Universitat Autònoma de Barcelona

Probabilistic Graphical Models for Document Analysis

Francisco Cruz Fernández

Probabilistic Graphical Models for Document Analysis

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

Bellaterra, September 21, 2016

Director:	**Dr. Oriol Ramos Terrades** Departament de Ciències de la Computació Universitat Autònoma de Barcelona
Thesis Committee	**Dra. Laurence Likforman-Sulem** Département Traitement du Signal et des Images Telecom ParisTech **Dr. Simone Marinai** Dipartimento di Ingegneria dell'Informazione Universit degli Studi di Firenze **Dr. Josep Lladós Canet** Departament de Ciències de la Computació Universitat Autònoma de Barcelona **Dr. Jerod J. Weinman** Department of Computer Science Grinnell College **Dr. Ernest Valveny Llobet** Departament de Ciències de la Computació Universitat Autònoma de Barcelona

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, Departament de cinències de la computaciò, Universitat Autònoma de Barcelona.

ISBN XXXX

Printed by Ediciones Gráficas Rey, S.L.

A Chloe, a mis padres, y a mi hermana...

The best way to predict the future is to invent it.
- Alan Kay

Agradecimientos

El periodo de tiempo transcurrido desde que comencé el desarrollo de esta tesis, hasta el día de hoy cuando escribo estas últimas páginas, ha sido el más más intenso y emocionante de mi vida. No solo en lo académico y profesional, sino en muchos otros aspectos personales que me han marcado y que siempre formarán parte de mi. Voy a intentar agradecer a todas las personas que han formado parte de mi vida estos años y me han apoyado para poder llegar hasta aquí. Me dejaré a alguien por el camino seguro, pero no me lo tengáis en cuenta, son muchas páginas escritas, y, aunque no os nombre, vosotros sabéis lo importante que habéis sido para mi.

En primer lugar, me gustaría agradecer al Centro de Visión por Computador (CVC) y a la Universidad Autónoma de Barcelona por su apoyo al concederme la beca FPI74-CVC que me ha permitido desarrollar esta tesis. También agradecer a las empresas ITESOFT-Yooz por la financiación de este trabajo, y por proporcionarme la experiencia empresarial y de colaboración entre grupos de investigación durante el proyecto DOD. También agradecer a los miembros del tribunal por aceptar la invitación a formar parte de esta tesis, y al grupo PRHLT por su valiosa colaboración en una de las partes presentadas.

En segundo lugar, quería agradecer a todo personal del CVC por su apoyo todos estos años y por estar siempre ahí cuando os he podido necesitar. No se donde estaré dentro de unos años, pero seguro que costará encontrar una empresa con una familia que valga tanto la pena. Un agradecimiento especial a todos los miembros del DAG por haberme introducido en el mundo de la investigación y haberme proporcionado todos los conocimientos que me han traido al día de hoy, siempre me he sentido muy bien en este grupo y esa sensación dice mucho del tipo de personas que lo forman. Muchas gracias al Dr. Marçal Rossinyol, Dra. Alicia Fornés, Dr. Dimosthenis Karatzas, Dr. Lluís Gómez, Dr. Hongxing Gao, Nuria Cirera, Albert Berenguel, Angelos Nikolaou, Suman Kumar, Sounak Dey y a las nuevas generaciones que se unieron hace poco. Gracias al Dr. Ernest Valveny por recibirme en mis primeros días en el CVC, y al Dr. Josep Lladós por su apoyo y gran dedicación al grupo.

En tercer lugar, agradecer a mi director de tesis Dr. Oriol Ramos Terrades. No se como puedo agradecer tu gran dedicación, apoyo y compromiso todos estos años. Se que has tenido que tener mucha paciencia conmigo, sobretodo en cuanto a tus interminables hojas de fórmulas, que seguro echaré de menos, y a tus tan valiosas explicaciones. Muchas gracias por todos los conocimientos que me has transmitido, las manías que seguro que me has pegado, y por haberme hecho llegar hasta aquí. Espero haber dejado también un poco de huella en ti, y estoy seguro que todo te irá

genial en el futuro. Gracias, de verdad.

En lo personal, estos años han supuesto muchos cambios en mi vida. Llegar a una ciudad nueva donde no conoces a nadie puede ser duro, pero afortunadamente ese no fue mi caso. Todo se lo debo a las grandes personas que he tenido la oportunidad de conocer y a los que, aunque tenga lejos, siempre consideraré mis amigos. Gracias por vuestras risas, por vuestra compañía y apoyo sin el cual no podría haber conseguido llegar hasta aquí. Gracias a Jon, Lluís, Carles, Toni, Rubén, David, Anjan, Alejandro, Camp, Ivet, Roger, Joan, Sandra, Elena, Gisela, Yainuvis, Marina, Hugo, y todos los que habéis formado parte de esta etapa. También a la gente de Ventilador Music, tan currantes como nadie, estoy seguro de que os irá genial. A la gente de Valencia, Lu, Mikel, Samuel, Vicente, gente de La Puebla y Uppsaleños, gracias por aguantarme y estar siempre ahí aunque nos veamos poco.

Agradecer por supuesto a mi familia, a mis padres Paco y Manoli, y a mi hermana Laura, por haber estado siempre a mi lado y darlo todo para que lograra lo que me propusiera. Gracias por vuestra inspiración y cariño, si he llegado hasta aquí es en gran parte gracias a vosotros. Gracias también a Pascale por haberme dado lo mejor que me ha pasado.

Y por último, a la persona más importante de mi vida. Chloe, me faltaría espacio para describir lo que ha significado para mi tenerte a mi lado estos años. Gracias por tu apoyo incondicional, por tu confianza y por tu sacrificio, sobretodo en este último año, sin ello no podría haberlo logrado. Gracias por contagiarme cada día de tu alegría, por ser como eres, por tu esfuerzo, por tu paciencia, por hacerme ver siempre el lado bueno de las cosas, por enfadarte conmigo, y por hacerme mejorar cada día. Espero que nuestro futuro sea al menos tan bueno como han sido estos últimos 7 años y que pueda devolverte todo lo que me has dado. Te quiero. Esta tesis es para ti.

Abstract

Currently, more than 80 % of the documents stored on paper belong to the business field. Advances in digitization techniques have fostered the interest in creating digital copies in order to solve maintenance and storage problems, as well as to have efficient ways for transmission and automatic extraction of the information contained therein. This situation has led to the need to create systems that can automatically extract and analyze this kind of information.

The great variety of types of documents makes this not a trivial task. The extraction process of numerical data from tables or invoices differs substantially from a task of handwriting recognition in a document with annotations. However, there is a common link in the two tasks: Given a document, we need to identify the region where the information of interest is located. In the area of Document Analysis this process is called Layout Analysis, and aims at identifying and categorizing the different entities that compose the document. These entities can be text regions, pictures, text lines or tables, among others. This process can be done from two different approaches: physical or logical analysis. Physical analysis focus on identifying the physical boundaries that define the area of interest, whereas logical analysis also models information about the role and semantics of the entities within the scope of the document. To encode this information it is necessary to incorporate prior knowledge about the task into the analysis process, which can be introduced in terms of contextual relations between entities. The use of context has proven to be useful to reinforce the recognition process and improve the results on many computer vision tasks. It presents two fundamental questions: what kind of contextual information is appropriate, and how to incorporate this information into the model.

In this thesis we study several ways to incorporate contextual information on the task of document layout analysis. We focus on the study of Probabilistic Graphical Models and other mechanisms for the inclusion of contextual relations applied to the specific tasks of region identification and handwritten text line segmentation. On the one hand, we present several methods for region identification. First, we present a method for layout analysis based on Conditional Random Fields for maximum a posteriori estimation. We encode a set of structural relations between different classes of regions on a set of features. Second, we present a method based on 2D-Probabilistic Context-free Grammars and perform a comparative study between probabilistic graphical models and this syntactic approach. Third, we propose a statistical approach based on the Expectation-Maximization algorithm devised to structured documents. We perform a thorough evaluation of the proposed methods

on two particular collections of documents: a historical dataset composed of ancient structured documents, and a collection of contemporary documents. On the other hand, we present a probabilistic framework applied to the task of handwritten text line segmentation. We successfully combine the EM algorithm and variational approaches for this purpose. We demonstrate that the use of contextual information using probabilistic graphical models is of great utility for these tasks.

Resumen

Actualmente, más del 80% de los documentos almacenados en papel pertenecen al ámbito empresarial. Avances en materia de digitalización de documentos han fomentado el interés en crear copias digitales para solucionar problemas de mantenimiento y almacenamiento, además de poder disponer de formas eficientes de transmisión y extracción automática de la información contenida en ellos. Esta situación ha propiciado la necesidad de crear sistemas capaces de extraer y analizar automáticamente esta información.

La gran variedad en tipos de documentos hace que esta no sea una tarea trivial. Un proceso de extracción de datos numéricos de tablas o facturas difiere sustancialmente del reconocimiento de texto manuscrito en un documento con anotaciones. No obstante, hay un nexo común en las dos tareas: dado un documento, es necesario localizar la región donde está la información de interés. En el área del Análisis de Documentos, a este proceso se denomina Análisis de la estructura del documento, y tiene como objetivo la identificación y categorización de las diferentes entidades que lo componen. Estas entidades pueden ser regiones de texto, imágenes, lineas de texto, celdas de una tabla, campos de un formulario, etc. Este proceso se puede realizar desde dos enfoques diferentes: análisis físico, o análisis lógico. El análisis físico consiste en identificar la ubicación y los límites que definen el área donde se encuentra la región de interés. El análisis lógico incluye además información acerca de su función y significado dentro del ámbito del documento. Para poder modelar esta información, es necesario incorporar al proceso de análisis un conocimiento previo sobre la tarea. Este conocimiento previo se puede modelar haciendo uso de relaciones contextuales entre las diferentes entidades. El uso del contexto en tareas de visión por computador ha demostrado ser de gran utilidad para guiar el proceso de reconocimiento y reforzar los resultados. Este proceso implica dos cuestiones fundamentales: qué tipo de información contextual es la adecuada para cada problema, y cómo incorporamos esa información al modelo.

En esta tesis abordamos el análisis de la estructura de documentos basándonos en la incorporación de información contextual en el proceso de análisis. Hacemos énfasis en el uso de modelos gráficos probabilísticos y otros mecanismos para proponer soluciones al problema de la identificacion de regiones y la segmentación de lineas de texto manuscritas. Presentamos varios métodos que hacen uso de modelos gráficos probabilisticos para resolver las anteriores tareas, y varios tipos de información contextual. En primer lugar presentamos un conjunto de características que pueden modelar información contextual sobre la posición relativa entre las diferentes regiones. Uti-

lizamos estas características junto a otras para en varios modelos basados en modelos gráficos probabilísticos, y los comparamos con un modelo sintáctico clásico basado en gramáticas libres de contexto. En segundo lugar presentamos un marco probabilístico aplicado a la segmentación de lineas de lineas de texto. Combinamos el proceso de inferencia en el modelo con la estimación de las líneas de texto. Demostramos como el uso de información contextual mediante modelos gráficos probabilísticos es de gran utilidad para estas tareas.

Contents

List of Tables

List of Figures

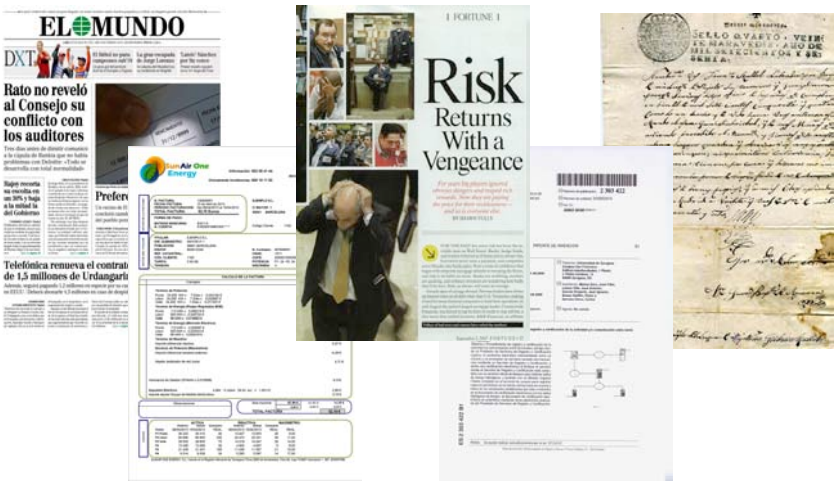
Chapter 1

Introduction

Throughout history, different civilizations have captured their knowledge and ideas in documents initially reserved to a small portion of the population. The invention of the printing press, as we know it today, led this information to a wide sector of the society, and foster the distribution of a large amount of documents from multiple knowledge areas. However, it was in the last century when the amount of documents produced increased considerably. The development of new information technologies, and the widespread use of personal computers, instead of reducing the amount of paper documents, it had the opposite effect. Nowadays is easier than ever to capture information in paper, and, despite the increasing use of personal devices such as tables or laptops, paper documents still have an important role in our lives.

The great advances in the last decades regarding digitization techniques and camera capture systems, have fostered the interest in digitally preserve large amounts of information that, up to that moment, was registered in paper documents. This phenomenon have been mainly reflected in the scope of the business activity. Many countries have declared regulations to force companies to have digitized versions of documents with financial or sensitive information. Besides, with the objective of a better information management, companies have also digitized their old paper databases and important documentation. Nowadays, the large majority of digitized documents belongs to this ambit [5]. Conservation of ancient and historical documents has also an important role in the increment of digitized documents. Multiple collections of invaluable relevance are being digitized and stored in digital libraries in order to preserve its contents against aging and possible loss of data. Besides, most of these collections can be object for many types of research studies [6]. Digitized documents offer an easy maintenance, offers loss-less storage, and efficient ways for transmission and perform information retrieval processes.

This situation has opened a new market niche to develop systems able to automatically extract and analyze the information contained in collections of documents. One of the most in-demand applications is the automatic extraction of information from forms or invoices, although there are many more such as automatic check processing,



(a)

Figure 1.1: Documents contain multiple types of information in multiple ways. The digitization process makes possible to automatically extract and process this information.

signature verification, document classification, or information retrieval, among others. In this dissertation we focus on the area of computer vision that tackles these tasks. This research field is referred to as Document Analysis and Recognition (DAR) [1], and aims at investigating and developing new and better ways in what regards to extract and process the information contained in collections of documents. The research on DAR includes many other research areas, such as image processing, pattern recognition, machine learning, and, more recently, big data technologies for storage and scalability purposes. Furthermore, it comprises many different, but related, research tasks, such as layout analysis, text line segmentation, text recognition, document classification, graphics recognition, etc.

In the next sections we describe the motivation that led to the realization of this thesis, we detail the main challenges of the addressed tasks and enumerate the set of contributions derived from the presented work. Last, we define the structure of this thesis.

1.1 Motivation and challenges

The automatic extraction of information from document images can be a process kind of ambiguous. This ambiguity is the result of the large diversity of types of documents that can be object of analysis (see Figure 1.1). Information can take different forms

[1] The term DAR is also referred in the literature as Document Image Analysis (DIA) or Document Image Analysis and Recognition (DIAR) depending on the author preferences.

or come from different sources, for instance, we could be interested in the extraction of some numerical values from tables or forms, or in the explicit recognition of the text. In any case, in order to extract the information of interest, is always necessary to identify the area of the document where it is located, and extract it in a format that facilitates posterior analysis processes.

The task responsible for this step is referred to as document layout analysis. Formally, layout analysis aims at identifying and categorizing the set of entities that compose the document. The set of entities depends on the task in question. An usual approach is to identify the set of blocks of text, figures, or tables. However, it is also common to perform a fine-grained analysis to detect specific parts or the document. A particular case of this analysis is text line detection. This task consist of two main processes that eventually may converge: the localization of the text line and its segmentation by defining a pixel-wise labeling of the text components. The detection of these entities is one of the main tasks within the document analysis process. Its relevance relies in the fact that many other tasks depend on the results obtained. On the one hand, the accuracy of important tasks such as signature verification or symbol recognition depends on the correct identification of regions that include the signature, and the symbol, respectively. On the other hand, many methods for text recognition or word spotting directly depend on the segmentation of the set of text lines. For instance, given the set of administrative documents shown in Figure 1.2, the final objective is to perform recognition of the handwritten text. Recognition systems could not be applied on the full page, or even on the handwritten region, but they require the explicit segmentation of the text line. Besides, an incorrect segmentation of the characters, or the inclusion of elements from other lines may result in recognition errors.

We aim to investigate new models to tackle these tasks based on the inclusion of contextual information. The role of context on many computer vision tasks, such as object detection or real-scene image segmentation, have proved to be of great importance in order to improve the quality of the results [7]. Recent advances in neuroscience have shown as our visual system make use of these contextual associations continuously and, in fact, we make use of this resource on many other perception tasks [8, 9]. We think that the inclusion of contextual information can add important benefits to the analysis of the document layout and to the segmentation of the text lines. For instance, given an document image, we can identify a particular text region with certain features. However, if we know that the document is a newspaper page, and detect other text regions under it, smaller in font size, and divided in columns, we can guess that the detected regions corresponds to the headline and the body of the new. In this case, the knowledge about the type of document and the surroundings of the detected regions corresponds to the context.

A natural way to represent context is defining the relations between the different entities. Probabilistic Graphical Models (PGMs) provide a powerful probabilistic framework to encode these relationships and to model local contextual constraints between variables. In this dissertation we study different approaches to the tasks of layout analysis and handwritten text line segmentation. We propose several ways

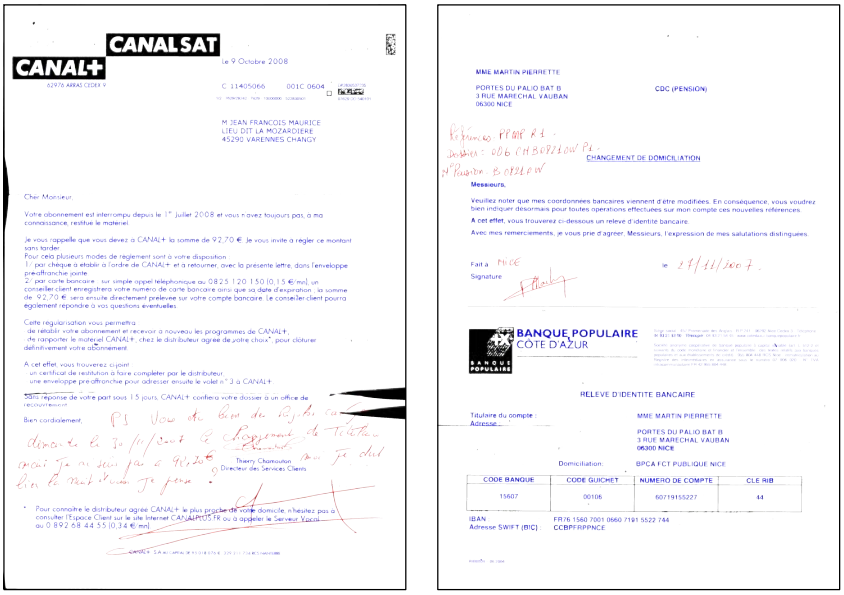


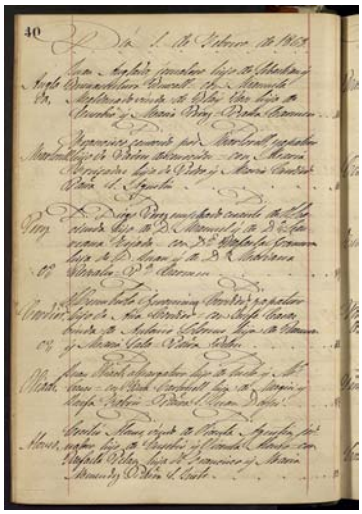
Figure 1.2: Two examples of administrative documents with handwritten annotations.

to encode contextual information based on the use of PGMs and other probabilistic techniques. In the next sections we describe the particularities and main challenges of these task.

1.1.1 Layout analysis

Layout analysis is usually defined as two separated tasks that converge to the complete analysis of the page: physical layout analysis[2], and logical layout analysis. Physical layout analysis aims to detect the entities of a document, such as text blocks or figures, by identifying its location and region boundaries within the page. For instance, given a page from a newspaper, we could focus on identifying homogeneous regions corresponding to the different paragraphs and categorize them as text regions in contrast to background areas. However, we can incorporate additional knowledge about the task in order to categorize each region according to its role within the ambit of document. According to our previous example, a text region could a headline, footnote, body or signature. The logical categorization of the set of functional entities, along with their inter-relationships is referred to as the logical structure of the document, and the process of analyzing this structure is logical layout analysis. Logical layout analysis methods aims at detecting the same homogeneous regions than the previous approaches, but assigning this set of meaningful labels according to its semantic within the page. In fact, the analysis of the logical structure is usually performed on the results of a physical layout analysis stage. In addition, in documents with complex

[2]Also referred to in the literature as geometric layout analysis.



(a)

(b)

Figure 1.3: Two examples of documents with different layout. a) Historical document including a set of structured records. b) contemporary document containing textual elements and images. This is an example of non-Manhattan layout where different regions have irregular shape.

layouts, the physical layout analysis process may require of some logical information and knowledge about the regions to perform its correct identification.

A common categorization of the types of layout distinguishes between Manhattan and non-Manhattan layouts. A Manhattan layout is characterized by rectangular regions that can be separable by vertical and horizontal cuts. Most machine-printed documents, such as newspapers or scientific publications fall in this category. On the contrary, non-Manhattan layouts are characterized by regions of irregular shape and an arbitrary location within the page. Another common classification refers to structured and non-structured documents. Collections of structured documents are characterized by having a predefined organization of its contents. This organization can be limited to a set of regions that are replicated along the page, as some documents that contain a list of records, or to the whole page, as the case of forms. However, non-structured documents are composed of regions that can be located at any place and do not have a relation with other documents within the same collection. Many methods have been developed to perform layout analysis on these types of documents. However, despite the great advances the task is still considered as an open and challenging problem. We summarized the main challenges of the task in four main categories:

- **Document diversity**. Among the main challenges it stands out the large variety of types of documents that can be object of analysis. For instance, regarding machine-printed documents, we can find newspapers, magazines, maps or administrative documents in all their possible forms. Each of these documents has

a particular layout with different challenges. Regarding handwritten documents the variability is even higher. Besides, due to the handwriting style it is common to find overlapped regions on which is not easy to identify its boundaries. Besides, unlike machine-printed documents, it is possible that several regions have different orientations. Historical documents present also some additional complications. They do not follow the formatting requirements of contemporary documents, and consequently their structure can be more difficult to extract.

- **Document degradation**. Although this is a challenge specially present on historical documents, it can also occur on contemporary ones. Ancient documents are usually degraded due to aging or bad conservation issues. Under this conditions they can include some elements that affect to the correct detection of its content, such as holes, ink blots, show-through effects or even missing parts. On contemporary documents, degradation can be result of missing parts result of the acquisition process. These issues can affect to the performance of methods that rely only on physical properties of the document, such as connected components or texture analysis.

- **Generalization**. Developing general methods for layout analysis is one of the main challenges. Many physical layout analysis methods can identify regions by analyzing the distribution of the connected components or the white space between regions. However, many of them usually rely on assumptions about the document structure and are difficult to generalize to very different collections. Some learning-based methods can be adapted to be run on different collections. Nevertheless, the scope of these methods is usually limited to rectangular regions or structured documents.

- **Encoding semantic information**. To perform logical layout analysis implies to be able to encode the functional or semantic information about the regions into the model. Many methods and techniques have been developed for this purpose. However, this is still an important challenge, and for some types of free-style documents this information can be difficult to encode.

We show in Figure 1.3 two documents from different collections that illustrate some of the concepts and challenges previously described. On the left, we show an example of historical document whose content is arranged in a set of records. This is an example of structured document where an structured entity is replicated along the page. This document is characterized for a handwritten longhand script and the existence of some ornamental items that causes overlapping between the different regions. In contrast, on the right, we show a contemporary page from a magazine. In this case the document mix a set of images with printed text regions of different categories, color and size. The comparison between these two documents shows the difficulty of developing general methods that account for such a different layouts.

1.1.2 Handwritten text line segmentation

The problem of detecting text lines was stated decades ago in the context of machine-printed text [10]. Since then, many methods for machine-printed documents have been proposed with remarkable results, to the point of be considered as a solved problem for this type of documents [11, 12]. Machine-printed text lines are expected to be uniform throughout the document, i.e. to share the same orientation and text style, as well as to be free of line overlapping and warping effects. However, when these conditions are not satisfied common methods for this task can fail. Common problems in the segmentation of machine-printed documents come from perspective distortions or curvatures at the page limits. Both situations are related with the acquisition process. Classical methods for this type of documents usually fail under these circumstances, since they assume a certain certain regularity in the skew angle, character size, and spacing between the characters and lines.

Segmentation of handwritten documents is still a challenging problem. In this category we include any document that contains handwritten text, from common letters, where we have an idea about where the text is located, to freestyle handwritten documents, where the handwritten text can be located at any area of the document, or mixed with machine-printed elements. The main challenges for the segmentation of handwritten text lines rely mainly on the large variability in writing styles and in the range of possible document layouts. We enumerate the main challenges concerning this categories in the following points:

- **Text orientation.** In general, it is difficult that all text lines in a document follow exactly the same orientation. This includes from slightly changes to extreme cases that cause text line overlapping. This situation can affect to some methods that assume continuity in the text orientation, or expect certain separation between lines.
- **Curved lines.** Directly related with the previous challenge, curved lines are one of the main challenges for text line segmentation methods. In this case, linear or piecewise linear approximation are in general not enough accurate and require of additional techniques.
- **Text line overlapping.** Another major challenge on this task is when text lines overlap with each other. In most cases this is produced by the contact between ascenders and descenders of characters with long strokes, although it is also produced when text lines are huddled together. This problem is common in some languages with complex characters and many diacritic or punctuation symbols such as Indian Bangla or Chinese.
- **Text style.** Different writers use different font type and size for their texts. Even along documents from the same writer it can be differences in the text style. This situation forces methods to not rely only on these features.
- **Document layout**. Handwritten text can be located at any part of the document. Text in regular letters is usually located at the center of the document or

split in several regions such as address, main body and signature. However, if we think in administrative documents with handwritten annotations, text can be written at any location of the page. In these cases usually is necessary to first detect the text regions and process them separately.

In Figure 1.4 we show several examples of these challenges. We must also highlight the case of historical documents, since they add several challenges to the task [13]. Further than the challenges arising from the document structure and document degradation described in the previous section, there are some artifacts that specially affect to the text line segmentation process. Regarding to the writing style, the variability in types of writers is now extended along the time. This implies multiple changes in the size of the characters, longhand scripts, space between lines or the inclusion of text ornamentation. This variability in combination with the common document degradation is an important challenge even for the best general segmentation methods. Regarding to the document structure, elements like holes, stains or text faint are also a drawback for the detection of text lines. In addition, show-through is specially disturbing for this task.

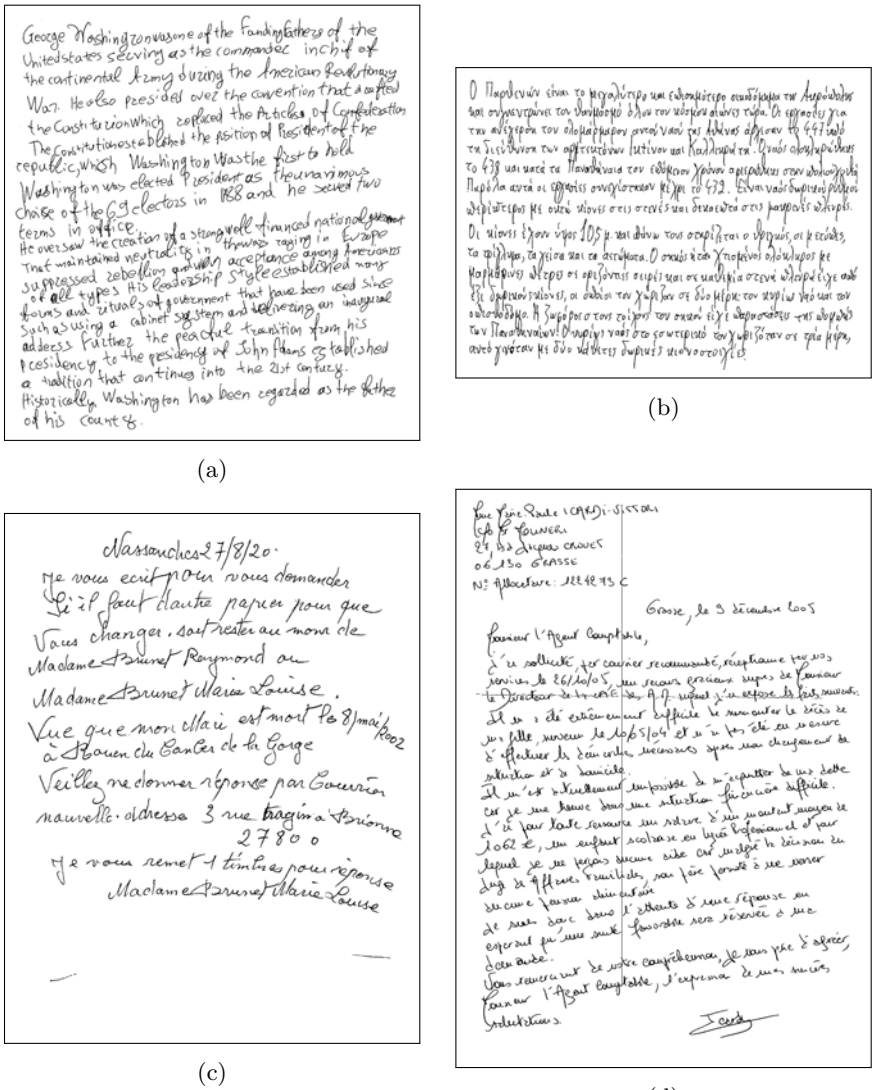
All these challenges promoted that many methods proposed in the last years focus on a particular kind of document collections. Others focus on specific problems, such as touching lines or curved lines, and others are tailored to particular scripts or document layouts that make them hard to generalize to other collections.

1.2 Goals and Contributions

The main objective of this thesis is to develop novel methods and techniques to incorporate contextual information and prior knowledge to the tasks of document layout analysis and handwritten text line segmentation. For this purpose, we present several approaches based on probabilistic graphical models and other statistical resources that encode this information and deal with the main challenges of these tasks. This thesis is framed within an industrial project that aims to develop systems for automatic processing of administrative documents. An important part of this thesis have been focus in the research, prototyping, and development of an application for handwritten text line segmentation devised to detect handwritten annotations on this type of documents. Next, we summarize the set of contributions produced on each part of this thesis.

Document layout analysis.

We propose several ways to include contextual information and prior knowledge into the models to encode the logical structure of the page. The proposed methods produced the following contributions:



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

Figure 1.4: Illustration of some of the main challenges for handwritten text line segmentation. (a) Multiple orientations and huddled lines. (b) Text line overlapping caused by ascenders and descenders. (c) Curved lines. (d) Example of multiple regions with different orientations.

1. **Relative location features for layout analysis.** We introduce relative location features to the document layout analysis process. On this features we can encode spatial relationships and semantic information from the set of entities. We combine them with other texture descriptors in order to perform the physical and logical analysis.

2. **Statistical method for layout analysis.** We present a statistical approach based on PGMs and the EM algorithm for layout analysis devised to structured documents. Our method learns the structure of the document, and perform the identification of the different entities in an iterative way. It can be applied to several types of layouts and it is flexible to light structure changes.

3. **Method for physical layout analysis based on CRFs.** We propose a method for physical layout analysis based on CRFs. Contextual information is encoded on both, the CRF structure and the set of features. Pairwise interaction produce an *smoothing* effect that deals with common degradation issues on historical documents.

4. **Comparative study between PGMs and 2D-SCFG.** We perform a comparative study between syntactic approaches and probabilistic graphical models. We aim to evaluate both approaches on modeling the logical structure. For this purpose, we developed, in collaboration with the PRHLT group from the Polytechnical University of Valencia, a method based on bi-dimensional stochastic context-free grammars for the segmentation of structured documents, and compare it against several PGMs and inference algorithms.

Handwritten text line segmentation.

We propose a general method for freestyle handwritten line segmentation. The main objective is to be able to process any handwritten document regardless of its layout and writing style. The use of contextual information can be useful to overcome the described challenges. For instance, if we define a neighboring system between connected components of text, the decision of labeling each component can be reinforced with the information of its neighbors. This can be useful to deal with curved text lines and character overlapping. We implement this contextual feature among others using a probabilistic graphical model. The main contribution of this work relies in the probabilistic framework designed for this task. We propose a model that can be easily extended with new knowledge about the problem and an iterative algorithm for computing inference and parameter learning on the model. We summarize the produced contributions in the following four points:

1. **General method for handwritten text line segmentation.** We designed our method with the objective to deal with a large variety of documents regardless of their type of layout. Besides, our method is script and language independent.

2. **Easily expandable with prior knowledge.** The proposed probabilistic framework can be easily extended with additional prior knowledge about the task by means of the inclusion of new feature functions.

3. **Probabilistic framework for parameter learning and inference.** We successfully combine the EM algorithm and variational approaches to perform parameter learning and inference on a probabilistic graphical model.

1.3 Thesis Structure

This thesis is structured in seven different chapters:

In Chapter 1, we have described the motivation that led to the realization of this thesis. We detailed the main challenges of the addressed tasks and enumerate the set of contributions derived from the work that we describe on this dissertation.

In Chapter 2, we do a review of the state-of-the-art techniques developed for both layout analysis and handwritten text line segmentation tasks. We highlight the main strengths and weaknesses for each approach in order to justify the work presented in this thesis.

In Chapter 3, we describe the theoretical fundamentals of the statistical models used for the development of the methods presented on this thesis. First, we present a review of the main types of probabilistic graphical models and its application to computer vision tasks. Second, we analyze different approaches to compute inference on these models. Third, we describe the Expectation-Maximization algorithm and its application to estimate the parameters of a Gaussian mixture model. Finally, we present the framework of bi-dimensional stochastic context-free grammars for the task of image segmentation.

In Chapter 4, we present the set of methods developed for the task of document layout analysis. We describe several approaches to encode contextual information and prior knowledge about the problem. We propose two approaches based on PGMs which diverge in the inference process. Besides, we perform a comparative study against 2D-SCFG for the analysis of structured documents.

In Chapter 5, we present the probabilistic framework developed for the task of handwritten line segmentation. We describe two approaches of the proposed method in order to evaluate the contribution of contextual information to the task.

Finally, in Chapter 6, we describe the conclusions achieved and the concluding remarks. Besides, we show the list of publications resulting of the work of this thesis.

Chapter 2

State-of-the-art in layout analysis

In this chapter we review the main approaches and methods developed for the tasks of layout analysis and handwritten text line segmentation in the last years. We highlight the main strengths and weaknesses for each approach in order to justify and contextualize the work presented in this thesis. First we go through the main approaches for physical and logical layout analysis. Then we revise the existing proposals for handwritten text line segmentation.

2.1 Introduction

Major advances in digitization techniques have promoted the interest in developing systems capable of automatically extracting and processing the information contained in digitized documents [10]. The increase on computational resources, together with large improvements on machine learning and pattern recognition techniques, have fostered the development of many methods for document layout analysis and segmentation.

Early works on document layout analysis focused on the physical identification of text regions from machine-printed documents. Later on, this process was extended to other types of documents with different layouts and to other types of entities, such as figures or tables. Logical analysis of the document structure was the next natural step in order to achieve complete analysis of the document contents. This step required the use of additional techniques of knowledge representation and syntactic approaches that account for the structure and relations between the different entities. Nowadays, notable results have been achieved on both tasks an on a large variety of types of documents. However, there are still many challenges to overcome and place to improve, either in processing complex layouts, as in the inclusion of prior information and modeling of the logical structure of the page. Analogously to the de-

velopment of layout analysis methods, important advances were achieved for text line segmentation on machine printed documents. However, in more complex documents, such as handwritten or historical documents, the problem is still open. In this chapter we review the main approaches and methods developed for these tasks, and analyze the main strengths and weaknesses of these approaches in order to demonstrate the contributions presented on this thesis.

2.2 Document layout analysis

In this section, we provide a detailed review of the main approaches developed for the two main tasks: physical and logical layout analysis. There have been many important works that survey the main approaches to these tasks [14, 15, 16, 17, 18]. Besides, there have been held several contests in order to evaluate these methods on benchmark contemporary datasets [19, 20, 1], and historical datasets [21, 22].

A common approach before proceeding with the analysis of a document, is to apply preprocessing techniques that aim to eliminate any imperfections that may affect the performance of methods. Probably the most common preprocessing operation is skew estimation [16]. The objective is to find the orientation angle with respect to the horizontal and vertical directions in order straighten out the document content. Other common operations are: noise reduction operations, perspective corrections, or deblurring processes.

2.2.1 Physical layout analysis

The main objective of this task is to detect and label maximal homogeneous regions corresponding to the text blocks, illustrations, math symbols, or any other entity that belong to the ambit of the document. The common approaches and methodologies applied to this task have been commonly organized within three main categories: bottom-up, top-down, and hybrid methods.

Bottom-up

Bottom-up approaches usually start the analysis at low-level entities such as pixels, which are grouped in other larger entities such as connected components, words, text lines, and ultimately in blocks. Initially this approach was used for the detection of text regions, but it can be also applied to detect other type of entities such as images or tables. An important technique used by many bottom-up method is the Run Length Smearing Algorithm (RLSA) [23]. This algorithm merges character into words, text lines and even text regions by smearing the text area to join the different text components. This method is usually more efficient than other pixel-based approaches, and have produced good results even on noisy and skewed documents. For instance, in [24], a preprocessing step based on the Hough transform and the

RLSA algorithm, is performed for skew estimation and to estimate line spacing. The merging process of the text components is then performed using RLSA smoothing operations. This process results on a set of small regions that are then connected in base of a connected component analysis based on a set of morphological features. Other methods diverge in the process of assembling the low-level entities. Some relevant works are [25, 26, 27]. In [25], the authors propose a similar procedure to analyze the set of connected components based also on morphological features. However in this case they do not work at pixel level, but they estimate the set of bounding boxes around each component, and use these boxes as the units for the sake of reducing the complexity of the method. The docstrum method is one of the most popular methods of this approach [27]. It aims at detecting text regions, tables or equations. First they apply a preprocessing step to remove noise from the image. Then, a clustering process is performed in order to group connected components, first into text-lines, and then in text blocks. The clustering process is performed in base of the distance and angle between neighbor components, which are analyzed within an histogram of features. Other relevant references on the bottom-up paradigm are the Voronoi-diagram-based algorithm in [28], the page decomposition algorithm in [17], or the text/graphics separation algorithm in [29]. Initial bottom-up methods were mainly devised to identify text regions with respect to the background. Other methods widened this process to identify also other types of entities such as images, graphics, or even discriminate between different types of text regions by combining segmentation and classification approaches [30]. Other recent works have focused in the the use of classifiers at pixel level, and then they group pixels according to the result of the classification [31].

Top-down

Top-down methods follow the opposite process for the detection of the different regions. The underlying idea is to start the analysis from the top-level entities or the full page description, and split it into smaller areas such as columns or wide regions. Then these entities are split until finding the wanted areas. The most popular approach form this family is the recursive X-Y-cuts algorithm [32]. The method recursively splits the document image into smaller rectangular regions according to the result of horizontal and vertical projection profiles computed on each iteration. The method estimates the cut areas in base of a set of thresholds from the histogram of projections, which is different for each level. In order to estimate the threshold and the stop criterion of the method, it relies on prior information about the task. Many other posterior methods have been inspired by this approach, eventually modified .

Many other approaches have focused on the analysis of projection profiles. And nowadays the naive version based on the analysis of peaks and valley on the resulting histogram is used as baseline for comparing the performance of other approaches [33, 34]. In [33], projection profiles are computed from the set of bounding boxes of the connected components. It is applied for the detection of text paragraphs, lines and words. In [34], the authors aim to detect text regions and lines in documents with a little skew. The approach is based on dividing the image in vertical strips

and computing projection profiles on each of them. Then, results on each strip are combined to build the final regions. Other important work is [35], where the authors present a method for the analysis of columns of text. They propose a parametric model for which they call global-to-local analysis, which infers the segmentation of text columns based on a set of assumptions of the shape of the regions.

Hybrid and other techniques

Hybrid methods combines the previous approaches to exploit the benefits from both methodologies. Some examples of hybrid methods are [36, 37, 38]. In [36], a top-down approach is used to locate the horizontal and vertical white separators between columns and large blocks of text. Then, connected components are merged to compose the set of text-lines and final regions on several orientations. In [37], the hybrid method consist in a bottom-up approach to perform a first guess of the region shapes and the location of the tab-stops. Then, these tab-stops are used to deduce the column layout and the reading order of the page following a top-down approach.

Other types of methods combine the detection process with classification techniques in order to discriminate between different types of regions, such as images, equations or different categories of text regions [30]. Neural networks have been also used for this purpose [39]. The use of neural networks avoids the manual selection of model parameters used by other classification methods. Another example is the use of binary classification trees for the classifications of extracted regions in several categories [40]. Text classification features have had an important role in the classification process. Texture descriptors are one the most used features to discriminate between the set of document entities [41, 42, 43]. Besides, many works also used morphological features, such as contour analysis the morphological distribution of text lines [44].

2.2.2 Logical layout analysis

Logical layout analysis aims to assign a set of meaningful labels to the set of regions that compose the document. To achieve this objective, methods proposed for this task require of a prior knowledge about the document structure, the semantic of the regions, and the relations between them. In this section we describe three main categories of logical layout analysis methods: tree-representation approaches, syntactic analysis-based approaches, and probabilistic methods.

Tree-representation

A popular approach in the last decades has been to represent the document logical structure by trees derived from a set of rules. The set of rules is defined according to the knowledge about the task. The layout analysis process consist in the construction of these trees starting from the physical entities detected. For instance, in [45], the authors propose a method to derive the logical tree and define the reading order

according to a set of deterministic rules. This method obtained very good results on several type of machine-printed documents such as magazines, scientific publications or newspapers. In [46] the tree derivation process is performed from a set of weighted rules. They propose a search algorithm through the different hypothesis in order to find the most probable logical tree. Some other methods have been proposed based on this approach, which usually diverge on the initial method to obtain the physical regions and the type of defined rules.

Syntactic analysis

Syntactic methods make use of context-free grammars to represent the knowledge about the structure of the documents [47]. The analysis of the document layout is therefore presented as a parsing process according to the defined grammar. Basically, the problem is reduced to finding the most likely parse tree for a given document structure. The process usually perform the physical and logical steps together, since the shape and relative location of the regions is usually encoded in the grammar together with its semantic value. In [32], the authors present a syntactic approach based on context-free grammars for the analysis of technical journals. Grammar rules were defined to aggregate pixels along the parsing process in order to form the different logical regions. More recently, probabilistic versions of context-free grammars have been also used for structural and logical layout analysis [48, 49]. Besides, the bi-dimensional approach for these grammars offers advantages for the definition of the rules, and the type of tasks that can be addressed by representing bi-dimensional contextual relations [50, 51]. A recent work using grammars proposed a discriminative grammar instead of the common generative approach used on previous approaches [52]. This work proved that discriminative models are more powerful than the previous probabilistic context-free grammars.

Probabilistic and learning-based methods

Probabilistic resources have been limited up to this point to weight the grammar rules or to the use of classifiers. The possibility of learning-based method have been explored in the past, with the objective of creating methods able to learn a structure and infer a solution given a document. Some early works incorporated learning systems able to discriminate between document categories [53]. The system incorporated specific derivation rules for each document category, which were applied according to the classification step. However, was the use of Hidden Markov Models (HMMs) what suppose a major change in the use of probabilistic methods. A method based on an extension of HMMs was proposed in [54] for page decoding. The authors proposed a generative model to manage bi-dimensional regions and infer the logical structure based in the analysis of sequences of small portions of the document. In posterior works, they presented a significant improvement on this method in terms of computational efficiency [55]. Another example of learning-based method is [56]. In this work, the authors first generate a set of overlapping zones according to the Voronoi

tessellation. Then, they propose an inference process based on a linear constrained optimization problem to find the optimal configuration of overlapping zones. The use of probabilistic graphical models have been also extended from HMMs to other models. In [11], the authors propose to use a Markov Random Field to model the statistical relationships between the different elements on the document. More recently, in [57] a hierarchical CRF is proposed for noise removal and text region labeling using a set of Globally Matched Wavelet (GMW) filters as features.

2.2.3 Discussion

We reviewed the main approaches developed for the problem of layout analysis. Methods for physical layout analysis have obtained good results on many types of documents. However, a common problem to all the described approaches relies in the adaptability to other collections. Most methods are based on assumptions about the document structure, and are difficult to generalize to different collections. Top-down methods, for instance, can only be applied to detect polygonal regions with a clear separation between them. Bottom up approaches usually fail if the regions are overlapped and if the document contains noise or degraded parts. Hybrid approaches, in the same way that get profit from the advantage of the other approaches, it is also weak against the cited situations. Probabilistic methods can deal with some of these problems, as the case of overlapping regions or the adaptability to other collections. Besides, in some cases they provide a probability of the final detection that can be uses as a measure to the quality of the results. However, they usually require of training data that is not always available, and the learning and inference processes can also lead to computational problems on particular configurations of complex models.

Logical layout analysis methods are able to encode hierarchical relations and semantic information about the entities, which is useful to guide the region detection process. However, many of them work on the already detected regions, so they can drag all the previous stated inconvenients. Instead, some of them combine the physical detection with the logical categorization process, which is very convenient to deal with ambiguities in the labeling process. Many logical layout analysis methods rely on a previous knowledge about the problem. The type of knowledge depends on the desired level of analysis. The more detailed categorization of the document entities, the more prior information is required. This condition presents a situation where very specialized methods are not able to generalize on other collections, unless new prior knowledge is added to the model. For this reason, some of the best reported layout analysis systems are usually focus to particular types of documents layouts. Methods that are able to learn this knowledge have been also reported. An important characteristic of learning-based of methods is the flexibility to assume changes with respect to the learned structure, for instance, in what regards to the region sizes or range of locations within the page. Regarding rule-based methods, the set of derivation rules can become rather arbitrary and, as we commented before, the adaptability to other collections of documents can be difficult. Besides, the use of deterministic rules may lead to ambiguities of the tree-construction process. Syntactic approaches based on grammars also suffer from the adaptability problems. Besides, the complexity of

the parsing process can be a drawback for many tasks, since it is exponential in the number of terminal symbols.

2.3 Handwritten text line segmentation

There have been many attempts to tackle the task of text line segmentation from different perspectives. As a particular case of physical layout analysis, common approaches are based in the bottom-up and top-down paradigms. However, further than the common paradigms, many other have emerged in order to cover the variations that handwritten documents present. We have categorized them into Hough-based methods, morphology-based methods, and other approaches. Several contest have been also held on benchmark datasets [58, 2, 3]. In the following, we highlight the main families of methods, as well as the advantages and disadvantages of some representative works for each of them.

Bottom-up

Bottom-up methods, analogously to layout analysis methods, are based on the analysis at pixel level or at connected component level, which are then combined to form characters, words and ultimately, lines. Next we explain the most popular mechanisms for the detection of text lines. Some methods use criteria based in geometric relationships between the components, as distance, angle, overlap degree or similarity [30, 59]. The CMM method [3], for instance, groups connected components that are mostly horizontally aligned. Other works propose a clustering method based in a metric distance computed using the Minimal Spanning Tree (MST) technique [60]. In [61] the authors define a text line as a cluster of connected components, and propose an optimization method to minimize a fitting function that combines the fitting error of each component to the line, and the distance between lines. In [62], the authors propose a probabilistic approach to estimate the text line density function based in an anisotropic Kernel, and use the level set method to determine the line boundaries. Other reference works on this paradigm are [42, 63, 64, 65].

Top-down

Between these most popular top-down approaches we must quote again the projection profile-based methods as the successful approaches [32]. Line segmentation methods based on this approach are based in projecting textual pixels on the vertical axis, which results in the histogram projection of the distribution of the text components. Maximum and minimum peaks shall represent, in an ideal case, the location of the text lines and line spacing, respectively. A example of this approach is the work in [66]. However, these methods are sensitive to orientation changes or to curved lines [67]. A common solution to deal with this problem is to divide the document into

vertical strips and compute histogram projections on each region. Then, merge the results of consecutive strips to form the final lines according to geometric properties [68, 69, 70, 71], or probabilistic approaches [72, 73]. Projection profile techniques are also widely used for the task deskewing documents [74]. In addition, it is common to use this approach to find an initial location of the lines, and then run another more sophisticated method to find them [75, 59].

Hough-based

This family of methods use the Hough Transform [76] to locate the different text lines [29, 77]. This transform extract some key points of the image and computes a set of lines that best fit to these set of points [78]. Other works propose a scheme to compute the alignments of the different line hypothesis obtained to extract the set of text lines [77]. Besides of this mechanism, the authors also use contextual information about neighbor hypothesis to enrich the final decision. In [29], the alignment in the Hough domain is performed by exhaustive search in several directions starting from the minimal points on the left of the document. Other important references are [79, 80, 75].

Morphology-based

Morphology-based methods, including smearing-based approaches, also produce good results [81, 82, 83]. These methods make use of the morphological properties of the documents to infer the location of the lines. In [84], the authors combine basic erosion and dilation operators with histogram projections to detect the text lines and deal with the overlapping problem. The run-length smearing algorithm (RLSA) is another example [23]. The fuzzy [85] and adaptive [58] versions of this algorithm have been applied to handwritten documents with remarkable results in the case of skewed and curved lines.

Other methodologies

Other types of methods have been developed for the purpose of segmenting text lines. Graph-based approaches [86, 87, 3] extract the graph representation of the document and exploit the graph properties to estimate the text line locations. In [86], for instance, the authors compute the skeleton of the text components to build a graph. Then they propose a mechanism based in the A^* algorithm to find the set of paths of minimum energy between the nodes that represent the space between the text lines. The approach of finding the minimum cost path have been also exploited using dynamic programming methods [88] or the Viterbi algorithm [73]. Active contours (snakes) [89] is another technique that has produced good results [90, 91]. Statistical approaches are less common in the literature for this task. Statistical resources have been used to model density functions to discriminate between text and background

[62], or as a tool in the post-process steps to split touching characters using a linear SVM classifier [73]. Another example is found in [72], where the decision of associating a touching component to one or another line is given by a set of bi-variate Gaussian distributions.

2.3.1 Discussion

We reviewed the main approaches to the task of handwritten line segmentation. Many are able to deal with the most challenging problems of this task, however, the effort to overcome these problems usually translates into a weakness against other situations.

Text line overlapping is one of the most challenging situations for all the approaches. General bottom-up methods are specially sensitive to noisy or crowded documents, since the merging process usually fails under these circumstances. Top-down methods based on projection profiles, and some methods based on morphological operators, are also sensitive to these conditions. As a consequence of this, many of these methods are difficult to generalize to other scripts or languages, since they usually require specific heuristics or post-process steps to detect overlapping and treat it aside [62].

Curved lines are also an important issue for projection-based and hough-based methods. Instead, methods based on bottom-up approaches or morphological operations are usually robust against this condition. A similar effect is produced if text regions contain text lines in multiple orientations. While, top-down approaches can deal with different page and region orientations, they can be affected by this effect. In general, bottom-up, hough-based and morphological-based methods are also robust to these conditions.

Chapter 3

Theoretical framework

In this chapter we give an overview of the main statistical techniques that have inspired the methods presented on this thesis. We review the main concepts about probabilistic graphical models and describe the main inference algorithms used in this work. In addition, we also describe the framework of bi-dimensional stochastic context-free grammars to be applied on a page segmentation task.

3.1 Introduction

The use of machine learning techniques and other statistical models have contributed to achieve great improvements in computer vision and document analysis tasks. With these methods we are able to efficiently learn from observed data, and to perform a set of deductions about unknown information in order to provide a solution to a problem. For the development of the methods presented in this thesis we have used a set of probabilistic techniques. Specially, we have focused in the use of Probabilistic Graphical models.

Probabilistic graphical models are a powerful formalism to encode relationships between multiple variables. It combines the robustness of probabilistic resources with the flexibility of graph theory. PGMs rely on the concept of conditional independence[1] to represent in a diagrammatic representation the probability distribution between a set of variables. It is an important concept, since it determines the structure of the model, and helps to simplify the estimation of the probability distribution in views to compute inference on the model.

There are several important properties of PGMs that justify their success in many

[1]The conditional independence condition states that two variables A and B are conditionally independent given a third variable C if, given the value of C, the knowledge about the value of B does not affect to the estimation of A, formally $p(A|B,C) = p(A|C)$.

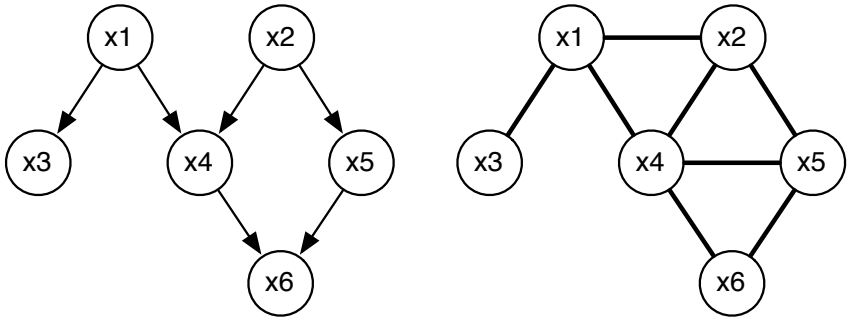


Figure 3.1: Example of Bayesian Network and Markov Random Field for a given set of variables.

computer vision tasks, among other fields. On the one hand, they are able to model contextual constraints between variables, and allow to incorporate additional assumptions in the form of prior probability distributions in the same graphical formulation. On the other hand, the graph representation provides an easy way to visualize and model the structure of a given problem. Last, there is an extensive research on the problem of computing inference and parameter learning that allows the application of this framework to a large collection of problems, specially in the case of problems where uncertainty plays a role.

Depending on the set of conditional independences that can be encoded in the model, and the family of distributions that derives from these relationships, we distinguish between directed models, also referred to as *Bayesian Networks*, and undirected models. Both representations model probability distributions of the variables, and depends on the definition of the problem to choose each of them. Among the main types of undirected models we highlight *Markov Random Fields*, *Conditional Random Fields*, and *Factor graphs*.

In this chapter we describe the theoretical background of the probabilistic models used for developing the methods we present throughout this thesis. First, we introduce the basic notation. Second, we describe the fundamentals of the main types of PGM. Then, we review the problem of inference and parameter learning and describe the used methods. Finally, we describe other statistical methods used in this thesis: EM algorithm, and Stochastic Context-Free Grammars.

3.2 Notation

A PGM is formally represented as a graph denoted by $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the set $\mathcal{V}$ represent the nodes on the graph, and $\mathcal{E}$ the set of edges. The set of nodes corresponds to a set of discrete random variables X. We denote the specific values of these variables as x, and concretely, as x_i for a particular node $i \in \mathbb{V}$. We refer to a set of unknown

variables as Y, and analogously, y and y_i denote the realization of the whole set and of a particular variable, respectively.

3.3 Bayesian networks

Bayesian networks (BN) are probabilistic graphical models represented by Directed Acyclic Graphs (DAG). An arrow in the model is used to establish a conditional dependence between the two involved variables. The definition of this kind of relationships allows to factorize the joint probability distribution $p(x)$ into a product of conditional and prior distributions as:

$$p(x) = \prod_{i \in \mathcal{V}} p(x_i | x_{\pi_i}) \tag{3.1}$$

where x_{π_i} denote the variables to which x_i is conditioned in the model, also referred to as *parent* variables. We depict an example of BN for a given set of variables in Figure 3.1a. According to the depicted model, the joint distribution between the variables is expressed as:

$$p(x) = p(x3|x1)p(x4|x1, x2)p(x5|x2)p(x6|x4, x5)p(x1)p(x2) \tag{3.2}$$

which results in a more compact and easy way to compute it, than the usual chain rule. BN can be used to model probability distributions on the condition that they satisfy the called *local Markov property*. This condition basically states that all the variables in the model have to be conditionally independent of any of their non-descendants given their parent variables. Formally:

$$\forall i \in \mathcal{V}, x_i \perp x_{N_i} | x_{\pi_i}$$

where x_{N_i} denotes the set of non-descendant variables. BN are used to model problems where exists a relation of causality between variables, such as speech recognition systems or some predictive applications. They have been applied to many research fields, such as computational biology, monitoring industrial processes, or decision support systems.

3.4 Markov random fields

Markov Random Fields (MRFs) are represented by an undirected graph that can contain loops. As in the case of BN, variables have also to satisfy the *local Markov property*. The notion of parents and descendants is substituted by a local neighborhood of connected variables, and analogously, this property states that a variable has

to be conditionally independent of any other variable in the model given all its neighbors. In addition, it must satisfy the called *global Markov property*, which states that any two subsets of variables (A, B) are conditionally independent given a separating subset C (where every path from a node in A to a node in B passes through C). Formally:

$$X_A \perp X_B | X_C$$

An important notion in MRF is the concept of clique. A clique refers to a subset of vertexes such that every two distinct vertexes in the clique are adjacent. According to this definition, a clique is *maximal* when it can not be extended by including any other vertex. In the example of Figure 3.1b, an example of maximal clique is the one composed of the variables $(x4, x5, x6)$.

According to the Hammersley-Clifford theorem [92], a probability distribution is represented by a MRF defined by the graph $\mathcal{G}$, if it satisfies the Markov properties with respect to $\mathcal{G}$, and its density function is strictly positive. In this case the probability distribution is a *Gibbs distribution*, and therefore $\mathcal{G}$ is a *Gibbs random field* that can be factorized over the set of cliques of the graph as:

$$p(x) = \frac{1}{Z} \prod_{c \in C} \Psi_c(x_c) \tag{3.3}$$

where $\Psi_c(x_c)$ are potential functions defined over the variables of the clique c, and C is the set of all cliques in $\mathcal{G}$. Potential functions are not necessarily probability functions. This is one of the main differences with respect to BN, which are factorized in a set of conditional and prior probability functions. This arbitrariness in the definition of the potential functions makes necessary a normalization step in order to define a proper probability measure. This normalization is represented by the term Z, also called partition function. The computation of Z is one of the main challenges for computing inference and parameter learning in MRF, since it sums over all possible configurations of x as:

$$Z = \sum_{x} \prod_{c \in C} \Psi_c(x_c) \tag{3.4}$$

which is not always possible to compute depending on the model. We discuss this problem in Section 3.5. Another common approach to define the joint distribution of a MRF relies on the definition of energy functions over the set of cliques. The energy of a clique is defined as:

$$\theta_c(x_c) = -\log \Psi_c(x_c) \tag{3.5}$$

since by definition we are sure that potential functions Ψ_c are strictly positive. In this way, we can define the energy of a MRF in terms of a sum of clique energies, which is easier to manage compared to the previous product of potential functions:

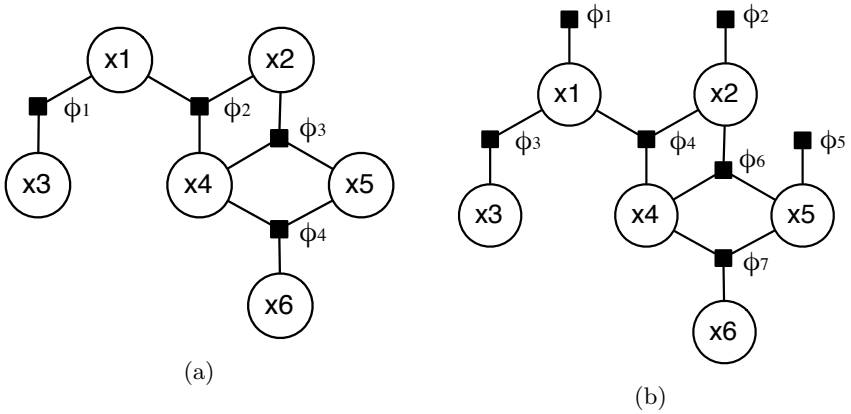


Figure 3.2: Two possible configurations of the factor graph linked to the previous models.

$$E(x) = \sum_{c \in C} \theta_c(x_c) \tag{3.6}$$

and then, define the joint probability distribution in terms of this energy as:

$$p(x) = \frac{1}{Z} \exp\{-E(x)\} \tag{3.7}$$

The use of MRFs is highly extended in many fields due to their ability to model contextual relationships between variables and to solve labeling and parsing problems. In computer vision, they have been applied to many tasks such as image restoration, stereo-vision, optical flow or image segmentation. On this type of problems, the most common MRF topology used is the pairwise MRF model. This type of MRF consist of the definition of potential functions over cliques of less than three variables. This includes potential functions over single variables, called *unary potentials*, and a set of potential functions on pairs of variables, called *pairwise potentials*. The most popular pairwise configurations are *grid-like* models and *part-based* models. The first one takes advantage from the grid structure from the pixels of an image and establish relations between neighbor pixels. In this case the cliques are defined from neighbor variables considering a 4 or 8 neighborhood. The second one is used for problems with a deformable structure between the variables such as pose recognition or other tasks with pictorial structures. The energy of these type of models is usually defined as:

$$E(x) = \sum_{i \in \mathcal{V}} \theta_i(x_i) + \sum_{\{i,j\} \in \mathcal{E}} \theta_{i,j}(x_{i,j}) \tag{3.8}$$

where $\theta_i(x_i)$ are the unary potentials, and $\theta_{i,j}(x_{i,j})$ the pairwise potential for each

pair of connected variables (i, j) on the defined MRF. Another important type of MRF is the conditional random field. Here, the main features do not rely on the structure of the model, but in the type of probability distribution modeled.

3.4.1 Conditional random fields

Conditional Random Fields (CRFs) are a type of graphical model used to encode a conditional probability distribution between a set of latent variables and a set of observations. Essentially a CRF follows the same principles than the ones described for MRFs, since it is subset of them, with the exception that a CRF is globally conditioned to the set of observations.

CRFs were introduced by Lafferty *et al.* in [93] as an alternative of MRF to label sequences of data. In contrast to the generative approach of MRF, which usually requires lots of data to learn the model parameters, CRFs are discriminative models that focus on the data distribution, and condition the set of labels to this data. For instance, in an image segmentation problem we can consider a set of latent variables y, representing the class labels, which are conditioned to a set of observations x from each image pixel that we want to segment. CRFs do not model the probability distribution over the observed variables x, which dependencies can be complex. This simplifies the modeling of the distribution between the two sets, which is the main advantages of CRFs. The conditional distribution modeled is also a *Gibbs distribution*, and therefore it can be defined as a product of factors on the cliques as:

$$p(y|x) = \frac{1}{Z(x)} \prod_{c \in C} \Psi_c(y_c, x_c) \tag{3.9}$$

In the same way than for MRF, we can also define the conditional distribution in terms of the energy associated to the set of clique potentials:

$$p(y|x) = \frac{1}{Z(x)} \exp\{-E(y, x)\} \tag{3.10}$$

note that the partition function depends on the set of observations, and only sums over the set of latent variables y as:

$$Z(x) = \sum_{y} \prod_{c \in C} \Psi_c(y_c, x_c) \tag{3.11}$$

Again, in the computation of the partition function relies the main difficulty to compute inference on these models. CRFs have been widely used on many labeling tasks, such as image segmentation and object recognition.

3.4.2 Factor graphs

Eventually, when we model a problem using a PGM, it can be convenient to have a representation that directly specifies the factorization of the model. This type of representation is called factor graph, and can be defined for both BN and MRF models. A factor graph is an undirected bipartite graph that connects a set of variables $i \in \mathcal{V}$ with a set of factors nodes $F \in \mathcal{F}$. It is an useful representation in order to easily visualize the factorization of a graphical model and to understand certain families of inference algorithms. On this type of graphical model, each factor is represented by the symbol ■, which is connected to the set of variables that are part of the definition of the factor. As we described before, the same MRF or BN structure is use to model a family of probability distributions that satisfies the Markov properties. Therefore, the structure of the factor graph linked to a particular MRF or BN is not unique, since it depends on the definition of the set of factors, and this depends on the definition of the problem. We show in Figure 3.2 two examples of factor graphs that are valid to represent the graphical models depicted in Figure 3.1.

3.5 Inference and parameter learning

When we use MRFs to model a particular problem, once we have defined the factorization of the model there are two main tasks to perform: *parameter learning*, and *inference*. Parameter learning refers to estimation of the parameters linked to the set of defined factors. A defined MRF describes a family of probability distribution, but by parametrizing the set of factors we can specialize the model to the specific problem addressed. Once we defined the model and its parameters the inference process consist on estimating the optimal configuration of the model variables so that it maximizes the probability distribution, or instead, minimizes the energy of the model. In this section we describe the main methods developed for the inference task. In particular, we focus on the methods devised to discrete MRFs, since it is the type tasks that we address in this thesis. However, many of the described approaches have approximations for the continuous case, and an extensive research have been also done for this type of models.

The inference problem on MRFs is usually formulated in terms of Maximum a posteriori (MAP) estimation as:

$$\hat{x} = \underset{x}{\operatorname{argmax}}\, p(x) \tag{3.12}$$

according to the probabilistic interpretation, or, defined in terms on an energy minimization process in case of defining the MRF as the sum of clique energies (see Eq. 3.6):

$$\hat{x} = \underset{x}{\operatorname{argmin}}\, E(x) \tag{3.13}$$

in any case, this problem is known to be NP-hard [94], and results intractable for most MRF configurations, except for some specific model topologies as tree-like structures. The main difficulty lies in the number of variables and the graph structure. For a reduced number of variables it is possible to compute exact inference in a tractable time, but when this number increases, the complexity also does in an exponential way. Regarding the structure, algorithms as Junction Tree are able to achieve exact inference on trees, although its complexity is also exponential with the size of the cliques. On real applications the exact computation is not possible, and we must resort to approximate methods. Fortunately, there have been an extensive research on this problem that have produced important results and methods.

From the energy minimization perspective, methods such as *simulated annealing (SA)* or *Iterated conditioned modes (ICM)* where proposed decades ago with important results in the field. The ICM algorithm [95] belongs to the family of local search methods. It relies in the idea that if all the variables but one were observed, it is easy to compute MAP estimation for this variable. Therefore, the algorithm iteratively computes the probability for each variable according to the value of the rest, which are then updated. SA algorithm instead relies on a meta-heuristic to maximize the optimization function. It is based on the Monte Carlo method to generate sample states of a thermodynamic system. Both algorithms have the inconvenient of falling on local maximum. However, they are still a good choice for many tasks.

More recently, other families of methods such as Graph Cuts (CG) approaches or Loopy Belief Propagation (LBP) have been proposed with remarkable improvements to the task. Besides, optimization techniques have been also improved with methods such as the LP primal-dual optimization and other dual approximations. In the next sections we review some these families of methods in detail.

3.5.1 Graph Cuts

Graph Cuts-based methods are a very popular approach due to its efficiency and high quality results achieved on many visual perception problems. They rely on the *max-flow/min-cut* theorem in order to minimize the energy of discrete MRFs [4]. The basic form of Graph Cuts is to construct a graph $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$, called *s-t graph*, with three types of nodes $\mathcal{V} = \{s, t, p\}$. Two of them are called terminal nodes, which correspond to the source s and sink t nodes in the context of *max-flow/min-cut* problem. The set p corresponds to the set of pixels $p \in P$ from the modeled image. There are also two types of edges defined as $(p, q) \in \mathcal{E}'$. First, t_s edges connects the source node s with each of the pixels p. Second, t_r edges connects analogously the sink t with the pixels. Each of the edges has a non-negative capacity $w(p, q)$. The objective of the method, is to find a cut $\mathcal{C}$ such that $\mathcal{G}' = (\mathcal{V}', \mathcal{E}' - \mathcal{C})$ of minimum cost, where at least each pixel is connected with a terminal node. The overall cost is defined as:

$$C(\mathcal{C}) = \sum_{(p,q)\in\{\mathcal{E}'-\mathcal{C}\}} w(p, q) \tag{3.14}$$

The use of this approach on some binary problems (such as denoising a binary image or binary segmentation) proved to obtain the exact solution in polynomial time. However, in problems with more than two possible labels the exact computation is not possible and approximations are required.

The previous definition describes Graph Cuts from the graph theory perspective, however, the usual interpretation of this algorithm on a computer vision task is done in terms of an energy minimization problem. Here, the objective is to find a labeling f that minimizes the energy:

$$E(f) = E_{data}(f) + E_{smooth}(f) \tag{3.15}$$

we can see the correspondence between this equation and the definition of the energy on a MRF in Eq. 3.8. Here, $E_{data}(f)$ represents the unary potentials, or the cost of assigning the labeling f according to the observed data. $E_{smooth}(f)$ represents the pairwise potentials. In the context of Graph Cuts is also defined as a *discontinuity preserving* function, and measures the cost of assigning neighbor pixels p, q with the labels f_p, f_q, respectively. The form of these functions is usually defined as:

$$E(f) = \sum_{p \in P} D_p(f) + \sum_{\{p,q\} \in \mathcal{N}} V_{p,q}(f_p, f_q) \tag{3.16}$$

where $\mathcal{N}$ is the set of neighbor pixels according to the defined structure. The definition of functions D_p and $V_{p,q}$ depends on the requirements of the problem. However, there are some restrictions that they have to satisfy. In the case of $V_{p,q}$ it has to be defined as a *metric* or *semimetrics*, and therefore satisfy the properties that define them. More information about this constraints can be found in the original paper [4].

The common use of Graph Cuts on multi-label problems is when $V_{p,q}$ takes the form of a Potts model [96]. On This model $V_{p,q} = u_{\{p,q\}} T(f_p \neq f_q)$, where $u_{\{p,q\}}$ is the cost of assigning to neighbor pixels the labels f_p and f_q, and $T(\cdot)$ is 1 when the argument is true or 0 otherwise. The Potts model in the frame of an energy minimization problem is related to the known multiway cut problem, and therefore, the problem can be solved using this approach. The multiway cut problem is also defined on a graph $\mathcal{G}' = \{\mathcal{V}', \mathcal{E}'\}$. The set of nodes is composed by two kinds of nodes $\mathcal{V} = \{l, p\}$. First, there is a set of *l-nodes*, which corresponds to each of the L labels of the problem. Second, there is a set of *p-nodes*, which corresponds to the P image pixels. The set of edges is also composed by two different sets. First, *t-links* connects each of the terminal nodes to the set of pixels. They have a weight assigned $w_{\{p,l\}} = D_p(l)$. Second, *n-links* connects the set of $p - vertices$ according to a defined neighborhood. Each *n-link* has also a weight such as $w_{\{p,q\}} = u_{\{p,q\}}$. An illustration of this structure is shown in Figure 3.3a. The labeling problem is then defined as finding the cut $\mathcal{C}$ that minimizes the energy defined in Eq. 3.16. The cut is constrained to include only one *t-link* between a pixel and the terminal nodes. An example of a multiway cut is shown in Figure 3.3b. It is proved that finding the minimum cut corresponds to the solution to the energy minimization problem on MRF.

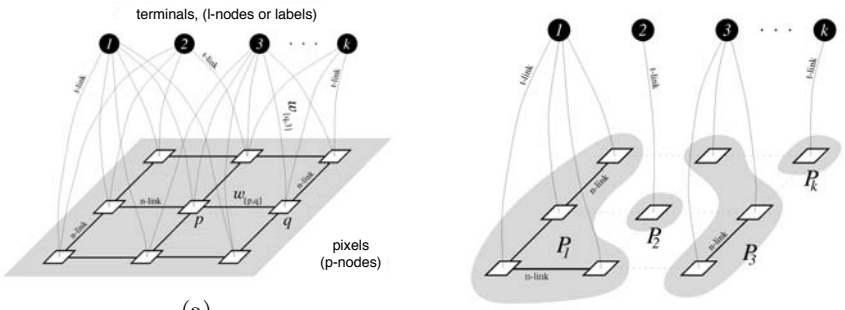


Figure 3.3: Illustration of the multi-label GC process. [4]

3.5.2 Belief propagation algorithms

Belief propagation (BP), also referred to as sum-product algorithm, is a *message-passing* algorithm widely used to compute inference on different families of PGMs. The algorithm is defined on the factor graph of the problem, and aims to compute the partition function Z and the marginal distributions for all variables $p_i(x_i)$, and for each factor on the model $p_F(x_F)$. The algorithm was initially formulated for tree-structured PGMs, for which it is possible to compute exact solution to these problems [97]. Later on, an extension of BP was proposed to its application on general graphs with loops, in which case it provides an approximation to the marginals and partition function. This version of the algorithm is referred to as *Loopy Belief Propagation* (LBP). In this section we first describe the main concepts of the BP algorithm for tree-structured graphs. Then, we describe the main variations of the loopy version and how it is applied to compute MAP estimation on computer vision tasks.

The basic idea of BP is to exchange a set of *messages* between variables and factors. The meaning of these messages can be understood as a conditional function of a variable or factor, given the information received from the rest of the model. Two types of messages are involved in the algorithm: variable-to-factor messages, $m_{x_i \to F}$, and factor-to-variable messages, $m_{F \to x_i}$. In order to introduce the formulation of each part of the algorithm, we define the set $\mathcal{N}(F)$ as the list of variables linked to a factor F, and the set $\mathcal{M}(i)$ as the list of factors to which the variable i is connected. First, variable-to-factor messages are defined as:

$$m_{x_i \to F(x_i)} = \sum_{F' \in M(i) \setminus \{F\}} m_{F' \to x_i} \tag{3.17}$$

which means that the message sent to a factor is the sum of all the messages received on i, except the one from this factor. Similarly, the factor to variable message is defined as:

$$m_{F\to x_i} = \log \sum_{x_F \setminus x_i} \exp\left(-E_F(x_F) + \sum_{j\in\mathcal{N}(F)\setminus\{i\}} m_{F'\to x_i}\right) \tag{3.18}$$

which is the sum of the factor potential with messages from all other variables, marginalized over all variables except the one associated with i. Note that for numerical reasons we are defining the algorithm in terms of log-factors and log-messages. Factor potential $-E_F(x_F)$ corresponds with the clique energy defined on Eq. 3.5. In the case of tree-represented graphs, the message order is determined by the structure of the tree. The common procedure is to first send messages from the leaves to a node previously defined as a root (*leaf-to-root*). Then, in a second step, it proceeds in the inverse order sending messages from the root to the leafs (*root-to-leaf*). In this kind of graphs is enough to perform these two steps once in order to ensure convergence and compute exact values of the partition function and the set of marginals.

In the case of general graphs, the main difference relies in the message-passing process. In graphs with loops, as grid-like graphs, no particular order can be defined since loops incorporate redundancy to the process. This condition results on non-exact messages, which are usually referred to as *beliefs* in the general form of the LBP algorithm. The usual procedure on loopy graphs is to initialize all messages to a fixed value, and then perform message updates iteratively in a defined order. The message-passing scheme can be addressed on two ways: sequential or parallel. Some works have study both approaches and proved that even though the sequential approach is faster, results remain almost the same [98]. In any case, messages are now approximations of the true beliefs, and therefore the partition function and marginals are approximations of the exact values. The formulation of the messages compared to the original BP introduces some changes. While the equation for computing factor-to-variable messages in Eq. 3.18 remains equal, the equation for variable-to-factor messages changes slightly to introduce a required normalization factor. Thus, being m' the message defined in Eq. 3.17, the normalized expression is:

$$\begin{aligned} \lambda &= \log \sum_{x_i} \exp(m'_{x_i\to F(x_i)}) \\ m_{x_i\to F(x_i)} &= m'_{x_i\to F(x_i)} - \lambda \end{aligned} \tag{3.19}$$

where λ is the normalization term that includes messages sent from other variables to the objective factor. It is common to perform several iterations of the message-passing process aiming at achieving more accurate messages. Once this process finish, it is possible to compute the partition function and the set of beliefs for each factor and variable. The factor beliefs are computed as:

$$p_F(x_F) = \frac{1}{Z_F} \exp\left(-E_F(x_F) + \sum_{j\in\mathcal{N}(F)\setminus\{i\}} m_{F'\to x_i}\right) \tag{3.20}$$

where the normalization term is defined as $Z_F = \log \sum_{x_F} \exp(p_F(x_F))$. Similarly, the variable belief is defined as:

$$p_i(x_i) = \frac{1}{Z_i} \exp \sum_{F' \in M(i)} m_{F' \to x_i} \tag{3.21}$$

where the normalization term is $Z_i = \log \sum_{x_i} \exp(p_i(x_i))$. In order to compute the partition function, the set of local normalization functions Z_F, Z_i have to be taken into account independently. Thus, the expression to compute Z is:

$$\begin{aligned} Z = &\sum_{i \in \mathcal{V}} (|M(i)| - 1) \left[\sum_{x_i} p_i(x_i) \log p_i(x_i) \right] - \\ &- \sum_{F \in \mathcal{F}} \sum_{x_F} p_F(x_F)(E_F(x_F) + \log p_F(x_F)) \end{aligned} \tag{3.22}$$

Loopy belief propagation have been successfully applied on many computer vision tasks with remarkable results [99, 100]. MAP inference can be performed by replacing the marginalization on Eq. 3.18 by a maximization, and updating the normalization function on Eq. 3.17 accordingly. Then the final MAP estimation is performed by finding the state of the variable x_i that maximizes its called *max-belief*:

$$\hat{x_i} = \underset{x_i}{\operatorname{argmax}}\ p_i(x_i), \quad \forall i \in \mathcal{V} \tag{3.23}$$

Despite the success of the algorithm on many tasks, in its general form it does not guarantee to converge to a fixed point, which for some problems it can lead to poor estimations of the marginals. In addition, the theoretical properties have not been well understood, and it is not clear how the algorithm performs so well on loopy graphs. Research on the convergence problems led to establish that fixed points from the BP algorithm correspond to extrema of the Bethe and Kikuchi free energy [101]. This motivated the the derivation of convergent extensions of the LBP algorithm [102] based on these principles, and many other variational methods were proposed to minimize these energies, such as the CCCP and UPS algorithms [103, 104].

3.5.3 Variational methods

Variational bayesian methods are a family of techniques used to compute an approximation to the exact probability distribution when exact estimation is not feasible. For a given probability distribution $p(x)$ defined by a PGM, a common interpretation of these methods relies on the approximation of the so-called free energy, which is defined as:

$$F(p) = E(p) - S(p) \tag{3.24}$$

where $E(p)$ corresponds to the energy, and $S(p)$ is the entropy of the model. The problem of finding the exact probability distribution $p(x)$ is reduced to minimize the function of $F(p)$, which for general PGMs, is defined as:

$$F(p) = -\sum_{c \in C} \sum_{x_c} p(x_c) \log \Psi_c(x_c) + \sum_{x} p(x) \log p(x) \tag{3.25}$$

where C is the set of cliques that compose the PGM and $\Psi_c(x_c)$ the potential function associated to this clique. The problem of computing this energy is that the estimation of the entropy term is also intractable in practice, since it sums over an exponential number of elements. Different variational methods diverge on how the entropy term is approximated. This problem have been addressed from two different perspectives, the called *mean-field* and *Kikuchi* approximations. In this section we define both approaches, with special emphasis on Kikuchi approximation, since we rely on it for the definition of some of the algorithms proposed on this thesis.

Kikuchi approximation

The Kikuchi approximation is based on estimating the entropy term as a combination of several marginal entropies as:

$$-S(p) = \sum_{x} p(x) \log p(x) = \sum_{\gamma \in \mathcal{R}} c_\gamma \sum_{x_\gamma} p(x_\gamma) \log p(x_\gamma) \tag{3.26}$$

where $\gamma \in \mathcal{R}$ denotes the so-called *regions* on which are defined a set of pseudo-marginals $p(x_\gamma)$. These regions are composed of a subset of variables from the model, which usually corresponds with the set of cliques C and its successive intersections. The term c_γ is the *Moebius* or *overcounting* numbers. The use of this approximation of the entropy term lead to an approximate probability distribution q on which the Kikuchi free energy is defined:

$$F_{Kikuchi}(q) = -\sum_{c \in C} \sum_{x_c} q(x_c) \log \Psi_c(x_c) + \sum_{\gamma \in \mathcal{R}} c_\gamma \sum_{x_\gamma} q(x_\gamma) \log q(x_\gamma) \tag{3.27}$$

As we stated in the previous section, it have been proved that fixed points of LBP correspond to extrema of the called *Bethe* free energy [102]. Bethe free energy is a particular case of the Kikuchi free energy, where the set of regions is defined in a particular way. We can see the parallelism between the first term on Eq. 3.22 and the expression of the Kikuchi free energy. In fact, an estimation of Z can be obtained by solving the following optimization problem:

$$-\log Z \approx \min_{q \in Q} F_{kikuchi}(q) \tag{3.28}$$

which is subjected to a set of constraints on the pseudo-marginals $q(x_\gamma)$ to ensure consistency and normalization of the probability distribution q. Q represents the family of distributions that satisfy this set of constraints. The constrained minimization problem can be solved by introducing Lagrange multipliers and solving the derived dual problem, which is convex and ensures to find the proper solution [101]. The solution to this problem for a particular configuration of the regions $\mathcal{R}$ lead to the definition of the *Generalized Belief Propagation* (GBP) algorithm, which is guaranteed to converge to a local minimum of the Kikuchi energy.

Mean Field approximation

The previous approach relies in the approximation of the entropy term as a sum of local entropies, in contrast, the mean field approximation face the minimization of the free energy by reducing the number of possible probability distributions to a set of tractable probability distributions $q \in \mathcal{Q}$. In this way, the problem is reduced to the following optimization:

$$q^* = \underset{q \in \mathcal{Q}}{\operatorname{argmin}}\ KLD(q(x)||p(x)) \tag{3.29}$$

where KLD is the Kullback-Leibler divergence between two probability distributions. Therefor, the objective of mean field methods is to find the distribution q that best approximates the original probability distribution. The KLD is defined as:

$$\begin{aligned} KLD(q(x)||p(x)) &= \sum_x q(x) \log \frac{q(x)}{p(x)} \\ &= \sum_x q(x) \log q(x) - \sum_x q(x) \log p(x) \\ &= -S(q) + \sum_{F \in \mathcal{F}} \sum_{x_F} q_F(x_F) E_p(x_F) + \log Z_p \end{aligned} \tag{3.30}$$

where $S(q)$ is the entropy of the distribution q, and $q_F(x_F)$ the marginal distribution over the variables in F. Note that the partition function Z_p do not depend on q, and therefore it is not necessary to compute in the minimization of the KLD. Once we solve the minimization and find q^*, we can approximate the set of marginals $q_F(x_F)$, and provide which is called the *mean field lower bound* of the true partition function as:

$$\log Z_p \geq -S(q) + \sum_{F \in \mathcal{F}} \sum_{x_F} q_F(x_F) E_p(x_F) \tag{3.31}$$

Different approaches to define the family of distributions $\mathcal{Q}$, and further generalizations of the mean field algorithm, have been tackled in the literature. Additional information can be found in the following references [105, 106].

There is a close relation between the well-known Expectation-Maximization algorithm and variational methods. Keeping in mind the essential aspects, both methods aim at estimating unknown probability distributions and are defined in terms of the KLD distance. In the next section we describe the Expectation Maximization algorithm in order to estimate the parameters of probability distributions defined on a set of latent variables.

3.6 Expectation-Maximization algorithm

In previous sections we describe a set of models and techniques widely used to model complex data distributions. The success of this type of techniques is possible due to the existence of advanced methods for parameter learning from a set of observations. However, there are tasks where the only data available for training these models is incomplete, and therefore, the parameter estimation becomes a difficult task. Although there are some attempts to perform robust parameter learning on missing data, most of these approaches are based in the same principles than the Expectation-Maximization algorithm. The Expectation-Maximization (EM) algorithm is a method that enables parameter estimation in probabilistic models with incomplete data. It is based in the maximum likelihood (ML) estimator, which is a method to estimate parameters in a probabilistic model. The EM algorithm generalize this concept to the case of incomplete data.

First, lets recall the definition of the ML estimation problem. We have a data set composed of N samples $\mathcal{X} = \{x_1, ...x_N\}$ that has a certain density function $p(x|\Theta)$ ruled by the set of parameters Θ. The likelihood estimate of the parameters given data is then defined as:

$$\mathcal{L}(\Theta|\mathcal{X}) = \prod_{i=1}^{N} p(x_i|\Theta) \tag{3.32}$$

the function describes how well the set of parameters explain the data. In a maximum-likelihood problem, the goal therefore is to to find the parameters $\hat{\Theta}$ that maximizes the $\mathcal{L}$. In practice, it is common to work with the logarithm of the likelihood function, called the log-likelihood, since it results more convenient to work with sums, and because the log-function is a strictly increasing function, which is also convenient in views of a maximization. The difficulty of this problem depends on the form of p. In the particular case of a Gaussian distribution $\Theta = (\mu, \sigma^2)$, the problem boils down to set derivatives of $\mathcal{L}$ to zero and solve for both parameters. However, this can not be so easy in other types of distribution and require more complex techniques.

The EM algorithm [107] allows to perform this maximization in the case of complex likelihood functions or when some data is missing. Following the previous definition, now we add to the observed data $\mathcal{X}$ the set of missing data $\mathcal{Y}$. The new log-likelihood function is then defined as $\mathcal{L}(\Theta|\mathcal{X}, \mathcal{Y}) = p(\mathcal{X}, \mathcal{Y}|\Theta)$. Again, the problem is to find the parameters Θ that rule the distribution p.

The EM algorithm consist of two iterative steps repeated until the convergence of the solution. First, the Expectation (E) step consist of the computation of the expected value of the log-likelihood function with respect to the unknown data given the observed data and the current parameter values as:

$$Q(\Theta|\Theta') = \mathrm{E}\left[\log p(\mathcal{X},\mathcal{Y}|\Theta)|\mathcal{X},\Theta'\right] \tag{3.33}$$

where Θ' are the current set of parameters. Then, in the Maximization (M) step the algorithm finds new values of Θ so that they maximize the value of the expectation in the previous step. This is:

$$\Theta = \underset{\Theta}{\operatorname{argmax}}\, Q(\Theta,\Theta') \tag{3.34}$$

The algorithm guarantees convergence to a local maximum of the log-likelihood function. Depending on the starting values the convergence can be to a local or global maximum of the function. This fact has promoted a variety of heuristic approaches to avoid that local maximums, and there are many works focused on the convergence of this algorithm [108, 109]. Here we present the general version of the EM algorithm. Depending on the application the algorithm can take several forms. In the following we present the derivation for the case of the parameter estimation of a Gaussian Mixture Model (GMM), one the the most common applications of the EM algorithm.

3.6.1 Gaussian Mixture Model estimation via EM

We tackle the particular case of the estimation of the parameters of a Mixture of Gaussian functions[110]. GMM are a very popular resource to model complex probabilistic density functions on continuous-valued feature vectors [111], and therefore are widely used in the pattern recognition community. The probability density function of a GMM is a wighted sum of M Gaussian functions defined as:

$$p(x|\Theta) = \sum_{k=1}^{M} w_k p_k(x|\theta_k) \tag{3.35}$$

where x represents the data, and $\Theta = \{w_1, ..., w_M, \theta_1, ..., \theta_M\}$. The weights satisfy that $\sum_{k=1}^{M} w_k = 1$, and $\theta_k = (\mu_k, \Sigma_k)$ for each component k of the mixture. Now, lets assume that there is another set of unobserved data $\mathcal{Y}$ that complements the observed data $\mathcal{X}$. The most common interpretation for $\mathcal{Y}$ is that contains the information of which component of the mixture has generated each sample x. Therefore, if we know the values of $\mathcal{Y}$ the log-likelihood function is defined as:

$$\log \mathcal{L}(\Theta|\mathcal{X},\mathcal{Y}) = \log(P(\mathcal{X},\mathcal{Y}|\Theta)) = \sum_{i=1}^{N} \log(w_{y_i} p_{y_i}(x_i|\theta_{y_i})) \tag{3.36}$$

The problem is that we do not know the distribution of the unobserved data. Since the EM computes the previous expected value with respect to this data, we must define an proper expression for it. Thus, we can think in an initialization of the parameters in Θ computed according to some criteria. This set of initial values is Θ'. Give this set of parameters, we can derive an expression for the set of hidden variables y in the model as:

$$p(y|\mathcal{X}, \Theta') = \prod_{i=1}^{N} p(y_i|x_i, \Theta') \tag{3.37}$$

where the probability for each sample is defined as:

$$p(y_i|x_i, \Theta') = \frac{w'_{y_i} p_{y_i}(x_i|\theta'_{y_i})}{\sum_{k=1}^{M} w'_k p_k(x_i|\theta'_k)} \tag{3.38}$$

Now, considering all this information, we can provide an expression for the function Q in Eq 3.33:

$$\begin{aligned} Q(\Theta, \Theta') &= \sum_{k=1}^{M}\sum_{i=1}^{N} \log(w_k p_k(x_i|\theta_k)) p(k|x_i, \Theta') \\ &= \sum_{k=1}^{M}\sum_{i=1}^{N} \log(w_k) p(k|x_i, \Theta') + \sum_{k=1}^{M}\sum_{i=1}^{N} \log(p_k(x_i|\theta_k)) p(k|x_i, \Theta') \end{aligned} \tag{3.39}$$

So, we have the expression corresponding to the E-step of the algorithm that computes the expected value of the log-likelihood function linked to initial set of parameters that defines the GMM. The next step consist of the estimation of new parameter values such that we maximize Q. This optimization can be done by setting partial derivatives of Q with respect to each type of parameter to zero, and solve for each of them. Since w_k is not related with the parameters in θ_k we can perform their maximization independently.

In the case of the expression for w_k we can introduce the Lagrange multiplier λ with the constraint that $\sum_k w_k = 1$. Solving the resulting equation gives:

$$w_k^{new} = \frac{1}{N} \sum_{i=1}^{N} p(k|x_i, \Theta') \tag{3.40}$$

Now we compute the expressions for the parameters of each Gaussian component $\theta_k = (\mu_k, \Sigma_k)$. For this generic case, we assume a mixture of d-dimensional Gaussian functions defined as:

$$p_k(x|\mu_k, \Sigma_k) = \frac{1}{(2\pi)^{d/2}|\Sigma_k|^{d/2}} e^{-\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k)} \tag{3.41}$$

In order to obtain update expressions for each parameter of the mixture, we combine Eq 3.41 in Eq 3.33 and solve for $\frac{\partial Q}{\partial \mu_k} = 0$ and $\frac{\partial Q}{\partial \Sigma_k} = 0$. The process provides the final equations for new parameter estimates according to the old set of parameters as:

$$\mu_k^{new} = \frac{\sum_{i=1}^{N} x_i p(k|x_i, \Theta')}{\sum_{i=1}^{N} (k|x_i, \Theta')} \tag{3.42}$$

$$\Sigma_k^{new} = \frac{\sum_{i=1}^{N} p(k|x_i, \Theta')(x_i - \mu_k^{new})(x_i - \mu_k^{new})^T}{\sum_{i=1}^{N} (k|x_i, \Theta')} \tag{3.43}$$

Chapter 4

Document layout analysis

In this chapter we present several methods developed for the task of document layout analysis. We propose several approaches to encode contextual information and prior knowledge about the problem. First, we present a set of features to encode relative location prior. Second, we propose an approach based on CRF for the segmentation of the document using different inference algorithms. Third, we present an statistical approach based on PGMs and the EM algorithm devised to structured documents. Last, we perform a comparative study against a method based 2D-PCFG for the analysis of structured documents.

4.1 Introduction

The ultimate aim of document analysis systems is to extract the information of interest from collections of documents. Within this process, layout analysis is the task in charge of identifying where this information is located based on the analysis of the document structure. Information of interest can be represented by multiple forms, making it necessary to incorporate to the analysis process a prior knowledge about the nature of the information to extract, and about any other circumstances that guide the identification process.

This prior knowledge can be defined according to contextual relations between entities. A contextual relation defines a set of circumstances which occur around a fact. For instance, if the objective is to identify signatures in a collection of letters, we can define that the region that contains the signature is more likely to be found at the bottom of the document, and it will be accompanied by other bigger text regions. Contextual relations between entities can be used to express different types of information. We distinguish between three main categories of relations: structural, hierarchical and semantic. First, structural relations express co-occurrences between entities and other characteristics, such as relative location or morphological restric-

tions. Second, hierarchical relations describe the distribution of each entity according to other more complex or simpler logical entities. Last, semantic relations establish conceptual links concerning the role of each entity. Several questions arise in what regards to the inclusion of contextual relations in our methods. (i) Which mechanism should we use to incorporate this information into the model?. (ii) Is this information provided, or does it have to be learned?.

In this chapter we present several answers to these questions in the ambit of the layout analysis task. We aim to model different types of contextual information from several sources, and encode it through different mechanisms, to guide the layout analysis process and improve the results on several tasks. We focus on the use of PGMs for this purpose. First, we tackle the problem of layout analysis as a classification task to find the most likely configuration of class labels for each pixel of the image. Our model is based on CRFs for MAP probability estimation. CRFs allow us to encode local contextual relations between variables, such as pairwise continuity constraints. Besides, we encode a set of structural relations between different classes of regions on a set of features. These features provide information about the relative location between the different classes of entities. Second, we perform a comparative study between our methods based on PGMs and the syntactic approach of 2D-PCFG. Context-free grammars are a common resource to encode structural and hierarchical relations, and the comparison against PGMs result of great interest for the field. We present a method based on this paradigm developed in collaboration with the PRHLT group from the Polytechnical University of Valencia. We define a grammar that accounts for a learned structure of a documents, and implement a derivation algorithm to find the most likely structure according to the grammar rules. Last, we propose a method for layout analysis of structured documents. We model structural relations and hierarchical dependencies between regions of interest by a Bayesian Network. Then, according to the modeled structure, we approximate the set of regions by a set of bi-dimensional Gaussian functions. We present an iterative algorithm based on the EM algorithm to compute maximum likelihood of the parameters, and use the previous set of relative location features as another source of input information to the model. We perform a thorough evaluation of the proposed methods on two particular collections of documents: a historical dataset composed of ancient structured documents, and a collection of non-structured contemporary documents. We summarize the set of contributions of this chapter in the following points:

1. **Relative location features for layout analysis.** We introduce relative location features to the document layout analysis process. On this features we can encode spatial relationships and semantic information from the set of entities. We combine them with other texture descriptors in order to improve the physical and logical analysis.

2. **Statistical method for layout analysis.** We present a statistical approach based on PGMs and the EM algorithm for layout analysis devised to structured documents. Our method learns the structure of the document, and perform the identification of the different entities in an iterative way. It can be applied to several types of layouts and it is flexible to light structure changes.

3. **Comparative study between PGMs and 2D-PCFG.** We perform a comparative study between syntactic approaches and probabilistic graphical models. We aim to evaluate both approaches on modeling the logical structure. For this purpose, we developed, in collaboration with the PRHLT group from the Polytechnical University of Valencia, a method based on bi-dimensional probabilistic context-free grammars for the segmentation of structured documents, and compare it against several PGMs and inference algorithms.

The rest of the chapter is organized as follows. In section 4.2 we describe the two layout analysis tasks addressed in this thesis. Then, in Section 4.3 we describe the set of features used by all the proposed models and how they are encoded within them. In Section 4.4 we describe the first proposed model based in CRFs. Section 4.5 describes the model using 2D-PCFG for layout analysis on structured documents. In Section 4.6 we describe the model proposed for layout segmentation based in fitting Gaussian functions. Then, in Section 4.7 we show the experimental evaluation and the comparative study between PGMs and 2D-PCFG. Last, we show the main conclusions of this chapter in Section 4.8.

4.2 Document collections

In this section we describe the two layout analysis tasks that we address in this chapter. Although the set of proposed methods can be generalized to other types of collections, we put special emphasis on these tasks, since they gather the main challenges of layout analysis, and will help to validate the set of contributions of our work. First, we present the Barcelona Historical Handwritten Marriages database. This is a collection of high interest for many historians, and the objective is to identify the different regions that compose a page in order to extract and analyze the information within them. Second, we present the PRImA database. A benchmark collection in the field of layout analysis.

4.2.1 BH2M: the Barcelona Historical Handwritten Marriages database

Marriage license books are handwritten documents that have been used in ecclesiastical institutions for centuries for registering marriages. Most of these books have a structure similar to an accounting book. Figure 4.1 shows an example of page of a marriage license book belonging to a collection of 291 books conserved at the Cathedral of Barcelona and conducted in the period from the year 1451 until 1905. The set of books includes approximately $550,000$ marriage licenses from 250 parishes. The pages in these books were orderly written, and although there are differences over the centuries, the layout in each page was quite rigid.

Every book is divided in two parts: the first part is an index of names, and the second part contains the marriage license records (see [112] for a more detailed

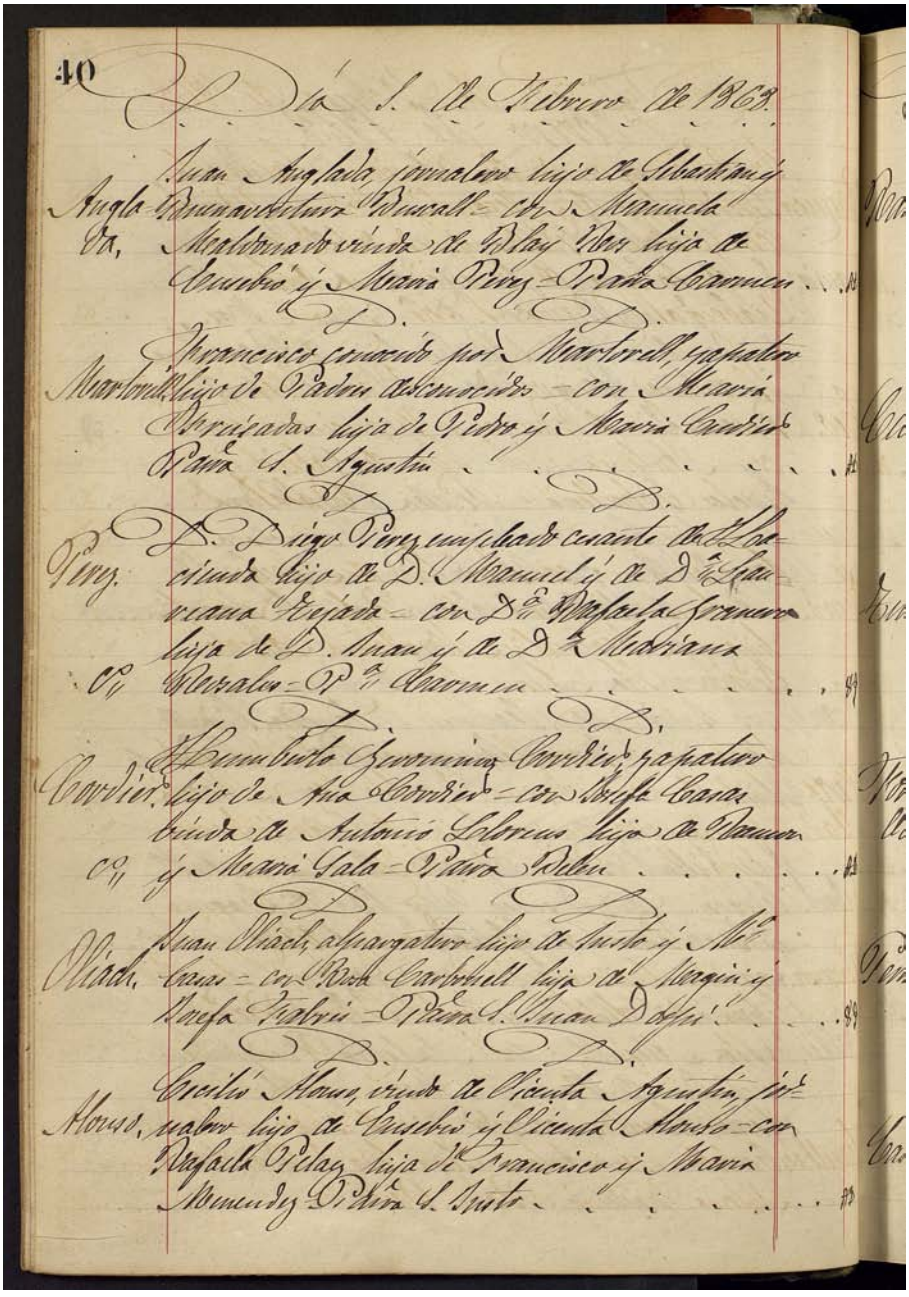


Figure 4.1: Example of page of a marriage license book from volume 208 containing six records.

description of this collection). In this thesis we focus on the analysis of the pages in the second part of the books. Each page is composed of several records, such that each one is associated with a marriage license. Each record has in turn a *husband name's block* (Figure 4.2.a), the *main block* (Figure 4.2.b), and the *tax block* (Figure 4.2.c). Note that the documents can have additional textual zones, like the date that can be seen at the beginning of the page (it can also appear in the middle of a page), and the two large calligraphic letters[1] that separate the consecutive records that were registered the same day. These additional zones were ignored in these experiments, i.e., they are considered like background because they were not considered relevant for subsequent transcription tasks. The process for creating the ground-truth requires marking the minimum rectangle containing the identified classes: *Body*, *Name* and *Tax*. All the pixels that did not belong to any of these regions are considered background.

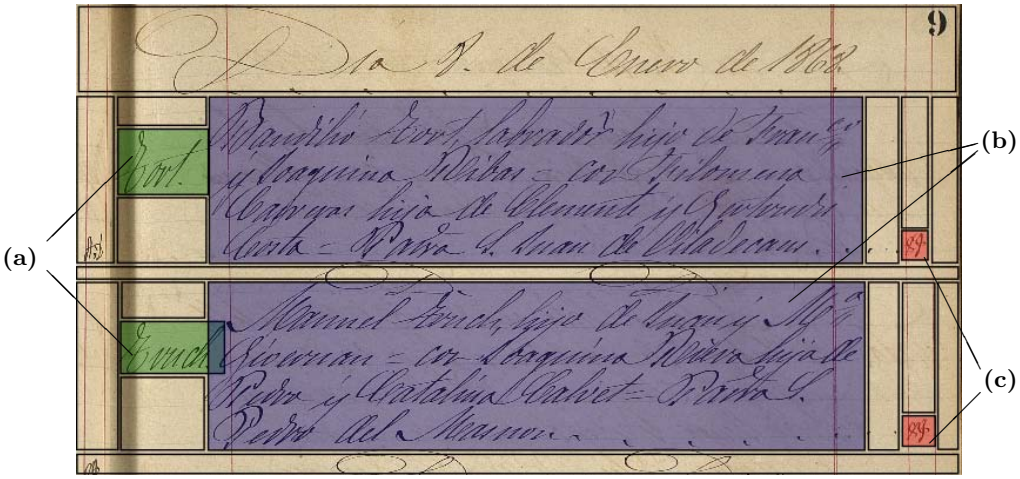


Figure 4.2: Example of the page segmentation problem for two records. Several background zones are considered and each record is composed of three parts: (a) Name (b) Body (c) Tax.

The final goal in these collection is to obtain the transcription of each marriage license. The correct segmentation of each page becomes a challenging task. The problem is to correctly isolate every record in a page, and to relate their corresponding parts, that is, the name, the body text and the tax associated with each entry. We focus on detecting the bounding boxes around the main parts of each record and relate them within the record logical entity. Note that a fine-grained detection of the frontiers of each zone would be ideal, but this is difficult because sometimes two zones overlap if rectangular bounding boxes are used. (see the lower record in Fig. 4.2). The problem of detecting the records can be stated as two different problems: first, to classify the textual zones into the previously mentioned classes of entities (*Background*, *Name*, *Body* and *Tax*); and second, to detect the complete set of records of each page.

[1]These letters are D. D. that is the abbreviation of "Dit dia" which means "The mentioned day".

Despite the great number of volumes in this dataset, currently only a few of them are being used on different tasks like handwriting recognition, word spotting, or layout analysis. A public database for the volume 69 of this collection is available and covers almost all the tasks of a document analysis pipeline (layout analysis, line segmentation, word segmentation, word spotting and handwriting recognition) [113]. However this database focus exclusively on the *Body* part of each record and do not include the other parts that compose them. For this reason, for the experiments reported in this chapter we focus on another book on the collection, concretely the volume 208. However, as the documents in all the volumes have the same structure, the proposed models could be applied to the remaining books equally.

4.2.2 PRImA Layout Analysis Dataset

The Pattern Recognition and Image Analysis dataset (PRImA) [1] is a benchmark dataset created for the evaluation of layout analysis (physical and logical) method. The dataset is composed of a set of realistic documents covering a wide variety of document layouts, including most of the important challenges in layout analysis. The collection is focus on pages from magazines and technical/scientific publications since they are most likely to be focus of digitization and automatic processing tasks. The set of magazines scans come from several publications about general news, business and technology information which contain a mixture of simple layouts, as Manhattan structures and newspaper-like structures, as well as complex layouts as non-Manhattan layouts, different font sizes and graphical content. We show several samples of this collection in Figure 4.3. In the case of technical documents they usually present common layouts from journals and conference proceedings, with both simple and complex layouts as well.

Documents in this dataset are in general less structured than the BH2M dataset, so we use it as an example for non-structured documents in order to validate the effect of the proposed resources to analyze the structure of a document. The entire dataset is composed 305 ground-truthed images whose ground-truth is described and stored according to the PAGE (Page Analysis and Ground truth Elements) framework [114]. For the proposed experiments we use a subset of the publicly available PRImA dataset, in order to match with the subset used for the ICDAR 2009 and 2011 page segmentation competitions [1]. Besides, we focus on the two classes of entities *text* and *images* without taking into account any logical label.

Figure 4.3: Sample images from the PRImA dataset.

4.3 Text classification features

In this section we describe the set of features used for the task of layout analysis. We describe two different feature descriptors used to capture discriminant information for both physical and logical layout analysis processes. On the one hand we describe a descriptor based in the Gabor transform to extract texture information from the document. This feature provide discriminant information to differentiate between text classes and between text and background elements. On the other hand we define the set of features that encode spatial relations between entities. From now onward, we refer to them as Relative Location Features (RLF). Both sets of features are applied to collections of structured and non-structured documents in order to evaluate their contribution to the task.

4.3.1 Document representation

Although our objective is to label each pixel with the most likely class, modeling the problem at pixel level is not always possible or the best choice. On the one hand there is the problem of model complexity. High resolution images are composed by several millions of pixels. If we build a graphical model composed of such amount of variables, computing inference and parameter learning can be intractable even for approximate methods. This also affects to the feature extraction process, since it can be difficult in terms of complexity. On the other hand, there is the problem of the type of information captured at pixel level. Feature extraction at pixel-level may not encode context information that can be useful for the kind of tasks that we face.

A common approach to overcome these problems is to model the image as a set of superpixels. A superpixel is a cluster of adjacent pixels that are grouped according to different criteria. Depending on the task they can be grouped according to texture similarity, color or other visual feature. Besides, they can be also grouped according to its semantic information. However, since there is no general superpixel generation method that works for every kind of images and scenarios, the process of dividing the image in these groups has to be done attending to the task requirements. In the case of document images the superpixel set can be set regarding to its texture. As a result of this process, image is divided into a set of superpixels containing text and background regions. However, common degradation in the case of historical documents, or any other source of noise can result in an superpixels over-estimation that may not be logical with the document content, and therefore, the final labeling may not be accurate.

For the proposed methods in this chapter we choose a representation for our document images based on a set of S regular cells of $N \times N$ pixels forming a grid layer over the pixels. Using this representation we reduce the computational cost of our methods, and at the same time we preserve the integrity of the image content and capture local contextual information by analyzing the group of cell pixels.

4.3.2 Texture features

Texture descriptors are regularly used in segmentation tasks to extract information about the spatial arrangement of color or intensities from areas of an image. One common descriptor is the Gabor filter. A Gabor filter is a linear filter based in the Gabor transform commonly used for texture analysis. It is defined by a sinusoidal wave of complex values modulated by an exponential function [115]. This exponential function is a Gaussian function centered in the origin of coordinates, with a parameter controlling the size of the function support[2]. In the frequency space, the Gabor transform is also defined by a Gaussian function, centered in the frequency f_0 and support inversely proportional to frequency f_0. Furthermore, in images the filter support has an elliptical shape tuned by three parameters γ, η and θ:

[2]We refer as support of a Gaussian function the region enclosing 99% of the energy.

$$\begin{aligned}\psi(x,y;f,\theta) &= \frac{f_0^2}{\pi\gamma\eta}e^{-\left(\frac{f^2}{\gamma^2}x'^2+\frac{f^2}{\eta^2}y'^2\right)}e^{i2\pi f x'}\\ x' &= x\cos\theta + y\sin\theta,\\ y' &= -x\sin\theta + y\cos\theta\end{aligned} \tag{4.1}$$

It is well-known that the Fourier transform of a Gaussian function is again a Gaussian function. In addition, if we scale the support of Gabor filters by a factor of k^{-m}, the support of their Fourier transform are proportional to k^m. In particular, given the definition of Gabor filters in Eq. (4.1), the support of Gabor filters in the spatial domain are ellipses with axis proportional to $\frac{\gamma}{f_{max}}k^m$ and $\frac{\eta}{f_{max}}k^m$. The values of η and γ are obtained according to the number of orientations n, the scaling factor k, and the overlapping degree q of filters in the Fourier space as:

$$\gamma = \frac{k-1}{k+1}\frac{\sqrt{-\log q}}{\pi}; \quad \eta = \frac{\sqrt{-\log q}}{\pi\tan\frac{\pi}{2n}}$$

The usual application of this descriptor to texture analysis in document images is to compute a bank of filters of this type for several orientations and signal frequencies. Some works have studied the best way to select the set of orientations and frequencies [116]. According to the recommendations in that work, we consider an exponential spacing of m frequencies: $f_l = k^{-l}f_{max}$, with $l = \{0,...,m-1\}$, where f_{max} is the maximum frequency fixed, f_l the lth frequency and k the scaling factor $k > 1$. Regarding the orientation angle, the most common approach is to set orientations uniformly as $\theta_b = b2\pi/n$, with $b = \{0,...n-1\}$, where n is the number of orientation to be considered.

Despite of the advantages of Gabor filters regarding the information captured and its flexibility there are some considerations to take into account. The main difficulty in the use of these filters is the computational cost of computing the responses for all the filters in the bank. The involved Fourier transform, for example, is one of the most expensive process to be computed in the computation of a Gabor response. A fast implementation of a bank of this type of filters was proposed in [117]. As a result of the application of this method on a pixel, we obtain a feature vector g of dimensions $n \times m$, which covers almost all the spectrum of frequencies up to the highest one f_{max}. Here, the feature $g_{l,b}$ is given by the filter response of frequency l and orientation b on a particular document pixel.

Recent studies suggest that the analysis of the relationships between filter responses can discriminate between different objects or classes in a classification task [118]. We choose a Gaussian mixture model (GMM) in order to estimate a likelihood probability $p(g \mid c)$ for of each possible class c given the descriptor g computed at pixel level for each pixel p. According to this mixture we are able to estimate the probability $p(c \mid s)$ for a given cell of the image $s \in S$ for each possible class c as:

$$p(c \mid s) = \frac{\sum_{p \in s} p(g_p \mid c) p(c)}{\sum_c \sum_{p \in s} p(g_p \mid c)\, p(c)} \tag{4.2}$$

where $p(c)$ is the *prior* probability for the class c. Next we explain the details of the GMM model used for this purpose.

GMM-based classifier GMM are widely used to approximate complex probabilistic density functions on continuous-valued feature vectors [111]. The resulting pdf is defined as the result of a weighted sum of M Gaussian components as:

$$p(g|\mu_i, \Sigma_i) = \sum_{i=1}^{M} w_i q(g|\mu_i, \Sigma_i) \tag{4.3}$$

where g is the feature vector described above and w_i corresponds with the weight of the Gaussian component i defined by the parameters μ_i, Σ_i. These weights are restricted to satisfy $\sum_{i=1}^{M} w_i = 1$. Each density function $q(g|\mu_i, \Sigma_i)$ correspond to one of the M Gaussian components and is given by the expression:

$$q(g|\mu_i, \Sigma_i) = \frac{1}{\sqrt{2\pi|\Sigma_i|}} \exp\left\{-\frac{1}{2}(g-\mu_i)'\Sigma_i^{-1}(g-\mu_i)\right\} \tag{4.4}$$

where μ_i and Σ_i are the mean and covariance matrix of the component i, repectively. Therefore, the final set of parameters that defines the mixture is: $\Theta_{GMM} = \{w_1, \mu_1, \Sigma_1, ..., w_M, \mu_M, \Sigma_M\}$. We perform the estimation of this values by means of the common Expectation-Maximization (EM) process for GMM as was explained in Section 3.6. Thus, given this mixture we can compute now the probability for each pixel $p_s \in s$ given its descriptor g_s for each considered class.

4.3.3 Relative location features

So far, we described how PGMs can encode prior information about the problem by modeling the relations in the graph structure or by the definition of factor functions that encode this information. However, pairwise relationships are limited to the relations between variables in a defined neighborhood, and the use of other high-level configurations implies an additional computational effort. Therefore, the remaining mechanism to encode contextual information and prior knowledge is through the set of features. In this section we define RLF and how they are used for our document layout analysis task.

The concept of relative location prior was introduced in [119] to encode this prior information in order to capture inter-class spatial relationships. That work was devised to segmentation of real scene images where exists a previous knowledge about

the most common location and about the semantic of the classes. We propose to apply this same approach to the analysis of the layout of document images. There are some tasks where a previous knowledge about the semantics of the classes can be used to guide the segmentation process, for instance, if we want to discriminate between the classes *body* and *tax* (see Figure 4.2), we can restrict the system to find *tax* pixels only in areas above and at the right side from *body* pixels. There are also some cases where we do not have a semantic knowledge but we know from the definition of the problem that there is a co-occurrence relation between the classes. In administrative documents, for instance, we may know that the signature appears always in some place at the bottom of the document.

To implement this set of features we follow several steps. First, we have to compute a set of probability maps for each pair of classes. These maps encode the probability of finding elements from one class in a particular location according to its relative position to elements of another class. Second, we need an initial estimation of the most likely class given a particular pixel or region. This step is usually performed by another classification process. The logic of this step relies is to have a starting point on which apply the relative location prior, and then correct this initial labeling according to it. Finally, combining these two steps we can define the set of RLF.

Probability maps Probability maps are a mechanism that allow us to represent the location where is most likely to find elements of one class with respect to the location of an element of another class. In other words, given two classes c_i and c_j, and one pixel from each class p_i and p_j, we say that the map $M_{c_i|c_j}(u, v)$ encodes the probability that the pixel p_i, with a relative displacement (u, v) from p_j, belongs to the class c_i as:

$$M_{c_i|c_j}(u, v) = P(c_i|p_i, p_j, c_j) \tag{4.5}$$

We learn a probability map for each pair of classes, including the map of one class respect to itself. Besides, the maps are not reciprocal, i.e. $M_{c_i|c_j}(u, v) \neq M_{c_j|c_i}(u, v)$, so that the required number of maps to compute will be, considering C classes, is C^2 maps.

The learning process of the probability maps is based on counting the number of times that each possible displacement between pixels from two different classes is given within the training set. More concretely, the map $M_{c_i|c_j}$ will be computed by calculating all the displacements between the pixels from the class c_i and the pixels from the class c_j. For example, let the pixels $p_i = (x_i, y_i)$ y $p_j = (x_j, y_j)$, the displacement between them is $(u, v) = (x_i - x_j, y_i - y_j)$. Once calculate this displacement, the value of the map $M_{c_i|c_j}$ at the position (u, v) is increased by one. Repeating the same calculation for each pair of pixels from the classes c_i and c_j and normalizing we obtain a region where the majority of the votes are grouped, which correspond to the most probably area. Going back to the document example in Figure 4.2, we show the probability map $M_{body|name}$ in Figure 4.4.

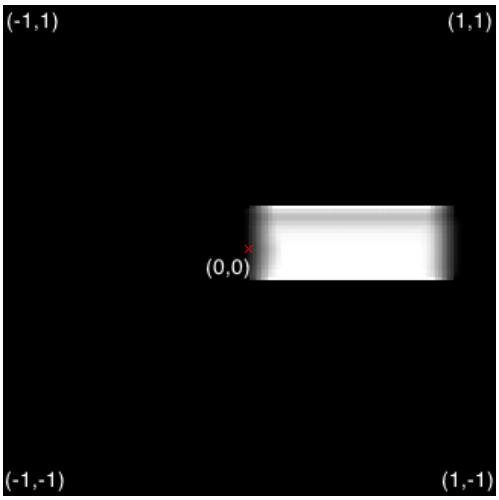


Figure 4.4: Probability map $M_{body|name}$ for the BH2M dataset.

Computing all the possible displacements between pixels of one class (original) with respect to another (reference) may result computationally expensive. In order to reduce the number of calculations we substitute the pixels of the reference class by the centroids of the cells according to our document representation. Since we still consider all the pixels of the original class, this process does not substantially affect to the global result of the map.

Encoding Relative Location Features We start from a image representation consisting in our set of cells S, and we calculate an initial class prediction $\hat{c}_s$ for each of them with a probability $p(\hat{c}_s|s)$ given by the GMM classifier in (4.2). The computation of the RLF is conducted as follows: each cell in the image will cast a total of C votes over each of the other cells. These votes can be understood as how likely is to assign a particular label to each of these cells, according to the initial labelling assigned to them, and the information provided by the probability maps. Thus, considering K cells, each cell recibes $C(K-1)$ votes from the rest.

In order to get more profit from the RLF, and following the approach in [119], we split the process to compute two different RLF called v_{other} and v_{self}. The first one gather the votes from cells with different initial class label, and the second one the votes that come from cells with equal labels. That division will allow us to assign different weights to each feature. The implementation of these features is formulated as:

$$
\begin{aligned}
v_c^{other}(s) &= \sum_{j \neq i : c_i \neq \hat{c}_j} \alpha_j M_{c_i|\hat{c}_j}(x_i - x_j, y_i - y_j) \\
v_c^{self}(s) &= \sum_{j \neq i : c_i = \hat{c}_j} \alpha_j M_{c_i|\hat{c}_j}(x_i - x_j, y_i - y_j)
\end{aligned}
\tag{4.6}
$$

where (x_i, y_i) and (x_j, y_j) are the coordinates from the centroids of the cells s_i and s_j, respectively. Votes are also weighted by $\alpha_j = p(\hat{c}_{s_j}|s_j)$, which is estimated in (4.2). We appreciate the importance of a confident initial classification.

As a last stage, we normalize the set of features in order to define a proper probability distribution. In this case we normalize to ensure that $\sum_{c=1}^{C} v_{c_s}^{other}(s) = 1$, and respectively for the values of $v_{c_s}^{self}$.

Logistic Regression-based classifier Now we show how to include the defined set of features into a model as local information of a cell. We define the probability function $p_{final}(c_s \mid s)$ that includes all the information gathered up to this point. That includes both the new Relative Location Features and the initial class prediction based on the Gaussian Mixture Model from Gabor features. More concretely, Relative Location Features are linearly combined with the texture-based features as follows:

$$
\begin{aligned}
P_{final}(c_s \mid s) = w^{tex} \log P(c_s|s) + \\
+ w_{c_s}^{other} \log v_{c_s}^{other}(s) + w_{c_s}^{self} \log v_{c_s}^{self}(s),
\end{aligned}
\tag{4.7}
$$

where the weights w^{app}, $w_{c_s}^{other}$ and $w_{c_s}^{self}$ are learned using a logistic regression model from labeled data, and $p(c_s|s)$ is the one defined in (4.2).

4.4 Conditional Random Fields

In this section we describe the first method proposed for the task of layout analysis. We tackle the problem as a labeling task where we want to compute the most likely configuration of the class labels for each pixel on the image. Then, we extract the homogeneous regions that share the same label to obtain the final configuration of the regions from the page.

This process provides the physical identification of the set of regions. In order to encode its logical meaning we resort to the set of Relative Location Features defined on the previous section. Using these features we obtain two main benefits: reinforce the classification process, and define a more complex set of labels for a given problem. For instance, for the task of layout analysis on the BH2M dataset, using only texture descriptors probably is not possible to discriminate between the different entities that compose a record further than detecting text regions, since the visual appearance and

text morphology is quite similar. However, introducing relative location prior, what was before a regular text region, can be now categorized according to its semantic role.

We rely on Conditional Random Fields for this task. Using this model, the objective is to compute MAP probability in order to find the configuration of the class labels for each pixel that maximizes the probability defined by the CRF. We use the document representation described in Section 4.3.1 for efficiency purpose. Therefore, we wan to obtain the set of labels for each cell on the image.

We define a grid-like CRF according to the cell representation composed of two kind of random variables. First, we define the set of observed variables X, which correspond with each of the cells. Second, we define a set of hidden variables Y linked to each of the observations, which represent the set of class labels. The result is an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where vertexes in $\mathcal{V}$ are the variables, and $\mathcal{E}$ is the set of edges of the graph. We also define two types of edges in $\mathcal{E}$. First, we have edges connecting each observed variable with its corresponding label. Second, we have edges connecting the set of hidden variables representing the level of interaction between them. For the proposed model, we consider a set of pairwise relationships in a 4-neighborhood of each cell. Thus, our MAP probability problem is formulated as:

$$\hat{Y} = \underset{Y}{\operatorname{argmax}}\, p(Y \mid X) \tag{4.8}$$

where $p(Y \mid X)$ is the probability of the CRF. As we described in the previous chapter, we can model this MAP problem in terms of energy minimization as:

$$P(Y \mid X) = \frac{1}{Z} \exp \left\{ \sum_{s} \psi(x_s, y_s) + \sum_{(s,k)\in\varepsilon} \phi(y_s, y_k) \right\} \tag{4.9}$$

where Z is the corresponding partition function. The term $\psi(x_s, y_s)$ represents local potentials of each cell s, which in our case is how likely is to assign the label y_s to the cell e_s, and $\phi(y_j, y_i)$ represent the pairwise potentials, which defines the penalty of assigning labels y_s, y_k to neighbor cells i, j.

We propose two approaches to define the local term $\psi(x_s, y_s)$ according to the described set of features. In the first place, we use the texture-based descriptor encoded by the GMM as we define in Eq. (4.2). In the second place, we use the combination of the previous descriptor with the RLF as we show Eq. (4.7). Analyzing the role of both set of features, we can validate the contribution of the RLF to the task.

In what regards to the pairwise term we propose a pairwise function that takes into account the knowledge about the problem. We define a set of parameters for each possible pair of classes. These parameters are learned according to the observations in the training set.

4.4.1 Inference and parameter learning

In the previous chapter we described several methods for the task of computing inference and parameter learning on PGM. For this work we aim to compare several of these methods in order to analyze their effect in a real problem. For the presented model we propose to compare three main algorithms from each family of methods. First, we use the $\alpha-\beta-swap$ version of Graph-Cut algorithm, which uses the MAP probability problem definition as an energy minimization process. In this case we use the implementation released in [4]. We define the pairwise potential using a Potts-based model where $\phi(y_s, y_k) = T_{y_s,y_k} \cdot \pi(y_s \neq y_k)$, where the factor $\pi(\cdot)$ is 1 if the condition is true and 0 if is false, and T_{y_s,y_k} is the cost associated to each situation. We use the same penalty cost for each pair of classes. The use of this function must favors the pixels included on the same regions to be labeled with the same label, while obtaining better defined boundaries between regions. Second, we use LBP as representative of the message-passing algorithms. For this algorithm we use the same pairwise function described above, but in this case we fix the value of T_{y_s,y_k} according to the prevalence on the training set of each pair of classes. This function is more strict in the labeling of the regions, since pairs of non-observed classes are penalized. Last, we use the ICM algorithm as a simple example of variational methods based in the search paradigm. We use the UGM toolkit [120] that includes implementations for both parameter learning and MAP estimation of these two approaches, among others.

4.5 2D probabilistic context-free grammars

In this section we present the method based on 2D-PCFG for the task of layout analysis. A context-free model is a natural way to model both the horizontal and vertical context of the problem, where there are dependencies among rows, columns and 2D regions. This type of grammars are able to represent efficiently contextual relations that are relevant for the task of document segmentation. Basically, the problem is reduced to finding the most likely parse tree t for a given document structure.

A Context-Free Grammar (CFG) G is a tuple (N, Σ, S, R), where N is a finite set of non-terminal symbols, Σ is a finite set of terminal symbols ($N \cap \Sigma = \emptyset$), $S \in N$ is the start symbol of the grammar, and R is a finite set of rules: $A \to \alpha$, $A \in N$, $\alpha \in (N \cup \Sigma)^+$.

A Probabilistic Context-Free Grammar (PCFG) $\mathcal{G}_s$ is defined as a pair (G, P), where G is a CFG and $P : R \to]0, 1]$ is a probability function of rule application, i.e. $\forall A \in N : \sum_{i=1}^{n_A} P(A \to \alpha_i) = 1$; where n_A is the number of rules associated with non-terminal symbol A. This type of grammars can be represented in Chomsky Normal Form (CNF) resulting in only two types of productions: binary rules $A \to BC$ and terminal rules $A \to c$ (where $A, B, C \in N$ and $c \in \Sigma$).

Bidimensional Probabilistic Context-Free Grammar (2D-PCFG) are a generalization of PCFG that are able to deal with bidimensional matrices. In this extension,

nonterminal symbols account for 2D regions. The binary rules of a 2D-PCFG have an additional parameter $r \in \{\mathrm{H}, \mathrm{V}\}$ that describes a spatial relation: horizontal concatenation (H) or vertical concatenation (V). Given a rule $A \xrightarrow{r} B\ C$, the combined subproblems B and C must be arranged according to the spatial relation constraint, i.e., horizontally adjacent and same height for $r = \mathrm{H}$ and vertically adjacent and same width for $r = \mathrm{V}$. This extension is enough in order to apply this framework to the kind of problems we are dealing with. The segmentation of a document image can be obtained as the most likely derivation given a 2D-PCFG, such that the region that defines the input image is recursively divided either vertically or horizontally into smaller rectangular regions.

Parsing Algorithm

Given a page image, the problem is to obtain the most likely parsing according to a 2D-PCFG. For this purpose, the input page is considered as a bidimensional matrix I with dimensions $w \times h$ and each cell of the matrix can be either a pixel or a cell of $d \times d$ pixels. Then, we define an extension of the well-known CYK algorithm to account for bidimensional structures. We have basically extended the algorithm described in [121] to include the probabilistic information of our model.

The CYK algorithm for 2D-PCFG is essentially a *dynamic programming* method, which fills in a parsing table $\mathcal{T}$. Following a notation very similar to [122], each element of $\mathcal{T}$ is a probabilistic nonterminal vector, where their components are defined as:

$$\mathcal{T}_{(x,y),(x+1,y+1)}[A] = \hat{P}(A \Rightarrow z_{(x,y),(x+1,y+1)}) \tag{4.10}$$

$$\mathcal{T}_{(x,y),(x+i,y+j)}[A] = \hat{P}(A \overset{+}{\Rightarrow} z_{(x,y),(x+i,y+j)}) \tag{4.11}$$

Each region $z_{(x,y),(x+i,y+j)}$ is defined as a rectangle delimited by its top-left corner (x, y) and its bottom-right corner $(x+i, y+j)$. We denote $\ell_{i\times j} = \ell(z_{(x,y),(x+i,y+j)})$ as the size $(i \times j)$ of the subproblem associated with a region $z_{(x,y),(x+i,y+j)}$. The probabilities $\hat{P}$ represent the probability of the most likely derivation from nonterminal A resulting in the region z.

If the size of the subproblem is larger than 1×1, then there exists some binary rule ($A \xrightarrow{r} B\ C$, with $B, C \in N$, and $r \in \{H, V\}$) and some split point k such that, in a similar way to [122], we can divide the problem in two subproblems:

$$\begin{aligned}\hat{P}(A \overset{+}{\Rightarrow} z_{(x,y),(x+i,y+j)}) &= P(\ell_{i\times j} \mid A) \max_{B,C} \{ \\ &\max_{1\le k<i} P(A \xrightarrow{H} B\,C)\ \hat{P}(B \overset{+}{\Rightarrow} z_{(x,y),(x+k,y+j)})\ \hat{P}(C \overset{+}{\Rightarrow} z_{(x+k,y),(x+i,y+j)})\ , \\ &\max_{1\le k<j} P(A \xrightarrow{V} B\,C)\ \hat{P}(B \overset{+}{\Rightarrow} z_{(x,y),(x+i,y+k)})\ \hat{P}(C \overset{+}{\Rightarrow} z_{(x,y+k),(x+i,y+j)})\ \}\end{aligned} \tag{4.12}$$

where a new hypothesis is computed from two smaller subproblems, such that the probability is maximized for every possible vertical and horizontal decomposition resulting in the region $z_{(x,y),(x+i,y+j)}$. It should be noted that the 2D-PCFG provides

syntactic and spatial constraints $P(A \xrightarrow{\mathrm{r}} B\,C)$, and we have also included the probability $P(\ell_{i\times j} \mid A)$ that a nonterminal A accounts for a problem of size $i \times j$.

The probability $P(\ell_{i\times j} \mid A)$ has two effects on the parsing process. First, it helps to model the spatial relations among every part of a given page because there is a specific nonterminal for each zone of interest. For instance, this can be seen in Figure 4.2 where the size of the background region on top of the page will be different from the size of the background zone over a tax region. Furthermore, many unlikely hypotheses are pruned during the parsing process due to its size information, hence, it speeds up the algorithm.

Considering the definition of the matrix parsing table $\mathcal{T}$ (Eq. (4.11)), the expression of the Eq. (4.12) can be rewritten to obtain the general term of the parsing algorithm. Thus, for all i and j, $2 \leq i \leq w,\ 2 \leq j \leq h$, we have:

$$\begin{aligned}
\mathcal{T}_{(x,y),(x+i,y+j)}[A] \;&=\; P(\ell_{i\times j} \mid A) \;\max_{B,C}\; \{ \\
&\max_{1\leq k<i}\; P(A \xrightarrow{\mathrm{H}} B\,C)\; \mathcal{T}_{(x,y),(x+k,y+j)}[B]\; \mathcal{T}_{(x+k,y),(x+i,y+j)}[C] \;\;, \\
&\max_{1\leq k<j}\; P(A \xrightarrow{\mathrm{V}} B\,C)\; \mathcal{T}_{(x,y)(x+i,y+k)}[B]\; \mathcal{T}_{(x,y+k),(x+i,y+j)}[C] \;\;\}
\end{aligned}$$

For subproblems of size equal to 1×1 and taking into account the definition of Eq. (4.10), the derivation probability of a single cell (size region equal to 1×1) can be marginalized according to the class label (terminal) c. Given that we need to calculate the most likely parsing, we can approximate the sum by a maximization, and considering some other usual assumptions the probability of the derivation of a single cell is:

$$\hat{P}(A \Rightarrow z_{(x,y),(x+1,y+1)}) \approx P(\ell_{1\times 1} \mid A) \;\max_{c}\; P(A \to c)\; P(c \mid z) \tag{4.13}$$

where $P(\ell_{1\times 1} \mid A)$ is the probability that nonterminal A derives a subproblem of size 1×1; $P(c \mid z)$ represents the probability that a cell (region) z belongs to class c, and $P(A \to c)$ is the probability of a terminal rule for terminal (class) c.

Taking into account the matrix $\mathcal{T}$ (Eq. (4.10)), we can rewrite the expression of Eq. (4.13) to obtain the initialization term of the parsing algorithm. Thus, for each region z of size 1×1, we have:

$$\mathcal{T}_{(x,y),(x+1,y+1)}[A] \;=\; P(\ell_{1\times 1} \mid A) \;\max_{c}\; P(A \to c)\; P(c \mid z_{(x,y),(x+1,y+1)}) \tag{4.14}$$

Finally, the most likely parsing of the full input page is obtained in $\mathcal{T}_{(0,0),(w,h)}[S]$ such that S is the start symbol of the 2D-PCFG. It is important to notice that all the probability distributions involved in the parsing process can be learnt from labeled data. The time complexity of the algorithm is $O(w^3h^3|R|)$ and the spatial complexity is $O(w^2h^2)$.

Note that an inconvenient of 2D-PCFG is the need of training data to learn the probabilities that drive the grammar. In the case of its application to document

segmentation this means that we can only learn this probabilities from documents with a fixed structure or limited number of possible layouts. For a new document whose structure has not been learned and differ from the observations, the grammar will try to fit it to the most likely according to the learned structures, which can result in inaccurate segmentation results.

Model Estimation

The model based on 2D-PCFG for parsing structured documents has, in turn, some probabilistic distributions that need to be learnt. First, the probability $P(c \mid s)$ that a certain cell s of the image belongs to class c is described in Section 4.3. We use the same two approaches to this probability than for the CRF-based model. This two approaches are the texture-based descriptor encoded by the GMM and defined in Eq. (4.2), and the combination with the RLF described in Eq. (4.7).

There are two additional distributions that we have to estimate: the probabilities $P(\ell_{i\times j} \mid A)$ and the probability of the rules of the grammar $P(A \to \alpha)$ from Eq. (4.12). In order to learn automatically these distributions, we performed a forced recognition of the training set. Given a certain document, the forced recognition was carried out by providing the probability $P(c \mid s)$ using the ground-truth information. Concretely, for each cell s belonging to class c^* we set $P(c^* \mid s) = 1$ and $P(c \mid s) = 0$, $\forall c \neq c^*$. The remaining distributions were considered equiprobable.

Hence, we obtain for each document the best parsing according to the 2D-PCFG model. On one hand, the probability distribution of the size for each non-terminal A was estimated according to the occurrences in the forced recognition of the training set as

$$P(\ell_{i\times j} \mid A) = \frac{n(A_{i\times j})}{n(A)} \tag{4.15}$$

such that $n(A_{i\times j})$ is the number of times that non-terminal A accounts for a region of size $i \times j$ in the training set, and $n(A)$ the total number times that non-terminal A accounted for a region of any size.

On the other hand, the probabilities of the rules of the grammar are estimated using the set of derivation trees obtained from the forced recognition of the training set as:

$$\begin{aligned} P(A \to c) &= \frac{n(A \to c)}{n(A)} \\ P(A \xrightarrow{r} B\,C) &= \frac{n(A \xrightarrow{r} B\,C)}{n(A)} \end{aligned} \tag{4.16}$$

where each rule probability is computed using the number of times that the rule

was used in the training set, normalized by the total number of rules with non-terminal A as left-hand symbol $n(A)$. In order to make the model able to account for unseen events, after these distributions were estimated, we also smoothed them by setting a minimum probability threshold.

4.5.1 Application to the BH2M database

Now we describe how the previous method was applied to the particular task of the segmentation of the records on the BH2M database. We describe the set of grammar rules defined to account for the particularities of this collection. We define two different sets of grammar rules:

- Cell combinations:

  These rules denote the horizontal and vertical concatenation of sub-regions in order to create larger regions of the terminal classes.

BackGround	$\longrightarrow$	Bg	
Name	$\longrightarrow$	Nm	
Body	$\longrightarrow$	Bd	
Tax	$\longrightarrow$	Tx	
BackGround	$\xrightarrow{\text{H}}$	BackGround	BackGround
BackGround	$\xrightarrow{\text{V}}$	BackGround	BackGround
Name	$\xrightarrow{\text{H}}$	Name	Name
Name	$\xrightarrow{\text{V}}$	Name	Name
Body	$\xrightarrow{\text{H}}$	Body	Body
Body	$\xrightarrow{\text{V}}$	Body	Body
Tax	$\xrightarrow{\text{H}}$	Tax	Tax
Tax	$\xrightarrow{\text{V}}$	Tax	Tax

- Document structure:

  These rules define the structure of the document given the set of detected regions and its geometrical information: For a set of regions with a certain probability of , and their probabilities to be some of the

Page	$\xrightarrow{\text{V}}$	BackGround	FullRegSeqBg
FullRegSeqBg	$\xrightarrow{\text{V}}$	FullRegSeq	BackGround
FullRegSeqBg	$\xrightarrow{\text{V}}$	FullReg	BackGround
FullRegSeq	$\xrightarrow{\text{V-RR}}$	FullRegSeq	FullReg
FullRegSeq	$\xrightarrow{\text{V-RR}}$	FullReg	FullReg
FullRegSeq	$\xrightarrow{\text{H}}$	BackGround	RegisterBg
RegisterBg	$\xrightarrow{\text{H}}$	Register	BackGround

$$
\begin{array}{llll}
\text{Register} & \xrightarrow{\text{H-NB}} & \text{FullName} & \text{BodyTax} \\
\text{FullName} & \xrightarrow{\text{H}} & \text{BackGround} & \text{NameBg} \\
\text{NameBg} & \xrightarrow{\text{H}} & \text{Name} & \text{BackGround} \\
\text{BodyTax} & \xrightarrow{\text{B-T}} & \text{Body} & \text{FullTax} \\
\text{FullTax} & \xrightarrow{\text{R-R}} & \text{BackGround} & \text{Tax}
\end{array}
$$

The specific relations 'RR', 'NB' and 'BT' model the vertical concatenation between records, and the geometrical correspondence between 'Name-Body' and 'Body-Tax' entities, respectively. The initial symbol of the grammar is *Page*. As we stated before, we learn the probability of each rule from the training set.

4.6 EM-based region fitting model

In this section we present the last method proposed for the task of layout analysis. The two previous methods encoded the set of contextual relations through different mechanisms. Our CRF-based method relied on the inclusion of relative location prior through the set of features, and the implicit contextual relations modeled by the graph structure. In the approach based on 2D-PCFG, part of the semantic about the classes of entities was modeled through the probabilistic rules of the grammar. The method that we propose now aims to encode the characteristics of the document structure also within the definition of the model itself, similarly than grammar rules. We rely on a Bayesian Network in order to encode dependence relations between model variables, and to provide a factorization of the probability distribution that accounts for the document structure.

Our model allows to represent many types of collections of documents as long as they are represented by some sort of structure. Basically, in the BN we encode the relations between the different types of entities that compose a page, and the dependencies between these entities and other variables from the model, such as our set of features, or any other information that we want to model. Then, according to the factorization of the BN, for a given segmentation hypothesis we can compute a probability that indicates how well this segmentation fits the actual content on the page.

We focus on the hypothesis that we can approximate the location and shape of the different entities on the document by a set of bi-dimensional Gaussian functions. Thus, we propose an iterative algorithm based in the EM-algorithm to estimate the parameters of each of these functions, according to the probability distribution modeled by the BN. In this way, we are able to find the configuration that better fits the structure of the page. Several questions arise from this hypothesis. First, for a given document the number of Gaussian components to fit is unknown. We need, at least, to have available a prior knowledge about the configuration of the page in order to initialize the set of components. Second, given the final configuration of the Gaussian function, we have to estimate the boundaries of the region that is representing.

In the next sections we define the probabilistic framework of the proposed method in base of the particular case of the defined layout analysis task for the BH2M collection. Documents on this collection are composed of a set of records, that are the same time are composed of a set of regions, distributed along the page. The location and size of these entities may vary through the set of pages. Although initially we do not know how many records compose the document, we do know how they are structured, and therefore, we could learn about the variations of their location and region sizes. However, note that this method can be also addressed to other tasks just by modifying the graphical model according to the characteristics of the problem. Next, we present the instance of the proposed method for this particular task.

4.6.1 Model

Our method is focus on structured documents where the entities that compose them are located in similar areas of the document but varying in its final size and location. This variability can be encoded by means of a bi-dimensional Gaussian function whose mean represent the location across the two coordinates, and whose variance defines the final size of the detected region. We aim to detect rectangular portions of the document, which can be eventually rotated.

We use a random sample of N pixels for the representation of the document content. We replace the previous representation in cells of pixels in order to increase the flexibility of the fitting process of the Gaussian functions, since we think that in this case the space between cell centroids and the rigidity of a grid was not enough accurate for this process. We define a set of variables linked to the set of random pixel locations selected from the document image. First, we define a set of random variables g and u, where g represent the local observations based in the features extracted at pixel level, and u is the set of coordinates (x, y) of each selected pixel. In addition, we define a set of unknown variables y, s, r, representing the regions, the records, and the number of records in the document, respectively. We treat the class background separately as we detailed in the next section. Therefore, for a given document we define a set of $L = y \times r$ possible labels plus the background class.

We depict the variable dependencies in the BN shown in Figure 4.5. First, texture descriptors g depend on the type of regions y (incoming arrow in the node g). Secondly, not all positions u in the document image are equally probable for all the semantic labels y and records s (incoming arrows in the node u). Finally, the values that can take s depend on the numbers of regions in the document r. Besides, we also represent the implicit dependence between a record and the regions that compose it.

We aim to fit a set of rectangular regions to the page, and to label each pixel within these regions to its corresponding region class. Therefore, we define our model as the following MAP probability estimation problem:

$$\hat{y}, \hat{s}, \hat{r} = \underset{y,s,r}{\operatorname{argmax}}\, p(y, s, r|g, u) = \underset{y_n,s_n,r}{\operatorname{argmax}} \prod_n \frac{p(y_n, s_n, g_n, u_n|r)p(r)}{p(g_n, u_n|r)} \tag{4.17}$$

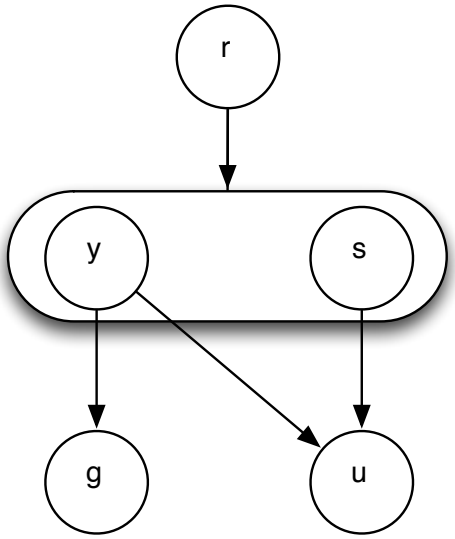


Figure 4.5: PGM showing dependences between variables.

thus, we reduce the problem to estimate the conditional densities $p(y_n, s_n, g_n, u_n|r)$ and $p(g_n, u_n|r)$ for each pixel n and each possible value of r. We learn the value of $p(r)$ for each possible value of r computing its relative frequency from the training set. We rely on the proposed BN to factorize the probability distribution $p(y_n, s_n, g_n, u_n|r)$:

$$p(y_n, s_n, g_n, u_n|r) = p(g_n|y_n)p(u_n|y_n, s_n)p(y_n)p(s_n|r) \tag{4.18}$$

which, without lose of generality can be rearranged as:

$$p(y_n, s_n, g_n, u_n|r) = p(g_n, u_n|y_n, s_n)p(y_n, s_n|r) \tag{4.19}$$

We observe that we can apply the EM algorithm to the factorization above, being $p(y_n, s_n|r) = p(y_n)p(s_n|r)$ the weights of the probability $p(g_n, u_n|y_n, s_n)$. To define $p(g_n, u_n|y_n, s_n)$ we rely again on the graphical model of Figure 4.5:

$$p(g_n, u_n|y_n, s_n) = p(g_n|y_n)p(u_n|y_n, s_n) \tag{4.20}$$

where $p(g_n|y_n)$ is estimated from the texture descriptor defined in Eq. (4.4) and Eq. (4.6), depending on the set of features we want to use. On the other hand, $p(u_n|y_n, s_n)$ is provided by the Gaussian function as:

$$p(u_n|y_n, s_n) = \exp\left\{-\frac{1}{2}(x_n - \mu_{y_n,s_n})^t \Sigma_{y,s}^{-1}(x_n - \mu_{y_n,s_n})\right\} \tag{4.21}$$

which correspond with the pdf linked to the bi-dimensional Gaussian function that fits the region y_n, s_n with parameters $\theta_{y,s} = (\mu_{y,s}, \Sigma_{y,s})$. Finally, we define $p(g_n, u_n|r)$ by summing conditional probabilities:

$$p(g, u|r) = \sum_{y,s} p(g_n, u_n|y_n, s_n)p(y_n, s_n|r) \tag{4.22}$$

Thus, our method depends on the configuration of the set of defined Gaussian functions. We define the total amount of parameters Θ as the union of all the parameters $\theta_{y,s}$. Once we have defined the equations of our model, we need to establish how to update the parameters of the mixture to find the corresponding regions.

EM for region estimation

In Section 3.6 we defined the EM algorithm for the estimation of the parameter of a Gaussian mixture. For this method we adopted the same general equations presented but adapted to the proposed probabilistic framework. Our adapted version of the EM algorithm follows the following steps: During the E-step we compute the expectation of the variables y, s, r conditioned to the observations g, u and the old set of parameters Θ':

$$\begin{aligned} Q(\Theta, \Theta') = &\sum_{y,s}\sum_{n=1}^{N} \log p(y_n, s_n|r)p(y_n, s_n, r|g_n, u_n, \Theta') + \\ &+ \sum_{y,s}\sum_{n=1}^{N} \log(p_k(g_n, u_n|\theta_{y,s}))p(y_n, s_n, r|g_n, u_n, \Theta') \end{aligned} \tag{4.23}$$

which in our case consist in the computation of Eq. (4.17). Then, in the M-step, the parameters of the Gaussian functions are updated. This process is repeated until the convergence of the algorithm. We rely in the update expressions computed in Section 3.6. These expressions, adapted to the proposed models are:

$$\begin{aligned} \mu_{y,s}^{new} &= \frac{\sum_n u_n p_{old}(y_n, s_n, r|g_n, u_n)}{\sum_n p_{old}(y_n, s_n, r|g_n, u_n)} \\ \Sigma_{y,s}^{new} &= \frac{\sum_n p_{old}(y_n, s_n, r|g_n, u_n)(u_n - \mu_{y,s}^{new})(u_n - \mu_{y,s}^{new})^t}{\sum_n p_{old}(y_n, s_n, r|g_n, u_n)} \end{aligned} \tag{4.24}$$

besides, we update the mixture weights as:

$$p(y,s|r)^{new} = \frac{1}{N}\sum_{n} p_{old}(y_n, s_n, r|g_n, u_n) \tag{4.25}$$

In the case of the background class, we define its conditional probability as:

$$p(background|g_n, u_n)^{new} = \frac{1}{1+\sum_{y_n,s_n} p_{old}(y_n, s_n, r|g_n, u_n)}$$

An update on the parameters of the Gaussian mixture results in a displacement and a re-scaling of the Gaussian ellipses to better fit the entities on the page. After some iterations these components are suppose to be centered on the regions that we want to detect. We set the convergence criterion in base of the Kullback-Leibler divergence between two consecutive estimation of Eq. (4.17). We summarize our method in Algorithm 1.

Algorithm 1: EM algorithm for region fitting

1. Θ initialization
2. E-step:
   a Estimate $p(y, s, r|g, u, \Theta')$, Eq. (4.17)
3. M-step: estimation of the region parameters
   a Update Θ: $\Theta \leftarrow \Theta'$, Eq. (4.24)
4. Repeat steps 2-3 until convergence
5. End

EM initialization

An inconvenience of the EM algorithm is its sensitivity to the initial configuration of the parameters. We need to provide an accurate initial estimation of both the number and the parameters of the Gaussian functions that we want to fit. With this purpose we define an additional set of features in order to capture the information about the common shape and location of any possible entity to detect.

Shape and location descriptors are both learned from the labeled data in the training set. Shape descriptors are learned by analyzing the height $h_{y,s}$ and width $w_{y,s}$ for each class y within a region s. Note that these features are not dependent to the number of regions within a page, since they just provide information about the dimension of each entity. In what regards to the location descriptor we distinguish

between the possible number of regions in s. We compute the euclidean distance $d_{y,s} = (dx_{y,s}, dy_{y,s})$ from the center of each entity y in a region s to the origin of the page for each of the observations.

According to these values we can initialize the set of Gaussian components. Initially the algorithm starts with a coarse estimation of the parameters $\{\mu, \Sigma\}$ for each of the components of the mixture. We define a component for each possible pair of elements $\{y, s\}$. Thus, the mean of the resulting Gaussian is defined by the pair $\mu_{y,s} = \{\mu_{dx_{y,s}}, \mu_{dy_{y,s}}\}$. To define the co-variance matrix we took into account that the variance of the size of a region is independent from its location. Since each of the co-variance matrices represent the size of each region we to approximate each co-variance by two flat Uniform distributions, one for each of the dimensions. Thus, according to the definition of these distributions and the relation between the variances of Gaussian and Uniform distributions, the final form of the co-variance matrix is defined by:

$$\Sigma_{y,s} = \begin{pmatrix} \frac{\mu^2_{h_{y,s}}}{12} + \sigma^2_{h_{y,s}} & 0 \\ 0 & \frac{\mu^2_{w_{y,s}}}{12} + \sigma^2_{w_{y,s}} \end{pmatrix} \tag{4.26}$$

where $\mu_{w_{y,s}}$ and $\mu_{h_{y,s}}$ are the mean width and mean height computed from the training set, respectively, and $\sigma^2_{w_{y,s}}$ and $\sigma^2_{h_{y,s}}$ the corresponding variances.

Final labeling

Given the distribution of each component along the page, we estimate for each of them the rectangular bounding box that comprises it. The last step of our method is to compute these rectangular regions that according to our assumptions should match up with the different regions of the page.

We defined each of these rectangles according to the computed parameters of each of the components of the mixture. On the one hand, the center of a region will be defined by the mean of its component $\mu_{y,s}$. On the other hand, in order to estimate the size of the region we define the co-variance matrix as $\Sigma_{y,s} = ADA^t$, where A represents the rotation matrix and D is a diagonal matrix that will define the dimensions of the region as: $[sx, sy] = 2\sqrt{3} \cdot diag(D)$. These values define a rectangular area on the image that will be labeled as the region $\{y, s\}$. Repeating this computation for every component of the mixture will give the final segmentation of the image.

4.7 Experimental evaluation

In this section we describe the set of experiments performed in order to validate the proposed methods on the task of document layout analysis. We presented two datasets as representatives of structured and non-structured collections of documents in Section 4.2. In the following sections we describe the metrics used to show the obtained results, the set of datasets selected, and the battery of experiments performed.

4.7.1 Metrics

We present our results in terms of precision, recall and F-Measure computed at pixel level. These metrics are commonly used in the task of segmentation and allow to compare with other works in the area. The precision rate measures the portion of detected pixels that are correct, while the recall measure indicates the portion of correct pixels that are detected with respect to the total. The harmonic mean between these two measures is the F-Measure. These metrics are used in the ICDAR 2009 Page Segmentation Competition together with another metrics to evaluate the contestant methods, so we able to directly compare our results with them. Precision, recall and F-measure are computed at pixel level as:

$$\begin{aligned} \text{Precision rate} &= \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \\ \text{Recall rate} &= \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \\ \text{F-Measure} &= \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \end{aligned} \tag{4.27}$$

4.7.2 Statistical hypothesis test

We are also interested in checking the contribution of the RLF from a statistical point of view. A common way to perform this verification is by means of a Student's t-test on the obtained results, however, as this measure is devised to be applied on Normal functions, and the F-measure does not satisfy this condition, we compute the Wilcoxon rank-sum test instead. This test is considered the non-parametric version of the Student's t-test, and is suitable for non-Normal functions. The test is devised to check whether one of two independent distributions tends to produce larger values than the other. With this verification on the results obtained by our methods, we are able to check whether the results produced by the experiments using RLF significantly better or not with regard a significance level α. The statistic U given by the test is computed by the expression:

$$U1 = R1 - \frac{n1(n1+1)}{2} \tag{4.28}$$

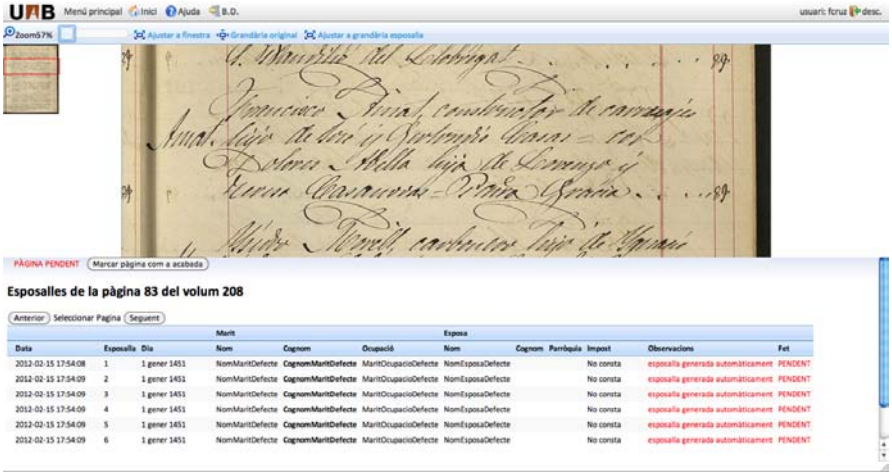


Figure 4.6: Ground-truth tool

where n1, in our case, is the number of results obtained only by the experiments using Gabor features on each page of the test set, and R1 is the sum of the F-measure obtained on each image. The statistic U is then given by the minimum value between U1 and U2, considering U2 the same calculation in the case of the results obtained by the RLF approach. We consider in our experiments a significance level of 5%, which means that values of U under this value will be considered statistically significant.

4.7.3 Ground Truth

The corresponding ground-truth for the volume 208 of the BH2M dataset was manually created for the realization of these experiments. The labeling process was done through a web-based crowd-sourcing tool developed in the Computer Vision Center (CVC) that allows to label different collections from the ongoing projects. The tool is devised to receive the contributions from any person interested in participate in the project, either by tagging the different elements or making text transcription. The labeling process consist in marking each of the three areas of a interest on each license by drawing the corresponding bounding boxes. A screen-shot from the labeling process of one page can be seen in Figure 4.6. The final ground-truth is given by a SVG file with all the registered information of the page. This output file contains the coordinates of the upper corners from the different bounding boxes and its vertical and horizontal length. The hierarchical distribution from the SVG tags allow to arrange the file in a set of licenses, which results very useful for our task. The entire volume contains 593 pages of which we label at pixel level the first 200 pages of the volume. We randomly split 150 pages for training, 10 pages as validation set and the remaining 40 for test. The resolution of the images is 300 dpi ($\approx 2750 \times 3940$ pixels).

4.7.4 Parameters and settings

Along the previous sections we described some parameters required for the definition of the methods proposed. In this section we describe this set of parameters and how they are fixed.

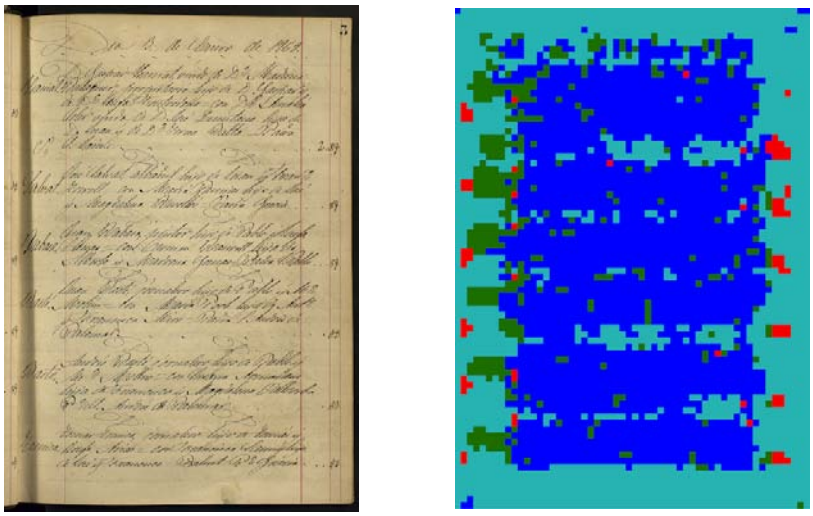
General settings

Now we describe the parameters concerning the set of feature descriptors and the document representation. In what regards to the document representation in cells, we evaluate cells of 25×25 and 50×50 pixels. Since we are interested in reducing the computational time of our methods and at the same time capture contextual information in the cell, considering sizes under these values is not a coherent option. On the other hand, considering sizes above these values could exclude small regions of interest producing some missing detection.

With respect to the parameters of the texture filter bank, we computed a 36-dimensional feature vector using 9 orientations and 4 frequencies of the filter. These values were chosen to ensure that the Gabor functions cover the frequency space. Additionally, we set the overlapping degree $q = 0.5$, $f_{max} = 0.35$, and the scaling factor $k = \sqrt{2}$. Finally, we also learnt the parameters of the logistic regression used in Eq. (4.7) from the training set. We apply this process over each image in the training set on a total of 3,000,000 pixels randomly selected. We randomly select 500,000 feature vectors from the resulting features from each class of the dataset. Thus, we use 500,000 samples to learn the GMM for each class. We set the number of components of each mixture in 36, the same than the number of components on the feature vector.

In what regards to the implementation of the probability maps, we set the size of a map in 200×200 pixels, so that the computed displacements have to be normalized by the image width and height and quantized to be represented in this range. We also fix the Map coordinates in the range $[-1, 1] \times [-1, 1]$ (see Figure 4.4). In addition, each map is normalized in order to define a proper probability distribution and ensure $\sum_c M_{c|c'}(u, v) = 1$.

We test all the proposed methods on the two sets of presented features. First we show results using only texture features, and compare with the RLF. In any case, the computation of the RLF was done according to the initial classification obtained by the GMM using texture features as we explain in Section 4.3.3. In order to understand the contribution of the RLF we show in Figure 4.7 the initial classification of a document. We appreciate as some cells at the left of the image are labeled with the class tax, when this class must appears only in the right side. This effect is expected to be corrected by the RLF.



(a) Original image. (b) Classification result.

Figure 4.7: Initial cell-level classification result of one image from BH2M dataset. The different classes are: tax (red), body (blue), name (green).

EM-based settings

We define two stop criteria for our EM algorithm. On the one hand, we fix the maximum number of iterations of the EM to 50, we empirically check that over this value there are not significant changes in the result. On the other hand, we established the convergence criterion according to the Kullback-Leibler divergence between two consecutive iterations for the distribution of $p(y, s, r|g, u)$. We fixed a convergence value for $\varepsilon \ll 0$. With respect to the parameters of the method, we fixed the number of random points in $20,000$ to ensure an acceptable computational time and a proper coverage of the page. The mean size of an image in the evaluated dataset is $4,000 \times 2000$ pixels, so we are processing a page using only the 2% of the total information on it. We select the set of random pixels ensuring an uniform distribution on the page.

In what regards to the parameter r that denote the number of records, we checked that this number in the training set is bounded from 5 to 7 records, so we will run our method for these three possible values of r, and keep the one with high probability.

Finally, we considered two criteria in order to evaluate the quality of the results. On the one hand we want to detect the logical layout of the page, which means to detect all the records and we will require the three parts in a record to be correctly identified. Therefore we provide a measure of global detection of regions. On the other hand we want to perform quantitative and qualitative comparison of this approach with respect to the other methods.

2D-PCFG settings

We used a 2D-PCFG to tackle the page segmentation problem on the BH2M dataset. Given the nature of the problem such that the documents have a known structure, we manually define the grammar as defined in Section 4.5.1. According to the model described in Section 4.5, we have to train several probability distributions. The probabilities of the productions of the grammar and the size probabilities for each non-terminal are estimated from the training data as explained in Section 4.5.

The 2D-PCFG model combines probability distributions that are learned independently, hence, there may be scaling problems when multiplying the different probabilities. For this reason, the resulting probability is obtained such that each distribution has an exponential weight that adjusted the scale of them. As a result, we have to tune three weights: the probabilities of the grammar $P(A \to c)$ or $P(A \xrightarrow{r} B\,C)$, the probability of a region $P(c \mid z)$ and the probabilities $P(\ell_{i\times j} \mid A)$. Then, the weights of the system are tuned using the downhill simplex algorithm by maximizing the average F-measure for classes *Name*, *Body* and *Tax* when recognizing the validation set.

4.7.5 Experiments

Cell size experiment

We define a document representation in a set of cells composing a rectangular grid. In this experiment we evaluate the effect of using different cell size for the task of layout analysis of the BH2M dataset. We compare two different cell sizes, 25×25 and 50×50 pixels.

The model selected to carry out this experiment is the CRF-based approach using the set of features composed by the combination of both Gabor features and the RLF. The results of the experiment are shown in Table 4.1. We appreciate that using cell size of 50×50 we obtain better results in general than using 25×25 pixels per cell. Initially we can think that using smaller cell sizes would produce better results, however, the distribution of the classes within the image explain these results. The area of the class *tax* is small in comparison with the classes *body and name*, so the division of this region into cells may produce that some *tax* cells are labeled with the class *background*, obtaining a small precision rate as can be seen in the results. The same effect is observed for the class *name* in less proportion. Considering the obtained results, for the rest of experiments we fix a cell size of 50×50 pixels.

Results on BH2M dataset

This section describes the experimentation carried out to evaluate the different models proposed for the task of layout analysis on the BH2M dataset. We emphasize on the logical analysis of the layout, since we want to detect each record and label each of its semantic parts accordingly. In the first place we perform a comparison

	Relative Location Features Cell size 25x25			Relative Location Features Cell size 50x50		
	Precision	Recall	F-measure	Precision	Recall	F-measure
Body	0.83	0.96	0.88	0.88	0.96	**0.92**
Name	0.70	0.72	0.71	0.77	0.73	**0.74**
Tax	0.31	0.65	0.41	0.69	0.66	**0.66**

Table 4.1: Results on 5CofM dataset comparing cell sizes of 25×25 and 50×50 pixels.

between the model based on PGMs in contrast to 2D-PCFG. We test on this experiment the performance of three inference algorithms for PGMs: GraphCut, Loopy Belief Propagation (LBP) and Iterated Conditional Models (ICM); and compare their performance against the proposed grammatical model. In the second place we show the result obtained by the EM-based method. In this case we perform a qualitative comparison of the record detection in contrast to the previous two approaches.

We resume in Table 4.2 the results obtained in the sequence of experiments performed. In views of the results we can identify four main discussion topics: the qualitative detection of the records, the comparison between set of features, the comparison between inference methods, and the comparison between the two proposed models.

Regarding the qualitative detection of the records, results show that in general for all the approaches the class *Body* is detected with good F-measure rates, whereas classes *Name* and *Tax* represent the most challenging part. This effect is related with the proportion of the document that represents each region, because it is more difficult to properly classify small regions. Errors are usually found at the boundaries of the regions, and therefore a missing cell of the *Tax* class have more impact on the results than the same cell of the *Body* class.

Regarding text classification features, we can clearly observe two different behaviours. PGMs performed significantly better with RLF features than when regular Gabor features are used. In general, the recognition of the three classes is always improved, specially in terms of precision rate. The improvement in the detection of the *Name* entities is remarkable, which demonstrates the improvement in the cell classification with respect to the initial Gabor-based classification (See Figure 4.7). The class *Tax* was the most challenging class for both sets of features. We see that altough the differences in F-measure is small, using Gabor features achieves less precision but more recall rates whereas RLF detects more precise areas and lower recall rates, since the search area focalized to the area given by the probability maps. To assert whether these differences are significant or not from a statistical viewpoint, we run a two-sided Wilcoxon rank sum test with a significance level of 5%. The test obtained values under the confidence level on every experiment based on PGMs, which proves that including RLF into the CRF model significantly improves the detection results of the regions from structured documents.

Now we analyze the results obtained by different inference methods. In general,

Model	Features	Class	Precision	Recall	F-measure
Graph Cut	Gabor	Body	0.89	0.87	0.88
		Name	0.27	0.82	0.40
		Tax	0.35	0.94	0.51
	RLF	Body	0.88	0.96	**0.92**
		Name	0.77	0.73	**0.74**
		Tax	0.69	0.66	**0.66**
LBP	Gabor	Body	0.89	0.83	0.86
		Name	0.27	0.74	0.39
		Tax	0.45	0.79	0.56
	RLF	Body	0.88	0.93	0.90
		Name	0.72	0.71	0.69
		Tax	0.85	0.43	0.55
ICM	Gabor	Body	0.89	0.83	0.85
		Name	0.27	0.74	0.39
		Tax	0.45	0.79	0.56
	RLF	Body	0.89	0.92	0.91
		Name	0.71	0.74	0.72
		Tax	0.90	0.45	0.58
2D-PCFG	Gabor	Body	0.91	0.95	**0.93**
		Name	0.73	0.86	**0.78**
		Tax	0.69	0.80	**0.71**
	RLF	Body	0.90	0.95	0.92
		Name	0.77	0.79	0.77
		Tax	0.78	0.65	0.68

Table 4.2: Classification results for different models and text classification features.

LBP is the method that obtained worst recognition rates, whereas ICM and Graph-Cut obtain similar values, being Graph-Cut the one with higher detection rates for the three classes, with F-measure 0.92, 0.74 and 0.76 for classes *Body*, *Name* and *Tax*, respectively. We believe that this differences rely on the definition of the pairwise potential. In the case of LBP we established the set of parameters according to the prevalence on the training set of each pair of classes. However, since elemants from the classes *Name* and *Tax* are less common, the resulting parameters are biased and favors the labeling of *Body* elements. However, in the case of the Graph-Cut approach all classes are considered equiprobable in this sense, and the only penalization comes from the assignation of different neigbor classes.

Regarding the results obtained by the 2D-PCFG model we see as this model obtain similar performance using both sets of features, and even results with Gabor features were slightly better than results provided by RLF features. 2D-PCFG is a powerful model that is able to take advantage of the knowledge about the document structure. Thus, grammars were able to overcome the lacks of Gabor features obtaining very good results for all classes without the additional spatial information provided by RLF. Given that we learn probabilistic information about the structure of the documents from training data, the model was able to successfully parse using the regular Gabor features. Figure 4.11 shows an example of recognition using 2D-PCFG and both sets of features. We can see how the overlapping region between *Name* and *Body* is classified as *Body*. Also, as results pointed out (see Table 4.2), Gabor features obtained higher recall and we can see that resulting regions are larger. On the other hand, RLF provided higher precision by adjusting better the size of the regions detected. Finally, recognizing the space between records is difficult due to the large calligraphic letters considered as background.

Comparing the performance of the different models, we see as in this task 2D-PCFG obtain better results than PGMs, more remarkable in the case of the *Tax* class. The 2D-PCFG model with Gabor features achieved F-measure 0.93, 0.78 and 0.71 for classes *Body*, *Name* and *Tax*, respectively. Results showed that grammars achieved a great improvement even using Gabor features, while PGMs in the proposed configuration requires of the additional set of features in order to capture the structure information as was expected.

Our CRF-based model classify cells but they do not identify the explicit segmentation in records. However, the most likely hypothesis according to the 2D-PCFG model provides a derivation tree that accounts for both the structure of the document and the segmentation in cells. Using this information we can compute the number of records detected in each document, and then, compute the percentage of documents in the test set where the number of records detected is correct. We consider that a record is detected when we identify the three regions that compose it with at least a 50% of precision rate. 2D-PCFG in combination with Gabor features computed the right number of records in 80% of the test documents, whereas with RLF features only 52.5% of the documents had the correct number of records detected. The lower rate in PGMs in mainly produced by the records where the *Tax* class is not detected or due to over-segmentation of the regions. This measure helps to assess the quality of the

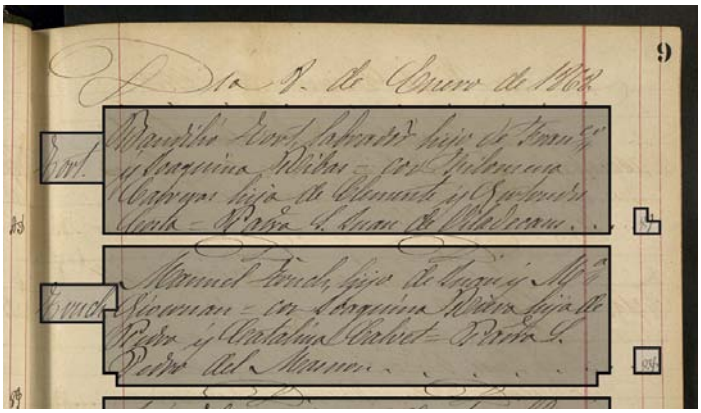


Figure 4.8: a) Ground-truth

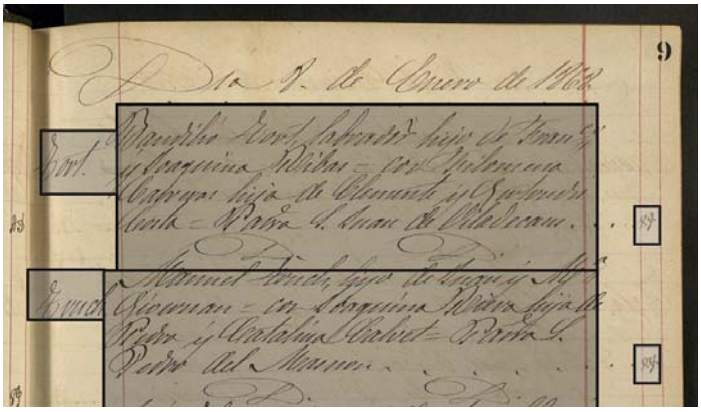


Figure 4.9: b) 2D-PCFG with Gabor

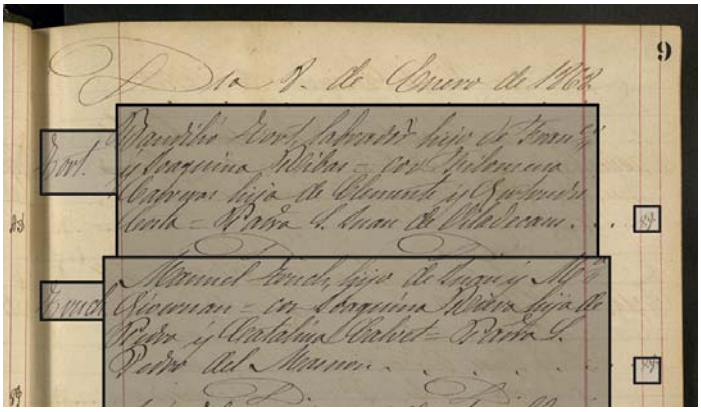


Figure 4.10: c) 2D-PCFG with RLF

Figure 4.11: Example of page segmentation and structure detection with 2D-PCFG and different text classification features.

Model	Features	Class	Precision	Recall	F-measure
EM-based	Gabor	Body	0.83	**0.93**	0.88
		Name	0.27	**0.99**	0.41
		Tax	0.20	**0.95**	0.32
	RLF	Body	0.85	0.93	0.89
		Name	0.39	0.94	0.54
		Tax	0.27	0.90	0.41

Table 4.3: Classification results for different models and text classification features.

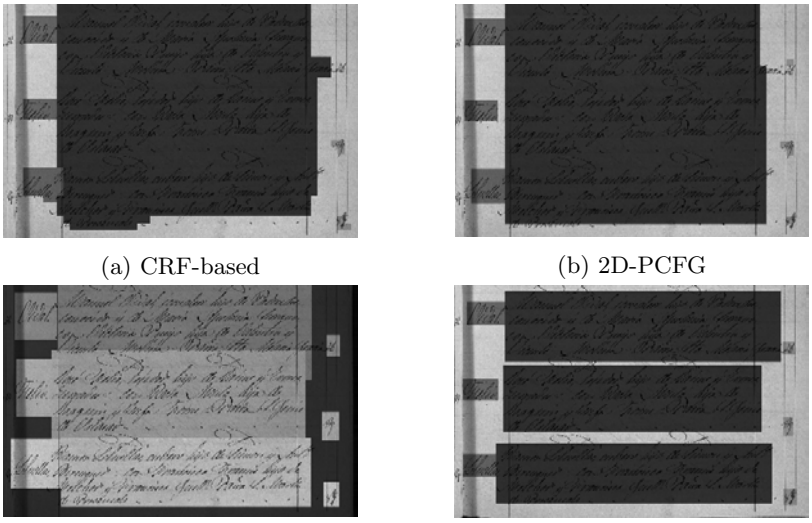
recognition such that we can see that in addition to the slight improvement of Gabor features with respect to RLF features, the number of records detected presented a significant improvement for the 2D-PCFG.

Experiments EM-based method

We show in Table 4.3 the results obtained for the three different classes and both set of features. Overall results show a similar trend in the detection of each class. Regions from classes *Body* and *Name* are detected with higher F-measure rates, whereas the class *Tax* still represent the most challenging part.

We see that in terms of F-measure rates the proposed method does not produce a significant improvement in the detection of the three classes, however we think that these values do not fairly reflects the real contribution of the method. The contribution can be seen in the Recall values with respect to the other methods. With the proposed method we are able to detect more than the 90% of all the regions using any set of features, which in the domain of our task should be considered as a great advantage. In addition, given that the objective of our method is to obtain a logical layout of the page, we also evaluated the obtained results by the number of pages properly segmented (*i.e., the number of detected records equal to the number of records in the page.*), getting an 90% of accuracy.

In comparison with the PGMs approach, the representation in cells of pixels produced that some of the small regions from classes *name* and *tax* were missed. That situation also affects to the method based in 2D-PCFG, where in some cases the *tax* region was undetected and therefore the detection of the full record was not correct. Another drawback of the previous methods in comparison with this approach is that the area between two records is merged within the *Body* class producing an overlapping of all the records and consequently in the case of the CRF was not able to return a proper segmentation of the page in records. These problems are improved in the proposed approach. The method is designed to detect the three elements of a record, so that in the case of detecting a record, all three parts will be always included on it. Besides, the representation with random pixels avoid the loss of small regions in comparison with the approach based in cells, since some pixel will be always included



(a) CRF-based (b) 2D-PCFG

(c) EM-based (d) Ground-Truth

Figure 4.12: Comparison between the results obtained by previous methods with respect to the proposed for a specific page.

in the detected area.

All the previous situations are illustrated in Fig. 4.12. Analyzing this image it is also possible to understand the loss in the precision rate. We see that using the proposed approach the regions obtained match with the expected results, although the obtained area for the classes *name* and *tax* is usually wider that the area in the GroundTruth. That effect is produced mainly by two reasons. On the one hand the area around these regions usually presents some noise that the EM tries to fit into the region, also in most of the records the *name* and *body* regions are overlapped, so it is common that some of the overlapped pixels influence the corresponding component of the mixture. On the other hand the variance of the size and location of these classes is higher than in the case of the class *body*. That provokes that at the beginning the components are initialized with larger values in the covariance matrix and because of the noise in the region is difficult to fit to the proper region.

In what regards to the use of RLF with respect to Gabor features, as we saw in the previous methods, the use of these features improve the results when the method do not take the structure into account, as in the case of CRFs. In the presented method the structure was already implicit in the definition of the classes, so the use of RLF in this case do not produce any significant improvement.

Features	Class	Precision	Recall	F-measure
Gabor	Text	0.87	0.93	0.90
	Image	0.74	0.88	0.79
RLF	Text	0.89	0.96	0.92 (0.0539)
	Image	0.80	0.85	0.82 (0.7007)

Table 4.4: Results obtained on the PRImA dataset by the CRF-based method (p-value from Wilcoxon test in brackets).

Results on PRImA dataset

We describe now the results obtained from the experiments performed on the PRImA dataset. For this dataset we can only show results from the CRF-based approach, since the rest of methods can no be applied to this collection of documents. Both methods, 2D-PCFG and the EM-based, are devised to structured documents and require of an structure to learn the form of the grammar, and distribution of the Gaussian functions, respectively. Therefore, we present results using the proposed CRF-based method. Following the same methodology than in the previous set of experiments, we present results for the two sets of features presented. Besides, regarding the inference algorithm, since Graph-Cut obtained the higher rates, we select this method for the evaluation on this dataset.

We show in Table 4.4. Due to the non-structured layout of this collection, the improvement produced by the inclusion of the RLF is not statistically significant according to the values obtained by the Wilcoxon test. Note that the p-value in the case of text regions is not far from 0.05, which means that even for this kind of loosely structured documents, RLF can help in segmenting text regions. However, for the class *image*, the obtained p-value (0.7) show that these features are useless, an expected outcome since images are located in any part of the document.

Making a comparison with the other methods on this task [1, 42] (Table 4.5, we appreciate as the detection rate in terms of F- measure is up to the highest results achieved in the contest ($84\% - 95\%$). In the case of our class image they do not concretely refer to this class but they consider the class Non-text where is included also graphs and other elements, so our results are not directly comparable with them. We show an example of the segmentation of a document in this dataset in Figure 4.13.

Method	Non-text	Text	Overall
DICE	0.66	0.92	0.90
Fraunhofer	0.75	0.95	0.93
REGIM-ENIS	0.67	0.92	0.88
Tesseract	0.74	0.93	0.91
Gabor features	0.79	0.90	0.85 (0.0539)
Gabor + RLF	0.82	0.92	0.87 (0.7007)

Table 4.5: Comparison with the submited methods to the ICDAR2009 page segmentation competition [1].

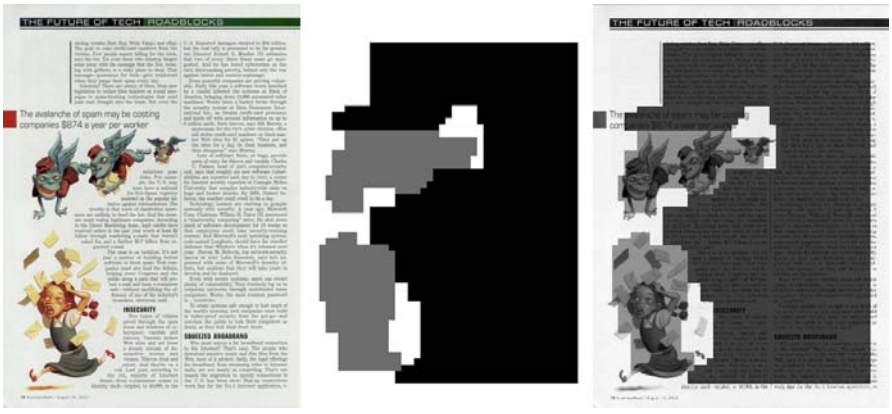


Figure 4.13: Result of the segmentation of one image from the PRImA dataset

4.8 Conclusion

In this chapter we tackled the problem of document layout analysis. We want to detect and segment the different regions that compose a document, and if it is possible, being able to categorize these regions in the set of logical labels that are defined by the task. We defined two set of features for this purpose. A set of texture descriptors to discriminate between the different classes of entities in a more physical way, and a set of relative location descriptors that contributes to the inclusion of the logical knowledge about these entities through its location on the page. We prove that, in models that do no take account of the structure of the document in an implicit way, the contribution of the RLF is significant to encode the logical definition of the classes as long as it can be encoded as a location restriction. On the contrary, in models that already learn in some way about the structure and the logical meaning of the classes, the effect of these features, although is not counterproductive, it does not deserve the computational effort to obtain them.

We proposed three different methods for detecting document regions. In the first place we model the problem as a MAP probability estimation of a CRF. We represent the CRF graph according to the grid defined by the cell representation of the image. The final segmentation is achieved through several inference methods using the information of both set of features and pairwise models. Graph-Cut-based method obtained the higher results. In this type of segmentation tasks, methods based on region cuts are be more accurate in order to set region boundaries. Message-passing approaches requires of more informative features and pairwise models to ensure that the information sent through the variables is rich enough to discriminate correctly. The label bias on the BH2M turn out to be determinant for this method, which was not able to classify the smaller classes correctly. This effect is also validated on the ICM algorithm, which do not rely on the pairwise potentials and also gets higher detection rates for this classes.

In the second place we propose two methods devised to structured documents that take into account this structure in the definition of the models. First, 2D-PCFG compute the most likely derivation tree of the document given the rules of the grammar. Second, we presented a model based in a simple CRF where the dependencies between variables are not considered. The graphical model here was devised to represent the dependencies between variables and to model the probability for each pixel conditioned to a set of Gaussian distributions. In this model, the region estimation process consisted in fitting a set of Gaussian functions on each candidate region. We propose an iterative scheme based in the EM algorithm to find the optimum configuration of the set of Gaussian.

We prove that these structured methods, are better choices for the analysis of layouts with a strong structure, Whereas in case of free layouts these methods can not be applied directly. However, models based in PGM are more flexible and can be adapted to any type of layout. They just need of a proper definition of the features and the relationships between variables.

Chapter 5

Handwritten text line segmentation

The task of text line segmentation arises as a particular case of physical layout analysis where the entities to segment are text lines of a text region. In this chapter we present our method for the task handwritten text line segmentation. We present two different approaches: First, a basic model used to prove our hypothesis. Second, we present the full probabilistic framework based on MRFs for inference and parameter learning. We combine this framework with an improved version of our initial approach to build our final method.

5.1 Introduction

Text line segmentation is an important part of the document analysis process. When we as humans read a text paragraph, it is easy to know that is structured in a set of lines and identify each of them. However, in order to develop automatic systems capable to read text, it is necessary a process to separate between the set of text lines.

The final objective of text line segmentation is to assign the same label to each of the components that compose a text line, being either pixels or connected components. The search process of these components is commonly performed following two possible approaches, depending on the type of document and the difficulty of the task. The first approach consist on finding a set of imaginary lines that connect the set of text characters that compose the text line. These lines are usually categorized as *baseline*, which connects the lower part of the characters bodies, *median line*, which connects the upper part of the character bodies, and the *upper* and *lower* lines, which joins the top and bottom part of the set of ascenders and descenders of the long characters, respectively [13]. By estimating these set of lines it is easy to label the component that compose the text line, and focus on particular areas in order to solve overlapping problems. The second approach is based on searching the set of aligned components

that compose a text line. This process can be performed using different techniques as we described in Chapter 2. This approach is more common for documents with complex layouts.

There are many solutions to the problem of text line segmentation on machine-printed documents that produce robust solutions in general. Text lines from machine-printed documents are characterized by having similar character size, interline spacing and the absence of overlapping and wrapping effect, which facilitates the task. However, segmentation on handwritten documents is still considered an open problem. The great variability on author styles and document layout are the main challenges. It is common to find multiple line orientations, overlapping between the ascenders and descenders from some characters, curved lines and irregular line sizes. In addition, it is common to find documents with annotations or corrections over the text that also complicate the task. On historical documents, apart from the described issues, document degradation and the inclusion of ornamentation and other entities are also an issue to take into account.

In this chapter we present a general method for the task of handwritten text line segmentation. It is based on the estimation of a set of regression lines to fit text lines in handwritten documents. First, we present an initial approach devised to validate our working hypothesis and detect the possible weaknesses of the method. Then, we present our probabilistic framework for inference and parameter learning. We successfully combine the EM algorithm and variational approaches for this purpose. Thus, we summarize the main contributions of this chapter as follows:

1. **General method for handwritten text line segmentation.** We designed our method with the objective to deal with a large variety of documents regardless of their type of layout. Besides, our method is script and language independent.

2. **Easily expandable with prior knowledge.** The proposed probabilistic framework can be easily extended with additional prior knowledge about the task by means of the inclusion of new feature functions.

3. **Probabilistic framework for parameter learning and inference.** We successfully combine the EM algorithm and variational approaches to perform parameter learning and inference on a probabilistic graphical model.

The remainder of this chapter is organized as follows. In Section 5.2 we describe an overview of the method proposed for handwritten line segmentation. Then, in Section 5.3 we describe the initial approach to this method based in GMM. In Section 5.4 we present the probabilistic framework and the final version of the method. Then, in Section 5.5 we show the experimental evaluation of the different methods. Last, we describe in Section **??** the collaborative European project for which the proposed methods were developed

5.2 Method overview

We define a handwritten document as a collection of text paragraphs written along the page, each of them composed of a set of text lines. Each of these lines is composed by a set of words and letters, that in turn, are defined by a set of foreground pixels. Our working hypothesis for the segmentation of text lines is the following: for a given text line, knowing the set of pixels that compose it, we can estimate a regression line through this set of pixels that is a good estimate of the original text line position. Besides, we can model a probability distribution between the set of pixels and the set of text lines. According to this distribution, a given pixel will have a higher probability to belong to its correspondent regression line than to another.

Our document definition covers from documents with an unique text region, to documents with several handwritten annotations along the page. In the last case, each text region can be written in different styles. In order to deal with this variability we designed our method to be applied on each paragraph separately, which implies a previous step in the document processing for identifying the set of paragraphs, and allow us to deal with complex document layouts. In order to clarify the notation, from now onward, and without loss of generality, we define our method assuming a single text region.

Text region segmentation

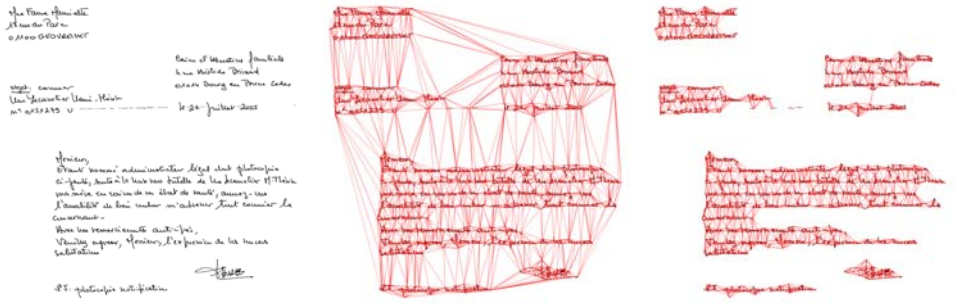
We rely on the Delaunay triangulation for the region segmentation process. The process is based on the text region extraction method defined in [123]. We compute the Delaunay triangulation from a set of selected pixel locations. The main idea of this method is that the Delaunay triangles that connect different text regions have longest sides compared with the triangles that connect components from the same region. We compute an histogram H of longest sides (LS) of the triangles and estimate a threshold over which a triangle is considered to connect text regions. Then, removing this set of triangles will divide the Delaunay graph into a set of sub-graphs that identify the set of text regions. Then, the histogram is smoothed by computing local averages over $2w + 1$ neighbor points:

$$H(x) = \frac{1}{2w+1} \sum_{i=x-w}^{x+w} h(i) \tag{5.1}$$

where x is the LS in pixels. From the resulting histogram we extract the maximum value l_p, which is related with the text line space. Then, the final threshold L_{th} is computed as:

$$L_{th} = \min\{x : x > l_p \text{ and } H(x) \leq 1\} \tag{5.2}$$

In Figure 5.1 we show an example of applying this process to a handwritten document with different text regions. We must highlight that our objective here is



(a) (b) (c)

Figure 5.1: Text region segmentation process. a) Original image. b) Delaunay triangulation computed on the set of selected random pixels. c) Result of the process after removing the selected triangles isolating several text regions.

not to perform the best possible text region segmentation, but to perform a coarse segmentation that help us to divide the problem and treat each of them accordingly to its particular characteristics.

Linear Regression

Regression based techniques are used when, given a cloud of points in the space in the form $\{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$, it is assumed that they are linearly correlated but corrupted by white noise. White noise distributions follow a Gaussian law, centered in the origin with a particular variance σ^2. Thus, for the set of text pixels that compose a particular text line, our objective is to estimate the regression line that better fits this set of pixels. Besides, the Gaussian distribution defined by its variance σ^2 is a good resource to use in order to define the probability for each pixel to belong to each line.

We propose to use a linear regression model where each line is defined by an equation of the type $y = ax + b$, being b the *y-intercept* and a the slope of the line. The measure to indicate the difference between the observed value for a pixel (x_n, y_n) and its predicted value on the regression line is the residue ε_n, and is defined as $\varepsilon_n = y_n - b - ax_n$. According to the set of residues, the usual way to estimate how well a regression line fits the set of pixels is given by the summation of all the square residues:

$$S_r = \sum_n \varepsilon_n^2 = \sum_n (y_n - b - ax_n)^2 \tag{5.3}$$

The objective therefore is to obtain the parameters of the regression line a, b that minimize S_r. This optimization problem can be solved by computing partial

derivatives on both parameters and finding its zero values:

$$\begin{aligned} \frac{\partial S_r}{\partial a} &= \sum_n -2x_n(y_n - b - ax_n) = 0 \\ \frac{\partial S_r}{\partial b} &= \sum_n -2(y_n - b - ax_n) = 0 \end{aligned} \tag{5.4}$$

Straightforward manipulation of the previous equations gives the known linear regression equations for a, b:

$$\begin{aligned} a &= \frac{\sum_n (x_n - \overline{x})(y_n - \overline{y})}{\sum_n (x_n - \overline{x})^2} \\ b &= \frac{1}{N} \sum_n (y_n - ax_n) \end{aligned} \tag{5.5}$$

in addition, σ^2 is computed by mean square error equation:

$$\sigma_l^2 = \frac{1}{N-2} \sum_n (y_n - b - ax_n)^2 \tag{5.6}$$

Thus, a regression line l is defined by the three parameters: $\{a_l, b_l, \sigma_l^2\}$, that at the same time define a Gaussian density function linked to its variance. The likelihood probability associated to this Gaussian function is:

$$q(x, y|l) \propto \exp\left\{ \frac{(y - b_l - a_l x)^2}{2\sigma_l^2} \right\} \tag{5.7}$$

The main problem that arises from our hypothesis is that for a given document, both the number of text lines and the set of pixels that composed each of them are unknown, since actually is the problem that we want to solve. We rely in the classic EM algorithm to overcome this situation by estimating the parameters that define the set of regression lines that better fit data.

The use of the EM algorithm in our problem can be illustrated in the following way: given a handwritten document, we initialize a set of L regression lines according to some criterion. Then, in the successive iterations of the method we will compute new estimates for the parameters a_l, b_l, σ_l^2, $\forall l \in L$ in order to find the configuration that better fits the set of text lines. The new estimation is computed in base of the probability distribution defined over the set of regression lines and the data.

In the following sections we describe two different approaches to model this probability distribution:

1. We define a first simple model in order to validate our working hypothesis. This model is based in the estimation of a Gaussian Mixture Model (GMM) where each component of the mixture is linked to a regression line.

2. Our second model is an extension of the previous one. We develop a probabilistic framework for computing inference and parameter learning on a MRF. We combine the EM scheme for the estimation of the parameters of the regression lines within the learning framework.

5.3 Gaussian approach

We select a set of N pixels that correspond with the gravity centers of the set of text connected components. Liked to this set of pixels we define the set of variables in our model. On the one hand we define a set of observed variables $e_n = (x_n, y_n)$, that represent the coordinates of the selected locations. On the other hand we define a set of hidden variables h to encode the text line labels for each location. Thus, we define our GMM with the following expression:

$$p(h = l|e, \Theta) = \prod_n p_n(e_n|h_n = l, \Theta)p(h_n = l) \tag{5.8}$$

therefore, our model combines a prior probability $p(h_n = l)$, that represents the mixture weights, and a likelihood probability $p_n(e_n|h_n = l)$, which are modeled by the set of Gaussian density functions linked to each line l as we formulated in Eq.(5.7). Last, the parameters in Θ correspond with the parameters that define each of the Gaussian components, which according to our model are the parameters of the regression lines: $\Theta = \{a_1, b_1, \sigma_1^2, ..., a_L, b_L, \sigma_L^2\}$, for each of the L regression lines of the mixture.

We use the EM algorithm for the estimation of the set of parameters of the mixture. For this purpose, the conditional expectation of the log-likelihood function described in Eq. (3.33), but adapted to our model, takes the form:

$$\begin{aligned} Q(\Theta|\Theta') &= \sum_{l=1}^{L}\sum_{n=1}^{N} log(p_n(e_n|h_n = l, \Theta)p(h_n = l))p(h_n = l|e_n, \Theta) = \\ &= \sum_{l=1}^{L}\sum_{n=1}^{N} p(h_n = l))p(h_n = l|e_n, \Theta) + \\ &+ \sum_{l=1}^{L}\sum_{n=1}^{N} log(p_n(e_n|h_n = l, \theta)p(h_n = l|e_n, \Theta) \end{aligned} \tag{5.9}$$

where $\theta \in \Theta$ refers to the subset of parameters of a specific component of the mixture. Now we can compute the update expressions to estimate the new parameter values for the regression lines. For this purpose we compute partial derivatives on Q with

respect to each of them. In addition we estimate the update expression for $p(h_n)$, which is independent from the rest. Thus, the set of update expressions results in:

$$\begin{aligned} a_l^{new} &= \frac{\sum_n (x_n - \bar{x})(y_n - \bar{y}) p_{old}(h_n = l|e_n)}{\sum_n (x_n - \bar{x})^2 p_{old}(h_n = l|e_n)} \\ b_l^{new} &= \frac{\sum_n (y_n - a_l^{new} x_n) p_{old}(h_n = l|e_n)}{\sum_n p_{old}(h_n = l|e_n)} \\ \sigma_l^{2,new} &= \frac{\sum_n (a_l^{new} x_n + b_l^{new} - y_n)^2 p_{old}(h_n = l|e_n)}{\sum_n p_{old}(h_n = l|e_n)} \end{aligned} \tag{5.10}$$

and the expression for the probability $p(h_n)$:

$$p_{new}(h_n) = \frac{1}{N} \sum_n p_{old}(h_n = l|e_n) \tag{5.11}$$

We observe the similarity between the obtained updates and the classical regression equations in Eq. (5.5). Indeed, these equations generalized in the degenerate case, where $p(h_N = l|e_n)$ is a Dirac distribution. Therefore we summarize our method in Algorithm 2. After the initialization of the the method we iterate between the EM steps until the convergence of the model. First, in the E-step we estimate the probability distribution of the mixture computing $p(h|e, \Theta')$ for the current set of parameters Θ'. Then, in the M-step we compute new estimates of the parameters of the mixture.

Algorithm 2: EM algorithm for regression line estimation

1. Θ initialization
2. E-step:
   a Estimate $p(h|e, \Theta')$
3. M-step: estimation of regression lines
   a Update Θ: $\Theta \leftarrow \Theta'$
4. Repeat steps 2-3 until convergence
5. End

In the next sections we describe the initialization process of the set of regression lines and the process to label each of the textual components once the method converges.

5.3.1 Initialization and final labeling

It is known that the EM algorithm is often sensitive to the initial choice of parameters. An inaccurate initialization of regression line parameters may lead the method to fall into a local maximum that do not correspond with the better text line fitting. For each text region detected by the region segmentation process we initialize the correspondent set of regression lines. This is an important step, since we have to ensure the initialization of enough regression lines in order to fit all the text lines. After the convergence of the method we analyze the results obtained in order to detect extra lines and to label each text component to the most likely text line.

Line initialization For each text region detected we compute an estimation of the number of regression lines to initialize and its initial parameters. We first estimate the main orientation of the text region. To do so, we rely again on the orientation of the Delaunay triangle sides. We compute the set of normalized vectors defined by each triangle side $\vec{v_s} = (\vec{v_1}, \vec{v_2})$ and keep only the ones with positive direction, e.g., vectors from left to right. Then, the estimate the orientation as the mean vector orientation.

Next, we estimate the number of regression lines to initialize on each paragraph. First, we rotate the text region according to the computed orientation to process it horizontally. Then, we estimate the text region height h_r and mean connected component height h_{cc}. Then, we define the number of regression lines in this region as $L = h_r/h_{cc}$.

Finally, we compute the initial configuration of the regression lines by distributing equitably the estimated number of lines along the paragraph. Variance $\sigma_{l,t}$ is then computed according the Eq. (5.5) assigning each connected component to the nearest regression line. Note that initially all the lines have the same orientation and interline space. This initialization process is devised to produce an over-estimation of the number of regression lines. We need to ensure that we fit enough lines, and then we can detect and remove the surplus ones.

Post-process and final labeling As consequence of the initial over-estimation of the number of text lines some of the regression lines may end-up the process without fitting any line, and therefore they have to be removed before the final labeling. We define a threshold ε of the probability $p(h = l)$ under which a regression line will be removed.

Final labeling is done at pixel level. Labeling at connected component level is not an option because of touching characters. Therefore, each text pixel $e = (x, y)$ will be labeled according to the most probable line as:

$$\hat{l} = \arg\max_{l} p(h = l|x, y) \tag{5.12}$$

5.3.2 Discussion

The proposed approach addresses the problem of fitting a set of regression lines through the text components in base of a GMM. This probabilistic model provides information at pixel level about how it fits to each of the regression lines. However, the type of information modeled in this model is only local to each pixel location, since we are not considering any contextual information or the interaction between neighbor pixels or regions. We think that modeling this information can be of great importance specially for this task, since it can help to reinforce the labeling decision of conflict areas such as touching components or the case of in curved regions. For the next approach to the proble, we propose to model the probability distribution with a MRF in order to capture this information.

In addition, in the previous model we estimate regression lines of the same length than the region width. This approach is correct for regular text regions as long as they are properly segmented. Even in that case, the existence of short lines or misaligned words can result in a wrong detection or in the confusion of the fitting process. In the next proposed model we solve this situation by defining segments instead of full lines. In this way we can account for broken lines, misaligned fragments of text, or even process several regions at the time.

Last, we consider that although the proposed initialization accounts for most of the region layouts, due to the sensitivity of the EM algorithm to this step we must provide a more intelligent. Therefore, in the next model we improve this step by adding some common techniques for a coarse line detection.

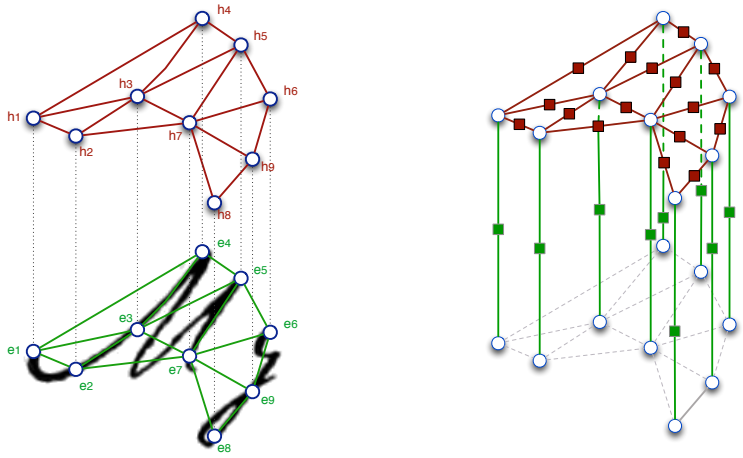
5.4 Probabilistic framework

In this section we describe the second model proposed for the task of handwritten text line segmentation. We follow the same hypothesis than for the previous method although we improve several aspects of the previous approach. First, we preset a more complex probabilistic model that consist of a MRF with an additional set of features. Besides we improve the linear regression model and refine the initialization and post-process steps.

We select a random set of N text pixels ensuring an uniform distribution along the document image in order to cover all the textual components. The use of a random sample reduce the complexity of the overall method, and according to previous works it should not significantly affects to the final result as long as the sample covers all the data.

We define a MRF model composed of two kind of random variables. On the one hand, we have random variables $e = (x, y)$ which correspond to pixel coordinates and, on the other hand, we have hidden variables, h, which denote the labels of text lines. The topology of our model is given by the Delaunay triangulation computed from the set of random pixels, as we show in Figure 5.2a. The result is an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where vertexes in $\mathcal{V}$ are the variables h and e. The set $\mathcal{E}$ is composed of two

kind of edges. First, we have edges between pixel coordinates e and the corresponding text line label. Second, we have edges between adjacent hidden variables h.



(a) Undirected graph: $\mathcal{G} = (\mathcal{V}, \mathcal{E})$

(b) Factor graph

Figure 5.2: Illustration of a region of our MRF. (a) Variables in green represent the *observed* pixels, e. In red, hidden variables h representing the text line labels. (b) Illustration of the two types of factors. Green factors are the v factors that relates the observed and the hidden values. Red factors are the u factors composed only by the hidden values.

We represent our MRF model by a factor graph composed of two type of factor functions in agreement with the two kind of edges describe above, see Figure 5.2b. First, we have factor functions modeling dependencies between observed pixels, e, and hidden variables, h. These are 3-order factors since pixel coordinates are two random variables and we denote them by Ψ_v, with $v \in [1, N]$. Second, we have factor functions modeling dependencies between pairs of hidden variables and we denote them by Ψ_u, where $u = (i, j)$ runs over the edges of the Delaunay triangulation. Thus, the MRF factorizes as a product of Ψ_u and Ψ_v as follows:

$$p(e, h|\Theta) = \frac{1}{Z(\Theta)} \prod_v \Psi_v(e_v, h_v|\Theta_a) \prod_u \Psi_u(h_u|\Theta_b) = \prod_v p_v(e_v|h_v, \Theta_a) p(h|\Theta_b) \quad (5.13)$$

where $\Theta = (\Theta_a, \Theta_b)$ is the set of shared parameters, i.e. all factors Ψ_v share the same parameters Θ_a, and similarly, all factors Ψ_u share parameters Θ_b. Note that the topology of $\mathcal{G}$ allow us to factorizes the MRF model as a product of conditional likelihood probabilities $p_v(e_v|h_v, \Theta_a)$ of pixels $e_v = (x_v, y_v)$ and the *prior* probability of hidden variables h, $p(h|\Theta_b)$. Given this later factorization is easy to see that we can also factorize the partition function $Z(\Theta)$ as:

$$Z(\Theta) = \prod_v Z_v(h_v, \Theta_a) Z_0(\Theta_b) \quad (5.14)$$

where $Z_v(h_v, \Theta_a)$ and $Z_0(\Theta_b)$ denote, respectively, the partition function of the conditional likelihood probabilities and the *prior* probability.

As the previous model, we rely on the classic EM algorithm [107] for the estimation of the new regression line parameters. This method essentially follows the same scheme. The main difference concerns the parameter learning step of the MRF model. First, in the E-step, we update the parameters of the *prior* probability $p(h|\Theta_b)$. We update these parameters using the proposed extension of the GBP, which we explain in section 5.4.2, to allow parameter learning. With the parameters learned we can approximate the posterior probability of each single hidden variable h_v given the coordinates e_v. Then, in the M-step, we update the parameters Θ_a, which correspond to the regression lines. In summary, our proposed scheme is Algorithm 3:

Algorithm 3: EM algorithm for MRF models

1. $\Theta = (\Theta_a, \Theta_b)$ initialization
2. E-step: parameter learning of *prior* probability
   a Update Θ_b: $\Theta_b \leftarrow \Theta'_b$
   b Estimate $p(h_v|e_v, \Theta'_a, \Theta_b)$
3. M-step: estimation of regression lines
   a Update Θ_a: $\Theta_a \leftarrow \Theta'_a$
4. Repeat steps 2-3 until convergence
5. End

In the remainder of this section we explain the linear regression scheme and how to estimate the new updates of its parameters Θ_a. Then we explain how to learn model parameters linked to the prior probability $p(h|\Theta_b)$. We will conclude this section with the definition of the feature functions used for the handwritten text line segmentation task.

5.4.1 EM algorithm for linear regression

We defined a set of factor functions that encode the information within the MRF. Each factor function is composed of a set of feature functions f_k and g_k where k runs in I_u or I_v depending whether the feature function is defined on Ψ_u or Ψ_v, respectively. These feature functions are embedded in factors as:

$$\begin{aligned} \log \Psi_u &= \sum_{k \in I_u} f_k(h_u|\Theta_b) \\ \log \Psi_v &= \sum_{k \in I_v} g_k(h_v, e_v|\Theta_a) \end{aligned} \tag{5.15}$$

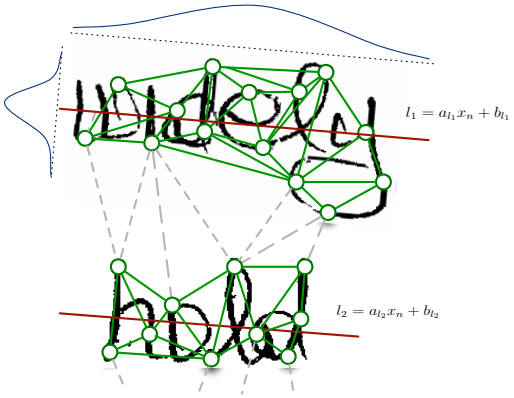


Figure 5.3: Hypothetical region of our graphical model that relates the pixels from words from consecutive lines. Messages sent through the dashed lines are supposed to favor a different label for each connected pixel.

we replace the above definitions and the MRF model of Eq. (5.13) in Q and we have:

$$\begin{aligned} Q(\Theta|\Theta') = \sum_u \sum_{k \in I_u} \left[\sum_{h_u} f_k(h_u|\Theta_b) p_u(h_u|\Theta'_b) \right] + \\ + \sum_v \sum_{k \in I_v} \left[\sum_{h_v} g_k(h_v, e_v|\Theta_a) p_v(h_v|e_v, \Theta'_a, \Theta'_b) \right] - \log Z(\Theta) \end{aligned} \tag{5.16}$$

with this expression we find the new parameter updates by finding the local maximum of Q, which correspond with the M-step. Thus, the partial derivative of Q for a parameter $\theta_k \in \Theta_a$ is:

$$\begin{aligned} \frac{\partial}{\partial \theta_k} Q(\Theta|\Theta') = \sum_v \sum_{k \in I_v} \mathrm{E}_{h_v} \left(\frac{\partial}{\partial \theta_k} g_k(h_v, e_v|\Theta_a) | e_v, \Theta'_a \right) - \\ - \sum_v \frac{\partial}{\partial \theta_k} \log Z_v(h_v, \Theta_a) = 0 \end{aligned} \tag{5.17}$$

For this model we rely again in a linear regression model, although with an slight modification. The objective again is to estimate a set of L lines in the form $y_v = a_l x_v + b_l$ with vertical variance $\sigma^2_{l,t}$ from the set of pixels that compose it. We introduce a new parameter in order to define a pair of bounds that defines a segment of l. The main reason is to fit lines of different sizes instead of considering an unique line size. These bounds are given with respect to the center of the segment c_l by the horizontal variance $\sigma^2_{l,h}$. Therefore, a line l is defined by the following five parameters: $\theta_a = \{a_l, b_l, c_l, \sigma_{l,t}, \sigma_{l,s}\}$ that define two Gaussian density functions linked to the

horizontal and vertical variances. The associated likelihood probabilities are:

$$\begin{aligned} p_t(x_v, y_v|h_v = l, \theta_a) &\propto \exp\left\{\frac{(y_v - a_l x_v - b_l)^2}{2\sigma_{l,t}^2}\right\} \\ p_s(x_v, y_v|h_v = l, \theta_a) &\propto \exp\left\{\frac{(x_v - c_l)^2}{2\sigma_{l,s}^2}\right\} \end{aligned} \tag{5.18}$$

for a pixel $e_v = (x_v, y_v)$ and a line l. These densities will provide a measure of how well a particular pixel fits a line. Figure 5.3 shows an example of a MRF region with two regression lines across two hypothetical words from consecutive text lines l_1 and l_2.

Now we formulate the update equations for each parameter by solving the derivative in Eq. (5.17) for each of them. For a given document the number of parameters to estimate is $|\Theta_a| = 5L$. Note that only parameters $\sigma_{l,t}^2$ and $\sigma_{l,s}^2$ appear on the partition function $\log Z_v(h_v, \Theta_a)$. The update expressions for Θ_a are similar than in our previous approach, although in this case the posterior $p_v(h_v = l|e_v, \Theta'_a, \Theta_b)$ is given by the inference algorithm explained later in section 5.4.2:

$$\begin{aligned} a_l^{new} &= \frac{\sum_v (x_v - \bar{x})(y_v - \bar{y}) p_v(h_v = l|e_v, \Theta'_a, \Theta_b)}{\sum_v (x_v - \bar{x})^2 p_v(h_v = l|e_v, \Theta'_a, \Theta_b)} \\ b_l^{new} &= \frac{\sum_v (y_v - a_l^{new} x_v) p_v(h_v = l|e_v, \Theta'_a, \Theta_b)}{\sum_v p_v(h_v = l|e_v, \Theta'_a, \Theta_b)} \\ c_l^{new} &= \frac{\sum_v (x_v - \bar{x}_v) p_v(h_v = l|e_v, \Theta'_a, \Theta_b)}{\sum_v p_v(h_v = l|e_v, \Theta'_a, \Theta_b)} \\ \sigma_{l,t}^2 &= \frac{\sum_v (y_v - a_l^{new} x_v - b_l^{new})^2 p_v(h_v = l|e_v, \Theta'_a, \Theta_b)}{\sum_v p_v(h_v = l|e_v, \Theta'_a, \Theta_b)} \\ \sigma_{l,s}^2 &= \frac{\sum_v (x_v - c_l^{new})^2 p_v(h_v = l|e_v, \Theta'_a, \Theta_b)}{\sum_v p_v(h_v = l|e_v, \Theta'_a, \Theta_b)} \end{aligned} \tag{5.19}$$

In addition, we also estimate the *prior* probability of each line l given the updated parameters Θ_a as:

$$p_v(h_v = l|\Theta_a, \Theta_b) = \frac{1}{N} \sum_v p_v(h_v = l|e_v, \Theta_a, \Theta_b) \tag{5.20}$$

The key point is that posterior probabilities $p_v(h_v = l|e_v, \Theta'_a, \Theta_b)$ are unknown and consequently we cannot update the parameters of the regression lines. To overcome this problem, we run an approximate inference algorithm that allow us to learn MRF parameters and estimate $p_v(h_v = l|e_v, \Theta'_a, \Theta_b)$.

5.4.2 Inference and Learning

In the previous section, we described how to estimate the parameters Θ_a linked to regression lines. However, parameters Θ_b remain unknown and still have to be learned.

Many parameter learning methods for MRF models relay on free energy methods. These are variational methods that seek density functions that approximate true marginals by beliefs functions that satisfies a set of constraints. Free energies are quite close to $Q(\Theta, \Theta')$ used within the EM algorithm and defined in Eq. (5.16), since both are defined in terms of the Kullback-Leibler divergence (KLD). For instance, the free energy associated to Belief Propagation (BP) algorithm is the Bethe energy as:

$$\begin{aligned} F_{Bethe}(p(h|e, \Theta'_a, \Theta_b)) = &\sum_u p_u(h_u|\Theta_b) \log p_u(h_u|\Theta_b) + \\ &+ \sum_v n_v p_v(h_v|e_v, \Theta'_a, \Theta_b) \log p_v(h_v|e_v, \Theta'_a, \Theta_b) \end{aligned} \tag{5.21}$$

where n_v are related to the number of neighbors of h_v, and can be negative. In our case, we have to include the information given by the likelihood functions of regression lines. So, we define the free energy as:

$$\begin{aligned} F(p(h|e, \Theta'_a, \Theta_b)) = &\sum_u p_u(h_u|\Theta_b) \log p_u(h_u|\Theta_b) + \\ &+ \sum_v c_v p_v(h_v|e_v, \Theta'_a, \Theta_b) \log p_v(h_v|e_v, \Theta'_a, \Theta_b) + \\ &+ \sum_v \sum_{k \in I_v} \sum_{h_v} g_k(h_v, e_v|\Theta'_a) p_v(h_v|e_v, \Theta'_a, \Theta_b) \end{aligned} \tag{5.22}$$

where parameters $c_v > 0$ are any positive real value. The approximate marginals and conditional marginals have to satisfy the usual constraints used in message-passing methods. We summarize them in Table 5.1. First, since p_u and p_v are marginal approximations, they have to be *normalized*. Second, we have to impose the *sum-normalization* constraint between $p_u(h_u|\Theta_b)$ and $p_v(h_v|e_v, \Theta'_a, \Theta_b)$ to ensure consistency between marginal estimation. Unlike usual message-passing algorithms and to well tie the estimated prior probabilities by the model with the observed data, we impose consistency between prior probability of single variables h_v, $p_u(h_v|\Theta_b)$, and posterior probability $p_v(h_v|e_v, \Theta'_a, \Theta_b)$. Finally, we have to ensure coherence between the observations, encoded in the empirical moments μ_k, and model prediction. This last set of constraints is the called *moment-matching* constraint and it provides the parameter learning step for the pairwise parameters and global prior probability, Eq. (5.20). Thus, the minimization of Eq. (5.22) results on a constrained minimization problem that can be solved by means of the Lagrange multipliers as:

Constraint	Formula	L. Multiplier
normalization	$\sum_{h_u} p_u(h_u\|\Theta_b) = 1$	ν_u
	$\sum_{h_v} p_v(h_v\|e_v, \Theta'_a, \Theta_b) = 1$	ν_v
sum-normalization	$\sum_{h_{u\setminus v}} p_u(h_u\|\Theta_b) = p_v(h_v\|e_v, \Theta'_a, \Theta_b)$	λ
moment-matching	$\sum_u f_k(h_u) p_u(h_u\|\Theta_b) = \mu_k$	θ_k

Table 5.1: Set of constraints for the optimization of (5.22)

$$
\begin{aligned}
L(p_u, p_v, \Lambda, \Theta, N) = \\
\sum_u \sum_{h_u} p_u(h_u|\Theta_b) \log p_u(h_u|\Theta_b) + \\
+ \sum_v c_v \sum_{h_v} p_v(h_v|e_v, \Theta'_a, \Theta_b) \log p_v(h_v|e_v, \Theta'_a, \Theta_b) + \\
+ \sum_k \sum_{h_v} g_k(h_v, e_v|\Theta'_a) p_v(h_v|e_v, \Theta'_a, \Theta_b) + \\
+ \sum_k \theta_k \left[\sum_u f_k(h_u) p_u(h_u|\Theta_b) - \mu_k \right] + \\
+ \sum_v \sum_{u \supset v} \sum_{h_v} \lambda_{v \to u}(h_v) \left[\sum_{h_{u\setminus v}} p_u(h_u|\Theta_b) - p_v(h_v|e_v, \Theta'_a, \Theta_b) \right] + \\
+ \sum_u \nu_u \left[\sum_{h_u} p_u(h_u|\Theta_b) - 1 \right] + \sum_v \nu_v \left[\sum_{h_v} p_v(h_v|e_v, \Theta'_a, \Theta_b) - 1 \right]
\end{aligned}
\tag{5.23}
$$

The Lagrangian is convex for all positive c_v over the sets of constraints defined [101]. The computation of partial derivatives of L with respect to $p_u(h_u|\Theta_b)$ and $p_v(h_v|e_v, \Theta'_a, \Theta_b)$ lead us to the expressions:

$$
\begin{aligned}
\log p_u(h_u|\Theta_b) &= -\nu_u - 1 - \sum_k \theta_k f_k(h_u) - \sum_{v \subset u} \lambda_{v \to u}(h_v) \\
\log p_v(h_v|e_v, \Theta'_a, \Theta_b) &= -\frac{\nu_v}{c_v} - 1 + \frac{1}{c_v} \sum_k g_k(h_v, e_v|\Theta'_a) + \frac{1}{c_v} \sum_{v \subset u} \lambda_{v \to u}(h_v)
\end{aligned}
\tag{5.24}
$$

where $\theta_k f_k(h_u)$ refers herein to the feature function $f_k(h_u|\Theta_b)$ defined in Eq.(5.15).

Now, we know that $\sum_{h_u} p_u(h_u|\Theta_b) = 1$, and similarly $p_v(h_v|e_v, \Theta'_a, \Theta_b)$, therefore, computing the exponential values on both sides of the equation, and summing for $\sum_{h_u}$

and $\sum_{h_v}$, we obtain the values for ν_v and ν_u as:

$$\begin{aligned}\nu_u &= -1 + \log \sum_{h_u} \exp\left\{ -\sum_k \theta_k f_k(h_u) - \sum_{v \subset u} \lambda_{v \to u}(h_v) \right\} \\ \nu_v &= -c_v + c_v \log \sum_{h_v} \exp\left\{ \frac{1}{c_v} \sum_k g_k(h_v, e_v | \Theta'_a) + \sum_{v \subset u} \frac{1}{c_v} \lambda_{v \to u}(h_v) \right\}\end{aligned} \tag{5.25}$$

The above are the corresponding partition functions for approximate marginals $p_u(h_u|\Theta_b)$ and conditional marginals $p_v(h_v|e_v, \Theta'_a, \Theta_b)$. Plugging the expressions for $p_u(h_u|\Theta_b)$ and $p_v(h_v|e_v, \Theta'_a, \Theta_b)$ on the primal problem we find the dual problem L^*:

$$L^*(\Lambda, \theta_b) = -\sum_u \log Z_u(\Theta_b, \Lambda) - \sum_v c_v \log Z_v(e_v, \Theta'_a, \Theta_b, \Lambda) - \sum_k \theta_k \mu_k \tag{5.26}$$

which is convex [124]. From the definition of the dual problem we can derive the expressions for its parameters to obtain the formulas that build Algorithm 4. We find the optimal prior probabilities $p_u(h_u|\Theta_b)$ and a posterior probabilities $p_v(h_v|e_v, \Theta'_a, \Theta_b)$ by finding the optimal parameters Λ and Θ minimizing the dual problem L^*, which is convex on $\lambda_{v \to u}(h_v) \in \Lambda$ and $\theta_k \in \Theta_b$. The partial derivative with respect to θ_k is:

$$\frac{\partial L^*}{\partial \theta_k} = \sum_{h_u} f_k(h_u) p_u(h_u|\Theta_b) - \mu_k = 0 \tag{5.27}$$

Observe that the gradient of the dual problem with respect to θ_k is 0 when *moment-matching* constraints are satisfied. Since the prior $p_u(h_u|\Theta_b)$ depends on parameter θ_k we apply line search strategy to find better updates of θ_k. We fix the length η of each step according to the Armijo conditions and the update step is:

$$\theta_k \leftarrow \theta'_k + \eta \left(\sum_{h_u} f_k(h_u) p_u(h_u|\Theta'_b) - \mu_k \right) \tag{5.28}$$

Θ_b are shared by all Ψ_u and we can perform several iterations before starting sending messages. In practice, this does not improve accuracy neither speed up the convergence. Therefore, we update once between each message-passing process.

The message-passing process consist of computing partial derivative with respect $\lambda_{v \to u}(h_v)$, being fixed model parameters Θ_b and Θ'_a found in the previous gradient descend and EM iteration, respectively. This block-gradient descend strategy has been successfully applied before in many other numerical schemes such as [101, 125]. Algorithm 4 follows the same ideas appearing in those papers. Partial derivative with respect to $\lambda_{v \to}(h_v)$ lead to the *sum-marginalization* constraint:

$$\frac{\partial L^*}{\partial \lambda_{v \to u}(h_v)} = \sum_{h_{u \setminus v}} p_u(h_u|\Theta_b) - p_v(h_v|e_v, \Theta'_a, \Theta_b) = 0 \tag{5.29}$$

For each hidden variable h_v, we fix Θ_b, Θ'_a is always fixed in this algorithm, consider the partial derivative with respect to $\lambda_{v\to u}(h_v)$ and let us say that $\lambda^{(new)}_{v\to u}(h_v)$ is the value where the gradient is 0. Then, from the *sum-marginalization consistency* constraint, we have:

$$\begin{aligned}
\frac{\partial L^*}{\partial \lambda_{v\to u}(h_v)} &= \sum_{h_{u\setminus v}} p_u(h_u|\Theta_b) - p_v(h_v|e_v,\Theta'_a,\Theta_b) = 0 \\
p_v(h_v|e_v,\Theta'_a,\Theta_b) &= \frac{e^{\lambda^{(new)}_{v\to u}(h_v)}}{e^{\lambda_{v\to u}(h_v)}} p_u(h_v|\Theta_b) \\
\log p_v(h_v|e_v,\Theta'_a,\Theta_b) &= \lambda^{(new)}_{v\to u}(h_v) - \lambda_{v\to u}(h_v) + \\
&\quad + \log p_u(h_v|\Theta_b) \\
\lambda^{(new)}_{v\to u}(h_v) &= \lambda_{v\to u}(h_v) + \log p_v(h_v|e_v,\Theta'_a,\Theta_b) - \\
&\quad - \log p_u(h_v|\Theta_b)
\end{aligned} \tag{5.30}$$

where we express the update message $\lambda^{(new)}_{v\to u}(h_v)$ in terms of the old messages. To estimate $p_v(h_v|e_v,\Theta'_a,\Theta_b)$ we add the last row of (5.30) over all the pairs u of hidden variables including h_v:

$$\begin{aligned}
\sum_{u\supset v} \lambda^{(new)}_{v\to u}(h_v) &= \sum_{u\supset v} \lambda_{v\to u}(h_v) + \\
&+ \sum_{u\supset v} \log p_v(h_v|e_v,\Theta'_a,\Theta_b) - \sum_{u\supset v} \log p_u(h_v|\Theta_b)
\end{aligned} \tag{5.31}$$

Then, defining A_v as the number of pairs u containing v and rearranging the terms:

$$\begin{aligned}
A_v \log p_v(h_v|e_v,\Theta'_a,\Theta_b) &= \sum_{u\supset v} \lambda^{(new)}_{v\to u}(h_v) + \sum_{u\supset v} [\log p_u(h_v|\Theta_b) - \lambda_{v\to u}(h_v)] \\
c_v \log p_v(h_v|e_v,\Theta'_a,\Theta_b) &= \nu_v - c_v - \sum_k g_k(h_v,e_v|\Theta'_a) - \sum_{u\supset v} \lambda^{(new)}_{v\to u}(h_v)
\end{aligned} \tag{5.32}$$

and adding both equations in both sides, we obtain the update for $\log p_v(h_v|e_v,\Theta'_a,\Theta_b)$:

$$\begin{aligned}
\log p_v(h_v|e_v,\Theta'_a,\Theta_b) &= \frac{1}{c_v + A_v} \log \psi_v(h_v|e_v,\Theta'_a) + \\
&+ \frac{1}{c_v + A_v} \sum_{u\supset v} [\log p_u(h_v|\Theta_b) - \lambda_{v\to u}(h_v)]
\end{aligned} \tag{5.33}$$

where $\log \psi_v(h_v|e_v,\Theta'_a) = \frac{\nu_v}{c_v} - 1 - \frac{1}{c_v}\sum_k g_k(h_v,e_v|\Theta'_a)$ and

$$p_v(h_v|e_v,\Theta'_a,\Theta_b) = \left(\psi_v(h_v|e_v)^{c_v} \prod_{u\supset v} \frac{p_u(h_v|\Theta_b)}{e^{\lambda_{v\to u}(h_v)}}\right)^{\frac{1}{c_v + A_v}} \tag{5.34}$$

We can now to compute the update $\lambda_{v\to u}^{(new)}(h_v)$:

$$\begin{aligned}\lambda_{v\to u}^{(new)}(h_v) = \lambda_{v\to u}(h_v) + \log p_v(h_v|e_v, \Theta'_a, \Theta_b) - \\ - \log p_u(h_v|\Theta_b)\end{aligned} \tag{5.35}$$

The above formulas are more compactly expressed if we define message functions by: $m_{v\to u}(h_v) = e^{\lambda_{v\to u}(h_v)}$ and then, the update rules are:

$$\begin{aligned}m_{v\leftarrow u}(h_v) &= \frac{p_u(h_v|\Theta_b)}{m_{v\to u}(h_v)} \\ m_{v\to u}(h_v) &= \frac{p_v(h_v|e_v, \Theta'_a, \Theta_b)}{m_{v\leftarrow u}(h_v)}\end{aligned} \tag{5.36}$$

In summary, we have the update expression to update the parameters of the feature functions, and the expression for the messages sent between the u and v. These update formulas arranged as shown in Algorithm 4 provide a block gradient descend method that can be parallelized. At each iteration we find new updates of the prior model parameters Θ_b, then we send messages, first from hidden variables h_v to pairs of hidden variables u, and then we combine them obtaining the new messages to update the posterior probability $p_v(h_v|e_v, \Theta'_a, \Theta_b)$ for the next EM iteration of the main algorithm.

5.4.3 Feature functions

In previous sections we defined a general pairwise MRF model adapted to the detection of an unknown number of text lines. This model allow a wide range of unary feature functions to estimate text line position and pairwise feature functions to model text line labels between adjacent pixels. Now we describe the set of feature functions f_k and g_k defined in Eq. (5.15) used for the task of handwritten text line segmentation.

Local fitting This function uses the information provided by the two Gaussian distributions defined in Eq. (5.18) with a slight modification inspired by [61]. It corresponds to a flattened Gaussian distribution on its maximum value. The width of this Gaussian plateau is controlled by a threshold S_l, which is computed as in [61] and it estimates the interline space above and below line l. In summary, we define this function as:

$$g_k(h_v = l, e_v|\Theta_a) \triangleq \begin{cases} \frac{(x-c_l)^2}{2\sigma_{l,s}^2} & if \quad d_l \leq rS_l \\ \frac{(y-a_l x-b_l)^2}{2\sigma_{l,t}^2} + \frac{(x-c_l)^2}{2\sigma_{l,s}^2} & if \quad d_l > rS_l \end{cases} \tag{5.37}$$

where d_l is the residue of the regression line, and $r \in [0, 1]$. This flattened procedure slightly modifies the computation of the partition function of likelihood probabilities

Algorithm 4: Message passing algorithm for constrained minimization of free energies. Z_u and Z_v are the partition function and η is a step length satisfying the Armijo condition.

Data: $\{\mu_k\}$: empirical moments.
Result: Θ_b, Λ: model parameters, $\{p_v(h_v|e_v, \Theta'_a, \Theta_b), p_u(h_u|\Theta_b)\}$ marginals.
Initialize: $\theta_k = 0$, $\theta_k \in \Theta_b$, $m_{v\to u}(h_v) = 1$;
while *not converged* **do**
  for $\forall k \in \{I_u\}$ **do**

$$\theta_k \leftarrow \theta'_k + \eta \left(\sum_{h_u} f_k(h_u) p_u(h_u|\Theta'_b) - \mu_k \right)$$

  for $\forall v$ **do**
    for $\forall u \supset v$ **do**

$$p_u(h_v|\Theta_b) = \sum_{h_{u\setminus v}} p_u(h_u|\Theta_b)$$

$$m_{v\leftarrow u}(h_u) = \frac{p_u(h_v|\Theta_b)}{m_{v\to u}(h_v)}$$

$$p_v(h_v|e_v, \Theta'_a, \Theta_b) = \frac{1}{Z_v} \left(e^{-\sum_k g_k(h_v, e_v|\Theta'_a)} \prod_{u\supset v} m_{v\leftarrow u}(h_u) \right)^{\frac{1}{c_v + A_v}}$$

    for $\forall u \supset v$ **do**

$$m_{v\to u}(h_v) = \frac{p_v(h_v|e_v, \Theta'_a, \Theta_b)}{m_{v\leftarrow u}(h_u)}$$

$$p_u(h_u|\Theta_b) = \frac{1}{Z_u} \left(e^{-\sum_k \theta_k f_k(h_u)} \prod_{v\subset u} m_{v\to u}(h_v) \right)^{\frac{1}{c_v}}$$

$p_v(e_v|h_v, \Theta'_a, \Theta_b)$ but it still depend only on the variances σ^2, and therefore, the update equations in Eq. (5.19) remain valid.

Line probability This function integrates the *prior* probability in Eq. (5.20) into the learning process described in Algorithm 4. This prior probability can be seen as a moment of the indicator function $[h_v = l]$ and consequently we can learn its associated parameter θ_l. We update the corresponding empirical moment μ_k with the line probability estimated in each iteration. In our case, for each line the empirical moment is $\mu_l = p_v(h_v = l|\Theta_a, \Theta_b)$. Thus, the function is defined as:

$$f_k(h_v = l|\Theta_b) \triangleq \theta_l[h_v = l] \tag{5.38}$$

With this function we expect to avoid to assign variables to surplus lines, and reinforce the regression lines with higher probabilities.

Pairwise function Pairwise functions encode the probability of assigning a set of labels to neighbor variables. In our task we encode in this function some assumptions about the configuration of the lines. For example, in a given document two connected variables are more likely to belong to the same text line, i.e. share the same label, or as much, to consecutive lines. Besides, some documents may have two connected variables from non-consecutive text lines, although they represent a few cases with respect to the most common layouts. We define our pairwise function according to those three possible scenarios. The function is defined on Ψ_u, and returns the parameter associated to each possible case:

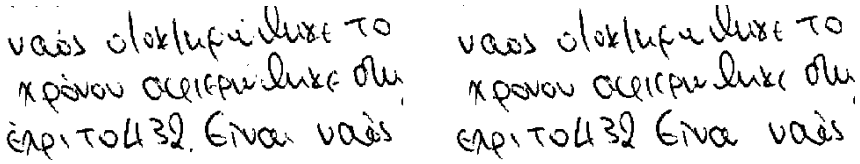
$$f_k(h_i, h_j|\Theta_b) \triangleq \begin{cases} \theta_0 & if \quad |h_i - h_j| = 0 \\ \theta_1 & if \quad |h_i - h_j| = 1 \\ \theta_2 & if \quad |h_i - h_j| \geq 2 \end{cases} \tag{5.39}$$

where $\theta_0, \theta_1, \theta_2$ are parameters in Θ_b shared for all pair of hidden variables in u and learned with Algorithm 4. The empirical moments μ_k for this function are learned from the training set by analyzing the frequency of each considered case. In summary, we have 5L parameters to estimate during the M-Step, and 3+L parameters in Θ_b to learn.

5.4.4 Initialization and final labeling

In this section we describe the steps required to configure our method for the task of handwritten line segmentation. First, we define the initialization step which is crucial for the good performance of the EM algorithm. Second, we describe the post-process and final labeling.

Initialization The initialization of our method for handwritten line segmentation consist of two steps. In the first place we detect the different text regions or paragraphs



(a) (b)

Figure 5.4: Example of the noise removal step in the initialization of the method. a) Original document image. b) Processed image after the removal of the noise components.

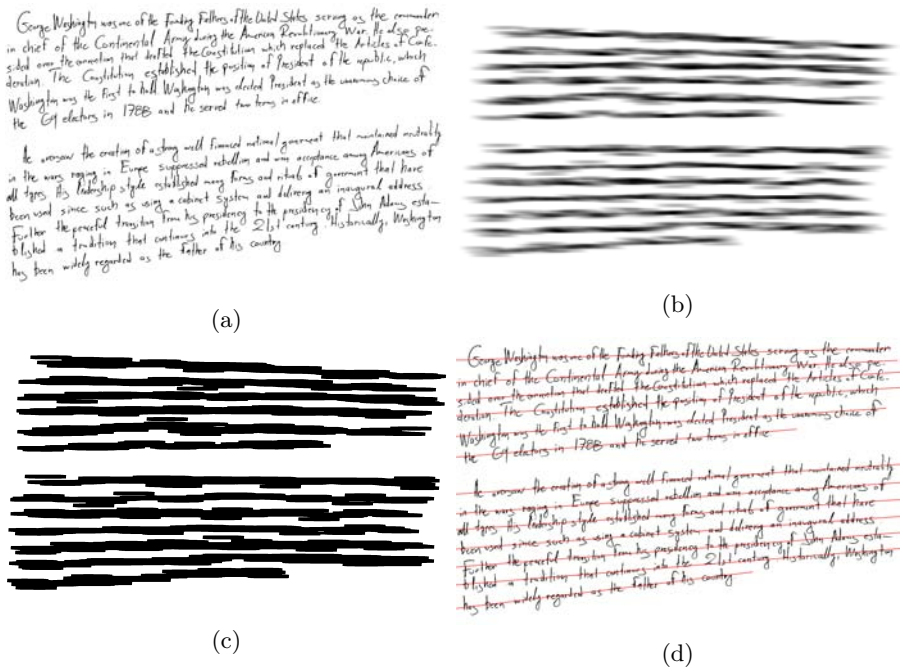
that compose the document image using the method described at the beginning of this chapter. Then, for each of them we initialize the parameters of the regression lines.

Noise removal Noise in documents are usually small connected components that can affect negatively to the line fitting process of our method. Noise elements can be produced in the scanning process, be small ink stains, or residues from previous processes such as machine-printed text separation step. In order to identify and remove noise connected components, we compute the mean area value $\bar{a}_{cc}$ and remove the ones which area is under a certain threshold of the mean t_{area}. In this way we achieve to remove most small components and obtain a cleaner image. During this process we can also remove some diacritics symbols that are part of the text, however, all the components will be assigned to a line in the final process of the method, so it will not be a problem in the final result.

Initial line hypothesis It is known that the EM algorithm is often sensitive to the initial choice of the parameters. An inaccurate initialization of the line parameters may lead the method to fall into a local maximum that do not correspond with the better text line fitting. We combine several common techniques to propose an initial set of regression lines, which will be refined along the iterations of the method.

- Blob estimation: We apply several steps based in the work [126] for skew correction and blob identification. We apply a set anisotropic 2D Gaussian filters of size $W \times H$ on a range of orientations α and select the one with better response on the projection profile. Then, we apply the Otsu binarization method to the filtered image in order to obtain a set of blobs that represent the approximate line locations.

- Overlapping detection: We analyze the obtained blobs in order to detect overlapping as result of touching or curved lines in the document. To do so, we compute the mean connected component height and divide the blobs proportionally to a threshold t_o of this value. Besides, we identify residual blobs result of filtering diacritics or noise components. We compute the ratio of text within



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

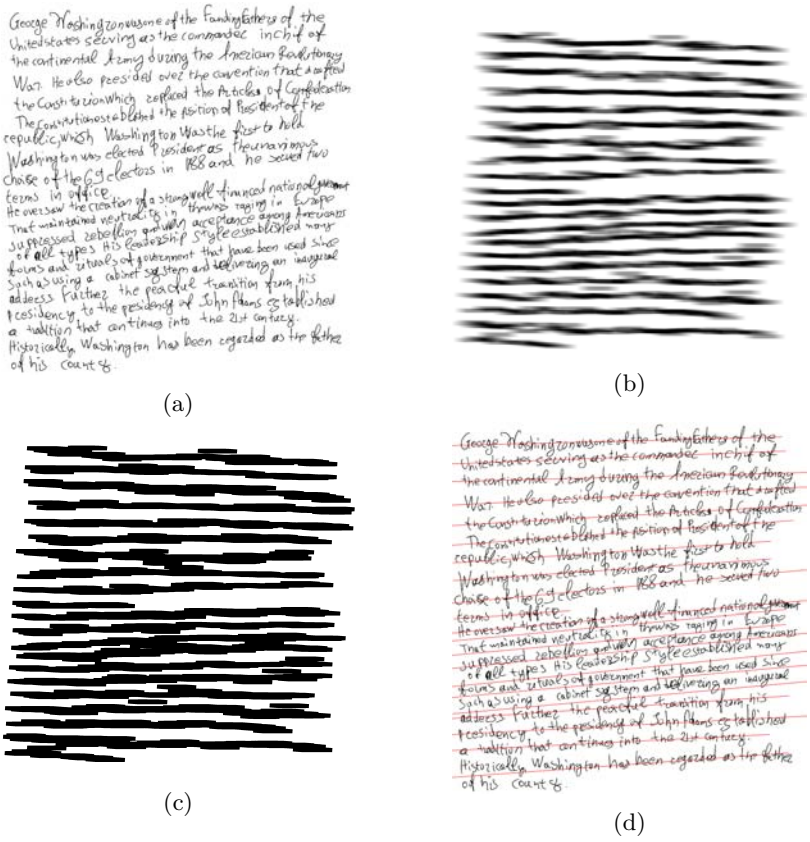
Figure 5.5: Example of an accurate initialization process on an easy document. a) Original image. b) Gaussian filtering. c) Resulting blobs. d) Initial regression lines.

each blob and remove the ones under a threshold t_r learned from the training set.

- Line estimation: The number of resulting blobs define the initial number of candidate lines. For each blob, we estimate the regression line parameters using the common line regression equations on the set of pixels that compose each of them.

As can be seen in the image Fig. 5.5, in documents with simple layouts where lines are properly separated, the initialization itself can be a good result of the detection of the lines. A simple labeling at pixel level by proximity to the line should be enough. In these cases, the execution of our posterior inference process will converge in a few iterations and will be useful to label ascenders and descenders, or other problematic characters. However, in complex documents with crowded or slightly curved text the process is more challenging and the initialization usually is not accurate enough. Fig 5.6 shows an example of a challenging image where only a few initial lines fit exactly the correct text line.

A straightforward consequence of the initialization step is the possible over estimation of lines. It is possible that a line is approximated by two or more initial line fragments. Besides, some diacritics from non-romance languages might be also approximated by a short line segments. This effect is not a drawback for our method,



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

Figure 5.6: Example of non accurate initialization process in a document image with crowded text. a) Original image. b) Gaussian filtering. c) Resulting blobs. d) Initial regression lines.

but the opposite. An initial over-segmentation is recommended, since we need to be sure that we fit the enough number of lines to cover all the text lines. In the case of initializing less than the correct number, some textual components will be probably assigned to the incorrect line, producing several miss detection.

Post-process and final labeling

In the post-process step we analyze the obtained result in order to detect and merge possible fragmented lines and remove surplus ones. After that, we label each of the textual connected components according to the probability given by the MRF model.

- Surplus lines removal: We remove the extra lines remaining after the algorithm

convergence. Extra lines are featured by a low probability close to zero. We detect and remove these lines by identifying the ones which probability is under an ε value fixed beforehand.

- Fragmented lines: The over-segmentation from the initialization step may lead to a fragmentation of a text line. Since our model is linear, the method deals with curved lines by splitting the line into two or more segments. We analyze the relative position between the lines in order to identify these cases and unify the fragments into a single line.

- Final labeling: For each variable e we select the line l that maximizes the probability $p_v(h_v = l|e_v, \Theta)$. We assign the connected component that contains the variable to the line only if all the variables within the component share the same label. Multiple labels in one component usually correspond with touching characters. In that case we label each pixel of the component by distance to the closest regression line.

5.5 Experimental evaluation

In this section we show the set of experiments designed for the task of handwritten line segmentation. We pursue several objectives in this section. In the first place we show qualitative and quantitative results obtained from the two presented approaches. These results allow to compare both methods and evaluate the impact of the inclusion of pairwise relationships as contextual information with respect to the local model. In the second place we prove the generality of our methods to be applied on collections of documents with different layouts and characteristics. For this objective we dispose of several benchmark collections among which are included historical documents, administrative annotated documents, letter-type documents and regular documents with centered text in many different languages. With these experiments we validate the hypothesis that led to the development of this method. Besides, we will discuss the future improvements and research lines opened after this work.

5.5.1 Metrics

Now we describe the set of metrics used to display the results of the different experiments. We adopted the evaluation protocol settled for the successive ICDAR handwritten segmentation competitions as the main evaluation measure to validate our results. This measure is widely used as a benchmark in the document analysis community to evaluate their LS methods, so that we shall be able to compare against the rest of published methods.

The performance evaluation method is based on counting the number of one-to-one (o2o) matches between the detected lines and the ground truth. This comparison is performed by computing the MatchScore table of [127]. The values within this

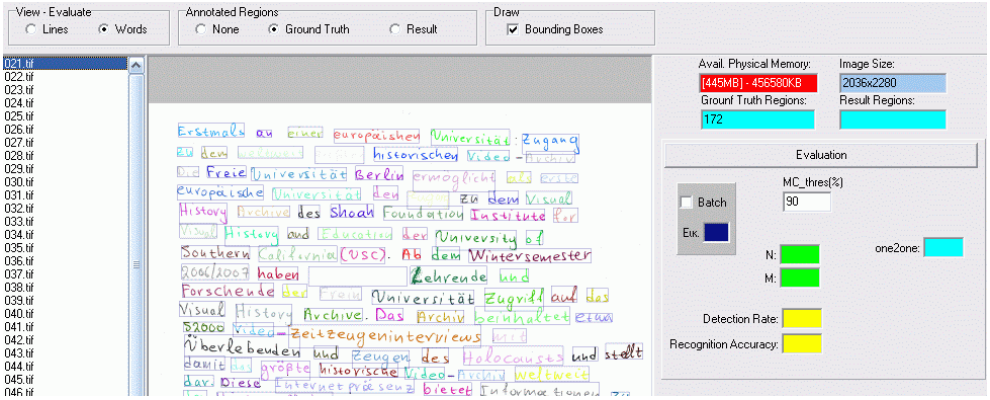


Figure 5.7: Interface of the ICDAR Evaluation tool for handwritten text line and word segmentation.

table measure the correspondence between the set of pixels G_i from a detected region i and a the set of pixels R_j from a region j in the ground truth computed as

$$MatchScore(i,j) = \frac{T(G_i \cap R_j)}{T(G_i \cup R_j)} \tag{5.40}$$

being $T(s)$ a function that counts the elements on each set s. Several indicators are then computed from this table. Being N the count of ground truth elements and M the count of result elements, the values of the Detection Rate (DR%) and Recognition Accuracy (RA%) are computed as

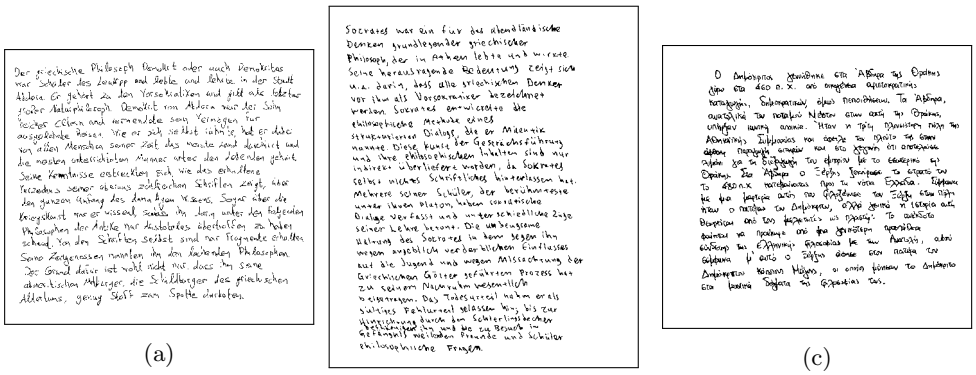
$$DR = \frac{o2o}{N}, RA = \frac{o2o}{M} \tag{5.41}$$

then, the metric F-measure value (FM%) is computed combining the previous values as

$$FM = \frac{2DR \cdot RA}{DR + RA} \tag{5.42}$$

To consider a one-to-one match is required that the matching score is equal or above an acceptance threshold T_a, we use the same threshold than the competitions fixed in $T_a = 95\%$. Figure 5.7 shows the interface of the evaluation tool provided by the competition site.

In the case of the collection annotated administrative documents, since we do not disposed of a ground truth with the characteristics required for this evaluation protocol, we will provide results in terms of RA, DR and FM.



(a) (b) (c)

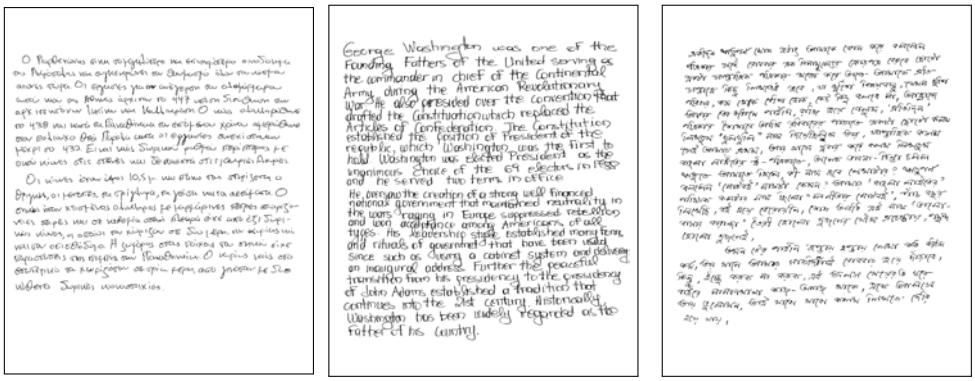
Figure 5.8: Three samples of document images from the ICDAR 2009 HW segmentation contest.

5.5.2 Datasets

Now we present the set of benchmark datasets selected for this experimental evaluation. We chose several collections of documents with different features and difficulties with the objective to prove the adaptability of our method to different types of layouts, languages and font types. Next we describe the main features of each dataset in detail.

ICDAR 2009 dataset
This dataset is composed of a set of binary scanned handwritten documents created by different writers. For the construction of the dataset each writer was ask to write the same extract of text according to their own writing style and language. The set of languages includes English, German, Greek and French. The full dataset contains 300 document images, which are divided in two partitions for training (100 images) and test (200 images). The test partition contain a total of 4043 text lines. The layout of the documents consist of a central paragraph of text and free of graphical or non-textual elements, although some of them may contain some small noise components. In general the lines are well separated with the exception of some touching characters in some documents. Only a few documents present multi-skewed text and light curvatures. Figure 5.8 shows some examples of documents.

ICDAR 2013 dataset
This dataset was an update of the previous one with the objective to include documents with an additional complexity. This dataset is composed of 300 images divided in 150 images for training and 150 images for test, being this last set composed of 2649 text lines. The layout of the documets is similar than the previous one, although it contains more complex and cramped documents as well as some multi-paragraph



(a) (b) (c)

Figure 5.9: Three samples of document images from the ICDAR 2013 HW segmentation contest.

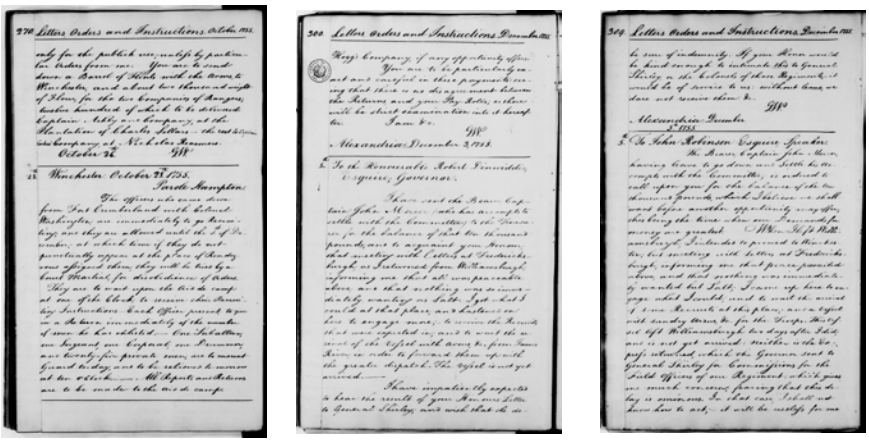
and multi-skewed documents. The main difference relies in the inclusion of new complex documents written in Indian Bangla with different content and document sizes. Figure 5.9 shows some examples of documents from this dataset.

George Washington dataset

This dataset is composed of 20 grey-level images from the George Washington Papers at the Library of Congress dated from the 18th century [128]. The documents are written in English language in a longhand script from two different writers. This database adds a set of different challenges with respect to the previous one due to the old script style, overlapping lines and a more complex layout. Also, the documents may contain non-text elements as stamps or line separators. We show several examples in Figure 5.10. For this dataset there is not a defined ground truth for the task of line segmentation, therefore we have manually created our own specially for this task and use the same ICDAR evaluation protocol. For this reason, it is not possible to compare our results with any other methods apart from previous works and the work in [86]. We present the results as an indicator of the adaptability of our method to historical documents.

Annotated administrative documents

Last, we have tested our method in a collection of administrative documents with handwritten annotations. This is a more heterogeneous and complex dataset, since contains documents with multiple text regions, each of them with different characteristics as orientation and writing style. The collection includes letter-type documents, annotations in machine-printed documents, information from bank checks and other irregular documents. The set of documents on which our method will be applied is the result of the application of a previous machine-printed text separation [129], in order to remove all possible not handwritten components. The dataset is written in



(a) (b) (c)

Figure 5.10: Three samples of document images from the George Washington dataset.

English and French languages and is composed by 433 document images. We show some examples of documents in Fig 5.11.

5.5.3 Parameters and settings

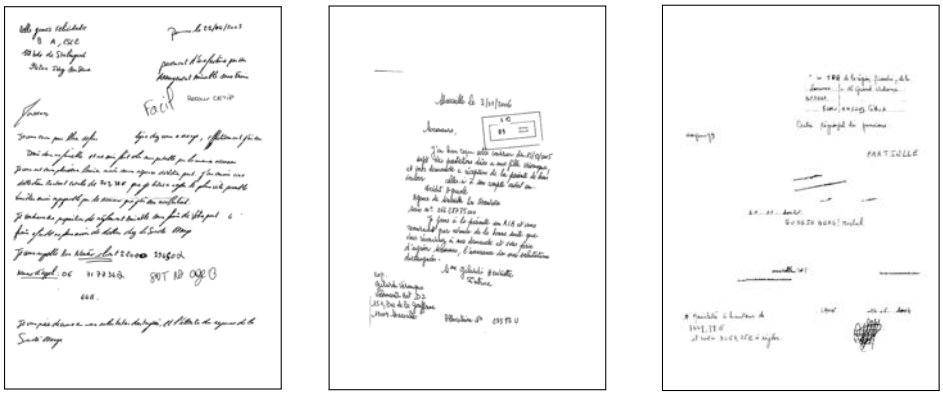
We defined in our two method proposals several parameters required in the different steps of the methods. In this section we describe each of them in detail as well as other settings needed for the set of experiments.

Gaussian model

In the case of the Gaussian model we only required to set the stop criteria of the EM algorithm. On the one hand we fixed the maximum number of iterations of the algorithm to 200. We empirically set this value to ensure a proper adjustment of the regression lines, since in the majority of the cases the convergence was reached under this value. On the second hand we established as the second convergence criterion the Kullback-Leibler divergence between the probability distributions computed on two consecutive iterations. We have considered a convergence value of $\epsilon = 10^{-4}$.

Pairwise model

For this model we dispose of several parameters for the successive steps of our algorithm. In the first place, in the initialization step we use a value for $t_{area} = 0.2 * \bar{a}_{cc}$ as threshold to filter up noisy components. Since we only want to perform a coarse filtering, this simple threshold achieves to remove most noise components and avoids to confuse the EM regression process. The removed components at the end of the



(a) (b) (c)

Figure 5.11: Three samples of document images from the dataset of annotated administrative documents.

process will be labeled in the same way than the rest.

In the second place, for the estimation of the initial line hypothesis we apply a skew estimation and blob detection based in [126], although we changed some parameter values according to the performed experiments. We apply a set of Gaussian filters with different orientations in the range of $[-40, 40]$ degrees, and a filter size of $\frac{1}{3}H_{cc} \times 10W_{cc}$ with a vertical and horizontal standard deviation of $\frac{1}{3}H_{cc}$ and $\frac{10}{3}W_{cc}$, respectively.

In the third place, in what regards to the configuration of the MRF we learned the parameters of the pairwise function from the training set of the ICDAR datasets. We consider this dataset as an accurate sample of common handwriting script, and therefore we use the same parameter values for the other experiments.We empirically validated that updating these parameters along the iterations of the method did not significantly improve the segmentation results, but the updating process increased the processing time. Therefore for all the shown experiments we maintained the same learned parameters.

Last, for the parameters of the fitting function in Eq. (5.37) we have used a ratio value $r = 0.3$ and a value for $\sigma = 0.1$ for the pixels outside this range. The maximum number of iterations of the EM was fixed to 50, with an additional stopping criterion based in a threshold of the Kullback-Leibler divergence (KLD) between two consecutive iterations distributions iterations of 1×10^{-4}.

5.5.4 Experiments

Random pixel selection

We aim at analyzing the impact of the density of random text pixels selected for the construction of the graphical model. We conduct this experiment on the ICDAR

2013 dataset for a pixel ratio of $1\%, 3\%, 5\%, 10\%$ and 15% of the total amount of text pixels. Table 5.2 shows the obtained results in terms of the F-measure, mean processing time, and its corresponding confidence intervals.

We see that using values above 5% does not produce significant improvements in the results, while the computational complexity increases considerably due to the large number of variables and connections in the MRF model. With a 1% of pixels, we obtain a 95.52% in almost four times less computational time compared to 5%. However, the reduction in the number of pixels may leave some text regions uncovered, which can lead to an incorrect segmentation. Besides, we observe that the confidence interval for higher number of pixels increases. This implies that the method becomes less stable. For the rest of experiments we select a 5% of text pixels as standard value, since it seems to provide a good trade-off between data representation and time complexity.

(%) of points	FM(%)	Time(mean)
1%	95.52 ± 1.46	$12.6\text{s} \pm 1.1$
3%	96.95 ± 1.24	$28.4\text{s} \pm 2.7$
5%	97.05 ± 1.17	$42.4\text{s} \pm 3.3$
10%	97.05 ± 1.18	$81.3\text{s} \pm 7.7$
15%	97.05 ± 1.25	$115.1\text{s} \pm 11.2$

Table 5.2: Results for different percentage of random points selected for the construction of the graphical model.

ICDAR 2009 dataset

We show in Table 5.3 the results obtained on the ICDAR 2009 Handwriting Segmentation dataset. We obtained an 95.20% FM for our first method using only local features and the set of Gaussian distributions as the probabilistic framework. We see as our method overcomes some of the other contestant methods, while there is still place for improvement in order to reach the top results.

We performed an exhaustive analysis of the results an identified three main error sources that we describe next. In the first place we identified a problem in the detection of short text lines usually located at the end of the document. This error comes directly from the estimation of the line probability in (5.11). Since the number of connected components that compose this type of lines is lower than the number of components from larger text lines, the computed probability is significantly smaller that the corresponding probability for the other regression lines. This promotes an snowball effect in the rest of iterations where this line probability becomes smaller and finally disappear from the computations. An example of this error is shown in Figure 5.12. We see as the last text line is not fitted by any regression line, and consequently its text components are be assigned to the closest regression line. The penalization of this situation in the evaluation protocol is twofold, since it affects

Method	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
HandwritingPAIS	4031	3973	98.49	98.56	98.52
CMM	4044	3975	98.54	98.29	98.42
Fernandez *et al.* [86]	4176	3971	98,40	95,00	96,67
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	37.20
Gaussian	**4061**	**3858**	**95.60**	**95.00**	**95.20**
Pairwise	**4044**	**3986**	**98.81**	**98.56**	**98.68**

Table 5.3: Results on the ICDAR2009 Handwriting Segmentation Contest [2]

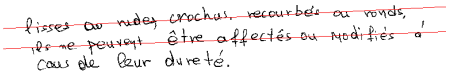


Figure 5.12: Example of excluded short lines at the end of a page.

to two o2o associations and the total number of lines is not correct. We estimate that this error correspond almost to the 40% of the reported errors, so the method performance should be significantly better once this error is fixed.

The second error source is related to the over-estimation of the number of lines in the initialization step. As a result of this we have observed two possible situations. The first one refers to extra lines that end up fitting some ascenders and descenders located between two text lines correctly estimated by other regression lines. This situation slightly affect the quantitative results by adding and extra detection and in some cases affecting to some o2o associations. The effect in the qualitative results may be worst since all the ascenders and descenders from a text line could be assigned to the extra line and in this way hinder the possible recognition of the text line. The second situation caused by this type of error occurs when the extra line crosses and intersects several other regression lines (see Figure 5.13 for an example). In this situation the incorrect line is trying to fit isolated text components from diacritic symbols or noise. At the end of the algorithm these type of lines get high probability due to the fitted components are not being assigned to any other line, and therefore, is not removed in the post-process step. This situation affects to both quantitative and qualitative results depending on how many text components are fitted by the incorrect line. We estimated this represent around the 50% of the reported errors.

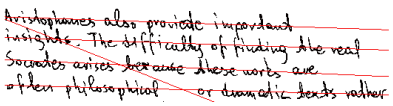


Figure 5.13: Example of a crossing line result of over-estimating the number of lines.

Last, we have grouped a set of not so common mistakes in the third group of errors. This type of mistakes represent around the 10% of the reported cases and are directly related with an under-estimation of the number of lines. Due to this problem, some lines may end up the process not being fitted by any regression line, or in the worst case, sharing a regression line with another text line, which results in the lose of two text lines.

The second proposed method based in the PGM have proven to overcome most of the issues of the first proposal. We obtain a 98.68% FM value, with a confidence interval $[98.23, 99.13]$. This result compares with the top methods of the competition and overcomes the result obtained by the GMM approach. In addition, the analysis of the results shows that 166/200 images reach 100% of FM, while the main errors are concentrated in a few error cases. This result demonstrates the contribution of the MRF model in the incorporation of contextual information and the improvement in the initialization and post-process step.

Despite of the improvements in the post-process step to remove extra lines, one of the most recurrent error sources comes again from the over-segmentation and non-removed extra lines. In this case the extra lines often fits small diacritics or noise, and are usually reduced to unique text connected components and in general the effect on the other lines is smaller compared with the previous method. In this case the quantitative results are penalized because of the extra detection, however, the effect in the qualitative results has been reduced and, in views to the recognition of the text, this issue should not affect to the global recognition, since the rest of the lines are correctly segmented. An example of an annoying extra line can be seen in Figure 5.14, although in the majority of the cases this is corrected properly in the post-process, this remaining errors affect to several one-to-one associations.

Another type of error observed is produced in reduced areas with several touching characters from consecutive lines. In those cases the connectivity in the graph between these characters may be high in comparison with other non-touching characters, and therefore the exchanged messages may favor the same labeling for all the text component instead of split between lines. An example of this effect can be seen in Figure 5.15.

ICDAR 2013 dataset

In Table 5.4 we show the result obtained on the ICDAR 2013 Handwriting Segmentation dataset. The additional complexity of this dataset is reflected in the results, where we obtain a 90.16 and a 97.05% of F-Measure for our Gaussian and Pairwise

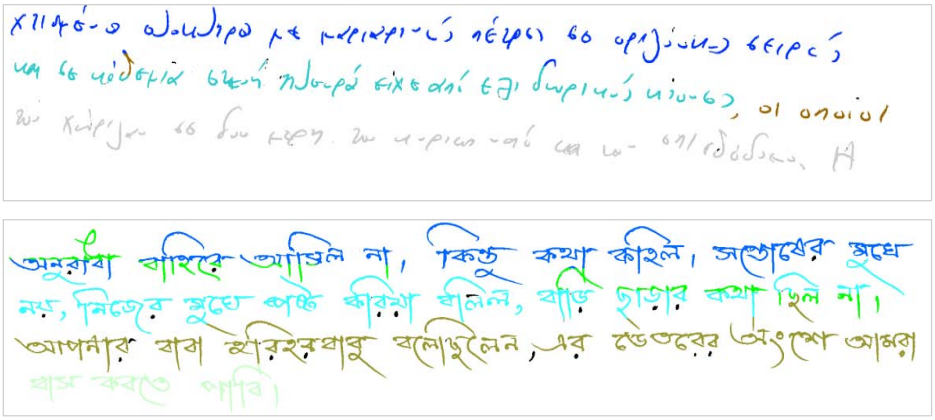


Figure 5.14: Two examples of extra lines not removed in the post-process step.

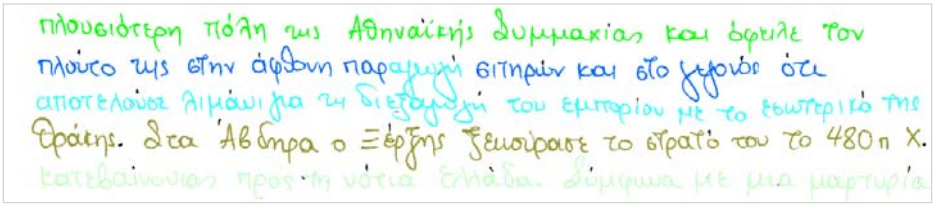


Figure 5.15: Segmentation error produced when several characters from the same word are highly connected with another text line. The messages sent between variables from these words favor the same labeling for every component in conflict.

approaches, respectively. In this case the behaviour of our first approach was similar than for the previous collection, and the new characteristics of the dataset stand out the weaknesses of this approach. We carry out the analysis of the results according to the results obtained by the Pairwise method.

In this case the errors found are also similar than the ones obtained in the previous experiment, although we observed some new situations due to the new sources of document layouts. In general, documents with cramped lines are well addressed by the detection of overlapped lines or the overestimation from the initialization step. In these cases it is possible that we initialize a line that covers two or more text lines in the document (see Figure 5.6). If this is not detected and we have initialized some extra lines, some of them usually end-up fitting the text components of the overlapped lines. However, there are still some cases where our method is not able to separate text lines that are too overlap as can be seen in the bottom example in Figure 5.16. In addition, our method is able to deal with light curvatures of the text by splitting the line in fragments or because the connectivity of the curved area favors it. However, in some cases as the top example in Figure 5.16 the linear approach of our regression is not enough and ends-up fitting components from other lines. Nevertheless, in the

Method	M	o2o	DR(%)	RA(%)	FM(%)
INMC	2614	2614	98.68	98.64	98.66
NUS	2645	2605	98.34	98.49	98.41
GOLESTAN-a	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
LRDE	2632	2568	96.94	97.57	97.25
Fernandez *et al.* [86]	2697	2551	96,30	94,58	95,43
QATAR-b	2609	2430	91.73	73.14	92.43
MSHK	2696	2428	91.66	90.06	90.85
Gaussian	**2715**	**2418**	**91.28**	**89.06**	**90.16**
Pairwise	**2647**	**2570**	**97.01**	**97.09**	**97.05**

Table 5.4: Results on the ICDAR2013 Handwriting Segmentation Contest [3]

Method	M	o2o	DR (%)	RA (%)	FM (%)
Fernandez *et al.*[86]	693	653	91,30	94,20	92,70
Cruz *et al.* [130]	631	551	82,60	87,30	84,80
Base line [86]	727	338	47,20	46,40	46,70
Proposed	**702**	**614**	**92.05**	**88.16**	**90.06**

Table 5.5: Results on the George Washington dataset.

practice our method is able to deal with the majority of these situations as seen in Figure 5.19.

In comparison with the rest of the methods we see as our results are slightly below the top methods in quantitative terms. We could perform a more exhaustive comparison if we had per image results, however we can provide some additional information in views of future comparisons. For instance, we obtained a total of 125/150 images labeled with a 100% of FM, while for the rest of images the 80% of the errors are related with over-segmented lines and extra lines fitting isolated components. In views of text recognition this errors should not be a problem to obtain a correct result. The other 20% or error sources are related with the overlapping of very close lines and some other problems arising from crowded areas of text (Figure 5.16). In addition, we also provide the confidence interval as $97.05\% \pm 1.25$ for an $\alpha = 0.05$.

George Washington database

Results obtained on the George Washington database are shown in Table 5.5. We performed this experiment using the same configuration than in the previous ones, the idea is to evaluate the adaptability to other datasets without parameter tuning or training. In addition, we have not taking into account some special characteristics

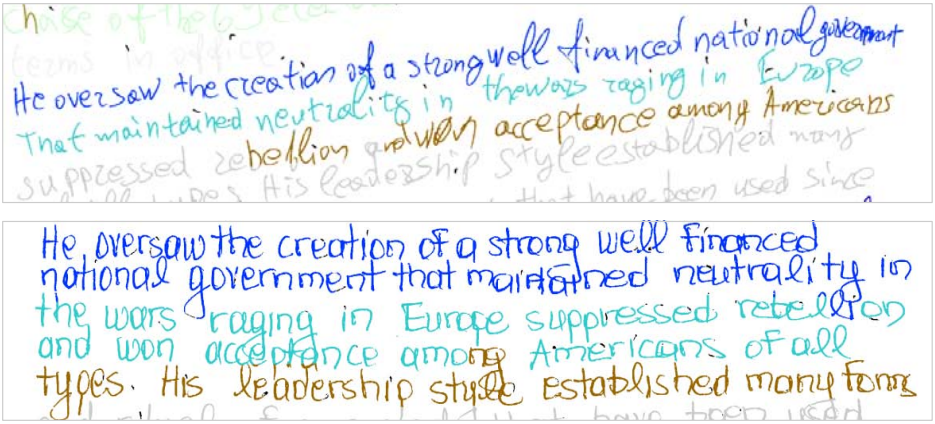


Figure 5.16: Two examples of having extra lines.

of this dataset that could be integrated into the model for a better performance.

Some of this characteristics are for example the non-textual elements. Some images contain stamps, separator lines between paragraphs or underlines, which detection may affect to the numerical results. These underlines are usually labeled in the ground truth as a component of the upper text line, while our methods could recognize it as a separated text line (see Figure 5.17). The same effect happens in some arbitrary separator lines, signatures of footnotes, which are labeled in the ground truth as a part or another line or as an independent line. This labeling issue, although does not have an impact in the qualitative results, and in an hypothetical text recognition, it does affect to the numerical result, since implies several one-to-one bad associations and an excess of the number of segmented lines. In addition, to process this documents we previously had to apply a binarization step to extract the connected components of text. We used the common Otsu [131] binarization technique, which incorporates some noise into the image that may affect to the overall result.

We obtained an 84.8% of FM value for our Gaussian model. The same problems observed in the experiments on ICDAR dataset are observed here, although they are concentrated in the cases described below. For instance, is common that two regression lines overlap and end up fitting the same line apart from the line that were supposed to fit. An example of this error is shown in Figure 5.18. This effect was handled by the graphical model by the incorporation of the pairwise information. These relationships force the model to not cross and therefore corrects this type of error. In addition, the new representation in line segments is able to fit smaller lines even if they are aligned in a different way in any location of the document. Overcoming this problem led to a 90.06% of FM value with an increase of the DR of almost a 10%. The labeling problem of underlines and similar structure still affects to the quantitative results, although the separate detection of these lines is not an issue in views of recognition.

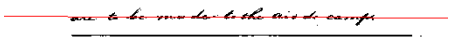


Figure 5.17: Example of separator line included in a wrong regression line.

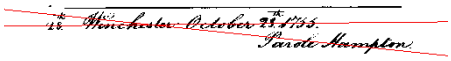


Figure 5.18: Two regression lines trying to fit on connected components from different lines.

Administrative annotated documents

As for the GW dataset, we use the same parameter configuration than for the ICDAR experiments. In this dataset one of the challenges is to detect the different text regions in order to process them separately. For instance, in Figure 5.11a we can appreciate at the bottom of the central block of text three text lines with different font that have to be labeled separately. We can see another example in Figure 5.11b, where several lines at the bottom of the document may be merged as the same line in the case of processing the full page. Our method behaves on these cases on two possible ways. In most of the cases the different text regions are detected in the initialization step and then processed separately. However, in some documents where the region segmentation is not achieved, text lines are approximated by several regression lines due to the initial over-segmentation.

On this experiment we are not able to compare to other works, since it is a non-published collection of documents, however we can compare against our previous work in order to validate the new model. Table 5.6 shows the results obtained. We see that the results are significantly improved with the new proposed approach. The observed improvement confirm the contribution of the proposed model. In addition, the result on this dataset proves the versatility of our method on complex layouts.

The main sources errors in this dataset are again related with the effect of over-segmentation of some of the text lines, and the labeling of residual non-textual elements as lines, which reduces the accuracy of the method. However, as in the previous dataset, this effects should not have a big impact on a posterior recognition task.

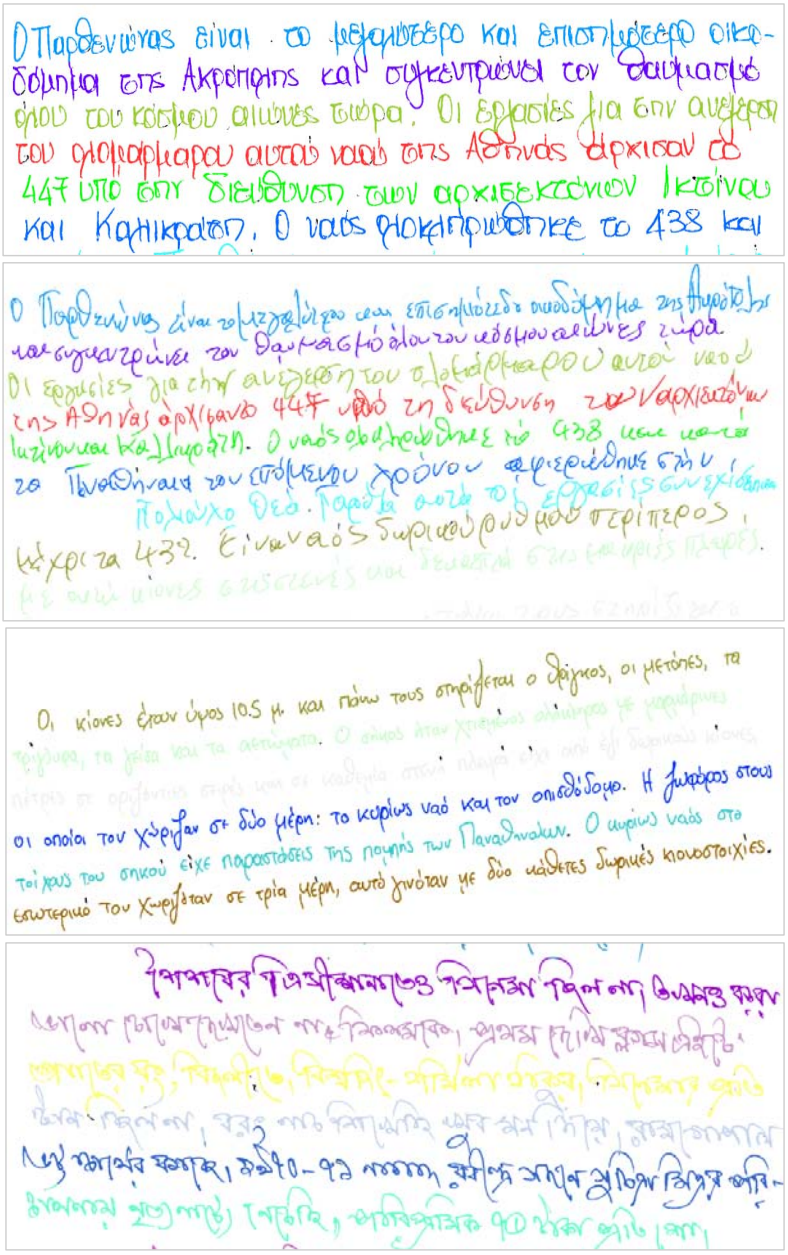


Figure 5.19: Results on four fragments of images from the ICDAR 2013 dataset that include crowded text and light curvatures.

Method	Precision	Recall	FM(%)
Gaussian	69.45	72.38	70.88
Pairwise	**79.75**	**82.70**	**81.19**

Table 5.6: Results on the administrative handwritten annotation dataset.

5.6 Conclusion

In this chapter we present two models for the task handwriting text line segmentation based on the estimation of a set of regression lines. Our models rely on the EM algorithm for the estimation of the regression parameters, and diverge in the probability distribution that model. First, we propose a GMM where each component of the mixture represents a regression line. Second, we propose a probabilistic framework that relies on a MRF model for parameter learning of neighboring pixels. We implement a message-passing-based algorithm to compute approximate inference and learning the model parameters. Our methods can be applied on documents with different layouts and features. Besides, our framework permits to easily extend the model with the inclusion of prior information by means of new feature functions.

We conduct several experiments with promising results on four collections of documents without model reconfiguration. The selected datasets include several types of layouts, historical an contemporary documents, from several writers and scripts. Besides, our method is able to deal with most of the situations regarding touching text lines and light curvatures of the text, which demonstrates the contributions of the proposed model. The results validate our initial hypothesis, since we prove that a set of regression lines can fit with high accuracy the actual text lines locations.

As future work lines, we consider to use higher order regression model that could lead to better approximation of curved and complex lines. Besides, we think that some of the current errors may be corrected with the inclusion of more informative feature functions. Note that we use a reduced and basic set of feature functions based on the pixel location and common pairwise interactions. We believe that the inclusion of more specific knowledge could improve the overall results. For instance, we can incorporate improved pairwise feature that analyzes the edge length and relative position between the connected variables. In addition, we plan to integrate new discriminant features in order to perform machine-printed/handwritten text separation and line segmentation within the same process. In this way we will be able to process administrative annotated documents directly without the need of previous steps.

Chapter 6

Conclusions and Future Work

In this thesis we have tackled the problem of document layout analysis and the specific case of handwritten text line segmentation. We proposed several methods based on probabilistic graphical models and other sources of contextual information for this purpose, and we have evaluated them on several collections to prove our hypotheses, and compare us against other works in the field. In this section we highlight the main conclusions extracted from the work presented in this thesis, and describe the research lines opened as a result of the performed contributions.

6.1 Summary and future work

Layout analysis

In what regards to the chapter of layout analysis, we wanted to analyze and develop methods that account for contextual relations. In order to define these relations we relied on both, the definition of the task, and the implicit characteristics of the datasets to which our methods were addressed. We presented several methods and statistical resources for this task, and performed a series of experiments to validate our hypotheses and contributions. In the following, we summarize the main conclusions extracted from this work.

We incorporated relative location prior into the models through the set of RLF. This is a good resource to introduce this kind of contextual information, since the formulation as a features allows to incorporate them within any model. We performed experiments to validate the contribution of these features to the different tasks, and concluded that its use produces significant improvements on the detection of the regions from structured documents. This was an expected behavior that was confirmed by our experiments. In the case of non-structured documents, the global effect of

these features was only significant, from a statistical point of view, in the detection of common regions, such as text regions. In less common regions, the probability maps, and therefore the set of features, is not informative enough to improve its recognition. Another way to validate the contribution of these features is analyzing its role through the different methods. We saw as the inclusion of RLF on the 2D-PCFG did not produce any improvement, quite the opposite, since the model already accounted for the relative location of the regions in the document. However, in the other models without an explicit representation of the document structure, the contribution of RLF was determinant.

We presented several methods for the task of layout analysis, each of them devised to encode some kind of contextual information. Our model based on CRF was designed to integrate the set of RLF and to encode pairwise contextual relationships between neighbor variables in the graph. The effect of these relationships was smoother regions in the final segmentation. We compared several inference algorithm, and conclude that the $\alpha - \beta - swap$ version of Graph-Cuts was the method that offer better response for this task. We presented a statistical method for layout analysis on partially structured documents. We modeled relations between regions and model variables by means of a Bayesian network, which allowed us to compute the probability of a particular segmentation result and select the best among several iterations of the method. Besides the EM-based algorithm proved to be able to account for changes in the location and size of the regions with respect to the learned structure. We obtained remarkable results with our method based on 2D-PFCG for layout analysis on the BH2M dataset. The proposed grammar accounted for the structure of the document composed of sets of records. However, the method was highly expensive in computational time, and therefore it is not clear if it can be used on practical applications.

Handwritten text line segmentation

In what regards to the handwritten line segmentation task, we presented two approaches based on the hypothesis of fitting a set of regression lines through the text line components. Our models rely on the EM algorithm for the estimation of the regression parameters, and diverge in the probability distribution that model.

First, we proposed a simple probability distribution based on a set of Gaussian distributions where each component of the mixture represented a regression line. With this model we demonstrated that our hypothesis was correct, and obtained good results on several benchmark datasets. However, this model was defined on local information at pixel level and we considered that using contextual information may lead to the correction of some of the detected error. This was the main motivation to develop the second proposed approach. We proposed a probabilistic framework that relied on a MRF. We successfully combined the EM process with an inference algorithm and a parameter learning process. Through this model we incorporated several types of contextual relationships between variables and between lines, that lead to significant improvements in all the datasets.

Our method can be applied on documents with different layouts and features. Besides, our framework permits to easily extend the model with the inclusion of prior information by means of new feature functions. The experiments performed aim to demonstrate also the generality of our method. We selected datasets that include several types of layouts, historical an contemporary documents, from several writers and scripts. We proved that our method is able to deal with most of the situations regarding touching text lines and light curvatures of the text, which demonstrates the contributions of the proposed model. The results validate our initial hypothesis, since we prove that a set of regression lines can fit with high accuracy the actual text lines locations.

As a consequence of this work, we considered that several research lines for future work have been opened:

- The transition to a higher order regression model could lead to a better approximation of curved and complex lines.
- The model is easy to extend with additional feature functions. The research of more informative feature functions, or the use of additional classifiers or other machine learning techniques to encode information through these function may lead to great improvements of the results.
- Several tasks of document analysis can be also included in the method through the set of functions. We consider that could be possible to perform machine-printed/handwritten text separation and line segmentation within the same process. Which would be of great interest for business-oriented applications.

List of Publications

Journals

- Francisco Cruz and Oriol Ramos Terrades, A probabilistic framework for handwritten text line segmentation. *Submitted to Pattern Recognition, July 2016.* (Q1)
- Francisco Álvaro, Francisco Cruz, Joan-Andreu Sánchez, Oriol Ramos Terrades, José-Miguel Benedí. Structure detection and segmentation of documents using 2D stochastic context-free grammars, Neurocomputing, Volume 150, Part A, 20 February 2015, Pages 147-154. (Q1)

International Conferences

- ICPR Francisco Cruz and Oriol Ramos Terrades. Document segmentation using relative location features. In *21st International Conference on Pattern Recognition (ICPR)*, pp. 1562-1565, 2012. (Oral)
- IBPRIA Francisco Álvaro, Francisco Cruz, Joan-Andreu Sánchez, Oriol Ramos Terrades and Jose Miguel Benedí. Page Segmentation of Structured Documents Using 2D Stochastic Context-Free Grammars. *In 6th IbPRIA*, Vol. 7887, pp. 133-40. (Oral)
- ICDAR Francisco Cruz and Oriol Ramos Terrades. Handwritten Line Detection via an EM Algorithm. *In 12th International Conference on Document Analysis and Recognition (ICDAR)*, pp. 718-722, 2013. (Poster)
- ICPR14 Francisco Cruz and Oriol Ramos Terrades. EM-based Layout Analysis Method for Structured Documents. *In 22nd International Conference on Pattern Recognition (ICPR)*, 2014. (Poster)

Bibliography

[1] A. Antonacopoulos, D. Bridson, C. Papadopoulos, and S. Pletschacher, "A realistic dataset for performance evaluation of document layout analysis," in *International Conference on Document Analysis and Recognition*, pp. 296–300, 2009.

[2] B. Gatos, N. Stamatopoulos, and G. Louloudis, "ICDAR 2009 handwriting segmentation contest," in *International Conference on Document Analysis and Recognition*, pp. 1393 –1397, 2009.

[3] N. Stamatopoulos, B. Gatos, G. Louloudis, U. Pal, and A. Alaei, "ICDAR 2013 handwriting segmentation contest," in *International Conference on Document Analysis and Recognition*, pp. 1402–1406, Aug 2013.

[4] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, pp. 1222–1239, Nov. 2001.

[5] P. Lyman and H. R. Varian, ""how much information", 2003. technical report retrieved from http://www.sims.berkeley.edu/how-much-info-2003 on 04/09/2016.," tech. rep., 2003.

[6] A. Esteve, C. Cortina, and A. Cabré, "Long term trends in marital age homogamy patterns: Spain, 1992-2006," *Population*, vol. 64, no. 1, pp. 173–202, 2009.

[7] A. Oliva and A. Torralba, "The role of context in object recognition," *Trends in Cognitive Sciences*, vol. 11, no. 12, pp. 520–527, 2007.

[8] B. Moshe, "Visual oobject in context," *Nature Reviews Neuroscience*, vol. 5, pp. 617–629, 2004.

[9] D. P. A. G. A. H. Joshua O. S. Goh, Soon Chun S. and M. W. L. Chee, "Cortical areas involved in object, background, and object-background processing revealed with functional magnetic resonance adaptation," *The Journal of Neuroscience*, vol. 45, no. 24, pp. 10223–10228, 2004.

[10] G. Nagy, "Twenty years of document image analysis in pami," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 38–62, Jan 2000.

[11] J. Liang, I. T. Phillips, and R. M. Haralick, "A statistically based, highly accurate text-line segmentation method," in *Proceedings of the Fifth International Conference on Document Analysis and Recognition*, pp. 551–554, Sep 1999.

[12] R. Plamondon and S. Srihari, "Online and off-line handwriting recognition: a comprehensive survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 63–84, Jan 2000.

[13] L. Likforman-Sulem, A. Zahour, and B. Taconet, "Text line segmentation of historical documents: a survey," *International Journal on Document Analysis and Recognition*, vol. 9, pp. 123–138, Apr. 2007.

[14] H. Fujisawa, Y. Nakano, and K. Kurino, "Segmentation methods for character recognition: from segmentation to document structure analysis," *Proceedings of the IEEE*, vol. 80, pp. 1079–1092, Jul 1992.

[15] R. M. Haralick, "Document image understanding: Geometric and logical layout," in *Conference on Computer Vision and Pattern Recognition*, pp. 385–390, IEEE, 1994.

[16] R. Cattoni, T. Coianiz, S. Messelodi, and C. M. Modena, "Geometric layout analysis techniques for document image understanding: a review," tech. rep., 1998.

[17] A. K. Jain and B. Yu, "Document representation and its application to page decomposition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 294–308, Mar 1998.

[18] S. Mao, A. Rosenfeld, and T. Kanungo, "Document structure analysis algorithms: A literature survey," 2003.

[19] A. Antonacopoulos, B. Gatos, and D. Bridson, "Icdar 2005 page segmentation competition," in *International Conference on Document Analysis and Recognition*, vol. 1, pp. 75 – 79, 2005.

[20] A. A., B. Gatos, and D. Bridson, "Icdar 2007 page segmentation competition," in *International Conference on Document Analysis and Recognition*, vol. 2, pp. 1279–1283, 2007.

[21] A. Antonacopoulos, C. Clausner, C. Papadopoulos, and S. Pletschacher, "ICDAR 2013 Competition on Historical Newspaper Layout Analysis (HNLA 2013)," in *International Conference on Document Analysis and Recognition*, pp. 1454–1458, 2013.

[22] A. Antonacopoulos, C. Clausner, C. Papadopoulos, and S. Pletschacher, "ICDAR 2013 Competition on Historical Book Recognition (HBR 2013)," in *International Conference on Document Analysis and Recognition*, pp. 1459–1463, 2013.

[23] K. Y. Wong, R. G. Casey, and F. M. Wahl, "Document analysis system," *IBM Journal of Research and Development*, vol. 26, pp. 647–656, Nov 1982.

[24] J. L. Fisher, S. C. Hinds, and D. P. D'Amato, "A rule-based system for document image segmentation," in *Pattern Recognition, 1990. Proceedings., 10th International Conference on*, vol. 1, pp. 567–572, IEEE, 1990.

[25] F. Esposito, D. Malerba, G. Semeraro, E. Annese, and G. Scafuro, "An experimental page layout recognition system for office document automatic classification: an integrated approach for inductive generalization," in *International Conference on Pattern Recognition*, vol. 1, pp. 557–562, IEEE, 1990.

[26] T. Akiyama and N. Hagita, "Automated entry system for printed documents," *Pattern Recognition*, vol. 23, no. 11, pp. 1141 – 1154, 1990.

[27] L. O'Gorman, "The document spectrum for page layout analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 11, pp. 1162–1173, 1993.

[28] K. Kise, A. Sato, and M. Iwata, "Segmentation of page images using the area voronoi diagram," *Computer Vision and Image Understanding*, vol. 70, no. 3, pp. 370 – 382, 1998.

[29] L. Fletcher and R. Kasturi, "A robust algorithm for text string separation from mixed text/graphics images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, pp. 910–918, Nov 1988.

[30] A. Simon, J.-C. Pret, and A. P. Johnson, "A fast algorithm for bottom-up document layout analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 273–277, Mar. 1997.

[31] C. An, H. Bird, and P. Xiu, "Iterated document content classification," in *International Conference on Document Analysis and Recognition*, pp. 252–256, 2007.

[32] G. Nagy, S. Seth, and M. Viswanathan, "A prototype document image analysis system for technical journals," *Computer*, vol. 25, pp. 10–22, July 1992.

[33] J. Ha, I. T. Phillips, and R. M. Haralick, "Document page decomposition using bounding boxes of connected components of black pixels," in *Book*, vol. 2422, pp. 140–151, 1995.

[34] P. Parodi and G. Piccioli, "An efficient pre-processing of mixed-content document images for ocr systems," in *International Conference on Pattern Recognition*, vol. 3, pp. 778–782 vol.3, Aug 1996.

[35] K. S. Baird, "Anatomy of a versatile page reader," *Proceedings of the IEEE*, vol. 80, pp. 1059–1065, Jul 1992.

[36] M. Okamoto and M. Takahashi, "A hybrid page segmentation method," in *International Conference on Document Analysis and Recognition*, pp. 743–746, Oct 1993.

[37] R. W. Smith, "Hybrid page layout analysis via tab-stop detection," *2013 12th International Conference on Document Analysis and Recognition*, vol. 0, pp. 241–245, 2009.

[38] K. Chen, F. Yin, and C.-L. Liu, "Hybrid page segmentation with efficient whitespace rectangles extraction and grouping," in *International Conference on Document Analysis and Recognition*, pp. 958–962, IEEE, 2013.

[39] D. X. Le, G. R. Thoma, and H. Wechsler, "Classification of binary document images into textual or nontextual data blocks using network models," *Mach. Vision Appl.*, vol. 8, pp. 289–304, Oct. 1995.

[40] R. Sivaramakrishnan, I. T. Phillips, J. Ha, S. Subramanium, and R. M. Haralick, "Zone classification in a document using the method of feature vector generation," in *International Conference on Document Analysis and Recognition*, vol. 2, pp. 541–544, IEEE, 1995.

[41] A. K. Jain, N. K. Ratha, and S. Lakshmanan, "Object detection using gabor filters," *Pattern Recognition*, vol. 30, no. 2, pp. 295–309, 1997.

[42] S. Kumar, R. Gupta, N. Khanna, S. Chaudhury, and S. D. Joshi, "Text extraction and document image segmentation using matched wavelets and MRF models," *Image Processing*, vol. 16, no. 8, pp. 2117–2128, 2007.

[43] Z. Kato and T.-C. Pong, "A markov random field image segmentation model for color textured images," *Image and Vision Computing*, vol. 24, no. 10, pp. 1103 – 1114, 2006.

[44] M. Bulacu, R. Koert, L. Schomaker, and T. Zant, "Layout analysis of handwritten historical documents for searching the archive of the cabinet of the Dutch queen," in *International Conference on Document Analysis and Recognition*, pp. 23–26, 2007.

[45] S. Tsujimoto and H. Asada, "Major components of a complete text reading system," *Proceedings of the IEEE*, vol. 80, pp. 1133–1149, Jul 1992.

[46] A. Dengel, R. Bleisinger, R. Hoch, F. Fein, and F. Hones, "From paper to office document standard representation," *Computer*, vol. 25, pp. 63–67, July 1992.

[47] A. Conway, "Page grammars and page parsing. a syntactic approach to document layout recognition," in *International Conference on Document Analysis and Recognition*, pp. 761–764, 1993.

[48] A. Jain, A. Namboodiri, and J. Subrahmonia, "Structure in online documents," in *International Conference on Document Analysis and Recognition*, vol. 1, pp. 844–848, 2001.

[49] J. Handley, A. Namboodiri, and R. Zanibbi, "Document understanding system using stochastic context-free grammars," *International Conference on Document Analysis and Recognition*, vol. 1, pp. 511–515, 2005.

[50] S. Crespi Reghizzi and M. Pradella, "A CKY parser for picture grammars," *Information Processing Letters*, vol. 105, pp. 213–217, Feb. 2008.

[51] F. Álvaro, J. Sánchez, and J. Benedí, "Recognition of on-line handwritten mathematical expressions using 2d stochastic context-free grammars and hidden markov models," *Pattern Recognition Letters*, vol. 35, no. 0, pp. 58 – 67, 2014.

[52] M. Shilman, P. Liang, and P. Viola, "Learning non generative grammatical models for document analysis," in *International Conference on Computer Vision*, vol. 2, pp. 962–969, IEEE, 2005.

[53] F. Esposito, D. Malerba, and G. Semeraro, "Multistrategy learning for document recognition," *Applied Artifictal Intelligence*, pp. 33–84, 1994.

[54] G. E. Kopec and P. A. Chou, "Document image decoding using markov source models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, pp. 602–617, June 1994.

[55] A. C. Kam and G. E. Kopec, "Document image decoding by heuristic search," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, pp. 945–950, Sep 1996.

[56] D. Gao, Y. Wang, H. Hindi, and M. Do, "Decompose document image using integer linear programming," in *International Conference on Document Analysis and Recognition*, vol. 1, pp. 397–401, Sept 2007.

[57] S. Chaudhury, M. Jindal, and S. Dutta Roy, *Model-Guided Segmentation and Layout Labelling of Document Images Using a Hierarchical Conditional Random Field*, pp. 375–380. Springer Berlin Heidelberg, 2009.

[58] B. Gatos, N. Stamatopoulos, and A. Antonacopoulos, "ICDAR 2007 handwriting segmentation contest," in *International Conference on Document Analysis and Recognition*, pp. 1284 –1288, 2007.

[59] S. Jaeger, G. Zhu, D. Doermann, K. Chen, and S. Sampat, *DOCLIB: a software library for document processing*, vol. 6067. 2006.

[60] F. Yin and C.-L. Liu, "Handwritten chinese text line segmentation by clustering with distance metric learning," *Pattern Recognition*, vol. 42, no. 12, pp. 3146–3157, 2009.

[61] H. I. Koo and N. I. Cho, "Text-line extraction in handwritten chinese documents based on an energy minimization framework," *Transactions on Image Processing*, vol. 21, no. 3, pp. 1169–1175, 2012.

[62] Y. Li, Y. Zheng, D. Doermann, S. Jaeger, and Y. Li, "Script-independent text line segmentation in freestyle handwritten documents," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 8, pp. 1313–1329, 2008.

[63] J. Kumar, W. Abd-Almageed, L. Kang, and D. Doermann, "Handwritten arabic text line segmentation using affinity propagation," in *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*, pp. 135–142, 2010.

[64] T. N. Dinh, J. Park, and G. Lee, "Voting based text line segmentation in handwritten document images," in *10th International Conference on Computer and Information Technology*, pp. 529–535, June 2010.

[65] M. Feldbach and K. Tonnies, "Line detection and segmentation in historical church registers," in *Proceedings of Sixth International Conference on Document Analysis and Recognition*, pp. 743–747, 2001.

[66] R. Manmatha and J. L. Rothfeder, "A scale space approach for automatically segmenting words from historical handwritten documents," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, pp. 1212–1225, Aug. 2005.

[67] U.-V. Marti and H. Bunke, "Text line segmentation and word recognition in a system for general writer independent handwriting recognition," in *Sixth International Conference on Document Analysis and Recognition*, pp. 159–163, 2001.

[68] E. Bruzzone and M. Coffetti, "An algorithm for extracting cursive text lines," in *International Conference on Document Analysis and Recognition*, pp. 749–752, Sep 1999.

[69] E. Kavallieratou, N. Fakotakis, and G. Kokkinakis, "An unconstrained handwriting recognition system," *International Journal on Document Analysis and Recognition*, vol. 4, no. 4, pp. 226–242, 2002.

[70] E. Kavallieratou, N. Dromazou, N. Fakotakis, and G. Kokkinakis, "An integrated system for handwritten document image processing," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 17, no. 04, pp. 617–636, 2003.

[71] N. Tripathy and U. Pal, "Handwriting segmentation of unconstrained oriya text," in *Ninth International Workshop on Frontiers in Handwriting Recognition*, pp. 306–311, Oct 2004.

[72] M. Arivazhagan, H. Srinivasan, and S. Srihari, "A statistical approach to line segmentation in handwritten documents," tech. rep., Document Recognition and Retrieval XIV SPIE, 2007.

[73] V. Papavassiliou, T. Stafylakis, V. Katsouros, and G. Carayannis, "Handwritten document image segmentation into text lines and words," *Pattern Recognition*, vol. 43, no. 1, pp. 369 – 377, 2010.

[74] B. Yu and A. K. Jain, "A robust and fast skew detection algorithm for generic documents," *Pattern Recognition*, vol. 29, no. 10, pp. 1599 – 1629, 1996.

[75] 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*, pp. 176–180, July 2009.

[76] P. Hough, "Method and means for recognizing complex patterns," Dec. 18 1962. US Patent 3,069,654.

[77] L. Likforman-Sulem, A. Hanimyan, and C. Faure, "A Hough based algorithm for extracting text lines in handwritten documents," in *Proceedings of the Third International Conference on Document Analysis and Recognition*, vol. 2, pp. 774–777 vol.2, Aug 1995.

[78] R. O. Duda and P. E. Hart, "Use of the hough transformation to detect lines and curves in pictures," *Commun. ACM*, vol. 15, pp. 11–15, Jan. 1972.

[79] G. Louloudis, B. Gatos, I. Pratikakis, and C. Halatsis, "Text line detection in handwritten documents," *Pattern Recognition*, vol. 41, no. 12, pp. 3758–3772, 2008.

[80] Z. S. Y. Pu, "A natural learning algorithm based on hough transform for text lines extraction in handwritten documents," in *In Proceedings of the Sixth International Workshop on Frontiers in Handwriting Recognition*, pp. 637–646, 1998.

[81] P. 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*, pp. 241–246, 2008.

[82] A. Nicolaou and B. Gatos, "Handwritten text line segmentation by shredding text into its lines," in *International Conference on Document Analysis and Recognition*, pp. 626–630, 2009.

[83] A. Alaei, U. Pal, and P. Nagabhushan, "A new scheme for unconstrained handwritten text-line segmentation," *Pattern Recognition*, vol. 44, no. 4, pp. 917 – 928, 2011.

[84] R. P. d. Santos, G. S. Clemente, T. I. Ren, and G. D. C. Cavalcanti, "Text line segmentation based on morphology and histogram projection," in *Proceedings of the 10th International Conference on Document Analysis and Recognition*, (Washington, DC, USA), pp. 651–655, 2009.

[85] Z. Shi and V. Govindaraju, "Line separation for complex document images using fuzzy runlength," in *Proceedings of the First International Workshop on Document Image Analysis for Libraries*, p. 306, 2004.

[86] D. Fernández-Mota, J. Lladós, and A. Fornés, "A graph-based approach for segmenting touching lines in historical handwritten documents," *International Journal on Document Analysis and Recognition*, vol. 17, no. 3, pp. 293–312, 2014.

[87] J. Kumar, L. Kang, D. Doermann, and W. Abd-Almageed, "Segmentation of handwritten textlines in presence of touching components," in *International Conference on Document Analysis and Recognition*, pp. 109–113, IEEE, 2011.

[88] M. Liwicki, E. Indermuhle, and H. Bunke, "On-line handwritten text line detection using dynamic programming," in *Ninth International Conference on Document Analysis and Recognition*, vol. 1, pp. 447–451, Sept 2007.

[89] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331.

[90] S. S. Bukhari, F. Shafait, and T. M. Breuel, "Script-independent handwritten textlines segmentation using active contours," in *International Conference on Document Analysis and Recognition*, pp. 446–450, 2009.

[91] S. S. Bukhari, F. Shafait, and T. M. Breuel, "Towards generic text-line extraction," in *International Conference on Document Analysis and Recognition*, pp. 748–752, 2013.

[92] J. Hammersley and P. Clifford, "Markov fields on finite graphs and lattices," 1971.

[93] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *International Conference on Machine Learning*, (USA), pp. 282–289, 2001.

[94] G. F. Cooper, "The computational complexity of probabilistic inference using bayesian belief networks," *Artif. Intell.*, vol. 42, no. 2-3, pp. 393–405, 1990.

[95] J. Besag, "On the statistical analysis of dirty pictures," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 48, no. 3, pp. 259–302, 1986.

[96] R. B. Potts and C. Domb, "Some generalized order-disorder transformations," *Proceedings of the Cambridge Philosophical Society*, vol. 48, p. 106, 1952.

[97] J. Pearl, "Reverend Bayes on inference engines: a distributed hierarchical approach," in *Proceedings of the National Conference on Artificial Intelligence*, pp. 133–136, 1982.

[98] M. F. Tappen and W. T. Freeman, "Comparison of graph cuts with belief propagation for stereo, using identical mrf parameters," in *International Conference on Computer Vision*, p. 900, 2003.

[99] J. Sun, N.-N. Zheng, and H.-Y. Shum, "Stereo matching using belief propagation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, pp. 787–800, July 2003.

[100] P. F. Felzenszwalb and D. P. Huttenlocher, "Pictorial structures for object recognition," *Int. J. Comput. Vision*, vol. 61, pp. 55–79, Jan. 2005.

[101] T. Heskes, "Convexity arguments for efficient minimization of the bethe and kikuchi free energies," *Journal of Artificial Intelligence Research*, vol. 26, pp. 153–190, June 2006.

[102] J. S. Yedidia, W. Freeman, and Y. Weiss, "Constructing free-energy approximations and generalized belief propagation algorithms," *IEEE Transactions on Information Theory*, vol. 51, pp. 2282–2312, July 2005.

[103] A. L. Yuille, "Cccp algorithms to minimize the bethe and kikuchi free energies: Convergent alternatives to belief propagation," *Neural Comput.*, vol. 14, pp. 1691–1722, July 2002.

[104] M. Welling and Y. Teh, "The unified propagation and scaling algorithm," *Advances in neural information processing systems*, vol. 14, pp. 953–960, 2002.

[105] E. P. Xing, M. I. Jordan, and S. Russell, "A generalized mean field algorithm for variational inference in exponential families," in *Proceedings of the Nineteenth Conference on Uncertainty in Artificial Intelligence*, pp. 583–591, 2003.

[106] J. Winn and C. M. Bishop, "Variational message passing," *Journal of Machine Learning Research*, vol. 6, no. Apr, pp. 661–694, 2005.

[107] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society, Series B*, vol. 39, no. 1, pp. 1–38, 1977.

[108] G. McLachlan and T. K. T, *The EM Algorithm and Extensions*. 1996.

[109] L. Xu and M. I. Jordan, "On convergence properties of the em algorithm for gaussian mixtures," *Neural Computing*, vol. 8, pp. 129–151, Jan. 1996.

[110] J. Bilmes, "A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models," tech. rep., 1998.

[111] Y. Tong, *The multivariate normal distribution*. Springer series in statistics, Springer-Verlag, 1990.

[112] V. Romero, A. Fornes, N. Serrano, J. A. Sanchez, A. H. Toselli, V. Frinken, E. Vidal, and J. Llados, "The esposalles database: An ancient marriage license corpus for off-line handwriting recognition," *Pattern Recognition*, vol. 46, no. 6, pp. 1658 – 1669, 2013.

[113] D. Fernández-Mota, J. Almazán, N. Cirera, A. Fornés, and J. Lladós, "BH2M: The barcelona historical, handwritten marriages database," in *22nd International Conference on Pattern Recognition*, pp. 256–261, 2014.

[114] S. Pletschacher and A. Antonacopoulos, "The page (page analysis and ground-truth elements) format framework," in *International Conference on Pattern Recognition*, pp. 257–260, Aug 2010.

[115] I. Fogel and D. Sagi, "Gabor filters as texture discriminator," *Biological Cybernetics*, vol. 61, no. 2, pp. 103–113, 1989.

[116] V. Kyrki, J.-K. Kamarainen, and H. Kälviäinen, "Simple gabor feature space for invariant object recognition," *Pattern Recognition Letters*, vol. 25, pp. 311–318, Feb. 2004.

[117] J. Ilonen, J.-K. Kamarainen, and H. Kälviäinen, "Fast extraction of multi-resolution Gabor features," in *14th International Conference on Image Analysis and Processing*, (Modena, Italy), pp. 481–486, 2007.

[118] J. Ilonen, J.-K. Kamarainen, P. Paalanen, M. Hamouz, J. Kittler, and H. Kalviainen, "Image feature localization by multiple hypothesis testing of gabor features," *Image Processing, IEEE Transactions on Image Processing*, vol. 17, pp. 311 –325, march 2008.

[119] S. Gould, J. Rodgers, D. Cohen, G. Elidan, and D. Koller, "Multi-class segmentation with relative location prior," *Int. Journal of Computer Vision*, vol. 80, no. 3, pp. 300–316, 2008.

[120] M. Schmidt, "UGM: A matlab toolbox for probabilistic undirected graphical models," 2013.

[121] S. Crespi Reghizzi and M. Pradella, "A CKY parser for picture grammars," *Information Processing Letters*, vol. 105, pp. 213–217, Feb. 2008.

[122] J. Goodman, "Semiring parsing," *Computational Linguistics*, vol. 25, no. 4, pp. 573–605, 1999.

[123] Y. Xiao and H. Yan, "Text region extraction in a document image based on the delaunay tessellation," *Pattern Recognition*, vol. 36, no. 3, pp. 799 – 809, 2003.

[124] D. P. Bertsekas, A. Nedić, and A. E. Ozdaglar, *Convex Analysis and Optimization*. Athena Scientific, 2003.

[125] A. G. Schwing, T. Hazan, M. Pollefeys, and R. Urtasun, "Distributed message passing for large scale graphical models," in *Proceedings of Computer Vision and Pattern Recognition*, 2011.

[126] M. Ziaratban and K. Faez, "An adaptive script-independent block-based text line extraction," in *20th International Conference on Pattern Recognition*, pp. 249–252, Aug 2010.

[127] I. T. Phillips and A. K. Chhabra, "Empirical performance evaluation of graphics recognition systems," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, pp. 849–870, September 1999.

[128] A. Fischer, A. Keller, V. Frinken, and H. Bunke, "Lexicon-free handwritten word spotting using character HMMs," *Pattern Recognition Letters*, vol. 33, no. 7, pp. 934 – 942, 2012.

[129] A. M. Awal, A. Belaíd, and V. P. D'Andecy, "Handwritten/printed text separation using pseudo-lines for contextual re-labeling," in *14th International Conference on Frontiers in Handwriting Recognition*, pp. 29–34, Sept 2014.

[130] F. Cruz and O. Ramos Terrades, "Handwritten line detection via an em algorithm," in *International Conference on Document Analysis and Recognition*, pp. 718–722, Aug 2013.

[131] O. Nobuyuki, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, pp. 62–66, Jan 1979.