UAB

Universitat Autònoma
de Barcelona

Contextual Word Spotting in Historical Handwritten Documents

A dissertation submitted by **David Fernández Mota** at Universitat Autònoma de Barcelona to fulfil the degree of **Doctor of Philosophy**.

Bellaterra, September 2014

Director	**Dr. Josep Lladós** Dept. Ciències de la Computació & Centre de Visió per Computador Universitat Autònoma de Barcelona
Co-director	**Dr. Alicia Fornés** Dept. Ciències de la Computació & Centre de Visió per Computador Universitat Autònoma de Barcelona
Thesis committee	**Dr. Basilis Gatos** Computational Intelligence Laboratory Inst. of Informatics and Telecom. & National Center for Scient. Research "Demokritos"
	Dr. Oriol Ramos Dept. Ciències de la Computació & Centre de Visió per Computador Universitat Autònoma de Barcelona
	Dr. Véronique Eglin Laboratoire d'InfoRmatique en Images et Systèmes d'information Université Claude Bernard Lyon
International evaluators	**Prof. Dr. Jean-Marc Ogier** Laboratoire Informatique, Image et Interaction Université de La Rochelle
	Dr. Andreas Fischer Department of Electrical Engineering Montreal Polytechnic

This document was typeset by the author using LaTeX 2$_{\varepsilon}$.

The research described in this book was carried out at the Computer Vision Center, Universitat Autònoma de Barcelona.

ISBN: 978-84-940530-7-8

Printed by Ediciones Gráficas Rey, S.L.

A mis padres, a mis hermanas y a Óscar

Agradecimientos

Hace alrededor de unos 10 años que hice mi Erasmus en Holanda. Yo llegué allí con toda la inexperiencia de alguien que nunca ha vivido fuera de su casa, sin saber que iba a ser de mí, y me encontré con un interesante proyecto de visión por computador. Fue allí donde se despertó mi curiosidad por este campo. Me fascinaba que un ordenador pudiera reconocer objetos en las imágenes. Años después, acabé en el Centro de Visión por Computador iniciando una tesis de análisis de documentos. Echando la vista atrás, es inevitable preguntarse si todo se originó en el momento que decidí irme de Erasmus.

En primer lugar quiero dar mi más sincero agradecimiento a mis directores de tesis, Josep Lladós y Alicia Fornés. Estoy convencido que sin ellos esta tesis no hubiera tenido el gran progreso que ha tenido. Les doy las gracias por todo el apoyo y los consejos que me han dado a lo largo de estos 5 años de tesis. Quiero agradecerles también la paciencia que han tenido conmigo a la hora de revisar mis artículos, y en especial esta tesis; y muy especialmente quiero agradecerselo a Josep porque cada día valoro mucho más el gran esfuerzo y dedicación que realiza para el buen funcionamiento del grupo de trabajo, y en general del CVC. Quiero darle mil gracias por, a pesar de las múltiples responsabilidades que tiene, siempre ha encontrado un hueco para guiarme, aconsejarme o simplemente dejarme ver el camino que muchas veces se distorsiona en momentos de estrés. Ha sido un gran placer trabajar contigo estos años y espero que sigamos trabajando juntos en esto de los documentos.

Siendo una tesis que ha sido desarrollado en un centro de investigación como es el CVC, no puedo dejar de agradecer a toda la gente que he conocido y me ha ayudado de alguna forma en la elaboración de esta tesis durante todos estos años. Agradecer al personal de informática y de administración porque sin ellos el centro dejaría de funcionar. En especial quiero agradecer a todos esos buenos compañeros y amigos que he conseguido tras elaborar esta tesis: Camp, Lluís, Jon, Ivet, Joan, Alejandro, Fran, Jorge, Toni, Carles, Anjan... Sin el apoyo de todos ellos no lo hubiera conseguido. Gracias por estar ahí.

No puedo olvidarme del DAG, ese grupo de “locos” que se dedican a analizar documentos. Gracias por todos los sabios consejos que me habéis ofrecido durante todo este tiempo. Han sido de gran ayuda y esta tesis lleva un poquito de todos vosotros. También quiero agradecer a la gente del Centro de Estudios Demográficos (CED) por la ayuda que han ofrecido para el acceso a los documentos que se han utilizado en está tesis. En especial quiero agradecer a Joana, Miquel y Ana el tiempo que han dedicado en este proyecto.

The research stay in USA have been important for me. I would like to thank Dr. R. Manmatha for his great supervision, guidance and advice during these 4 months. It has been a pleasure for me and honour to work with one of the pioneer of the word spotting.

Vorrei fare un ringraziamento speciale a Simone Marinai. Per i suoi consigli, per avermi guidato nel mio lavoro e anche per la sua amicizia. E’ stato un grande piacere lavorare con Voi e spero di continuare questa collaborazione in futuro.

Quiero agradecer a los amigos, que soportan tanto los buenos como los malos momentos.

Durante todos estos años he realizado muy buenos amigos. Considero que todos ellos, en cierto modo han influido en el desarrollo de la persona que hoy en día soy. En especial quiero agradecerles a la “secta” que sean como son. Me considero muy afortunado de tener el grupo de amigos que tengo. Durante todos estos años habéis logrado motivarme, apoyarme, alégrame, y darme el calor de esos amigos que siempre agradeces en momentos difíciles. Juan, Ángela, Anna Ballester, Mireia, Anna Bou, Alexandra, Cristina, Lidia, Nerea, Silvia, Raquel, Diego, Jose Luís, Ximo, Miguel Tomás, Miguel, Ismael, Toni, Sergio, Amalia, Vero... gracias por estar ahí!

Me gustaría acabar dando las gracias a mi familia, por el aprecio y apoyo de todos estos años, por sentiros cercanos, a pesar de la distancia. Agradecer a mis *hermanas* todo el apoyo que me han dado durante toda mi vida. Siempre han estado ahí en todas las decisiones que he tomado. A mi *padre* por su insistencia en mi formación. Sé que para ti es muy importante que haya llegado donde he llegado, y aprecio mucho esa incansable insistencia para que sea alguien en esta vida. No me puedo olvidar de mi *madre*, mi súper mamá. Cada día me sorprendes más y más por todo lo que haces por tus seres queridos, y en especial por tus hijos. Sé que el día que te dije que me venía a Barcelona, sentiste un poco de tristeza en tu corazón, pero aun así no dejaste que aflorará y no has dejado de darme todo tu apoyo y amor durante todos estos años. Quiero agradeceros todo lo que habéis hecho por mí y por ser tan comprensivos en ese momento de mi vida que todos ya sabéis. OS QUIERO!

Finalmente gracias a ti, por estar ahí, por ser mi punto de apoyo, por darme ánimos, por alegrarme el día y por quererme tanto o más como te quiero yo a ti.

Abstract

There are countless collections of historical documents in archives and libraries that contain plenty of valuable information for historians and researchers. The extraction of this information has become a central task among the Document Analysis researches and practitioners. There is an increasing interest to digital preserve and provide access to these kind of documents. But only the digitalization is not enough for the researchers. The extraction and/or indexation of information of this documents has had an increased interest among researchers. In many cases, and in particular in historical manuscripts, the full transcription of these documents is extremely difficult due the inherent deficiencies: poor physical preservation, different writing styles, obsolete languages, etc.

Word spotting has become a popular an efficient alternative to full transcription. It inherently involves a high level of degradation in the images. The search of words is holistically formulated as a visual search of a given query shape in a larger image, instead of recognising the input text and searching the query word with an ascii string comparison. But the performance of classical word spotting approaches depend on the degradation level of the images being unacceptable in many cases . In this thesis we have proposed a novel paradigm called contextual word spotting method that uses the contextual/semantic information to achieve acceptable results whereas classical word spotting does not reach.

The contextual word spotting framework proposed in this thesis is a segmentation-based word spotting approach, so an efficient word segmentation is needed. Historical handwritten documents present some common difficulties that can increase the difficulties the extraction of the words. We have proposed a line segmentation approach that formulates the problem as finding the central part path in the area between two consecutive lines. This is solved as a graph traversal problem. A path finding algorithm is used to find the optimal path in a graph, previously computed, between the text lines. Once the text lines are extracted, words are localized inside the text lines using a word segmentation technique from the state of the art.

Classical word spotting approaches can be improved using the contextual information of the documents. We have introduced a new framework, oriented to handwritten documents that present a highly structure, to extract information making use of context. The framework is an efficient tool for semi-automatic transcription that uses the contextual information to achieve better results than classical word spotting approaches. The contextual information is automatically discovered by recognizing repetitive structures and categorizing all the words according to semantic classes. The most frequent words in each semantic cluster are extracted and the same text is used to transcribe all them.

The experimental results achieved in this thesis outperform classical word spotting approaches demonstrating the suitability of the proposed ensemble architecture for spotting words in historical handwritten documents using contextual information.

Resumen

Existen incontables colecciones de documentos históricos en archivos y librerías repletos de valiosa información para historiadores e investigadores. La extracción de esta información se ha convertido en una de las principales tareas para investigadores del área de análisis de documentos. Hay un interés creciente en digitalizar, conservar y dar acceso a este tipo de documentos. Pero sólo la digitalización no es suficiente para los investigadores. La extracción y/o indexación de la información de estos documentos tiene un creciente interés entre los investigadores. En muchos casos, y en particular en documentos históricos, la completa trascripción de estos documentos es extremadamente difícil debido a dificultades intrínsecas: preservación física pobre, diferentes estilos de escritura, lenguajes obsoletos, etc.

La búsqueda de palabras se convierte en una popular y eficiente alternativa a la transcripción completa. Este método conlleva una inherente degradación de las imágenes. La búsqueda de palabras se formula holísticamente como una búsqueda visual de una forma dada en un conjunto grande de imágenes, en vez de reconocer el texto y buscar la palabra mediante la comparación de códigos ascii. Pero el rendimiento de los métodos de búsqueda de palabras clásicos puede verse afectado por el nivel de degradación de las imágenes, que en algunos casos pueden ser inaceptables. Por esta razón, proponemos una búsqueda de palabras contextual que utiliza la información contextual/semántica para obtener resultados donde los métodos de búsqueda clásica no lo logran un rendimiento aceptable.

El sistema de búsqueda de palabras contextual propuesto en esta tesis utiliza un método de búsqueda de palabras basado en segmentación, y por tanto es necesaria una segmentación de palabras precisa. Documentos históricos manuscritos presentan algunas dificultades que pueden dificultar la extracción de palabras. Proponemos un método de segmentación de palabras que formula el problema como la búsqueda del camino central en el área que hay entre dos líneas consecutivas. Esto se resuelve mediante un problema de grafo transversal. Un algoritmo de búsqueda de caminos es utilizado para encontrar el camino óptimo en el grafo, calculado previamente, entre dos líneas de texto. Una vez las líneas se han extraído, las palabras son localizadas dentro de las líneas de texto utilizando un método del estado del arte para segmentar palabras.

Los métodos de búsqueda clásicos pueden mejor utilizando la información contextual de los documentos. Presentamos un nuevo sistema, orientado a documentos manuscritos que presentan una estructura a los largo de sus páginas, para extraer la información utilizando información contextual. El sistema es una eficiente herramienta para la transcripción semiautomática que utiliza la información contextual para obtener mejores resultados que los métodos de búsqueda convencionales. La información contextual es descubierta automáticamente reconociendo estructuras repetitivas y categorizando las palabras con su correspondiente clase semántica. Se extraen las palabras más frecuentes de cada clase semántica y así el mismo texto es utilizado para transcribir todas ellas.

Los resultados experimentales obtenidos en esta tesis mejoran los resultados de los métodos clásicos de búsqueda de palabras, demostrando idoneidad de la arquitectura propuesta para la búsqueda de palabras en documentos históricos manuscritos utilizando la información contextual.

Contents

List of Figures

s

List of Tables

Chapter 1

Introduction

This chapter presents the motivation and objectives of this thesis. We briefly overview the Document Analysis and Recognition research field and, in particular, the analysis of historical documents. Afterwards, we review the word spotting architecture, and discuss about its applicability to handwritten documents. Following, we introduce the Five Century of Marriages project (5CofM), and show the characteristics of this collection of books. We introduce one of the main characteristic of this kind of documents: the contextual information, which can be used to extract the information of this kind of documents. Finally, we overview the main difficulties found in such task, and summarizes the objectives and contribution of this work.

1.1 Historical Handwritten Documents

Paper documents have been the best effective form of communication during the history, but there is an increasing interest to digitally preserve and provide access to historical document collections in libraries, museums and archives, due the advantages of computer systems in terms of storage, retrieval, transmission and automatic processing of documents. This kind of documents are valuable cultural heritage, as they provide insights into both tangible and intangible cultural aspects. Historical archives usually contain handwritten documents. Examples are manuscripts written by well known scientists, artists or writers; as well as letters, trade forms or administrative documents kept by parishes or councils that help to reconstruct historical sequences in a given place or time. While machine printed documents, under minimum quality conditions, are easy to be read by OCR systems, handwritten document recognition is still a scientific challenge.

The **Digital Humanities** are an area of research, teaching, and creation concerned with the intersection of computing and the disciplines of the humanities. The relation and interaction between the humanities discipline with digital resources and computing is the major and the best scientific and educational development. The new digital area, as the printing was in the last Humanism, gives the chance of developing, expand and disclose the knowledge. The digital humanities provides a huge variety of objects of study, from the design and preservation of digital collections to cultural data analysis.

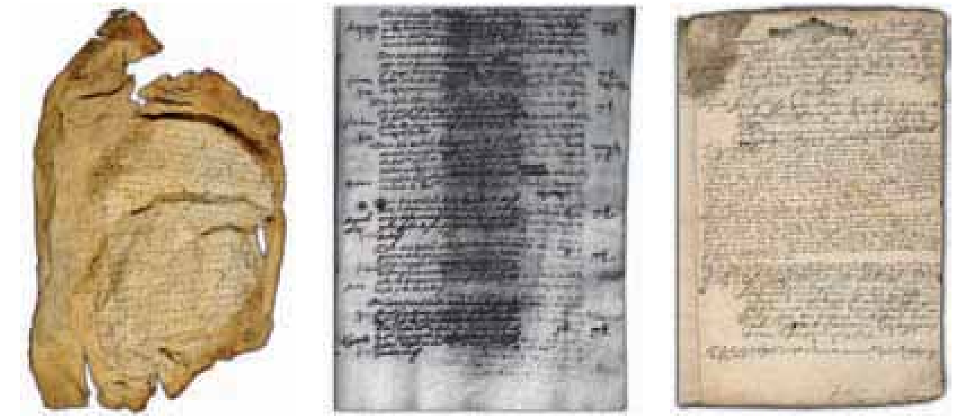
From the point of view of the humanities a historical document (or scientifically called historical source) is defined as any written or graphical material which provides information to reconstruct the past. Historical sources are diverse and they are classified as primary and secondary sources. A *primary source* is a source mentioning some new idea, creative thought, or data originating in that source, and not derived from another author or another source. A *secondary source* is any source who uses other sources to relate a fact. Secondary sources are not the originators of new ideas, creative thoughts, or data; they merely act as a conduit for such information. We are going to focus our classification in primary sources, which come directly from the past. This can be divided in written and non-written sources.

- *Non-written sources* comprise a big variety of objects whose objective were not to inform, but they can be used for the research. Artistic sources (as architecture, sculpture and paint), daily objects (as apparels, tools, instruments, etc.), photography, movies, records and oral fonts are part of this kind of sources.
- *Written sources* are documents, as the name suggests, that are written independently of the language or medium. This kind of documents can be handwritten or printed, and depending of their content are classified in five categories. The first one is *documentary source*. It consists in all the generated documents along the history. Most of these documents are stored in historic archives and can be classified as: politics (documents related with politic activity as speeches, manifests, agreements, etc.); legal (structured documents containing laws, orders, treaties, etc.); statistics (documents of records with a high repetitive structure of information among their pages, as parish books, census, civil records, counts, etc.); cartography (documents containing maps, floor plans, etc. that can be classified as non-written, but they usually contains text); numismatics (coins and bills). The second one is *periodicals*. Basically press and magazines. These documents are so recent, since the 18th century and until the 19th century they are not regular. The third category is *literary or scientific works*. They are works written in a specific period with a big variety of topics. Some examples are novels, poems, essays, etc. The fourth category consist in *memories and personal diary*. They are special documents because they are not usually adjusted to reality. The authors give their point of view of the reality. And finally, *letters*. The objective of these documents was the communication between people.

Among the written sources, we can establish a second level of classification dependently on the structure of the contents. Documents which present a repetitive structure along their pages, *fixed-structure*, and documents with a *free-structure*. The first ones usually are documents of records like census, civil records, etc. The second ones do not follow a structure and the stored information is diverse. In the first category we can include documents of the type legal or statistic. The rest of documents do not present a structure and are classified as free-structure.

The digitalization of large collections of documents is the first stage of the Document Image Analysis and Recognition (DIAR). This field of pattern recognition tries to cover the need of the humanistic people: improvement of the readability and searching, and to extract semantic information to the documents. The interest of DIAR, besides to the preservation on digital format, is the recognition and transcription of the document to a machine readable format. In fact, there are some difficulties related with the digitalization of the documents. The first one is the access to these historic documents. Although there is a huge amount of old documents in archives and churches all over the world, the access is allowed only to some experts historians, because of safety reasons (documents are very valuable), and also because of the delicate state of the paper (there is an important degree of paper degradation).

The second difficulties are related with the curation state of the documents. The physical lifetime degradation of the original documents, related to the frequent handling and careless storage, produces holes, spots, broken strokes, ink bleed, winkles, etc., or even accidental fires. Figure 1.1 shows some examples of degraded documents. The third type of difficulties are introduced in the scanning process. It might introduce difficulties such as non stationary noise due to illumination changes, show-through effect, low contrast, warping, etc.



(a) Status of *The Great Parchment Book* collection after a fire

(b) Microfilm of the volume 45 of *Llibre de Esposalles*

(c) *Archives Historiques de Châtillon-Sur-Chalaronne*

Figure 1.1: Some examples of degraded historical documents.

Handwriting recognition systems have been reported in the literature as capable of transcribing handwritten documents up to certain precision. Full transcription [50, 178, 216] is difficult however good performance can be achieved when searching individual words. Thus a strategy to search words in handwritten documents would be to apply a HWR system to that document and then to search the words in the output text. Another strategy is known as handwritten word spotting. It is defined as the pattern analysis task which consists in finding keywords in handwritten documents. The term of word spotting was defined in the early of the eighties. It has been used in speech recognition [95, 134, 162] to detect the presence of certain keywords in a message, instead of fully transcribe it. This methodology was applied in handwritten recognition first in 1996 by Manmatha et al. [122]. The important contribution of this work is that an image matching approach is sufficient for retrieving keywords in documents, without the need of recognizing the text. These techniques are used whena complete transcription is not required, or the existing technologies can hardly transcribe documents with poor quality. Rather than attempting to transcribe text to its full extend, only checking the existence of special keywords (names, dates, cities, etc.) may be enough.

Handwritten word spotting is one of the central topics in this thesis. In the following sections, handwritten word spotting is described with more details, together with some current difficulties and limitations which inspire the main contributions of the present thesis.

1.2 Handwritten word spotting

Handwriting recognition usually transcribes the entire documents taking into account the prior probabilities of the lexicon units of a given lexicon. In this framework, a large amount of data is required to train the probabilities and the language models. However, this such

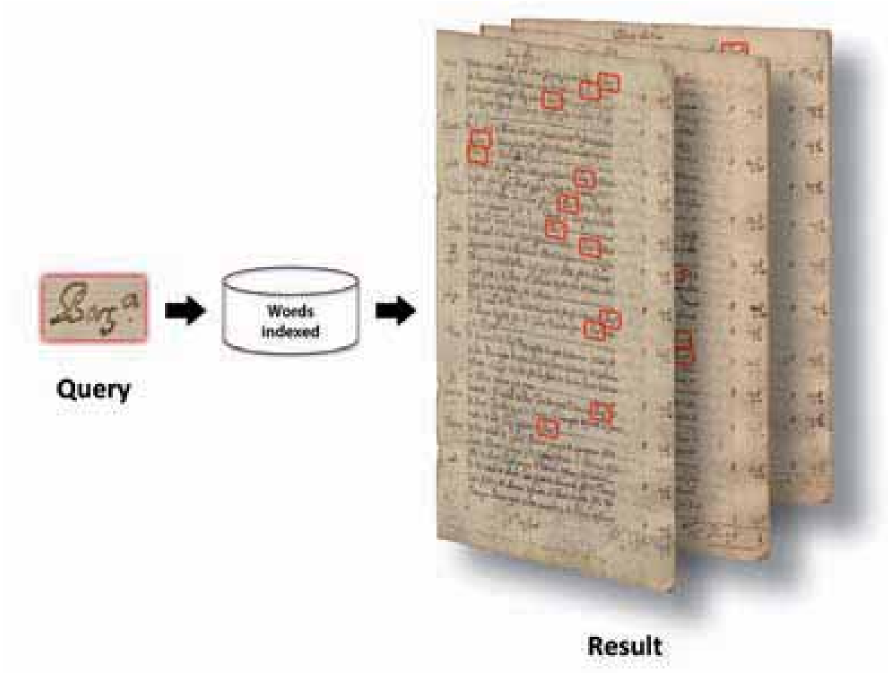


Figure 1.2: The components of a word spotting approach.

training data may not exist, or may require a tedious manual creation. Moreover, the information contained in a collection of manuscripts may not be useful to train recognizers for another collection because of different periods of time with different script styles, or different disciplines, with non-intersecting and restricted lexicons. Another handicap of these approaches is the cost in terms of computation and training set requirements.

Handwritten word spotting formulates the recognition of words as one of the pattern recognition problems. It is not necessary the transcription of the words because the search is done by similarity between the shapes. An image matching approach is sufficient for retrieving keywords in documents. The problem is then converted to a validation problem rather than a recognition problem, i.e. to validate whether a word image matches a given query with a high score. Several works have been developed using this methodologies applied to handwritten documents [1, 6, 59, 67, 96, 154, 159], industrial applications [19, 30] or even forensics [135].

The components of a word spotting approach are (i) a collection of documents or database (words indexed) and (ii) an input element denoted as query (see Figure 1.2). The output (result) of a word spotting approach should be the localization in the collection of documents (or sub-images of this collection) that are similar to the query. Two main types of word spotting approaches exist depending of the representation of the input query:

- Query-By-String (QBS): the input is a text string [59, 67]. Character models are learned off-line and at runtime the character models are combined to form words and the probability of each word is evaluated [27, 49, 30].
- Query-By-Example (QBE): the input is an image of the word to search and the output is a set of the most representative (sub)images in the database containing a similar

word shape [122, 151, 193].

The QBS has the advantage of flexibility to search any kind of keyword. However, labelled datasets are required in order to train the recognition engine. At the other hand, QBE methods can achieve sufficient accuracy to be useful in a practical scenario and it does not require learning (it only requires collecting one or several examples of the keyword). As Manmatha et al. discuss in their work [122], these methods are mostly based on image matching. These methods are worth of attention when labelled training data is not available or would be too expensive to collect. Therefore, the best option depends on the application.

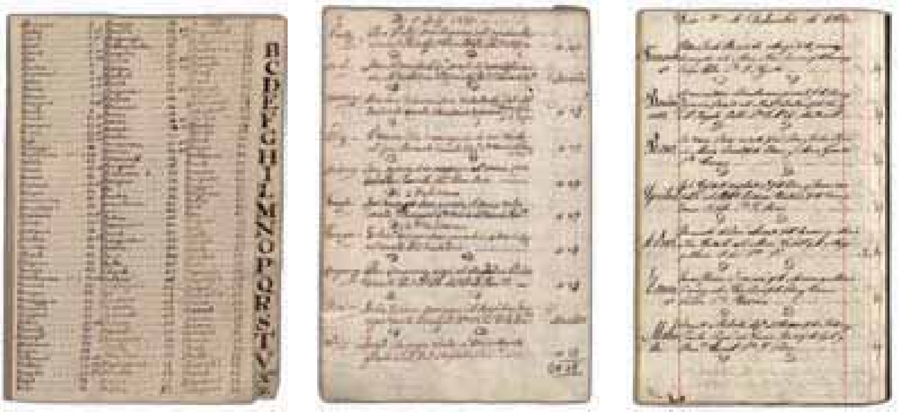
In word spotting a key decision is the feature representation. In the literature there are two methodologies to extract the features: segmentation-based and segmentation-free approaches. *Segmentation-based* approaches [151] require each document image to be segmented at word level, taking advantage of the knowledge of the structure of the document. The main problem of these approaches is that word classification is strongly influenced by over or under-segmentations. *Segmentation-free* approaches [69, 126, 169] are not affected by this problem. The image is divided into patches (e.g. using a sliding window) and the query word image is classified regarding each patch. Since all the regions are compared, these methods are computationally costly in terms of time. Therefore, the best option depends on the application. In the next section, the application of interest of this thesis is presented and the choice of the approach is justified.

1.3 Motivation

1.3.1 The project Five Centuries of Marriages (5CofM)

On September 27, 1409, Pope Benedict XIII (Pedro Martínez de Luna), visited Barcelona and granted the new Cathedral a tax on marriage licenses (*esposalles*) to be raised on every union celebrated in the Diocese. This tax was maintained until the third decade of the 20th century. Between 1451 and 1905, a centralized register, called *Llibres d'Esposalles* recorded all the marriages and the fees posed on them according to their social class, on an eight-tiered scale. This exceptional documentary treasure, conserved at the Archives of the Barcelona Cathedral, comprises 291 books with information on approximately 600.000 marriages celebrated in over 250 parishes, ranging from the most urban core of the city to the most rural villages in the periphery of the Diocese. Their impeccable conservation for over four and a half centuries is a miracle in a region where parish archives have suffered massive destruction through several episodes along the last 200 years. Nothing similar is known to exist anywhere else. The Figure 1.3 shows the formal continuity of the source along time.

Each book contains the marriage of two years, and each book was written by a different author. Each book of the collection is split in two parts. The first one is an index with all the husbands' surnames that appear in the volume and the page number where it appears (see Figure 1.3a). The indexes of the books have the same structure: several columns, where each column is composed by a surname, several dots and the number of page where this surname appears. The second part consist of the marriage licences (see Figures 1.3b and 1.3c). The layout of the marriage licences consists of separate registers, each in one paragraph. Each register is divided in three parts. In the left part we can find the husband's surname. Each surname is next to the record of the wedding. In the right part we can find the tax of the wedding, which depends on the social status of the couple. The central part corresponds to the record. The structure of each of these registers is comprised by certain characteristics, such as the date of the wedding, the groom's name, the job and origin, the groom's parents'



(a) 1617: index of volume 69 (b) 1726: volume 127 (c) 1860: volume 200

Figure 1.3: *Llibre d'Esposalles* (Archive of Barcelona Cathedral, ACB).

names, the bride's name, her father's name, and her home-town. We can see an example in Figure 1.4.

The *Five Centuries of Marriages (5CofM)* project[1] is a long-term research initiative based in the data mining of this unique documentary source. Bringing together social and computer scientist. The core of this project is the construction of a database built with documents which have never been the object of scientific use or study. The proposal of this project promotes a joint scientific contribution, where technology meets history and social science: the *Center of Demographic Studies (CED)* brings the experience in historical population studies, and the *Computer Vision Center (CVC)* provides the technological transfer related to the field of Computer Vision, in particular Document Image Analysis System.

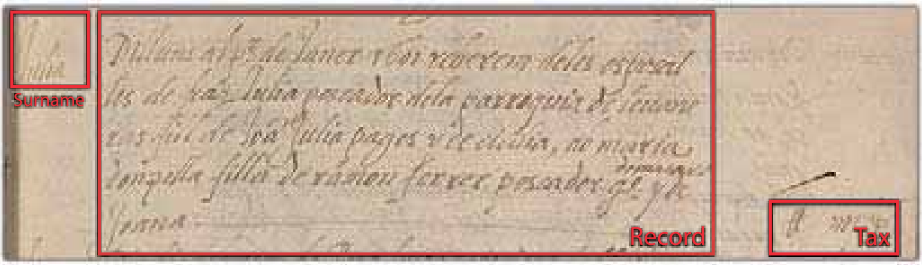
Some published works shows the relation between the social and the computer scientists and the good progress of the project. The migration of the population is studied [25] through the centuries in Barcelona and surrounding. The international transmission of social status is the focus of the work of Pujadas et al. [149], or the mortality along the years is the study of Villavicencia et al. [198]. The computer vision has collaborated in this project providing word spotting [5, 140], handwriting recognition [32, 67], layout segmentation [7, 35] and image enhancement approaches [63].

1.3.2 Contextual Handwritten Word Spotting

Word spotting has become a popular and efficient strategy in the access by content to historical manuscriptswhen explicit recognition is not possible. Due to the quality of physical preservation, the writing styles, and the obsolete languages, the full transcription of such documents is extremely difficult. In many applications, once the documents are digitized for preservation purposes, search contents-wise is the main purpose. Here is when the use of object retrieval approaches using visual features gains relevance.

This thesis provides the tools to extract the information from the documents of the *5CofM* project, and could be extended to handwritten documents which similar characteristics. These particular characteristics are a similar structure/layout along all the pages of the

[1] http://dag.cvc.uab.es/infoesposalles/


Surname
Record
Tax

Literal trasncription: Dilluns al pr(imer) de Janer 1601 reberem de les esposal / les de fran(cesc) Julia pescador de la parroquia de levane / ras fill de Joa(n) Julia pages y de cicilia, a(m)b maria / donzella filla de ramon ferrer pescador q(uondam) de mataro y de / Joana
4 sous

English literal translation: Monday the first of January 1601 received the marriage license fee / for francesc Julia fisherman from the levaneres parish / son of Joan Julia peasant and cicilia with Maria / maiden daughter of ramon ferrer fisherman deceased de mataro and / Joana
4 sous

Figure 1.4: Layout of the marriage licences and literal transcription of an example of the volume 60 (1601).

volumes, repetitive records containing wedding licences and semantic/contextual information which relates the words insides the records.

The use of context can significantly improve the recognition of individual objects. In computer vision, it is an emerging trend [29]. Usually word spotting is built based solely on the statistics of local terms. The use of correlated semantic labels between codewords adds more discriminability in the process. Three levels of context can be defined in a word spotting scenario. First, the joint occurrence of words in a given image segment. Second, the geometric context involving a language model regarding the relative 1D or 2D position of objects. Third, the semantic context defined by the topic of the document. A number of document collections convey an underlying structure. This structure is natural in records describing demographic events such as census, birth, marriage, or death records. This structure is characterized by a page arranged in records (paragraphs) or tables. In a finer level each unit (record) uses to follow a syntactic structure. The analysis of the contents in such documents can not be solved by raw transcription, but word spotting is a good alternative for record linkage (linking names for genealogical analysis) or search of people, places, and events.

The main hypothesis of the thesis is that the use of context boosts the recognition. In historical demographic documents, some words have high probability of co-occurrence. For example, if we have genealogy linkage, we can learn joint probabilities between family names, some common words in the record like ”married to” determine the position of the searched ones, migration movements from geographic areas also generate clusters of family names that can be linked to city names, etc.

Historians use the three levels of context to understand and extract the information of the documents. Firstly, some words are contextually related and the probability of apparition is higher in presence of other related words. The words ”married to”, showed in the above example, limit the words that appears around them. Secondly, in highly structured documents, some words always appear in the same position. This information can be used to find some words close to the fixed ones. And finally, historians uses the semantic knowledge to find information. For example, the interpretation of words will be driven by the topic of the document. If the document is about marriages licences, words as ”husband”, ”wife”, ”parish”, etc. are likely to appear.

The contribution of this thesis is to find words into the documents using the contextual information, like the historians do. The joint concurrence, the geometric context and the semantic context are used to boost the results obtained by the word spotting approaches of the state of the art.

1.4 Thesis Objectives

The general objective of this thesis is to perform research around the problem of word spotting in highly structured historical handwritten documents. More particularly, the objective is twofold: from the scientific point of view and from an application point of view.

From the **scientific** point of view, the goal is to study the existing word spotting approaches oriented to handwritten documents, and outperform these methodologies using the contextual information of the documents. This objective, it can divided in the following sub-objectives:

- The first objective focuses on the layout segmentation of historical handwritten documents. This thesis is developed using a segmentation-based word spotting approach and efficient word segmentation is required. It covers the study of the methods oriented to recent and historical documents. The study includes the evaluation of the existing techniques to improve and restore damaged documents.
- The second objective is centred in the study and development of a descriptor for handwritten documents. A word spotting approach requires the use of a descriptor to match the words to find. A good descriptor should guaranty intra-class compactness and inter-class separability. It should be tolerant to noise, degradation, occlusions, distortion and elastic deformations typically found in handwritten documents.
- The third objective is to improve the existing techniques of word spotting using contextual information. The approach proposed is an unsupervised structural word spotting, which analyses the documents and extracts the structural information of them. The approach recognizes repetitive structures in the documents and categorizes all the words to, afterwards, to spot words into the classes. It must include the revision of the methods of word spotting of the literature, and study of the different methods to align text and the revision of the state of the art in syntactic analysis.
- As additional objective, is the construction of a framework for validating the proposed methodology for structural handwritten documents. There is a lack of public database where the words have semantic and structural relationship.

From the **application** point of view, the goal is to develop a whole system for structural handwritten word spotting that achieves good results in a real-world problem. As we have mentioned before this thesis has been in the framework of the *5CofM* project, and the main objective is to develop a system that aims the process of extracting information from the collection of books *Llibre d’Esposalles*.

1.5 Thesis Contributions

With the aim to accomplishing the objectives related above, in this thesis has been done the contributions explained below. We identify three main contributions aligned with the key components of a word spotting system, namely layout segmentation, feature extraction and

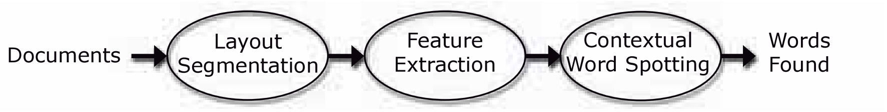


Figure 1.5: Blocks of the thesis.

retrieval (see Figure 1.5). The first block includes the components that segment the documents in its constituent parts. The second block is oriented to the study and development of an efficient descriptor adapted to historical handwritten documents. And the last block constitute the methodologies developed to search information using contextual information.

1. A method for layout segmentation in historical handwritten documents.

   An accurate line segmentation improves the word segmentation in handwritten documents, and it has been proved with the comparative study done using an accurate line segmentation and projection-based line segmentation. The contribution is the proposal of a line segmentation approach which tackles with the main problems in handwritten documents: touching components, curvilinear text lines and horizontally-overlapping components. Once the lines are extracted, we have applied the method developed by Manmatha and Rothfeder [123] to segment the words.

2. A descriptor to characterize handwritten word images.

   After the review of the state of the art methods for word spotting in handwritten documents, we have developed a pseudo-structural descriptor for handwritten word images.

   - The study shows the influence of the selection of key points and the associated features in the performance of word spotting processes. In general, features can be extracted from a number of characteristic points like corners, contours, skeletons, maxima, minima, crossings, etc. A number of descriptors exist in the literature using different interest point detectors. But the intrinsic variability of handwriting varies strongly on the performance if the interest points are not stable enough. As result we show that the performance of a handwritten word spotting approach does not only rely on the descriptor but also on the key point detection method.
   - The descriptor proposed is a pseudo-structural descriptor organized in hash structure. It is a modification and extension of the LOCI descriptor [70], which is oriented to characters, to handwritten word images. This descriptor encodes the frequency of intersection counts for a given key-point in different direction paths starting from this point. Once word images are encoded using a Loci-based descriptor, the indexation structure is organized as a hashing-like way where features are encoded as index keys and words are stored in a hashing structure. As a result, a robust descriptor in front of noise and elastic deformations is obtained.

3. A context-aware word spotting approach.

   The main contribution consist in the proposal of the following methods for extracting the repetitive structure, classify and to do contextual word spotting of the documents automatically.

   - Unsupervised approach to identify repetitive structures based in the frequency and position of words. The target documents of this thesis are documents which presents a repetitive structure along all the pages (e.g. administrative documents, licences, records, etc.). There are some words that always appear in the same positions along all the repetitions. These words are considered as *key-words* and they will be the anchor points to classify the words by classes. As a contribution, a method for detecting the *key-words* is proposed. The method is based the Longest Common Subsequence (LCS) algorithm [120] to determine the order of the *key-words*.

- Aligning word images. Once the keywords are detected, a method for searching them into the documents is proposed. Key-words are a sorted list of words that always appear in the same order. The method consists in using the advantage of the prior known order of appearance of the query words. Given an ordered sequence of query word instances, the proposed approach performs a sequence alignment with the words in the target collection.
- Structural word spotting. The key-words establish structural information layer in the documents. All the words between two key-words are classified as part of the same class, because these words are semantically related. The main contribution associated to this goal consists in the proposal of two methods to spot words in handwritten documents using the structural information. The first method allows finding words in the different semantic segments using a classical word spotting architecture. The second approach uses Markov Logic Networks (MLN) [155] to recover true positives and remove false positives from the output of the word spotting approach.
  - Contextual word spotting. The main contribution is a system of word spotting to find words semantically related. The search of words is done using a classical word spotting approach inside each semantic segment.
  - MLN-based parsing. The last contribution complements the previous one. The system is improved using the semantic information using the weighted grammars and neural networks. The method relies on Markov Logic Networks to probabilistically model the relational organization of handwritten records.

The work developed in this contributions opens a field of structural searches. A word can be found using implicit information, like the class it belongs to. E.g. the search of the name of a man marriage record is focussed in a the husband information segment of the record. The relation between words boosts the structural searches. E.g. search two specific words that are in the same record, license, etc., implies a contextual spotting of words. A great range of possibility is opened to do more sophisticated searches according to the needs of the user.

4. Construction of a database for the evaluation framework.

The main contribution is an image database of historical handwritten marriages records stored in the archives of Barcelona cathedral, and the corresponding meta-data addressed to evaluate the performance of document analysis algorithms. The contribution of this block is twofold:

- First, it presents a complete ground truth which covers the whole pipeline of handwriting recognition research, from layout analysis to recognition and understanding.
- Second, it is the first dataset in the emerging area of genealogical document analysis, where documents are manuscripts pseudo-structured with specific lexicons and the interest is beyond pure transcriptions but context dependent.

1.6 Thesis outline

The present dissertation is structured in order to respect the aforementioned blocks of contributions, with a preceding chapter used to introduce the necessary context (related work and system overview), and a final chapter of conclusions. The particular structure is as follows:

- The state of the art of layout analysis and word spotting in handwritten historical documents is review in Chapter 2. First, the main layout analysis methods are overview. Following, we make a classification of the word spotting methods of the literature. The classification is done by learning-based and learning free methods as main categories. Finally, a summary of the chapter is done
- The layout segmentation in handwritten documents is defined in Chapter 3. The proposed algorithm formulates line segmentation as finding the central path in the area between two consecutive lines. This is solved as a graph traversal problem. A graph is constructed using the skeleton of the image. Then, a path-finding algorithm is used to find the optimum path between text lines. Afterwards, a word segmentation technique of the state of the art is applied to localize the words inside the text lines.
- In the Chapter 4 is evaluated the word spotting approaches of the state of the art and showed the importance of choosing the proper descriptor and the key points. We show the performance using different configurations of key points. And finally, we propose a new pseudo-structural descriptor based in local features oriented to historical handwritten documents.
- The contextual word spotting framework is defined in the Chapter 5. The framework model is composed by two levels. The first one is oriented to the knowledge discovery. The contextual/semantic information is extracted automatically, and then, the extraction of the information is performed using the contextual information in the second layer.
- The use of the Markov Logic Networks to improve the results of the word spotting approaches is presented in the Chapter 6. The use of grammars in highly structures documents is used to improve the performance of the word-spotting approach.
- The Chapter 7 presents a new database of historical handwritten documents. The database allows validating from layout analysis to recognition and understanding. An image database of historical handwritten marriages records has been created from the archives of Barcelona cathedral, and the corresponding meta-data addressed to evaluate the performance of document analysis algorithms.
- Conclusions are presented in the Chapter 8. Firstly a summary of the main contributions is performed. Afterwards, we discuss about the proposed approaches and their corresponding experimental results. Finally, future work is esposed.

Chapter 2

State of the art

In this chapter, firstly, the main layout analysis methods are overview. Following, we make a classification of the word spotting methods of the literature. The classification is done by learning-based and learning-free methods as main categories. Finally, a summary of the chapter is done.

The aim of a handwritten word-spotting system is to detect keywords in document images. This problem can be contextualized in the field of computer vision as well as in the field of pattern recognition, more specificity, the sub-field of document analysis.

The topic of handwritten text analysis has been present in pattern recognition for decades. This thesis is focussed in the problem of word spotting and in the influence of the context information in the improvement of word spotting. In the following, we review the standard techniques used in handwritten word spotting and the classical architecture of a word spotting approach. Since we work on segmentation based word spotting, first, a brief study of the different layout analysis techniques is provided. Then the topic of word spotting is analysed more in depth, and the more relevant literature is reviewed. It is assumed that the reader is familiar with the basic concepts of image processing, pattern recognition and statistics.

2.1 Layout analysis

Commonly, the input to any document analysis system is an image of a whole page, and therefore the first usual step is a segmentation of this document image into its constituent parts: paragraphs, lines and words. Word segmentation is normally a chained process consisting of a line segmentation and then, a word segmentation.

Here we introduce the most important of the layout analysis techniques in the literature. In Chapter 3 we have a done an exhaustive review of the most relevant techniques for the present thesis.

Line segmentation

One of the most basic techniques for text line segmentation is the method of *projection profiles* [187]. This method assumes that the lines of the document have been written

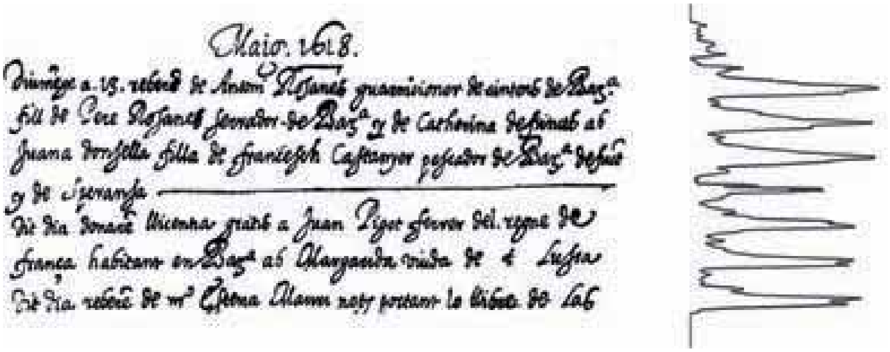


Figure 2.1: Horizontal projection profile of a cropped region.

horizontally. If I_{ij} denotes the image pixel with coordinates (i, j), then the projection profile is the sum of pixel values along each row of the image:

$$P_i = \sum_{j=1}^{w} I_{ij}, \tag{2.1}$$

where w denotes the width of the image. Lines are split at the rows for which P_i is a local minimum, since they are likely to correspond to line gaps. This idea is illustrated in Figure 2.1.

Since the method of profiles was originally proposed for line segmentation in typed text documents [187], it is expected to work only in very clean handwriting conditions. Therefore, its accuracy is likely to decrease in more realistic scenarios, such as skewed text, touching and overlapping lines, complex layout, curved lines, etc. For this reason some improvements have been proposed. Marti and Bunke [129] use the same principle as in Equation 2.11 but summing black/white transitions instead of pixels, to be more robust to writing fragmentation. Some works, for instance the one by Takru and Leedham [192] or by Zahour et al. [212] propose improved variants that are robust against touching and overlapping lines. Piecewise projection profiles can be combined to be robust to curved lines as well, as in the work of Arivazhagan et al. [13].

However, in general an explosive number of line segmentation techniques exist, probably too numerous to discuss here. The reader is referred to the Chapter 3 for a comprehensive collection of text line segmentation techniques.

Word segmentation

Once the lines have been segmented, a word segmentation step is performed to separate the text line into its constituent words. This is usually achieved by computing connected components and then by evaluating the distances between these connected components, for instance by means of the convex hull distance [119]. Because words are normally formed by one or more connected components, the gap distances above a threshold can be interpreted as gaps between words, and distances below the threshold as gaps between characters. The threshold can be chosen either empirically or as the output of a classification or clustering algorithm, such as k-means [46] or Otsu's thresholding method [142].

Improvements for this approach include supervised segmentation where the input consists of features computed from the word gaps (such as gap distance, angle, etc.) and the output is a set of two classes, corresponding to inter-word or inter-character gaps [188]. Although this method can be more accurate, of course a labelled dataset is required for learning the classifier. Scale-space techniques [123] have also been used for improved segmentation.

Finally, there are some applications which may require a more sophisticated segmentation strategy, such as transcript matching [195] (i.e. aligning text line images with their textual transcriptions). In such a scenario, word segmentation is of critical importance and a separation error may have a severe impact in subsequent steps. Therefore, that work proposes to produce multiple segmentation hypotheses, expecting that the global optimization method will automatically select and discard the necessary ones. This is reminiscent of the early approaches of word recognition using multiple segmentation hypotheses combined with dynamic programming [21].

2.2 Word spotting

Spotting is the task of locating a particular element without explicitly recognizing the content. The objective is, given an sample element, to find similar elements among a collection of elements, regardless the content or meaning of the elements.

Extrapolating this topic to the field of textual documents, we refer to word spotting. It is the task of locating a particular keyword without explicitly transcribing the content. This methodology is useful when the documents are not transcribed. It is an alternative to the entire transcription of the documents, which can be a hard task in historical handwritten documents due the difficulties that usually present these documents (lifetime degradation, holes, show-throw, etc.).

Formally, handwritten word spotting is defined as the pattern analysis task that consists in finding keywords in handwritten document images. The term of "word spotting" was introduced within the speech recognition field in the 70s. The original idea was to find a pattern in a continuous one-dimensional signal. It was an unconstrained system where the words are not segmented and there is not grammar to be considered in the sentence. The first approaches used different methodologies to reach this objective. Dynamic programming algorithms have become increasingly popular in automatic speech recognition [134, 138, 173]. There are two reasons why this has occurred: First, the dynamic programming strategy can be combined with a very efficient and practical pruning strategy so that very large search spaces can be handled. Second, the dynamic programming strategy has turned out to be extremely flexible in adapting to new requirements.

Hidden Markov Model (HMM) is used to retrieve spoken messages in several works [95, 162]. The idea is to make a parallel network of keyword and background filler models and use them to find the parts of the continuous signal similar to the input template. The Viterbi algorithm is a dynamic programming algorithm for finding the most likely sequence of hidden states that results in a sequence of observed events [113, 183]. The algorithm finds the best path maximizing the given probabilities.

At the early 90s, this methodology was applied to typewritten documents. The idea is to apply the same concept in a two-dimensional signal, which can be easily segmented into words. One of the first works [28] segmented the words using morphological operations and the feature vectors were computed using the external shapes and the internal structures of the words. Afterwards, keyword spotting was performed using a Viterbi search through the HMM network created by concatenating the keyword and non-keyword HMMs in parallel.

In handwritten documents, the concept of word spotting was introduced by Manmatha et al. [122] in 1996. The words of several documents are indexed doing clusters of classes. The matching is done using two different algorithms: Euclidean Distance Mapping and the

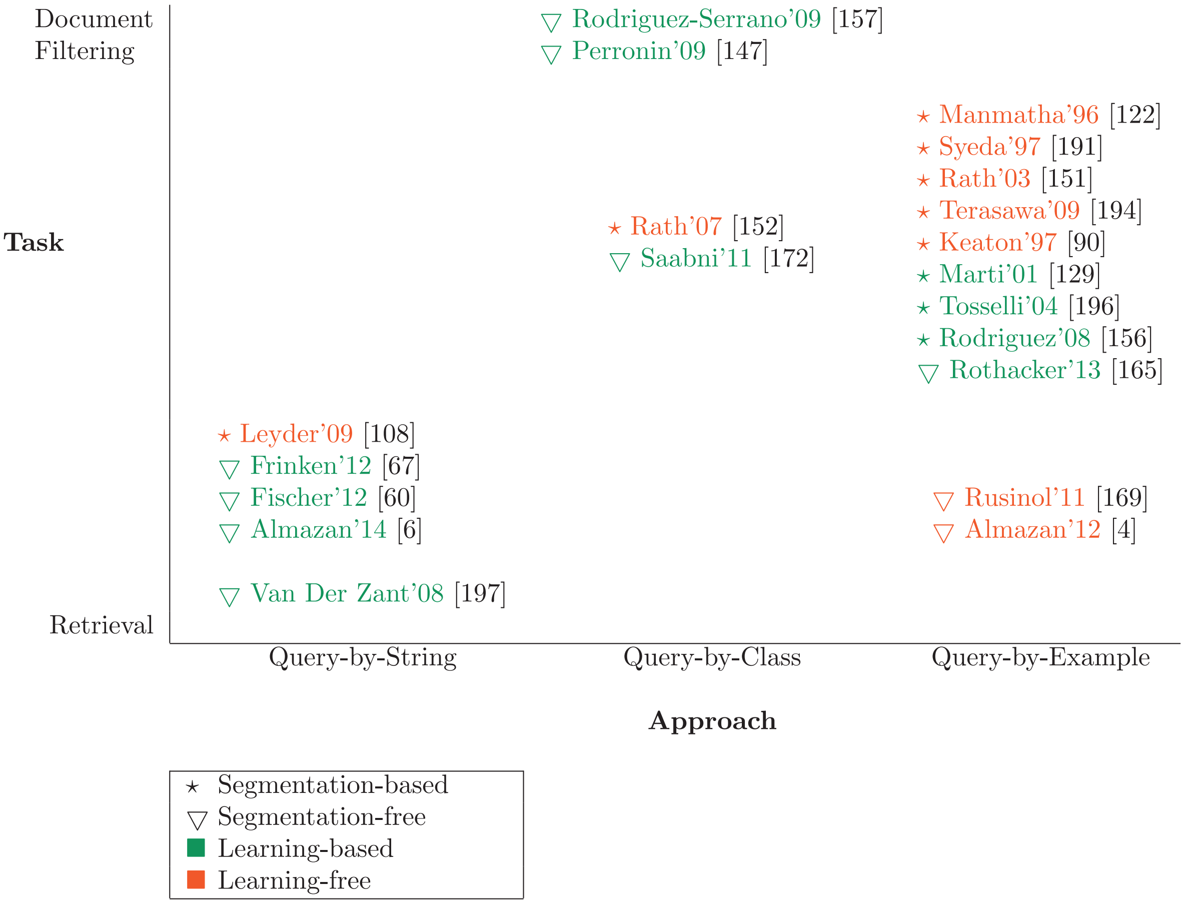
SLH algorithm. The indexation is done using the first n samples of each class.

During the last decades numerous approaches have been developed. All of them have the same objective: find keywords in handwritten documents. Depending on the point of view, they can be classified in different ways:

- Some methods were designed for *retrieval* purposes, while others were designed for *classification* purposes. The difference between them is that the methods that do retrieval find instances of the input keywords, nevertheless filtering purposes detect the presence of several keywords to identify a document as part of one class or another.
- Some of the methods in the literature need a *learning* process to compute a representative model. These methods usually need big quantities of labelled documents to make the model that will be used in the classification step. However, *learning-free* methods do not need labelled documents. These methods extract the features of the words and they do the classification computing a similitude measure between them.
- The methods that need segmented words are classified as *segmentation-based*. These methods need to analyse the layout the documents to detect and segment the words. The drawback of these methods is that the performance is influenced by the accuracy of the segmentation, but the advantage is that they usually are fast and easy of compute. In the other hand, there are methods *segmentation-free* that do not need to segment the words. The words are usually located using a sliding window that goes through all the document.
- Finally, the methods can be classified depending on the format of the input. If the input query is a word image, the methods are classified as *Query-by-Example (QBE)*. But if the input is a string, they are classify as *Query-by-String (QBS)*. There is a third way to classify the methods is *Query-by-Class (QBC)*. In this case a class is used as input of the method. Previously a classification of the words is done and several samples of the same words are used to find similar words in the documents.

The most relevant approaches of handwritten words spotting developed during the last decades are reviewed next. The study classifies the methods depending whether they need a previous learning process or not. Inside of each category, the methods are studied in deep and classified using the other categories showed above. Table 2.1 shows a summary of the most relevant methods in the literature.

Table 2.1: Classification of the most relevant methods



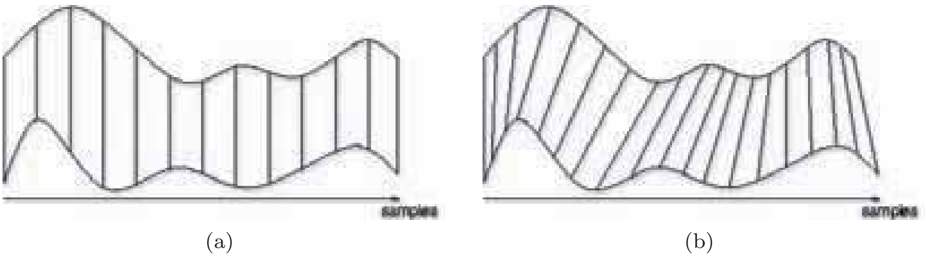


Figure 2.2: Alignment of two time series using linear stretching and Dynamic Time Warping (extracted from [151]): (a) linearly stretched, with 1-1 comparison and (b) non-linear alignment with DTW.

2.2.1 Learning-free methods

The classical architecture of the learning-free methods has three steps. First a layout analysis is applied to detect the text and segment the words. Then, for each segmented word, the feature vector is computed, and finally, the similarity between the input query and the corpus is computed. Manmatha et al. [122] were pioneers proposing an approach based in this architecture. From the binarized word images the matching is done in two phases. First, the number of words to ve matched is pruned using the areas and aspect ratios of the word images. Next, the actual matching is done using a matching algorithm. Two different matching algorithms are tried here. One of them only accounts for translation shifts, while the other accounts for affine matchings. Syeda et al. [191] adapted the principle of geometric hashing to localize the queries in handwritten images. The descriptor uses corner features on the curves to compute the feature vector. Keaton et al [90] use the profile and the cavities of the words to compute the feature vector. A probabilistic graph matching based on Bayesian evidential reasoning is used to find the best match between the keywords.

Although several works were developed in the next years, the break-through reference paper was the work developed by Rath and Manmatha [151]. The words are segmented using projection-based methods and the sequential features are computed using the upper and lower profiles, the number of foreground pixels, and the number of transitions black/white in each column of the word. The similarity of the words is computed using a Dynamic Time Warping-based (DTW) approach. The main advantage is that DTW computes the distance between two series optimizing the alignment, and it can distort (warp) the time axis, compressing or expanding when necessary (Figure 2.2).

Rath and Manmatha extended their word spotting approach with the clustering methodology [152]. The feature vectors computed from the word images are clustered using the k-means algorithm. Similar words are clustered into the same group speeding up the transcription of documents. The user transcribes only one word in each cluster and all the words in this cluster are automatically transcribed.

The main advantage of the work described above is the robustness in front of deformations, allowing some variety in the handwritten style. But the disadvantages are that the performance depends on a good word segmentation and the complexity of the problem: $O(n^2)$. All distances between words have to be computed.

Some of the works of the literature try to increase the performance removing the layout analysis step (or partially remove it). Terasawa et al. [194] extract the features from the text line. A sliding window is used to go through to the text line and compute the features. The descriptor used is the slit style HOG feature, which is a gradient-distribution-based feature

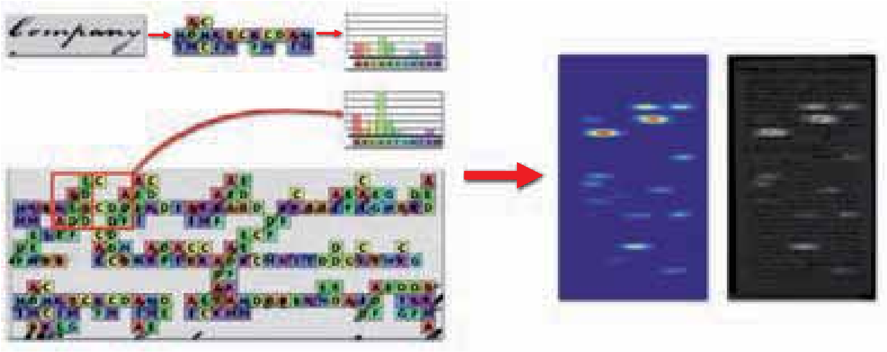


Figure 2.3: Architecture of the work developed by Rusiñol et al. [169].

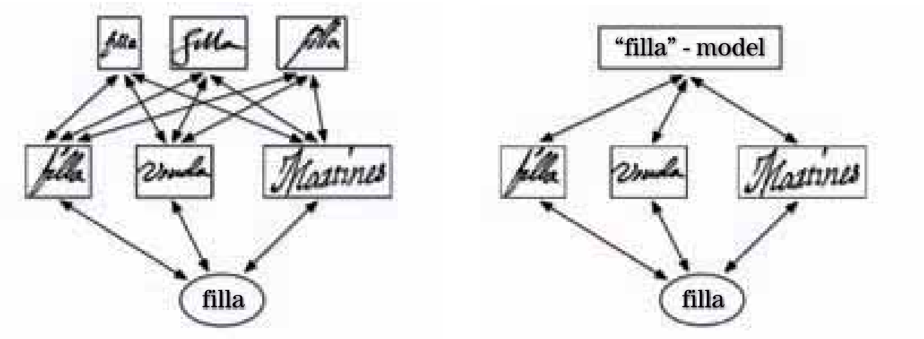
with overlapping normalization and redundant expression. The similarity is computed using a DTW-based algorithm.

But even when the word segmentation step is avoided (such as in the last described approach), the performance still depends on the good text line segmentation. Some works avoid the layout analysis by applying a sliding window over all the document. Rusiñol et al. [169] apply the Bag of Visual Words (BoVW) methodology in handwritten documents. The features are computed using a sliding window that goes through all the document. A SIFT-based descriptor is used at three different scales, and the low-gradient features are removed. The codebook is computed using the k-means algorithm. The test documents are split into overlapping local patches. The matching is computed using a voting scheme of probabilities (Figure 2.3).

Almazán et al. [4] use the same architecture to match the words. The method is an unsupervised approach geared to large databases. Documents and query images are divided in equal-sized cells and represented with HOGs. The score of a region is calculated using a sliding window that computes the convolution with the query.

The methods reviewed above are query-by-example approaches that use as input query a word image. The main problem of these approaches is that results depend on the input query, they do not handle different handwritten styles and spelling alternatives. But the advantage is that they do not need prior knowledge. Nevertheless, query-by-string approaches use text queries as inputs (from the keyboard), can handle different handwriting styles and spelling alternatives. But, they need to build a writing model. Leydier et al. [108] build a model for each letter in the alphabet and provide information about the linking behaviour. When the user types the query word, the approach generates different spelling variations and links the letters according to the rules. The feature vector are computed using gradients and ZOI locations. The matching is done searching the lowest score path in a pre-computed tree.

After the review done in this section, we conclude that the learning-free methods are a good alternative for spotting words in handwritten documents because training data is not required, they are independent of to the alphabet and language, and they are a fast image retrieval case if indexing-hashing techniques are applied. But the main disadvantages are that they do not handle with multiple writers and the query-by-string methodology is difficult to apply.



(a) (b)

Figure 2.4: Comparison of the input query with the documents: (a) against the templates and (b) against the model.

2.2.2 Learning-based methods

The learning-based methods belong to different paradigm than the learning-free methods. They create a character model of characters using training data and the comparison is done against the model and not against the templates (Figure 2.4). Saabni et al. [172] use Active Shape Models for Dynamic Time Warping (ActiveDTW). A word class is modelled by abstract parameters instead of samples. The priori probability of the query is estimated finding the best path.

The advantages of the learning-based methods are: higher precision and recall rates, they are flexible systems, can be used for document filtering and database search, and they allow the inclusion of more data (e.g. language models) to further increase the recognition. However they need a large training set and the fixed costs for training the system have to be taken into account, whenever new writing styles appear.

These approaches usually focus the extraction of the features on the entire text line, because text lines can be easily segmented (more than words). But, the text lines have different lengths and it is not known how many words exist in that text line and where their boundaries are. So, the objective is to transform a text line into a sequence and to use methods for sequential data analysis. Although the learning-based methods use text lines, there are also some methods in the literature that use words instead of text lines. The extraction and representation of the features are similar to text line methods. Next, we explain how the features are extracted and we show some examples.

The feature extraction is done using a sliding window that extracts features at each position. Feature vectors are concatenated into a sequence. A wide variety of features exist in the literature. Marti et al. [129] use geometric features. A combination of six local and three global features. The position and orientation of upper-most pixel and bottom-most pixel, the three first momentums, the number of transitions and the fraction of black pixels between highest and lowest black pixel (Figure 2.5a). Tosselli et al. [196] use grey-level and derivative features. The image is first divided into grid of squared cells whose size is small fraction of the image height. Then each cell is characterized by the following features: normalized grey level, horizontal grey level derivative and vertical grey level derivative (Figure 2.5b). Rodriguez et al. [156] use local gradient histogram features. A Gaussian filter is applied to obtain horizontal and vertical gradients G_x and G_y, and to compute the magnitude and direction of $(G_x, G_y)^T$ (Figure 2.5c). The result is a sequential representation as we can see in the Figure 2.5d.

Once the features are extracted using any of the existing approaches of the literature, the next step is to efficiently evaluate a matching score between the query and the text line. The more used methods, from a mathematical point of view, are: Hidden Markov Models (HMM), Bidirectional Long-Short Term Memory Neural Network (BLSTM NN) and Word Graph (pure posterior probability). Let us further describe the principles of these methods.

HMM-based

The HMM is composed by a set of discrete set of discrete states $s_1, s_2, \ldots$, a transition probability matrix (A_{ij}), the emission probability distributions $b_1, b_2, \ldots$, the starting probabilities for each state and the feature space of the observations. The character-models are built using a linear topology and a mixture of Gaussians (Figure 2.6a), and the words are composed by concatenating character models accordingly (Figure 2.6b).

Once the models are computed, the main problem matching the keywords is the score normalization because it is hard to estimate. There are several ways to normalize the models. The first one is the Cohort model. The cohort model [56] in psycholinguistics and neurolinguistics is a model of lexical retrieval. The *cohort model* is based in the concept that auditory or visual input stimulates neurons as it enters into the brain, rather than at the end of a word. The main drawback of this model is that it needs a set of words, and it does not usually exist for handwritten keyword spotting, hence the Cohort model is basically never used for handwritten keyword spotting.

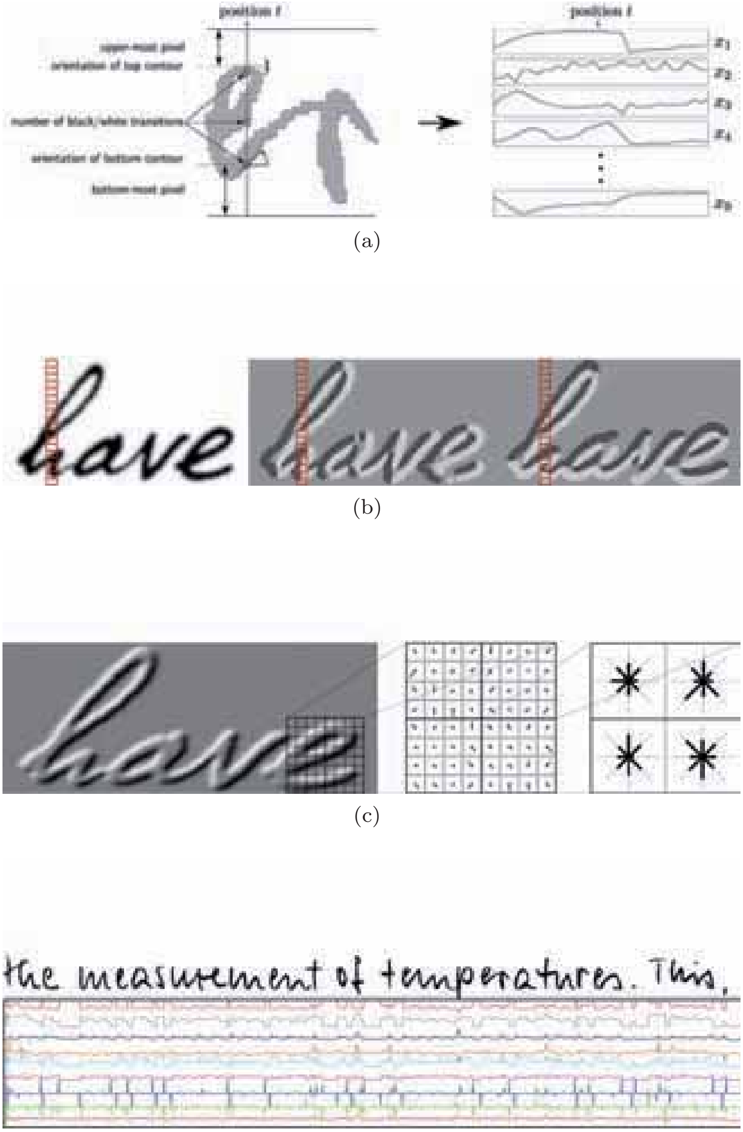
The second score normalization uses *predictor features*. A set of features is extracted and classified separately to determine whether or not a keyword is found. Perronin et al. [147] use HMM with the Viterbi algorithm and combining Fisher Kernel. The approach trains a HMM from a set of positive example images. For each class they compute the normalized gradient vector for each image in the training set (positive and negatives images). Finally, a classifier is trained for each keyword class.

The third one uses *Background model* [60]. It consists in modelling the distribution $P(x)$ of all the words that might be encountered in the system. This can be easily computed using the Viterbi algorithm. The advantage of this score normalization is that no language information (vocabulary, etc.) is needed.

The score normalization described above has the problem that they force some structure in the sequence, because each character is a sequence of HMM states. To remove as much of this constraint as possible, we could loop directly on the states. Using *Gaussian Mixture models* [157] as score normalization, different states are not necessary any more, instead one universal state is sufficient. A Gaussian mixture model is a probability density function defined by a set of Gaussians (mean, covariance matrix) and a coefficient c_i for each Gaussian. The result is a simple structure where the ordering of the frames is disregarded and it is fast to compute.

BLSTM NN-based

Bidirectional Long-Short Term Memory Neural Networks (BLSTM NN) [67] are bidirectional recurrent Neuronal Networks (NN) with specialized Log-Short Term Memory (LSTM) hidden layer. The LSTM is composed by a core the values are stored in cells. The information flows into and out of each cell and it is controlled by gates. A forget gate can reset the core's value to 0. The cells are differentiable memory cells and overcome the vanishing gradient problem: Backpropagation for several time-steps is hard and recurrent NN can therefore not be trained to consider more than just a few time steps. In Figure 2.7 we can observe


position t
upper-most pixel
orientation of top contour
number of black/white transitions
orientation of bottom contour
bottom-most pixel
position t
x_1
x_2
x_3
x_4
x_9
have
the measurement of temperatures. This,

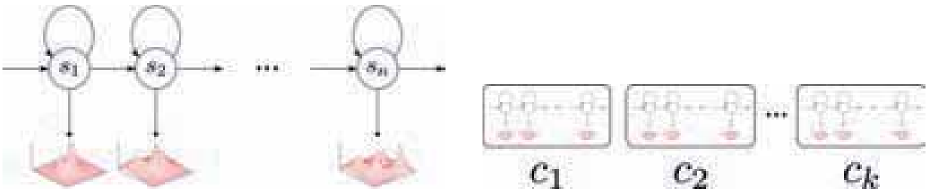
(a)

(b)

(c)

(d)

Figure 2.5: Some sequential features used in learning-based methods and representation of a text line: (a) geometric features; (b) grey-level and derivative features; (c) local gradient histogram features and (d) Sequential representation of a text line (figures extracted from [65]).



(a) (b)

Figure 2.6: Building the models of the HMM. (a) Character-models. (b) Word-models (figures extracted from [65]).

a scheme of the system. The system computes the character probability vector for each position. Each row represents a character or a special character (e.g. when a space between two words appear). The matching score is the product of the path's nodes computed using dynamic programming. The search is done computing the best path in the matrix that follows the characters of the searched word.

2.3 Summary

This Chapter has reviewed the core methodologies, technical ideas and previous works that concern to handwritten word spotting. Table 2.1 shows the main techniques used in handwritten word spotting classified by the task, the kind of the input query, if they are segmentation-free or not and if they are learning-based or not.

Different ways have been proposed to accurately search handwritten words in documents. First, Learning-free methods have been reviewed, from the first proposed works to the most recent using more sophisticated techniques. We have showed methods in the literature that are segmentation-based and works that use a sliding window to localize the words in the documents. We have also reviewed works based in Query-by-String and Query-by-Example.

Secondly, we have overviewed the works in the literature that are Learning-based. These methods estimate the posterior probability of transcribing a word. We have introduced the classical architecture of a Learning-based keyword spotting from the feature extraction to the matching step. We have reviewed the methods that are based in text lines and the methods that are word-based.

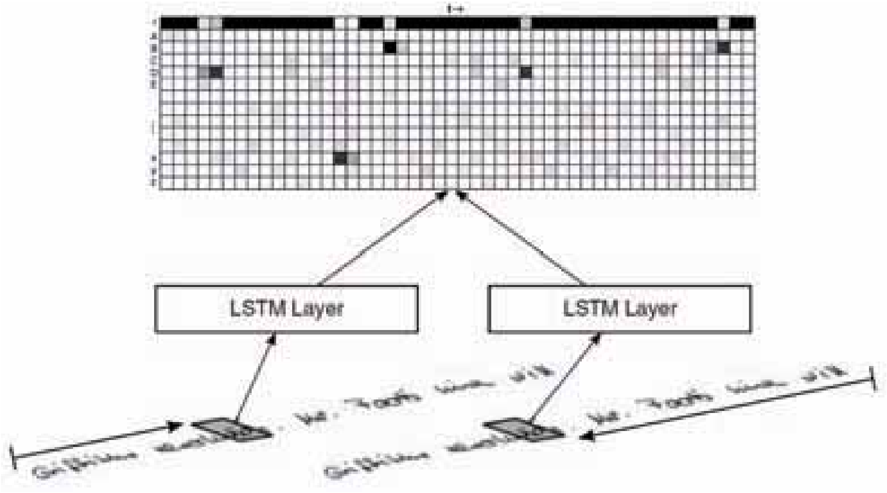


Figure 2.7: An illustration of the mode of operation of the BLSTM neural network. For each position, the output layer sums up the values of the two hidden LSTM layers (figure extracted from [65]).

Chapter 3

Line Segmentation

Layout segmentation in handwritten documents is an important task in the recognition of historical documents. The good localization of the words inside the documents is an important process because many recognition techniques require accurate word segmentation. Most of the word segmentation approaches assume that a line segmentation approach is available and often use simple existing techniques. But, handwritten document images contain text lines with multiple orientations, horizontally-overlapping components, touching and overlapping characters between consecutive text-lines and different document structures making line segmentation a difficult task. In this chapter we present a new approach for handwritten text line segmentation solving the main difficulties in handwritten documents. The proposed algorithm formulates line segmentation as finding the central path in the area between two consecutive lines. This is solved as a graph traversal problem. A graph is constructed using the skeleton of the image. Then, a path-finding algorithm is used to find the optimum path between text lines. Afterwards, a word segmentation technique of the state of the art is applied to localize the words inside the text lines.

3.1 Introduction

Layout segmentation and, in particular, line segmentation is a key step to guarantee a good performance in handwriting recognition. Line segmentation and word segmentation are critical early processing steps for several document analysis tasks, such as word spotting [151, 154, 181], and text recognition[98]. The objective is to extract word images from the documents. The importance in this task is because many word recognition techniques require an accurate word segmentation. Wrongly segmented word images will cause the recognition step to fail in handwritten document recognition systems. Most of the word segmentation approaches assume that a line segmentation approach is available and often use simple existing techniques. However, it is important to have an accurate line segmentation to obtain better results in word segmentation. The segmentation of lines is not a trivial process. Historical documents have several difficulties that can get complicated the segmentation of text lines. First, the physical lifetime degradation of the original documents, related to the frequent handling and careless storage, produces holes, spots, broken strokes, ink bleed, winkles, etc.

Second, if the scanning process has not been rigorous, it might introduce difficulties such as non stationary noise due to illumination changes, show-through effect, low contrast, warping, etc. Third, the inherent irregularity of handwriting is also a problem. Besides these general difficulties, the characteristics of the handwriting and the configuration of the text lines may provoke additional difficulties. First, curvilinear baselines due to the non-straight pen movement. Second, lines of crowded writing styles, which are more difficult to segment because they are close to each other and increase the overlapping. Third, the presence of touching and horizontally-overlapped components [82] when ascenders and descenders exceed the lower and upper bounds. And finally, punctuation and diacritic symbols, which are located between lines and introduce confusion in the decoding process of the physical structure. In Fig. 3.1 we illustrate some of the difficulties described above.

The main contribution of this work is an accurate line segmentation approach for handwritten documents. Although the final goal is to segment the words, an accurate line segmentation is required to get the entire words, even the words are touching or overlapped. Although accurate algorithms for locating text lines in machine printed documents have been proposed [78, 141], they have shown some drawbacks in handwritten documents and there is still room for improvement. Several text-line segmentation algorithms for handwritten documents have been proposed (see Section 3.2).

The line segmentation process in handwritten documents can be divided in two tasks: localization and segmentation. Localization means to find the position of the text line, for example by its baseline, or central axis. Segmentation refers to a pixel-wise labelling. Localization has a good performance in highly structured documents, when text lines are isolated (as they follow rule lines or form boxes [111]) or when the methods are designed ad-hoc to a particular layout and document type. But when the stated difficulties of handwritten documents are present: touching lines, curvilinear text lines and horizontally-overlapping components, the performance decreases and the accurate segmentation is very difficult. Finer analysis processes are performed, especially in touching parts. Some methods use connected components to group the touching parts to the closest text lines [97, 181], while other methods are more accurate and analyse touching parts to split them accurately [102, 145]. The segmentation is done taking into account the properties and the shape of the studied area [182]. In this chapter we present a line segmentation approach that in addition to the general difficulties of historical documents, tackles with these problems without loosing the generality, so the approach is writer-independent, layout-independent, and is able to cope with skew and warping disturb.

The algorithm presented in this work, first, estimates the localization of the text lines, and second, segments the text lines. The accuracy of the location is not primordial because our algorithm is focused in finding the optimal path with minimum cost in-between two consecutive lines in the image background. This is solved as a graph traversal problem. Hence, the skeleton of the background image is converted to a graph. After finding potential starting and ending nodes, minimum cost paths between pairs of starting and ending nodes are searched. Local cost functions associated to the graph nodes are defined to find the best continuation path in terms of the potential configurations. The problem of touching text lines is solved adding virtual edges between the candidate nodes that are around the involved characters.

Once the lines are segmented, the words are segmented using the approach developed by Manmatha and Rothfeder [123] to segment words in old handwritten documents. In their work the lines of the documents are extracted using grey-level projections and word segmentation is done by filtering with anisotropic Laplacians in scale space. Essentially, they optimize for the scale by which each line should be filtered. In their work they showed that this approach for word segmentation works better than a gap metric segmentation approach.

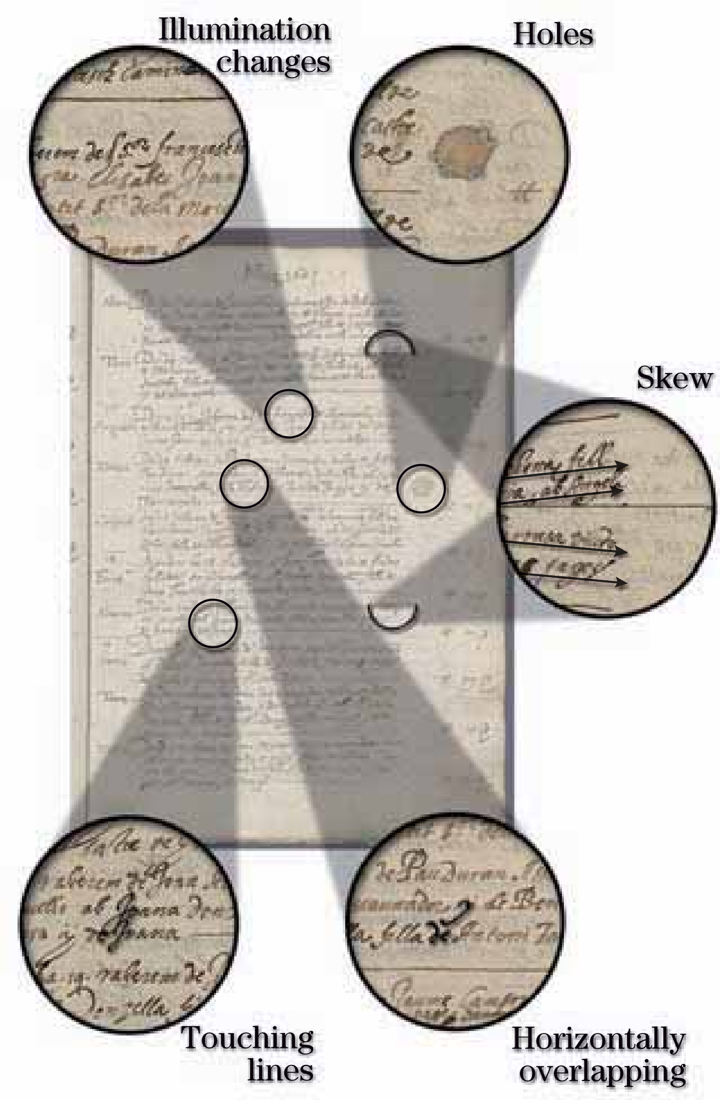


Figure 3.1: Some difficulties in historical handwritten documents: illumination changes, holes, skew, horizontally-overlapping, and touching lines.

The rest of the chapter is structured as follows. In section 3.2 the state-of-art is reviewed. Section 3.3 describes the proposed line segmentation method. Section 3.4 describes our proposal for line-wise word segmentation. Section 3.5 shows the experimental results. Finally, we present the conclusions in the last section of the chapter.

3.2 Related Work

There are a wide variety of word segmentation methods reported in the literature, but they have mostly been developed for machine printed characters, where the text typically has inter-word gaps that are much larger than the inter-character gaps (gaps between characters within one word). The results obtained using these methods in handwritten documents are poor and unsatisfactory. As far as we know, the major part of the approaches are oriented to specific datasets, or are elements of integrated systems for specific tasks, for example, bank check and postal address recognition [136].

Most of the literature on handwritten document recognition is focused on modern handwritten databases. Handwritten recognition systems are often tested using specific documents created for the purpose of testing these systems and, thus, are forced in how they are created [130]. Historical handwritten documents present more drawbacks and difficulties to solve, as mentioned before. Feldbach and Tonnies [57] have proposed a bottom up method for historical church documents that needs to be set according to the type of handwriting.

The literature on word segmentation shows three approaches. The first one - the gap metric approach - uses connected components (CC) to extract words. Many papers assume that each CC belongs to only one word and gaps between words are greater than the gaps between characters. Seni and Cohen [177] evaluated eight different distances measures between pairs of connected components. The second one considers the word extraction as involving determining whether each position in a text line belongs to a letter of a word or a space between two words [118]. For this task a Hidden Markov Model is used. The third approach is that of Manmatha and Rothfeder [123], which showed that their algorithm works better than gap metrics for word segmentation. This approach has been previously described.

Handwritten text line segmentation has received interest over the last years [111]. In addition to relevant publications, a series of competitions on this topic has been organized in international events (e.g. the ICDAR2009 Handwritten Segmentation Contest [69], with 12 participants, or ICDAR2013 Handwritten Segmentation Contest [190], with 14 participants). Observing the existing methods to segment handwritten documents, we propose a classification into 5 categories: projection-based, Hough-based, component grouping, morphology-based operations and other methods.

3.2.1 Taxonomy of methods

Projection-based methods are based in projection profiles. Black pixels are projected on the vertical axis. The maxima and the minima of the resulting histogram correspond to regions with large and low horizontal density of pixels. The lines are obtained computing the average distance between the peaks of the histogram [123, 207]. Some techniques [13, 145] can deal with variations in the text orientation, but they are sensitive to the size of characters and the gaps between successive words. To solve these problems, some methods [88, 89] detect areas where two lines are merged due to long ascenders or descenders and compute local histograms to split the lines. The *PAIS* method [69] improves a line segmentation approach based in projections applying the knowledge of estimated line-distance and reasonable black-to-white traversal numbers.

Most of these techniques are simple and easy to implement, but they do not work efficiently with multi-skewed text lines, touching components and horizontally-overlapping component configurations.

Hough-based methods [115] describe parametric geometric shapes (straight lines, circles and ellipses are the most usual) and identify geometric locations that suggest the existence of the sought shape. They are proper methods to detect lines because text lines are usually parallel, and consequently in the Hough space they generate a configuration consisting of aligned peaks at a regular distance. Although these methods handle documents with variations in the skew angle between text lines, they are not accurate when the skew varies along the same text line, i.e. curvilinear lines. In addition, these methods can not achieve an accurate segmentation of touching or overlapping lines.

Grouping-based methods, also known as bottom-up strategies, group components according to a specific property. Most of the works belonging to this category are based in searching for components that are horizontally aligned. In [209, 210] connected components are organized in a tree structure in terms of a metric distance, and grouped by a minimal spanning tree (MST) algorithm. The *CMM* method [69] groups components that are horizontally aligned. The *JadavpurUniv* method [69] analyzes dimension features of the components to determine the handwriting style and to set the threshold values for inter-word spacing. Li et al. [110] propose an approach based on density estimation criteria to cluster components. Although grouping methods usually present problems to segment touching text lines, the method described in [110] includes a post-process that detects and splits them. Feldbach and Tonnies [57] join baselines segments, computed in a pre-process step, in historical church registers, similar to the main experimental focus of this chapter. Kumar et al. [101] present a method that computes the similarity between text components based on local orientation detection and shortest path in graphs. The proposed method can handle with printed documents and complex layouts in handwritten documents, however like the other grouping-based methods, it fails to segment touching text lines.

Morphology-based methods have been used in many works for layout analysis, especially when documents contain text blocks that are strictly oriented horizontally or vertically, i.e. columns and lines. Smearing-based operators can be seen as morphological methods with horizontal structuring elements. Particular examples are the methods described in [45, 168]. They combine the two fundamental morphological operations (dilation and erosion) with horizontal projections and run-length smearing algorithm (RLSA) respectively. Other methods [124, 175] use anisotropic Gaussian kernel or local estimation count map.

In the *EPITA Research and Development Laboratory (LRDE)* method [69], a morphological watershed transform is computed once the document is smoothed using an anisotropic Gaussian filter. Shi et al. [182] propose a technique based on a generalized adaptive local connectivity map which uses a steerable directional filter. In the *Ecole de Technologie Superieure (ETS)* method [69], the text is smeared using a modified version of Weickest's coherence-enhancing diffusion filter to segment lines. Alaei et al. [2] use strip-like structures to decompose the text block in vertically parallel structures. Each one is labelled using their grey intensity and applying a morphological dilation operation. Nicolau et al. [139] shred text images into strips along the white gaps in between text lines. Saabni et al. [172] propose a method that computes an energy map of the input text block image and determines the seams that pass across text lines.

These kind of methods also have problems in documents with overlapping of adjacent text lines. To overcome this problem, some morphology-based works define ad-hoc heuristics

[33] or min-cut/max-flow graph cut algorithm [91].

Graph based: Some approaches use graphs to compactly represent the image structure keeping the relevant information on the arrangement of text lines. Energy (or cost) functions are used to establish the optimal path between nodes that segment the lines. In [103] the segmentation is proposed as a graph cut problem. The graph is built using either the pixels or the connected components of the image as nodes, which are linked to its neighbours through edges.

The *PortoUniv* method [69] represents the image as a graph, which is used to find the minimum energy paths between the borders of the page using an efficient dynamic programming approach. The robustness of projection based methods is combined with the flexibility of graph-based methods by Wahlberg et al. in [200]. The graph is constructed using the foreground of the image.

Other methods: There is a miscellanea of other methods that can not be classified into any of the main categories described above. Kass et al. [87] use active contours to explore the borders of the image objects with relevant differences between the foreground and the background in characteristics like brightness or colour. Bukhari et al. [22, 23] adapt active contours (snakes) over the ridges of the geometry of the gray level image to detect the central axis of parts of text lines. The method properly localizes the text lines in the documents, even with the difficulties explained above. In case of touching components lying in two different text lines (a connected component lies over two text lines), they are horizontally of vertically cut depending on the slope of underlying ridges into equal number of parts. However, this can split words into different lines (e.g. ascenders or descenders of the words are split in the above or below text line). Liwicki et al. [114] use dynamic programming to find text lines, computing minimum cost paths in between consecutive text lines. Stafylakis et al. use a Viterbi algorithm to segment the text-lines [189].

3.2.2 Discussion

In order to summarize the above described methods for segment lines in handwritten documents, Table 3.1 overviews their strengths and weaknesses and Table 3.2 shows the type of documents (binary or grey level) usually used as input. The graph based methods are not included in the taxonomy because the methodologies used in these kind of approaches are too diverse and, therefore they are unable to be generalized under a common assessment. The rest of the methods are compared (Table 3.1) according to the following criteria: if the method works in printed and/or handwritten documents; if the method handles variations in the skew angle between text lines and when the skew varies along the same text line (curved lines) or not; if the method can solve the problem of horizontally-overlapping components; and if the method can properly split touching lines or not.

Table 3.1: Comparative of text line detection methods.

	Printed documents	Handwritten documents	Skewed documents	Curved lines	Over-line	Touching lines
Projection-based	++	−−	+	−−	−−	−−
Hough-based	++	−	++	−−	−−	−−
Grouping-based	++	+	++	+	+	−−
Morphology-based	++	+	++	+	++	−−
Our method	++	++	++	+	++	+

Table 3.2: Classification of the methods according to the input images.

	Gray level	Binary
Projection-based	[123]	[13] [88] [89] [145] [200]
Hough-based		[115]
Grouping-based		[208] [209]
Morphological operations	[2] [124] [181]	[33] [45] [91] [168] [175]
Other methods	[22] [23] [87] [182]	[103] [114] [189]

Text line segmentation in printed documents is a problem that has been solved from different approaches with satisfactory results. However, when dealing with handwritten documents, state-of-the-art methods, specially projection and Hough-based, present some difficulties to properly segment the lines. Errors in segmentation are usually due to noise and the non-rigid structure of this kind of documents. These irregularities lead to the three main problems that are present in the handwritten text line segmentation: curved lines, horizontally-overlapping lines and touching lines. Methods based on morphological operations and grouping-based methods are able to deal with curved and horizontally-overlapping lines. However, the segmentation of two touching lines still remains as an unsolved problem among state-of-the-art methods.

Besides the taxonomy presented in Table 3.1, an additional criterion that is worth to be considered is whether the approach requires a learning process [85, 211] or not [24]. The methods based on projection profiles have good accuracy when they are applied to documents with the same structure layout and style. The main problem of these approaches is their adaptability. They have to learn their models for every new document, for those that present a new structure layout, or a new handwriting style or a different time period. This kind of methods need some samples of every type of documents to learn a model. However, such samples are not always easily available. In addition to this drawback, learning-based methods have a higher computational cost, even though the learning process is an off-line process. The methods without a learning process are more adaptive. They can robustly extract the lines of any kind of documents and the computational cost is lower. However, the performance decreases when dealing with a close collection because they are not adapted to the specificities of that set.

Most of the methods presented above localize text lines with a high accuracy, but only a few of them focus their methods to solve the problem of overlapping and touching components [83, 143, 160]. Even so, Kang et al. [83] require a learning process to define local configurations of touching components. Ouwayed et al. [143] split touching components following the descending parts of the characters, which is a common property of most characters in the Arabic alphabet. Rohini et al. [160] localize touching components extracting the core region (space between consecutive text lines) using horizontal projections profiles. Then, the method needs a pre-process to deskew the curved text lines.

We have also classified the above methods according to the kind of input image: binary or grey-level. Usually, projection-based methods use binary images as input, except the method of Manmatha et al. [123] that uses a modified version of [74] extended to grey-level images. Hough-based and grouping-based methods use binary images in their approaches because they perceptually group basic primitives (key-points or connected components). Morphological-based and other methods have a large diversity of algorithms and each one uses a different type of images.

From the comparative shown in Table 3.1, we can conclude that the main challenges are the segmentation of touching, horizontally-overlapping and curvilinear lines. The main contribution of the approach proposed in this chapter is its robustness and high performance when segmenting lines under the above mentioned problems. From the comparative in Table 3.2, we conclude that most methods use binary images as input. In our approach, we assume that images have been previously binarized so it simplifies the process. However we will show how the method is robust to binarization noise so this process is not critical in the pipeline.

3.2.3 Contribution

We present an approach inspired by graph representation methods. Graphs are a useful tool to capture the structure of the image objects (lines and words in our case). In addition,

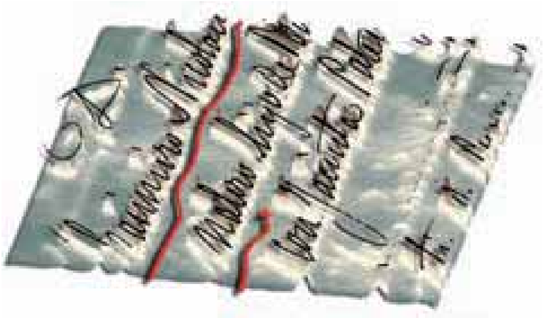


Figure 3.2: In a 3D-view, text lines can been seen as peaks and the space between them as valleys.

graph theory offers solid and elegant methods. Graph vertices are usually constructed from pixels or connected components. Graph edges represent spatial relations between connected components and are usually weighted by the distance between the connecting vertices [20, 208].

In this thesis has been developed a segmentation-based handwritten word spotting, thus, the main objective of this work is to localize text lines and to solve the problem of touching lines adding new *virtual edges* to the graph. These characters are split using some heuristics which evaluate the spatial information around the area involved. This technique is not oriented to a specific writer, style or alphabet, and it is able to cope with multi-oriented text lines and historical documents. The approach presented in this work belongs to the group of methods which do not need a learning process to segment lines, therefore it does not need labelled samples. Next, we explain this approach in detail.

3.3 Line Segmentation Approach

Humans tend to write text in blocks, and they usually use the same space between lines. In a 3D-view of the intensity image, this characteristic can be seen as a valley: if we compute the distance function and hence see the topography of the image, the in-line space is seen as a valley and the words as crests. Using this observation, we first compute the distance function in the input image, which corresponds to the skeleton of the background. Afterwards, we detect paths through the valleys of the document. The paths consist of background points at equal distance to the words above and below. We use a path-finding algorithm to select which paths are the best to segment the lines. In Fig. 3.2 we can observe a representation of this characteristic.

The goal of our method is to automatically locate and segment text line regions in handwritten documents. The system consists of two big stages, as shown in Fig. 3.3. The first stage is the enhancing of the documents and the localization of the text lines. Then, the second stage is the line segmentation. We compute the skeleton of the background image. All the possible pixel-paths are computed using an iterative thinning function. Then the paths are converted to a graph, which will be used to find the optimal paths that segment the text lines. Then, the best paths that segment the text lines are found. For this purpose, we adapt the A-star path finding algorithm. Finally, a consistency checking step is applied. Let us further describe the different steps.

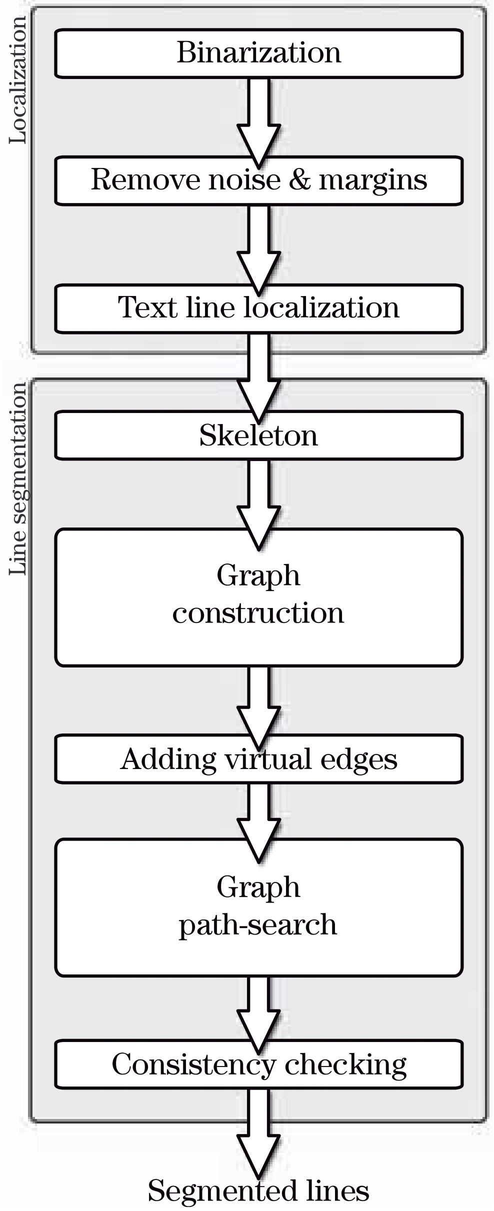
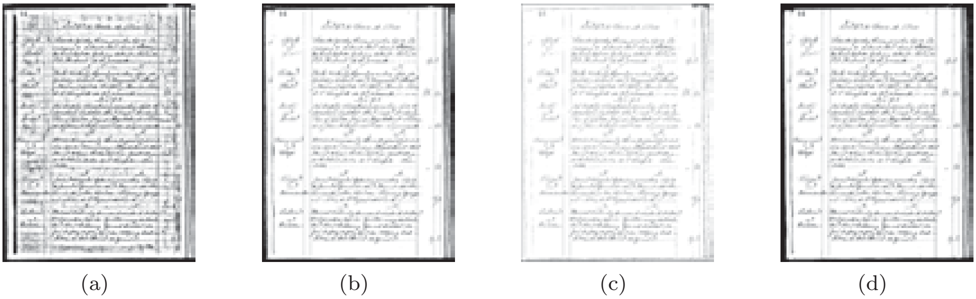


Figure 3.3: Flowchart of the proposed approach.

3.3.1 Localization

The localization step is preceded by an enhancing process of the documents. The main idea is to use the valleys that appear between the text lines to segment them. In a 3D interpretation of the images the words of the text lines represent can be roughly associated to crests, and the noise of the documents can introduce hills that produce the diversification of the valleys. This fact introduces new possible paths and the computational cost increases proportionally to the noise. The number of valleys is reduced applying morphological operations to smooth the image and to reduce the hilltops.

First, the image is binarized using the Otsu's method [142]. Several binarization methods have been tested in our document images: Niblack (Fig. 3.4a) generates noisy images, Sauvola (Fig. 3.4c) loses important information and the characters are thinned, Bernsen (Fig. 3.4b) generates good results but the computational time is very high. Otsu (Fig. 3.4d) obtains a clear image, without noise, the characters do not loose pixels and are well defined, and in addition, it is the fastest method.



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

Figure 3.4: Binarization of the documents using several methods. (a) Niblack's method. (b) Bernsen's method. (c) Sauvola's method. (d) Otsu's method.

The skeleton is a simplification of the topology of the image. In this work, the skeleton of the distance function applied to the background allows to obtain the seams between text lines (valleys of the distance transform image). Since the images are originally binary (white paper and black ink), computing the skeleton in the grey-level image would not give the same result so it would obtain distorted skeleton with extra-segments in images due the scanning process. Consequently, the path search algorithm has to analyze all these extra paths, so the computation cost increases exponentially. For this reason, we prefer to binarize the input images.

The scanning process introduces some distortions. One of them is the set of page margins (a black area around the page image). We delete these margins by applying morphological operations and selecting the biggest blobs in the periphery of the document. Then, a median filter is applied with a mask of size $4x4$ to remove the speckle noise. The problem of using this kind of filters is that they provoke a thinning of the characters. An alternate sequential filter of opening and closing operations is applied to correct this problem.

Once the image is binarized, text lines are localized using a projection-base method. To get a best accuracy, a rough estimation of the skew of the document is computed using the Wigner-Ville distribution [143].

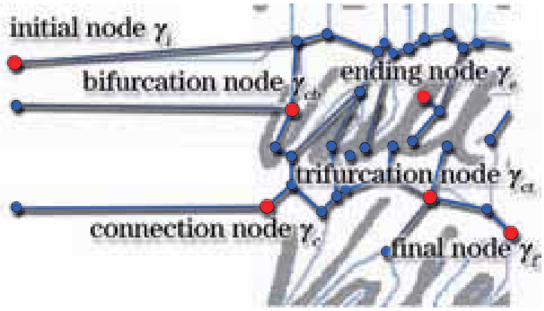

initial node γ_i
bifurcation node γ_{cb}
ending node γ_e
trifurcation node γ_{ct}
connection node γ_c
final node γ_f

Figure 3.5: Types of graph nodes computed from the skeleton image.

3.3.2 Graph construction

The proposed method for segmenting lines in handwritten documents is based on searching the pixel paths of minimum cost on the skeleton of the background image between the left and the right margins of the previously localized text lines.

For the sake of efficiency, instead of directly processing the skeleton at pixel level, it is approximated by a graph $G = (V, E)$ that preserves its structure. The set of vertices V of the skeleton graph represents characteristic points (terminal and intersection points), and the set of edges E represents sequences of consecutive skeleton points between vertices. Formally a skeleton graph G is represented as an attributed graph $G = (V, E, L_V, L_E)$ where L_V and L_E are two labelling functions that assign attributes to nodes and edges respectively. The labelling functions L_V and L_E are defined as follows.

Given a vertex $v \in V$, the attributes assigned to it are denoted as: $L_V(v) = [N_v, x_v, y_v, t_v]$ where N_v denotes the number of neighbours, $(x_v,\ y_v)$ are the coordinates of the pixel and t_v is the type of the node out of $\{\gamma_i,\ \gamma_f,\ \gamma_e,\ \gamma_c,\ \gamma_{cb},\ \gamma_{ct}\}$ (Fig. 3.5). These types of nodes represent the following configurations:

- γ_i: an *initial node* is a terminal pixel (with only one neighbour) of the skeleton located at the left margin of the image (first column of the image pixels).
- γ_f: a *final node* is a terminal pixel of the skeleton located at the right margin of the image (last column of the image).
- γ_e: an *ending node* defines the end of a path in the central part of the image. It is a terminal pixel of the skeleton located at any place of the image except the first and the last column of the image pixels.
- γ_c: a *connection node* represents a corner in a path, i.e. it has two incident edges (skeleton paths) with an important change in the orientation.
- γ_{cb}: a *bifurcation node* defines a branch (three incident edges). It is a pixel of the skeleton which has three neighbours.
- γ_{ct}: a *trifurcation node* defines a crossing (four incident edges). It is a pixel of the skeleton which has four neighbours.

An edge $e = (v_s, v_t) \in E$ stores the current path of chain pixels joining the source vertex $v_s \in V$ and the target vertex $v_t \in V$; the Euclidean distance between v_s and v_t; and the type of edge: *true edge* (when there is a true path of pixels between v_s and v_t) or *virtual edge*.

The problem of touching text lines is solved by adding virtual edges. Due to the geometry of the distance function image, two touching words provide a discontinuity in the path, i.e.

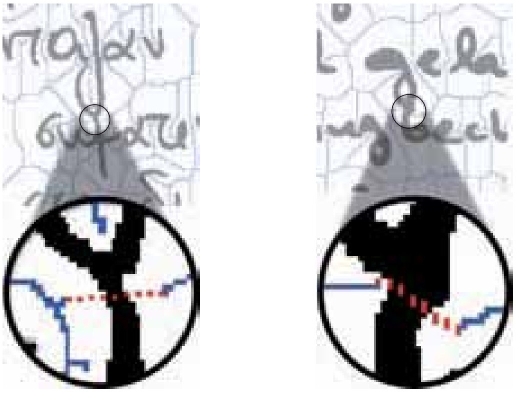


Figure 3.6: Adding virtual edges between two *ending nodes*.

the skeleton computed in this area creates two or more *ending points* around the place where the touching problem appears. These *ending nodes* are used to solve this problem. We connect these nodes using new edges. These new edges are known as *virtual edges* (Fig. 3.6). These *virtual edges* are sub-path candidates. So they allow to reconstruct the broken path traversing the touching characters through the minimum path. A virtual edge is created between two ending nodes γ_e when they are very close. The threshold radius R_v (see Eq. 3.1) of the area around an ending node to search for other connecting nodes is experimentally set proportional to the size of the image. The mean of the separations between the text lines is estimated in the localization process. When t_e is a virtual edge, then p_e is empty.

$$R_v = \frac{\sum_{i=2}^{n} y_i - y_{i-1}}{n} \tag{3.1}$$

3.3.3 Graph path-search

Once the skeleton of the background image has been converted to a graph, the problem of text line finding is translated into searching for shortest paths in the graph according to some considerations. A-star (A*) is a computer algorithm that is widely used in path finding and graph traversal. Hart et al. [76] described the algorithm as an extension of the Dijkstra's 1959 algorithm [43].

The algorithm proposed in this work is a modified version of the classical A-star algorithm. In the classical algorithm the starting and the target point should be established. In our problem, we do not know a priori which node, among the final nodes, is the target node. For each *initial node* γ_i of the graph, the algorithm iteratively searches a minimum cost path until a final node γ_f is reached.

The objective of this step is to find the best path in the valley between two crests, or text lines, (Eq. 3.2). The solution is found as a minimum energy path in the graph G between an initial node $v_1 \in \{\gamma_i\}$ [1] and a final node $v_z \in \{\gamma_f\}$. A path P contains a sorted list of

[1] For the sake of understanding we denote $v \in \{\gamma_i\}$ to represent a node belonging to the category of initial nodes (equally for the rest of types).

nodes and edges. The path P can be seen as a representation of the valley path between two text lines calculated based on the skeleton. We denote a path P^j as follows:

$$P^j = [v_1^j, e_1^j, v_2^j, e_2^j \dots, e_{z-1}^j, v_z^j] \tag{3.2}$$

where $v_1^j \in \{\gamma_i\}$ and $v_z^j \in \{\gamma_f\}$.

The cost of a path P^j is the accumulated cost of the local transitions (local paths) between consecutive nodes. Formally:

$$p(P^j) = c(v_1^j, v_2^j, e_1^j) + c(v_2^j, v_3^j, e_2^j) + \cdots + c(v_{z-1}^j, v_z^j, e_{z-1}^j). \tag{3.3}$$

Given an initial node $v_1 \in \{\gamma_i\}$, the algorithm searches for the minimum cost path that reaches a final node $v_z \in \{\gamma_f\}$ in the opposite side of the page.

The algorithm searches the best path in the state-space S, where each state represents a partial path explored to this point. The algorithm explores in a state S_i all the possibles states $[S_i^1, S_i^2, \dots, S_i^m]$ to go. An intermediate state S_i corresponds to a graph node v_n^j, and the next possible states correspond to all the possible next graph nodes v_{n+1}^j that are connected to the node v_n^j.

The transition from a state S_i to the next state S_{i+1} is computed in terms of some heuristic functions that model local configurations (explained with more details in the next paragraphs). Each transition has a cost of moving from a state to the next state according to a weighted combination of four predefined heuristics corresponding to four possible local configurations. Briefly, the heuristics give the cost of a path taking into account its trend, the bound of each text line, and also to avoid the possible backward paths and to solve the problems of touching-components and horizontally-overlapping objects. The next state corresponds to the minimum cost transition from the node v_n^j to the neighbour v_{n+1}^j through the edge e_n^j (chosen among all the possible nodes v_{n+1}^j connected to v_n^j). The cost of the path-step v_n^j to v_{n+1}^j through the edge e_n^j is denoted as $c(v_n^j, v_{n+1}^j, e_n^j)$. Formally:

$$c(v_n^j, v_{n+1}^j, e_n^j) = \alpha_1 * h_1(v_n^j, v_{n+1}^j, e_n^j) + \cdots + \alpha_4 * h_4(v_n^j, v_{n+1}^j, e_n^j) \tag{3.4}$$

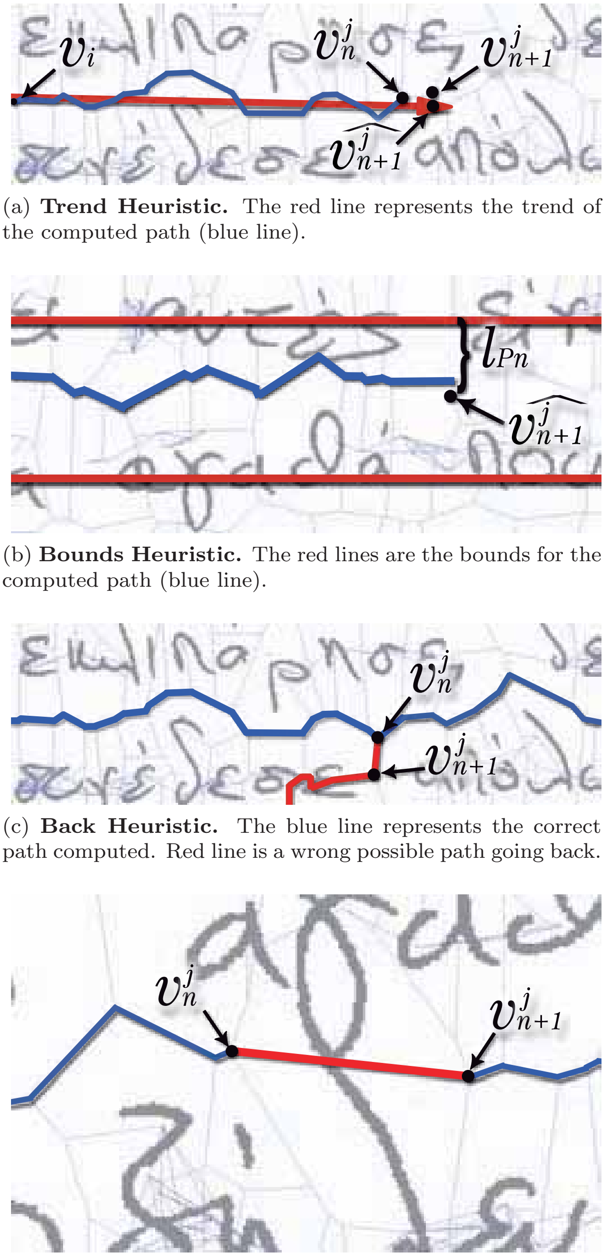
where α_i are the corresponding weights computed experimentally ($\alpha_1 = 1, \alpha_2 = 0.5, \alpha_3 = 0.01$ and $\alpha_4 = 0.2$), $v_n^j \in \{\gamma_i, \gamma_e, \gamma_c, \gamma_{cb}, \gamma_{ct}\}$, $v_{n+1}^j \in \{\gamma_f, \gamma_e, \gamma_c, \gamma_{cb}, \gamma_{ct}\}$ and e_n^j is the edge between v_n and v_{n+1}, which contains information as if it is virtual or real. To simplify the notation $x_{v_{n+1}^j}$ is denoted as x_{n+1}^j and $y_{v_{n+1}^j}$ is denoted as y_{n+1}^j.

Let us further describe the four heuristics that are considered to model the cost function.

H1.- Trend Heuristic. Humans write text lines following a uniform direction, without abrupt orientation changes. Although a text line presents a curvilinear orientation, the local orientation trend predicts the smoothest continuation path. This property is used to fix the path to this trend and to avoid the possibility of sharp curves in the computed paths.

We use the trend of the path, computed by a linear regression, to compute the cost of the new node in the path taking into account the nearby nodes in the y axis. Given the estimated trend point $\widehat{v_{n+1}^j}$,the starting node $v_i \in \{\gamma_i\}$ and the source node v_n^j, the cost is the sum of the respective differences between those three points and the target node v_{n+1}^j. Formally:

$$\begin{aligned} h_1(v_n^j, v_{n+1}^j, e_n^j) =& |f_{lr}(v_{n+1}^j) - y_{n+1}^j| \\ &+ |y_i^j - y_n^j| + |y_{n+1}^j - y_n^j| \end{aligned} \tag{3.5}$$



(a) **Trend Heuristic.** The red line represents the trend of the computed path (blue line).

(b) **Bounds Heuristic.** The red lines are the bounds for the computed path (blue line).

(c) **Back Heuristic.** The blue line represents the correct path computed. Red line is a wrong possible path going back.

(d) **Virtual Paths Heuristics.** A virtual path (red line) connects to ending nodes to solve the problem of touching lines.

Figure 3.7: Illustration of the different heuristics.

where $f_{lr}(v^j_{n+1}) = \widehat{y^j_{n+1}}$ is a linear regression obtained from all the nodes that compose the temporal path $[v^j_1, e^j_1, v^j_2, e^j_2 \ldots, v^j_n]$, where v^j_1 is a starting node and v^j_n is the source node and $\widehat{v^j_{n+1}}$ is an estimation of v^j_{n+1}.

H2.- Bounds Heuristic. Humans tend to write text lines parallel each other. We also take into account that it is not usual to cross lines when writing. The objective of this heuristic is to fix the path inside the upper and lower bounds of two text lines, defining a band along which the path can not surpass (Fig. 3.7b). We fix the path between the upper and the bottom limit of each line. Some documents contain multi-skewed lines. To correct this problem, the bounds of each line are adapted dynamically at each iteration in terms of the current trend of the path. The graph edges located between the words of the same text line have a higher cost than the edges located between different text lines. Formally:

$$h_2(v^j_n, v^j_{n+1}, e^j_n) = \begin{cases} 0 & \text{if } f_{hb}(v^j_n, v^j_{n+1}) \leq l_{P_n} \\ f_{hb}(v^j_n, v^j_{n+1}) & \text{otherwise} \end{cases} \tag{3.6}$$

where $f_{hb}(v^j_n, v^j_{n+1}) = \mid f_{lr}(v^j_{n+1}) - y_{n+1^j} \mid$ and l_{P_n} is the limit estimated previously of this path.

H3.- Backwards Heuristic. Following the premises of the last two heuristics (H1 and H2), paths cannot go back abruptly. They have to follow a trend and it has to be inside a bound. Another premise is that the target of our path-finding algorithm is located on the right margin of the document, so paths that go backwards are penalized with a high cost.

The direction of the path is checked, and if it goes back, we increase the cost directly proportional to the retracted distance (Fig. 3.7c). Although the cost of this path is high, in some cases the algorithm chooses this option because there is no alternative or it is too expensive. Formally:

$$h_3(v^j_n, v^j_{n+1}, e^j_n) = \begin{cases} 0 & \text{if } x^j_{n+1} - x^j_n \geq 0 \\ d_e(v^j_n, v^j_{n+1}) & \text{otherwise} \end{cases} \tag{3.7}$$

where d_e is the Euclidean distance.

H4.- Virtual Paths Heuristic. As we have explained before, the problem of touching text lines is solved by adding virtual edges to the graph. In the construction process of the graph, we introduce virtual edges between intermediate ending points (Fig. 3.7d). We use this kind of edges when there is no alternative path (or the cost of the other paths is too high), or the alternative path has a high deviation passing through the words of the text lines above or below. Formally:

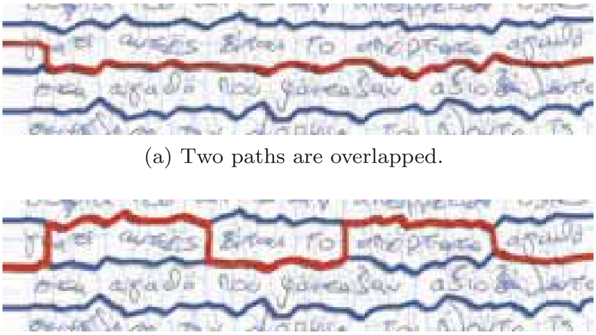
$$h_4(v^j_n, v^j_{n+1}, e^j_n) = d_e(v^j_n, v^j_{n+1}) \tag{3.8}$$

if e^j_n is a virtual path.

In summary, the algorithm computes the best path in the valley between two crests, or text lines, for each initial node $v^j_i \in \{\gamma_i\}$. For each node $v_n \in V$, its branches $v_{n+1} \in V$ are expanded, and the *heuristic cost* h is computed from v^j_n to v^j_{n+1}. The branch with the minimum cost (sum of the *real cost* and the *heuristic cost*) is chosen. The *real cost* is the cost computed from the initial node v^j_i to the current node v^j_n. Contrary, the *heuristic cost* is an estimation of the cost from v^j_n to v^j_{n+1}. The algorithm ends when it finds an ending node $v^j_f \in \{\gamma_f\}$. The algorithm is summarized in **Algorithm 1**.

Algorithm 1 Path finding.

```
$LIST\_PATHS = NULL$;
for all $v_i^j \in V$ do
    $OPEN\_LIST = null$;
    $CLOSE\_LIST = null$;
    Insert $v_i^j$ node in $OPEN\_LIST$
    $P* = null$;
    while $OPEN\_LIST \neq empty$ do
        Select the first node $v_n^j \in V$ from $OPEN\_LIST$, remove it from
        $OPEN\_LIST$, and put it on $CLOSED\_LIST$
        if $v_n^{\in}\{\gamma_f\}$ then
            exit
        end if
        Expand node $v_n^j$, generating the set $V_M \subseteq V$, of its successors that are not
        already ancestors of $v_n$ in $P*$
        for all $v_{n+1}^j \in V_M$ do
            $Cost = getHeuristic(v_n, v_{n+1}^j, v_i^j, e_n)$
            if $v_{n+1}^j$ is not in $OPEN\_LIST$ then
                Insert $v_{n+1}^j$ node in $OPEN\_LIST$
            else
                $v_{eq} = getOpenListNode(v_{n+1}^j)$;
                if $Cost(v_{n+1}^j) < Cost(v_{eq})$ then
                    Update $v_{eq}$ with new cost
                end if
            end if
        end for
    end while
    the path $P*$ is obtained by tracing a path along the pointers from $v_n^j$ to $v_i^j \in$
    $\{\gamma_i\}$.
    Add $P*$ to $LIST\_PATHS$
end for
```



(a) Two paths are overlapped.

(b) One path is overlapped on two other paths.

Figure 3.8: Examples of the different types of overlapping between paths.

3.3.4 Consistency checking

Although the proposed method is able to cope with noise, some documents may present high levels of degradation. It results in wrongly segmented lines, in particular two consecutive paths may be overlapped (Fig. 3.8). To avoid this problem, the consistency checking process is a post-process that looks for the overlapped paths and splits them accordingly. During the graph path-search step, the algorithm checks the overlapped edges. Then, paths that share edges are split.

Two kinds of overlapping can appear. The first one occurs when two paths are overlapped (Fig. 3.8a). The second one occurs when a path is overlapped with two other paths (Fig. 3.8b). In the first case, the overlapping is solved taking in account two facts: the high variability (in Y axis) in its nodes and the distance between the starting node and the closest text line estimated in the localization step. Text lines that start in the middle of an estimated text line, will be penalized. In the second case, the overlapping is solved removing the path which is overlapped in the other two paths.

3.4 Word Segmentation

To extract the words form the text lines we improve the method of Manmatha and Rothfeder [123] for word segmentation based on a more accurate line segmentation approach. In [123] an effective word segmentation method for noisy historical documents is proposed. They propose a scale-space approach where the image is first dissected into lines using a grey-level projection profiles analysis technique. The line image is filtered with an anisotropic Gaussian filter at several scales in order to produce blobs which correspond to portions of characters at small scales and to words at large scales. The appropriate scale for finding words is automatically determined by optimizing a function over scale space.

We show the influence of the line segmentation step. When text lines are straight and close to the horizontal, methods based in the analysis of projection profiles work well both for line and word segmentation. But in several cases text lines present difficulties for such methods due the problems showed in Section 3.1. The objective is to replace the line segmentation in [123] with the line segmentation presented above.

The method is a continuation of the line segmentation approach explained above. The words of each text line are extracted using the method of Manmatha and Rothfeder [123] and we briefly summarize the used approach. The input is a grey-scale image, then, for each line, it is converted to grey-scale and used as input to the algorithm of word segmentation. The line image is then filtered with an anistropic Laplacian filter. The filtered output is thresholded to create a set of blobs. At a certain scale the blobs are more likely to correspond to words. This scale is automatically found by doing an optimization over scale space. The blobs at this optimum scale correspond to the words.

The last step is a post-processing stage were false positives are deleted and over-segmentations are corrected. The output of the word segmentation contains a set of connected components (CC) from each line. Assume that box b_j contains CC $\{a_1, a_2, \ldots, a_m\}$, where a_j represents both the CC and its area. We consider the box b_j its a valid box if

$$\frac{max\{a_1, a_2, ..., a_m\}}{a_j} > 0.1 \tag{3.9}$$

and

$$0.99 > \frac{width(b_j)}{heigth(b_j)} > 0.1 \tag{3.10}$$

To solve the problem of over-segmentation, two boxes (b_k, b_h) on the same line which overlap by more than 10%, assume that the box b_k contains CC $\{a_{k1}, a_{k2}, \ldots, a_{kn}\}$ and the box b_h contains CC $\{a_{h1}, a_{h2}, \ldots, a_{hn}\}$. We consider that both boxes allow to the same word if

$$max\{a_{k1}, a_{k2}, \ldots, a_{kn}\} == max\{a_{h1}, a_{h2}, \ldots, a_{hn}\} \tag{3.11}$$

This means that the largest area of b_k must be the same of the largest area of b_h.

We can observe the importance of a good line segmentation in the Table 3.3. This table shows the accuracy for the new approach of optimized line segmentation instead the projection based method that Manmatha and Rothfeder use in their work.

In the table, true positives are those for which a bounding box is generated for a real word. Missed words are those for which no bounding box is generated. Over segmentation occurs when two or more bounding boxes are generated for one word. Under-segmentation occurs when two or more words lie within one bounding box. Since the ground truth is generated on a per word basis, three words in a box count as three errors. Extra boxes are wrong boxes or false positives. The total errors column is the sum of errors in the other columns.

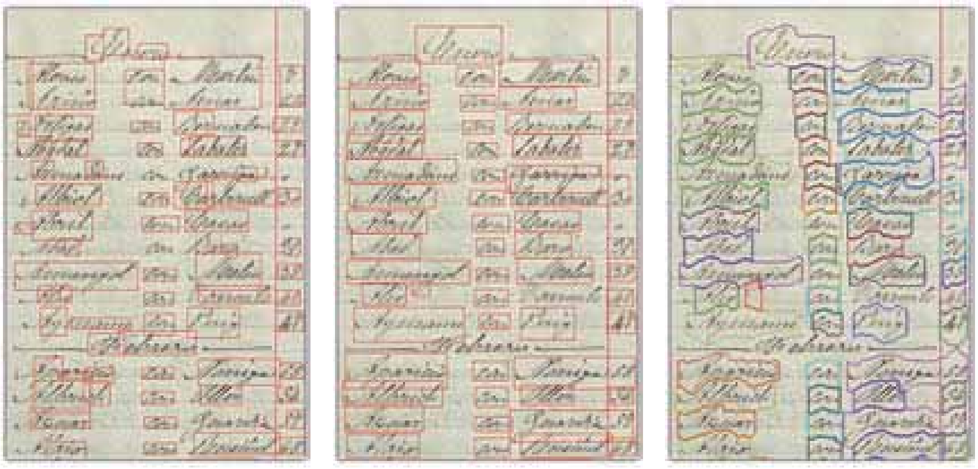
	TP(%)	MW(%)	OS(%)	US(%)	EB(%)	TE(%)
OLS+MR	79.35	4.78	0.53	14.97	0.37	20.65
OLS+MR+NOPP	73.37	1.84	0.82	14.88	9.10	26.63
MR	52.39	9.79	26.12	9.63	2.06	47.61

Table 3.3: Results combining an optimized line segmentation with Manmatha and Rothmeder's method (OLS+MR), combining an optimized line segmentation with Manmatha and Rothmeder's method (without post-processing) (OLS+MR+NOPP) and the results of Manmatha and Rothmeder's method (MR). The metrics used are: True Positives (TP), Missed Words (MW), Over Segmentation (OS), Under Segmentation (US), Extra Boxes (EB) and Total Errors (TE).

Using a post-process to remove extra boxes and over segmentation, we obtain better results. The number of words increase from 73% to 79%, the over segmentation is reduced, under segmentation is similar and extra boxes are reduced drastically from 9% to 0.37%. Instead, missing words increase when we apply post-processing. Applying post-processing we discard mostly extra boxes produced by the margins and noise in the document which may be detected as words. But, in some cases, the post-processing detects bounding boxes which are overlapping, as part of the same word, and they are joined.

Comparing our new approach with the work developed by Manmatha and Rothfeder with the same images, we observe a clear improvement. True positives increase from 52% to 79%, missed words decrease from 9% to 4%, over segmentation decrease drastically from 26% to 0.5% and extra words are also reduced. But the under segmentation increases using our method from 9% to 14%, because the words are close and, even for humans it is difficult to separate. Note that if a word is divided into two boxes it counts as two under segmentation errors. This explains the large number of under segmentation errors.

Figure 3.9b and 3.9a show the qualitative results using our optimized line segmentation and the Manmatha and Rothfeders method. We have used bounding boxes to show the results because second method uses bounding boxes. Although, the accuracy of the word segmentation is higher using an optimized line segmentation (3.9c).



(a) (b) (c)

Figure 3.9: Results using a projection base method to segment lines (3.9a) and results using an optimized line segmentation (3.9b). In both methods the words are extracted using the approach developed by the Manmatha and Rothfeder's. In image 3.9c observe the accuracy of the optimized line segmentation, instead of using bounding boxes.

A good word segmentation is an important step for the layout segmentation-based methods. The performance is better when it is used an accurate word segmentation approach because is the words are split wrongly, the results of the word spotting will be affected.

3.5 Experimental Results and Discussions

We have experimented with five databases with increasing level of difficulty: the datasets from ICDAR2009 [69] and ICDAR2013 [190] Handwritten Segmentation Contest, the UMD Database [84], George Washington's manuscripts [150, 153] and the Barcelona Historical Handwritten Marriages database (BH2M) (see Chapter 7). The metrics used to evaluate the performance of our approach are the ones used in the ICDAR2013 Handwritten Segmentation Contest.

3.5.1 Metrics

To make the results comparable, the performance evaluation method used in this work is the same that the used in ICDAR2013 Handwritten Segmentation Contest [190]. It is based on counting the number of matches between the entities detected by the algorithm and the entities in the ground truth. A *MatchScore(i,j)* table was used, representing the matching results of the *j-th* ground truth region and the *i-th* resulting region.

$$MatchScore(i,j) = \frac{T(G_j \cap R_i \cap I)}{T((G_j \cup R_i) \cap I)} \tag{3.12}$$

Let I be the set of all images points, G_j the set of all points inside the j ground truth region, R_i the set of all points inside the i result region, $T(s)$ a function that counts the points of set s.

A region means the set of foreground pixels likely to belong to a text line in each segment. A match is only considered if the matching score is equal to or above a specific acceptance threshold. Let N and M be the amount of pixels of the ground-truth and result elements respectively, and *o2o* is the number of one-to-one matches, we calculate the detection rate *DR* and recognition accuracy *RA* as in ICDAR2013 Handwritten Segmentation Contest [190]. Formally:

$$DR = \frac{o2o}{N}, \quad RA = \frac{o2o}{M} \tag{3.13}$$

A performance metric, called F-Measure *FM*, is computed combining the values of the detection rate (DR) and recognition accuracy (RA):

$$FM = \frac{2DR * RA}{DR + RA} \tag{3.14}$$

3.5.2 Datasets

In this work, we have used 5 different dataset. All of them cover a complete state of the on handwritten documents. We have used historical and modern databases, and the documents were written in different languages. The use of these datasets show the robustness of our method in front of different kind of documents.

ICDAR2009 and ICDAR2013. The documents of the ICDAR2009 text line segmentation contest, consisting of 200 document binary images with 4034 text lines, came from several writers that were asked to copy a given text. None of the documents include any non-text elements (such as graphical lines, drawings, etc.), and were written in several languages (English, French, German, and Greek). The documents of the ICDAR2013 text line segmentation contest are similar to the previous one. The main difference is that there are more languages involved by including an Indian language. It consists in 150 document binary images with 2649 text lines.

These databases were specifically created for a competition on line segmentation. The text lines are roughly straight and horizontal. There are only few documents that present multi-skewed text. The text lines are well separated and, in a few cases, we observe overlapping between ascenders and descenders from adjacent lines. The touching line problem only appears in some few cases. A sample of a handwritten document image of these datasets can be seen in Fig. 3.10a and 3.10b.

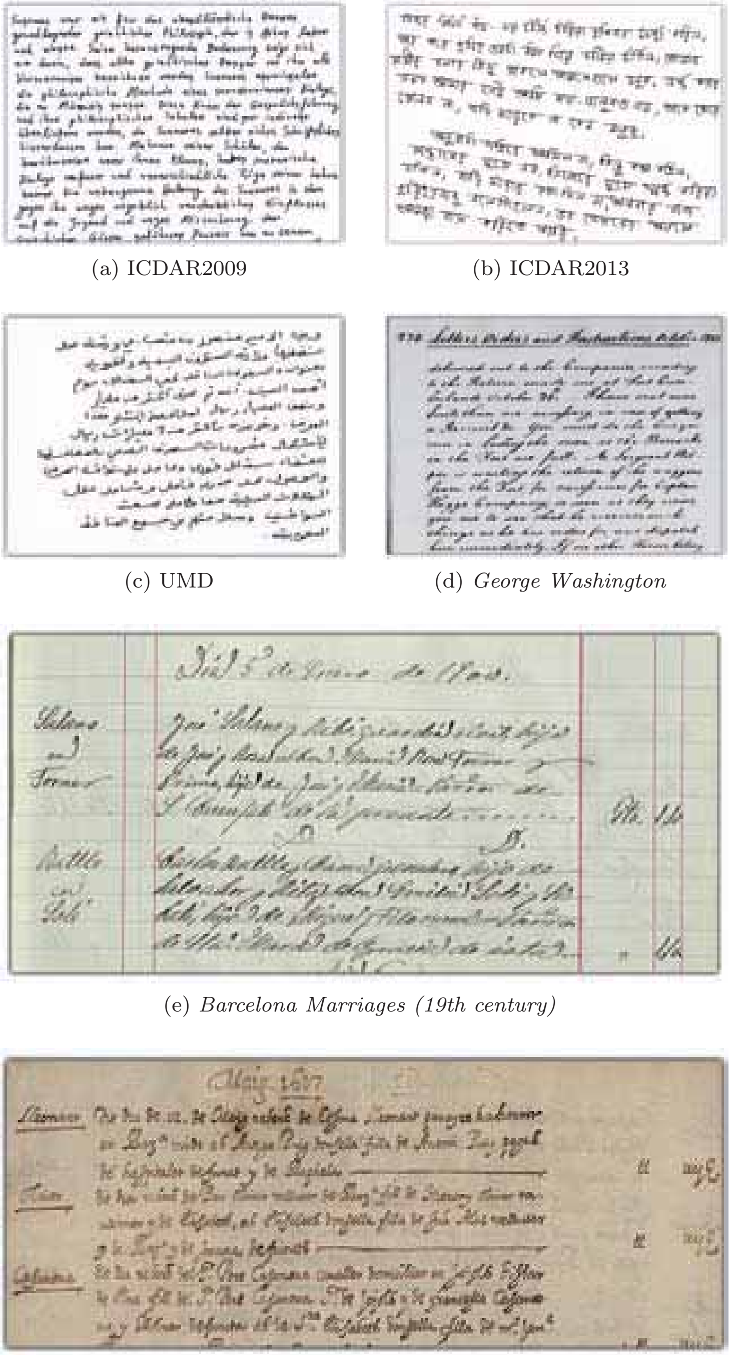
UMD. The UMD data set was collected by the members of the Language and Media Processing Laboratory at the University of Maryland. It consists in 123 Arabic documents, containing 1670 text lines. The images in this dataset present complex layouts and different levels of noise and degradation. We can observe an example of this database in Fig. 3.10c.

George Washington. This dataset consists of 20 pages and 715 text lines, from the George Washington collection. It is sampled from different parts of the original collection (at the Library of Congress). The images are in grey-level and scanned from microfilms. There are two writers in the 20 selected document pages. It is a well-known real historical handwritten database and, although the documents present good quality, some of the typical difficulties appear in some cases. The handwriting style in the Washington dataset is roughly straight and horizontal, and it contains ascenders and descenders from adjacent lines which are touching each other. We can observe an example of this dataset in Fig. 3.10d.

Barcelona Marriages. There are approximately 90, 500 pages, written by 244 different writers. The documents of this collection are in colour and they are degraded by lifetime and frequent handling. We show two examples of these documents in Figs. 3.10e and 3.10f. This collection presents an old handwriting style with all the difficulties of a real historical handwritten database (see Chapter 7): touching lines, large and big strokes with many overlapped characters between lines, horizontally-overlapping components and multi-skewed text lines. We have used two datasets in order to show the performance of our method with different handwriting styles. The first one consists of 30 documents with 964 text lines from the 19th century. The second one contains 94 documents with 3252 text lines from the 17th century.

3.5.3 Experiments and Results

We have performed five different experiments with the five datasets explained above, plus a synthetic dataset. Each experiment includes the results of the method pre-



(a) ICDAR2009

(b) ICDAR2013

(c) UMD

(d) *George Washington*

(e) *Barcelona Marriages (19th century)*

(f) *Barcelona Marriages (17th century).*

Figure 3.10: Samples of the datasets used in this Chapter.

sented in this work and the results of a classic method based in projections [158]. The objective of this comparison is to show the difference between localizing and segmenting text lines. We prove that, even our localization is coarse, we obtain a high accuracy in the text line segmentation. The parameters used in our method have been experimentally computed, but it is important to notice that we have used the same configuration for all the experiments. Therefore, we show how robust is the method to different collections using the same configuration. The values are $\alpha_1 = 0.61$, $\alpha_2 = 0.369$, $\alpha_3 = 0.001$ and $\alpha_4 = 0.20$.

ICDAR2009 & ICDAR2013 experiments. In the first experiment we have compared the results of our approach with the results of the participants of the ICDAR2009 and ICDAR2013 Handwritten Line Segmentation Contests and a classical line segmentation method based in projections [158] as baseline. The performance obtained using the ICDAR2009 and the ICDAR2013 datasets are shown in Table 3.4 and 3.5 respectively.

First, to have a baseline reference, the performance of a classical algorithm based on projections has been measured. It obtains a FM of 85.86% using the ICDAR2009 and 76.51% using the ICDAR2013 database. This low performance is because projection-based methods have difficulties segmenting handwritten documents with skew or touching components, and do not work properly when ascenders and descenders of two consecutive lines are horizontally-overlapped.

To better assess the performance of our method regarding the state of the art, in Table 3.4 we can see the comparison with all the methods presented in the ICDAR2009 Handwritten Line Segmentation Contest. These results are reprinted from the contest report [69]. Here we focus on the analysis of the most outstanding methods. The *CUBS* method is based on an improved directional run-length analysis. The *ILSP-LWSeg-09* method uses a *Viberti* algorithm to segment lines. The *PAIS* method is based on horizontal projections. And the *CMM* method uses labels to identify words of the same line. For more details on the rest of the methods the reader is referred to Section 3.2.

The main problem of the *CUBS* method is the overlapping of adjacent text lines. The *ILSP-LWSeg-09* method has the problem with the function that minimizes the distance is that it is different depending on the document. The *PAIS* is a projection-based method, and this kind of methods have a problem in highly multi-skewed text lines. Finally, the *CMM* method is a grouping-based method, so it fails to distinguish touching text lines.

Table 3.5 shows the comparison with all the methods presented in the ICDAR2013 Handwritten Line Segmentation Contest. These results are reprinted from the contest report [190]. Here we focus again on the analysis of the most outstanding methods. The *INMC* method is based in an algorithm of energy minimization using the fitting errors and the distances between the text lines. The *NUS* method uses a seam carving algorithm to segment lines. The *GOLESTAN* method divides the document in regions to compute the text lines using a 2D Gaussian filter.The *CUBS* method is based in a connectivity mapping using directional run-length analysis. And the *IRISA* method combines blurred images and connected components to segment the lines.

The *INMC* method has as main problem that needs a learning process to estimate

Table 3.4: Evaluation results using the ICDAR 2009 Database, where *M* is the count of result elements, *o2o* is the number of one-2-one matches, *DR* is the Detection Rate, *RA* is the Recognition Accuracy and *FM* is the harmonic mean. The number of ground-truth elements *N* is 4034.

	M	o2o	DR(%)	RA(%)	FM(%)
CUBS	4036	4016	99.55	99.50	99.53
ILSP-LWSeg-09	4043	4000	99.16	98.94	99.05
PAIS	4031	3973	98.49	98.56	98.52
CMM	4044	3975	98.54	98.29	98.42
CASIA-MSTSeg	4049	3867	95.86	95.51	95.68
PortoUniv	4028	3811	94.47	94.61	94.54
PPSL	4084	3792	94.00	92.85	93.42
LRDE	4423	3901	96.70	88.20	92.25
Jadavpur Univ	4075	3541	87.78	86.90	87.34
ETS	4033	3496	86.66	86.68	86.67
AegeanUniv	4054	3130	77.59	77.21	77.40
REGIM	4563	1629	40.38	35.70	37.90
Proposed	4176	3971	98.40	95.00	96.67
Base line (Project.)	4081	3834	86.36	86.37	85.86

a cost function that imposes the constrains on the distances between text lines and the curvilinearity of each text line. The *GOLESTAN* method localize text lines, the segmentation is done obtaining the dilation of synthetic paths, which represents the localization of the text lines. Difficulties as touching lines and overlapping are not solved. The *IRISA* method localize properly the handwritten text lines, the problems of crossing lines and overlapping are considered, but the touching lines problem is omitted.

Our method has obtained a FM of 96.67% and a 95.43% on respective databases. It is worth noticing that the performance of some methods, having a high performance, decreases when they are applied to other document collections, especially historical ones. An example is the *CUBS* method which in the ICDAR2009 got a FM of 99.53% and a 97.45% in the ICDAR2013. But nevertheless, the method presents robustness in front of different databases. In the following paragraphs we will show how our method is robust under different conditions, and how the performance keeps in a reasonable level even when the distortion is high in some other historical datasets.

The objective of the next experiments is to show that our method is addressed to segment lines accurately and not only the localization.

UMD experiment. In this experiment we have evaluated the performance of our method with respect the classical approach based in projections, using the UMD

[3] This method was a preliminary version of the approach presented in this work. The method proposed in this work increases the accuracy of the touching-components segmentation with an improved version of the heuristics.

Table 3.5: Evaluation results using the ICDAR 2013 Database. The number of ground-truth elements N is 2649.

	M	o2o	DR(%)	RA(%)	FM(%)
INMC	2650	2614	98.68	98.64	98.66
NUS	2645	2605	98.34	98.49	98.41
GOLESTAN-a & -b	2646	2602	98.23	98.34	98.28
CUBS	2677	2595	97.96	96.94	97.45
IRISA	2674	2592	97.85	96.93	97.39
TEI (SoA)	2675	2590	97.77	96.92	97.30
LRDE	2632	2598	96.94	97.57	97.25
ILSP (SoA)	2685	2546	96.11	94.82	95.46
QATAR-b	2609	2430	91.73	93.14	92.42
NCSR (SoA)	2646	2447	92.37	92.48	92.43
QATAR-a	2626	2404	90.75	91.55	91.15
MSHK	2696	2428	91.66	90.06	90.85
CVC[3]	2715	2418	91.28	89.06	90.16
Proposed	2697	2551	96.30	94.58	95.43
Base line (Project.)	2430	1836	77.50	75.55	76.51

Table 3.6: Evaluation results using several approaches in *UMD* Documents. The number of ground-truth elements N is 1951.

	M	o2o	DR(%)	RA(%)	FM(%)
Proposed	**2189**	**1814**	**92.97**	**82.86**	**87.63**
Base line (Project.)	2314	1270	65.09	54.88	59.55

dataset. The performance obtained using the UMD dataset is shown in Table 3.6. Our approach obtained a FM of 87.63% and the baseline method based in projections obtained a FM of 59.55%. The performance of simple projection-profile analysis methods fails also in Arabic documents, even more if the present touching lines in their documents. The results of the UMD database can be compared with the results of the work [23]. They compare their work with several datasets of the literature and with different databases (included the UMD database). We observe low performance results (73.52%) using the UMD database. This method makes an accurate localization of the lines, but fails in the segmentation when the lines present touching components.

George Washington experiment. In this experiment we have evaluated the performance of our method with respect the classical approach based in projections, using the George Washington dataset. The performance obtained using the George Washington Dataset is shown in Table 3.7. Our approach obtained a FM of 92.70%, slightly lower than in the ICDAR2009 and in the ICDAR2013 dataset. The baseline method based in projections obtained a poor FM of 46.70%.

Table 3.7: Evaluation results using several approaches in *George Washington* Documents. The number of ground-truth elements N is 715.

	M	o2o	DR(%)	RA(%)	FM(%)
Proposed	**693**	**653**	**91.30**	**94.20**	**92.70**
Base line (Project.)	727	338	47.20	46.40	46.70

Table 3.8: Evaluation results using several approaches in *Barcelona Marriages* Documents (19th century). The number of ground-truth elements N is 964.

	M	o2o	DR(%)	RA(%)	FM(%)
Proposed	**981**	**964**	**83.00**	**81.60**	**82.20**
Base line (Project.)	1276	630	65.30	49.30	56.10

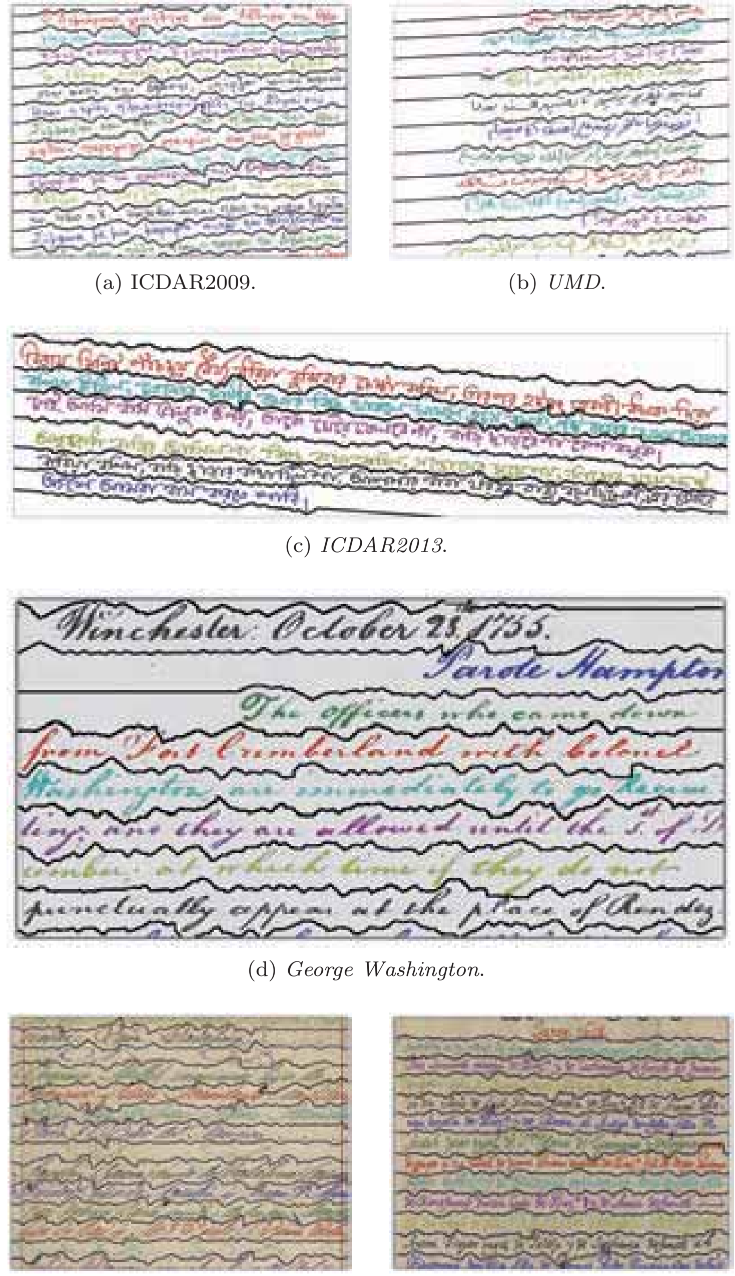
Barcelona Marriages experiment. In the last experiment, we have compared the performance of our method with the results obtained from the classical approach based in projections, using the two Barcelona Marriages datasets. Tables 3.8 and 3.9 show the results obtained in the datasets of the 19th and 17th century respectively. The results using this dataset are good taking into account the quality of the documents. We have obtained a *FM* of 82.20% using the dataset of 19th century and a FM of 86.3% with the dataset of 17th century. These FM rates are lower than the obtained using the George Washington dataset. We have also computed the FM rate using the classic method based in projections and the results are poor (56.10% and 63.60% respectively). As it was expected, the performance of simple projection-profile analysis methods, that are standard techniques in machine-printed documents, is very poor in historical manuscripts with variations in the script styles, lines that touch and overlap ones to the others, noise, etc.

In Fig. 3.9 the reader can observe some qualitative results obtained in the five databases. As we can observe the Barcelona Marriages datasets are more complex than the other two datasets, and sometimes touching components are not properly segmented. However, the problem of horizontally-overlapping components is solved in most of the cases (Fig. 3.11f). Our method can find a path through the ascenders and descenders of the text lines without any problem. We can observe that our method properly segments documents containing text lines of different length (Fig. 3.11d and 3.11e) and skew (Fig. 3.11a and 3.11c).

Other experiments. To evaluate the robustness of our method in front of the typical difficulties of handwritten documents, we have generated some synthetic images (Fig. 3.12) from the ICDAR2009 dataset. The first difficulty appears when the doc-

Table 3.9: Evaluation results using several approaches in *Barcelona Marriages* Documents (17th century). The number of ground-truth elements N is 3252.

	M	o2o	DR(%)	RA(%)	FM(%)
Proposed	**1013**	**865**	**87.40**	**85.30**	**86.30**
Base line (Project.)	1064	651	66.40	61.10	63.60



(a) ICDAR2009.

(b) *UMD*.

(c) *ICDAR2013*.

(d) *George Washington*.

(e) *Barcelona Marriages (19th century)*.

(f) *Barcelona Marriages (17th century)*.

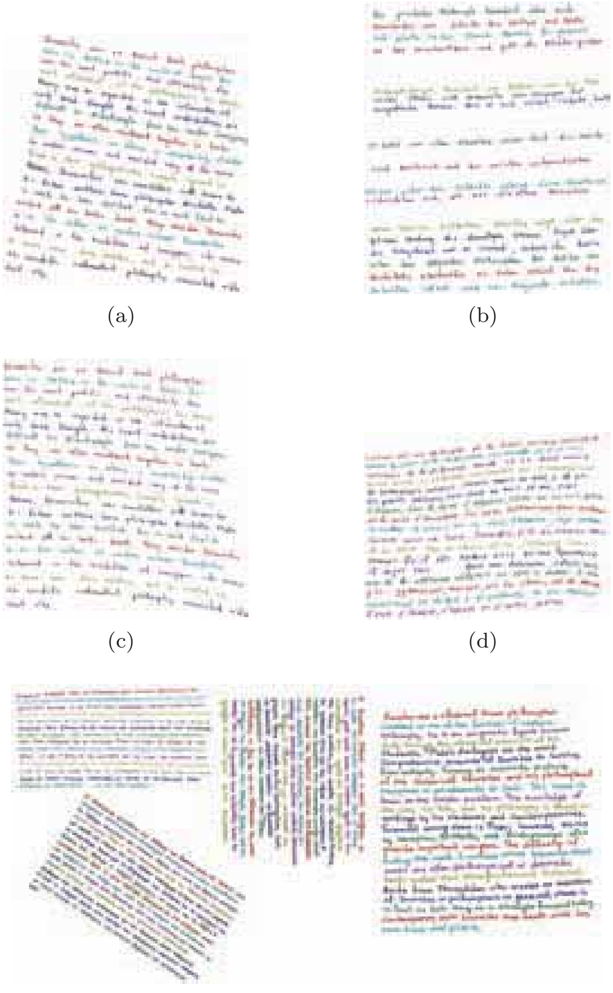
Figure 3.11: Qualitative results obtained using our approach.

ument presents a high skew (Fig. 3.12a). We have rotated −5 degrees document. The second one appears when the line spaces between text lines are not homogeneous (Fig. 3.12b). We have spaced the lines using different sizes. Sometimes the text lines present a falling curvature (Fig. 3.12c), or text lines have different orientation between them (Fig. 3.12d). Finally, some documents present several text blocks, even in different orientations (Fig. 3.12e). These three images have been manually modified. As the objective of this work is the line segmentation in text blocks, and not the layout segmentation, we have used the approach developed by Cruz et al. [34] which is oriented to extract text blocks in historical documents. So, the segmentation method has been applied after segmenting each text block. In the same image we have introduced text with 0, 90 and 45 degrees. We can observe that in all these cases our approach properly segments the documents.

Our method is also able to cope with noisy documents as we can observe in Fig. 3.13. The presence of noise influences on the computation of the skeleton of the background image. However the variation in the skeleton does not influence on the segmentation of the text lines. We can observe the results obtained using a clean image in Fig. 3.13a and a salt-and-pepper noisy image in Fig. 3.13b. In this figure we have simulated the case when the document is too much noise and the pre-process cannot clean the document completely. The result of the pre-process is a noisy image, but nonetheless, the segmentation is done in a proper way.

Some documents contain spots and show-through that can be a problem in the segmentation process. As we can observe in Fig. 3.14 our method allows to solve this problem thanks to the pre-processing step. In Fig. 3.14a and 3.14c we observe some noise in the document (spots), which can slightly modify the path, as we can observe in the Fig. 3.14a, but, in both cases, our approach finds a path to segment the lines. There are some cases where the documents present drawings and graphical lines (Fig. 3.14b). Our method does not remove this kind of noise, but segments the lines searching a path between these graphical elements. The problem of show-through is showed in the Fig. 3.14d. The pre-process solves this problem and the paths are properly computed. For more complex cases we can use specific pre-processing methods to remove spots, graphical lines, drawings and show-through. We have removed all the steps to clean the image included in the pre-process in all the experiments presented above. We have only done the binarization of the images. The objective is to simulate the noisy documents where the pre-process cannot remove the noisy completely. We show the good performance of the approach presented in this paper in this kind of documents.

There are two main problems that can vary the number of initial nodes regarding the number of text lines (Fig. 3.15). The first problem appears when there are comments between the text lines. They are usually smaller than the size of the text lines and they are completely touching the above and below text lines (see Fig. 3.15a). In these cases the paths to segment the lines (upper, bottom and in-line text line) are almost overlapped, and the consistency checking process will remove the most overlapped path. Consequently, it joins one of the text lines with the in-line text. The second problem appears when the length of the text line is very short (compared to the rest of the lines) and it is horizontally overlapped (Fig. 3.15b). Then the segmentation of the text lines becomes ambiguous because it is difficult to determine



(a) (b)

(c) (d)

(e)

Figure 3.12: Qualitative results over synthetic images generated from ICDAR2009 dataset. (a) Document with a pronounced skew. (b) Document with different line spacing between text lines. (c) Document where the text lines fall. (d) Document with multi-oriented text lines. (e) Document with several text blocks, each one with a different orientation.

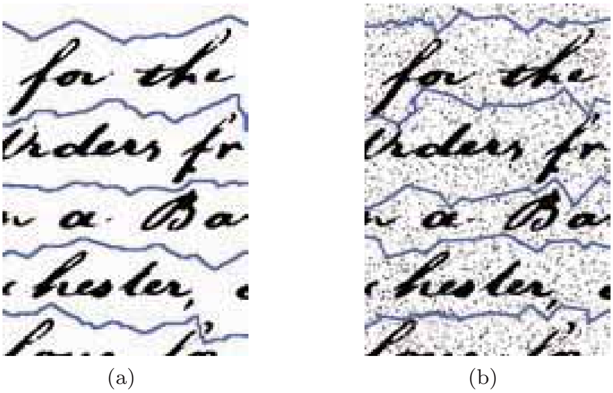


Figure 3.13: The influence of noise in line segmentation. (a) Clean image. (b) Salt & pepper noise image.

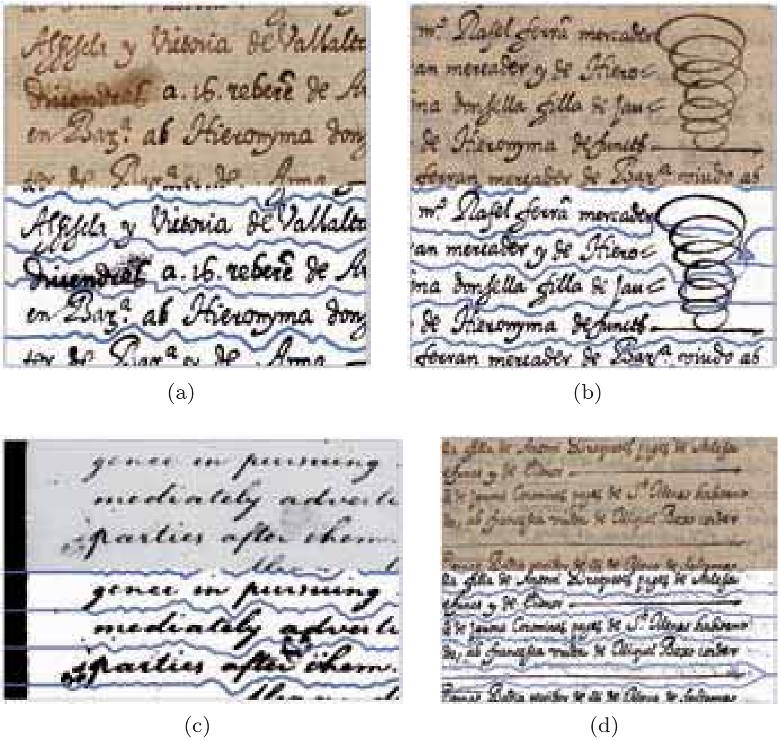
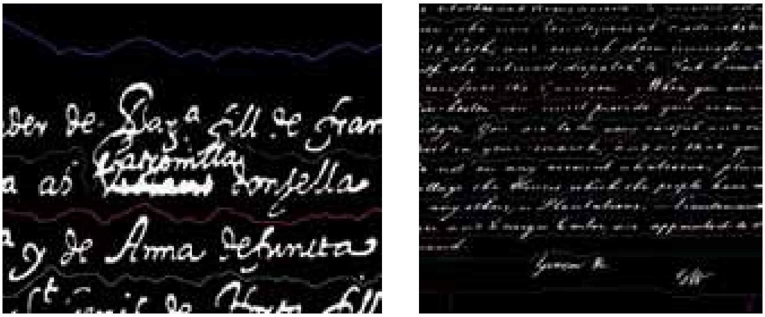


Figure 3.14: The influence of noise in line segmentation: spots, blobs, graphical lines and show-through problems.



(a) Text comments between the text lines.

(b) Short horizontally overlapped text lines.

Figure 3.15: Difficulties that can vary the number of initial nodes.

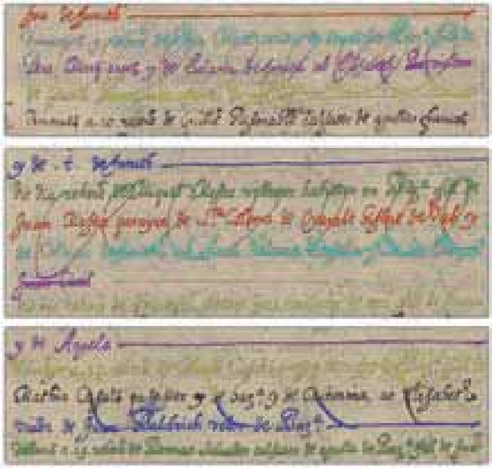


Figure 3.16: Segmenting non-text elements.

whether text fragments belong to the same text line or not. These two problems increase the difficulty of segmenting text lines and it can be objective of specific works. However, since the number of these specific cases is very low in the databases used in our experiments, this is not a critical issue.

Finally, the presence of non-text elements like graphical lines or drawings is not a handicap for our method because it does not really depend on a script or text like writing style, but only on the structure. In the Barcelona Marriages dataset it is usual to end text lines with an horizontal stroke, or to cross out wrong registers. Our approach is able to tackle with this difficulty as we can observe in Fig. 3.16.

3.5.4 Discussion

The configuration used in the experiments has been computed experimentally using the ICDAR2009 dataset. This configuration has been used in all the experiments independently of the dataset. This fact shows the robustness of our method in front of different types of handwritten documents, whether they are historical or modern.

We can observe the evolution of the performance of the methods using the five databases: the two first ones are datasets where the problem of touching, skew and horizontally-overlapping text lines is scarce (ICDAR2009 and ICDAR2009 dataset). The second one is dataset written in Arabic. It is written using a left justification and presents touching and overlapping in a few cases. The third one is a historical dataset with some noise introduced by the life-time, and with some of the problems explained before (George Washington dataset) The last one is a dataset with noisy and degraded documents, where the percentage of touching components and horizontally-overlapping is very high (Barcelona Marriage dataset).

The last experiment shows the robustness of our method in front of the typical difficulties in handwritten documents. The method is able to cope with noisy documents and with documents that contain drawing lines or comment lines.

The overall conclusion is that a baseline method based on classical projection profile analysis does not work well with manuscripts. The main difficulties for this kind of methods were anticipated in the introduction: the more is the skew and degree of touching and overlapping between consecutive lines, the lower is the accuracy of the segmentation. In general, projection profiles detect the approximate position of the text lines, but do not allow an accurate segmentation. When analysing other methods designed for handwritten line segmentation, the proposed approach is ranked in the top positions. The robustness of the method was proved when it was applied to databases with increasing levels of difficulty. Our method kept its performance over 80% of *FM*. Hence we can conclude that the proposed method is highly able to cope with the different types of problems that appear in historical documents, in particular with multi-oriented lines, overlapped components and touching lines. Conversely, as it is observed in [125], the winner method of the *ICDAR2009* Handwritten Segmentation Contest with a *FM* of 99.5%, drastically decreases the performance to a *FM* of 56.1% using low resolution and noisy handwritten documents.

3.6 Conclusions

In this chapter we have presented a robust line segmentation approach, which segments lines in any kind of handwritten documents. It is not designed for a specific category of documents, coping with both historical and modern ones. The proposed approach finds the path which is located at the same distance in the area between two consecutive text lines. The skeleton of the background image is used to convert it to a graph. This graph is used to find the best path to segment text lines using a minimum cost path-search algorithm.

One of the key contributions is the ability of the method to segment lines pixel-wise and not only locate then, as some approaches in the literature.

We have tested the method in five databases with different difficulties: ICDAR2009, UCDAR2013, UMD, George Washington and Barcelona Marriages database. With this extensive experimentation, we have proved that our method is able to deal with different conditions of degradations and in front of modern and historical documents. Even in the worst scenario, such as the Barcelona Marriage datasets that present many difficulties, our approach obtains a FM of 82.20%, while other state of the art methods dramatically decrease their performance when the documents contain skewed or multi-oriented, touching and overlapping lines.

In summary, a robust method to segment handwritten lines that outperforms the state-of-art (Table 3.1) in historical documents have been presented. The method works in an unsupervised framework so it is writer invariant. It tackles with multi-skewed, touching and horizontally-overlapped lines.

The main objective of this thesis is a contextual segmentation-based word spotting. The words of the text line segmented are necessary for the method. Once the text line segmentation approach presented in this Chapter is applied, words can be extracted using any state of the art method. In our case we have shown the importance of the line segmentation in the Section 3.4, where the method of word segmentation of Manmatha and Rothefeder is improved applying our line segmentation method before their algorithm.

Chapter 4

Word Spotting

Word spotting is a content-based retrieval strategy that, due to the impossibility of a recognition process with enough quality, lean towards to a visual object detection approach. The key idea of word spotting relies upon representing word images with robust features and a subsequent classification scheme. The chosen feature space is a crucial decision. It must be representative and scalable enough to distinguish among a high number of classes (words) but invariant to the inherent variations within the same class (noise, distortion of handwriting, writing styles, etc.). In this chapter we evaluate the word spotting approaches of the state of the art and we show the importance of choosing the proper descriptor and the key points. We show the performance using different configurations of key points. And finally, we propose a new pseudo-structural descriptor based in local features oriented to historical handwritten documents.

4.1 Introduction

Given a set of documents, Handwritten Word Recognition (HWR) is the general task of creating a computer readable representation. Word spotting, in contrast, is the related information retrieval task. The goal is to create a ranked list of all documents, sorted according to their relevance, which is the likelihood of the keyword occurring in the document. Due to its effectiveness, word spotting has been largely used for historical document indexing and retrieval, not only for old printed documents [128], but also for old handwritten ones [66, 93, 107, 108, 133, 152].

It is important to remark that, although the use of a language model is very common in HWR, it may be worthless when dealing with historical documents. It may be due to different reasons: the lack of enough training data to compute lexicon frequencies, unseen vocabularies (names, cities) or non stable over the time, “on the fly” querying (the user selects a collection and wants to search an arbitrary word in it), etc. Word spotting is a content-based retrieval strategy where, due to the impossibility of a recognition process with enough quality, leans to a visual object detection approach. The key idea of word spotting relies upon representing word images with robust features and a subsequent classification scheme. The chosen feature space and the selection of the proper key points are a crucial decision.

The research done in word spotting can be distinguished according to different properties. From the point of view of the task word spotting approaches can be categorized for document filtering [157] and retrieval purposes [6]. Taking into account the input query, there are three classes: Query-by-String [67], Query-by-Class [172] and Query-by-Example [165]. There are some methods in the literature that need a learning process to compute the model for the matching step [60], and other methods that are learning-free [169]. Finally, the last classification of methods is done taking into account if they need a layout analysis step to segment the words (segmentation-based) [194], or if the methods search the words directly in the documents without a segmentation step (segmentation-free) [165].

Next, we briefly review most of the representative works of handwritten word spotting. For more details, we encourage the reader to read Chapter **??** for a deep study of the different categories showed above.

The task of word spotting as detecting a word in an image has been initially proposed in [104] for printed text and a few years later by Manmatha et al. [122] for handwritten text. The first methods adopted approaches common in optical character recognition (OCR) and used of pixel-wise comparison of the query and the test image (or selected parts thereof, called zones of interest (ZOI)). Notable works in this domain include XOR comparison [121], Euclidean distance [150], Scott and Longuet-Higgins distance [122], Hausdorff distance of connected components [117], the sum of Euclidean distances of corresponding key points (corner features) [166], and feature extraction using the moments of the background pixels [19].

There is a fundamental issue that not always gets the deserved relevance. It is the primitives, or interest points, on which the features are computed. Two families can be differentiated, namely appearance-based and object-based primitives.

Appearance-based methods extract descriptive features from all the image pixels in terms of the photometry. Different arrangements can be considered when analyzing the pixels, so an implicit spatial information is encoded. A typical implementation is inspired by spatial pyramid methods where the descriptors are extracted on a regular grid and different scales. Almazan et. al [4] divide the images in equal-sized cells. For each one a HOG descriptor is computed combined with an exemplar-SVM framework. Gatos et. al [69] perform a template matching of block-based image descriptors. Rothacker et. al [165] localize the descriptors on a regular grid and uniform scale. The feature vector is constructed using a dense SIFT descriptor. Almazan et. al [5] adopt the Fisher Vector (FV) representation computed over SIFT descriptors densely from the word image. Other methods use column-wise feature descriptors. Frinken et. al [67] compute global and local features in each column. Rodriguez-Serrano et. al [159] combine Marti and Bunke [129], Zoning [199] and LGH [156] features. These methods train first the models, using the information of the entire image. Once the model in trained, the images are compared and the candidates are ranked using a similarity measure, commonly a Dynamic time Warping (DTW) or Hidden Markov Model (HMM-based) similarity. Object-based methods segment local interest points from the image, and extract features on each individual object. As in other pattern recognition domain, typical interest points in images are key points like corners or crossings, edges, skeletons or regions.

The main objective of this thesis is show the improvement of the word spotting approaches when we introduce the semantic information into the search. So, we engage in the study of several descriptors designed for handwritten word spotting. We classify the state of the art methods in different categories according to the classical taxonomy that divides pattern recognition into statistical and structural approaches. We compare three word representation models, namely sequence alignment using DTW as a baseline reference, a bag of visual words approach as statistical model, a pseudo-structural model based on a deformable HOG-based shape representation, and a structural approach where words are represented by graphs.

As stated above, the first objective of this chapter is the study of the different methods of word spotting in the literature, but the performance of a handwritten word spotting approach does not only rely on the features but also on the interest point model over which the features are computed. In this chapter we present a study of the different interest points that can be used for the extraction of the features.

Finally, we propose a new descriptor that addresses the problem of handwritten word spotting in historical documents following a query-by-example strategy. The descriptors of the state of the art are oriented to modern handwritten documents without all the difficulties that are present in historical documents. We propose a holistic approach, using shape matching techniques that perform good results in historical and modern documents. Our approach is inspired in Loci characteristics [70] and allow to aggregate spatial and pseudo-structural information in the descriptor.

The rest of the chapter is structured as follows. In section 4.2 is done a review of the state of the art. Section 4.3 describes the key points which are selected for the evaluation of the interest points. Section 4.4 describes the proposed descriptor. Section 4.5 shows the experimental results. Finally, we present the conclusions in the last section of the chapter.

4.2 Analysing Word Spotting Approaches

First we will study several word spotting approaches in the literature. We analyse the performance of the descriptors when dealing with word recognition. We compare four families of morphological models applied to word classification in a QBE word spotting framework in historical manuscripts. The performance of the descriptors is therefore not only assessed in terms of the retrieval quality, but also other attributes like their computational cost and their ability of being integrated in a large scale retrieval application.

As baseline model, we have selected the well known approach of Rath and Manmatha [150, 153] where word images are represented as sequences of column features and aligned using a Dynamic Time Warping algorithm. We have chosen this representation because it can be considered the first successful handwritten word spotting approach in the state of the art.

In addition, other description models are analysed according to the classical taxonomy that divides pattern recognition into statistical and structural approaches. Hence, we present three different description models that can be considered statistical, pseudo-structural or hybrid, and structural, respectively. The three models proposed in the evaluation are: the bag of visual words representation based on SIFT features proposed by Rusiñol et al. [169], as statistical descriptor; the deformable HOG-based shape descriptor (nrHOG) presented by Almazan et al. [3] , as pseudo-structural representation; and finally a graph-based representation following the work of Dutta et al. [47], as structural descriptor.

4.2.1 Representation Models for Handwritten Words

As we have introduced above we have implemented three models as representative of different classes to evaluate the importance of shape word representation in a word spotting context. This section overviews the different approaches. The reader is referred to the corresponding original works for detailed descriptions.

Baseline Model

As a reference system, we have chosen the well-known word spotting approach described by Rath and Manmatha [150, 153], which is based on the Dynamic Time Warping algorithm. The Dynamic Time Warping (DTW) algorithm was first introduced by Kruskal and Liberman [100] for putting samples into correspondence in the context of speech recognition. Word images are represented as a sequence of column-wise feature vectors, and DTW can distort (or warp) the time axis, compressing it at some places and expanding it at others, finding the best matching between two samples.

Bag-of-visual-words Descriptor

The representation of the word images is based on the bag-of-visual-words (BoVW) model powered by SIFT [116] descriptors. Firstly a reference set of some of the word images is needed in order to perform a clustering of the SIFT descriptors to build the codebook. For each word image in the reference set, the SIFT descriptors is densely calculated over a regular grid of 5 pixels by using the method presented by Fulkerson et al. [68]. Three different scales using bin sizes of 3, 5 and 10 pixels size are considered. These parameters are related to the word size, and in this case have been experimentally set.

Once the SIFT descriptors are calculated, by clustering the descriptor feature space into k clusters we obtain the codebook that quantizes SIFT feature vectors into visual words. The k-means algorithm is used to perform the clustering of the feature vectors. In the experiments carried out, a codebook is used with dimensionality of $k = 20.000$ visual words.

The SIFT descriptors are extracted from the word images, and we quantize them into visual words with the codebook. Then, the visual word associated to a descriptor corresponds to the index of the cluster that the descriptor belongs to. The BoVW feature vector for a given word is then computed by counting the occurrences of each of the visual words in the image.

The main drawback of bag-of-words-based models is that they do not take into account the spatial distribution of the features. In order to add spatial information to the orderless BoVW model, Lazebnik et al. [106] proposed the Spatial Pyramid Matching (SPM) method. This method roughly takes into account the visual word distribution over the images by creating a pyramid of spatial bins.

Deformable HOG-based Shape Descriptor

The nrHOG feature extraction approach is based on the computation of HOG features in a given set of k x k points, denoted as focuses, over the shape to be described. These focuses, which can also be seen as an adaptable mesh, are automatically positioned with the objective of being distributed along the shape pixels.The approach has two sequential steps: a first step devoted to compute the location of the focuses following an iterative region partitioning algorithm and a second step where regions centred over the focuses are extracted and described using HOG features.

Graph-based Descriptor

For the Graph-based descriptor a graph matching approach [47] is adapted to the word spotting problem. The graph representation for a word image is constructed from its skeleton information. The skeleton of the image is polygonally approximated using a vectorization algorithm [164].

Once words are represented by graphs, word spotting is solved by a graph matching algorithm. To avoid the computational burden, a method based on graph serialization is proposed. Graph serialization consists in decomposing graphs in one dimensional structures like attributed strings, so graph matching can be reduced to the combination of matchings of one dimensional structures.

4.3 Key Point Evaluation

The starting hypothesis of this study is that the performance of a handwritten word spotting approach does not only rely on the features but also on the interest point model over which the features are computed. We have compared different segmentation strategies to extract interest points. In particular, we have extracted the descriptor from foreground pixels, background pixels, local extrema (end points, corners and crossings), contours and skeletons. For each interest point scheme, we have computed features using the descriptor presented in this chapter (Section 4.4) and the Shape Context descriptor [16].

Let us further describe the four key points that are considered to compute the features.

Local extrema

Key points are computed using local extrema points based in the work of España-Boquera et. al [53]. The proposed method consists in automatically detecting a set of points from the image and classifying them by supervised machine learning techniques

[71, 185]. The computation of local vertical extrema of foreground pixels is done in three steps: first, the contour of the image is obtained by searching positions within a column between a background and foreground pixel; second, the selected points are grouped into lines following a proximity criterion; and finally, the maxima of the upper contour and the minima of the lower contour are computed using sliding window and checking whether the central point is the maximum (or minimum) of the window.

The classification of the points in five classes (ascender, upper, lower, descender and the rest of the points) is done using two Multilayer Perceptrons (MLP) (see Fig. 4.1a). The first of the MLPs has two outputs with a softmax activation function to determine whether the input data is a lower baseline point or not, and the other MLP has five outputs (with also a softmax activation function) corresponding to the five classes previously described: ascender line point, upper baseline point, lower baseline point, descender line point and any other point.

A feature vector is computed for each key point. Then, and in order to split the key points in the three regions, the upper and lower points are classified as part of the main body area.

Skeleton-based

In this case, key points are extracted from the skeleton of the word image, as Wang et. al propose in [202].

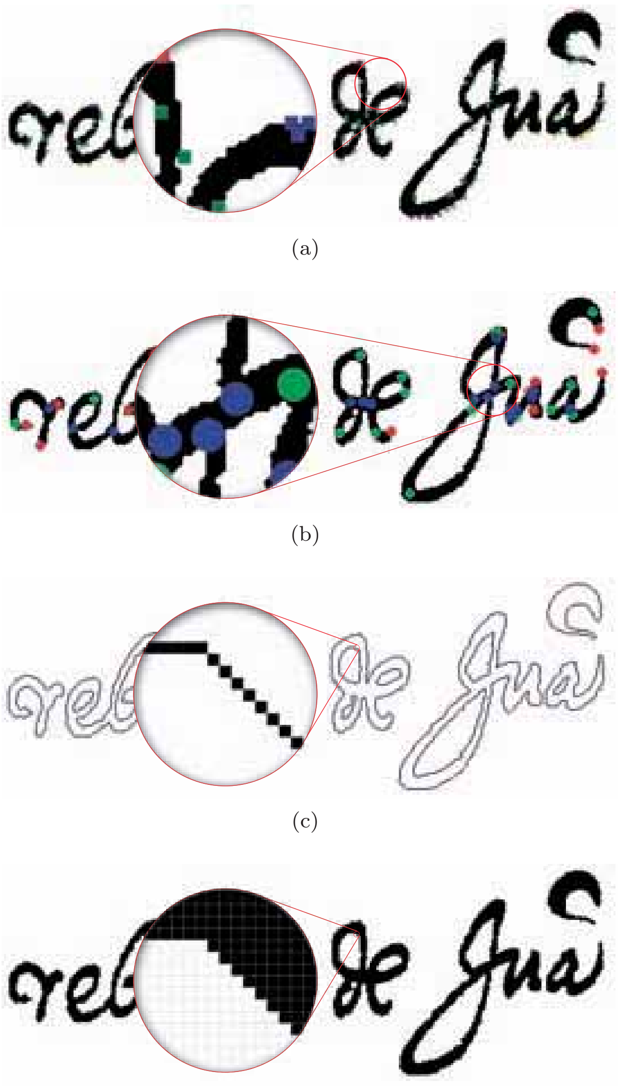
After obtaining the skeleton of the text, structural interest points are detected. The interest points are referred to three types of points. They are respectively starting and ending points, branch points including junction and crossing points and high-curved points (see Fig. 4.1b). Since the skeleton is one-pixel width, for each black pixel (skeleton pixel), a 3x3 mask is applied to check the nearest 8 neighbours of the pixel. If there is only one black pixel among 8 neighbours, the reference pixel is considered as a starting or ending point. For branch points, they employed Hit-and-Miss transformation [215], a basic binary morphological operation, which is generally used to detect particular patterns in a black-and-white image. To detect high-curved points, the curvature of a point is estimated using the angle between two vectors.

Contour

The contour of the words can be used as key points. The subtraction of the eroded image is computed to get the contour. Every pixel of the contour is used as key point of the word image (see Fig. 4.1c).

Background and Foreground

The key points can be the foreground and background of the word image. Thus, all background/foreground pixels are used as key points (see Fig. 4.1d).



(a)

(b)

(c)

(d)

Figure 4.1: Examples of the different interest points used. (a) Local extrema point detection (red point – ascender point, blue point – upper point, yellow point – lower point, purple point – descender point and green point – the rest of the points), (b) Structural key point detection (red point – starting/ending point, green point – high-curved point, blue point – branch point), (c) Contour key points, (d) Foreground and background key points.

4.4 Loci-based descriptor

We have classified the word spotting approaches in three categories: statistical, pseudo-structural (or hybrid) and structural approaches. The first approaches usually need a costly learning process to create a model to classify the words. But, sometimes there is a lack of labelled documents and it is not possible to train a model. Structural approaches are focussed in the extraction of the features using local characteristics of the images, like skeletons, intersections, etc. Later, the similarity is computed using this features, or converting them to other structures like graphs. But, in any case, the similarity is to small sensitive variations in the features. On the other hand, pseudo-structural methods use the features of the structural methods, and analyse the region using statistical features. This kind of approaches is more robust because analyses regions of the image, instead of centring the descriptor in a characteristic of the image. In addition, this kind of methods does not need to construct a training model.

We propose a descriptor based on pseudo-structural features. Our approach is inspired in characteristic Loci feature [48, 70]. Given a word image, a feature vector based on Loci characteristics is computed at some characteristic points. Loci characteristics encode the frequency of intersection counts for a given key-point in different direction paths starting from this point. As key-points are used contours, foreground pixels, background pixels or skeletons, depending on the application.

The spotting strategy proposed in this chapter has two requirements that can be separated in two major modules: the learning and the retrieval stage (Fig. 4.2). First, word images have to be mapped to a feature space using a Loci-based descriptor. Second, it is necessary to design an indexation structure allowing to formulate queries of word images and to retrieve similar instances from the database in terms of shape similarity. The indexation structure is organized in a hashing-like way where features are encoded as index keys and words are stored in a hashing structure. Hence, features describing words have to be chosen so that they allow to cluster the space in classes in an unsupervised way, and also the features have to satisfy properties as coping with distortions and obtaining compact representations.

The system consists of five stages (Fig. 4.3). The first step normalizes the word images. The objective is to locate the word in the centre of a template blank image, according to its centre of mass. Otherwise, two words with different lengths would have different grid sizes, resulting in different feature vectors for similar word parts. The second step of our system is a region extraction. For the sake of assessing the influence of key points in word spotting, the BoW scheme could be applied to the whole image features. However, it does not preserve spatial information. Due to this we divide word images into regions, so the features are associated to each region. Horizontally, the word image is divided in three equal-size regions. Vertically, the word image is divided in n equal-size regions. At the end we have an irregular grid, so the associated features for each cell have a rough spacial meaning.

The next step consists in the extraction of the key points. We can use several key point extraction methods of the literature. In Section 4.3 the influence of different categories of interest points is assessed. In particular, four methodologies are used to detect them in the regions of the word image. The first one uses local extrema

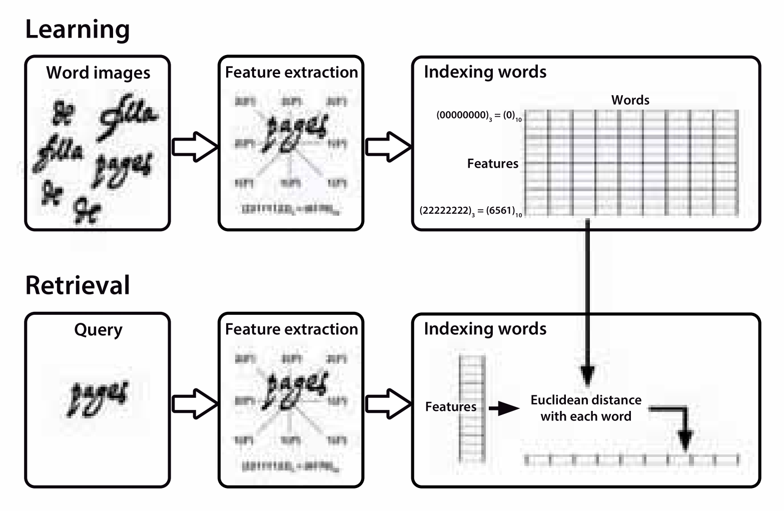


Figure 4.2: Outline of the approach.

points to select the characteristic points. The second one extracts the key points analyzing the skeleton of the word image. The third one is based in the contour of the word image. And the last one uses the foreground and the background regions of the image. In the feature extraction step, each key point is analyzed and its feature vector is computed. A fixed-size window is located in the centre of the key point. The window is analysed using the Loci descriptor presented in this Chapter. Finally, the feature vectors of a region are decoded using a codification algorithm.

In the following subsections we further describe the steps of our word spotting approach.

4.4.1 Word Normalization

The key points are distributed among horizontal regions, i.e. depending whether they appear in the top, bottom or central part of the word; and also into vertical regions. For this reason, the word length plays a crucial role in computing the similarity between words. If we divide all the word images using the same number of region/cells, long words would have cells with a big amount of pixels in contrast to short ones with cells containing very few pixels (see Fig. 4.4a and 4.4b). The main reason is that the use of different grid sizes for each word would result in feature vectors of different sizes, and therefore, needing more complex matching techniques. So, our purpose is to describe the word images with a feature vector of the same length, so that the distance between the feature vectors can be easily performed (e.g. Euclidean distance).

For this purpose, we follow the idea proposed by Fornés et. al [61]. Every word image is located in the centre of a template blank image, according to its centroid (see Fig. 4.4). The advantages of the normalization before the feature extraction are: the same number of cells is used for describing the characters of each word and the

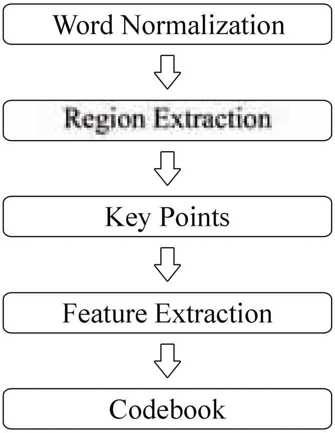


Figure 4.3: Flowchart of the proposed approach for word spotting based on Loci features.

feature vector of a short word (see Fig. 4.4d) is completely different from that long word (see Fig. 4.4c).

4.4.2 Region extraction

Methods based on histograms of features do not store the spatial relation of the key points. However, the use of the spatial relation can greatly increase the representation of a visual descriptor [201]. The spatial information of the key words is stored splitting the word images. The histogram of codewords is computed for each region. The word image is divided into a grid of *3 x n* equal-sized regions (see Fig. 4.4c and 4.4d). Horizontally, the word image is divided into three parts to delimit the ascenders. descenders and the main body. Vertically, the word has been divided in different parts.

4.4.3 Key Points

In order to create a histogram of visual words, we first need to extract local information from the image. This local information can be extracted using characteristic points of the image, edges or regions. We have evaluated 4 configurations (explained in Section 4.3). The first two ones are based in the location of characteristic points. The third one is based in edges, and the last one in regions.

4.4.4 Feature Extraction

The characteristic Loci features were devised by Glucksman and applied to the classification of mixed-font alphabetic, as described in [70]. A characteristic Loci feature is composed by the number of the intersections in the four directions (up, down, right and left) (Fig. 4.5). For each key point in a binary image, and each direction, we

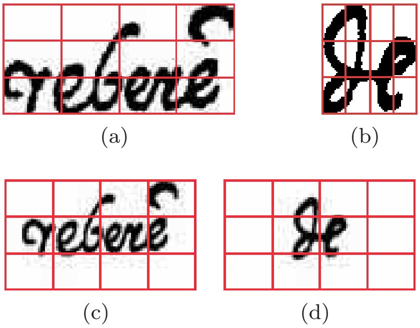


Figure 4.4: Two words with the same grid size: (a) and (b). Here, the number of columns used for describing each character is different. – Words located in a blank template image: (c) and (d). Here, the number of columns used for describing each character is similar.

count the number of intersections (an intersection means a black/white transition between two consecutive pixels). Hence, each key-point generates a codeword (called *Locu number*) of length 4.

This chapter presents a new feature descriptor based in the characteristic Loci features. We have introduced three variations on the basic descriptor:

- We have added the two diagonal directions, as we can see in Fig. 4.5. This gives more information to the descriptor and more sturdiness to the method.

- The number of the intersections is quantized. We have bounded the number of intersections in intervals. Each direction has a different quantization. This upper-bounding generates a more robust feature.

- The key-points detector will be seen in Section 4.4.3. We have evaluated several configurations: background and foreground pixels, local extrema points, contour and skeletons.

The skeleton of the image is computed by an iterative thinning until reaching lines of 1 pixel width.

The feature vector is computed by assigning a label to each key point pixel as shown in Fig. 4.5. The features are computed according to the number of intersections with the key points computed in right, upward, left and downward directions. In previous works, the characteristic Loci method has been applied to digits and isolated letter recognition. In this work, to reduce the dimension of the feature space, the maximum number of intersections has been limited to 3 values (0, 1 and 2). By delimiting the number of possible values we reduce the number of combinations. The length of the feature vector is proportional to the number of possible values. For example, with 3 possible values and 8 directions, we obtain 3^8 (6.561) combinations; with 4 possible values we have 3^4 (65.536). The computational and time cost increases in an exponential way making a costly system.

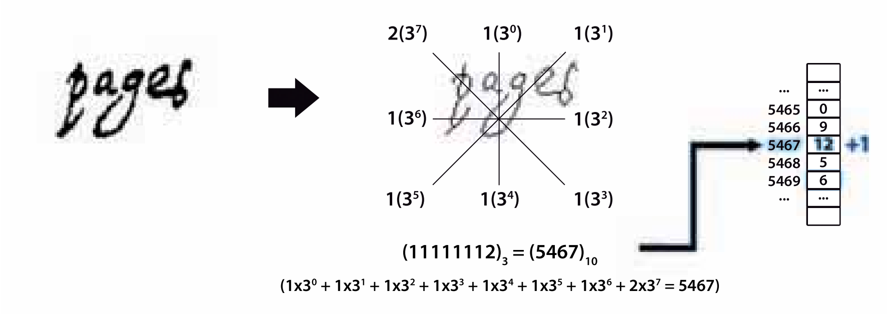


Figure 4.5: Characteristic Loci feature of a single point of the word page.

Table 4.1: Intervals for each direction in characteristic Loci feature.

	Values		
direction	**0**	**1**	**2**
Vertical	{0}	[1, 2]	[3, +∞]
Horizontal	{0}	[1, 4]	[5, +∞]
Diagonal	{0}	[1, 3]	[4, +∞]

Characteristic Loci feature was designed for digit and isolated letter recognition, and the number of intersections was bounded. The original approach uses the same interval in all directions. In this work we have also bounded the number of intersections. For each direction we have defined a different interval for each value. The horizontal direction has a larger interval than the vertical direction. In the original approach the digits or characters have a similar height and width, but in our approach the width of the words is usually bigger than the height. According to the dimensions of the words the range of the intervals is directly proportional. Diagonal directions are a combination of the two other directions. Table 4.1 shows the intervals for each direction.

According to the above encoding, for each background pixel, an eight-digit number in base 3 is obtained. For instance, the *Locu number* of point P in Fig. 4.5 is $(22111122)_3 = (6170)_{10}$. The *Locu numbers* are between 0 and 6.561 ($= 3^8$). This is done for all background pixels. In this case, the dimension of the feature space becomes 6.561. Each element of this vector is a *Locu number*, and represents the total number of background pixels with this *Locu number*.

4.4.5 Codebook

The retrieval process of this approach consists in organizing the feature codewords in a look up table M (Fig. 4.2). Columns of M represent the words (w) of the database. Rows correspond to all the possible combinations that can appear using characteristic Loci features (f). *M(f, w)* means that the feature f is presented in word w. For this work, we have 8 directions and each one has three different values. So, we have $3^8 (= 6.561)$ possible combinations. The feature vector is like a histogram of *Locu* *numbers.*

Classification process consists in searching the best matching of the query with all the words of M (Fig. 4.2). The chosen query is used to extract the vector of features. This vector is used to match the query with all the words of the database. In the retrieval step we have applied two distance formulations, namely Euclidean and Cosine, to match similar words.

4.5 Experimental results

The objective of this thesis is to show how the semantic information of the documents can improve the results of the word spotting approaches. Accordingly, we have performed three different experiments that analyse some of the works developed in the state of the art, and shows the importance of choosing not only the proper descriptor, if not to choose the proper key-points. First, we study the performance of the three classes of methods trough the approaches presented in Section 4.2. Second, we compared the key-points configurations listed in Section 4.3. And finally, we show the results obtained with the descriptor presented in this chapter.

To perform the experiments we have used the BH2M database (for more details see chapter 7). All the approaches presented are segmentation-based, so we assume that the words are segmented using the ground truth.

4.5.1 Experiments of the Analysis

This experiment evaluates the three classes of methods of word spotting: statistical, pseudo-structural and structural-based. The objective is to show the importance of choosing a proper descriptor depending on the problem.

We have selected 6544 word snippets with 1751 different transcriptions from the database. All the words having at least three characters and appearing at least ten times in the collections were selected as queries. There are 514 queries corresponding to 32 different words.

In order to evaluate the performance of the different word representation methods in a word spotting framework we have chosen the well-known mAP metric (Eq. 4.2). It derives from Average Precision (AP) (Eq. 4.1). AP provides a single number instead of a plot. It measures the quality of the system at all recall levels by averaging the precision for a single query:

$$AP = \frac{1}{RDN} \times \sum_{k=1}^{RDN} (Precision\, at\, rank\, of\, k^{th}\, relevant\, document) \tag{4.1}$$

where RDN is the number of relevant documents in the collection.

mAP is the mean of AP over all queries. Most frequently, arithmetic mean is used over the query set.

$$mAP = \frac{\sum_{q=1}^{Q} AP(q)}{Q} \tag{4.2}$$

Table 4.2: Retrieval results.

Descriptor	*mAP*
nrHOG	56.06
Loci-based	40.06
BoVW	30.00
DTW	19.20
Structural	9.70

where Q in the number of queries.

In Figure 4.6 an example of the qualitative results for a given query is presented. We can see that most of the methods present some false positives in the first ten responses. However, it is interesting to notice that this false positive words are in most of the cases similar to the query in terms of shape. In Figure 4.6, when asking for the word "*Farrer*", we obtain similar results such as "*Farer*", "*Ferrer*", "*Carrera*", "*Fuster*" or "*Serra*".

In Table 4.2 the evaluation measures are presented. We can see that the nrHOG method outperforms the other three. The BoVW and nrHOG methods extract the word representation directly from the gray-level word images, whereas the DTW-based and the structural methods need a preliminary step of binarization. In spite of degraded word images that might be difficult to binarize hindering the performance of the word representations that need a preprocessing step including binarization.

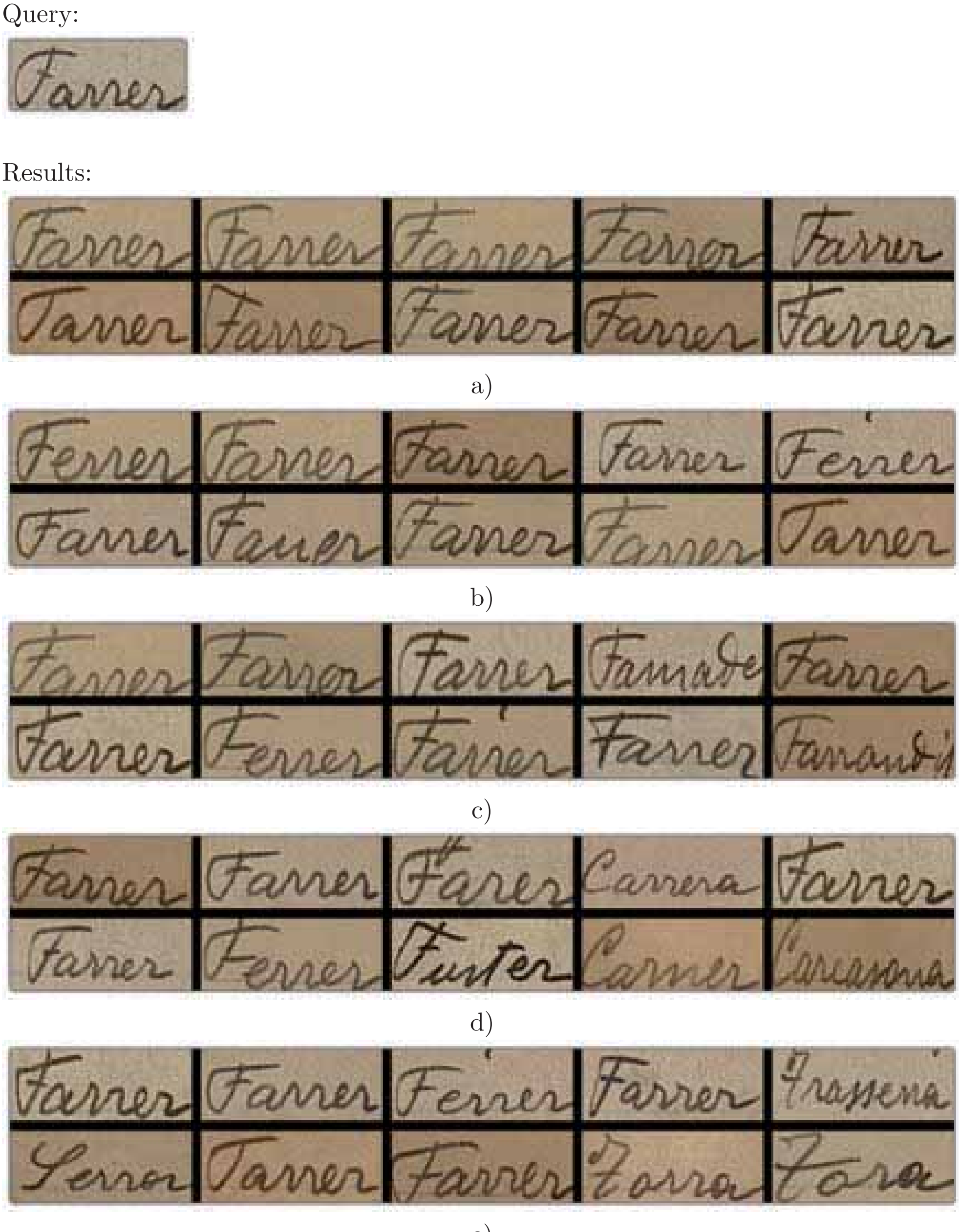
DTW is very time consuming. Computing distances among words with the DTW method is very time consuming since the algorithm has a $O(n^2)$ complexity. BoVW needs a training process, and therefore needs a labelled training dataset, that in most of the cases does not exist.

Finally, we can see that the structural method performs poorly. This word representation needs a preliminary step of transforming the raster image to a vectorial image in order to build a graph representing the words. When working with historical documents, slight degradations affect too much to the vectorization process yielding to an important performance loss of this word representation. However, it seems from the performance that the vectorization step does not have much effect on the performance but in time complexity for computing the paths. This can happen due to the appearance of noise in the image for which the vectorization method can create spurious critical points which increases the path computation time. Since the appearance of spurious points does not change the overall structure of a path, the performance will not drastically change.

4.5.2 Key point experiments

In order to show the importance of the selection of suitable key points, we first describe our performance evaluation protocol in terms of the dataset, descriptors, metrics and experiments.

To test and validate the best key points scheme, we have computed the results using two feature descriptors that present similar architecture. Both are pseudo-



a)

b)

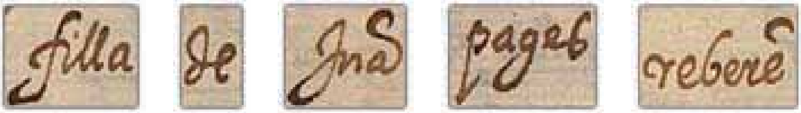
c)

d)

e)

Figure 4.6: Example of qualitative results. (a) nrHOG method, (b) Loci-based method, (c) BoVW method, (d) DTW-based method and (e) Structural method.

structural descriptors that needs to extract key-points. Once the key-points are com-



(a) ”filla” (b) ”de” (c) ”Juan” (d) ”pages” (e) ”rebere”

Figure 4.7: Examples of the classes used.

puted, the feature vectors are computed. The first one is the descriptor proposed in this chapter (see Section 4.4): (*Loci*). The second one is the *Shape Context* descriptor [16]. It describes the coarse distribution of the neighbouring shape with respect to a given point on the shape. For a given point in the shape contour, a coarse histogram of the relative coordinates of the remaining points is computed. The basic idea is to pick n points on the contours of a shape. For each point p_i on the shape, consider the $n1$ vectors obtained by connecting pi to all other points. The set of all these vectors is a rich description of the shape localized at that point but is far too detailed. The key idea is that the distribution over relative positions is a robust, compact, and highly discriminative descriptor. So, for the point p_i, the coarse histogram of the relative coordinates of the remaining $n1$ points, $h_i(k) = \#\{q \neq p_i : (q - p_i) \in bin(k)\}$ is defined to be the shape context of p_i. The bins are normally taken to be uniform in log-polar space.

We have selected 5 of the most representative words in these documents. For each word, we have selected five random samples to compute the mean of each class. In Fig. 4.7 we can observe an example of each word.

The metric used to analyse the performance of the descriptors using the different key points is the mean Average Precision metric (mAP)

In the case of the Shape Context descriptor, we need to create a codebook from the word image samples. For this purpose, we have used 5 documents (1707 words). We have used a equal-sized cells to compute the key points.

The experiments have been organized, for the horizontal division, in 3 equal-sized regions, and vertically, the image has been divided in 3, 4 and 5 equal-sized regions. Given these configurations, we have done a comparative evaluation and analysed all the key point – descriptor pairs.

	Loci			Shape Context		
	n=3	n=4	n=5	n=3	n=4	n=5
Skeleton Based	0.34	0.59	0.55	0.18	0.18	0.20
Local Extrema	0.32	0.51	0.52	0.47	0.44	0.53
Contour	0.31	0.55	0.55	0.53	0.53	0.69
Foreground	0.27	0.49	0.49	0.58	0.50	0.67
Background	0.83	0.86	0.84	0.60	0.60	0.67

Table 4.3: Region extraction: *Grid Cells.*

We can observe that the number of regions is an important parameter: when increasing the number of regions, the performance increases in most cases. E.g. we can see in Table 4.3 that using contours as key points the performance increases in both descriptors: from 31% to 55% and from 53% to 69%. Using the spatial information, the feature vectors are more discriminant because the are localized in the word image.

From the point of view of the selection of the key points, the more the key points, the better the performance. The skeleton based method obtains the worst results because the computed key points are centred only in bifurcations and extrema points. However, local extrema, contour and foreground methods obtain similar results, better than the skeleton based method because the number of key points is higher and more sparse than in the skeleton based approach. The background key points outperform all the other configurations independently of the descriptor that is chosen because the density of information is higher, and they are less sensible to variations.

4.5.3 Loci descriptor experiments

The objective of the experiment is to evaluate the performance using different size masks. Although the key points have been evaluated in Section 4.3, we perform the results using background and foreground pixels as key points to validate the results and using 10 key-words as queries.

We have selected the well known approach developed by Rath and Manmatha [151] as baseline to compare it with our proposed methodology. As we have commented above, we consider this approach as the first successful handwritten word spotting approach in the state of the art. They present an algorithm for matching handwritten words in noisy historical documents using Dynamic Time Warping (DTW). To evaluate the retrieval performance of the system with a query word against a handwritten document image, we use the standard precision (Eq. 4.3) and recall (Eq. 4.4) measures. We have done the following experiments:

- We have evaluated the performance using different characteristic pixels (background and foreground pixels of the word image) and using different mask sizes (size of regions of interest to compute the number of intersections for each key-point).
- We have compared our descriptor with the algorithm developed by Rath et al. [151].
- We have evaluated different distance measures (Euclidean and cosine distance) fixing the size mask.

$$Precision = \frac{|\{relevant\, docuemnts\} \bigcap \{retrieved\, documents\}|}{|\{retrieved\, documents\}|} \tag{4.3}$$

$$Recall = \frac{|\{relevant\, docuemnts\} \bigcap \{retrieved\, documents\}|}{|\{relevant\, documents\}|} \tag{4.4}$$

In table 4.4, we show the performance of the word retrieval system using different characteristic pixels as reference. The precision and recall table is computed according

Table 4.4: Accuracy using different characteristic pixels and mask sizes.

	Background pixels		Foreground pixels	
size mask	Precision	recall	precision	recall
15	68, 3%	66, 6%	39, 1%	77, 9%
20	76, 3%	68, 6%	43, 1%	79, 9%
40	78, 6%	64, 5%	44, 9%	79, 6%
80	89, 0%	67, 0%	46, 5%	79, 1%
100	89, 0%	67, 0%	46, 5%	79, 1%

to different mask sizes. We observe that when we increasing the mask size, the results are better. Using 80 or more pixels we obtain similar results. As we check in the evaluation of the key points (Section 4.3), the performance using background pixels we obtain better results. Using background pixels as reference, the number of pixels that gives information to the feature vector is higher than using foreground pixels.

Figure 4.8 shows the results of our experiments compared with the algorithm of Rath et al. [151] using the same database In our experiments we have used different distance measures to compute the distance between words descriptors. All the experiments are done using a mask size of 100 pixels. First, we have compared the performance using background and foreground pixels using Euclidean distance as measure. We can observe that we obtain better results using background pixels. We have used these measures because the Euclidean distance gives us the magnitude of difference between two word images, while cosine distance is a normalized measure which gives us a measure of how similar two word images are.

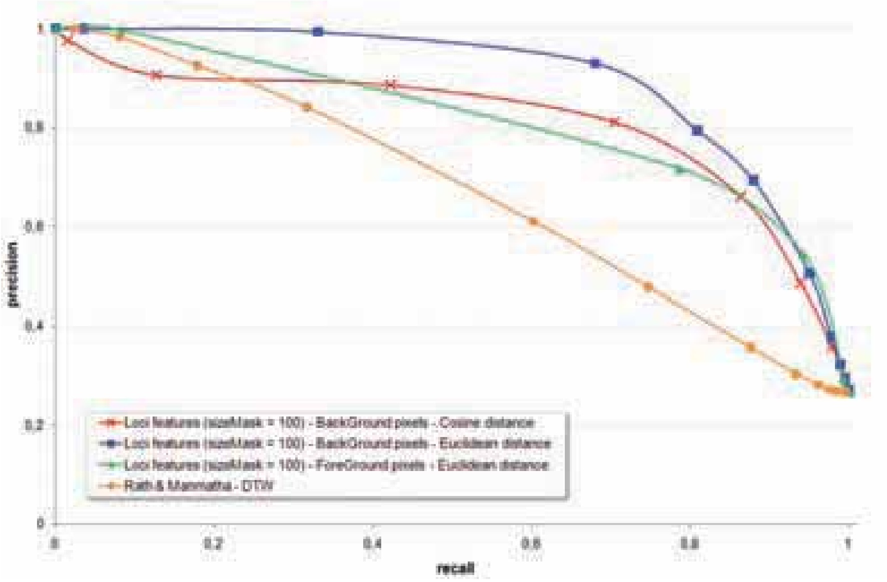


Figure 4.8: Experimental results. Precision-Recall curve.

In Figure 4.8 we also see the performance using different distance measures (Euclidean and cosine distance). The results are better using the Euclidean distance than using the cosine distance. Finally, we evaluate the performance in comparison to the algorithm of Rath and Manmatha. In their work they use Dynamic Time Warping to compute the distance between words. We can observe that our approach obtains better results. The better performance of our approach is due to the nature of the descriptor. While Rath and Manmatha use a pixel-based column descriptor, our descriptor captures more global information, including the structure of the word strokes.

Finally, Table 4.2 shows the comparative results of the state of the art methods and the results for the Loci descriptor. We observe that nrHOG and Loci obtaind the best results. Pseudo-structural methods outperform the rest of the methods. These kind of methods presents a compact signature that can be easily indexed, and do not need a training process. Thus scales much better than a DTW-based and BoVW-based methods.

4.6 Conclusions

In this chapter we have done a study of some methods of the state of the art categorized in three classes: statistical, pseudo-structural and structural methods. We have shown the importance of choosing a proper descriptor for each specific task. Statistical methods usually learning a model of the documents, but sometimes there is a lack of labelled documents and it is not possible to compute the model. In the other hand, structural models extract features from some characteristics of the word images, like skeletons, intersections, etc. The main problem is that these methods are not flexible enough in front of variations in the handwritten. Pseudo-structural methods are more robust in front of variations in the writer style. These methods analyse the information of the characteristics and around them.

We have also shown that the selection of the key points has the same importance than the selection of the descriptor. We have evaluated several key points in front of two descriptors that follow a similar architecture. The results show that the density of the key points in word images stores more information and increases the robustness in front of variation of the styles.

Finally, we have presented a new descriptor, which is based in a previous work oriented to characters. We have modified the original descriptor to adapt it to words. We have incorporated more information for each key point, and we have introduced spatial information dividing the image into cells.

Despite the descriptor presented in this chapter do not obtain the best results, the objective of this chapter was not to develop a new descriptor for the literature. The objective of this thesis is to show how the semantic information of the documents can improve the results of the word spotting approaches. Therefore, the study of the existing methods shows that pseudo-structural methods perform better in front of these kind of documents, and the descriptor presented in this chapter reinforce the robustness of these methods.

Chapter 5

Contextual Word Spotting

The use of contextual information in computer vision is a growing trend. Documents highly structured, like parish books, census, civil records, counts, etc., presents inherent contextual/semantic information that can be used to improve the results of classical word spotting approaches. In this chapter we introduce a new framework to spot words using the contextual information. This framework is composed by two layers. The first one discovers the contextual information automatically, and the second one uses this information to spot the words that appears more frequently in the documents. Our framework has been evaluated using the *BH2M* database, which presents a highly structured collection of marriage license documents. Compared with classical word spotting approaches, our framework achieves better results when we introduce contextual information in the search process.

5.1 Introduction

Word spotting has become a popular an efficient alternative to full transcription of documents. For example, word spotting is especially useful when the purpose is to retrieve information on people or events in historical or genealogical research from archives residing in municipalities. Classical word spotting approaches deal with the problem in a holistic way. It is done a visual search of a given query in a large image. Stated as a retrieval problem, and depending on the degradation level of images and the objective precision and recall, a number of false positive or true negative responses can be obtained. This level of error may be unacceptable, depending on the functional requirements of the application. There is an emerging trend of using contextual/semantic information to improve the performance of object recognition [29] in computer vision.

Over the last two decades much progress has been made in object detection using machine learning techniques. Usually recognition is built based solely on the statistics of local terms. Contextual information can provide more relevant information for the recognition of an object than intrinsic object information. The context of an object in the scene can be defined in terms of other recognised objects and their mutual dependencies. The use of correlative semantic labels between individual objects adds more discriminability in the process. Three types of context can be defined. First, the *object co-ocurrence* in a given image segment. Second, a *geometric context* involving a language model regarding to the relative 1D or 2D position of objects. Third, a *global or semantic context* defined by the topic of the document and consisting of semantic classes.

In document analysis, the use of the context for document recognition and contents extraction is very useful. Textual and graphical items in images are tokens of information carrying messages under a visual language, i.e. a document is communication sentence made by humans to be understood by humans. The three levels of contextual information are clearly found in document images. Hence, is a record, the probability of finding repetitive sorted words is high. Text recognition, either machine printed or handwritten, usually involves a language model that drives the recognition. Finally, the use of grammars describing the semantic context in a document understanding scenario is increasingly common, for example, the paleographic knowledge when reading historical documents.

The classical word spotting approaches can outperform the results using the contextual/semantic information. Informally speaking, contextual word spotting can be described as taking advantage of the lexical or syntactical structure of the sentence, i.e. the position of the query word in a sentence and the words before or after it (its context). If we integrate the joint probabilities of appearance of different words, we can overcome the individual misrecognition of one of them. Semantic word spotting [99] can be seen as a variant where the semantic categories of words reinforce the position where they can appear.

Knowledge discovery describes the process of automatically searching large volumes of data for patterns that can be considered knowledge about the data. The knowledge discovered can improve the extraction of information from the documents. Some types of documents, e.g. form-like records (birth, marriage, death), present a repetitive structure in the words within text lines along the pages. This repetitive structure defines the contextual/semantic information, which can be used to perform better result in the information extraction. In this chapter we will study implicit syntactic structures that can be roughly discovered from certain patterns consisting in repetitions of sequences of non consecutive words.

According to the presence of such patterns consisting of sequences of non-consecutive words, we will refer as *sequential context* the knowledge about the order of appearance of a certain set of words, among which is the query. Given this assumption, we benefit from the order, searching for the query word between the preceding and following words according to the known sequence. Therefore, *contextual word spotting* can be defined as the searching of words into the documents, but the search is influenced by the information that appears around it. For highly noisy or distorted query words, the search within a repetitive structure documents allows to reinforce the result by the neighbour words in the document, and hence to restrict the search position.

Our proposed approach is inspired in previous works where the alignment of text sequences is used to correct errors between different editions of the same book [206], or to align original [205] and translated editions [204]. In other cases the semantic

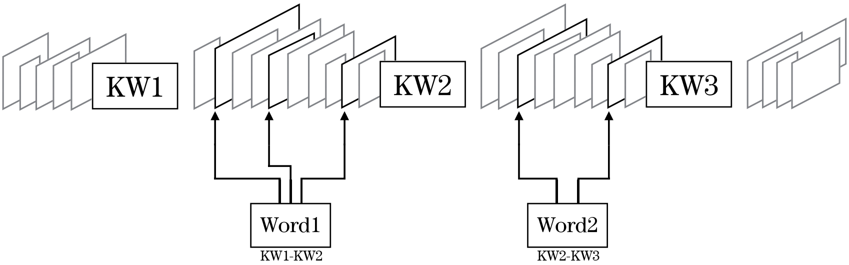


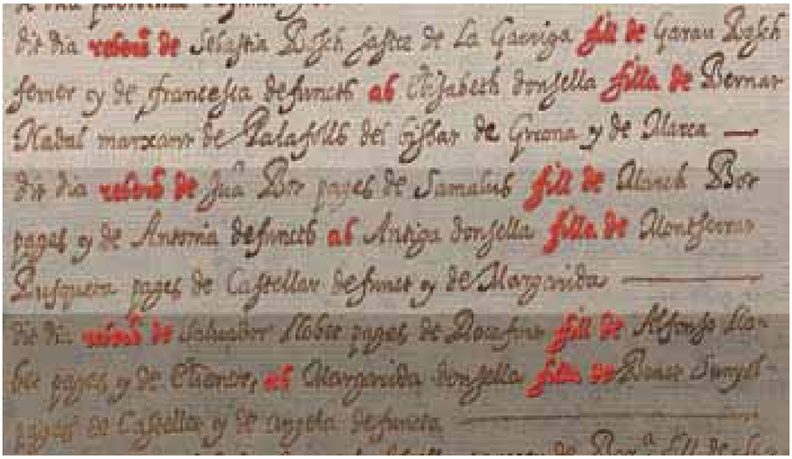
Figure 5.1: Framework model: key word discovering and contextual information extraction.

information is used for retrieval purposes [99] – words related semantically are given as similar, although the transcription and the shape of the word is different–. Other works adapt classical approaches, as Dynamic Time Warping [98] or Hidden Markov Models [58] to align the transcriptions of several documents.

The objective of the work presented in this chapter is twofold. First, documents are analysed to discover the contextual information, each word is categorize by semantic class, and subsequently, the most frequent words of each semantic class are extracted and spotted in the documents. The framework model presents two big layers. The first one is oriented to the extraction of key words, which establish the contextual information, and the alignment of these key words in all the documents. Key words are instances in the text that split the document by semantic information. This layer is composed by two steps. The first one analyses the handwritten documents to extract the repetitive structure. The structure is deduced analysing the frequency of the words and the location inside the text. Several key words are extracted after the analysis, which will be used to establish the contextual information. The second step consists in the alignment of the selected key words. The sequence of key words is searched with the premise that they appears in the same order inside the documents. This fact establishes the semantic classes, which groups several words that are semantically related.

The second layer is oriented to spot words by semantic information. Each block of words is analysed, and the words that appear more frequently are selected to be transcribed. This layer is inspired in the work of Rath et al. [152]. Word images are grouped in clusters of similar words. Each one is annotated and automatically all the words are transcribed. In our method we analyse the words by semantic information. The search is more accurate and the number of clusters is higher. The layer is divided in two steps. The first one applies a density-base clustering algorithm to each semantic class and the words that appear most frequently are extracted. The second step selects several examples of each cluster, and similar words are spotted in the respective semantic class.

Figure 5.1 shows the proposed model. The first level discovers the key words (*KW1*, *KW2*, etc.), which establish the contextual information, and the second level analyses the high frequency words between these key words. The spotting is done relating the contextual information for each semantic class.



Dit dia reberem de Sebastia [illegible] jaspa de La Garriga fill de Garau [illegible]
ferrer y de Francesca defuncts ab Elisabeth donzella filla de Bernat
Nadal marxant de Palafrugell del [illegible] de Girona y de Marca —
Dit dia reberem de Jua [illegible] pages de Samalus fill de Aleixh [illegible]
pages y de Anna defuncts ab Antiga donzella filla de Montserrat
[illegible] pages de Castellar defunct y de Margarida ——
Dit dia reberem de Salvador [illegible] pages de [illegible] fill de Alfonso [illegible]
[illegible] pages y de Elionor ab Margarida donzella filla de [illegible]
pages de Castellar y de Angela defuncta ——

Figure 5.2: Example of a repetitive structure. The key words are repeated along the three licence marriages.

We have observed experimentally that structured documents present semantic information. A particular case is *BH2M* database (for more details see chapter 7). This collection, which has been used for the experimental framework of this work, presents a repetitive structure along its pages. Each license store the information of a wedding, and the information is stored using the same structure along all the licenses. Figure 5.2 shows an example of repetitive structure. It shows three license marriages from the BH2M database. We can observe that the same key words in each license. These key words define the structure of the document.

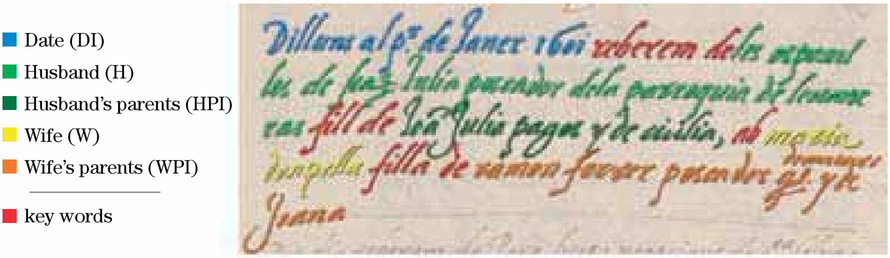
A semantic class is a sequence list of words which are related semantically. These classes are established by the key words. Figure 5.3 shows and example of a marriage license of the *BH2M* database. Key words are coloured in red. The semantic classes are established between them. The first semantic class (coloured in blue) is the day of the wedding. Following appears the husband and husband's parent semantic classes. After the keyword *ab* appears the wife and wife's parent semantic classes.

The rest of the paper is organized as follows. Firstly, in section 5.2 the approach is explained in deep. Secondly, in section 5.7 the experiments and results are presented. We finally draw the conclusions in section 5.8.

5.2 Outline of the Architecture

Figure 5.4 summarizes the outline of the framework model introduced in this chapter. As we have explained before, the framework model has two layers. The first one is oriented to the knowledge discovery, and the second one to analyse the most frequent words by semantic class.

The first layer establishes the contextual/semantic information through two steps. The first step consists in discovering the key words that shape the contextual information to the documents. The imposed constrains of the selected key words are:


🟦 Date (DI)
🟩 Husband (H)
🟩 Husband's parents (HPI)
🟨 Wife (W)
🟧 Wife's parents (WPI)
🟥 key words

Dilluns al pr(imer) de Janer 1601 reberem de les esposal les de fran(cesc) Julia pescador de la parroquia de levane ras fill de Joa(n) Julia pages y de cicilia, a(m)b maria donzella filla de ramon ferrer pescador q(uondam) de mataro y de Joana	Monday the first of January 1601 received the marriage license fee for francesc Julia fisherman from the levaneres parish son of Joan Julia peasant and cicilia with Maria maiden daughter of ramon ferrer fisherman deceased de mataro and Joana
Literal transcription	English literal transcription

Figure 5.3: Semantic classes are established by the key words.

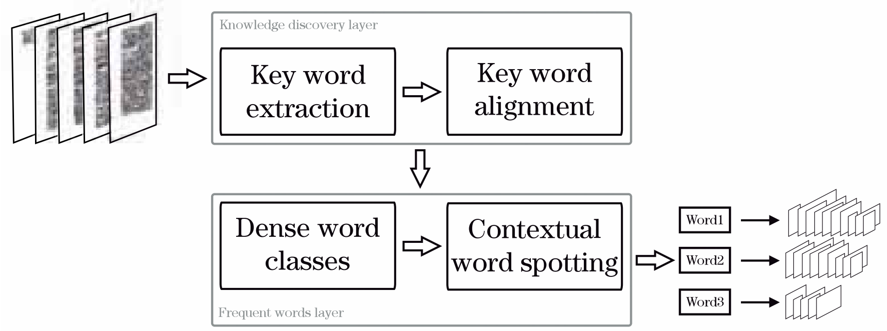


Figure 5.4: Flowchart of the proposed approach.

they have to repetitively appear along the text, and they have to appear always in the same order. Once the key words are selected, the second step of the layer is the key word alignment. Key words are searched in the documents in the same sequence order. The auto-alignment intra document detects the word sequences with regular repetition.

The second layer take advantage of the semantic information established in the first layer. The first step analyses each semantic class to discover the words that appear more frequently. The discovering is performed using a density-based clustering algorithm which groups similar word images and select the densest clusters. Next, the second step uses the words of this cluster to spot similar words in all the documents. Let us further describe the different steps.

5.3 Key Word Extraction

Contextual information is a great value in highly structured documents and can be used to improve the results of classical word spotting approaches. The contextual information is discovered through the key words, which establishes where the semantic classes are located in the text. We introduce a new unsupervised approach to discover the key words in handwritten documents.

Automatic key word extraction is the task to identify a small set of words, key phrases, key words or key segments from a document that describe its meaning [79]. Key words are useful for readers, scholars, business, social and others text' purposes. They are used to give an insight of the presented text. They give a clue about the text concept so that readers decide their interest. Their importance for research engines is to precise inquiry results and shortens the response time. Key words can be viewed as classifiers.

A collocation is just a set of words occurring together more often than by chance in a text document. Collocation is a syntactic and semantic unit whose exact and unambiguous meaning. The collocation can be of two types i.e. rigid or flexible. Rigid collocation are those n-grams that always occur side by side and appear in same order whereas flexible collocation are n-grams that can have intervening words placed between them or can occur in different order [38].

The approaches for automatic key word indexing can be categorized in: key word extraction and key word assignment. In key word extraction, words occurred in the document are analysed to identify apparently significant ones, on the basis of properties such as frequency and length. In key word assignment, key words are chosen from a controlled vocabulary of terms, and documents are classified according to their content into classes that correspond to elements of the vocabulary [132]. Existing methods about automatic keyword extraction can be divided into four categories: statistic, linguistic, machine learning and other methods.

Statistical methods tend to focus on non-linguistic features of the text such as term frequency, inverse document frequency, and position of a key word [39]. The benefits of purely statistics are their ease of use, limited computation requirements, and the fact that they do generally produce good results. However some methods pay attention to linguistic features such as part-of-speech, syntactic structure and semantic qualitative tend to add value, functioning sometimes as filters for bad key words [75, 79]. Other way to extract key words is using supervised learning from the examples. Machine learning approach employs the extracted key words from training documents to learn a model and applies the model to find key words from new documents [86, 213, 214]. Some of the approaches about key word extraction mainly combine the methods mentioned above or use some heuristic knowledge in the task of key word extraction, such as the position, length, layout feature of the words, etc. [80]. Bag-of-Words (BoW) is a widely used model in a variety tasks in Natural Language Processing (NLP). In this model, the text is represented as the bag of its words, disregarding grammar and even word order but keeping multiplicity. he bag-of-words model is commonly used in methods of document classification, where the (frequency of) occurrence of each word is used as a feature for training a classifier [176].

Some methods have been developed to key word extraction for handwritten documents, as the work Saabni et al. [171]. Their algorithm is based on geometric features (contours of the complete word-parts and strokes, such as dots and detached short comments). The search for a key word is performed by the search for its word-parts, including the additional strokes, in the right order. The use of these allows to use any feature-based matching technique, such as DTW or HMM. They show the good performance of the HMM based system when it is provided an adequate training sample (which is not always possible in historical documents). The slightly modified DTW algorithm provides better results even with a small set of training samples.

We introduce a new statistical approach for unsupervised key word extraction based in the frequency and the position of the key-words. We assume that the collocation of the key words is rigid. The key words always appear in the same order and relative position. The approach developed studies the frequency of the words and the relative position between them to determine the key words that achieve the two premises: to appear repeatedly along the text and always in the same sequence order. The approach consists in two steps: key word candidates and LCS-based key word extraction.

Key word candidates

Discovering key words in all the collection of documents is not a trivial task. Several techniques are used in the literature, as we have seen above, to discover key words. All of them are oriented to categorize the documents or to find specific key words inside the documents. In our work the objective is to discover the key words in highly structure documents.

We assume that the layout of the documents is known, and the records (paragraphs that contains the wedding information) are segmented. The segmentation of the physical layout is out of the scope of this chapter. We use the approach of Cruz et al. [35] which is oriented to extract text blocks in historical documents. They propose to use relative local features in a conditional random field framework to perform document segmentation tasks. The segmentation of he lines has been described in Chapter 3.

We haven selected 100 random records of the documents to discover the key words.

The objective is to compute the frequency of the words of each record, in front of the rest of the selected records. We describe in detail the entire process next:

1. Firstly, the features for each word are computed. The descriptor used is a deformable HOG-based shape descriptor [3]. It is the descriptor that performs the best results in the comparative of the chapter 4.

2. Once the features are computed, the Euclidean distance between the words of each selected record, and the rest of records is computed.

$$d(p,q) = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2} \tag{5.1}$$

   where p and q are the feature vectors of dimension n.

3. For each record, we compute the histogram of the words (Eq. 5.2). After the normalization of the histograms (Eq. 5.3), the words with a high frequency are discarded as candidates of key words. High frequencies in text usually represent stop-words. These words are not representative as key words because they appear along all the text randomly. Words with small frequencies are also rejected, because they appear sporadically and we are interested in words that appear in all the records (or in most of the records).

$$h(i) = \sum_{i=1}^{n}\sum_{j=1}^{m} d(i,j) \tag{5.2}$$

$$h(i) = \frac{h(i) - h_{Min}}{h_{Max} - h_{Min}} \tag{5.3}$$

   where $h(i)$ is each input word, h_{Min} is the minima among all the data and h_{Max} is the maxima among all the data points.

Key word candidates are selected after removing high and low frequencies in the histograms of similar words. We achieve the most frequent premise established above, but the candidates do not follow the same sequence order along the text. Next we introduce the step that analyses and establish this constraint.

LCS-based key word extraction

After selecting the key word candidates from the selected records, the next step is to select the words that accomplish the premises established previously: the list of key words has to appear in most of the records and they have to conserve the same sequence order in all of them. The second premise is achieved using the Longest Common Sequence (LCS) algorithm. LCS is basically a global alignment method which gives the longest sequence preserving the long range order between two sequences.

For a string example, consider the sequences "**t**hi**si**sa**test**" and "**t**e**s**t**i**ng123**test**ing". An LCS would be "**tsitest**".

There are a number of algorithms to compute LCS in the literature [42]. The standard dynamic programming algorithms has $O(mn)$ time and space requirements, where m and n are the length of the inputs sequences. For long input sequences, this algorithm has very large memory requirements. Therefore we adopt an $O(mn)$ time and linear space LCS algorithm [77] to calculate the LCS length without computing the actual LCS sequence itself. There is also a $O(nloglogn)$ time LCS algorithm for sequences where no elements appear more than once within either input sting [81]. This algorithm is not suitable for our purposes because the sequence of words may include repeated words.

The method developed for the purpose of selecting the key words from the candidates is based in the LCS algorithm and frequencies. It is composed by several steps shown next:

1. **LCS alignment**: The first step consists in the alignment of the candidates of each record with the rest of records. Given a set of sequence order of words P, the algorithm LCS aligns the sequence of words P_i and P_j.

   $$w(i,j) = LCS(p_i(h(i)), p_j(h(j))) \tag{5.4}$$

   where p represents the list of selected words (h) to be aligned, i and j are not equal and the LCS algorithm is defined as follow:

   Let two sequences be defined as follows: $P_i = (p_i^1, p_i^2 ... p_i^m)$ and $P_j = (p_j^1, p_j^2 ... p_j^n)$. The prefixes of P_i are $P_i^1, 2, \ldots m$; the prefixes of P_j are $P_j^1, 2, ...n$. Let $LCS(P_i^k, P_j^l)$ represent the set of longest common subsequence of prefixes P_i^k and P_j^l. This set of sequences is given by the following.

   $$LCS(P_i^k, P_j^l) = \begin{cases} 0 & \text{if } k = 0 \text{ or } k = 0 \\ LCS(P_i^{k-1}, P_j^{l-1}) & \text{if } p_i^k = p_j^k \\ longest(LCS(P_i^k, P_j^{l-1}), LCS(P_i^{k-1}, P_j^l)) & \text{if } p_i^k \neq p_j^k \end{cases} \tag{5.5}$$

   To find the longest subsequences common to P_i and p_j, compare the elements p_i^k and p_j^l. If they are equal, then the sequence $LCS(P_y^{k-1}, Y_j^{l-1})$ is extended by that element, p_i^k. If they are not equal, then the longer of the two sequences, $LCS(P_i^k, P_j^{l-1})$, and $LCS(P_i^{k-1}, P_k^l)$, is retained. (If they are both the same length, but not identical, then both are retained.) Notice that the subscripts are reduced by 1 in these formulas. That can result in a subscript of 0. Since the sequence elements are defined to start at 1, it was necessary to add the requirement that the LCS is empty when a subscript is zero.

2. **Alignment in sequence order**: The next step searches the paths between each sequence of aligned words w_i and the rest of the sequences (see Figure 5.5). Each path is labelled as follows: for each pair of aligned words $w(i,j)$,

source words without label (L_i), are labelled with a new one and the new labels are added to the target words (L_j). Target words that are aligned with labelled words, incorporate the label most frequent in the list of labels of the source word to their list of nodes. The algorithm is summarized in **Algorithm 2**.

Algorithm 2 Alignment in sequence order.

```
$L$ is empty initiated;
$lastL = 0$;
for all $w_i^j \in w$ do
    if $i \neq j$ then
        if $L(i)$ is empty then
            $lastL = lastL + 1$;
            $Add(L(i), lastL)$;
            $Add(L(j), lastL)$;
        else
            $Add(L(j), mostFrequent(L(i)))$;
        end if
    end if
end for
```

We can see an illustration portrayal of the Algorithm 2 in the Figure 5.5. The right part represents the paths computed using the LCS algorithm between the sequences of words. The left part represents how the information is stored. Each candidate stores the labels of the paths which is involved. Coloured labels represent the current paths of the diagram. Grey labels represent future states.

3. **Path selection**: Finally, paths (L) that covers more than 60% of the records are selected as key word paths (D).

Once the paths are discovered, we select K samples randomly of each path. K is determined by the length of the shortest selected path.

$$K = min(length(D(i))) \tag{5.6}$$

5.4 Key Word Alignment

Once the key words are selected from a sample of the collection (in our case using 100 records), the next step consists in localize the keywords in all the collection. An alternative to search the words is using the classical word spotting approaches. These approaches are methods widely accepted to extract the information of Historical Handwritten Documents, but although we can get pretty good results, it seems that they are reaching their ceiling. A quality leap in word spotting approaches can be achieved by introducing contextual information (in this particular case the previous knowledge of the sequence order of the words). The objective is to use this kind of information to improve the results obtained in the classical word spotting approaches.

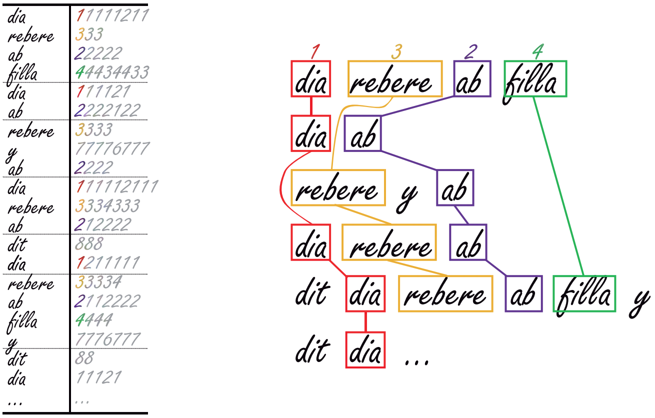


Figure 5.5: Representation of the LCS-based algorithm to align sequence of words. Coloured labels represents the current state. Gray labels represents a possible future state.

Concretely, our approach is inspired in time series alignment algorithms [17, 105, 137], to search instances of handwritten words in the same order of a given input list of words.

The proposed algorithm in this step starts with the premise that all the words of a list D are included in a list W (longer than D) repetitively N times (in the same order of appearance). The objective of the algorithm is to localize all the instances of D in W (where W is the list of words od the documents). There are several methods in the literature to do alignment of characters, words, instances, etc. , but of all them present the same problem: they are oriented to ascii code. The algorithms compare the characters, words, etc. [17, 206] to discover if they are equals or not. In our problem the similitude between word images is given by a distance. We propose an algorithm based in the main methodologies of sub-string matching [132]. This algorithm localizes a word d_i of the list D in the list W (w_j) taking in account the sequence order of the words and the similitude distances between the words localized behind (w_{j-1}) and before (w_{j+1}) of w_j.

Our approach follows the following steps: given a sorted list of words and the licence documents, we first compute the features for each word image. Next, the words in the word list are aligned with the words in the marriage licences. A visual scheme of the model is shown in Figure 5.6. Next, we describe this process in detail.

Feature extraction

We assume that the words of the documents are previously segmented, so the words are extracted and stored in the same sequential order that appears in the documents. The descriptor. The feature vector of each word image is computed and stored in the same sequence order that appears in the documents. We use the same descriptor that previous steps: the deformable HOG-based shape descriptor [3].

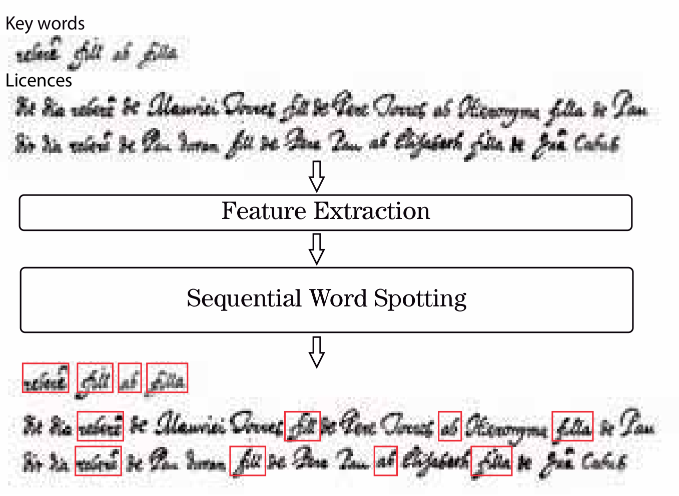


Figure 5.6: Illustration of the key-word alignment process.

Sequential word spotting

The objective of Sequential Word Spotting is to find similar instances of words from the list of input queries, but taking into account the ordered sequence previously established.

Formally, given a sequence of handwritten query words $D = a_1, a_2, a_3, \ldots, a_n$ and a list of target words $W = w_1, w_2, w_3, \ldots, w_m$, where $n < m$ and $D \subset W$, the alignment is formulated by a mapping function f. Thus, let f be defined as a mapping function between the words in sequence D to some words in sequence W. Hence the mapping $j = f(i)$, where $i \in [1, \ldots, n]$ and $j \in [1, \ldots, m]$, represents that $a_i = w_j$. An order constraint $f(i) < f(i+1)$ is imposed in the mapping, in other words the the words in D appear in W following the same order. The alignment algorithm returns a valid mapping $f(j)$ where $j = 1, \ldots, m$ with the minimum cost.

The mapping function f selects the instances of W taking into account the similarity of the words and the sequence order of D. For each input instance a_i, the r most similar instances $\{w_{f^1(i)}, w_{f^2(i)}, \ldots, w_j, \ldots, w_{f^r(i)}\}$ from W are selected. Each candidate w_j is a possible solution of f and therefore it is explored to find the optimal solution. Then, for the next input instance a_{i+1} the instance candidates of W have to appear after w_j. In other words, the instances that appear before the position $j+1$ are not taken in account. The candidate chosen for d_i is the w_j with the minimum cost. The final cost is the cumulative sum of the distances between the d_i words with the corresponding $w_{f(i)}$ candidates. A visual illustration of the mapping function f is shown in Figure 5.7.

The mapping function f align the words of the list D with the first closest instances in the list W. To find the rest of the key words of the list D in the list W, the mapping function is applied until the list W is empty or have less items than D. The first item of the list W is the next item of the last item selected in the last iteration

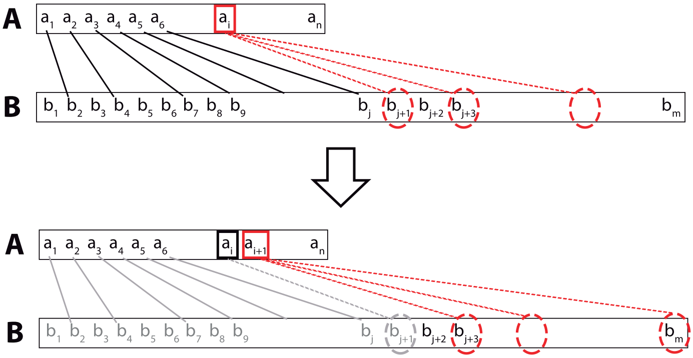

A
a_1 a_2 a_3 a_4 a_5 a_6 a_i a_n
B
b_1 b_2 b_3 b_4 b_5 b_6 b_7 b_8 b_9 b_j b_{j+1} b_{j+2} b_{j+3} b_m
A
a_1 a_2 a_3 a_4 a_5 a_6 a_i a_{i+1} a_n
B
b_1 b_2 b_3 b_4 b_5 b_6 b_7 b_8 b_9 b_j b_{j+1} b_{j+2} b_{j+3} b_m

Figure 5.7: Illustrative example of the mapping function f. The mapping function generates some candidates for the input query a_i. From the candidates, the query word is mapped to the word b_{j+1}. Then, the candidates for the query a_{i+1} are generated starting from the position of the previous mapped word b_{j+1}, taking into account the similarity to the input query a_{i+1}.

$W = (w_{f(i)+1}, w_{f(i)+2}, \ldots, w_m)$.

Once the key words are aligned in the documents, the contextual/semantic information is established and the semantic classes are assigned.

5.5 Dense Word Classes

In one of the pioneering works of word spotting, Rath et. al [152] cluster first similar snippet of word images. So they are simultaneously retrieval given a user's query. They use k-means clustering algorithm and various agglomerative clustering approaches to group similar instances. But, the main problem of these clustering algorithms is to select the proper number of clusters. Inspired by this idea, we have incorporated the search for the most frequent in each semantic class. So contextual word spotting is shown as an efficient tool for semi-automatic transcription.

There are different methods in the literature to choose the number of clusters [203]. They can be classified in two big groups depending on how the number of clusters are chosen. The first one is a manual method. The number of clusters is chosen based on the experimentation. Although for some applications, users can determine the number of clusters using their experience. The second group is automatic, or pseudo-automatic. Algorithms of this group use an index, or several indices, to obtain a measure that allows to choose the best number of clusters. There are several validity indices. They can be classified in two groups: external and internal validity indices.

External validity indices are used when true class labels are known. Some examples of external validity indices are: *Rand index*, measures the similarity between two data clustering; *Adjusted Rand index* is the corrected-for-chance version of the Rand index; *Mirkin index* considers only object pair in different clusters for both partitions and finds the dissimilarity.

Internal validity indices are used when true class labels are unknown. Some ex-

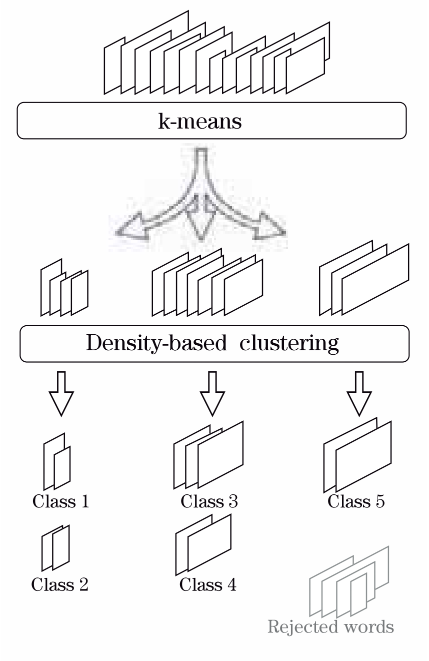


Figure 5.8: Illustration of the density-based clustering.

amples of internal validity indices are: *Silhouette index* [167] computes the average distance of a point from the other points of the cluster to which the point is assigned; *Davies-Boulding index* [40] is a function of the ratio of the sum of within-cluster scatter to between-cluster separation; *Calinski-Harabasz index* computes the sum of the squares of the distances between the clusters centroids and the mean of all objects in a cluster from their respective cluster centre.

The tool for semi-automatic transcription introduced in this chapter localizes similar instances, and uses the same text to transcribe all of them. There are two ways to select the words that we want to transcribe. The first one is manually, the user selects/crops the word images. In this case, this step of the system consists in selecting several instances of each word for each semantic class. The second one is selecting automatically the words that appear a major number of times. The objective is to transcribe most of words of the documents efficiently. So, words that appears a high number of times in each semantic class are the best candidates to be transcribed.

We have developed an approach that combines a manual and an automatic method to select the number of clusters. The approach has two levels (Figure 5.8):

- The first level uses geometric shape properties as features: aspect ratio, height and width. Images are clustered using the k-means algorithm. The parameter k is fixed to 3. We group word images in small, medium and large words. In this case, the objective is to find the completeness and not the homogeneity of the clusters. The completeness is achieved when all the members of a given class are assigned to the same cluster. The homogeneity is when each clusters contains only members of a single class.

- The second level clusters each group of the first level using a shape descriptor. The descriptor used in this work is a deformable HOG-based shape descriptor [3]. The descriptor of the comparative of Chapter 4 that get the best results. As we have commented above, we do not know in advance the number of clusters, so we have used a density-based algorithm for infer the optimum number of clusters [54]. It discovers clusters of arbitrary shape efficiently even for large spatial databases. The key idea is that for each point of a cluster the neighbourhood of a given radius has to contain at least a minimum number of points, i.e. the density in the neighbourhood has to exceed some threshold. The shape of a neighbourhood is determined by the choice of a distance function for two points p and q, denoted *dist(p,q)*.

The objective is discover the major number of clusters to transcribe a high number of words. If we cluster all the words of the documents at the same time, the results gotten can be ambiguous and not as expected. Documents highly structured with semantic information, usually present high repetitions of words in their classes, but in a massive clustering, these clusters by class can be scattered. Instead, clustering the words by semantic classes (fixed in the last step of the approach), the number of clusters are higher and oriented to their semantic information. For example, the name "David" is a man name. It can appear in the part of the license that talks about the husband, but rarely appears in the other parts. So, clustering the semantic classes that contains the husband's information, it will appear a cluster containing examples of this word. But in a massive clustering, probably it will disappear due to other most populated clusters. The output of this step are semantic clusters, which contain similar instances of the same word and the related semantic information.

5.6 Contextual Word Spotting

In the work of Rath et. al [153] the output of the clustering process is used to transcribe the words. But the problem is when the homogeneity of the clusters is low. Each cluster not only contains members of a single class. This fact produces some wrong transcriptions of the words that are not clustered properly. To solve this problem, we introduce the contextual/semantic information in the spot process. The search of the words by semantic class achieves better results than spot the same word in all the collection. For example, the name *Montserrada* (female name in old Catalan) and *Montserrat* (male name in old Catalan) have the same origin and they are very similar. Searching these words in their respective semantic classes, *Montserrada* in the wife information and Montserrat in the husband information, we get better results than searching in all the documents. The semantic class of wife information contains mainly female names, and rarely appears male names, so the misclassification between *Montserrada* and *Montserrat* will be less than when we search in the entire document.

We introduce the contextual word spotting that searches by semantic class. But one of the premises is to know the semantic class to search previously. Our method is based in some methods of the literature that uses the query by expansion technique to get more accurate results [31, 12, 11, 170]. The objective is to use several instances of the same input query class to improve the representation of this input query.

The contextual word spotting approach searches the words previously selected in the density-based clustering step (Section 5.5). Each word is related with a semantic class, and the search is focused in the related one. The search is refined using several instances $Q = (q_1, q_2, q_3, \ldots, q_n)$ of the same words (selected previously also in the last step). For each instance is computed the feature vector $V = (v_1, v_2, v_3, \ldots, v_n)$ where $v_i = f(v_i)$ and f is the function that extract the features from v_i . Next, the mean of V is computed to extract a representation r of the input query. Finally, it is computed the distance between r and all the words of the semantic class related. Closest words are extracted sorting the distances.

5.7 Experimental Results

The framework presented in this chapter is composed by different steps. Each one of these steps has been evaluated independently, although the performance of each step depends of the result of the previous one. The experiments have been done using the BH2M database (see Chapter 7 for more details). These documents contain the marriage licenses of Barcelona and surroundings. The documents of this collection are structured by paragraphs. The structure of each of these registers is comprised by a certain information items, such as the date of the wedding, the husband's information, the husband's parents, the wife's information and the wife's parents. These information can be split using some characteristic words that appears in the text of the paragraphs.

5.7.1 Key word Extraction

The first step of the framework is the automatic localization and extraction of the key words of the documents. The documents used to perform our experiments are composed by license marriages, where each one contains similar information and structure. The general sub-parts of a licence (L) are, in appearance order, the date-related information (D), the husband-related information (H) and the wife-related information(W). These three sub-parts are joined by keywords as follows: date (*DI*) and husband parts are connected by *rebere de* (*we received from* in old Catalan), husband and wife (*HI* and *WI*) information are connected by *ab* (abbreviation of *with*). Both husband and wife information parts are divided in two sub-parts, being his/her information and the corresponding parts (*HPI* and *WPI*) information. These two last parts are connected by the key words *fill de* and *filla de* (*son of* and *daughter of*) respectively. Figure 5.3 shows an example of the sub-parts of marriage licence. Formally, the syntactic rules are:

$$\begin{array}{ll} D \wedge H \wedge W & \Rightarrow L \\ DI \wedge rebere \wedge de & \Rightarrow D \\ HI {\vee} (fill {\wedge} de {\wedge} HPI) & \Rightarrow H \\ ab {\wedge} WI {\vee} (filla {\wedge} de {\wedge} WPI) & \Rightarrow W \end{array}$$

We can observe that the fixed words in the licence are the key words, which help us to split the information semantically. The objective of this step is to extract these words automatically. Some of these key words are composed by two words, one of which is the word *de*. It is a stop-word that can appears in any part of the text and cannot be considered as key word. So the key words of our documents are: *rebere*, *fill*, *ab* and *filla*.

Table 5.1: Number of items of each class and percentage of record that contain the class of the top-15 classes in the database. Left column covers all the database. Right column are the statistics for the 100 selected records.

All			Selected		
word	#items	frequency	word	#items	frequency
de	11,346	1,740 (100%)	de	656	100 (100%)
y	2,420	1,469 (84.43%)	y	131	83 (83%)
pages	1,799	1,013 (58.22%)	**ab**	**100**	**100 (100%)**
ab	**1,730**	**1,729 (99.38%)**	**rebere**	**98**	**98 (98%)**
rebere	**1,694**	**1,693 (97.30%)**	pages	96	57 (57%)
filla	**1,313**	**1,310 (75.29%)**	**filla**	**74**	**74 (74%)**
barcelona	1,267	835 (47.99%)	donsella	69	69 (69%)
donsella	1,264	1,264 (72.64%)	barcelona	65	47 (47%)
dia	1,074	1,071 (61.55%)	dia	60	60 (60%)
dit	1,030	1,010 (58.05%)	dit	55	54 (54%)
fill	**1,015**	**1,011 (58.10%)**	**fill**	**53**	**53 (53%)**
en	887	774 (44.48%)	en	52	45 (45%)
a	690	681 (39.14%)	a	41	40 (40%)
defunct	616	569 (32.70%)	habitant	38	35 (35%)
habitant	602	579 (33.28%)	defunct	37	34 (34%)

In Table 5.1 we can observe the top-15 classes of the database. It shows the data for all the database (left part of the table) and the data for the 100 selected records (right part). For each class the table shows the number of times the class appears in the dataset and the frequency of apparition of the class in the records. The classes *de* and *y* are the most populated classes and they are considered as key word candidates, but the approach presented removes high populated classes considering them as stop-words. The order of both lists are quite similar (it is a good indicator that the selected records are a good representation of the database), and the key words extracted from the syntactic rules are positioned equally.

The classes *ab*, *rebere* and *filla* appear in the 99.38%, the 97.30% and the 75.29% of the records of all the database respectively, and in the 100%, the 98% and the 74% of the selected records. The class *fill* appears in the position eleven of the both lists, and the number of times that this class appears are 58.10% and 53%. It is a small frequency because there are a high number of records where the husband's parent information does not exist.

We have performed the experiments using the 100 records referred above. We have used a deformable HOG-based shape descriptor (see Section 5.3) to compute the similarity between the word images. In Figure 5.9 we can see an example of each class selected as keyword by the approach developed. The sequence order of the words in the figure is the same that the established by the approach. We can observe that the class *dia* has been selected as key word. In the Table 5.1 we observe that the frequency is 61.55% and 60% in all de database and the selected records respectively. The frequency is above, or equal, to the threshold established as good frequency: 60%. Although this class has not been considered as key word in the syntactic rules, the results achieved make this class as good candidate of key word because it meets the premises and because this word is used to indicate the day of the wedding. In the other hand, the class *fill* is rejected because the frequency is 58.10% and 53%, under the fixed threshold. This threshold has been computed experimentally to get the optimum results. A lower threshold means a major number of key words not desired.

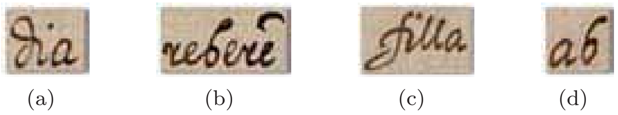


Figure 5.9: Result of the classes selected as key words: (a) *dia*,(b) *rebere*,(c) *ab* and (d) *filla*.

Concerning to the other classes, despite the high frequency, are not consider as key word because they do not meet one of the premises established previously (see Section 5.3): the position of the instances change along the text and does not appear in the same position with respect the other candidates.

5.7.2 Key word Alignment

The second step of the framework is the alignment of the selected key words in all the dataset. Following the last step, we have used the same descriptor to extract de features and compute the similarity between the word images. The metric used in order to evaluate the performance of the alignment of the key words is the mean Average Precision (mAP), which provides a single-figure measure of quality across recall curves (for more details the reader is referred to Chapter 4 Section 4.5).

We have evaluated the performance of the approach comparing our method with a classical Word Spotting approach. The classical architecture extracts the features of all the words of the dataset and the input queries, and computes the similarity between them using a similarity measures. Next, the words are sorted using this similarity. Table 5.2 shows the comparison of the methods for each key word and the mean of all of them. It can be seen that we have outperformed the original word spotting method. In all cases the accuracy is increased. We can observe that the performance of the word *fill* is slightly lower to the rest of the words because the word *fill*, which is part of the word *filla*, introduces confusion in the results.

Table 5.2: Retrieval results of the alignment process.

	word spotting	Alignment
mean	97.01	98.02
dia	97.10	98.03
rebere	97.05	98.01
ab	98.00	99.00
filla	94.10	95.04

5.7.3 Dense Word Class

Once the alignment is done over all the documents, the semantic classes are fixed and the words are labelled semantically. We have performed two experiments to compute the words that appear more frequently. Both experiments have been performed using the density-based clustering algorithm showed in section 5.5, but with a different configuration.

The evaluation of the clustering process has been done using V-measure [163]. V-measure is an entropy-based measure which explicitly measures how successfully the criteria of homogeneity and completeness have been satisfied. V-measure is computed as the "mean" of distinct homogeneity and completeness scores, that is, V-measure can be weighted to favour the contributions of homogeneity or completeness. A clustering result satisfies homogeneity if each one of its clusters contains only data points which are members of a single class, and a clustering result satisfies completeness if all the data points that are members of a given class are elements of the same cluster.

The first experiment is performed using all the words of the dataset. The clustering algorithm has two parameters: the number of objects k in a neighbourhood of an object (minimal number of objects considered as a cluster), and the neighbourhood radius (*eps*). The first one is fixed to three elements, and the second one has been experimentally fixed to 0.805. In the Table 5.3 we can observe the obtained results.

Table 5.3: Performance of the density-based clustering using all the words of the database.

# words	56,645
# clusters	38
completeness	92.07
homogeneity	76.74
V-measure	83.55

The results of the second experiment are shown in Table 5.4. It shows the results for the approach introduced in this chapter to select the most frequent word in each semantic class. The clustering is done in two levels. The first one divides the words by geometric shapes properties in short (S), medium (M) and large (L) words. Afterwards, each one is dense-clustered. This experiment has been performed using the fourth first months of the book. We can observe that the completeness of the first level is high in all the semantic classes and the homogeneity is low. The priority

Table 5.4: Performance of the density-based clustering by semantic class using the first 4 months of the database.

Semantic class	level	cluster	compl.	homog.	V-mea.	#classes	#words
KW1-KW2	1st	-	93.18	21.73	35.24	-	1349
	2nd	S	96.10	98.54	97.30	13	
		M	98.37	83.15	90.26	15	
		L	100.00	81.19	90.08	5	
KW2-KW3	1st	-	91.06	21.35	34.59	-	2789
	2nd	S	99.11	41.72	58.72	7	
		M	99.79	82.83	90.52	24	
		L	98.38	86.94	92.31	15	
KW3-KW4	1st	-	90.05	21.15	34.26	-	779
	2nd	S	100.00	56.89	72.52	4	
		M	100.00	92.58	96.15	4	
		L	100.00	100.00	100.00	13	
KW4-end	1st	-	89.70	20.04	32.76	-	2327
	2nd	S	100.00	35.06	51.91	8	
		M	100.00	77.71	87.46	22	
		L	100.00	86.79	92.93	8	
mean	2nd		99.31	76.95	85.01	138	7244

of this level is to cluster all the instances of the same class inside the same cluster. Several classes in the same cluster are allowed. We can observe that the results in the second level present a good performance for the V-measure. It indicates that we have a class in each cluster, and all the instances in a cluster belong to the same class.

Comparing both experiments we can observe that the number of clusters is higher when clustering is computed by semantic classes (138) than clustering all the words of the documents (38). The completeness and homogeneity of the cluster is better clustering by semantic classes. And in addition, clustering by semantic removes the most frequent stop-words, which usually do not contain relevant information of the documents and they are not interesting for document interpretation purposes.

5.7.4 Contextual Word Spotting Results

Once the semantic classes are discovered and the most frequent words are selected, the last step is the contextual word spotting. Each selected word is related to a semantic class. For example, the word *Pere* is a man name, and taking into account the syntactic rules previously mentioned (Section 5.7.1), it can only appear in the semantic class *husband (H)* or *husband's parent (HPI)*. So, we can decide if we would like to search all the husbands called *Pere*, or we are interested only in the father's names.

We have performed four different experiments (Table 5.5). The first two one are performed using deformable HOG-based shape descriptor (nrHOG), and the last two using the LOCI descriptor introduced in the Section 4.4 of the Chapter 4. The

Table 5.5: Performance of the contextual word spotting approach.

Descriptor	input words	classical WS	contextual WS
nrHOG	random	71	77
	perfect	79	83
LOCI	random	60	68
	perfect	66	71

results have been performed using two different configurations. The first one selects N random samples of each cluster of the last step of the framework. And the second one selects by hand the samples for each cluster. It is a prefect dataset where all the instances belong to the same class.

We can observe that the accuracy improves when the contextual information is used. Classical word spotting strategies is improved their results introducing contextual information.

5.7.5 Additional experimental cases

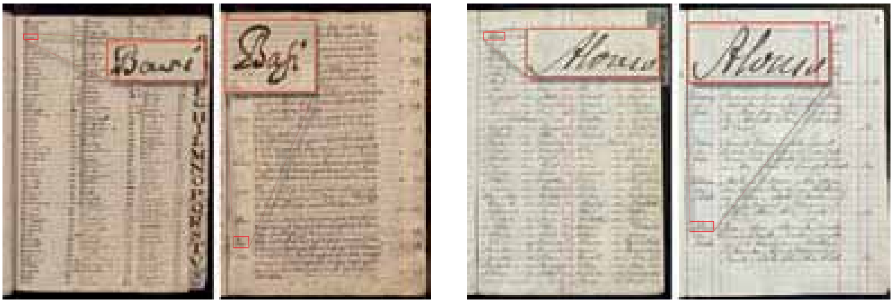
We have performed two additional experiments related with two of the novelties introduced in this chapter. The first one performs the alignment of a pre-set long list of words in front of all the documents. The second experiment shows the advantage of searching using the semantic information.

Sequential word spotting for index alignment

In this experiment we have used a two volumes of the *BH2M* collection. Both volumes have similar characteristics: each book contains the marriage licenses of one or two years. For each book, there is a separate set of index pages. The indices list the husband surnames (in some cases, the wife surnames too) in a pseudo-alphabetical order: the names appear in the chronological order of marriage, listed in sections corresponding to the initial letter. The indices were therefore written a posteriori. In some cases, by a different writer many years later. The indices contain errors and in some cases the page numbers do not correspond to the actual position of the surname entries. The objective of sequential word spotting is to align the indices with the surnames of the licenses, overcoming wrong pagination and improving the performance of individual word spotting reinforced with the context. For the experiments, we have selected the following books:

- *Volume A*: Multi-writer. This volume (see Fig.5.10a) is from the 17th century. The indices and the marriage licences were written by a different author. We have used all the words from the index that begin by the letter *A*. Thus, there are 186 words in the indices, and 1302 words in the licenses.

- *Volume B*: Single-writer. This volume (see Fig. 5.10b) is from the 19th century. Both the indices and the marriage licences are written by the same author. We have used the words that begin with the letter *B*. There are 182 word indices, and 1489 words in the licences.



(a)

(b)

Figure 5.10: Samples of the datasets used in this work: (a) *Volume A*: Multi-writer. (b) *Volume B*: Single-writer. The left column corresponds to indices, and the right column corresponds to the marriage licenses. We observe in the detail an example of words that belong to the same licence. Notice the handwriting variability of the word "Basi" in Volume A.

We have evaluated the performance of the approach comparing our method with a classical Word Spotting approach using several descriptors. The proposed descriptors can be categorized in two groups: single-writer and multi-writer. The first group contains descriptors that obtain very good results in a single-writer scenario, but do not seem robust enough for a multi-writer scenario. The Blurred Shape Model (BSM) descriptor [52] is based on computing the spatial distribution of shape pixels in a set of pre-defined image sub-regions. The Histogram of Oriented Gradients (HOG) descriptor [37] takes the pixel gradient information as the basis to extract features. The approach developed by Rath and Manmatha [151] uses the sequential information of graphemes to extract the features and Dynamic Time Warping (DTW) to compute the distance between them. The deformable HOG-based descriptor (nrHOG) [3] used in along the experiments of this chapter. The second group uses the work of Almazan et al.[5], where an attribute-based approach learns how to embed the word images in a more discriminative space, where the similarity between words is independent of the handwriting style.

The metrics used to evaluate the performance of the experiments are:

- Number of True Positives (TP). A TP is considered when a word of the index is aligned with the correct license.

- TP TOP 5. Instead of evaluating when the searched word is returned in the first position of the vector of (closer) distances, we compute when the word is returned in the top 5 closer positions of the vector (TP TOP 5).

- Mean Distance. We evaluate the mean distance (Mean Dist.) between the computed position and the correct position of the query. This measures how far is the license selected compared to the correct position. So, a small value means that, although the alignment fails, the position of the returned licence is close to the correct position. Formally, let a_i be the word in A assigned to b_j

through the mapping f, and $b_{j'}$ the correct instance of a_i in B (i.e. $b_{j'} = a_i$). The position distance between the computed position and the right position is defined as $pos\_dist(j, j') = |j - j'|$. The mean distance is computed as the average of the *pos_dist* of the failed aligned words.

Table 5.6 shows the comparison of several single-writer descriptors in the single-writer volume. It can be seen that we have outperformed the original word spotting method. In all cases, the accuracy is increased and the mean distance is reduced. Table 5.7 shows the results computed using a single (nrHOG) and a multi-writer descriptor (attributes). Both descriptors are evaluated using a single and a multi-writer dataset. We can observe that we outperform the original word spotting method again. The accuracy is increased and the mean distance is reduced, specially in the case of the single-writer descriptor over the multi-writer dataset. In such a case, one may conclude that the alignment in context is more helpful when the word shapes are dissimilar, and therefore, the shape descriptors less reliable.

Descriptor	TP (Align.)	Mean Dist. (Align.)	TP (WS)	Mean Dist. (WS)	TP TOP 5 (WS)
nrHOG	79 (42.47%)	20.47	18 (9.68%)	210.62	51 (27.42%)
BSM	64 (34.41%)	13.52	10 (5.38%)	168.68	27 (14.52%)
HOG	70 (37.63%)	14.32	20 (10.75%)	258.95	40 (21.51%)
DTW	50 (26.88%)	18.31	7 (3.76%)	214.92	25 (13.44%)

Table 5.6: Evaluating single-writer descriptors using a single-writer dataset (Volume B). *WS* means a classical Word Spotting approach. *Align.* means our proposed alignment for word spotting.

Volume	Descriptor	TP (Align.)	Mean Dist. (Align.)	TP (WS)	Mean Dist. (WS)	TP TOP 5 (WS)
Vol. A (multi-writer)	nrHOG	10 (5.49%)	39.13	2 (1.10%)	683.80	7 (3.85%)
	attributes	68 (37.36%)	11.16	9(4.95%)	238.48	27 (14.84%)
Vol. B (single-writer)	nrHOG	60 (32.26%)	15,74	12 (6.45%)	254.26	31 (16.67%)
	attributes	79 (42.47%)	20.47	18 (9.68%)	210.62	51 (27.42%)

Table 5.7: Evaluating a single-writer descriptor (nrHOG) and a multi-writer (attributes) descriptor using a single and a multi-writer dataset. *WS* means a classical Word Spotting approach. *Align.* means our proposed alignment for word spotting.

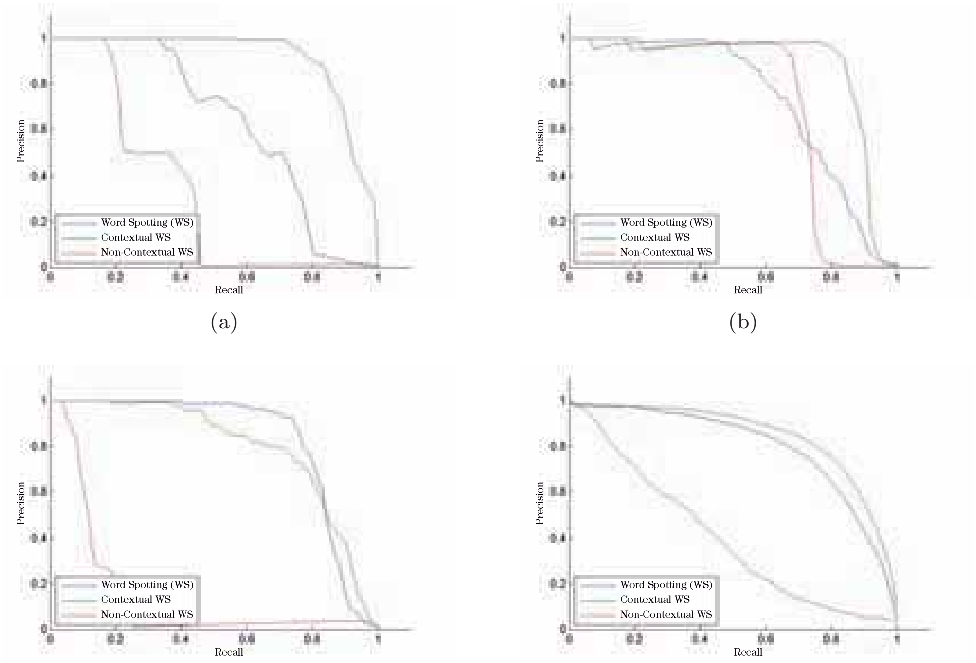
One of the difficulties of this work is to spot a specific instance among a big collection of words. For example, to find the surname *Alonso* that appears is a specific record. Usually, a classical word spotting approach computes all the similar instances to the input query (e.g.- all the similar instance to the surname *Alonso*), but in this case we have to find only one instance. In classical word spotting approaches, this task becomes difficult to achieve because the results is sorted list of words. Hence we have evaluated the word spotting results using the top 5 (we consider that the target instance is in the top 5). But even in that case, the results has been improved in all the experiments.

Semantic searching

Semantic search seeks to improve search accuracy by understanding searcher intent and the contextual meaning of terms as they appear in the searchable data space to generate more relevant results. In our particular problem, we can perform searches like "*All the husbands called Pere*" because the semantic information is established previously. The semantic search focusses the search of words in a specific semantic class. We have showed along all the experiments that contextual/semantic information improve the results achieved by classical word spotting approaches. But contextual information can help us to make the searches more accurate.

The experiment reported here reported how the contextual information helps to achieve better results. In Figure 5.11 we can see the results for several specific words and the mean of all the words. Each plot shows the results for a classical word spotting architecture (all the instances are considered true positives), the results for the contextual word spotting (previously showed in this chapter) and the results when we do a semantic search using a classical word spotting (the words with the same transcription and labelled with the same semantic class of the input query will be considered true positive).

We can observe in all the samples that the semantic search using the classical word spotting achieves poor results. Words with different semantic class are in the results, and they are considered false positives. If we are searching a word of a specific semantic class (e.g. "*All the husbands called Pere*"), instances which do not keep both premises are wrong results. We can observe in Figure 5.11c that the classical word spotting achieves better results than the semantic word spotting. The word "Pere" is a very common name, and it appears in all the semantic classes, which can introduce confusion in the search. Figure 5.11b shows the results for the word "*dilluns*" (*Monday* in old Catalan). We can observe that classical word spotting and the semantic search using the classical word spotting perform similar results. The days of the week are localized in a specific place of the record and they cannot appear in other semantic classes, so the results of the search are similar independently whether contextual information is used or not. Finally, Figure 5.11d shows the mean of the results. We have performed 99 different semantic searches. The set of words is composed by names, surnames, jobs, towns, etc., each one related with a specific semantic class. In the major part the results are achieved. but the instances which appears in all the semantic classes, like jobs, that the result are similar, or lower, to the word spotting approach. In conclusion, the results achieved are better as more



(a)

(b)

(c)

(d)

Figure 5.11: Contextual words spotting results: (a) "*Catherina*": WS(62%); Sem.WS(91%); Non-Sem.WS(32%), (b) "*dilluns*": WS(73%); Sem.WS(83%); Non-Sem.WS(71%), (c) "*pere*": WS(83%); Sem.WS(79%); Non-Sem.WS(15%) and (d) *mean*: WS(79%); Sem.WS(83%); Non-Sem.WS(39%) .

specific is the semantic class.

5.8 Conclusions

In this chapter we have presented a new framework model based in two layers. The first one makes a knowledge discover to extract the semantic/contextual information. Contextual information is extracted through the analysis of frequent terms or repetitive words along the documents. The key words have to follow two premises: the list of key words have to appear along all the text in a repetitive pattern and they have to appear always in the same sequence order. The second layer is centred in the information extraction using the contextual information. The spot of the words is done according to the semantic classes, previously computed.

The main contribution of this chapter is twofold. The first one is a method of knowledge discovery that analyses and extracts the key words of the documents. The key words establish the semantic information of the documents and group words that are semantically related. The second contribution is the information extraction using the contextual information.

We have tested the framework using the *BH2M* database. This collection presents a repetitive structure along their licenses. The performed experiments show the importance of the contextual information to achieve better results than classical word spotting. We have showed two possible applications of the framework presented in this chapter. The first one make the alignment of an index in front of the documents. In this case the sequence order of the words allows to improve the results of classical word spotting approaches. The spot of a word depends of the words that are before and behind. The second application complements the search of words using the semantic information. The search of the words is done specifying the semantic class.

Chapter 6

Markov Logic Network-based parsing

In this chapter we propose the use of Markov Logic Networks (MLN) to improve the results of word spotting. MLN is a very powerful statistical relational learning model that provides a very rich representation. The use of MLN to model a grammatical structure offers more flexibility in the definition of the rules, incremental and simple learning, with respect to traditional language models used in handwriting recognition. Markov Logic Networks can be used to learn the probability of the order in which the different words appear and integrate this information with the output of the Word-Spotting retrieval. As experimental setup, the *BH2M* collection has been used. This collection is highly structured and the grammar has been experimentally extracted. Each segment of the record has a pre-defined grammar, that it is used to improve the performance of the word-spotting approach. The propose approach has been evaluated using different scenarios, achieving good good results.

6.1 Introduction

Throughout this thesis, we show an alternative way to use the context to improve the performance of word spotting. In historical demographic documents, some words have high probability of co-ocurrence. For example, if we have genealogic linkage, we can learn joint probabilities between family names, some common words in the record like ”married to” determine the position of the searched ones, migration movements from geographic areas also generate clusters of family names that can be linked to city names, etc. We particularly focus 0n the syntactic context intra-sentences. The use of dictionaries is a common approach to model this context [72]. However, there is the drawback that lexicons constructed generically from a language do not work properly in historical documents where the contents are very specific in terms of topic and time period. The use of closed dictionaries is corpus-specific and practically unfeasible.

In artificial intelligence, one of the open questions is concerned with techniques for combining expressive knowledge representation formalisms (such as relational and first-order logic) with principled probabilistic and statistical approaches used to learn and infer. Probabilistic and statistical methods refer to the use of probabilistic representations and reasoning mechanism grounded in probability theory, such as Bayesian networks, Hidden Markov Models and probabilistic grammars, and the use of statistical learning and inference techniques.

A stochastic context-free grammar (SCFG, or probabilistic context-free grammar, PCFG) is a context-free grammar where a probability is associated to each production rule. In SCFG, the probability of a derivation is the product of the probabilities of the productions. SCFG has been used in different domains, such as Natural Language Processing. In these applications, SCFG are modelled as grammars, typically specified in syntaxes where the rules are absolute. Some speech recognition systems use SCFGs to improve their probability estimate and thereby their performance [18].

Uncertainty and complex relational structure characterize many real-world application domains. Statistical learning is related to uncertainty while relational learning deals with relational information. Statistical relational learning (SRL) [94] attempts to combine the best of both. SRL is a combination of statistical learning which addresses uncertainly in data and relational learning which deals with complex relational structures. There is an increasing interest to develop SRL approaches such as stochastic logic programs [36], probabilistic relational models [64], relational Markov models [9], structural logistic regression [148], and others.

We propose the use of Markov Logic Networks (MLN) [155] to improve the results of word spotting according to the stated hypothesis. We introduce grammar rules to classical word spotting methodologies to improve the results achieved. We have developed a word spotting approach guided by grammars. MLN is a very powerful statistical relational learning model that provides a very rich representation. The use of MLN to model a grammatical structure offers more flexibility in the definition of the rules, incremental and simple learning, with respect to traditional language models used in handwriting recognition. Markov Logic Networks can be used to learn the probability of the order in which the different words appear and integrate this information with the output of the Word-Spotting retrieval. Fabian et al. [55] proposed a Markov Logic Model which incorporates the contextual information in the form of expectations of a dialogue system to perform semantic processing in a spoken Dialogue System.

As experimental setup, the *BH2M* collection has been used (for more details see Chapter 7). The documents are semi-structured in records (paragraphs). Each record contains the information of a marriage using a regular structure, but with some variations from one period to another, or from one social status to another (for more details see Chapter 7). As we have seen in the Chapter 5 this collection contain contextual/semantic information related with the documents. This contextual information is defined through the key words, which are instances in the text that split the document by semantic information. Each segment of the record has a predefined grammar, which can be extracted experimentally. The objective is to do a words spotting guided by the grammar predefined previously.

The rest of the Chapter is structured as follows. Section 6.2 shows an overview of the MLN. Section 6.3 describes the proposed MLN-based method. Section 6.4 shows the experimental results. Finally, we present the conclusions in the last section of the chapter.

6.2 Markov Logic Networks

Markov Logic Network (MLN) is one of the most well-known methods proposed for SLR [44, 186]. MLNs extend first-order logic and associate a weight to each formula. Semantically, they can represent a probability distribution over possible worlds using formulas and their corresponding weights. Semantically, weighted formulas are viewed as templates for constructing Markov Networks. This yields a well-defined probability distribution in which worlds are more likely when they satisfy a higher-weight seto of ground formulas. Intuitively, the magnitude of the weight corresponds to the relative strength of its formula; in the infinite-weight limit, Markov Logic reduces to first-order logic.

Markov Logic has already been used to efficiently develop state of the art models for entity resolution, ontology induction, information extraction, social networks, collective classification, and many other problems important to the Semantic Web. Several applications are developed using MLN as a basis to infer some knowledge of the world. In [184] the application of MLN as a language for learning classifiers is investigated. In [73] a goal recognition framework based on MLN is presented.

First-Order Logic

A *first-order knowledge base* (KB) is a set of sentences or formulas in first order logic. Formulas are built using four types of symbols: *constants*, *logical variables*, *functions*, and *predicates*. Constant symbols represents objects in the domain of interest. Logical variables symbols range over the objects in the domain. Functions represent mappings from tuples of objects to objects, and *predicates* representing relations among objects in the domain or attributes of objects.

A formula is *satisfiable* if there is exists at least one world in which is true. The basic inference problem in *first-order logic* is to determine whether a KB entails a formula. If a world violates even a formula, it has probability zero. A KB can thus be interpreted as a set of hard constraints on the set of possible worlds. Markov logic networks soften these constraints so that when a world violates a formula in the KB it becomes less probable, but not impossible. The fewer formulas a world violates, the more probable it is.

Markov Networks

A *Markov Network* (also known as *Markov random field* is a model for the joint distribution of a set of variables $X = (X_1, X_2, \ldots, X_n)$ [109]. It is composed of an undirected graph G and a set of potential function ϕ_k. each node of the graph is a variable, and each clique has a potential function. The join distribution represented by Markov network is given by

$$P(X = x) = \frac{1}{Z} \prod_k \phi_k(x_{\{k\}}) \tag{6.1}$$

where $x_{\{k\}}$ is the state of kth clique. Z, known as the *partition function*, is given by $Z = \sum_{x \in X} \prod_k \phi_k(x_{\{k\}})$.

Inference in Markov networks is #P-complete. The most widely used method for approximate inference is Markov chain Monte Carlo (MCMC) [10], and in particular Gibbs sampling, which proceeds by sampling each variable in turn given its blanket.

Markov Logic

A Markov logic network L is a set of pairs (F_i, w_i), where F_i is a formula in first-order logic and w_i is a real number. Together with a finite set of constants $C = \{c_1, c_2, \ldots, c_{|C|}\}$, it defines a Markov network $M_{L,C}$ (Equation 6.2) as follows:

- $M_{L,C}$ contains a binary node for each possible grounding of each predicate appearing in L. The value of the node is 1 if the ground predicate is true and 0 otherwise.

- $M_{L,C}$ contains a feature for each possible grounding of each formula F_i in L. The value of this feature is 1 if the ground formula is true and 0 otherwise. The weight of the feature is the w_i associated with F_i in L

$$P\left(X = x\right) = \frac{1}{Z} exp \left(\sum_j w_j f_j \left(x\right) \right) \tag{6.2}$$

A world is an assignment of truth values to all possible ground atoms. Each state of the Markov network presents a possible world. The probability distribution over possible worlds x specified by the ground network is calculated by Equation 6.2, where $f_j\left(x\right)$ is the number of true groundings for F_i in x and Z is the partition function that is used to make the summation of all possible groundings adds up to one.

Inference

Recall that an MLN acts as a template for a Markov Network. Therefore, we can always answer probabilistic queries using standard Markov network inference methods on the instantiated network. Inference has two main phases in MLNs. In the first phase, a minimal subset of the ground Markov network is selected. Many predicates that are independent of the query predicates may be filtered in this phase. As a result, the inference can be carried out over a smaller Markov network. In the second phase the inference is performed on the Markov network using Gibbs sampling [26] where the evidence nodes are observed and are set to their values.

6.3 Method

The results of the word spotting are improved using the contextual information of the documents and MLN. In this section the database used in the experiments is illustrated briefly, the word spotting approach used in the first step is presented, and

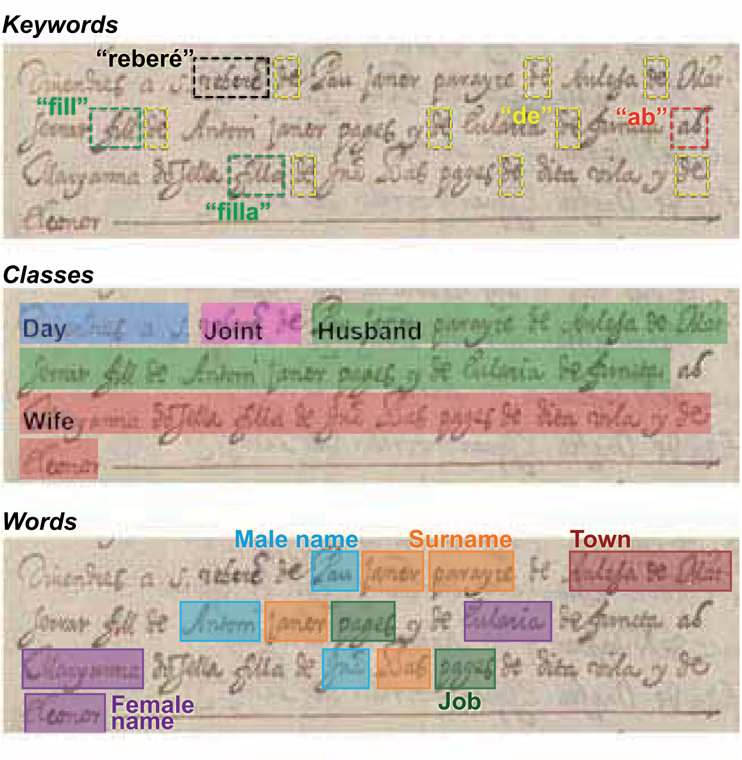


Figure 6.1: Grammar structure of a record.

the rules used to learn and infer the contextual information of the documents are discussed.

6.3.1 Structure of the Dataset

The marriage license records present a regular structure in all of them even if the number of words in each part changes from record to record (Figure 6.1 - *Classes*). The first part of the record specifies the day that the marriage took place. The next words are the key-words *Reberé de* (which means *Received from* in Catalan). Following these two words, information related to the husband is found. And finally, information related to the wife is showed after the key-word *ab* (*with* in Catalan).

There are some key-words in the records (See Figure 6.1 - *Keywords*) that always appear in all of them: *reberé*, *de*, *ab*, *fill* and *filla* (*receive*, *the*, *with*, *son* and *daughter* in old Catalan). These key-words always appear in the same order in the registers, and they usually precede a certain category of information. For example, the day of the marriage always appears before the key-word *reberé*, and after that, the information of the husband is written. The information about the husband is close to the key-word *ab* that indicates the beginning of the wife's information. There are some key-

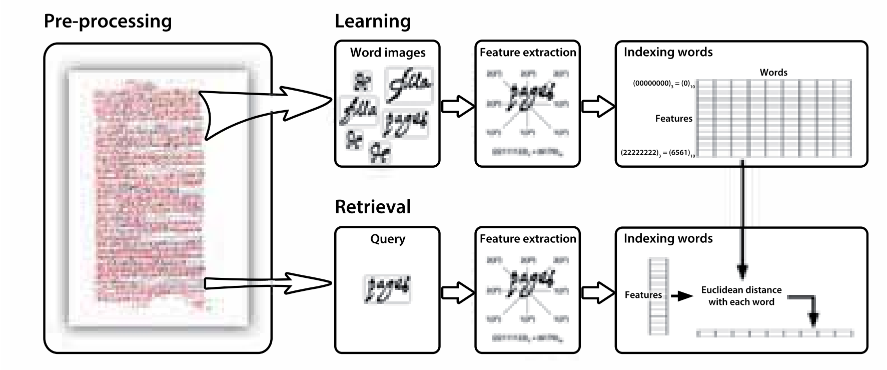


Figure 6.2: Outline of the Word-Spotting approach used in the thesis.

words which indicate some specific information, for instance, after the word *fill*, the husband's father name appears, and after the word *filla*, there is the wife's father name. In this work, we assume that these key-words are localized previously using the approach introduced in the chapter 5.

The other words can be classified in different categories (See Figure 6.1 - *Words*), and usually appear at different positions inside the record.

6.3.2 Word Spotting Approach

The word spotting approach used in this work is the approach presented in the Chapter 4, and follows a query-by-example strategy. For the sake of readability, we summarize it here. Thus, given a query image it locates all the instances of the same word class into the documents, which have been previously indexed. Shape matching techniques are used in the holistic approach. The descriptor used is inspired by Loci characteristics [70], aggregating pseudo-contextual information.

The spotting strategy can be separated into two major modules (Figure 6.2): the indexing and the retrieval stage. First, word images are indexed considering a feature space considering shape features. Second, word images are used as queries and similar instances from the database in terms of shape similarity are retrieved.

Once the words are segmented, one feature vector is computed for each word and is stored in a suitable hash structure. The descriptor is an adaptation to word images of the descriptor devised by Glucksman [70]. A characteristic Loci feature is composed by counting the number of intersections along eight directions (up, down, right, left, and the four diagonals). For each background pixel in the binary image, and for each direction, we count the number of intersections (black/white transitions between pixels). Hence, each key-point generates a codeword (called *Locu number*) which corresponds to a position inside the features vector. Each generated position increments the count in that position of the feature vector. The feature vector can be seen as a histogram of *Locu numbers*.

Basically, the retrieval process consists in organizing the feature codewords in a look up table *M*, whereas the classification process consists in searching the best matching of the query with all the words of M.

6.3.3 Markov Logic Networks for Marriage Records

Since marriage records have a regular but not fixed structure it is possible to model this structure with statistical grammar.This allows to identify the most probable parse of a sentence given a probabilistic context-free grammar (CFG). This grammar is then translated into an MLN.

A Markov Logic Network [41] is a probabilistic logic which applies the ideas of a Markov network to the first-order logic, enabling uncertain inference. The MLN can be considered as a collection of first-order logic rules to each of which it is assigned a real number, the weight. Each rule represents a rule in the domain, while the weights indicate the strength of the rule (see Section 6.2). Next, we describe the grammar translated into an MLN.

To use the MLN framework in our application we mapped the structure of the records in the marriage dataset in a weighted Context-Free grammar (CFG) in Chomsky normal form:

$G = (V, \Sigma, PR, R)$

where V are the non-terminal symbols, R is the start variable and corresponds to the part of the record selected. Σ are the terminal symbols, which define all the tokens (in our case handwritten words) that appear in the document. And finally, PR are the production rules. Next, we define the CFG for the day and the husband part of the record.

The first part of the record is the day information. It covers from the beginning of the record until the key-words *rebere de*. The Husband information is the next part of the record. It covers from the next word of the key-words *Rebere de* until the key-word *ab*. The Wife information is the last part of the record. It covers from the next word of the key-word *ab* until the end of the record. The CFG is defined by the following production rules (PR):

$$\begin{array}{rcl}
R & \rightarrow & D\ P\ H\ [HF]\ ab\ W\ [WF] \\
D & \rightarrow & name\ name\ [name] \\
P & \rightarrow & rebere\ de \\
H & \rightarrow & names \\
HF & \rightarrow & fill\ de\ names \\
W & \rightarrow & names \\
WF & \rightarrow & filla\ de\ names \\
names & \rightarrow & name\ [name]\ [name]\ [name]
\end{array}$$

where the non-terminal symbols are R, D, P, H, HF and $names$. R is the start variable and corresponds to the entire record, D that is for the part of the record that represents the day of the wedding, P is the key-words (*rebere de*), H is the part of the record that represents the part of the husband information, HF is the (optional) part of the record that covers the information of the husband's parents, W is the part of the record that represents the part of the wife information, WF is the (optional) part of the record that covers the information of the wife's parents and $names$ is a list of non-recognized words. The terminal *de* represents the word *de*, *rebere* represents the word *rebere*, *fill* represents the word *fill*, *ab* represents the word *ab*, *filla* represents the word *filla* and *name* is a class which represents all the other words.

To translate this into an MLN we encode each production rule as a clause, for instance $R \to D\ P$ becomes $D \wedge P \Rightarrow R$. The next step is to denote the position of the words or phrases in the record. For this purpose, each terminal or non-terminal is described as a predicate with two arguments that denote the beginning and end of a record or phrase as well as positions between words. Therefore a record with n words has $n+1$ positions. The MLN formulation is the following:

```
// Definition of R
D(a,b) ∧ P(b,c) ∧ HI(c,d) ∧ ab(d,e) ∧ WI(e,f) => R(a,f)
H(a,b) ∧ HF(b,c) => HI(a,c)
H(a,b) => HI(a,b)
W(a,b) ∧ WF(b,c) => WI(a,c)
W(a,b) => WI(a,b)

// Definition of D
name(a,b) ∧ name(b,c) ∧ name(c,d) => D(a,d)
name(a,b) ∧ name(b,c) => D(a,c)

// Definition of P
name(a,b) ∧ de(b,c) => P(a,c)
name(a,b) ∧ rebere(b,c) => P(a,c)

// Definition of H
names(a,b) ∧ !LessThan(b,a) => H(a,b)

// Definition of HF
fill(a,b) ∧ de(b,c) ∧ names(c,d) ∧ !LessThan(d,a) => HF(a,d)

// Definition of W
names(a,b) ∧ !LessThan(b,a) => W(a,b)

// Definition of WF
filla(a,b) ∧ de(b,c) ∧ names(c,d) ∧ !LessThan(d,a) => WF(a,d)

name(a,b) ∧ name(b,c) ∧ name(c,d) ∧ name(d,e) ∧ !LessThan(e,a) => names(a,e)
name(a,b) ∧ name(b,c) ∧ name(c,d) ∧ !LessThan(d,a) => names(a,d)
name(a,b) ∧ name(b,c) ∧ !LessThan(c,a) => names(a,c)
name(a,b) ∧ !LessThan(b,a) => names(a,b)
```

Here, `a` and `b` indicate the positions of the words. To encode the sequential nature of the records we shall consider a predicate `Succ(j,i)` that states that the position `j` follows position `i`. To reinforce the order of the words, a new predicate is used in this part of the record: `LessThan(e,a)`. It states that the position of the last word `e` goes after the first word `a`.

We should then match the ideal record structure with the noisy output generated by the key-word spotting on the handwritten registers. For this purpose we define a `WordSpot (hword,pos)` predicate that assigns the class `hword` to the word at position `pos`. Possible classes can be considered as filler models and are the occurrences of the key-words *"de"* and *"rebere"* as well as non-recognized words that are labeled as *"short"*, *"medium"*, or *"long"* according to their length. Obviously, in the handwriting recognition there could be false positives and false negatives and this should be reflected by suitable production rules that link the non-terminals with the output of the key-word spotting:

```
// de
WordSpot("de",i) ∧ Succ(j,i) => de(i,j)
WordSpot("short",i) ∧ Succ(j,i) => de(i,j)

// rebere
WordSpot("rebere",i)∧Succ(j,i) => rebere(i,j)
WordSpot("medium",i)∧Succ(j,i) => rebere(i,j)

// name
WordSpot("long",i) ∧ Succ(j,i) => name(i,j)
WordSpot("medium",i) ∧ Succ(j,i) => name(i,j)
WordSpot("short",i) ∧ Succ(j,i) => name(i,j)
```

The above rules take care of possible errors in the recognition. For instance, *de* can correspond either to a word recognized as *de* or to a generic short word.

The specification of the grammar is done by defining extra-rules. These extra-rules make more specific the grammar to the problem presented. next, we present the defined rules.

```
// Extra-Rules
Succ(j,i) ∧ Token("de",j) ∧ Succ(k,j) => de(j,k)
Succ(j,i) ∧ Token("fill",j) ∧ Succ(k,j) => fill(j,k)
Succ(j,i) ∧ Token("filla",j) ∧ Succ(k,j) => filla(j,k)

Token("de",i) ∧ Token("fill",j) ∧ Succ(j,i) => !de(i,j)
Token("TMEDIUM",j) ∧ Token("fill",i) ∧ Succ(j,i) => fill(i,j)
Token("TSHORT",j) ∧ Token("fill",i) ∧ Succ(j,i) ∧ Succ(k,j) => de(j,k)

Token("de",i) ∧ Token("filla",j) ∧ Succ(j,i) => !de(i,j)
Token("TMEDIUM",j) ∧ Token("filla",i) ∧ Succ(j,i) => filla(i,j)
Token("TSHORT",j) ∧ Token("filla",i) ∧ Succ(j,i) ∧ Succ(k,j) => de(j,k)

!(fill(a,b) ∧ fill(c,d) ∧ Succ(b,a) ∧ Succ(d,c))
!(fill(a,b) ∧ fill(b,c) ∧ Succ(b,a) ∧ Succ(c,b))
!(filla(a,b) ∧ filla(c,d) ∧ Succ(b,a) ∧ Succ(d,c))
!(filla(a,b) ∧ filla(b,c) ∧ Succ(b,a) ∧ Succ(c,b))
```

The first three rules encode the words *"fill"*, *"filla"* and *"de"* to terminals *fill*, *filla* and *de* respectively. The fourth rule restrict the apparition of the word *"de"* before the word *"fill"*: before the word *"fill"* never appears the word *"de"*. The next rule complements the last one. If a word *"fill"* appears, and before there is a non-recognized word, it is a terminal fill. The next rule fixes the terminal *de* when a word *"fill"*, or *filla*, appears before a non-recognize word. The next rules are similar to the last three ones, but they are applied to the word *filla*. The last two rules reinforce that two terminals *"fill"*, or *"filla"*, cannot be together.

If there are homonyms belonging to different parts of the record, such as `"medium"` (*rebere* or *name*), then we have to ensure that only one of these parts is assigned. The ambiguities in the lexicon are solved making mutual exclusion rules for each pair of parts as described in the following where the numbers before each rule denote the corresponding weight. The weight of the rules is computed in the training process, but we can establish an initial weight to change the certitude of the rule (in this case very high, meaning certitude).

```
// Mutual exclusion rules
!de(i,j) ∨ !rebere(i,j)
!de(i,j) ∨ !fill(i,j)
!de(i,j) ∨ !filla(i,j)
!de(i,j) ∨ !name(i,j)
!de(i,j) ∨ !names(i,j)
!rebere(i,j) ∨ !fill(i,j)
!rebere(i,j) ∨ !filla(i,j)
!rebere(i,j) ∨ !name(i,j)
!rebere(i,j) ∨ !names(i,j)
!fill(i,j) ∨ !name(i,j)
!fill(i,j) ∨ !names(i,j)
!filla(i,j) ∨ !name(i,j)
!filla(i,j) ∨ !names(i,j)

!D(i,j) ∨ !P(i,j)
!D(i,j) ∨ !R(i,j)
!D(i,j) ∨ !H(i,j)
!D(i,j) ∨ !HF(i,j)
!D(i,j) ∨ !HFI(i,j)
!D(i,j) ∨ !W(i,j)
!D(i,j) ∨ !WF(i,j)
!D(i,j) ∨ !WFI(i,j)
!P(i,j) ∨ !R(i,j)
!P(i,j) ∨ !H(i,j)
!P(i,j) ∨ !HF(i,j)
!P(i,j) ∨ !HFI(i,j)
!P(i,j) ∨ !W(i,j)
!P(i,j) ∨ !WF(i,j)
!P(i,j) ∨ !WFI(i,j)
```

```
!H(i,j) ∨ !HF(i,j)
!H(i,j) ∨ !HFI(i,j)
!W(i,j) ∨ !WF(i,j)
!W(i,j) ∨ !WFI(i,j)
!HF(i,j) ∨ !R(i,j)
!WF(i,j) ∨ !R(i,j)

D(a,b) ∧ P(b,c)
H(a,b) ∧ HF(b,c)
W(a,b) ∧ WF(b,c)

!D(a,a)
!P(a,a)
!R(a,a)
!H(a,a)
!HF(a,a)
!W(a,a)
!WF(a,a)
!R(a,a)
```

The last step for using MLN is the training of weights associated to rules. The weights are learned taking into account labeled training data. The training data are the records recognized with the word spotting integrated with information from the ground-truth. Rules that are most often true will obtain higher weights while rules that are sometimes violated (for instance due to errors in the word spotting approach) will obtain lower weights.

Each record in the training set is described by assigning the appropriate values to the previous predicates. An example is shown next. It corresponds to a record where the text `dit dia rebere de` has been recognized as `"short"`, `"short"`, `"rebere"`, `"short"` (in this case the `de` key-word was not properly recognized).

```
WordSpot("short",0)
WordSpot("short",1)
WordSpot("rebere",2)
WordSpot("short",3)

R(0,4)
D(0,2)
P(2,4)

name(0,1)
name(1,2)
rebere(2,3)
de(3,4)

Succ(1,0)
Succ(2,1)
Succ(3,2)
Succ(4,3)
```

Here, in the first part of the training file, we define the words in the record that are recognized by word spotting, then the position and order of each non-terminal. At the end we define the order of the terminals. Likewise, the test data are generated from the output of the word-spotting approach without considering the ground truth information. For each record the following information is generated:

```
WordSpot(”rebere”,0)
WordSpot(”short”,1)
WordSpot(”medium”,2)
WordSpot(”short”,3)
WordSpot(”TWORD”,4)
WordSpot(”TWORD”,5)
WordSpot(”TWORD”,6)
WordSpot(”TWORD”,7)
WordSpot(”TWORD”,8)
WordSpot(”TWORD”,9)
WordSpot(”TWORD”,10)
WordSpot(”FILL”,11)
WordSpot(”de”,12)
WordSpot(”TWORD”,13)
WordSpot(”TWORD”,14)
WordSpot(”TWORD”,15)
...

R(0,28)
H(0,10)
HF(11,16)
W(17,20)
WF(20,28)

Succ(1,0)
Succ(2,1)
Succ(3,2)
Succ(4,3)
...

LessThan(0,1)
LessThan(0,2)
LessThan(0,3)
LessThan(0,4)
...
```

The structure of training and test files is similar, but the non-terminals are not defined in the test files. The position of the non-terminals, and therefore the labeling of parts of the record according to the two main classes (D and P) is obtained by running the MLN inference on test records.

6.4 Experiments and Results

The Chapter 5 of this thesis shows the importance of using contextual information in the word spotting. The approach is performed in two layers. The first one discovers the contextual information, and the second one, using this contextual information, extracts the most frequent words of each semantic class. The experiments of this chapter are performed using the best results achieved in the first layer.

The contextual information is extracted localizing the key words of the documents. Once the key words are computed, the semantic classes are established in the documents. The performance of this layer have influence in the results of the contextual information. This is the reason that we have developed a MLN-based parsing to validate the key words computed. The use of grammars guiding the output of the word spotting achieve better results in the localization of key words.

The experiments have been performed using the *BH2M* database (see Chapter 7 for more details). The searched keywords are *reberé*, *fill*, *ab*, *filla* and *de*. We have used 188 random records and we have performed two different experiments (Tables 6.1 and 6.4. The first experiment is oriented to validate the key words that split the semantic class that involves de day information $D \rightarrow name\ name\ [name]\ rebere\ de$. The second one validates the key words of the second part of the license marriage: husband (H) and wife (I). In both experiments, the objective is to validate the results achieve in the alignment process of the Chapter 5. In particular, the key words validated are *reberé*, *fill* and *filla*. In both experiments, we have trained using 50 records and we have used different configurations changing the number of registers. In the first configuration, we have trained and test using the same 50 records (from the 50 documents explained before). In the second one, we have trained with 50 records, and test with 188 records. For the third configuration, we have removed the 50 records used in the training step. Some records present big distortions in the output of the word-spotting due to a bad word segmentation. For instance there are some cases with over-segmentation or under-segmentation, producing a non-well-formatted structure. These records have been removed in the experiment explained above. In the last one, we have introduced the non-well-formatted-records.

Some examples of the weighted rules obtained after training by using 50 records are shown below.

```
−2.407 !D(a1,a2) v !P(a2,a3) v R(a4,a3)
0.4005 D(a1,a2) v !nom(a1,a3) v ...
−1.198 D(a1,a2) v !nom(a1,a3) v ...
0.3854 P(a1,a2) v !de(a3,a2) v ...
−0.140 P(a1,a2) v !rebere(a3,a2) v ...
−304.7 de(a1,a2) v !WordSpot("DE",a1) v ...
−3.402 de(a1,a2) v !WordSpot("TSHORT",a1) ...
152.15 rebere(a1,a2)v!WordSpot("REBERE",a1)...
−0.055 rebere(a1,a2)v!WordSpot("TMEDIUM",a1)...
```

```
0       nom(a1,a2) v !WordSpot(”TLONG”,a1) ...
76.945  nom(a1,a2)v!WordSpot(”TMEDIUM”,a1)...
4.4305  nom(a1,a2) v !WordSpot(”TSHORT”,a1)...
545.56  de(a1,a2) v !WordSpot(”DE”,a1)...
155.84  rebere(a1,a2) v !WordSpot(”REBERE”,a1)...
0       !de(a1,a2) v !WordSpot(”DE”,a1) v ...
0       !de(a1,a2) v !WordSpot(”DE”,a3) v ...
0       rebere(a1,a2) v !WordSpot(”TSHORT”,a2)...
0       de(a1,a2) v !WordSpot(”TSHORT”,a1) v ...
1.7022  !rebere(a1,a2) v !WordSpot(”REBERE”,a1)...
851.67  !de(a1,a2) v !rebere(a1,a2)
872.73  !de(a1,a2) v !nom(a1,a2)
878.00  !rebere(a1,a2) v !nom(a1,a2)
893.99  !D(a1,a2) v !P(a1,a2)
974.20  !D(a1,a2) v !R(a1,a2)
899.24  !P(a1,a2) v !R(a1,a2)
9.8652  !D(a1,a2) v !P(a2,a3)
1046.2  !D(a1,a1)
970.24  !P(a1,a1)
733.35  !R(a1,a1)
−3.212  D(a1,a2)
−4.065  P(a1,a2)
−2.424  R(a1,a2)
−243.0  de(a1,a2)
−156.6  rebere(a1,a2)
−3.401  nom(a1,a2)
0       WordSpot(a1,a2)
0       Succ(a1,a2)
```

We can observe that each rule has a weight corresponding to its importance The MLN algorithm analyses the train dataset to establish the weight of the rules. The logic network is made taking into account the train samples. The most used rules, or induced, are highly weighted. The rules that are not induced, are weighted as negative samples. For example, the rule: `851.671 !de(a1,a2) v !rebere(a1,a2)` has a high weight because it means that one word cannot be both *”DE”* and *”REBERE”*.

Using these weighted rules, we have computed the results shown in Table 6.1 for the *Day* information part. It can be seen that we have outperformed the original word spotting method. In all the experiments, we have reduced the number of false positives and we have increased the true negatives samples. In addition to this, the precision is increased in all the cases, as shown by the F_1 score.

In Table 6.2 we can observe an example of the results. The columns are: the word spotting, the MLN output and the ground truth values. We can observe that the output of the word spotting has two wrong items. The word spotting approach computes the word `DE` in the first position of the record, and the word `REBERE`, situated in the third position, is localized correctly. Once the MLN is applied, the word `DE` is replaced by the class `nom` and the class `REBERE` is discovered in the third position. We can observe that the second position is `SHORT`, it is a different class of the ground

Experiment	Method	TP	FP	TN	FN	Precision	Recall	F1
50 samples	MLN	63	**24**	**75**	13	**0.72**	0.82	**0.77**
	Word Spotting	63	51	48	13	0.55	0.82	0.66
188 samples	MLN	278	**98**	**270**	31	**0.73**	0.89	**0.81**
	Word Spotting	278	195	173	31	0.58	0.89	0.71
138 samples	MLN	200	**69**	**198**	35	**0.74**	0.85	**0.79**
	Word Spotting	215	144	125	18	0.59	0.92	0.72
150 samples	MLN	215	**78**	**244**	23	**0.73**	0.90	**0.80**
	Word Spotting	215	157	166	22	0.57	0.90	0.70

Table 6.1: Accuracy for the *Day* experiments.

Table 6.2: Example of the results in the *day* part of a licence marriage.

word spotting	MLN	Groundtruth
DE	name	name
SHORT	SHORT	name
MEDIUM	REBERE	REBERE
DE	DE	DE

truth. But it is not a false positive because the classes `SHORT`, `MEDIUM` and `LONG` are considered as equivalents to the class `nom`.

In Table 6.4 we show the results for the *Husband* information part. We observe that the results are quite similar to the results of the day part. The number of false positives are reduced and the number of true negatives increased. In addition, the precision is increased in all the cases.

If we compare both tables, we observe that the day part achieves better results than the husband part. The husband part is composed by more words than in the day part. The number of states in the Markov network exponentially increase with the number of words. Despite this, the algorithm discovers some false positives and increase the number of true negatives.

In Table 6.3 we can observe an example where the MLN algorithm introduces wrong results. The grammar used in these experiments is experimentally determined, and the number of words before and after the terminal *filla* is fixed up to four. We can observe that the number of words in this example is significantly higher. This situation is not contemplated in our grammar and the behaviour of the algorithm is inexact. The first word `DE` is not recognized and the MLN algorithm introduces a word `DE` before the word `FILLA`. Finally two words are misplaced.

6.5 Conclusions

The objective of the work presented in this chapter is a continuation of the work developed in the Chapter 6. Once the key words are aligned in the first layer of the framework, a MLN-based parsing is applied to validate the results achieved. The idea

Table 6.3: Example of the results in the *husband* part of a licence marriage.

Word-Spotting	MLN	GrountTruth
MEDIUM	WORD	WORD
SHORT	WORD	DE
MEDIUM	WORD	WORD
DE	DE	DE
MEDIUM	WORD	WORD
MEDIUM	WORD	WORD
DE	DE	DE
MEDIUM	WORD	WORD
LONG	DE	WORD
SHORT	WORD	WORD
LONG	WORD	WORD
FILL	FILL	FILL
DE	DE	DE
MEDIUM	WORD	WORD
MEDIUM	WORD	WORD
MEDIUM	WORD	WORD
SHORT	WORD	DE
DE	DE	WORD
MEDIUM	WORD	WORD
DE	DE	DE
MEDIUM	WORD	WORD

Experiment	Method	TP	FP	TN	FN	Precision	Recall	F1
50 samples	MLN	28	**2**	**688**	22	**0.93**	0.56	0.70
	Word Spotting	40	5	638	10	0.88	0.80	0.84
188 samples	MLN	91	**30**	**2262**	68	**0.75**	0.57	0.65
	Word Spotting	122	47	2097	30	0.72	0.80	0.76
138 samples	MLN	63	**28**	**1574**	46	**0.69**	0.57	0.63
	Word Spotting	82	42	1459	20	0.66	0.80	0.72
150 samples	MLN	78	**29**	**1118**	59	**0.72**	0.56	0.63
	Word Spotting	89	36	930	25	0.71	0.78	0.74

Table 6.4: Accuracy for the *Husband* experiments.

is to do a word spotting guided by a grammar. We have proved that, using MLN, we reduce the number of false positives and increase the true negatives. Accordingly, we have shown that, using the spatial information, which relates the words of the documents, the results of the word-spotting approaches can be improved.

We have proved that using grammar as guide of the word spotting, the results are improved. The contextual information of highly structured documents is used to remove false positives and discover true negatives.

Chapter 7

BH2M: the Barcelona Historical Handwritten Marriages database

Due the lack of public available databases of historical handwritten documents, with the corresponding ground truth covering all the steps of the interpretation process, a database has been created for validating the entire system presented in this thesis. The database allows validating from layout analysis to recognition and understanding. An image database of historical handwritten marriages records has been created from the archives of Barcelona cathedral, and the corresponding meta-data addressed to evaluate the performance of document analysis algorithms. The contribution creating this database is twofold. First, it presents a complete ground truth which covers the whole pipeline of handwriting recognition research, from layout analysis to recognition and understanding. Second, it is the first dataset in the emerging area of genealogical document analysis, where documents are pseudo-structured manuscripts with specific lexicons and the interest is beyond pure transcriptions but context dependent.

7.1 Introduction

Digitization of historical manuscripts is a priority for archives and libraries worldwide. This process aims first the digital preservation of the documents, but it raises another challenge, the access to contents. The document analysis community has been very prolific during the last decade in the release of tools for (semi)automatic recognition and annotation of historical manuscripts, as services to scholars in social sciences and humanities. Layout analysis [7, 35], including line [15, 23] and word segmentation [174, 180], and recognition, including transcription [32, 127, 161] and word spotting [4, 67, 154, 159], are the most usual tasks of a knowledge extraction process. Whatever is the task, to ensure robust methods, benchmarking databases are needed.

A Ground Truth (GT) designed to validate the interpretation of historical manuscripts has to incorporate several difficulties that hinder the recognition processes and therefore allow to test the algorithms in real and difficult conditions. First, the intrinsic physical effects of degradation over time that result in show-trough, bleed-through, stains, holes, etc. This phenomenon is easily guaranteed if the images come from real books. Second, the use of archaic languages, which requires specific lexicons in the GT to train classifiers. Third, the semantics of the contents themselves, which is probably the most difficulty in a GT generation because it involves the use of contextual knowledge (a palaeographer interprets historical manuscripts using the knowledge of the time and theme context). Hence, it is preferable to have domain specific datasets than generic ones with complex additional meta-data.

A number of reference historical manuscript datasets are being used by the document analysis community. In historical handwriting recognition and word spotting, a key reference is the *George Washington* dataset [152]. It is written in English language and contains 20 pages from a single writer. *Parzival* [60] is a multi-writer historical database and contains 47 pages written in medieval German language. *Rodrigo* [179] is a single writer database written in Old Spanish and contains 853 pages. Databases containing modern handwriting are also commonly used. Among them, one example is the *IAM* database [131]. It is a multi-writer database which contains instances of handwritten English text and it consists of 1,539 pages. The *CASIA* database [112] consists of a dataset written in Chinese and another one in English. It contains 1,074 handwritten texts in on-line format. The *IFNENIT* [146] is a multi writer database of Arabic handwritten texts consists of 2,200 images.

Among the different categories of historical documents, there is a growing interest in the analysis of census, birth, marriage, or death records. A number of crowd-sourcing campaigns exist to transcribe information from such types of documents[1], and relevant works have been published [14, 92]. Search centered at people is very important in historical research, including family history and genealogical research. Queries about a person and his/her connections to other people allow focusing the search to get a picture of a historical context: a person's life, an event, a location at some time period. In this scenario, the challenge is not the transcription, but the understanding of the documents. This allows advanced tasks such as intelligent information extraction, summarization or knowledge discovery [51, 144].

Although such a big effort among the research community for the analysis and interpretation of digitized parish and civil records, there is a lack of specialized GT. Taking advantage of the application project 5CofM in which this thesis is involved, we have contributed with a database consisting in 174 images of manuscripts from the 17th century written over two years[2]. The meta-data provided allows to test the performance of the complete interpretation pipeline, from the layout analysis to the recognition. A second contribution of this work is that, to the best of our knowledge, it is the first database in the area of genealogy research. Hence, the contents of these documents can be appropriately mined and exploited by means of advanced artificial intelligence and machine learning techniques. This is the reason why this database is not only a benchmark for the document analysis community. For many other disciplines it is also an everlasting source of information, that will be a legacy to the following generations.

The rest of the chapter is organized as follows. Section 7.2 describes the marriage

[1] http://familysearch.org/, http://www.ancestry.com/

[2] The database is available at http://dag.cvc.uab.es/5cofm-ground-truth

licenses books collection, the main difficulties and the possible uses, and Section 7.3 draws the conclusion and shows directions for future improvements.

7.2 The *BH2M* books collection

In the 15th century, a centralized register of marriages called *Llibres d'Esposalles* was created in Barcelona. Its purpose was to record all the marriages occurred in Barcelona and surroundings, as well as keep record of the *marriage fee* that was paid. It is nowadays conserved at the Archives of the Barcelona Cathedral and comprises 244 books with information on approximately 550,000 marriages held between 1451 and 1905 in over 250 parishes. For more details the reader is referred to Chapter 1, Section 1.3.1.

Books in the collection *Llibres d'Esposalles* consist of two consecutive sections. The first section is an index with all the husbands' surname on the volume, and the page number where they appear (Fig. 7.1a). The second section is the marriage licenses (Fig. 7.1b, 7.1c and 7.1d). One can clearly appreciate the continuity of the layout during the centuries along the books, but with significant differences in the handwriting styles.

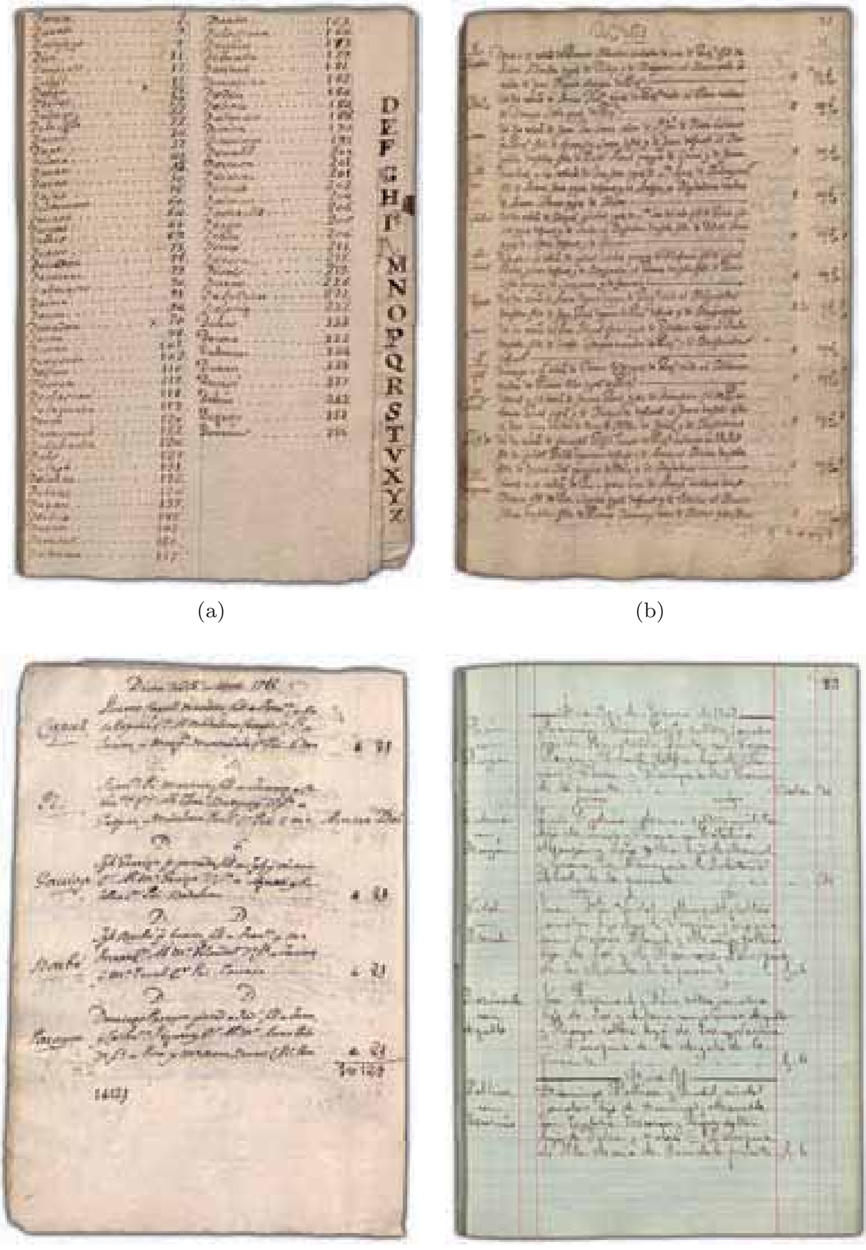
Marriage licenses have also a structured layout (Fig. 7.2), which divides the document in three well-defined parts, or columns: the *husband's surname*, the *license* and the *marriage fee* column. An important characteristic of these documents is that all the licenses present a quite regular structure that can be represented by a syntactic model. The BH2M database has been generated with crow-sourcing tasks under the EU ERC project *Five Centuries of Marriages (5CofM)*. The database consists of two parts. First the social science view that contains the transcribed marriage licences with normalized names, and second, the GT for document analysis and recognition tasks which is presented in this paper.

7.2.1 Difficulties/Challenges

The analysis of historical handwritten documents is not a trivial process due the intrinsic difficulties of this kind of documents (Fig. 7.3): the physical lifetime degradation of the original documents(holes, spots, etc.) and the difficulties introduced in the scanning process. Besides these general difficulties, the characteristics of the handwriting and the configuration of the text lines may provoke additional difficulties (the presence of touching and horizontally-overlapped components [82], curvilinear baselines, etc.). Finally, These documents do not present a standard nomenclature, with different word spelling, special symbols, abbreviations, cross-outs, comments between lines and other recognition challenges. In addition, new words are constantly appearing along their pages. For more details, the reader is referred to the Chapter 1.

7.2.2 Marriage Licenses

The main contribution of the work presented in this chapter is a publicly available database from the collection of Barcelona marriage license books. The dataset pre-


D E F G H I M N O P Q R S T V X Y Z

(a)

(b)

(c)

(d)

Figure 7.1: *Llibre d'esposalles* (Archive of Barcelona Cathedral). (a) 1599: index of volume 60, (b) 1617: volume 69, (c) 1788: volume 158, (d) 1902: volume 242

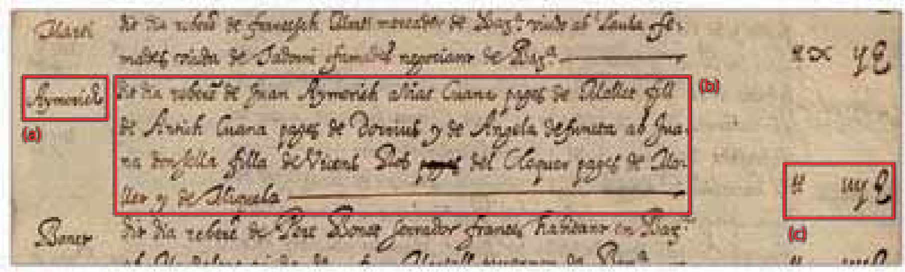


Figure 7.2: Structure of the documents: (a) husband's surname, (b) license, (c) fee of the wedding.

Number of:	Train	Validation	Test	Total
Pages	100	34	40	174
Licenses	998	339	403	1,740
Lines	3,132	1,065	1,301	5,498
Words	32,416	11,089	13,140	56,645
Unique word classes	3,060	1,535	1,757	3,360
Words that appear 1 time	1,831	942	1,100	1,710
Words that appear 2 times	397	213	234	552
Words that appear >2 times	832	380	423	1,098
Out of Vocabulary (OOV) words	–	594	1,082	–

Table 7.1: *BH2M*: Figures.

sented here corresponds to the volume 69, which contains 174 handwritten pages. This database has been compiled using the marriage licenses. This book was written between 1617 and 1619 by a single writer in old Catalan. Table 7.1 shows the figures of this volume.

Generally speaking, the database consists of annotated images. The annotations consist of an XML hierarchical structure (from individual words to blocks of text). The minimum unit of information is a bounding box of an individual word, with the corresponding transcription. Additional attributes like line, register number, or category, are associated to words. This representation allows to easily retrieve a GT from the meta-data adapted to the tasks to be tested (segmentation, word spotting, handwriting recognition, etc.). Three levels in the XML meta-data associated to images can be identified. The first one is designed to evaluate tasks for layout analysis, the second one for text transcription, and the third one for context dependent interpretation. Let us further describe these three levels of meta-data.

Layout structure

Document structure and layout analysis is the first processing phase applied to each page image in order to decompose it into regions. Here, each page consist of different physical blocks: text blocks, paragraph, lines and words. The top layer corresponds to the text blocks of the page. A document has three text blocks: left block or

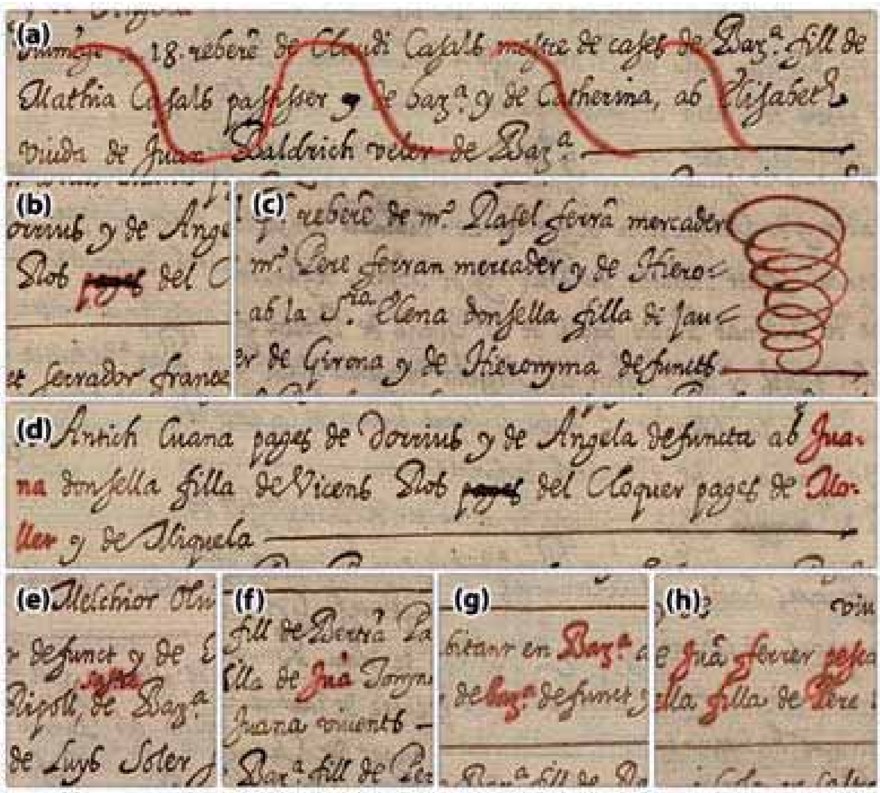


Figure 7.3: Examples of difficulties: (a) a graphical element voiding a license, (b) cross-out words, (c) a license with a special drawing, (d) the words *Jua-na* and *Mo-ller* are split in two lines, (e) the word *sastre* is written between two lines, (f) Abbreviation of *Juana* using the special symbol ˜, (g) Abbreviations of Barcelona (Bar^a) written with different shape and (h) presence of touching and horizontally-overlapped components.

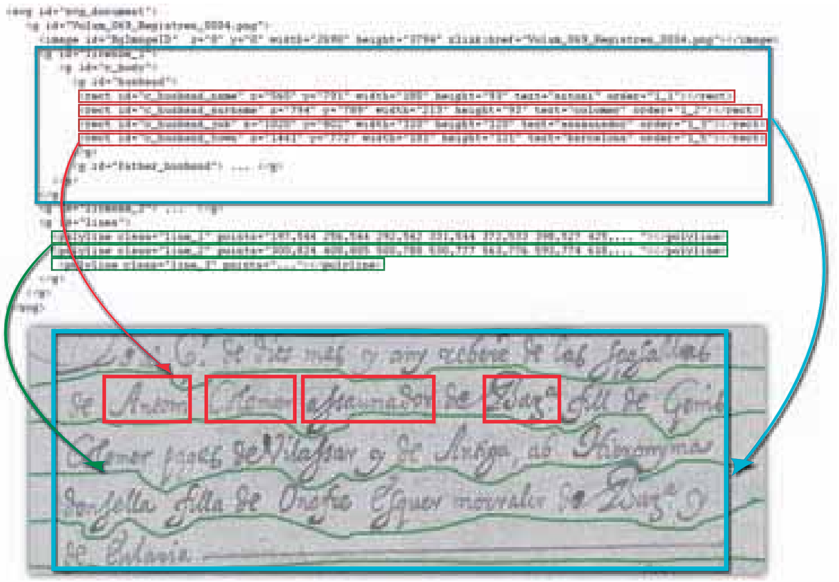


Figure 7.4: A simplified example of a XML file structure containing the information of the documents. We can observe how the information is stored in the XML file and the correspondence of this information in a graphical representation.

husband's surname, right block or *fee* and central block or *license*. This database has been created using the *license* text block. The second level corresponds to segmented lines. An accurate segmentation of the text lines is provided, including the ascenders and descenders of the text line (see Fig. 7.4).

The third level consists of text words (Fig. 7.4 – red boxes). The attributes stored for each word are the following:

- The ID of the page where the word is.
- The bounding box coordinates of the word.
- The text block where the word is located (in case the word is inside a text block).
- The text line where the word is located.
- The appearance order in the text line.
- The word also stores information about special cases. For example, if the word is part of a title, a comment, a cross-out word, etc.

The layout meta-data allows to evaluate algorithms for layout segmentation, from individual words to the text blocks. Depending on the granularity level to evaluate (words, lines, blocks) practitioners must select to corresponding objects in the XML files.

Transcription

The handwritten text was literally transcribed by volunteers using a crowd-sourcing platform [8, 62]. Given the complexity of the handwriting style due to many subtle spelling variants and the language itself, a posterior revision by experts demographers was performed to ensure its correctness. Along the 1,760 licenses, there are more than 56,000 different handwritten words which later conform 3,360 unique word classes. Table 7.1 summarizes some basic figures of the *BH2M* text transcriptions. From the text level document analysis point of view, the presented database is a suitable and syntactically enriched benchmark for various tasks such as word spotting or handwriting recognition.

Word transcriptions follow certain rules in order to unequivocally codify the text. As example, spelling mistakes are not corrected, abbreviations are not expanded; superscript characters are denoted by the precedence of the upper symbol " ˆ " and bounded by brackets (e.g. Barnˆ(a)); superscript of word ending character *m* (an upper stroke over the last characters of the word) is denoted by symbol $ (e.g. rebere$). The complete list of rules transcriptions are:

- It is made a literal transcription of the words, taking care of capital letters, accent and punctuation marks.

- Words with superscript characters are transcribed using the symbol ($\wedge$). Superscripts charter will be written after the symbol, and they will be between brackets (Fig. 7.5a).

- Abbreviated words, which suppress the last character of the word, will be ended with the symbol ($) (Fig. 7.5b).

- Cross out words, which can be read, will be transcribed between square brackets ([*text*]) (Fig. 7.5c).

- Comments between two text lines, because the writer forgot to write, will be written after the symbols ($\wedge\wedge$) and they will be written between brackets (Fig. 7.5d).

- Intentional spaces will be transcribed by 5 spaces (Fig. 7.5e).

- Illegible words will be transcribed by 5 *x* between two forward slash ($/xxxxx/$) (Fig. 7.5f).

- Final lines, which stablish the end of the record, will transcribed using 5 dashes ($-----$) (Fig. 7.5g).

- When a license is entire cross out, text lines will be between square brackets ([[*textline*]]) (Fig. 7.5h).

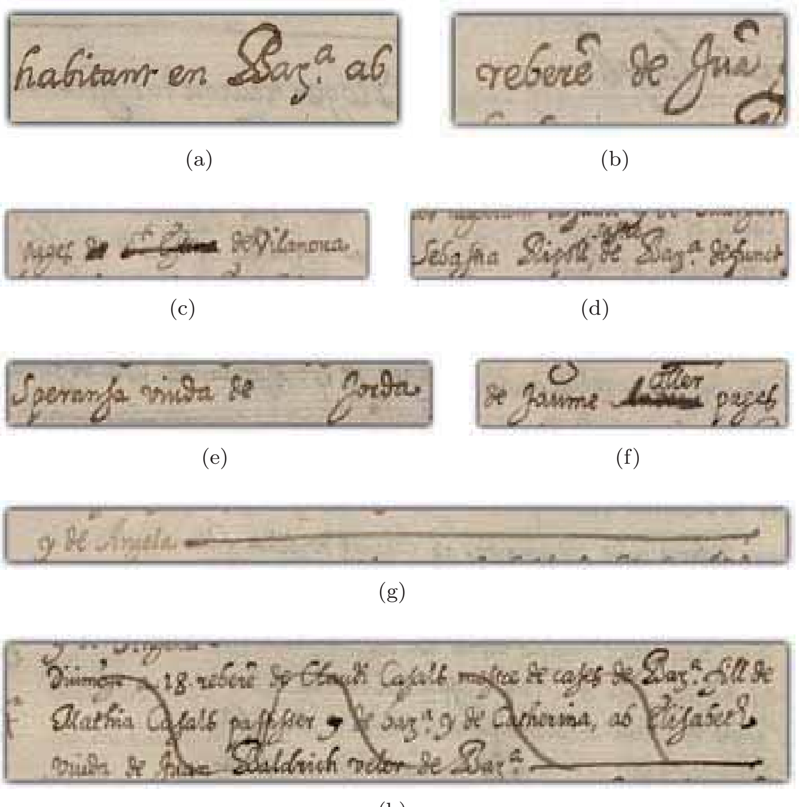

(a) (b) (c) (d) (e) (f) (g) (h)

Figure 7.5: Rule transcriptions. (a) *habitant en Barc∧(a)*, (b) *Rebere$ de Jua$*, (c) *pages* [*de*] [*S∧(t)*] [*Esteva*] *de Vilanova*, (d) *Sebastia Ripoll, ∧∧(sastre) de Barc∧(a). defunct*, (e) *Speransa viuda de Jorda, (f) y de Angela −−−−−*, (g) [[*Diumenge a. 18 rebere$ de...*]] [[*Mathia Casals pastisser y...*]] [[*viuda de Juan Baldrich...*]]

Semantic information

In addition to word transcription and location, the atomic units of this database are also labelled with meta-text information. Next, we describe the mentioned semantic information.

Each marriage license has the purpose of accounting for a prospective marriage. Hence the licenses contain similar information and structure, albeit the structure can vary over centuries. The general sub-parts of a license (L) are, in appearance order, the date-related (D) information, the husband-related (H) information and the wife-related (W) information. These three sub-parts are joined by keywords as follows: date (*DI*) and husband parts are connected by *rebere de* (*we received from* in old Catalan), husband and wife (*HI* and *WI*) information are connected by *ab* (abbreviation of *with* in old Catalan). Both husband and wife parts can be divided in two sub-parts, being his/her own information (e.g. name, surname, home-town) and the correspondent parents (*HPI* and *WPI*) information (*e.g.* father's name, deceased parents). These two last parts are connected by the keywords *fill de* and *filla de* (*son of* and *daughter of* in old Catalan) respectively. Formally, the syntactic rules are:

$$\begin{array}{ll} D \wedge H \wedge W & \Rightarrow L \\ DI \wedge rebere \wedge de & \Rightarrow D \\ HI \vee (fill \wedge de \wedge HPI) & \Rightarrow H \\ ab \wedge WI \vee (filla \wedge de \wedge WPI) & \Rightarrow W \end{array}$$

The semantic information is stored in the text word layout. The words are first categorized in several classes: *husband*, *wife*, *husband family*, *wife family* and other information. Each of these general class tags can be accompanied by more specific sub-class tags in a semantic and ontology-like way. In the XML example in Fig. 7.4 we show that *license 1* contains two classes: *husband* and *father husband*. In this case, each category contains a list of related tags: *husband name*, *husband surname*, *husband job* and *husband town*.

This class labelling system should not be seen as a complete parsing tree, since there are no word-level semantic labels (*e.g.* in a compound name, both words have the same label), but as a useful source of syntactic and semantic information. This semantic information can certainly be used to improve classic document analysis tasks approaches. But we believe that the real and novel value of this database is beyond this, as we will discuss next.

7.2.3 Uses of the database

Handwritten databases, either contemporary like *IAM* [131] and historical like *George Washington* [152], are generally designed for handwriting recognition. The main novelty of BH2M is that it covers the whole pipeline, including the specificity of the domain. So it also allows to test context dependent interpretation algorithms.

As stated in the introduction of this chapter, this database has some distinctive potentials. First of all, these highly structured documents require not only specific lexicons, but also specific language models. Secondly, and due to immigration, these documents contain a large amount of unknown new words (e.g. names, surnames, places), which are challenging for handwriting recognition approaches that deal with open vocabulary. Concretely, the test set contains a 32% of Out of Vocabulary Words (see Table 7.1), which are the number of running words for each partition that do not appear in the other partitions. Finally, these documents can be used for research in information extraction: the system can associate a semantic category to each word.

7.3 Conclusion

In this chapter we have presented the Barcelona Historical Handwritten Marriages database. The data is compiled from a marriage license book collection and it is a useful tool for different research lines in handwriting document analysis. We have introduced a database that can be used for different research tasks, covering from layout analysis to text recognition. In addition to the classical document analysis tasks, the database can be used for research in knowledge extraction, cross-linkage and document understanding.

Chapter 8

Conclusions and Future Work

In this chapter, we summarize the contributions of this thesis to the field of historical document image analysis. In particular, the use of context to improve classical word spotting approaches. Afterwards, we discuss the performance of the proposed methods and their limitations. Finally, future work is presented.

This thesis has addressed the task of word spotting in handwritten documents, in particular in documents that are highly structured, like parish books, census, civil records, counts, etc. These kind of documents present inherent contextual/semantic information that can be used to improve the results of the classical word spotting approaches. We have introduced a framework that analyses the documents and extracts the contextual information automatically. This information is used to transcribe more efficiently the words that appear more frequently in the documents.

This last chapter is organized as follows. In Section 8.1, the summary and contributions of this work are described, whereas in Section 8.2, we discuss about the advantages and limitations of the proposed methods. Finally, Section 8.3 proposes future work.

8.1 Summary and contribution

The main contribution of this thesis has been the proposal of a contextual word spotting framework for highly structured historical handwritten documents. The main hypothesis is that the use of context boosts the recognition. It consists in a framework that, after the analysing the layout of the image (in which the lines and words are segmented), it finds words into the documents using the contextual information.

Despite the main contribution of this thesis has been to propose a contextual word spotting method in highly structures documents, more particularly, the contribution can be divided in two folds: from the scientific point of view and from an application point of view. Let us summarize the main contributions.

From the **scientific** point of view, the contribution has been the study of the existing word spotting approaches oriented to historical handwritten documents and the improvement of the results achieved by these approaches incorporating the contextual information in the search of words. Next we summarize the particular contributions.

Layout Analysis. The contextual word spotting framework introduced in this thesis is a segmentation-based word spotting approach, therefore an efficient word segmentation is needed. Historical handwritten documents present some common difficulties that can difficult the extraction of the words. We have proposed a line segmentation approach that formulates the problem of line segmentation as finding the central part path in the area between two consecutive lines. This is solved as a graph traversal problem. A path finding algorithm is used to find the optimal path in a graph, previously computed, between the text lines. Once the text lines are extracted, words are localized inside the text lines using a word segmentation technique of the state of the art.

Word Spotting. The main contribution of this thesis is a contextual word spotting framework, so a previous study of the methods of the literature oriented to handwritten documents was required. We have evaluated several approaches of the state of the art classifying them according to the classical taxonomy that divides pattern recognition into statistical and structural approaches. In this study, we have shown that the performance of a handwritten word spotting approach does no only rely on the features but also on the interest point model over which the features are computed. This study shows that pseudo-structural methods are more robust in front of variations in the writer style, and dense interest points store more information of the images and increase the robustness in front of variations of the styles.

Taking into account the previous study, we have developed a pseudo-structural descriptor that uses a set of dense interest points to compute local features. This descriptor is oriented to historical handwritten documents. The descriptor encodes the frequency of intersection counts for a given interest point in different direction paths starting from this point.

Contextual Word Spotting. Classical word spotting approaches can be improved using the contextual information of the documents. We have introduced a new framework, oriented to handwritten documents which present a highly structure along their pages, to extract information using contextual information. The main contribution has been two fold. First, it automatically discovers the contextual information by analysing the documents. The approach recognizes repetitive structures and categorizes all the words according to semantic classes. Second, an efficient tool for semi-automatic transcription is oriented to discover the most frequent words in each semantic class. This approach uses a density-based clustering method to extract the most frequent words in the different semantic classes. The representative of each cluster is manually labelled, so the rest of the equivalent instances within the cluster are automatically spotted.

MLN-based Parsing. Contextual information presents a big variety of possibilities to improve the results of classical word spotting approaches. We have proposed the use of Markov Logic Networks (MLN) to improve the results of classical word spotting approaches. The introduced approach uses MLN to model a grammatical structure that offers more flexibility in the definition of rules, incremental and simple learning, with respect to traditional language models used in handwriting recognition. The output of classical word spotting approaches has been integrated with a grammar that has been previously weighted using the probability of the order in which the different words appear.

Validation Framework. Finally, the last contribution has been the construction of a validation framework for contextual word spotting. An image database of historical handwritten marriage records has been created from the archives of the Barcelona cathedral. The contribution creating this database is twofold. First, it presents a complete ground truth which covers the whole pipeline of handwritten recognition research, from layout analysis to recognition and understanding. Second, it is the first dataset in the emerging area of genealogical document analysis, where documents are pseudo-structured manuscripts with specific lexicons and the interest is beyond pure transcription but context dependent.

From the **application** point of view, the main contribution has been to develop a system for structural handwritten word spotting that achieves good results in a real-world problem. The development of this thesis has been involved in the framework of the *5CofM* project, and the main objective has been to develop a system for historians and demographic researches in order to create a database of the entire collection of books in a shortest time. The system developed in this thesis provides tools for the extraction and transcription of the documents. Without these tools, all the process would have been manually done.

8.2 Discussion

There are countless collections of historical documents in archives and libraries that contain plenty of valuable information for historians and researchers. The extraction of this information has become a central task among the Document Analysis researches and practitioners. There is an increasing interest to digital preserve and provide access to these kind of documents. But only the digitalization is not enough for the researchers. The extraction and/or indexation of information of this documents has had an increased interest among researchers. In many cases, and in particular in historical manuscripts, the full transcription of these documents is extremely difficult due the inherent deficiencies: poor physical preservation, different writing styles, obsolete languages, etc.

Word spotting has become a popular an efficient alternative to full transcription. It inherently involves a high level of degradation in the images. The search of words is holistically formulated as a visual search of a given query shape in a larger image, instead of recognising the input text and searching the query word with an ascii string comparison. But the performance of classical word spotting approaches depend on the degradation level of the images being unacceptable in many cases . In this thesis we have proposed a novel paradigm called contextual word spotting method that uses the contextual/semantic information to achieve acceptable results whereas classical word spotting does not reach.

Our word spotting architecture requires pre-segmented words. Thus the first contribution of the work has consisted in a robust line segmentation approach. One of the main contributions is that the method allows to segment lines pixel-wise and not only locate then, as some approaches of the literature, in any kind of handwritten documents. It is not designed for a specific category of documents, coping with both historical and modern ones. And it is able to segment words in an unsupervised framework so it is writer invariant and tackles with the main difficulties in historical documents: multi-skewed, touching and horizontally-overlapped lines.

Concerning the study of word spotting approaches of the state of the art, we conclude that it has the same importance to choose the proper descriptor than to choose the proper interest points (where the local features are computed). Historical handwritten documents present variations in the writer style, despite the documents were written by the same author, and noise that difficult the recognition. The descriptor used for this kind of documents should be robust in front of these difficulties. After the study, we conclude that pseudo-structural descriptors are more robust, and the use of dense interest points store more information of the image. From these premises, we propose a new descriptor to address some configurations that are not included in the study. We have reached similar results to other pseudo-structural descriptors, and in some databases, the developed descriptor achieved better results.

The proposed contextual word spotting framework introduces a double contribution. The first one is an approach to automatically discover the semantic/contextual information of the documents highly structured. The second contribution is a method that, from the contextual information, it extracts the information of the documents. We have showed that both approaches achieve good results. In the first case, the key words computed meet perfectly with the grammar established in these documents. In the second case, the extraction of information is compared to classical word spotting approaches where, in all the cases, we achieve better results. In addition, we have showed two possible applications of the framework developed. The first one makes the alignment of an index in front of the documents. The achieved results outperform classical word spotting approaches due to the sequence order of the words. The second proposed application complements the search of words using the semantic information. The approach allows to define the semantic class where the search has to be done. The results achieve better results than classical word spotting because the search is focussed in specific semantic classes, and the rest of words of other classes are removed previously.

Finally, we have developed an approach that uses MLN to improve the results of classical word spotting approaches. The objective is to combine the output of the word spotting with a grammar previously weighted. The experiments performed have achieved the results of the word spotting approaches. In most of the cases false positives have been removed and the true negatives have been increased. The precision is outperformed in all the experimentation done. But the system has a limitation, when the grammar increases, the number of states and probabilities increase in an exponential way in the MLN.

8.3 Future Work

Despite the advances performed in the word spotting approaches for historical handwritten documents introducing the contextual information, this thesis opens new issues to be further addressed.

Layout segmentation. Taking into account the main contribution of this thesis, the layout analysis could be improved using contextual information. The main difficulty in historical documents is the touching lines. When the ascenders and descenders of two lines are completely overlapped, the segmentation becomes difficult to solve. The segmentation can be improved analysing the information of neighbouring pixels. The contextual information can help us to segment the words accurately.

Segmentation-free. The contextual word spotting framework introduced in this thesis is a segmentation-based approach, and the results of the word representation step is critical in the performance of the whole system. The second open issue is the modification of the framework to be segmentation-free. The objective is to find repetitive patterns analysing the documents and to find the relation between them, as graphical models set relations between objects.

Sequential descriptor. The descriptor developed in this thesis has been used to validate the study done of the word spotting approaches of the literature. The results achieved with this descriptor are similar to other descriptors, or even outperform them in some cases. The objective is to use probabilistic models. For example, Hidden Markov Models (HMM) or Bidirectional Long-Short Term Memory Neuronal Network (BLSTM NN). The descriptor should be adapted for methods that analyse sequential input data.

Other structured documents. Another open issue is to apply the framework introduced in this thesis to other structured documents. But there is a lack of public of databases where the words have semantic and structural relationships. To address this issue, the creation of a new ground truth of other highly structured documents might be the main task. Instead of manually labelling every word, the creation of this ground truth can be started using the framework introduced here. The most populated would be automatically labelled requiring human operations to validate the results.

Framework. Finally, the database used for the experiments should be increased. The database contains documents written by only one writer. The writer style variability is small and multi-writer descriptor have been left out of the scope of this thesis. For this reason, documents of the collection of different authors should be obtained in order to perform further experimental results concerning to multi-writer approaches.

Publications

The following publications are a consequence of the research carried out during the elaboration of this thesis and give an idea of the progression that has been achieved.

Journals

- David Fernández-Mota, Pau Riba, Alicia Fornés, Josep Lladós. A graph-based approach for segmenting touching lines in historical handwritten documents. *International Journal on Document Analysis and Recognition*. 2014.

- Josep Lladós, Marçal Rusiñol, Alicia Fornés, David Fernández-Mota, Anjan Dutta. On the Influence of Word Representations for Handwritten Word Spotting in Historical Documents. *International Journal of Pattern Recognition and Artificial Intelligence*. 2012.

International Conferences and Workshops

- David Fernández-Mota, Pau Riba, Alicia Fornés, Josep Lladós. On the Influence of Key Point Encoding for Handwritten Word Spotting. In *International Conference on Frontiers in Handwriting Recognition*. 2014.

- Pau Riba, Jon Almazán, Alicia Fornés, David Fernández-Mota, Ernest Valveny, Josep Lladós. e-Crowds: a mobile platform for browsing and searching in historical demography-related manuscripts. In *International Conference on Frontiers in Handwriting Recognition*. 2014.

- David Fernández-Mota, Jon Almazán, Núria Cirera, Alicia Fornés & Josep Lladós. *BH2M*: the Barcelona Historical Handwritten Marriages database. In *International Conference on Pattern Recognition*. 2014.

- David Fernández-Mota, R. Manmatha, Josep Lladós & Alicia Fornés. Sequential Word Spotting in Historical Handwritten Documents. In *Workshop on Document Analysis Systems*. 2014.

- David Fernández-Mota, Simone Marinai, Josep Lladós & Alicia Fornés. Contextual Word Spotting in Historical Manuscripts using Markov Logic Networks. In *Workshop on Historical Document Imaging and Processing*. 2013.

- Dimosthenis Karatzas, Faisal Shafait, Seiichi Uchida, Masakazu Iwamura, Lluis Gomez, Sergi Robles, Joan Mas, David Fernández-Mota, Jon Almazán, Lluis Pere de las Heras. ICDAR 2013 Robust Reading Competition. In *International Conference on Document Analysis and Recognition.* 2013.

- Lluis Pere de las Heras, David Fernández-Mota, Alicia Fornés, Ernest Valveny, Gemma Sanchez, Josep Lladós. Perceptual retrieval of architectural floor plans. In *Workshop on Graphics Recognition.* 2013.

- Lluis Pere de las Heras, David Fernández-Mota, Ernest Valveny, Josep Lladós, Gemma Sanchez. Unsupervised wall detector in architectural floor plan. In *International Conference on Document Analysis and Recognition.* 2013.

- David Fernández-Mota, R. Manmatha, Alicia Fornés, Josep Lladós. On Influence of Line Segmentation in Efficient Word Segmentation in Old Manuscripts. In *International Conference on Frontiers in Handwriting Recognition.* 2012.

- Jon Almazán, David Fernández-Mota, Alicia Fornés, Josep Lladós, Ernest Valveny. A Coarse-to-Fine Approach for Handwritten Word Spotting in Large Scale Historical Documents Collection. In *International Conference on Frontiers in Handwriting Recognition.* 2012.

- David Fernández-Mota, Alicia Fornés, Josep Lladós. Handwritten Word Spotting in Old Manuscript Images Using a Pseudo-Structural Descriptor Organized in a Hash Structure. In *Iberian Conference on Pattern Recognition and Image Analysis.* 2011.

Awards

- Best paper award of the 2nd Workshop on Historical Document Imaging and Processing (HIP'2013).

Bibliography

[1] T. Adamek, N. OConnor, and A. Smeaton. Word matching using single closed contours for indexing handwritten historical documents. *International Journal of Document Analysis and Recognition (IJDAR)*, 9(2-4):153–165, 2007.

[2] A. Alaei, P. Nagabhushan, and U. Pal. Piece–wise painting technique for line segmentation of unconstrained handwritten text: a specific study with persian text documents. *Pattern Analysis and Applications*, pages 381–394, 2011.

[3] J. Almazan, A. Fornes, and E. Valveny. Deformable hog-based shape descriptor. In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, pages 1022–1026, Aug 2013.

[4] J. Almazán, A. Gordo, A. Fornés, and E. Valveny. Efficient exemplar word spotting. In *Proceedings of the British Machine Vision Conference*, pages 67.1–67.11. BMVA Press, 2012.

[5] J. Almazan, A. Gordo, A. Fornés, and E. Valveny. Handwritten Word Spotting with Corrected Attributes. In *ICCV 2013 - IEEE International Conference on Computer Vision*, pages 1017–1024. IEEE, Dec. 2013.

[6] J. Almazan, A. Gordo, A. Fornés, and E. Valveny. Word spotting and recognition with embedded attributes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2014.

[7] F. Alvaro, F. Cruz, J.-A. Sánchez, O. Ramos Terrades, and J.-M. Benedí. Page segmentation of structured documents using 2d stochastic context-free grammars. In J. a. Sanches, L. Micó, and J. Cardoso, editors, *Pattern Recognition and Image Analysis*, volume 7887 of *Lecture Notes in Computer Science*, pages 133–140. Springer Berlin Heidelberg, 2013.

[8] A. Amato, A. Sappa, A. Fornés, F. Lumbreras, and J. Lladós. Divide and conquer: Atomizing and parallelizing a task in a mobile crowdsourcing platform. In *CrowdMM*, 2013.

[9] C. R. Anderson, P. Domingos, and D. S. Weld. Relational markov models and their application to adaptive web navigation. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, pages 143–152, 2002.

[10] C. Andrieu, N. de Freitas, A. Doucet, and M. Jordan. An introduction to mcmc for machine learning. *Machine Learning*, 50(1-2):5–43, 2003.

[11] R. Arandjelović and A. Zisserman. Multiple queries for large scale specific object retrieval. In *British Machine Vision Conference*, 2012.

[12] R. Arandjelovic and A. Zisserman. Three things everyone should know to improve object retrieval. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2911–2918, June 2012.

[13] M. Arivazhagan, H. Srinivasan, and S. Srihari. A statistical approach to line segmentation in handwritten documents. In *Document Recognition and Retrieval XIV SPIE*, pages 6500T–1–11, 2007.

[14] S. Athenikos. Wikiphilosofia and pananthropon: Extraction and visualization of facts, relations, and networks for a digital humanities knowledge portal. In *ACM Conference Hypertext and Hypermedia (Hypertext 2009)*, 2009.

[15] M. Baechler, M. Liwicki, and R. Ingold. Text line extraction using dmlp classifiers for historical manuscripts. In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, pages 1029–1033, Aug 2013.

[16] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(4):509–522, Apr. 2002.

[17] L. Bergroth, H. Hakonen, and T. Raita. A survey of longest common subsequence algorithms. In *String Processing and Information Retrieval, 2000. SPIRE 2000. Proceedings. Seventh International Symposium on*, pages 39–48, 2000.

[18] R. Beutler, T. Kaufmann, and B. Pfister. Integrating a non-probabilistic grammar into large vocabulary continuous speech recognition. In *Automatic Speech Recognition and Understanding, 2005 IEEE Workshop on*, pages 104–109, Nov 2005.

[19] A. Bhardwaj, D. Jose, and V. Govindaraju. Script independent word spotting in multilingual documents, 2008.

[20] Y. Boykov and O. Veksler. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1222–1239, 2001.

[21] R. Bozinovic and S. Srihari. Off-line cursive script word recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 11(1):68–83, Jan 1989.

[22] S. Bukhari, F. Shafait, and T. Breuel. Script-independent handwritten textlines segmentation using active contours. In *International Conference on Document Analysis and Recognition*, pages 446–450, 2009.

[23] S. Bukhari, F. Shafait, and T. Breuel. Towards generic text-line extraction. In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, pages 748–752, Aug 2013.

[24] S. S. Bukhari and T. M. Breuel. Layout analysis for arabic historical document images using machine learning. In *International Conference on Frontiers in Handwriting Recognition*, pages 635–640, 2012.

[25] A. Cabré, J.-M. Pujadas-Mora, and M. Valls Fígols. Estimating continuous local and regional historical populations from marriage records. a case study in the barcelona area, 1451-1860. In *European Population Conference 2014*, Jun 2014.

[26] G. Casella and E. I. George. Explaining the Gibbs Sampler. *The American Statistician*, 46(3):167–174, 1992.

[27] J. Chan, C. Ziftci, and D. Forsyth. Searching off-line arabic documents. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 1455–1462, 2006.

[28] F. Chen, L. Wilcox, and D. Bloomberg. Word spotting in scanned images using hidden markov models. In *Acoustics, Speech, and Signal Processing, 1993. ICASSP-93., 1993 IEEE International Conference on*, volume 5, pages 1–4 vol.5, April 1993.

[29] M. J. Choi, A. Torralba, and A. S. Willsky. Context models and out-of-context objects. *Pattern Recognition Letters*, 33(7):853 – 862, 2012. Special Issue on Awards from {ICPR} 2010.

[30] C. Choisy. Dynamic handwritten keyword spotting based on the nshp-hmm. In *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*, volume 1, pages 242–246, Sept 2007.

[31] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *IEEE International Conference on Computer Vision*, Rio de Janeiro, Brazil, October 2007.

[32] N. Cirera, A. Fornés, V. Frinken, and J. Lladós. Hybrid grammar language model for handwritten historical documents recognition. In J. a. Sanches, L. Micó, and J. Cardoso, editors, *Pattern Recognition and Image Analysis*, volume 7887 of *Lecture Notes in Computer Science*, pages 117–124. Springer Berlin Heidelberg, 2013.

[33] E. Cohen, J. Hull, and S. Srihari. Control structure for interpreting handwritten addresses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1049–1055, 1994.

[34] F. Cruz and O. Ramos. Handwritten line detection via an em algorithm. *International Conference on Document Analysis and Recognition*, 2013.

[35] F. Cruz Fernandez and O. Ramos Terrades. Document segmentation using relative location features. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 1562–1565, Nov 2012.

[36] J. Cussens. Loglinear models for first-order probabilistic reasoning. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, UAI'99, pages 126–133, 1999.

[37] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893 vol. 1, June 2005.

[38] B. Das. Extracting collocations from bengali text corpus. *Procedia Technology*, 4(0):325 – 329, 2012. 2nd International Conference on Computer, Communication, Control and Information Technology(C3IT-2012).

[39] B. Das, S. Pal, S. K. Mondal, D. Dalui, and S. K. Shome. Automatic keyword extraction from any text document using n-gram rigid collocation, 2013.

[40] D. Davies and D. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1:224–227, 1979.

[41] J. Davis and P. Domingos. Deep transfer: A markov logic approach. *AI Magazine*, 32(1):51–53, 2011.

[42] S. Deorowicz. Solving longest common subsequence and related problems on graphical processing units. *Software: Practice and Experience*, 40(8):673–700, 2010.

[43] E. Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, pages 269–271, 1959.

[44] P. Domingos and M. Richardson. Markov logic: A unifying framework for statistical relational learning. In *PROCEEDINGS OF THE ICML-2004 WORKSHOP ON STATISTICAL RELATIONAL LEARNING AND ITS CONNECTIONS TO OTHER FIELDS*, pages 49–54, 2004.

[45] R. Dos Santos, G. S. G. Clemente, T. T. I. Ren, G. G. D. Cavalcanti, and R. P. D. Santos. Text line segmentation based on morphology and histogram projection. *International Conference on Document Analysis and Recognition*, pages 651–655, 2009.

[46] R. O. Duda, P. E. Hart, and D. G. Stork. *Nonparametric techniques (chapter 4)*. Pattern Classification. Wiley-Interscience, 2000.

[47] A. Dutta, J. Llados, and U. Pal. Symbol spotting in line drawings through graph paths hashing. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pages 982–986, Sept 2011.

[48] A. Ebrahimi and E. Kabir. A pictorial dictionary for printed Farsi subwords. *Pattern Recognition Letters*, 29:656–663, Apr. 2008.

[49] J. Edwards, Y. W. Teh, R. Bock, M. Maire, G. Vesom, and D. A. Forsyth. Making latin manuscripts searchable using gHMM's. In L. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 385–392. MIT Press, 2005.

[50] A. El-Yacoubi, M. Gilloux, R. Sabourin, and C. Suen. An hmm-based approach for off-line unconstrained handwritten word modeling and recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(8):752–760, Aug 1999.

[51] D. W. Embley, S. Machado, T. Packer, J. Park, A. Zitzelberger, S. W. Liddle, N. Tate, and D. W. Lonsdale. Enabling search for facts and implied facts in historical documents. In *Proceedings of the 2011 Workshop on Historical Document Imaging and Processing*, HIP '11, pages 59–66, 2011.

[52] S. Escalera, A. Forns, O. Pujol, P. Radeva, G. Snchez, and J. Llads. Blurred shape model for binary and grey-level symbol recognition. *Pattern Recognition Letters*, 30(15):1424 – 1433, 2009.

[53] S. España Boquera, M. Castro-Bleda, J. Gorbe-Moya, and F. Zamora-Martinez. Improving offline handwritten text recognition with hybrid hmm/ann models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(4):767–779, April 2011.

[54] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. of 2nd International Conference on Knowledge Discovery and*, pages 226–231, 1996.

[55] A. Fabian, M. Hernandez, L. Pineda, and I. Meza. Contextual semantic processing for a spanish dialogue system using markov logic. In I. Batyrshin and G. Sidorov, editors, *Advances in Artificial Intelligence*, volume 7094 of *Lecture Notes in Computer Science*, pages 258–266. Springer Berlin Heidelberg, 2011.

[56] A. Faizakov, A. Cohen, and T. Vaich. Gaussian subtraction (gs) algorithms for word spotting in continuous speech. In P. Dalsgaard, B. Lindberg, H. Benner, and Z.-H. Tan, editors, *INTERSPEECH*, pages 1793–1796. ISCA, 2001.

[57] M. Feldbach and K. Tonnies. Line detection and segmentation in historical church registers. In *International Conference on Document Analysis and Recognition*, pages 743–747, 2001.

[58] S. Feng and R. Manmatha. A hierarchical, hmm-based automatic evaluation of ocr accuracy for a digital library of books. In *Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries*, JCDL '06, pages 109–118. ACM, 2006.

[59] A. Fischer, A. Keller, V. Frinken, and H. Bunke. Hmm-based word spotting in handwritten documents using subword models. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 3416–3419, Aug 2010.

[60] A. Fischer, A. Keller, V. Frinken, and H. Bunke. Lexicon-free handwritten word spotting using character hmms. *Pattern Recognition Letters*, 33(7):934–942, may 2012.

[61] A. Fornés, V. Frinken, A. Fischer, J. Almazán, G. Jackson, and H. Bunke. A keyword spotting approach using blurred shape model-based descriptors. In *Proceedings of the 2011 Workshop on Historical Document Imaging and Processing*, HIP '11, pages 83–90, 2011.

[62] A. Fornés, J. Lladós, J. Mas, J. M. Pujades, and A. Cabré. A bimodal crowdsourcing platform for demographic historical manuscripts. In *Proceedings of the First International Conference on Digital Access to Textual Cultural Heritage*, DATeCH '14, pages 103–108, 2014.

[63] A. Fornés, X. Otazu, and J. Lladós. Show-through cancellation and image enhancement by multiresolution contrast processing. In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, pages 200–204, Aug 2013.

[64] N. Friedman, L. Getoor, D. Koller, and A. Pfeffer. Learning probabilistic relational models. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'99, pages 1300–1307, 1999.

[65] V. Frinken and H. Bunke. Continuous handwritten script recognition. In D. Doermann and K. Tombre, editors, *Handbook of Document Image Processing and Recognition*, pages 391–425. Springer London, 2014.

[66] V. Frinken, A. Fischer, H. Bunke, and R. Manmatha. Adapting BLSTM neural network based keyword spotting trained on modern data to historical documents. In *Proceedings of the Twelveth International Conference on Frontiers in Handwriting Recognition*, pages 352–357, 2010.

[67] V. Frinken, A. Fischer, R. Manmatha, and H. Bunke. A novel word spotting method based on recurrent neural networks. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(2):211–224, Feb 2012.

[68] B. Fulkerson, A. Vedaldi, and S. Soatto. Localizing objects with smart dictionaries. In *Computer Vision - ECCV*, volume 5302 of *Lecture Notes in Computer Science*, pages 179–192. 2008.

[69] B. Gatos and I. Pratikakis. Segmentation-free word spotting in historical printed documents. In *Document Analysis and Recognition, 2009. ICDAR '09. 10th International Conference on*, pages 271–275, July 2009.

[70] H. A. Glucksman. Classification of mixed-font alphabetics by characteristic loci. Technical report, 1969.

[71] J. Gorbe-Moya, S. Espaa-Boquera, F. Zamora-Martnez, and M. J. C. Bleda. Handwritten text normalization by using local extrema classification. In A. Juan-Cscar and G. Snchez-Albaladejo, editors, *PRIS*, pages 164–172. INSTICC PRESS, 2008.

[72] R. Grishman. Information extraction. *The Handbook of Computational Linguistics and Natural Language Processing*, pages 515–530, 2003.

[73] E. Ha, J. P. Rowe, B. W. Mott, and J. C. Lester. Goal recognition with Markov logic networks for player-adaptive games. In *AIIDE*. The AAAI Press, 2011.

[74] J. Ha, R. M. Haralick, and I. T. Phillips. Document page decomposition by the bounding-box project. In *International Conference on Document Analysis and Recognition*, page 1119, 1995.

[75] M. H. Haggag. Keyword extraction using semantic analysis. *International Journal of Computer Applications*, 61(1):1–6, January 2013.

[76] P. Hart and N. Nilsson. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems, Science, and Cybernetics*, pages 100–107, 1968.

[77] D. S. Hirschberg. Algorithms for the longest common subsequence problem. *J. ACM*, 24(4):664–675, Oct. 1977.

[78] J. Hull. Document image skew detection: Survey and annotated bibliography. *Series in Machine Perception and Artificial Intelligence*, pages 40–66, 1998.

[79] A. Hulth. Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, EMNLP '03, pages 216–223, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

[80] A. Hulth. *Automatic Keyword Extraction: Combining Machine Learning and Natural Language Processing*. VDM Verlag, Saarbr&#252;cken, Germany, Germany, 2008.

[81] J. W. Hunt and T. G. Szymanski. A fast algorithm for computing longest common subsequences. *Commun. ACM*, 20(5):350–353, May 1977.

[82] M. Jindal, R. Sharma, and G. Lehal. Segmentation of horizontally overlapping lines in printed indian scripts. In *International Journal of Computational Intelligence Research*, volume 3, pages 277–286, 2007.

[83] L. Kang and D. Doermann. Template based Segmentation of Touching Components in Handwritten Text Lines. In *International Conference on Document Analysis and Recognition*, pages 569–573, 2011.

[84] L. Kang, J. Kumar, P. Ye, and D. Dermann. Learning text-line segmentation using codebooks and graph partitioning. In *International Conference on Frontiers in Handwriting Recognition*, 2012. 63-68.

[85] L. Kang, J. Kumar, P. Ye, and D. Doermann. Learning Text-line Segmentation using Codebooks and Graph Partitioning. In *International Conference on Frontiers in Handwriting Recognition*, pages 63–68, 2012.

[86] A. A. Kardan, F. Farahmandnia, and A. Omidvar. A novel approach for keyword extraction in learning objectsusing text mining and wordnet. In *2nd World Conference on Information Technology (WCIT-2011)*, volume 1, pages 788–792, 2013.

[87] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of computer vision*, pages 321–331, 1988.

[88] E. Kavallieratou, N. Dromazou, N. Fakotakis, and G. K. Kokkinakis. An integrated system for handwritten document image processing. *International Journal of Pattern Recognition and AI*, 17(4):617–636, 2003.

[89] E. Kavallieratou, N. Fakotakis, and G. K. Kokkinakis. An unconstrained handwriting recognition system. *International Journal on Document Analysis and Recognition*, pages 226–242, 2002.

[90] P. Keaton, H. Greenspan, and R. Goodman. Keyword spotting for cursive document retrieval. In *Proceedings of the Workshop on Document Image Analysis*, pages 74–81, 1997.

[91] D. Kennard and W. Barrett. Separating lines of text in free-form handwritten historical documents. In *Document Image Analysis for Libraries*, pages 12–23, 2006.

[92] D. J. Kennard, A. M. Kent, and W. A. Barrett. Linking the past: Discovering historical social networks from documents and linking to a genealogical database. In *Proceedings of the 2011 Workshop on Historical Document Imaging and Processing*, HIP '11, pages 43–50, 2011.

[93] A. Kesidis, E. Galiotou, B. Gatos, and I. Pratikakis. A word spotting framework for historical machine-printed documents. *International Journal on Document Analysis and Recognition*, 14(2):131–144, 2011.

[94] H. Khosravi and B. Bina. A survey on statistical relational learning. In *Proceedings of the 23rd Canadian Conference on Advances in Artificial Intelligence*, AI'10, pages 256–268, 2010.

[95] K. M. Knill and S. Young. Speaker dependent keyword spotting for accessing stored speech. Technical report, Cambrige University Engineering Department, 1994.

[96] T. Konidaris, B. Gatos, K. Ntzios, I. Pratikakis, S. Theodoridis, and S. Perantonis. Keyword-guided word spotting in historical printed documents using synthetic data and user feedback. *International Journal of Document Analysis and Recognition (IJDAR)*, 9(2-4):167–177, 2007.

[97] H. Koo and N. Cho. Text-line extraction in handwritten chinese documents based on an energy minimization framework. *Trans. Img. Proc.*, pages 1169–1175, 2012.

[98] E. M. Kornfield, R. Manmatha, and J. Allan. Text alignment with handwritten documents. In *Proceedings of the First International Workshop on Document Image Analysis for Libraries (DIAL'04)*, DIAL '04, pages 195–209. IEEE Computer Society, 2004.

[99] P. Krishnan and C. Jawahar. Bringing semantics in word image retrieval. In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, pages 733–737, Aug 2013.

[100] J. B. Kruskal and M. Liberman. The symmetric time-warping problem: from continuous to discrete. In D. Sankoff and J. B. Kruskal, editors, *Time Warps, String Edits, and Macromolecules - The Theory and Practice of Sequence Comparison*, chapter 4. 1983.

[101] J. Kumar, W. Abd-Almageed, L. Kang, and D. Doermann. Handwritten arabic text line segmentation using affinity propagation. In *IAPR International Workshop on Document Analysis Systems*, pages 135–142, 2010.

[102] J. Kumar, L. Kang, D. Doermann, and W. Abd-Almageed. Segmentation of Handwritten Textlines in Presence of Touching Components. *International Conference on Document Analysis and Recognition*, pages 109–113, 2011.

[103] K. S. Kumar and A. Namboodiri. Learning segmentation of documents with complex scripts. *Indian conference on Computer Vision, Graphics and Image Processing*, pages 749–760, 2006.

[104] S.-s. Kuo and O. Agazzi. Keyword spotting in poorly printed documents using pseudo 2-d hidden markov models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 16(8):842–848, Aug 1994.

[105] E. Lawler and D. Wood. Branch-and-bound methods: A survey. In *Operations Research*, volume 14, pages 699–719, 1966.

[106] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 2169–2178, 2006.

[107] Y. Leydier, F. Lebourgeois, and H. Emptoz. Text search for medieval manuscript images. *Pattern Recognition*, 40(12):3552–3567, 2007.

[108] Y. Leydier, A. Ouji, F. LeBourgeois, and H. Emptoz. Towards an omnilingual word retrieval system for ancient manuscripts. *Pattern Recognition*, 42(9):2089–2105, 2009.

[109] S. Z. Li. *Markov Random Field Modeling in Image Analysis*. Springer Publishing Company, Incorporated, 3rd edition, 2009.

[110] Y. Li, Y. Zheng, D. Doermann, and S. Jaeger. Script-independent text line segmentation in freestyle handwritten documents. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1313–1329, 2008.

[111] L. Likforman-Sulem, A. Zahour, and B. Taconet. Text line segmentation of historical documents: a survey. *International Journal of Document Analysis and Recognition*, pages 123–138, 2006.

[112] C.-L. Liu, F. Yin, D.-H. Wang, and Q.-F. Wang. Casia online and offline chinese handwriting databases. In *Proceedings of the 2011 International Conference on Document Analysis and Recognition*, ICDAR '11, pages 37–41, 2011.

[113] W. Liu and H. Weisheng. Improved viterbi algorithm in continuous speech recognition. In *Computer Application and System Modeling (ICCASM), 2010 International Conference on*, volume 7, pages V7–207–V7–209, Oct 2010.

[114] M. Liwicki, E. Indermuhle, and H. Bunke. On-line handwritten text line detection using dynamic programming. *International Conference on Document Analysis and Recognition*, pages 447–451, 2007.

[115] G. Louloudis, B. Gatos, I. Pratikakis, and C. Halatsis. Text line detection in handwritten documents. *Pattern Recognition*, pages 3758–3772, 2008.

[116] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, Nov. 2004.

[117] Y. Lu and C. Tan. Word spotting in chinese document images without layout analysis. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, volume 3, pages 57–60 vol.3, 2002.

[118] F. Luthy, T. Varga, and H. Bunke. Using hidden markov models as a tool for handwritten text line segmentation. In *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*, volume 1, pages 8–8, Sept 2007.

[119] U. Mahadevan and R. Nagabushnam. Gap metrics for word separation in handwritten lines. In *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on*, volume 1, pages 124–127 vol.1, Aug 1995.

[120] D. Maier. The complexity of some problems on subsequences and supersequences. *J. ACM*, 25(2):322–336, apr 1978.

[121] R. Manmatha and W. B. Croft. Word spotting: Indexing handwritten archives, 1997.

[122] R. Manmatha, C. Han, and E. M. Riseman. Word spotting: a new approach to indexing handwriting. In *Computer Vision and Pattern Recognition, 1996. Proceedings CVPR '96, 1996 IEEE Computer Society Conference on*, pages 631–637, Jun 1996.

[123] R. Manmatha and J. L. Rothfeder. A scale space approach for automatically segmenting words from historical handwritten documents. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(8):1212–1225, 2005.

[124] R. Manmatha and N. Srimal. Scale space technique for word segmentation in handwritten documents. In *Scale-Space Theories in Computer Vision*, volume 1682, pages 22–33. 1999.

[125] V. Manohar, S. Vitaladevuni, H. Cao, R. Prasad, and P. Natarajan. Graph clustering-based ensemble method for handwritten text line segmentation. In *International Conference on Document Analysis and Recognition*, pages 574–578, 2011.

[126] S. Marinai. Text retrieval from early printed books. *International Journal on Document Analysis and Recognition (IJDAR)*, 14(2):117–129, 2011.

[127] S. Marinai. Text retrieval from early printed books. *International Journal on Document Analysis and Recognition (IJDAR)*, 14(2):117–129, 2011.

[128] S. Marinai, E. Marino, and G. Soda. Indexing and retrieval of words in old documents. In *Proceedings of the Seventh International Conference on Document Analysis and Recognition*, pages 223–227, 2003.

[129] U. Marti and H. Bunke. Using a statistical language model to improve the performance of an HMM-based cursive handwriting recognition system. *International Journal on Pattern Recognition and Artificial Intelligence*, 15(1):65–90, 2001.

[130] U.-V. Marti and H. Bunke. A full english sentence database for off-line handwriting recognition. In *Document Analysis and Recognition, 1999. ICDAR '99. Proceedings of the Fifth International Conference on*, pages 705–708, Sep 1999.

[131] U.-V. Marti and H. Bunke. The iam-database: an english sentence database for offline handwriting recognition. *IJDAR*, 5(1):39–46, 2002.

[132] O. Medelyan and I. H. Witten. Thesaurus based automatic keyphrase indexing. In *Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital Libraries*, JCDL '06, pages 296–297, 2006.

[133] R. Moghaddam and M. Cheriet. Application of multi-level classifiers and clustering for automatic word spotting in historical document images. In *Proceedings of the Tenth International Conference on Document Analysis and Recognition*, pages 511–515, 2009.

[134] C. Myers, L. Rabiner, and A. Rosenberg. On the use of dynamic time warping for word spotting and connected word recognition. *Bell System Technical Journal, The*, 60(3):303–325, March 1981.

[135] S. N Srihari, H. Srinivasan, C. Huang, and S. Shetty. Spotting words in latin, devanagari and arabic scripts. In *Vivek: Indian Journal of Artifical Intelligence*, volume 16, pages 2–9, 2006.

[136] G. Nagy. Twenty years of document image analysis in pami. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(1):38–62, Jan 2000.

[137] G. Navarro and R. Baeza-yates. Very fast and simple approximate string matching. In *Information Processing Letters*, pages 65–70, 1999.

[138] H. Ney and S. Ortmanns. Dynamic programming search for continuous speech recognition. *Signal Processing Magazine, IEEE*, 16(5):64–83, Sep 1999.

[139] A. Nicolaou and B. Gatos. Handwritten text line segmentation by shredding text into its lines. In *International Conference on Document Analysis and Recognition*, pages 626–630, 2009.

[140] M. R. nol and J. Lladós. Boosting the handwritten word spotting experience by including the user in the loop. *Pattern Recognition*, 47(3):1063 – 1072, 2014.

[141] L. O'Gorman. The document spectrum for page layout analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1162–1173, 1993.

[142] N. Otsu. A Threshold Selection Method from Gray-level Histograms. *IEEE Transactions on Systems, Man and Cybernetics*, pages 62–66, 1979.

[143] N. Ouwayed and A. Belaid. A general approach for multi-oriented text line extraction of handwritten document. *International Journal on Document Analysis and Recognition*, 2011.

[144] T. Packer and D. Embley. Cost effective ontology population with data from lists in ocred historical documents. In *HIP 2013*, 2013.

[145] V. Papavassiliou, T. Stafylakis, V. Katsouros, and G. Carayannis. Handwritten document image segmentation into text lines and words. *Pattern Recognition*, pages 369–377, 2010.

[146] M. Pechwitz, S. S. Maddouri, V. Mrgner, N. Ellouze, and H. Amiri. Ifn/enit - database of handwritten arabic words. In *In Proc. of CIFED 2002*, pages 129–136, 2002.

[147] F. Perronnin and J. A. Rodriguez-Serrano. Fisher kernels for handwritten wordspotting. In *Proceedings of the 2009 10th International Conference on Document Analysis and Recognition*, ICDAR '09, pages 106–110, 2009.

[148] A. Popescul and L. H. Ungar. Structural logistic regression for link analysis. In *Proceedings of the Second International Workshop on Multi-Relational Data Mining*, pages 92–106, 2003.

[149] J.-M. Pujadas-Mora, G. Brea, and A. Cabré. Intergenerational transmission of social status and occupations at the barcelona area, 16th 17th centuries. In *European Population Conference 2014*, Jun 2014.

[150] T. Rath and R. Manmatha. Word image matching using dynamic time warping. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages II–521 – II–527 vol.2, 2003.

[151] T. M. Rath and R. Manmatha. Word image matching using dynamic time warping. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages 521–527, June 2003.

[152] T. M. Rath and R. Manmatha. Word spotting for historical documents. *International Journal on Document Analysis and Recognition*, pages 139–152, 2007.

[153] T. M. Rath and R. Manmatha. Word spotting for historical documents. *International Journal On Document Analysis and Recognition*, pages 139–152, 2007.

[154] T. M. Rath, R. Manmatha, and V. Lavrenko. A search engine for historical manuscript images. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '04, pages 369–376. ACM, 2004.

[155] M. Richardson and P. Domingos. Markov logic networks. *Machine Learning*, 62(1-2):107–136, feb 2006.

[156] J. A. Rodriguez and F. Perronnin. Local gradient histogram features for word spotting in unconstrained handwritten documents. In *Proceedings of the 1st International Conference on Handwriting Recognition (ICFHR'08)*, Aug. 2008.

[157] J. Rodriguez-Serrano and F. Perronnin. Handwritten word-spotting using hidden Markov models and universal vocabularies. *Pattern Recognition*, 42(9):2106–2116, 2009.

[158] J. A. Rodríguez-Serrano and F. Perronnin. Handwritten word-spotting using hidden markov models and universal vocabularies. *Pattern Recognition*, pages 2106–2116, 2009.

[159] J. A. Rodriguez-Serrano and F. Perronnin. A model-based sequence similarity with application to handwritten word spotting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2108–2120, 2012.

[160] S. Rohini, R. Uma Devi, and S. Mohanavel. Segmentation of touching, overlapping, skewed and short handwritten text lines. *International Journal of Computer Applications*, pages 24–27, 2012.

[161] V. Romero, A. FornéS, N. Serrano, J. A. SáNchez, A. H. Toselli, V. Frinken, E. Vidal, and J. LladóS. The esposalles database: An ancient marriage license corpus for off-line handwriting recognition. *Pattern Recogn.*, 46(6):1658–1669, June 2013.

[162] R. Rose and D. Paul. A hidden markov model based keyword recognition system. In *Acoustics, Speech, and Signal Processing, 1990. ICASSP-90., 1990 International Conference on*, pages 129–132 vol.1, Apr 1990.

[163] A. Rosenberg and J. Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning(EMNLP-CoNLL)*, pages 410–420, 2007.

[164] J. Rosin and G. West. Segmentation of edges into lines and arcs. *Image and Vision Computing*, 7(2):109–114, 1989.

[165] L. Rothacker, M. Rusiñol, and G. Fink. Bag-of-features hmms for segmentation-free word spotting in handwritten documents. In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, pages 1305–1309, Aug 2013.

[166] J. Rothfeder, S. Feng, and T. Rath. Using corner feature correspondences to rank word images by similarity. In *Proceedings of the Computer Vision and Pattern Recognition Workshop*, pages 30–30, 2003.

[167] P. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, Nov. 1987.

[168] P. Roy, U. Pal, and J. Lladós. Morphology based handwritten line segmentation using foreground and background information. In *International Conference on Frontiers in Handwriting Recognition*, pages 241–246, 2008.

[169] M. Rusiñol, D. Aldavert, R. Toledo, and J. Llados. Browsing heterogeneous document collections by a segmentation-free word spotting method. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pages 63–67, Sept 2011.

[170] M. Rusiñol and J. Llados. The role of the users in handwritten word spotting applications: query fusion and relevance feedback. In *Frontiers in Handwriting Recognition (ICFHR), 2012 International Conference on*, pages 55–60, Sept 2012.

[171] R. Saabni and J. El-Sana. Keyword searching for arabic handwritten documents. *The 11th International Conference on Frontiers in Handwriting recognition (ICFHR 2008)*, pages 271–277, 2008.

[172] R. Saabni and J. El-Sana. Language-independent text lines extraction using seam carving. In *International Conference on Document Analysis and Recognition*, pages 563–568, 2011.

[173] H. Sakoe and S. Chiba. A dynamic programming approach to continuous speech recognition. In *Proceedings of the Seventh International Congress on Acoustics, Budapest*, volume 3, pages 65–69, 1971.

[174] A. Sarkar, A. Biswas, P. Bhowmick, and B. Bhattacharya. Word segmentation and baseline detection in handwritten documents using isothetic covers. In *Frontiers in Handwriting Recognition (ICFHR), 2010 International Conference on*, pages 445–450, Nov 2010.

[175] R. Sarkar, S. Moulik, N. Das, S. Basu, M. Nasipuri, and M. Kundu. Suppression of non-text components in handwritten document images. In *International Conference on Image Information Processing*, pages 1–7, 2011.

[176] T. Scheffer and S. Wrobel. Text classification beyond the bag-of-words representation.

[177] G. Seni and E. Cohen. External word segmentation of off-line handwritten text lines. *Pattern Recognition*, 27(1):41 – 52, 1994.

[178] A. W. Senior and A. J. Robinson. An off-line cursive handwriting recognition system. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):309–321, 1998.

[179] N. Serrano, F. Castro, and A. Juan. The rodrigo database. In N. C. C. Chair), K. Choukri, B. Maegaard, J. Mariani, J. Odijk, S. Piperidis, M. Rosner, and D. Tapias, editors, *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*. European Language Resources Association (ELRA), may 2010.

[180] K. Sesh Kumar, A. Namboodiri, and C. Jawahar. Learning segmentation of documents with complex scripts. In P. Kalra and S. Peleg, editors, *Computer Vision, Graphics and Image Processing*, volume 4338 of *Lecture Notes in Computer Science*, pages 749–760. Springer Berlin Heidelberg, 2006.

[181] Z. Shi, S. Setlur, and V. Govindaraju. Text extraction from gray scale historical document images using adaptive local connectivity map. In *Document Analysis and Recognition, 2005. Proceedings. Eighth International Conference on*, pages 794–798 Vol. 2, Aug 2005.

[182] Z. Shi, S. Setlur, and V. Govindaraju. A steerable directional local profile technique for extraction of handwritten arabic text lines. In *International Conference on Document Analysis and Recognition*, pages 176–180, 2009.

[183] R. Shinghal and G. T. Toussaint. Experiments in text recognition with the modified viterbi algorithm. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-1(2):184–193, April 1979.

[184] V. A. Silva and F. G. Cozman. Markov logic networks for supervised, unsupervised and semisupervised learning of classifiers. In *IV Workshop on MSc Dissertation and PhD Thesis in Artificial Intelligence (WTDIA)*, 2008.

[185] P. Simard, D. Steinkraus, and M. Agrawala. Ink normalization and beautification. In *Document Analysis and Recognition, 2005. Proceedings. Eighth International Conference on*, pages 1182–1187 Vol. 2, Aug 2005.

[186] P. Singla and P. Domingos. Entity resolution with markov logic. In *Proceedings of the Sixth International Conference on Data Mining*, ICDM '06, pages 572–582, 2006.

[187] S. Srihari and V. Govindaraju. Analysis of textual images using the hough transform. *Machine Vision and Applications*, 2(3):141–153, 1989.

[188] S. Srihari, H. Srinivasan, P. Babu, and C. Bhole. Handwritten arabic word spotting using the cedarabic document analysis system. In *Proc. Symposium on Document Image Understanding Technology (SDIUT-05), College Park, MD*, pages 123–132, 2005.

[189] T. Stafylakis, V. Papavassiliou, V. Katsouros, and G. Carayannis. Robust text-line and word segmentation for handwritten documents images. In *International Conference on Acoustics, Speech and Signal Processing*, pages 3393–3396, 2008.

[190] N. Stamatopoulos, B. Gatos, G. Louloudis, U. Pal, and A. Alaei. Icdar 2013 handwriting segmentation contest. *International Conference on Document Analysis and Recognition*, pages 1402–1406, 2013.

[191] T. Syeda-Mahmood. Indexing of handwritten document images. In *Document Image Analysis, 1997. (DIA '97) Proceedings., Workshop on*, pages 66–73, Jun 1997.

[192] K. Takru and G. Leedham. Separation of touching and overlapping words in adjacent lines of handwritten text. In *Frontiers in Handwriting Recognition, 2002. Proceedings. Eighth International Workshop on*, pages 496–501, 2002.

[193] K. Terasawa and Y. Tanaka. Locality sensitive pseudo-code for document images. In *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*, volume 1, pages 73–77, Sept 2007.

[194] K. Terasawa and Y. Tanaka. Slit style hog feature for document image word spotting. In *Document Analysis and Recognition, 2009. ICDAR '09. 10th International Conference on*, pages 116–120, July 2009.

[195] C. I. Tomai, B. Zhang, and V. Govindaraju. Transcript mapping for historic handwritten document images. In *IWFHR*, pages 413–418. IEEE Computer Society, 2002.

[196] A. H. Toselli, A. Juan, J. Gonzlez, I. Salvador, E. Vidal, F. Casacuberta, D. Keysers, and H. Ney. Integrated handwriting recognition and interpretation using finite-state models. *IJPRAI*, 18(4):519–539, 2004.

[197] T. van Der Zant, L. Schomaker, and K. Haak. Handwritten-word spotting using biologically inspired features. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(11):1945–1957, Nov 2008.

[198] F. Villavicencio, J.-M. Pujadas-Mora, F. Colchero, and A. Cabré. Adult mortality in catalonia in the 16th and 17th centuries. In *European Population Conference 2014*, Jun 2014.

[199] A. Vinciarelli, S. Bengio, and H. Bunke. Offline recognition of unconstrained handwritten texts using hmms and statistical language models. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 26:709–720, 2004.

[200] F. Wahlberg and A. Brun. Graph based line segmentation on cluttered handwritten manuscripts. *International Conference on Pattern Recognition*, pages 1570–1573, 2012.

[201] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3360–3367, June 2010.

[202] P. Wang, V. Eglin, C. Garcia, C. Largeron, and A. McKenna. A comprehensive representation model for handwriting dedicated to word spotting. In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, pages 450–454, Aug 2013.

[203] R. Xu and I. Wunsch, D. Survey of clustering algorithms. *Neural Networks, IEEE Transactions on*, 16(3):645–678, May 2005.

[204] I. Yalniz and R. Manmatha. Finding translations in scanned book collections categories and subject descriptors. In *SIGIR*, pages 465–474, 2012.

[205] I. Z. Yalniz, E. F. Can, and R. Manmatha. Partial duplicate detection for large book collections. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, CIKM '11, pages 469–474. ACM, 2011.

[206] I. Z. Yalniz and R. Manmatha. A fast alignment scheme for automatic ocr evaluation of books. In *Proceedings of the 2011 International Conference on Document Analysis and Recognition*, ICDAR '11, pages 754–758. IEEE Computer Society, 2011.

[207] B. Yanikoglu and P. A. Sandon. Segmentation of off-line cursive handwriting using linear programming. *Pattern Recognition*, 31(12):1825 – 1833, 1998.

[208] F. Yin. Handwritten text line extraction based on minimum spanning tree clustering. *International Conference on Wavelet Analysis and Pattern Recognition*, pages 1123–1128, 2007.

[209] F. Yin and C. Liu. Handwritten text line segmentation by clustering with distance metric learning. *International Conference on Frontiers in Handwriting Recognition*, pages 229–234, 2008.

[210] F. Yin and C. Liu. Handwritten chinese text line segmentation by clustering with distance metric learning. *Pattern Recognition*, pages 3146–3157, 2009.

[211] K. Zagoris, I. Pratikakis, A. Antonacopoulos, B. Gatos, and N. Papamarkos. Handwritten and machine printed text separation in document images using the bag of visual words paradigm. In *International Conference on Frontiers in Handwriting Recognition*, pages 103–108, 2012.

[212] A. Zahour, L. Likforman-Sulem, W. Boussalaa, and B. Taconet. Text line segmentation of historical arabic documents. In *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*, volume 1, pages 138–142, Sept 2007.

[213] C. Zhang. Automatic keyword extraction from documents using conditional random fields. In *Journal of Computational Information Systems*, volume 4, pages 1169–1180, 2008.

[214] K. Zhang, H. Xu, J. Tang, and J. Li. Keyword extraction using support vector machine. In *Proceedings of the 7th International Conference on Advances in Web-Age Information Management*, WAIM '06, pages 85–96, 2006.

[215] D. Zhao and D. G. Daut. Morphological hit-or-miss transformation for shape recognition. *Journal of Visual Communication and Image Representation*, 2(3):230 – 243, 1991.

[216] M. Zimmermann, J.-C. Chappelier, and H. Bunke. Offline grammar-based recognition of handwritten sentences. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(5):818–821, May 2006.

Final Acknowledgment

The authors thank the *CED-UAB* and the Cathedral of Barcelona for providing the images. This work has been partially supported by the Spanish projects TIN2008-04998, TIN2009-14633-C03-03, TIN2011-24631, TIN2012-37475-C02-02 and CSD2007-00018, by the EU project ERC-2010-AdG-20100407-269796 and by a research grant of the UAB (471-01-8/09)