Analyse d'images de documents : segmentation du contenu

Mehdi Felhi

UNIVERSITÉ DE LORRAINE

UFR mathématiques et informatique **École doctorale IAEM Lorraine**

Département de formation doctorale en informatique

Document image segmentation : content categorization

THÈSE

présentée et soutenue publiquement le 10 juillet 2014

pour l'obtention du

Doctorat de l'Université de Lorraine

(spécialité informatique)

par

Mehdi Felhi

Composition du jury

Invité :	Redolphe Duret	Océ Print Logic Technologies SA - A Canon Group Company
Rapporteurs :	Nicole Vincent	Professeur, Université Paris Descartes
	Jean-Yves Ramel	Professeur, École Polytechnique de l'Université de Tours
Examinateurs :	Mario Vento	Professeur, University of Salerno
	Beatriz Marcotegui	Maître de Recherche, MINES ParisTech
	Sylvain Lazard	Directeur de recherche, INRIA Nancy Grand Est
Directeur de thèse :	Salvatore Tabbone	Professeur, Université de Lorraine

Laboratoire Lorrain de Recherche en Informatique et ses Applications — UMR 7503

Loria
Laboratoire lorrain de recherche en informatique et ses applications

Mis en page avec la classe thloria.

Acknowledgments

First, I would like to express my deepest gratitude and appreciation to my advisors in Océ Print Logic Technologies (Canon-Group). Dr. Nicolas Bonnier (Currently, Team Leader at Canon Information Systems Research Australia) and Christophe Leynadier (Team Manager at Océ PLT), for giving me the opportunity to work in the innovation team. Furthermore, I would like to thank Dr. Maria-Valezzka Ortiz-Segovia for supervising me during the previous two years and for providing insightful discussions about my research project. Working in Océ PLT was an important stage in my professional life. For that, I would also like to thank the managers Redolphe Duret and Even Reneka.

Dr. Nicolas Bonnier has been a great advisor who was always available to discuss and help me to examine properly my research. He provided a very helpful research environment and helped me to work on several research projects and to collaborate with other PhD students. He encouraged me to write papers and attend conferences.

On the other side, I would like to express my sincere thanks to my Professor Salvatore-Antoine Tabbone who gave me the opportunity to work with him in his team. He helped me to develop my own academic style during the past three and a half years with his great advice and precious guidance. He has been supportive and has given me the freedom to pursue various projects without objection.

I would also like to thank Profs. Nicole Vincent, Jean-Yves Ramel, Sylvain Lazard, Mario Vento and Beatriz Marcotegui for serving on my PhD Committee and for their support and useful feedback about my research.

I shall also mention all my colleagues in the LORIA laboratory as well as my colleagues in Océ, namely Bart, Jonathan, Rachid, Santosh, Ahmed, Mehdi, Fernande, Marine, Taha, Hacen, Younouss, Kristyn and Albrecht. I have really enjoyed working with them.

Special thanks go to my friends that I met in Nancy, namely Salim, Aymen, Mohamed, Ghazi, Moutie, Chedy, Bilel, Walid, Oussema D., Oussema J., Amani, Nabil, Nihel who have been and still great friends. I would like also to thank and dedicate this thesis to my friends Wael, Hassen, Stoura, Nasr, Morsi, Ala, Said, Firas and Bilel.

My deep thanks go to my wife Ferdouas whose love, encouragement and advice were the main source of my success. Last but not least, I would like to dedicate this thesis to all my family. Especially, my mother Sonia, my father Habib, my brother Zied, my sister Myriam, my grandfathers, Sammouha, Ramzi and Anas. Thanks for their support and sincere kindness.

I dedicate this thesis to
my beloved parents Habib & Sonia,
my grandparents Batam & Mima
and to my wife Ferdaous
for their constant support and
unconditional love.
I love you all dearly !

Table of contents

List of figures

Part I

General Introduction

The popularity of document image analysis (or processing) and content-based analysis in both industrial and research contexts is growing at an explosive rate in today's world. This is motivated by a huge need from many applications that could exploit the development of this domain. These applications include indexing, document parsing, data compression, or image quality improvement. Previously, a traditional way to solve the content-based retrieval problem was to index for each document a set of descriptors manually provided by experts [1]. It is in this context that Doermann [2] elaborated a survey studying the classical document image retrieval methods. Edwards et al. [3] describe in their paper a generalized HMM based method that allows to transcript scanned, handwritten mediaeval Latin manuscripts. This method helps users to access to their documents by a simple full text search. Tiwari et al. [4] propose a search engine that browses automatically US patent images in order to find key words specified by users. For that, they designed a search system whose objective is to evaluate the similarity between the queries entered by the user and the US patent database. On the request of the user, the images are automatically grabbed and represented through an autocorrelogram. Content-based image retrieval interested also several scientists for video applications. Li et al. [5] designed an object model that helps browsing image and video databases. For that, they propose a description-based technique in addition with a content-based technique to improve the precision of their system. In fact, they combine an illumination invariance color channel normalization step, and a feature localization step. The second step merges color, texture, and shape features to identify images. However, after the success of the character recognition systems, a natural solution to index the images was found. This solution relies on recognizing the words included in the documents. In fact, text can be recognized automatically and then used in order to retrieve machine printed documents using textural queries either with or without the ASCII notation. The most important challenge when recognizing text is the localization and the segmentation steps. These steps are commonly known as the detection and the extraction text parts of the process. Insuring the localization step is the main goal of a Document Image Processing (DIP) system before analysing and recognizing all the detected parts. As it is described by Shazia et al. in [6], a typical DIP system is composed of three main steps :

— Preprocessing
— Feature extraction
— Classification

The preprocessing step mainly consists in acquiring the image, enhancing its quality if its original quality is poor, and possibly localizing the regions of interest by binarizing the image. The second step is probably the most critical and the most important step in a DIP procedure. In fact, extracting the features from the regions of interest of the image varies from one method to another and similarly the results depend on the type of the document. A feature is considered as a good one if it allows distinguishing the different regions of interests and eliminating undesired

zones and/or components. The performances of one feature depend from the kind of document images and from its contents. For this reason, we can find a lot of class of approaches in the literature. Each of them performs more or less good results for each type of documents. In the next part of this thesis, I will enumerate some of the most famous existing class of methods. Finally, a DIP system ends by classifying all the different parts based on the extracted feature(s) and on the diversity of the collected data. According to the application and to the extracted feature, the classification step can be either supervised or unsupervised. Similarly, the classification step can be performed either in the original and linear space or in a non-linear one using a kernel as a transformation. A document image is composed of different types of regions. Typically, it contains text regions, image regions, background(s) and graphics. The text processing consists essentially in finding text components and clustering them into words, text-lines and paragraphs before recognizing it. Note that segmenting text should in some cases start with detecting the prospective skew generated by the scan process. This step is essential before analyzing page layouts. Image regions are pictures that are either generated from cameras (photographs) or designed by an artist. Finally, graphical regions consist in lines, tables, and curves etc. The main purpose of DIP is to find an efficient method that automatically segments the document image into the above-mentioned regions. Document image analysis consists in extracting structural and functional information from the segmented image in the same way as a person would. The segmentation is a very important task before the analyzing step. In fact, efficient document analysis systems need information about the region's entity in order to perform the appropriate processing. For instance, if the system detects a text region it will submit the corresponding portion to the Optical Character Recognition (OCR) stage. Document image segmentation became the most important operation in the document image analysis procedure since technology has progressed to nearly resolving the document OCR issue, with recognition accuracy rates higher than 99% [7]. The performance of functional analysis depends also on the performance of the segmentation step. However, using these techniques before segmenting the document image will imply a lot of failures. For instance, applying an OCR system directly on video text without detection stage will result in poor character recognition rates (between 0 and 45% of recognition [8]). Text is often considered one of the most relevant structures to extract in document images and in real scenes. Commonly, in an image, this structure contains a rich source of semantic information [9, 10] that could be exploited in different applications such as, automatic indexing, data compression, or image quality improvement.

Let's briefly enumerate and define some of the possible applications that can benefit from a good DIP stage :

— Automatic indexing : After extracting and recognizing text zones, we can obtain an adequate amount of lexical keywords that label and identify each category of the document image. For example, a digital library can be automatically constructed starting from a

dataset of scanned document images. Another example of application using an automatic indexing is the automatic distribution. In fact, as we mentioned before, a good DIP stage implies a good document image analysis result. In this case, it is easy to detect the different fields of a document, such as, title, authors, sender, and department name etc. and then opt for an automatic distribution of letters or orders. Finally, it is possible to create a digital archive simply by restoring document images by means of a good DIP.

— Data compression : Many approaches applied on the document compression domain have already been elaborated. It is well known that the location of each class of content in the document allows determining an optimal compression method. In fact, an adaptive compression system could be applied for each class separately. DjVu (www.djvu.org) is one of the most famous existing document image compression technologies. The main idea behind this technology is the separation between foreground and background regions in a scanned document image. The inventors of this system [11] consider that text and graphical regions compose the foreground while background is constructed from the combination of images, photos and textures. Under the same resolution, the authors of [11] demonstrated that their produced djvu files require up to ten times less memory space than a classical JPEG file while maintaining a similar and satisfactory quality of the image.

— Improving printed image quality : Falkenstern et al. propose in [9] an innovative method permitting to automatically select a printer color rendering intent using semantic information (keywords). In their paper, the authors use a set of metadata corresponding to each image in order to select the best rendering intent capable to perform an optimal print job. In order to deal with the different reproduction objectives, the International Color Consortium (ICC) offers four different intents. Each one produces a different color reproduction. For that, an a priori knowledge concerning the image is necessary to make the best decision. The authors of [9] present a new system that insures this decision. The result of the selection system depends from several preferences of the user described in terms of metadata. For example it depends from the colorfulness or the details that exist in the image. In this work [9], the metadata are manually provided. However, it can be applied on a dataset automatically labelled.

— Other applications : DIP allows classifying document images by means of visual similarities. This helps to improve the accuracy and the efficiency of the search engines. Let us take as an example a document image database that contains a large set of images. After processing, we can obtain several information concerning the documents and then make a visual-based classification of the dataset. Moreover, this classification helps to understand the nature of the processed document. This information is very important to adapt a document analysis system for each category.

It is also possible to categorically reconstruct a digital document similar to the scanned one provided two main information : the perceptual content and semantic content. The perceptual content is composed of several attributes such as the color, shape, texture etc. The semantic content consists not only in the text content, but also in the relationship between all the objects included in the image. Besides the DIP field, text extraction in images and video frames interested also many researchers for similar applications ; namely image content description, semantic content extraction and interpretation, keyword detection, and indexation of images with text. Furthermore, existing text extraction methods have been developed for specific functions such as licence plate recognition [12].

Text detection and extraction is a complex task which consists in localizing and separating text from homogeneous or complex backgrounds or graphics, mostly in order to recognize characters by means of an OCR. Several challenges overcome when achieving this task such as the low contrast between the text and the background, the edgy and textured non-text regions, the non-uniform background, and the orientation of the text lines (horizontally, vertically, diagonally, etc). Before designing any text extraction system, it is very important to know the kind of text to be detected. In fact, many possible sources of variation may overcome during the design of the image containing the text or during the acquisition step. Indeed, text can artificially overlay the image as it is the case of the caption text in video frames. Text can also be present in architectural plans and maps for description as a legend or a description. Finally, we can find text in scene images. Extracting text in real scene images (called also natural images) is considered as the most complicated case. In fact, in this kind of images, text presents many variations. Below, we report the main possible variations due to the acquisition :

- Illumination : Camera flash effect or ambient light influence the acquisition and then the image rendering.
- Alignment : In the literature, most of the existing papers are interested in horizontal text. Unfortunately, this hypothesis is not satisfied in many cases. Text can suffer from numerous perspective distortions.
- Scale : such work makes some assumptions concerning the size interval of text characters. However, text size is varying a lot because of the zoom level or the resolution of the camera.
- Contrast : Although the text is generally considered as contrasted zones, in natural images text can be low contrasted either due to the intrinsic parameters of the camera or due to the fact that the color values of the background and the foreground are close.

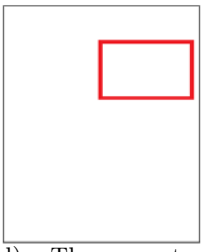
The research project associated with my PhD thesis was initiated by Océ Print Logic Technologies-Canon Group (Créteil, France) in collaboration with the LORIA laboratory (Nancy, France) under a “CIFRE contract”. The objective of this thesis is to propose a generic and full image segmentation system for document image processing and for text extraction in real scene images

overcoming the abovementioned variations in order to improve the printing quality of images. This system should separate a document image into different classes (text, pictures, lines and background(s)). It should also be able to extract text inside the pictures. Given the several challenges and variations of images that could exist in real world, we can imagine the following and the most complicated case that interested us when specifying the project. A complex document image can contain text regions and pictures. These pictures can represent real scene images containing relevant text zones. A simple user of the system can scan this complex document image in order to recover separately its contents. The scan can generate (or not) a skew deformation. Figure 1 summarizes the different tasks that the desired system should perform.

(a) A skewed document image. The skew is due to the scan operation. Text lines are represented in blue color, while the separator is represented in dark. Note that text regions could be over homogeneous background or over complex backgound (inside a photo region)

(b) The first step consists in deskewing the document image

(c) All text regions should be extracted



(d) The system should be able to localize the photo regions : here, a bounding box (in red) surrounds this region of interest

FIGURE 1 – Illustration of the desired system : The document image is divided into two main layers : a binary mask layer that indicates the text line locations, and another layer that surrounds the photo regions

In accordance with this specification, we planned our project as follows :

— We designed a skew detection method able to readjust the orientation of the document image.

— We suggested a page segmentation approach to separate and identify the regions of interest (text, lines, pictures and background).

— We proposed a text extraction method independent to the size and orientation of the text lines/characters.

Thus, the organization of this thesis is as follows. First, I report some of the existing document image segmentation methods in Chapter 1 of Part II. Second, I enumerate in Chapter 2 of the same Part the different categories of text extraction approaches that we can find in the literature.

Part III is devoted to the description and the discussion of our proposed approaches. In the Chapter 1 of this part, I describe our contributions in terms of page segmentations ; namely, our skew detection methods (Section §1.2) and our method designed for document image content classification (Section §1.3). Chapter 2 is dedicated for describing our proposed text extraction solutions. For that, I describe our new text detection system based on stroke descriptors in Section §2.2. Next, I present and discuss our stable text line method designed to extract multi-oriented text lines in Section §2.7. Finally, I summarize the different contributions of this research in Chapter 3.

Part II

State of the art

1

Page segmentation and document layout analysis

1.1 Introduction

Page segmentation is a crucial step in document layout analysis. It allows locating and identifying the different region types that compose a document image. Mainly, these contents consist of text, images, graphics and background(s). The arrangement manner of these regions as well as the document structure defines the document layout. The complexity of layout analysis lies on the class of layout to be considered. We can distinguish two main layout classes :

— Rectangular (or Manhattan) layout :
Layout is considered to be rectangular when all the components belonging to the same content type could be bounded by non-overlapping rectangular boxes (i.e. text, images and graphics could be separated by a set of horizontal and vertical line segments). Many well-known methods addressed the rectangular layout segmentation problem. We discuss in this chapter some of them.

— Non rectangular (or non-Manhattan) layout :
Recently, the physical arrangement of document images (magazines, newspapers etc.) is not limited to the rectangular layout. This is due to the use of trimmed images and the insertion of tilted and artistic text lines in documents. The use of page segmentation method designed for rectangular layouts for such type of document layouts damage the segmentation rates. In fact, the segmentation results could contain overlapped bounding boxes or completely fail to detect some regions. Similarly, these methods give low precision rates when the documents are skewed.

1.2 Existing methods – overview

Categorizing the state-of-the-art works in page segmentation field is a hard task since there are different particularities relative to each existing method. In fact, these methods consist in general in complex processing systems where each system is composed of several stages. Usually these stages consist of the preprocessing, processing and refinement steps. Many existing papers have been devoted for categorizing the different classes of document image analysis and page segmentation methods according either to the representation point of view, to preprocessing step or to the proposed features. Nevertheless, these studies may overlook some individuality of a number of existing methods. Therefore, several page segmentation approaches might belong to more than one class. Furthermore, some classes may overlap. In this section we discuss differently this classification issue. We first propose to expose a categorization criterion in order to set a decisive factor before classifying methods. Then, we describe in details the different classes of existing methods belonging to each categorization criterion separately. We also discuss an example for each class. Finally, we conclude this section by an overview that summarizes the advantages and drawbacks of the existing approaches. By observing the page segmentation literature we can dissociate methods based on the representation point of view, the preprocessing or the proposed features.

1.3 Representation point of view

If we take into account the document image representation point of view, we can dissociate two main classes ; column-level representation and local representation.

1.3.1 Column-level representation

The particular geometric layout of document images has been widely exploited for segmentation. Many works aims to segment and separate regions of document images by taking into account the structure of the document. In fact, several layouts present regular columns that can be separated by horizontal and vertical lines. For this type of images we distinguish three classes of methods ; namely top-down, bottom-up and hybrid methods. Top-down methods first split the document image into a set of disjoint regions. These regions are then classified and subdivided in terms of rows and columns. The most well known top-down technique is projection profile, recursive X-Y cut (RXYC). Bottom-up based techniques use geometric proximity information in order to merge or isolate regions. These techniques start from the pixel level and evolve to group pixels and reach higher level that describes certain geometric model. The most common bottom-up methods are based on connected components. Finally, hybrid methods combine both top-down and bottom-up approaches. In this paragraph we discuss top-down and bottom-up

methods.

1.3.1.1 Top-down methods

A top-down approach starts processing at a high level and then iteratively separates the content of the document into columns, paragraphs and text lines by assuming some a priori knowledge as the inter-column, inter-line space sizes and the shape of the layout. Typically, this approach consists of a sequence of hypotheses that help to segment the image by means of a depth backtracking search. Usually, the resulted segmentation can be likened as the result of a decision tree. One of the most commonly used top-down approaches is the recursive X-Y cuts called also recursive projection profile cuts. This method consists in a recursive process that split a document image into a sequence of rectangular blocks. The projections are calculated along the horizontal and the vertical axes of the document images at each step of the process. Each projection is equivalent to the sum of all the pixel values along the desired direction (where the pixel values can vary between 0 and 255 for grayscale images or between 0 and 1 for binary images). The subdivision is then insured by cutting these projections at the valleys level (a valley is determined by means of its width and its depth). The set of horizontal and vertical cuts alternate step by step. The recursive process ends when the resulted rectangular blocks in one step are the same as those of the previous step. In the following of this part we discuss the recursive x-y cut method of H. Wang et al. [13]. The authors described a top-down approach that consists of three main steps. First, H. Wang et al. [13] propose to scan the rows and columns of the document image in order to calculate the number of dark pixels. Then, they alternately accomplish horizontal and vertical projections to separate character components. A thresholding step is performed for the grouping the character components that belong to the same text row. The result consists in a set of rectangular bounding boxes or also character groups. These boxes are then merged to constitute possibly text lines. Finally a decision step is performed to check whether the resulted clusters are text, photos or graphic regions. This classification decision is made by the use of a linking list that describes the structure of the image. The authors propose also to create large blocks that group separately neighboring bounding boxes of the same type. To informally describe the classification step, an a priori knowledge about the structure and the size of the different type of regions is made. For instance, a sequence of neighboring blocks of similar heights and separated by spaces of the same sizes are considered to be text-line blocks. Similarly, sequences of vertically arranged text lines of the same sizes (height) represent paragraphs. On the other hand, graphics and photos are made of large blocks with more or less density values of dark points.

Discussion The top-down technique approaches are efficient enough for scanned documents (magazines and journals etc.) where the geometry of the layouts is known in advance. However, for

non-manhattan layouts the assumptions of the existing methods are not satisfied. Furthermore, usually the top-down approaches rely on morphological operations for the cutting and merging steps which make them very dependent to the text scale. They also suffer from some limitations especially in detecting skewed texts since they rely on structural features of the column-based documents, especially horizontal text lines and both horizontal and vertical spacing.

1.3.1.2 Bottom-up methods

According to the clustering approach applied in their corresponding works, Ouwayed et al. [14] classify existing bottom-up methods in five main categories ; namely K-Nearest Neighbors (K-NN), Hough transform, smoothing, repulsive-attractive network and minimal spanning tree.

K-NN This category consists of clustering methods where the text candidates are merged using similarity between neighboring connected components. This technique has been exploited in several papers [15, 16].

Hough transform Since the text characters of the same text lines are usually aligned on a straight line. The Hough transform represents a good characteristic to cluster connected components or points into separated zones. Generally, the process step starts by detecting the orientation of the text lines by means of Hough transform. Then, a validation step is performed to eliminate false alarms knowing the structure of neighboring components [17].

Smoothing One of the most well-known smoothing methods is the Run Length Smoothing Algorithm (RLSA) [18] that scans the white runs existing in both horizontal and vertical directions. The white runs are determined by means of either a fixed or an adaptive threshold.

Repulsive-attractive network Oztop et al. [19] presented an energy minimizing dynamical system that allows interacting with the document text image by way of an attractive and a repulsive energies. These energies are defined over a component network.

Minimum spanning tree A spanning tree is a subgraph that connects all the vertices together. Yin et al. [20] use this kind of trees to define and characterize the proximity of connected components.

Discussion Mostly, bottom-up methods perform better results than top-down methods for non-manhattan layouts. However, the processing is slower and requires more parameters. The clustering step also needs a priori knowledge about the document contextual structure.

1.4 Preprocessing

1.4.1 Compressed domain methods

Compressed domain-based methods define approaches that are based on a transformation of the original image to the frequency domain. As example, I introduce and discuss in this paragraph an existing page segmentation method that works in the compressed domain using Haar discrete wavelet transform. Audithan and Chandrasekaran [21] established a page segmentation method that exploits mainly the Discrete Wavelet Transform (DWT) of the original image, with some morphological operators as post processing step. We can summarize the method [21] in four points :

— Extracting the Haar Discrete Wavelet Transform of the original image.
— Removing non-text zones by a dynamic threshold.
— Applying morphological operators in detail components : by dilating the sub-bands (LH, HL, HH).
— A region is considered to be a text region if it represents edges in all of the dilated components.

The Haar discrete wavelet transform is one of the most used transforms in image processing. It is also considered to be the simplest wavelet ; indeed, the mother wavelet is defined as follows :

$$\Psi(t) = \begin{cases} 1 & \text{if } 0 \leq t < 1/2 \\ -1 & \text{if } 1/2 \leq t < 1 \\ 0 & \text{otherwise} \end{cases} \tag{1.1}$$

Haar wavelets can be written as follows :

$$\Psi_k^n(t) = 2^{n/2}\Psi(2^n t - k) \tag{1.2}$$

This results a recursive orthogonal decomposition in Hilbert space. Note that Haar DWT is the fastest wavelet transform since its corresponding coefficients are either 1 or -1. Actually, the Haar discrete wavelet transform generates two different kinds of components from an input image ; the average component (LL) and the detail components (LH, HL, HH). These components are also called sub-bands. The LL sub-band is the result of a low-pass filter, while the others (LH, HL and HH) express the high frequency and can detect three kinds of edges at the same time. S.Audithan et al. [21] apply the Haar discrete wavelet transform to images expressed in grayscale and obtained as a result four obvious features of the image :

1. The LL components express the average of the image
2. The detail components detect different type of edges :

— The HL sub-band detects vertical edges.
— The LH sub-band detects horizontal edges.
— The HH sub-band detects the diagonal edges.

The next step aims to delete non-text regions in each detail component sub-band by a dynamic threshold. The authors of [21] chose a dynamic threshold to distinguish text from graphic edges. In fact, they suppose that the intensity of the text edges is higher than that of the other edges. After calculating the adaptive threshold T that depends on the contrast of the original image, they proposed to binarize the different sub-bands. Then, in order to connect isolated text edges, the authors apply morphological dilation [22]. In fact, a text region is considered to be a compact zone that contains connected edges in all directions especially in vertical, horizontal and diagonal directions.

Discussion The most important advantage of this method is its speed. The implementation of the Haar wavelet is easy and the execution is fast. Besides, the Haar Discrete Wavelet Transform, which detects edges notably, permits also to efficiently detect text lines which contain an important density of edges in all directions (horizontal, vertical and diagonal). Furthermore, this approach enables the elimination of classical contours by means of morphological operators. However, the method [21] does not deal with the overlapping problem (the case when the text is over a photo region) and it separate the input image into two independent regions. Moreover, despite the use of a dynamic threshold, this method is limited to a specific resolution (size) of document images.

1.4.2 Gabor-based methods

The 1-dimensional Gabor function has been introduced by Dennis Gabor in 1946. Then, Daugman [23] extended it to the 2-dimensional case. The Gabor function has been widely used in the pattern recognition and computer vision fields. In fact, this function is considered as a good feature for texture recognition. This particularity has been exploited in many papers for detecting and separating text from the other regions in document images. For instance, Pati et al. [24] proposed a Gabor-based approach for segmenting Indian bilingual document images. The authors use a set a filters called bank of Gabor filters. The authors claim that the use of the filter bank presents mainly three different advantages ; namely, the lower bound of space bandwidth product is achieved due to the Gabor function. Furthermore, the receptive field profiles of the simple cells in the visual pathway are similar to the shapes of the Gabor filters. Finally, for each direction a specific band-pass filter is associated. To separate the text from the non-text area in document images, the authors used a multi-channel filtering approach. The Gabor filters are applied on separated blocks called block energy where the size is equal to 15×15 pixels.

Discussion The advantage of Gabor-based methods lies in their specific band-pass filters applied to different orientations. The Gabor filters allow detecting the texture appearance of text characters that present edgy regions along all directions. However, applying these filters requires a partitioning of the image into blocks. This implies a lot of misdetections if the text characters are larger than the patch size. Contrariwise, if the text is much smaller than the block size, the Gabor-based methods may cause false alarms since non-text regions are possibly merged with textured regions.

1.4.3 Skew detection methods

The performances of functional analysis depend mainly on the performances of the segmentation step. For instance, applying an OCR [25] system directly on skewed document images without skew angle detection will results in a lot of failures in recognition rates. To improve results, several techniques have been proposed for the detection of skew. In this paragraph, we introduce some of these techniques. One of the most often used techniques is Projection Profiles (PP) (and extensions of PP) [26, 27, 28]. For instance, the method proposed by G. Nicchiotti et al. [28] introduces an extension of the PP called Generalized Projections (GP) able to distinguish among a set of black pixels which belong to noise from which belong to text. In this class of methods, the skew angle is estimated by minimizing a cost function applied in different angles. However, the performances of PP based methods depend directly on the type of the document image. For instance, S. Li et al. [27] have mentioned that these methods are very sensitive to the layout of the document image and are not robust to changes in font and size of text.
Other methods are based on connected components such as Nearest Neighbor based method (NN) [29] and Hough Transform based methods (HT) [30, 29]. The NN methods have been used as a clustering method in order to estimate the skew angle. Concretly, a histogram of angles is calculated by determining the nearest neighbor of each connected component and the angle between them. Similarly, HT based methods draw the Hough plan of each document image and consider the peak value as the skew angle. These methods are also sensitive to the type of the document and are computationally expensive especially if the document contains a high number of connected components.
Furthermore, morphological mathematic based methods [31, 32] have been used for skew detection by applying morphological operators in order to transform text lines to a bounding box. But these methods are not robust to noise and to the choice of the parameters of morphological operators. Furthermore, most of the described skew detection approaches are dedicated to simple layout documents.

1.5 Features

The text region identification is always the most difficult task when segmenting document images. For this reason, existing methods employed some features that aim to separate this region from the rest of the image. In this section we expose two categories of methods ; color-based methods and connected component-based methods.

1.5.1 Color-based methods

One of the fundamental principles of conventional image segmentation is the use of the color as a characteristic to distinguish text, image, and background regions. These distinctions between text, image, and background, referred to as features, are extracted during image processing stages to identify text, image, and background objects by mean of various well-known techniques such as wavelet transform, segmentation, or feature extraction. Fuzzy C-Means (FCM) employs two simple statistical features, namely, mean and standard deviation of blocks of pixels. Several features have been used for FCM models. Chuai-aree et al. [33] have proposed a FCM model using three basic observations as follows : First, the image pixel colors are lighter than those of background in gray scale level. Second, pixels that differ slightly in mean or standard deviation are considered belonging to the same object and finally, the background pixels have high mean (bright) and low standard deviation value. A classification is then established for text, image, and background objects as follows :

— Pixels representing a text region have distinctively higher feature values than their background (which are usually lighter or darker to enhance legibility and readability), hence high standard deviation.
— Pixels representing an image have relatively low mean and standard deviation since the grayscale level usually appears darker than those of textual pixels.
— Pixels representing a background object have relatively close to zero standard deviation value and lighter background color.

The FCM method minimizes the dispersion of feature attributes between the center of each region and each block to determine a pixel membership. This step encompasses two processing stages. First each pixel is painted the same color code for the region, whereby the resulting images were classified by color-coded regions and mapping these color-coded regions to the original image is performed. This reserved mapping process is called defuzzification. Second, Neighboring smoothing should be carried out. Smoothing is called for to render pixel level correction caused by the above mapping inexactness (fuzzy classification). In the defuzzification step, the authors propose then to attribute labels to each region regarding to wavelet coefficients. According to the work of J. Li et al. [34], the largest wavelet coefficient distribution is chosen as the background, the region which has the smallest wavelet coefficient distribution, is the image, and the remaining is the

text. Thereby each region is tied to the appropriate categories, i.e., text, image, and background thus the categorization performed and the results have been improved by the smoothing phase which consist in correcting wrong labels.

1.5.2 Connected component-based methods

In their study, Marinai et al. [35] categorize some handwritten and ancient scanned document image analysis systems in the class of connected component-based methods. They justify the employment of the connected component level by the fact that the shape handwriting characters is varying a lot and then it needs a local recognizing system before clustering components into groups. In this category of methods, the authors cite the work of Marinai [36] that identify relevant words by means of a modified Dynamic Time Warping (DTW) after clustering connected components. They also cite the graphical based method of Barbu et al. [37] that represents the document image in a graph where the nodes symbolize the connected components. Saitoh et al. [38] propose a full document segmentation system based on connected components. The authors claim that this system is shape independent and robust to the skewing effect. For this purpose, they first extract and classify connected components. Then, neighboring components are merged into larger blocks. This step allows creating the different zones i.e. body, title or footer zones. Saitoh et al. propose the use of a tree structure using influence ranges. A tree preorder traversal is then performed to obtain an order of the text.

Discussion It is clear that most of the connected component-based approaches need finally a bottom-up approach after the classification step to merge components into zones. This procedure implies the use of a merging criterion (as the morphological operations or the decision trees). Usually, the choice of the criterion is crucial for the quality of the segmentation.

1.6 Performance evaluation

The choice of the appropriate evaluation method is a very important step that permits us to evaluate objectively the efficiency of different algorithms. However, the establishment of performance evaluation method is also a complex task in document analysis domain since it depends on many criteria, especially the choice of the test database and the method of the ground-truth generation. Note that the comparison between segmentation results and the ground truth is an efficient and common alternative to the visual checking method which is subjective, poor, unreliable and time consuming. Typically, an evaluation method consists of four main steps :

— The selection and the diversity of the images of the test database.
— The choice of the relevant information contained in the document image.
— The method of generation and representation of the ground-truth.

— The choice of the heuristics that describe the performances.

This procedure depends also on the type of documents and the regions of interest chosen by the author of the algorithm to be tested. In this section, I will describe first the performance evaluation steps. Then, I present in its second part some common datasets used by the document analysis scientific community.

1.6.1 Performance evaluation steps

Basically, in order to evaluate an algorithm, we should compare its segmentation results to a manually labeled dataset. This dataset is called the "ground truth". Hence, the more similarity there is between segmentation results and ground truth, the more the algorithm is efficient. The comparison is assured by means of statistics that describe the accuracy, precision, wrong segmentation etc.

1.6.2 Ground truth representation

Daniel Lopresti et al [39] define the ground truth as "a set of reference (truth) files produced by (human) interpreters". Oleg Okun and Matti Pietikainen [40] define it as what we should obtain after an ideal segmentation. We can also define the ground truth as a representation of the perfect result of the image document segmentation step. Hence, the generation of the ground truth (ground truthing) is often done manually or semi-automatically. Actually, the ground truthing is an expensive task (to scan images, to segment them manually, to represent the results and to maintain them etc.). Furthermore, this task depends on many factors like the type of the contained information and its representation. Consequently many works tried to establish automation processes to facilitate ground truthing. We will discuss in more details the automation process in the next paragraph. The objective of the ground truth is then to allow its users to compare their segmentation results (working from the same original dataset) with it.

By and large ground truth files are represented in Extensible Markup Language (XML) format. The XML allows its users to benefit from many advantages like :

— Data size : Instead of representing the segmentation results in an image format (TIF, JPEG, BMP etc.) with the same size of the segmented image which can be a wide format one, the use of the XML allows significantly compressing the data size of the ground-truth database.

— Simplicity : XML is readable to both machines and human and is a universal language.The use of this language allows organizing the writing and the reading operations. Moreover, XML parsers are easy to build.

— Extensibility : This is an important challenge when creating the ground truth dataset. Thus, new information might be added and old data.

Generally, a Document Type Definition (DTD) is provided with the set of the XML files. The DTD offer to XML parsers some rules to validate or not the file.

The document image is composed of many regions of interest. Hence, the XML file should describe each region differently and provide relevant information about it such as its position, orientation, or size.

1.6.3 Generation of the ground truth

Over the past few years, much work had been done in order to automate the creation of the ground truth. Presently, we find several interfaces (software) that deal with this problem and allow users to label each region and specify automatically associated attributes. Below are some of the most known tools of ground truthing.

Aletheia Alethia is a semi-automated ground-truthing tool developed by the PRImA group [41], destined for page layout analysis. This software is used to generate the ground truth of the PRImA dataset for ICDAR competition series [42]. The procedure is described with detail in [43]. The process begins with a connected component analysis. The result of this analysis is then exploited to construct regions that might be specified with "imprecise" region boundaries. For instance, the tool allows users to change the drawing modes and gives them the choice between drawing arbitrary polygons or rectangles to determine region boundaries. We note that the use of polygon shape to delimit boundaries is a well-organized solution to perform more precision about the position of each region by minimizing the empty space. Finally, Aletheia create a ground truth file in XML format or in its own format.

Trueviz Trueviz [44] is a tool, implemented in Java code language, which allows users to read and write easily XML ground truth files. Basically, it visualizes input (images with their correspondent segmentation in XML format) and permits to edit them by outlining titles, paragraphs, lines, words etc. TrueViz is designed for Page Layout Analysis. Therefore, TrueViz read only one type of document which is specified in a file called TrueViz.DTD.

Agora Agora [45] consists in interactive document image analysis software dedicated to historical document images. This tool helps to study this kind of images by segmenting it into classes.

ViPER ViPER (from the University of Mariland [46]) is designed for video text detection applications, especially, for ground truth generation. Typically it provides a frame by frame interface for users in order to produce metadata. It is also a visualization tool of video analysis

results and contains metrics for evaluation. Hence, it offers the possibility to produce automatically the performance evaluation. ViPER is widely used in video text detection/extraction field since it is easy to use, and permits the use of different output data configuration. The output result of the segmentation is often stored in a XML file given that it is a universal language, readable in different software that cannot interface with ViPER.

1.6.4 Common image databases

As we mentioned in the introduction of this chapter, the performance evaluation is a task that depends on the context, thus, there is neither agreement about a unique database nor about the regions of interest described in the ground truth. That's why we find in the literature several databases destined to different performance evaluation applications. In this section, we describe three of the most known datasets. These datasets are : PRImA dataset [41], Dataset of the University of Washington (UWASH) [48] and MediaTeam dataset [47].

1.6.4.1 PRImA dataset

This database was designed by PRImA group from the University of Salford [41, 49]. Typically, it contains several document images with the corresponding ground truth. Before selecting the images, the authors of [49] defined several criteria that must satisfy any database dedicated for an evaluation aim. According to [49], there are three main qualities that characterize a usable dataset which are :

— Realism : a realistic dataset must contain different categories of real and common kind of documents (such as newspapers, magazines and advertisements).
— Comprehensibility : the dataset must include different information in order to enable best evaluation.
— Flexibility : it should allow its users to select subsets depending on specific conditions and should be easy to browse.

The database is composed of two main parts :

— The first part contains some information (metadata) such as resolution, dimensions, the presence (or not) of images and/or graphic, colored or uncolored images etc. The designers of this dataset claim that the use of this metadata and attributes is important for users who want to browse and select a specific set of document for evaluation. This information could also be useful to adapt appropriate algorithm or parameters to each kind of document.
— The second part contains images (in *.tif format) and corresponding ground truth files (in XML format).

1.6.4.2 UWASH Database

UWASH database [48] is composed of three CD-ROM designed by the University of Washington and contain principally simple layout document images, where the entities could be expressed by simple bounding rectangles. The ground truth was generated automatically using a system called automatic table ground truth generation. This database contains documents in English and Japanese languages.

1.6.4.3 MediaTeam document database

MediaTeam database [47] contains different kind of documents categorized by their type. Mainly, it is composed of advertisements, articles, business cards, color segmentation images etc. Therefore, the images contain less number of complex layouts and the regions are represented using bounding rectangles.

1.7 Conclusion

In this this chapter we described and categorized some of the existing document image segmentation approaches. We also cited some of the existing ground-truthed databases used for the evaluation.
Table 1.1 summarizes the different classes of the existing document image segmentation methods together with the main advantages and disadvantages in terms of functional and computational performance. We can remark that the top-down methods are the fastest, but they are the most vulnerable to the structure of the document image. In the opposite way, connected component-based methods are the most precise. However, they often suffer from a high computational cost. Our aim in this thesis is to propose a method that is less sensitive to the type of the document (in terms of structure, scale and orientation) than the existing methods. For that, we focused mainly on the connected component-based class of methods. In the same time, we tried to provide solutions to descrease the computational cost. Namely, by estimating the orientation of the document and by proposing parallelizable process to extract and describe our regions of interest. In fact, a document image can contain thousands of connected components.
In Chapter 1 of Part III, I describe our contributions in this domain. More precisely, I will discuss our skew detection methods that allow estimating the orientation of the scanned documents. Then, I describe our document image segmentation method.

Techniques	Advantages	Disadvantages
Top-down methods	— Perform good segmentation results for manhattan layouts where the text regions are well contrasted — Are not time consuming since the complexity is low — Easy implementation	— Sensitive to the non-manhattan layouts — Are not able to detect isolated text components — Classical top-down methods are not able to detect text line from skewed document images — Do not deal with cropped images
Bottom-up methods	— Less sensitive to the non-manhattan layouts comparing to the top-down methods — For classical document images satisfying the contrast requirement, these methods works well and performs good extraction rates	— Computationally expensive comparing to the top-down methods — Consist of split and merge steps. These steps require some criteria that might vary from an image to another — Depend on the structure of the document images
Compressed domain methods	— Do not require prior information about the image structure — These method can combine local and global information by combining the frequency and the spatial information of the image — Less sensitive than the previous methods to the document structure	— Vulnerable to the image resolution. In fact, the discrete transforms do not take into account the size of the image. Thus, the precision varies if the resolution increases or decreases. — Region selection has an inherent dependence on the text size. — Mainly, these methods are more precise to detect horizontal and vertical text lines and less precise to detect arbitrary rotations

Color-based methods	— Have low computation complexity — Performs satisfactory results for a wide category of document images (especially, classical magazines and newspapers) — Less sensitive to the orientation of text lines	— Less immune to noise than the compressed domain and the connected component-based methods — Without the presence of apparent peaks (for brighter regions) or valleys (for darker regions), these methods do not work well. — The recognition rates fall down when the document image resolutions are low — Spatial and structural details are generally ignored
Connected component-based methods	— Allows characterizing text regions by means of features — Less sensitive to the scale changing	— High complexity — Necessitate some prior knowledge about the size and the orientation of the text regions

TABLE 1.1: Summary of the principal document image segmentation classes

2

Text extraction in real scene images

2.1 Introduction

Text detection and extraction is a complex task which consists in localizing and separating overlapping text from complex backgrounds or graphics mostly in order to recognize characters by means of the OCR. To achieve this goal we should overcome several problems. Typically, these challenges are : the low contrast between text and non-text, the shape of the text components, color homogeneity and texture appearance.

— Contrast between text and non-text : This issue is very common when processing real images. Indeed, very often text regions overlap graphics or backgrounds that present close grayscale or color values. Consequently, the task of the localization becomes very difficult. Thus, many researchers included the contrast information in the pre-processing step.

— Text component shape : One of the characteristics of the strings in the text is that they form compact shapes. This feature has been adopted in many approaches by exploiting the connected component analysis. However, this technique could produce incorrect results by connecting characters to other pixels belonging to the background mainly in the case of text/graphics separation.

— Color homogeneity : Often, a readable text presents a unique color different than that of its background. So, some authors assume that the letters are monochromatic and neglect exceptions when there is some noise in the image.

— Texture appearance : Texture analysis techniques are one of the most applied approaches destined to solve the problem as they allow localization of the text that presents a texture appearance. This kind of technique may cause false detections when there are some texture structures in the background.

Consequently, the task of the localization is a complex and challenging work. For instance, many authors included for example the contrast information analysis in their preprocessing step to deal with the contrast problems. Other researchers presented shape descriptors to well characterize the varying shape of text characters. In the remainder of this introduction, we will enumerate some existing works from the literature. Existing methods could be classified in four main groups : Region-based, color-based, statistical-based and connected component-based methods. Region-based methods are either based on the edgy characteristic of the text or based on its texture appearance. Shivakumara et al. present in [50] a region-based system to detect multi-oriented text lines by proposing a frequency-based method to detect multi-oriented text lines based on a Laplacian approach. The authors propose the use of the Maximum Gradient Difference (MGD) to detect text region candidates. The use of the MGD requires the definition of a local window size which implies the misdetection of big sized text regions. Color-based methods include all the works that consider that text is a homogeneous region. In fact, a readable text presents a unique color different than that of its background. Some authors assume then that the text characters are monochromatic. For instance, Gllavata et al. [51] use the K-means clustering technique in order to distinguish text components. This method is quite fast and provides good segmentation rates if the text is totally uniform. However, the detection results fall down if the image is noisy or in case of low contrast between the foreground and the background which is mostly the case of real-scene image with complicated background. Threshold-based methods can be also considered in this group of color-based methods. These methods are known to be fast and robust for simple images. For example Otsu [52] applies a global thresholding to extract text from background by the use of histograms. Other methods [53, 54] extend this technique and propose local thresholding methods, but all these methods are sensitive to the background variations and to the text size. Statistical based class encompass probabilistic models such as Conditional Random Field model (CRF) [55], and the Markov Random Field model [56]. These methods perform good results in case of complex background. These models combine unary properties of the text (geometrical features) with the binary or contextual properties (similarity and spacing between the components that represent the text). Connected component-based methods rely usually on a preprocessing step that allow to extract text candidates and then a processing step resulting in recognizing text by means of text intrinsic features like strokes [57, 58]. As mentioned above, most of these approaches perform more or less well for extracting some text types depending on the contrast and the quality of the image (noisy or not). In accordance with the categorization done previously, we plan this chapter as follows : first, we discuss the region-based methods, then the color-based methods. After that, we examine the statistical-based methods. In a fourth stage, we discuss the connected component methods. Along these three steps we report some examples for each category and we mention the main advantages and drawbacks of each of them.

2.2 Region-based methods

Most of the methods of text extraction described in the literature rely on some a priori knowledge based on textual characteristics. These methods apply general approaches with or without taking into account the contrast of the image regions.

2.2.1 Gradient characteristic

The Maximum Gradient Difference (MGD) is one of the most known region-based techniques used to detect text lines. The MGD in a given pixel is defined as the difference between the maximum and the minimum of values inside a sliding window centered on the same pixel and applied to the gradient of the image. The window size depends directly on the characters size. In fact, the text regions are supposed to have high MGD values while the MGD values of the background pixels are small. Wong and Chen [53] presented a text detection method based on the MGD. Their method consists of seven main steps :

— Identification of potential text line segments : for this, the authors apply the MGD to the grayscale image. By assuming that positive gradients correspond to a text to background transition while negative values are due to a background to text transition when the text is brighter than the background (the reverse is true ; when the text is brighter than background), Wong and Chen select segments which contain MGD values that exceed a threshold ; in order to eliminate false alarms they decided to exclude segments that have a small number of transitions (background to text and text to background) and blocks having mean and variance that are not included in a certain range.

— Text blocks detection : this step consists of merging potential text line segments

— Filtering the text blocks : to minimize false positives, blocks that are not included in a certain height-to-width ratio interval are removed

— Adjusting boundaries : to include all the parts of the letters belonging to the detected block. For this, the authors include outside pixels having MGD values close to the MGD average of the initial block

— Bi-color clustering : The authors of [53] suppose that in one block there are only two dominant colors, the foreground and the background colors. For that, they propose to pick the two principal colors from the color histogram of each text block to separate text from non-text regions.

— Filtering the artifacts : This step aims to eliminate noisy artifacts. Indeed, Wong et al. [53] distinguish the noise from text area by establishing some characteristics of the letters based on connected components.

— The image processing terminates by a contour smoothing step ; the authors apply structuring elements called the classical pruning structuring element pairs.

The different refinement steps of this approach decrease false alarms and the misdetection rate. For example, after the MGD step, the missed text regions will be added due to the adjustment buandaries step. Likewise, false candidate text line will be deleted at the end of the filtering artefact step. However, this method is quite slow (many refinement steps) and fails when the text lines are not horizontal.

2.2.2 Contrast characteristic

Shivakumara et al. [59] explored the contrast information by classifying first the image as a high or low contrast image in order to adapt the processing step to detect as well as possible boundaries of text lines. The approach of Shivakumara et al. [59] is composed of four main steps :

— Classification of high and low contrast images.
— Edge feature and texture based method for text detection [60].
— Threshold selection for high and low contrast images.
— False positives filtering.

The authors claim that the major cause of failure of most of the text extraction methods is due to the unsupervised threshold used to separate text from non-text. Thus, they propose to classify images into high and low contrast ones. This approach begins by dividing an input image (256 x 256 pixels) into 16 blocks that will be categorized into high and low contrast blocks by using two different heuristics. The first heuristic considers a block as high contrast block when the number of its Sobel edges is more than the number of the Canny edges, otherwise it will be considered as a bloc with low contrast. The authors used the arithmetic filters (to blur the image blocks) and the median filter (to eliminate noise pixels) as a preprocessing step. As a second step, the authors use a collective classification that groups previous information to judge if the image is high contrast image or not. An edge-based method [60] has been applied to detect text lines boundaries using an unique threshold, and then they propose two dynamic thresholds that correspond to the high and low contrast images respectively, these two numbers are calculated using the average gradient edge map. Finally, the method ends by removing false alarms by means of two opposite features : straightness and cursiveness. A detected text block is extracted when the number of the straight edge (the centroid of a component falls on the component itself) components is more than cursive edge ones. P. Shivakumara et al. [59] established an approach based on a high/low contrast classification and they succeed to notably improve the performance of an edge-based method [60]. As a matter of fact, this method is automatically adaptable to different kinds of documents, specially, those containing photos that can represent either high or low contrast with the text. However, the authors fixed the size of the blocks they use (to classify the image and calculate thresholds) to 16 by 16 pixels window block. This assumption has led to some incorrect results (misdetection and false alarms), particulary, in graphical image

documents which contain a lot of homogenous regions (greater than the size of the window).

2.2.3 Texture appearance

Wu et al. [61] propose a text extraction system using the texture appearance of the text regions. They summarize their system in four main stages as follows :

— A texture segmentation scheme is applied to select region candidates.
— Strokes are extracted from the detected regions.
— The background is separated from the foreground.
— The text is binarized in order to create final text bounding boxes.

In the first step of their algorithm, the authors distinguish text from non-text by means of texture appearance. In fact, text presents high frequency. Text characters of the same text line present also similar heights and spacing. Additionally, they constitute a straight line. For this purpose, Wu et al. propose the use of nine Gaussian derivative filters throughout three different scales. To the output of these nine filters, the authors create a feature vector at each pixel in the image composed of nine estimated energies. Based on their corresponding feature vectors, the pixels are then clustered using the K-means algorithm into three different clusters. Only, one of these clusters is considered as the text region. In the end of this step, a morphological operation is applied in order to link text fragments together and refine the segmentation results. This method is less sensitive in terms of scaling then the other classic region-based method. In fact, the authors propose the use of a pyramid scheme by proposing filters of different scales. However, authors consider all the regular textures as text which reduces significantly the precision of their algorithm in complex background images.

2.3 Color-based methods

The idea behind the color-based methods is the use of the color information in order to binarize the image before extracting the text. In fact, the text regions are usually considered as a set of monochromatic components. For this, many works have used the grayscale or the color histograms to automatically binarize the images by applying either thresholding or clustering approaches. In this section, we introduce one of the most known color-based method proposed by Gllavata et al. in [51]. This method aims to characterize the text from non-text regions using two different features ; namely, the pixel colors and the wavelet coefficients before validating the segmentation results using a commercial OCR. The authors propose in [51] a text segmentation system composed of four main steps :

1. Resolution Enhancement.
2. Text Color Estimation.

3. Feature selection and normalization : color and wavelet coefficients.

4. Pixel classification.

The first step aims to overcome the OCR limitations when the image represents low resolution. Thus, the authors propose to normalize all the images under process to a resolution equal to 300 dpi by means of a cubic interpolation. After the resolution enhancement step, the processed image is used to estimate the dominating text color. After a color histogram quantification, the authors estimate two different histograms ; the first, corresponds to the text color while the second corresponds to the background color. The maximum and the minimum of the histogram difference between these two histograms are considered to be the final text and background colors. The third step of the proposed method [51] consists in selecting two main features of the text regions ; color and wavelet features. For each pixel, the final feature vector is composed of the normalized red, green, and blue pixel color components added to the standard deviation of the 5/3 wavelet coefficients. To calculate the wavelet coefficients, the method use a small sliding window (of 3 by 3 pixels) centered in each pixel. The authors consider that the wavelet coefficients represent low values inside the text characters while they are high at the boundaries. Hence, the constructed feature vector can distinguish the text from non-text pixel according to the authors. Thus, the semi-supervised clustering k-means algorithm is applied in order to classify the image into two main clusters. The first class consists in the text regions identified by means of the Euclidian distance between its corresponding center and the ideal feature vector of a text block. The second class is composed of the background regions.

The main advantage of this method is the fact that it combines an intrinsic text characteristic (namely the color histogram) with a structural text characteristic (wavelet coefficient). The authors demonstrate the segmentation performances of their method by evaluating its recognition rates over a test dataset under different resolutions (72 dpi and 300 dpi). However, as the most color-based methods, this approach mainly suffers from three limits :

— Its dependence to the text scale changing : Even if the authors propose a resolution normalization step. The size of the sliding window is fixed to 3 by 3 pixels. This implies automatically several misdetections if the text characters are much higher than the window size.
— The vulnerability to noise : This method works well when the text characters are perfectly monochromatic. However, the text/background separation is not well performed in case of noisy images.
— The vulnerability to complex backgrounds : in fact, textured background do not present a uniform color which affect the results of the histogram difference.

2.4 Statistical-based methods

Mathematical models have also been employed to detect text. These statistical methods are not only used for the classification or the training issues but also for modelling the different spatial relationship between the different components of the text. In fact, these models allow combining intrinsic and extrinsic characteristics of the text. For example, it is known that the characters belonging to the same word or to the same text line present several distinguishing features comparing to the non-text regions (such as noise, homogenous regions or textured zones). For a given text line, we can mention the following particularities of the text characters : they present the same color, the same size, the inter-character spacing is uniform, and text characters are mostly aligned to the same fitted straight line. Some researchers propose the use of existing mathematical models to understand and then extract these characteristics. For instance, the random field models CRF and MRF present several assets permitting to represent and detect spatial and statistical particularities of the text. In this section, we discuss an example of statistical-based text detection papers that employ a CRF model.

On the one hand, each text character contains a rich set of characteristics that can help to perform a good classification process. On the other hand, statistical dependencies between text characters exist and it is one of the most important criteria that allow distinguishing the text. Hence, classical graphical or generative models (like the naive Bayes model) represent an adequate tool to represent the dependencies inside the same text zone. However, modelling an efficient and accurate naive Bayes classifier for extracting text is a hard task. In fact, graphical models represent the joint probability distribution of two variables (x and y). The first variable represents the observations, while the second represent the predictions. If we consider that x is the extracted features from the text candidates and y is the decision (text or non-text), it is trivial to conclude that there is a kind of dependency between features belonging to the same region. As a consequence, representing the joint probability distribution requires modelling the unary probability distribution of the observed entities which can lead to a difficult task. For such kind of data, Lafferty et al. [62] propose the use of the conditional distribution without the need of explicitly representing the probability distribution of the observation. This approach is taken by conditional random fields. Li et al. [55] apply in their work a conditional random field to segment text regions from images with complex background. For this, the authors introduce three potentials to model their segmentation system. These potentials are called unary, pairwise, and contextual.

— Unary potential : it incorporates local visual information that describes relationship between the local visual information (observation) and the label of the site (text or non-text).
— Pairwise potential : it reflects also the local visual information by reflecting the dependency

between neighboring labels.

— Contextual potential : the authors propose to add this term in order to solve local ambiguities that both unary and pairwise potentials cannot distinguish. Concretely, it allows segmenting text pixels from complex backgrounds using global information.

2.5 Connected component-based methods

Two kind of connected component based-methods have been reported in the text detection literature. The first group exploits the fact that text characters present homogeneous color components while the second group consider text lines or words as separated connected components after merging similar and neighbour text parts. The second group use naturally region based preprocessing steps such as morphological operations in order to cluster text characters into text lines. In this section, I present different existing works that developed connected component solutions for their text detection systems.

2.5.1 Text character level

The main advantage of this kind of methods is the fact that they allow not only localizing the text regions but also extracting it, which facilitates the OCR work. One of the most well known and efficient connected component based methods is proposed by Chen et al. in [58]. The authors present in their article a new stroke based descriptor applied on the text candidates. This method starts by enhancing the contrast of the image and extracting maximally stable extremal regions. Then, the non-text regions are filtered out. In fact, small and big components are disregarded. Similarily, components that present high number of holes or a high aspect ratio are eliminated. Next, a text descriptor is proposed. For that, Chen et al. [58] present an algorithm for finding stroke widths in each part of the text component candidate. This algorithm is based on the distance transform applied on the extracted connected components. Afterwards, the authors present their text descriptor that evaluates the stroke width variation inside the text character candidate. This approach ends by clustering text character components into text lines. This method offers several advantages such as its low complexity and its good segmentation rates comparing to the existing methods. Conversely, this method presents some important limits. For instance, the authors assume that the extracted stable regions are made of separated text characters. This assumption is not correct if we want to extract cursive text. Hence, it is not able to detect this kind of text because of the aspect ratio and/or the high number of holes inside a cursive text region. Furthermore, the authors do not deal with non-horizontal text lines. Besides, some of their assumptions are not fulfilled in case Japanese and Chinese langages.

2.5.2 Text line level

Phan et al. [63] propose to use the maximum gradient difference in order to cluster text components. The authors exploit the contrast between the text and the background and assume that the transitions between these two regions are enough to extract a compact region and consider it as a text line candidate. Hence, the text detection problem becomes a classification problem where the collected connected component candidates will be identified as text and non-text regions. For that, Phan et al. [63] begin with clustering the content of the transformed image using K-means into two clusters. The first one presents low maximum gradient difference values and is considered as non-text region (or background) while the second cluster is composed of text candidates. These text component candidates are considered as text lines (or words). Phan et al. present in their paper some characteristics of text lines and then propose some features in order to refine the results of the clustering step by eliminating non-text regions. Concretely, the authors present three main features. Namely, the ratio aspect, the area and the edge area of the text connected component candidates and they apply some rules in order to disregard non-text regions. This method presents good results when the text is well contrasted and horizontally oriented. However, it suffers from several drawbacks such as its dependence to the text scale variation and to the image kind. In fact, the refinement step relies on fixed thresholds that depend on the text size and font.

2.6 Conclusion

This chapter has been devoted for discussing existing text detection and extraction methods. Table 2.1 shows the different existing classes of text extraction methods. This table recalls the main advantages and drawbacks of each family of approaches.
On the one hand, region-based and color-based methods can perform efficient segmentation processes if the assumptions concerning the image are satisfied. On the contrary, the segmentation could lead to bad classification of the image content. Concretely, text could be misdetected and background regions could be considered as text zones. On the other hand, connected component-based methods are more appropriate for detecting text of arbitrary scales, colors and arrangements. However, most of the existing methods make a lot assumptions in order to increase their precision rates making their method vulnerable to the type of image.
In this thesis, we chose to propose a connected component-based approach that deals with different kind of text (regardless its scale, color and arrangement). We describe this method in Part III - Chapter 2.

Techniques	Advantages	Disadvantages
Region-based methods	— Perform good detection rates when it is easy to define a region homogeneity criterion — Comparing to the classical edge-based methods, these methods are less vulnerable to noise	— Most of these methods are computationally expensive and necessitate a large memory — Depend on the order in which pixels are processed — The precision of the region-based methods depend on the used segmentation criterion and the type of images — Less immune than the other approaches to low contrasted regions. For instance, monochrome text detection methods working in the grayscale space suffer from the fact that some different colors present similar projection values (e.g., yellow and white). — Mostly, these methods end with a refinement step that eliminates very big and small text regions. Prior information of the text size is then necessary

Color-based methods	— Do not require a priori knowledge about the image — The computation complexity depend neither on the number of regions nor on the number of components and then are not computationally expensive comparing to the other approaches	— Vulnerable to noise and low contrasted images — Most of these methods take into account only local details and do not include the spatial interaction between text components. However, text regions present regular textures that distinguish them from the non-text regions — Few existing methods combine color with structural information — Natural images present many physical phenomena (like noise and shadows). These phenomena imply the non-homogeneity of text regions and then impact the segmentation precision

Statistical-based methods	— Combine structural and color information — Do not rely on complex algorithm — The training step allows reducing the number of assumptions — These methods are able to detect text from images with complex background	— Training requires collecting and groud-truthing various text and non-text regions — Text detection rates depend on the training dataset — Overtraining decrease significantly the detection rates by affecting the classification results — Most of these methods necessitate initialization which can influence the results
Connected component-based methods	— Immune to the scale changing — Work better than the color and region-based methods for text extraction in complex backgrounds — Allow using descriptors and feature vectors to well describe text characters and text lines. This implies good precision and recall rates.	— Computationally expensive. The complexity depends on the number of text candidates — Results depend on the descriptors and the classification processes — Mostly, prior information about the text size and/or orientation affects the extraction rates of small, big and non-horizontal text regions.

TABLE 2.1: Summary of the text extraction in real scene images approaches

Part III

Proposed approaches

1

Document image segmentation

1.1 Introduction

As mentioned in the state-of-the-art part, most of the existing document image segmentation methods depend on the structure of the document. In fact, the performances of the segmentation step may fall down when segmenting non-manhattan or skewed documents. Recognizing and grouping text regions into blocks is a hard and complicated task. For that, many researchers make a lot of assumptions concerning the color, orientation and/or the orientation of the text regions.

In this thesis, we aim to make the segmentation step independent of the structure of the document image. More precisely, we designed a segmentation system that deals not only with the skewed document images but also with the different text sizes in the same image. For that we proposed two main contributions to the document image segmentation field. Concretely, we proposed two skew detection methods and a document image categorization approach.

This chapter is composed of two main parts :

— Skew detection : We discuss in the begining of this chapter our two skew detection methods that allow detecting the skew of the scanned documents and then adjusting their orientations. These method help to extract text region candidates with different contrasts and independently from the layout and the type of document images. Both methods consist of two main steps : the segmentation step and angle estimation step. We first proposed the use of the Maximum Gradient Difference (MGD) for segmenting the document images and detecting the text lines. Then, the result of this segmentation step is considered as an input to the R-signature or to the Ridgelet-based approach in order to estimate the skew angle.

— Document image categorization : In this part, we present a robust stroke-based segmentation system that aim to detect and separate : text, lines, photos and background(s). First,

we introduce the system overview in Section §1.3.1. Then we describe our stroke-based feature designed for detecting text and line candidates in Section §1.3.2. Section §1.3.3 is devoted for background/image segmentation. Next, we discuss the line detection and text clustering steps in Section §1.3.4 and §1.3.5. Finally, we report the experimental segmentation results and a comparison with existing page segmentation methods in Section §1.3.6.

1.2 Skew detection methods

I present in this section our two skew detection methods. The main difference between these two methods lies in the estimation step. In fact, the first one (method 1) uses the R-signature shape descriptor to estimate the orientation of the text regions while the second (method 2) is based on the Ridgelets. In the end of this section we discuss the differences between these two methods and the advantages of each one.

1.2.1 R-signature based method for skew detection

1.2.1.1 Maximum Gradient Difference (MGD)

As we mentioned earlier, our method begins with a segmentation step in order to extract text regions. The segmentation process is based on the maximum gradient difference technique introduced in several video text extraction methods [53, 63].
In accordance with Wong et al.'s method [53], the first step involves calculating the horizontal gradient G of the image I :

$$G = I \star g, \tag{1.1}$$

where, $g = [-1 \quad 1]$.

This step is followed by selecting the maximum and the minimum values of the calculated gradient within a local window centered at each pixel p of size $w \times 1$.

$$\begin{aligned} \forall i, j, \mathrm{MGD}(i, j) &= \max(G(i, p), j - n < p < j + n) \\ &\quad - \min(G(i, p), j - n < p < j + n), \end{aligned}$$

where, $n = \frac{w}{2} - 1$.

Note that, even if the text is over a complex background or onto image regions, MGD values corresponding to text regions are often superior to MGD values of the background or images. The ideal window size depend directly on the characters size. In fact, authors of [53] claim that one of the best choices of the value w is a value that approximates the size of the characters in the text line. Specifically, the 1-D window should be slightly bigger than the characters length.

1.2.1.2 R-signature : A shape descriptor for skew angle detection

The R-signature is a shape descriptor that was introduced by Tabbone et al. [64] in 2005. This descriptor is based on Radon transform and represents a robust approach designed to identify complex shapes.
Let $I(x, y)$ be a image. The Radon transform [64] of the image I is defined by :

$$T_{R^I}(\rho, \theta) = \int\int I(x, y)\delta\left(x\cos(\theta) + y\sin(\theta) - \rho\right) dxdy, \tag{1.2}$$

where $\delta(.)$ is defined as follows :

$$\delta(x) = \begin{cases} 1 & \text{if} \quad x = 0 \\ 0 & \text{otherwise} \end{cases}$$

In our case we are interested only in the binary domain since we detect the text region in the preprocessing stage before the skew detection step.
Let D be a binary shape (a text region in our case). Then, the image I could be represented as follows :

$$f_D(x, y) = \begin{cases} 1 & \text{if } (x, y) \in D \\ 0 & \text{otherwise} \end{cases}$$

To be more explicit, Radon transform describes the scattering data obtained from length intersection of all the lines L_i with the function I for all θ_i and ρ (see Fig. 1.1).

Under these notations, R-signature is defined in [64] as follow :

$$R_I(\theta) = \int_{-\infty}^{+\infty} T^2_{R^I}(\rho, \theta), \tag{1.3}$$

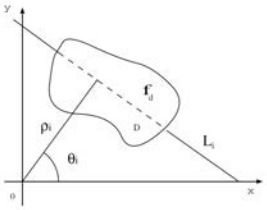


FIGURE 1.1 – Radon transform

1.2.1.3 Algorithm

The main idea of the proposed approach is to detect text lines in a compact representation and then estimate the skew angle (which is the orientation of the text lines) by means of R-signature. In order to improve precision rate of our method we adopted a successive refinement approach. We describe in this paragraph the algorithm of the proposed approach. Fig. 1.2 shows a sample result of the proposed algorithm.

1. At first, we calculate MGD_θ transform of the original image I in three directions $\theta \in \{\alpha_1, \alpha_2, \alpha_3\}$;

$$\mathrm{MGD}_\theta = Rotate_{(-\theta)}\left(\mathrm{MGD}\left(Rotate_\theta\left(I\right)\right)\right), \tag{1.4}$$

   where, $Rotate_\theta(I)$ is the rotation of the image I by an angle θ.

2. On each pixel location (i, j) we calculate the value minimizing MGD as follows :

$$\forall i, j, \mathrm{MGD}_1(i, j) = \mathrm{argmin}_\theta\left(\mathrm{MGD}_\theta(i, j)\right), \tag{1.5}$$

   The resulted MGD transform (MGD_1) is thresholded using a first threshold T_1. Hence, we obtain a binary image B_1 (see Fig. 1.4(b) and Fig. 1.4(c)).
3. We estimate the skew angle θ_0 by means of R-signature : the first search size is set to 1° (see Fig 1.4(f)) ;
4. We calculate then MGD in θ_0 direction and apply a second threshold $T2 > T1$ to calculate a second binary image B_2 (see Fig. 1.2(e) and Fig 1.2(f)) ;
5. We search the final skew angle $\theta_f \in [\theta_0 - 1, \theta_0 + 1]$ by applying R-signature to B_2 : the final search size is set to 0.01° (see Fig. 1.4(g)).

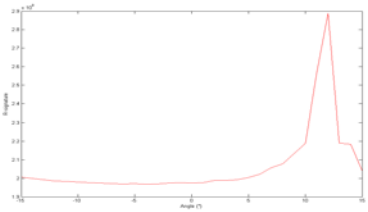
1.2.2 Ridgelet based method for skew detection

In order to improve the precisions of our system, we propsed a second solution for estimating the skew detection of document images. This solution is based on the Ridgelet transform.

(a) A skewed document image : here, the skew angle is set to 12°

(b) MGD_1 is the argmin of the MGD in three directions

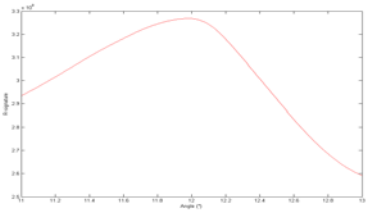
(c) B_1 : The binarized MGD_1 image using the threshold T_1



(d) R-signature of the binary image B_1 with a search size = 1° : θ_0 corresponding to the angle that have a maximum score in the R-signature histogram is calculated. In this example $\theta_0 = 12°$

(e) MGD_f is the MGD of the original image in direction θ_0

(f) B_2 : We used a threshold T_2 to binarize MGD_f



(g) R-signature of the binary image B_2 with a search size = 0.01° : θ_f corresponding to the angle that has a maximum score in this R-signature histogram corresponds to the estimated skew angle : here $\theta_f = 12.01°$

FIGURE 1.2 – Illustration of different steps of the proposed algorithm

1.2.2.1 Ridgelet transform

Many publications related to line detection, texture detection, denoising and symbol representation have been established using ridgelet transform. In fact, this transform describes the singularities of lines included in images and especially the orientation of objects in the document image.

The continuous ridgelet transform was described with details by Candès in [65] and is defined as follows :

$$R_I(a,b,\theta) = \int\int \psi_{a,b,\theta}\,(x,y)\,I(x,y)dxdy, \tag{1.6}$$

where $\psi_{a,b,\theta}\,(x,y)$ and I denote respectively the ridgelets and the image. Ridgelets are defined from a 1-D wavelet-type function $\psi(u)$:

$$\psi_{a,b,\theta}(x,y) = \frac{\psi(x\cos(\theta) + y\sin(\theta))}{a^{1/2}}, \tag{1.7}$$

Given this definition, the continuous ridgelet transform could be seen as a 2-D continuous wavelet transform applied to the set of parameters (ρ_i, θ_i) instead of the parameters (x_i, y_i). Note that the first set of parameters describes here the position of lines while the second expresses point positions.
Similarily to the paper presented by Do et al. [66] lied on our ridgelets implementation a 1-D wavelet transform to the "angle" columns of the radon transform in order to obtain the ridgelet coefficients of our segmented image. Thus the ridgelet transform is linked to wavelet transform via the radon transform.

Figure 1.3 shows the relation between the radon transform and the ridgelet transform.
In our work we apply this transform to the segmented image containing text lines and background in order to estimate the text orientation.

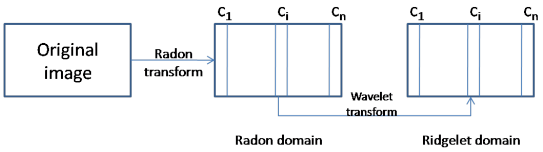


FIGURE 1.3 – The relation between radon and ridgelet transforms

We describe in the next section the method that we adopted for the purpose of finding the skew angle of the document.

1.2.2.2 Angle estimation using the Ridgelet transform

As suggested from the previous paragraph, a ridgelet based method is applied in order to find the skew angle of the document image. However, most of the papers that discussed the ridgelet transform applied a hard thresholding to the ridgelets so as to extract relevent coefficients that describe the image. In our case, we are interested only on determining a single angle which is the skew angle for each document. For this, we propose in this section a robust method to detect the angle in question.
The criteria that we adopted in our work consists in localizing the peaks that represent typical

linear singularities of the binarized image obtained from the segmentation step. We used the determinant $Det_H(\rho_p, \theta_p)$ of the Hessian matrix $H(f)$ at each point $p = (\rho_p, \theta_p)$ (see equation 1.8) to measure the sharpness of peaks in the ridgelet transform f. In fact, this determinant (called also discriminant) describes at each point the local curvature. Finally, we sum the absolute values under each angle column to obtain the ridgelet peaks profile (Some examples are shown in Fig 1.4(f) and Fig 1.4(g)). The angle that maximizes the peaks profile will be considered as the skew angle θ_0 (see equation 1.9).

$$H(f)(\rho_p, \theta_p) = \begin{pmatrix} \frac{\partial^2 f}{\partial^2 \rho}(\rho_p, \theta_p) & \frac{\partial^2 f}{\partial \rho \partial \theta}(\rho_p, \theta_p) \\ \frac{\partial^2 f}{\partial \theta \partial \rho}(\rho_p, \theta_p) & \frac{\partial^2 f}{\partial^2 \theta}(\rho_p, \theta_p) \end{pmatrix} \tag{1.8}$$

$$\theta_0 = \text{argmax}_\theta \left(\sum_\rho |Det_H(\rho, \theta)| \right). \tag{1.9}$$

1.2.2.3 Algorithm

We maintain the main structure of the algorithm proposed in Section §1.2.1.3. However, we substitute the third and the fourth steps in ordre to improve the precision by using the Rigelet transform. Fig. 1.4 shows a sample result of the proposed algorithm.

1. The MGD_θ transform of the original image I is calculated in three directions $\theta \in \{\alpha_1, \alpha_2, \alpha_3\}$. See Equation 1.4.
2. For each pixel (i, j) the value minimizing MGD is calculated as it was defined in Equation 1.5.

   The resulted MGD transform (MGD_1) is thresholded using a first threshold T. Hence, we obtain a binary image B (see Fig. 1.4(b) and Fig. 1.4(c)).
3. We estimate the skew angle θ_0 by means of ridgelet peaks profile described previously : the first search size is set to 1° (see Fig 1.4(f)) ;
4. We search the final skew angle $\theta_f \in [\theta_0 - 2, \theta_0 + 2]$ by applying our ridgelet peaks profile to B : the final search size is set to 0.1° (see Fig. 1.4(g)).

1.2.3 Experimental results

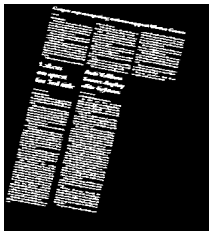
We have evaluated our two proposed methods on the open dataset [2] provided by Chou et al. [67].

This dataset is composed of 500 document images generated by scanning a collection of different documents. These images are selected to represent different kind of documents (news-

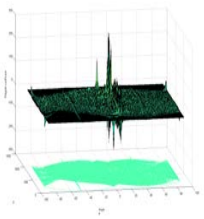
2. http ://ocrwks11.iis.sinica.edu.tw/ dar/Download/WebPages/Skew.htm

(a) A skewed document image : the skew angle is equal to 12°

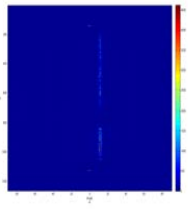
(b) MGD_1 is the argmin of the MGD in three directions



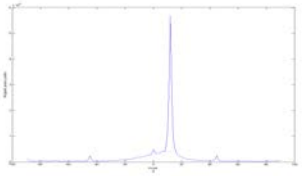
(c) B : We used a threshold T to binarize MGD_1



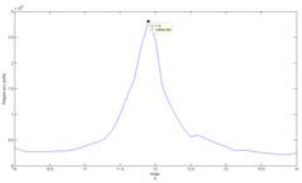
(d) Ridgelet transform of the binarized image B : The figure shows a concentration of peaks for $\theta = 12°$



(e) The determinant of the Hessian matrix is applied on each point : We remark high values for $\theta = 12°$



(f) Ridgelet peaks profile of the binary image B with a search size $= 1°$: θ_0 corresponding to the angle that have a maximum score in the profile is calculated. In this example $\theta_0 = 12$



(g) Ridgelet peaks profile of the binary image B with a search size $= 0.1°$ is applied in the interval $[\theta_0 - 2, \theta_0 + 2]$: θ_f corresponds to the angle that have a maximum score in this profile corresponds to the estimated skew angle : here $\theta_f = 11.9°$

FIGURE 1.4 – A sample result of the proposed algorithm

papers, books, magazines, and journals) and were produced by scanning different documents at 300 dpi. The designers of this database restricted the maximum possible angle to $\pm 15°$.
Chou et al. [67] divided the dataset into 5 categories according to the language and the type of the documents :

1. English documents ;
2. Chinese and Japanese documents ;
3. Documents containing large-scale figures ;
4. Documents containing forms or tables ;
5. Multilingual documents.

For comparison purpose, we selected five other existant methods ; Wavelet [27], PCP [67], PJ [68], TC [69] and CC [70]. The experimental results of these methods on the dataset are available in [27] and [67].
In our evaluation, we decided to decrease the images resolution to 150 dpi in order to accelerate the calculation time. The parameter values of the algorithm are empirically determined : $w = 10$, $T_1 = 90$ and $T_2 = 100$. We also fixed $(\alpha_1, \alpha_2, \alpha_3) = (-10°, 0°, 10°)$ to cover the interval $[-15°$ $15°]$ well. The performance measures are the error mean and the error variance comparing to the ground truth established by the designers of the dataset. Tables 1.1, 1.2, 1.3, 1.4 and 1.5 show the performance of the five existing methods and the proposed methods on each category of the database. These tables show also the performances of the top 80% error rates.

Comparing to the methods [27] and [67] that provide the best results in the litterature for the adopted dataset provided by Chou et al. [67], we remark that our proposed methods perform competitve performance rates especially in terms of variance error. In the next paragraph, we will discuss one of the advantages of the proposed methods.

1.2.4 Multiple skew angles detection

In contrast to the methods [27] and [67], our approaches deal separatly with the connected components (text lines provided by the MGD transform). We remarked that this property provides additional information comparing to [27] and [67]. As an example, we demonstrate in Fig. 1.5 that our approach (method 1) is able to distinguish different dominant skew angles corresponding to the same image ; we remark in Fig. 1.5(c) that the R-signature histogram presents two main peak values. These values approximate the orientation of each of the two skewed pages belonging to the original image of Fig.1.5(a). Note that method 2 presents the same multiple skew angles detection property.

Method	Mean		Variance	
	All images	Top 80%	All images	Top 80%
Method 1	0.240	0.168	0.033	**0.013**
Method 2	0.247	0.187	**0.030**	0.014
Wavelet	0.256	0.208	0.088	0.015
PCP	**0.149**	**0.102**	0.129	0.096
PJ	0.230	0.153	0.206	0.140
TC	0.185	0.148	0.180	0.131
CC	0.166	0.115	0.144	0.109

TABLE 1.1 – Performances on the 1^{st} category

Method	Mean		Variance	
	All images	Top 80%	All images	Top 80%
Method 1	**0.114**	0.071	**0.013**	**0.004**
Method 2	0.132	**0.060**	0.032	0.005
Wavelet	0.126	0.068	0.035	0.005
PCP	0.139	0.088	0.143	0.070
PJ	0.496	0.254	0.591	0.263
TC	0.171	0.108	0.155	0.091
CC	0.180	0.132	0.192	0.096

TABLE 1.2 – Performances on the 2^{nd} category

1.2.5 Conclusion for the skew detection methods

We proposed two robust and precise skew detection methods based on maximum gradient difference. First, the maximum gradient difference serves to segment image into text regions and non-text regions. Second, two strategies were presented in order to estimate the skew angle.

1. In the first method, we proposed the use of the R-signature
2. In the second method we used the determinant of the Hessian matrix of the Ridgelet transform to estimate this angle

Experimental results show that our new skew detection methods perform very well in terms of error variance which means that the proposed methods are robust. Besides, we demonstrated that our methods are able to detect two different dominant skew angles in the same document image. We noticed that method 1 is faster than method 2 and quite precise when the document images contain mainly text regions. However, when the document image presents other regions such as figures and tables, method 2 is able to select only text regions and then gives more satisfactory estimation results. As a futur work, we plan to evaluate the performances of the proposed methods to detect multiple dominant skew angles in scanned document images and to detect the skew in complex document images (that contain text regions over non-uniform

Method	Mean		Variance	
	All images	Top 80%	All images	Top 80%
Method 1	0.488	0.440	0.022	0.014
Method 2	0.485	0.429	0.041	0.034
Wavelet	0.499	0.450	**0.019**	**0.011**
PCP	**0.231**	**0.178**	0.135	0.011
PJ	7.787	3.419	9.049	4.934
TC	0.249	0.183	0.223	0.144
CC	0.345	0.223	0.325	0.186

TABLE 1.3 – Performances on the 3^{rd} category

Method	Mean		Variance	
	All images	Top 80%	All images	Top 80%
Method 1	0.172	0.116	0.021	0.009
Method 2	**0.111**	**0.059**	**0.015**	**0.005**
Wavelet	0.125	0.071	0.021	0.008
PCP	**0.111**	0.062	0.127	0.073
PJ	0.160	0.096	0.163	0.105
TC	0.150	0.078	0.180	0.084
CC	0.139	0.075	0.146	0.078

TABLE 1.4 – Performances on the 4^{th} category

Method	Mean		Variance	
	All images	Top 80%	All images	Top 80%
Method 1	0.154	0.083	0.037	0.004
Method 2	0.115	0.062	0.020	0.004
Wavelet	**0.071**	**0.040**	**0.006**	**0.002**
PCP	0.077	0.051	0.075	0.050
PJ	2.050	0.208	5.816	0.264
TC	0.176	0.105	0.240	0.072
CC	0.197	0.129	0.230	0.125

TABLE 1.5 – Performances on the 5^{th} category

backgrounds).

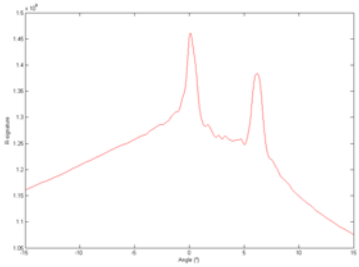
1.3 Document image categorization

In this section, I present our document image categorization method that aims to detect and recognize the different regions of interest in the document. Namely, we are interested in separating and identifying the following regions :

— Text regions

(a) An example of a document image I that presents two dominant skew angles : the first skew angle is equal to 6° while the second is set to 0°

(b) $\mathrm{MGD}_1(I)$



(c) R-signature of the binary image B calculated by thresholding $\mathrm{MGD}_1(I)$: Two main peaks corresponding to the angles 6° and 0° are detected

FIGURE 1.5 – Localizing two different dominant skew angles

— Line regions (separators)
— Photo regions
— Background

Note that in the begining of the document image processing we suppose that the document image is already deskewed.

1.3.1 System overview

Our system consists of four main steps :

1. The Global Stroke Width Variation (GSWV) feature : this step aims to extract the text and line candidates.
2. Active contours model and variation study : at this stage the background and the photo

regions are identified.

3. SVM classification : here, we introduce a new feature vector that allows the text/line separation. At the end of this step, the lines and separators are extracted and the text components are identified.
4. Adaptive projection profile for text clustering : the text components are clustered according to their intensity and sizes. An adaptive projection profile is applied to cluster paragraphs and isolated text lines.

Figure 1.6 shows the system overview diagram and the input/output at each step.

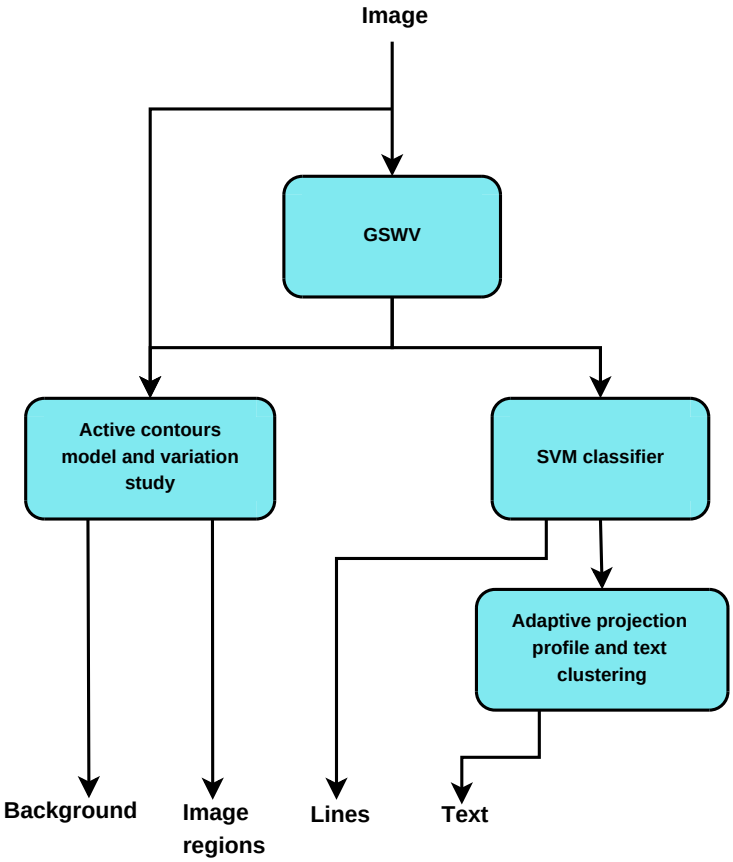


FIGURE 1.6 – System overview

1.3.2 Text and lines extraction using Global Stroke Width Variation (GSWV)

As discussed in the introduction, text and line stroke widths are generally uniform. In this section we describe our text/line descriptor called Global Stroke Width Variation (GSWV) that allows the identification of these regions and then eliminating the photo and background regions by estimating the component stroke width variations. First, the image is binarized using the Maximally Stable Extremal Regions (MSER) blob detection method [71]. A MSER region is

a region that is either brighter or darker than its neighboring background. In order to detect these regions, a set of successive thresholds is applied on the grayscale image which implies a set of binary images ; these images are composed of connected component regions. The stability of a given region is calculated by determining the inverse of its relative area variation when the threshold is increased. Therefore, a readable text and contrasted lines constitute stable regions. The GSWV feature is then applied on both darker and brighter components separately. Before introducing our new feature, let's introduce the following function :

$$D(C) = dist(sk_C) \tag{1.10}$$

where sk_C, is the pruned skeleton corresponding to the component C obtained by the method [72] that eliminates undesired branches. $dist(C)$ is the distance function that calculates the Euclidian distance separating each component pixel of the component C from the nearest background pixel. The pruned skeleton corresponding to text or line component have not (or have very few if there is a noise) branches pointing toward the component's boundary. Hence, to each stroke branch is associated a skeletal branch and then the function D can estimate half the perpendicular projection of the stroke width in each pixel of the skeleton. $D(C)$ has then the same variations as the stroke width of the component C. GSWV feature is calculated as follows :

$$GSWV(C) = \frac{\text{std}(D(C))}{\text{mean}(D(C))} \tag{1.11}$$

where $std(D(C))$ is the standard deviation corresponding to the vector $D(C)$.
Given a component C, if its stroke width is varying a lot comparing to its stroke width average then $GSWV(C)$ is high. Inversely, the $GSWV(C)$ presents a very small value if the stroke width of C is uniform which is the case of text and lines. At the end of this step, all the components having a GSWV below a certain threshold t_1 are considered as text/line candidates and t_1 is determined empirically.

1.3.3 Active contours model and variation study

Once the text and line candidates are extracted, we want to detect the photo and background regions. This step is composed of three main stages : Text/line inpainting, active contours for image segmentation and photos and background identification.

1. Text/line inpainting : In the literature, the inpainting is used to reconstruct lost elements/area in the image. Many works have been devoted to restore natural or artificial missing and damaging portions in the image [73, 74]. Most of the existing works are based on Partial Differential Equations (PDE) where the missed portions are filled by means of diffusion-based techniques [74]. Other works are based on texture analysis and morpholo-

gical operations to reconstruct images [74]. However, these methods are mainly dedicated to restore natural images and depend to several parameters. In this paragraph, we introduce an efficient morphological-based inpainting method to eliminate text and line candidates from document images. For this matter, we identify each text/line candidate by two heuristics :

$$maxStrokeWidth(C) = \max(D(C))$$

$$r(C) = \begin{cases} 1 & \text{if } C \text{ is a dark MSER} \\ -1 & \text{if } C \text{ is a bright MSER} \end{cases}$$

For each component C we propose Algorithm 1.

Algorithm 1 Inpainting process

Data: I ; I is the original image
Result: Ip ; I_p is the inpainted image
initialization
if $(r(C) > 0)$ **then**
  $R_c(Ip) = R_c(open(I, disk, maxStrokeWidth(C)))$; R_c is the area that occupy the connected component C, the structuring element is a disk and its size is equal to maxstrokewidth(C).
else
  $R_c(Ip) = R_c(close(I, disk, maxStrokeWidth(C)))$;
end

Note that the complexity of this algorithm does not depend on the number of components since we sample the set of components based on the maxstrokewidth histogram and we apply the same opening/closing structuring element to all the component belonging to the same interval.

2. Active contours for image segmentation : So far we have constructed the inpainted image Ip. The remaining portions of the document image contain either background or photo regions. These two classes have different characteristics as their intensities and color variations. The goal of this stage is to partition Ip into two different regions and to identify each one. For this reason, we propose the use of Chan and Vese active contour model [75] to separate the two regions that have different color distributions (different means). Mathematically, by given the curve $Cv = \partial\omega$ with $\omega \subset \Omega$ an open subset, and two unknown constants cv_1 and cv_2, denoting $\Omega_1 = \omega$ and $\Omega_2 = \Omega - \Omega_1$, Chan and Vese have proposed to segment an image J by minimizing the following energy with respect to cv_1, cv_2 and

Cv.

$$F\left(cv_1, cv_2, Cv\right) = \nu \left|Cv\right| + \lambda_1 \int_{Cv} \left|u(x,y) - cv_1\right|^2 \mathrm{d}x\mathrm{d}y + \lambda_2 \int_{Cv} \left|u(x,y) - cv_2\right|^2 \mathrm{d}x\mathrm{d}y$$

where ν defines the smoothness of the curve Cv. This parameter controls the segmentation result. The variation of ν leads to either a sub- or over-segmentation. In fact, if the smoothness parameter is very low, then we neglect the variations inside each region and this could lead to an over-segmentation. Contrarily, a high value of it leads generally to a sub-segmentation by decreasing the effect of the second and the third terms in the equation below.

3. Photo and background identification : after the segmentation process, we obtain a set of regions having different colors. We propose to evaluate the color variation inside each region by the use of the following expression :

$$V(A) = \frac{\mathrm{std}(I(A))}{\mathrm{mean}(I(A))} \tag{1.12}$$

where $V(A)$ denotes the variation of the region A, $I(A)$ is the vector that includes the pixel values of the region A. Intuitively, the V value is high small when the region is homogeneous. In fact, the intensity distribution is concentrated on the mean value. This characteristic allows distinguishing background(s) from the photo regions that present usually higher variations. For this reason, a threshold t_2 is determined empirically to separate the two regions. Finally, we construct the bounding boxes that delimit the photo regions.

1.3.4 SVM classifier

We suppose that the text and line regions do not belong to the photos, we eliminate then the text/line candidates that are included in the photo frames extracted during the previous step. This exclusion reduces the false positives since the photos could contain some textures and shapes that are similar to text or lines. Furthermore, most of the page segmentation systems ignore the fact that the photos contain text or lines. The role of this classification step is to separate text and lines. First, we define the following heuristics that are used for this separation :

— Relative thickness (RT)

$$RT(C) = \frac{Ar(sk_C)}{\mathrm{mean}(D(C))} \tag{1.13}$$

Where $Ar(sk_C)$ expresses the area that occupy the skeleton sk_C relative to the component

C. We notice that this heuristic have high values for lines since their stroke widths are very small comparing to the length of the line components.

— Elongation (El) :

$$El(C) = \frac{\text{majoraxis}(C)}{\text{minoraxis}(C)} \tag{1.14}$$

It calculates the ratio between the major and the minor axis of the ellipse that corresponds to the normalized second central moments as the component C. This parameters expresses the elongation of the component C whatever its orientation. Generally, separators and lines have very high elongations comparing to the text words.

— Compacity ($S(C)$) : This scalar varies between 0 and 1 and calculates the density of the component area comparing to its corresponding convex hull area. Note that affine lines have high solidity values. Contrarily, curved lines present low solidity values. Therefore, we define S_1 as follows :

$$S_1(C) = 2 * |S(C) - 0.5| \tag{1.15}$$

This parameter is defined to separate text and lines based on the solidity information. In fact, except some words/fonts, the text presents average solidity values comparing to the lines. Hence, lines present normally S_1 values higher than those of the text.

As discussed, all these parameters could help to separate text from line regions. Therefore, we define the following feature vector FT :

$$\forall C, \quad FT(C) = \begin{pmatrix} RT(C) \\ El(C) \\ S_1(C) \end{pmatrix} \tag{1.16}$$

Then we learn a linear SVM classifier on a training dataset that contains a set of labeled components (1 for text and -1 for lines). Thereby, this classifier permits to separate text and lines. In this work, we consider only thin and long lines in order to identify separators and we do not perform any segmentation process on the other type of lines.

1.3.5 Adaptive projection profile for text clustering

Projection profile methods are base on morphological operations which rely on predefined parameters that depend on the size of the text. The choice of these parameters is made arbitrarily based on the a priori knowledge about the text's size. However, the same image could contain different text regions with different sizes (paragraphs and titles). In this case, most of the existing projection profile methods fail to properly cluster and detect text regions. Some works try to overcome this problem by the use of a multi-resolution representation of the image such as the well known pyramid structure [76]. The multi-resolution representation methods consist generally

on unsupervised approaches and need a predefined parameter which is the number of levels. The wrong choice of this parameter can lead the merging of two different text regions. We propose in this section an adaptive projection profile approach to cluster text regions. First, we start by clustering the set of the text components according to their stroke widths and their colors. For that, we use the mean-shift clustering method [77] which does not require a predefined number of clusters. We assume that two components C_1 and C_2 belonging to the same class need to verify :

1. $\max\left(\frac{\text{mean}(D(C1))}{\text{mean}(D(C2))}, \frac{\text{mean}(D(C2))}{\text{mean}(D(C1))}\right) < a$
2. $|\text{mean}(I(C1)) - \text{mean}(I(C2))| < b$

As known, the log of a quotient is equal to the difference between the logs of the numerator and denominator. Hence, for each component C, the input of the mean-shift model is defined as follows :

$$M(C) = \begin{pmatrix} \log(\text{mean}\,(D(C1))) \\ \text{mean}\,(I(C)) \end{pmatrix}$$

The bandwidth which corresponds to the standard deviation authorized in each class is then equal to :

$$B = \begin{pmatrix} \log(a) \\ b \end{pmatrix}$$

In the end of this step, we apply the projection profile approach described in [78] to each class independently. By accomplishing this process, we can avoid the merging problem described in the beginning of this section. Note that we suppose in this step that text is either horizontal or vertical. For that, we apply only horizontal and vertical projections to group text regions.

Figure 2.16 summarizes the output of the different segmentation process steps. This figure shows the results of the GSWV feature, the image and background segmentation, the line extraction, the text clustering result and the overall final result. The final result shows a segmentation of the different regions in the document image. Note that the text in the original image contains different scales, orientations (horizontal and vertical) and different contrasts.

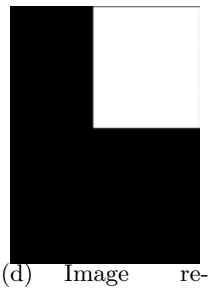
1.3.6 Experimental results

We determined empirically the parameters of our algorithm, namely $t_1 = 0.5, t_2 = 0.2, a = 1.22, b = 40$. In order to evaluate our method, we compared the segmentation rates of our system to those of the existing page segmentation methods that participated in ICDAR page segmentation competition [42]. A benchmark composed of a dataset (called PRImA dataset), a ground truth and a set of results are publicly available. Note that PRImA dataset consists of several ground-truthed document images. Each image is composed of paragraphs, lines, separators and photo regions. Figure 1.8 shows a comparison between the segmentation and recognition results

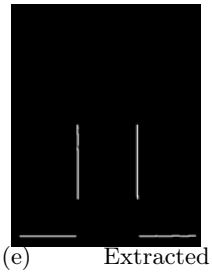
(a) The original document image

(b) Text and line candidates after the first step : GSWV

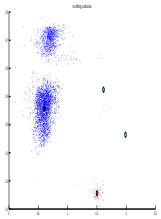
(c) Inpainted image



(d) Image region after the active contours segmentation step

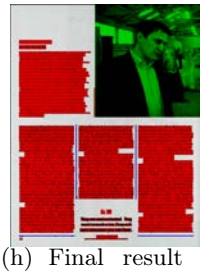


(e) Extracted lines (separators) after the SVM classification step



(f) Mean-shift clustering : here four classes have been selected

(g) Clustered text components : each color corresponds to one cluster



(h) Final result : image region is highlighted in green, text in red and lines in blue

FIGURE 1.7 – An illustration of the several steps of the proposed approach

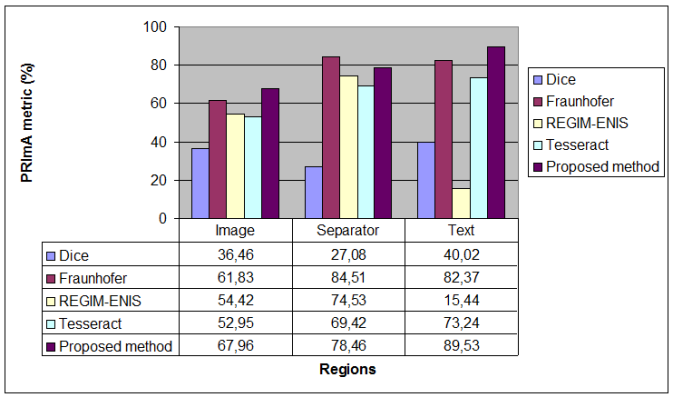


	Image	Separator	Text
Dice	36,46	27,08	40,02
Fraunhofer	61,83	84,51	82,37
REGIM-ENIS	54,42	74,53	15,44
Tesseract	52,95	69,42	73,24
Proposed method	67,96	78,46	89,53

FIGURE 1.8 – Evaluation Results : PRImA measure comparison for different regions

of four existing methods with those of our method using PRImA metric (results and metric are available in [42, 79]). This full-recognition scenario permits to independently evaluate the recognition performances of all the page segmentation methods for different regions (image, separators and text). The results reported in Figures 1.8 show an overall advantage of our method especially in detecting text and image regions. This advantage is due to the fact that we use of a new approach to define and cluster text regions adapted for each scale. Furthermore, the inpainting and the active contours segmentation steps imply good recognition rates for the image regions.

1.3.7 Conclusion for document image segmentation method

We described in this section our new stroke-based page segmentation system. We started by extracting text and line candidates by the use of our GSWV feature that estimate the stroke width variation of each connected component in the image. Then, we described our inpainting process that removes these text/line candidates in order to process the background and photo regions. The photo regions are separated from the background(s) as a result of an active contour segmentation model followed by a variation study. The lines are separated from text by the use of a linear SVM classifier applied on a set of discriminant features. Finally, we proposed to use an adaptive projection profile process to cluster the text regions by means of mean-shift model. Our method consists in a hybrid approach since it combines connected component and region-based processes. It is able to segment and identify lines, background(s), photo regions and multi-scale text. Results on the public dataset PRImA demonstrate the precision and the good recognition rates of our method.

1.4 Conclusion

In this chapter, I have introduced two main contributions in the document image segmentation field.

Our document image categorization approach described in Section §1.3 relies on a stroke descriptor that we apply on the different connected components of the document image in order to identify text and line candidates. Then, the method groups text candidates into horizontal and vertical text regions. However, after scanning documents we could obtain skewed pages. This results in skewed text lines. Hence, the system would fail in grouping text regions. For that, we decided to propose an approach for deskewing the document image (in Section §1.2) before performing the categorization task. Comparing to existing methods, our skew detection and document image categorization methods presented competitive precision rates on two public datasets.

2

Text extraction in real scene images

2.1 Introduction

Our work could be considered as a connected component approach for extracting text in real scene images and more precisely as a stroke-based approach that extends the existing methods of [57, 58]. In general, text strokes present a uniform width and color. Several studies exploit this property. For instance, the Stroke Width Transform (SWT) introduced in [57] permits to determine the width in each part of the connected component and to select those that have low variations (below a certain threshold) considering them as text character candidates. The SWT is calculated by determining the distance separating the two opposite gradients along the edges of the connected component. This method performs efficiently good detection rates when the text is highly contrasted. Alternatively, the gradient along the character's edge is not well determined and the calculation of the stroke width might fail. H. Chen et al. [58] propose a new version of SWT-based on the distance function that overcomes the aforementioned problem. The authors of [58] calculate the distance that separates each pixel in the considered connected component from the background using the distance transform, and then spread the maximum distances along the stroke width. Finally, they consider that the real width of the stroke is twice the resulted transform. However, these methods might fail when the text character presents some variations in its global stroke width (like the case of the letter "w" in Times New Roman font) and when the character overlaps another component. Furthermore, methods [57, 58] do not deal with oriented text lines as most of the existing methods of text detection and the authors consider only the grayscale level of the image neglecting the color information. This information of color is necessary in many cases as most of these methods may fail when the text is not well contrasted or when the text is under non-uniform lighting conditions (i.e. shadows, camera flash, reflections, etc.). In fact, these conditions will affect the recognition process as the color/grayscale of the text characters may change or may not be well contrasted. The existing methods in the field of text

detection perform good detection rates only when : (a) there is an ideal lighting environment, and (b) there is no reflections in the real scene.

Motivated by the works [57, 58], we propose in Section §2.2 a new text detection method based on stroke information. In Section §2.2, we describe the proposed approach regarding our text candidate selection process and present our proposed stroke-based descriptors. Section §2.3 is devoted to the experimental results, where we evaluate and compare our method with some of the existing methods. Next, in Sections §2.4 and §2.5, we discuss the dependency of the proposed approach to the resolution of the image and its computational processing time. Next, we present our conclusion concerning this method in Subsection §2.7.

Furthermore, we proposed during this thesis another approach to detect multi-oriented text lines. I discuss this method that we called stable text line regions in Section §2.7. This method is designed to improve the performances of the method described in Section §2.2. We demonstrate that this method is robust to text of various orientations and is capable to improve greatly the detection rates.

2.2 Proposed text extraction approach

As shown in Figure 2.1 our system consists of six main stages. Namely,
1- Pre-processing step : consists in extracting text candidates using the Maximally Stable Extremal Regions (MSER) [71] from the different color channels. In this step we introduce a selection criterion for the redundant components based on HOG.
2- Graph construction : connected components that are most likely to be non-text are eliminated and similar components are connected by edges.
3- Text descriptors : equal thickness stroke branches, local stroke width variance, and a new method to calculate the global stroke width variance are introduced as descriptors. The first text descriptor is applied on each node of the graph. The nodes that have very low probability to be text (using the text descriptor as a criterion) are eliminated. Then, we refine the connections (edges) of the graph using a DFS criterion in order to keep only text line candidates. The choice of the graph attributes and the DFS algorithm will be discussed in the next sub-sections. The other two descriptors will be exploited in the training part of the classification and refinement step.
4- Graph refinement : this step aims to eliminate undesired nodes and edges of the graph.
5- Graph cuts : at the end of this step, each node of the graph (connected component) is labeled as text or non-text.
6- Classification and refinement : text lines candidates are constructed and false alarms are eli-

minated using a training step.

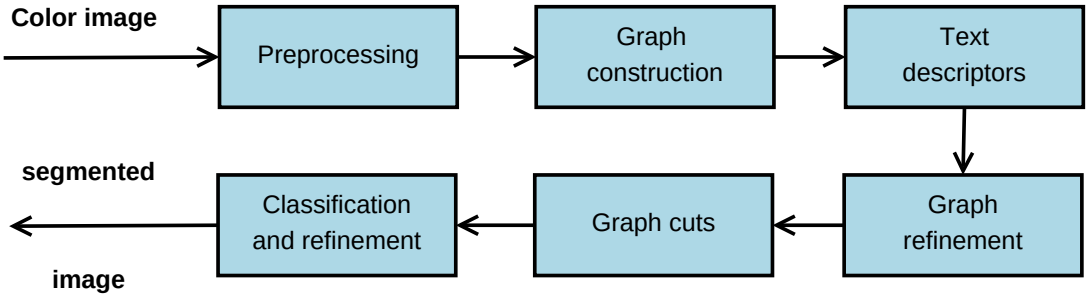


FIGURE 2.1 – System overview

2.2.1 Preprocessing step

Before detecting the connected components in the image, a fitering step is necessary to avoid misdetection and/or false detection due to the noise contained in text and non-text regions. We choose the anisotropic diffusion technique [80] for its ability to preserve the contrast between text and background areas while removing noise and smoothing homogeneous regions.

2.2.1.1 Extracting connected component in a one-dimensional color space

Once the image is filtered, we apply the MSER algorithm [71] to extract connected components. Basically, this technique is used as a blob detection method. A MSER region is a region that is either brighter or darker than its neighboring background. This method demonstrated its superiority comparing to several existing texture patch descriptors [81] in terms of robustness and efficiency. Moreover, the MSER has been used as a preprocessing step for the text detection method in [58] where it was shown that MSER is capable to detect easily text characters as homogeneous components.
In our work, we employed the library of [82] that includes an implementation of the MSER. The MSER function extracts the co-variant areas from an image and considers them as connected components. By fitting ellipses to the extracted regions, this implementation permits drawing elliptical frames around each connected component. The fit is ensured by calculating the center of each region along with the independent components of the variance of the set of pixels composing the considered component. The expressions of the mean μ^c and the variance Σ^c of each region R_c are as follows (See equation 2.1) :

$$\mu^c = \frac{1}{|R_c|} \sum_{x \in R_c} x, \Sigma^c = \frac{1}{|R_c|} \sum_{x \in R_c} (x - \mu^c)^T (x - \mu^c) \tag{2.1}$$

where, $|R_c|$ expresses the area of the set of pixels that belong to the region R_c.
Note that, $\mu^c = (\mu_1^c, \mu_2^c)$ is a 2-dimensional vector and the total number of independent components of the variance is equal to three ; $\left(s_1^c, s_2^c, s_{1,2}^c\right)$ (See Equation 2.3). These five scalars define an ellipse equation where the center is equal to the mean vector μ^c :

$$(x - \mu^c)^T S_c^{-1} (x - \mu^c) = 1 \tag{2.2}$$

where,

$$S_c = \begin{pmatrix} s_1^c & s_{1,2}^c \\ s_{1,2}^c & s_2^c \end{pmatrix} \tag{2.3}$$

In our work, we utilize the parameters of each ellipse to characterize the shape of its corresponding component. To accomplish this task, we project the matrices $\left(S_c^{-1}\right)_c$ onto the eigenspace as follows :

$$S_c^{-1} = P_c^{-1} \begin{pmatrix} \lambda_1^c & 0 \\ 0 & \lambda_2^c \end{pmatrix} P_c \tag{2.4}$$

Hence, the major axis a_c and the minor axis b_c of the ellipse are calculated as follows :

$$a_c = max\left(\lambda_1^c, \lambda_2^c\right), b_c = min\left(\lambda_1^c, \lambda_2^c\right) \tag{2.5}$$

We noticed that the a_c/b_c ratio could approximate the height/width ratio of the connected component regardless of its orientation. This approximation performs much better precision than the classical bounding box technique when text characters are skewed.

In general, the height/width ratio of the text character is close to 1. For this reason, most of the connected component-based methods filter out components that present height/width ratio values greater than 1 from the preprocessing step. However, this value depends on the font and on the character itself. For instance the “i” presents a height/width value much higher than 1. To overcome this problem, we choose to use the mean height/width ratio value of text line candidates as a feature for our classification (See §2.2.6) and not in the preprocessing step.

2.2.1.2 LCH color space

In this subsection we introduce the CIE LCH color space used for extracting text candidates and explain the usefulness of adapting it for a multi-channel text extraction purpose. The CIE LCH color space has been developed by the International Comission on Illumination (Commission Internationale de l'Éclairage) in order to describe the visible color spectrum. It consists of a three dimensional color model (or space) that reproduces the colorfulness of an image using the

three following axes (see Figure 2.2) :
- Lightness or brightness L : it is a vertical axis that varies from 0 (black) to 100 (white).
- Chroma or saturation C : the chroma values vary along the radius of a circle from center 0 (unsaturated colors) to 100 (saturated or pure colors).
- The hue H : for a given saturation value, the hue is an angle that varies from 0° to 360° and represents the color of a given pixel (0° =red, 180° =green, 270° =blue).

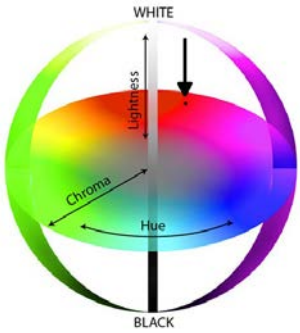


FIGURE 2.2 – The CIE LCH color model [3]

The axes of the LCH color model are independent one to another. This property will help us to properly detect perceptual regions throughout each axis separately. More importantly, the LCH color space is a device independent color model, which means that it's not related to a specific device or material (i.e. scanner, camera or printer) unlike the RGB and the CMYK color spaces that are device dependent spaces.

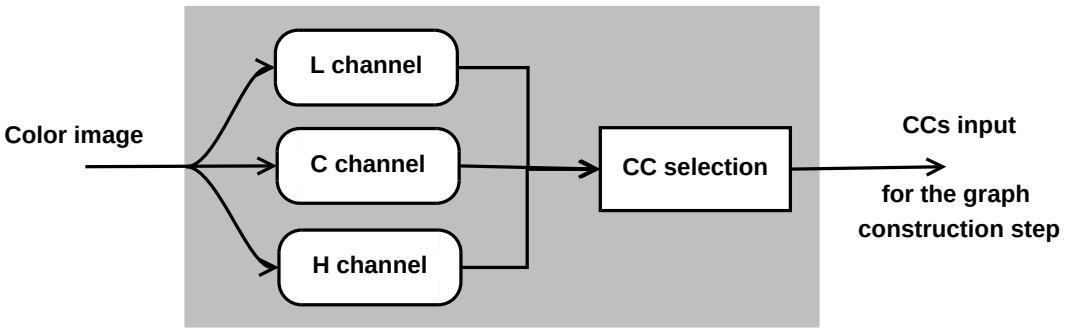


FIGURE 2.3 – Flowchart of the preprocessing step using the LCH color space

3. http ://toyoinkthailand.blogspot.fr/p/gravure-ink.html

2.2.1.3 Histogram of Oriented Gradients Correlation for connected component selection

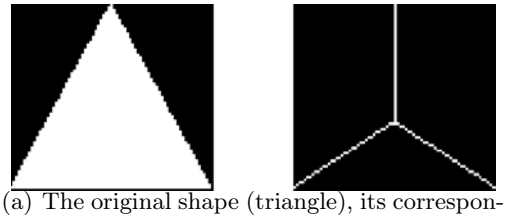
We use the MSER blob detection technique on each axis of the LCH space separately in order to detect all the stable components. However, by applying three times the MSER on the same image, we may extract some redundancies. For instance, a text component could be contrasted relative to its background in terms of saturation and lightness at the same time. In this subsection, we describe our selection criterion to choose the best text candidate if one region is extracted more than once. Note that we consider that a component is redundant if 90% of its pixels are detected at least in two different channels.

The HOG [83] transform is one of the most commonly used descriptors in the computer vision and pattern recognition fields. This transform allows characterizing shapes or zones based on the distribution of their contour directions. To calculate this transform on a region R we start by subdividing it into small cells of n-by-n pixels. Then, each pixel will vote for one class in an orientation-based histogram according to the direction of the gradient in that pixel : each class corresponds to one orientation in the histogram. Let's call n_{cl} the number of classes in the orientation-based histogram. This vote is weighted by the intensity of the gradient on each pixel. After all, the cells are grouped into spatially separated blocks in order to locally normalize the gradients. This process aims to minimize the contrast variation dependency in the same image. In general, the blocks are rectangular (R-HOG) but could be also circular (C-HOG). The HOG consists on the vector of the normalized cell orientation-based histograms, where the number of channels of the histogram corresponds to the number of the defined classes. Therefore, this descriptor depends mainly on the size of the cells, the number of cells in each block, and the number of classes or channels in the histogram. Several text detection methods propose the use of the HOG as a descriptor for text [84, 85]. However, most of these methods (especially the region-based ones) are not robust with respect to the changes in the size of text characters since the results depend on the cell and block sizes. For instance, using a small cell size will emphasize the details and could not define properly the global shape of a big character. On the other hand, if the character size is much smaller than the cell/block sizes, a lot of non-text regions will be considered in the HOG and then the text will be hardly detected. In this work, we propose to adapt the HOG transform as a text descriptor while avoiding the above-mentioned issues. Mainly, the idea behind defining this descriptor is to select the best candidates extracted from the different color channels. Moreover, this descriptor aims to filter out the non-text regions before the processing step. We assume that the best text character candidates respect the following criterion C_1 : "the orientations of the gradients belonging to the same stroke of a text component vary symmetrically along the height-axis of the component". In fact, noisy text components which present noisy contours or many gaps will not satisfy this assumption. So if we apply this criterion

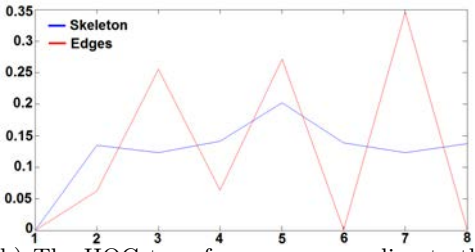
on the redundant components (that occur more than once over the three different color channels), we could select the best candidate. Many non-text shapes do not respect at all this criterion such as triangles which can be filtered out to decrease the complexity of the process. Note that the criterion C_1 implies the following criterion : "for each text character, the orientations of the gradients along the same stroke vary in the same way as in its corresponding skeleton". In the present work, we propose to compare the HOG corresponding to the connected components and the HOG corresponding to their skeletons using the following correlation coefficient :

$$HOG_{C_i} = \frac{(H(C_i) - \text{mean}(H(C_i)))\,(H(sk_i) - \text{mean}(H(sk_i)))}{\text{std}\,(H(C_i))\,\text{std}\,(H(sk_i))} \tag{2.6}$$

where, $H(C_i)$ is the HOG transform corresponding to the component C_i and $H(sk_i)$ corresponds to the HOG transform applied to its skeleton. In order to remain invariant to character size, we consider only one cell per each component. In order to guarantee an accurate precision, we calculate the orientation-based histogram along the eight principal directions ($n_{cl} = 8$). After applying this feature to the redundant components in the three color channels, we compare their corresponding HOG_{C_i} and select the component that corresponds to the channel that gives the highest score. Figure 2.4 shows a comparison between the HOG correlation coefficients corresponding to different shapes (text and non-text). Notice that the histograms are more correlated if the criterion C_1 is respected. Figure 2.5 shows an example of the connected component candidate selection criterion applied on the L, C and H channels.

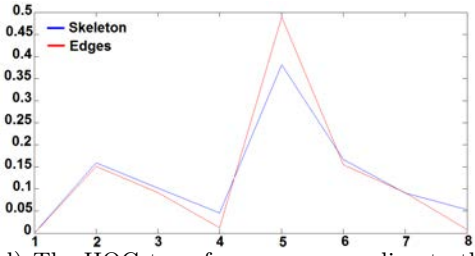


(a) The original shape (triangle), its corresponding skeleton



(b) The HOG transforms corresponding to the shape and to its skeleton : the two histograms are not correlated

(c) The original shape (the character N), its corresponding skeleton



(d) The HOG transforms corresponding to the text character and to its skeleton : the two histograms are correlated

FIGURE 2.4 – A comparison between the HOG of two different shapes and the HOG of their corresponding skeletons

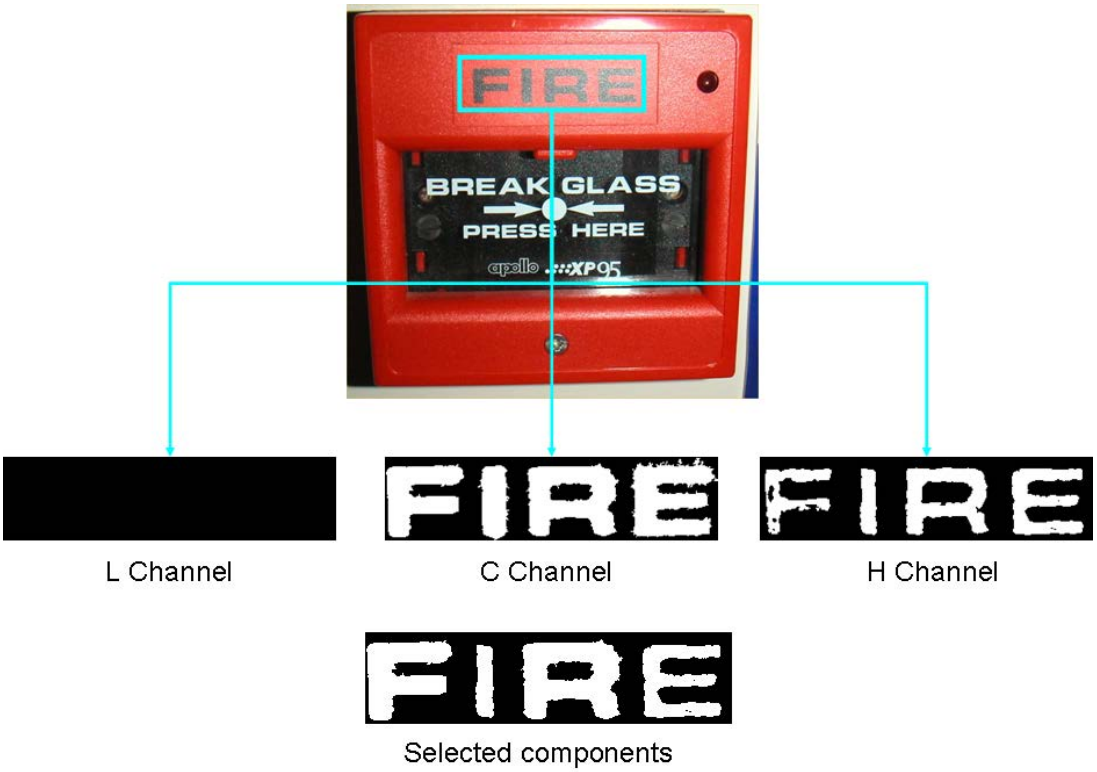


FIGURE 2.5 – Best candidate selection using the HOG correlation criterion : The word "FIRE" has been extracted in the channels C and H at the same time. However, it can be noticed that the appearances of the text candidates change according to their corresponding channels. In case of redundancy, our criterion allows selecting the best candidate which presents rougher contour and clearer interior and exterior appearance. The characters "F" and "R" are extracted from the C channel, while the "I" and "E" characters are extracted from the H channel.

2.2.2 Graph construction

Let I be a color image. We consider the selected connected components obtained from the previous step as the nodes of the graph. Then, we measure the relationship between nodes in order to cluster them by computing similarities among them. The edges of the graph should connect similar and neighboring nodes. For that reason, we propose four binary features that will quantify the similarity and the neighboring of two text candidate regions c_i and c_j (see Table 2.1). The proposed features are chosen in such a way that they are independent from the scale and the orientation of the text.

- Relative spatial distance (RSD) : given that text components belonging to the same text line are close to each other, neighboring nodes are more likely to have the same label (text or non-text).
- Scale ratio (SR) : only the lengths of major axes of components that form the graph are compared since text characters have similar major axis lengths contrary to the minor axes case : for instance, the character "i" presents very low minor axis length compared to the character "m".

However, the two letters have similar major axis lengths.
- Relative position (RP) : the angle of the edge that links the respective centers of each couple of nodes in the graph is calculated. This feature will be the input of our DFS (see §2.2.4) approach that refines the connections of the graph in order to maintain only the connections that look like text lines (regardless of its orientation or its curvature).
- Color difference (CD) : given the assumption that neighboring text characters have the same color.

Feature	Definition
Relative Spatial Distance	$RSD(c_i, c_j) = \frac{\|\mu^{c_i} - \mu^{c_j}\|}{b_{c_i} + b_{c_j}}$
Scale Ratio	$SR(c_i, c_j) = \frac{a_{c_i}}{a_{c_j}}$
Relative Position	$RP(c_i, c_j) = tan^{-1}\left(\frac{\mu_2^{c_i} - \mu_2^{c_j}}{\mu_1^{c_i} - \mu_1^{c_j}}\right)$
Color Difference	$CD(c_i, c_j) = \|I(c_i) - I(c_j)\|$

TABLE 2.1 – binary features for two nodes (c_i,c_j)

As discussed earlier, the nodes of the graph will contain text component candidates and the edges of the graph will link only similar and neighboring components in order to cluster text lines. For this purpose, we trained a linear SVM classifier with a set of nodes consisting of neighboring text characters belonging to the same text line, and a second set composed of random pairs of non-text components. We employed the features RSD, SR, and CD as an input to the SVM classifier. Finally, the results of the classification allow the construction of the edges of the graph. Note that the RP feature is not used during this step, but it will be exploited later for the graph refinement stage (Section §2.2.4).

2.2.3 Text descriptors

Two main reasons motivated us to propose a new and generalized version of SWT to describe the text. The first reason is the fact that the width of text characters has the same value for all the character pixels along the gradient direction of its contour. This width could be approximated by two times the maximum distance value (separating character pixels from the background) of each gradient direction. This maximum value is located in the middle of the stroke. This idea was proposed in [58]. However, this value is unique, so it is useless to calculate the width value for all the pixels belonging to the character components. The second reason consists in eliminating many false reject cases. Generally, a text character has not only one "global" uniform stroke width but also more than one "local" uniform stroke width as we mentioned in the introduction. In [58] the authors did not take into account this fact and eliminate only components having high "global" stroke width variances. Three new text descriptors based on stroke information

are introduced in the following subsections, namely, the equal thickness branches descriptor, the local stroke width variation descriptor, and the global stroke width variation descriptor.

2.2.3.1 Equal thickness branches descriptor

Our first descriptor is based on skeleton and distance transforms. The skeleton should describe fairly the structure of the corresponding text component candidate. Thus, we need to prune the skeletal branches that are not useful for the analysis or that make small contributions to the description of its structure. The idea behind pruning the skeletal branches is based on the fact that branches pointing toward the component's boundary are unnecessary ones [86]. Therefore, we propose to thin the skeleton based on the morphological methodology described in [72]. In order to determine the equal thickness branches, we begin by detecting the intersection and the end points of the pruned skeleton. These points delimit the branches of the pruned skeleton. Note that after pruning, the conserved branches in the skeleton are branches with small variations in terms of thickness. This is the reason for calling them "equal thickness branches". Then, we calculate the percentage of these branches for each text component candidate. Intuitively, text components obey the following criterion C_2 : "components presenting high percentages of "equal-thickness" parts are most-likely to be text components". The percentage is calculated as follows :

$$P_c = \frac{||R_c| - 2\sum_{x \in sk_c} dist(x)|}{|R_c| + 2\sum_{x \in sk_c} dist(x)} \tag{2.7}$$

where, sk_c pixels represent the remainder parts of the skeleton transform of the component C after the pruning step. The variable $dist$ is the distance separating the pixels belonging to the skeleton from the background. Note that the value of P_c is varying in the interval [0, 1]. The lower the P_c value, the more-likely C is a text character. We propose to use an ascendant function of the parameter P_c as the probability of a component of being text.
By performing a statistical analysis of a set of text characters with their corresponding P_c values, we concluded that the resulted histogram of a given text component can be approximated by a folded normal distribution [87]. Since a folded normal distribution has a zero probability mass on the negative part of the x-axis, we can approximate the variance of this distribution by means of maximum likelihood estimator. Therefore, the Probability Density Function (PDF) of a component C being text could be written as follows :

$$f(C) = \frac{2}{\sqrt{2\pi\hat{\sigma}^2}} exp\left(-\frac{P_c^2}{2\hat{\sigma}^2}\right) \tag{2.8}$$

where, $\hat{\sigma}^2$ is the estimated variance. Figure 2.6 shows in the same plot the normalized histogram obtained from the P_c values corresponding to the set of the text components and the estimated probability density function $f(C)$.

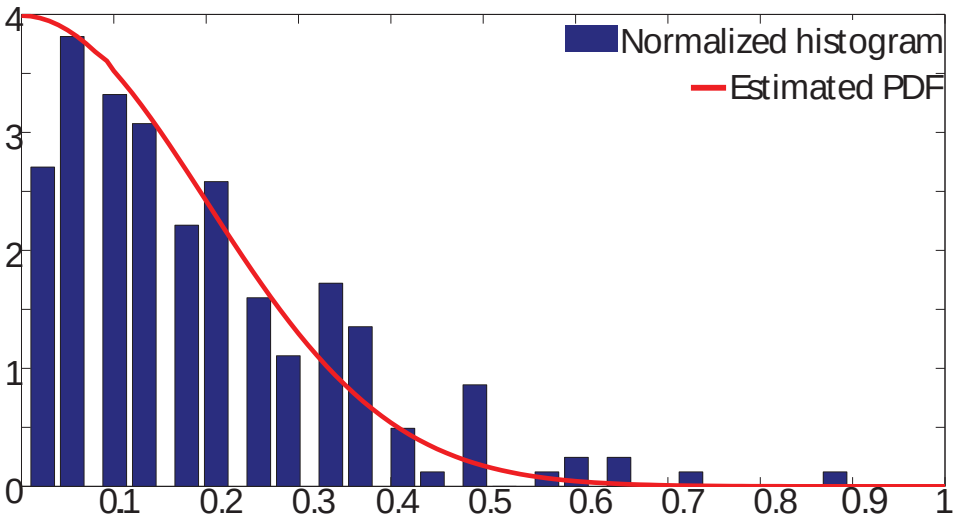


FIGURE 2.6 – The estimated PDF of the component C being text by approximating the normalized histogram. The x-axis shows the domain where the variable C is varying between 0 and 1 and the y-axis illustrates the corresponding PDF values.

2.2.3.2 Local Stroke Width Variation (LSWV) descriptor

The feature vector $D(C)$ described in the previous chapter (See Equation 1.10) has been used to describe the local stroke width variation ($LSWV$). We assume that the variations of the stroke width along the text character are very small comparing to the stroke width values themselves. Notice that (whatever the text font is) the local variation is small. In order to describe this characteristic, we introduce the following equation :

$$LSWV(C) = \text{mean}\,(L(C)) \tag{2.9}$$

where, $L(C)$ is a transform applied on the skeleton of the character candidate C, and is calculated as follows :

$$L(C) = \frac{1}{D(C)} \frac{\partial^2 D(C)}{\partial s^2} \tag{2.10}$$

where, s denotes the intrinsic coordinate of one pixel in the $D(C)$ path.
Note that the LSWV value is high when the relative second derivative of the stroke width function is high, which means that there are high local variations. Conversely, this value is low when the relative second derivative is small, which means that the local stroke width is varying smoothly or uniformly (in this case LSWV value is close to zero). We can obtain a finite difference approximation to the second derivative part by using the following expression :

$$\frac{\partial^2 D(C)}{\partial s^2} \approx D(C, s+1) - 2D(C, s) + D(C, s-1) \tag{2.11}$$

where, $D(C, s)$ corresponds to the value of $D(C)$ function in the pixel s.
The calculation of Equation (2.11) is equivalent to a 1-dimensional Laplacian convolution applied on the $D(C)$ function. Since each branch pixel has exactly two neighbors (except the end points and the intersection points that are ignored), the kernel of this convolution can be written as follows :

1	1	1
1	-2	1
1	1	1

FIGURE 2.7 – Convolution kernel for $LSWV$ calculation

Figure 2.8 illustrates the calculation process of the LSWV descriptor of two different shapes. Notice that a text character component presents a high concentration of low values in the histogram of L (Equation 2.10) compared to the L values of a non-text component. This is due to the fact that the stroke width is varying smoothly which is not always the case for a non-text component.

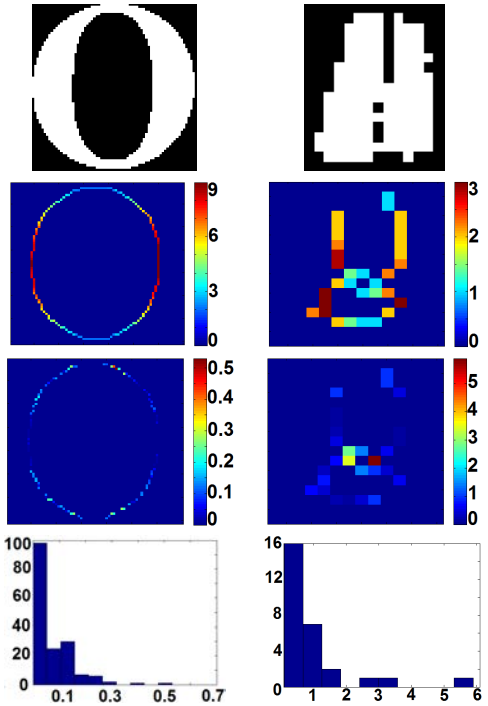
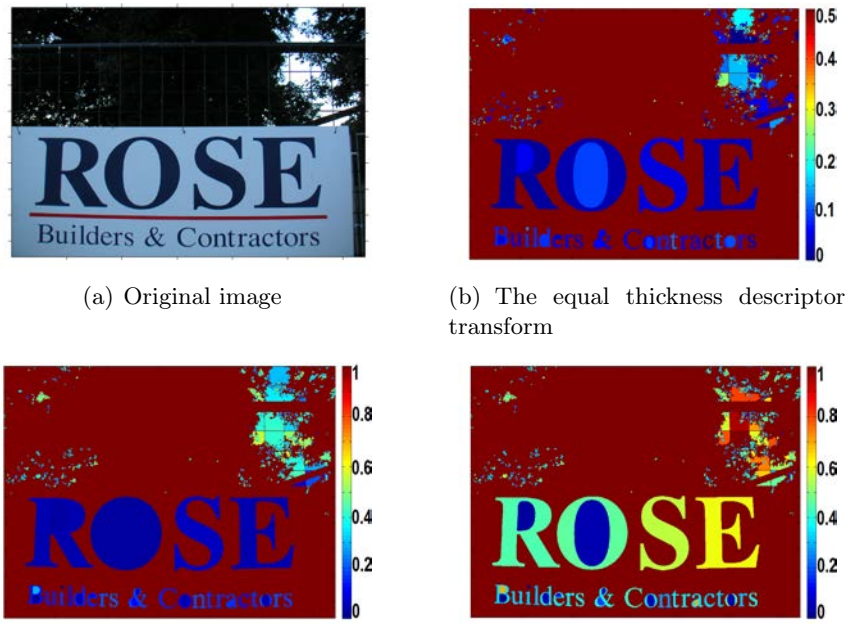


FIGURE 2.8 – An illustration of the LSWV transform : left : a text component, right : a non text component. Top to bottom : the original binary component C - $D(C)$ - Convolution kernel for LSWV applied on $D(C)$ - the histogram of L

2.2.3.3 Global Stroke Width Variation (GSWV) descriptor

So far we have introduced a local feature to describe the smoothness of the stroke width variations. In this subsection we propose the use of the GSWV descriptor that we introduced in Section §1.3.2 as a complementary feature to characterize the global variations of the text components. This feature allows quantifying the following property of the text characters C_3 : "a text component has a small global stroke width variation" (see Equation 2.12). This property varies from a text font to another, for instance, while text characters of the "Arial" font have a small stroke width variation, some of the characters of the "Times New Roman" font present uniform stroke width branches that do not have the same values (for example the character "v" is composed of two branches having different stroke widths).

$$GSWV(C) = \frac{\text{std}(D(C))}{\text{mean}(D(C))} \tag{2.12}$$



(a) Original image

(b) The equal thickness descriptor transform

(c) The local stroke width variance transform ($LSWV$)

(d) The global stroke width variance transform ($GSWV$)

FIGURE 2.9 – An illustration of the stroke width based descriptors applied on the text candidate components. Here we can see that the proposed descriptors could separate text from non-text. In fact, for all the plotted transforms, the text components have almost the same values which are smaller than those of the non-text components.

Figure 2.9 shows an illustration of the text descriptor transforms applied on the extracted components after the preprocessing and the graph construction steps. Note that text regions often have smaller values than the rest of the regions in the image. This figure demonstrates that the use of our set of descriptors can process complex font types. In fact, we notice that for

the "S" and "E" in the word "ROSE" the GSWV values are slightly higher than the other text components. This is due to the global stroke variations. However, these variations are smooth, which explains the fact that they present small LSWV values. For other non-text shapes, both, the LSWV and the GSWV transforms present higher values.

In Section §2.2.6, we demonstrate by a training process the complementarity of these two features.

Algorithm 2 DFS approach for text line refinement

Data: $RP = RP(c_i, c_j)_{i,j}$;
$Edge_i$: for each edge i we calculate its corresponding depth (length of the cluster) and then select all the edges that belong to this cluster ;
PTS : define the valid set of edges that link similar/neighboring nodes ;
Result: D_i : corresponding depth of the edge i ;
C_i : set of edges corresponding to the longest path from edge i ;
$[D_i, C_i] = DFS(RP, Edge_i, PTS_i)$;

if *(isempty(valid_ neighbors(*$Edge_i$*)))* **then**
  $D_i = 1$;
  $C_i = Edge_i$;
else
  v = valid_neighbors($Edge_i$) ;% two edges are valid neighbors iff they are verifying the Equation 2.13 ;
  $PTS = \{PTS\} - \{Edge_i\}$;
  $[D_{vm}, C_{vm}] = argmax(DFS(RP, Edge_{v_k}, PTS), v_k)$; % argmax function takes the attributes of the function DFS corresponding to the longest possible depth.
  $D_i = D_i + D_{v_m}$;
  $C_i = [C_i, C_{v_m}]$;
end

2.2.4 Graph refinement

Before performing the graph cuts and segmenting the image, we need to refine the connections (edges) between nodes according to the text orientation appearance. In this stage we propose a modified DFS process to be applied only to the edges of the graph. This stage eliminates edges that do not belong to a same text line through the parameter RP from Table 2.1. In fact, we suppose that the difference between the angles of two successive edges $(RP(i,j), RP(j,k))$ that link three text characters (i, j, k) should not exceed a certain threshold, even if the text line presents a curvature. Hence, the connections that do not respect the following equation are eliminated.

$$\theta_0 < |RP(i,j) - RP(j,k)| \pmod{2\pi} \tag{2.13}$$

where θ_0 is a threshold that we have found empirically.
Furthermore, by performing a DFS along the edges of the graph we induce a given node to be part of the cluster that contains the greatest possible number of connected nodes that obey criterion C_2 (See §2.2.3.1). Algorithm 2 shows the pseudo-code of the proposed DFS approach.

By the end of this step, we have defined all the valid edges of the text line candidate graph which will be the input of the graph cuts segmentation process.

2.2.5 Graph cuts step

The objective of this step is to identify and minimize a cost function to separate the "text" of "non-text" components. The input to this segmentation step is the graph G described in the previous paragraphs. In this section, we discuss the expression of the objective function of Equation 2.14 that serves to segment the graph (and thus the image) and to label the previously identified text candidates.

$$\begin{aligned} E(G) &= E_{data}(G) + E_{smooth}(G) \\ &= \sum_{C \in G} D_C(l_C) + S \sum_{(C_1,C_2) \in G} V_{C_1,C_2}(l_{C_1}, l_{C_2}) \end{aligned} \tag{2.14}$$

where,

$$S = \alpha f(0) \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

The set of expressions in Equation 2.14 shows the general form of the function applied on the graph. These expressions contain mainly four important terms :
- $D_C(l_C)$: This function calculates the cost of the node belonging to the class. We use the expression of the probability density $f(C)$. Explicitly, we consider the cost of the node belonging to the "text" class equals to $f(C)$. Furthermore, we consider that the cost of the node belonging to the "non-text" class equals to $1 - f(C)$.
- $V_{C_1,C_2}(l_{C_1}, l_{C_2})$: This is the function that quantifies the neighborhood between nodes $C1$ and $C2$. In our study, this is equal to the color difference between them (i.e. CD).
- S : This matrix is determined empirically. It is used to penalize the score when two neighboring nodes belong to two different classes.
- α : A very small value of α makes the process of classification dependent only on the data. However, a high value of α causes a dependence of the classification process on the structure of the graph and not on the data . In order to account for both, data and structure, we determine this value experimentally.
We propose to use the Graph Cuts method in order to solve this problem of energy minimization described above. The resulted graph eliminates non-text characters and maintains the final text

characters before the classification and refinement step.

2.2.6 Classification and refinement

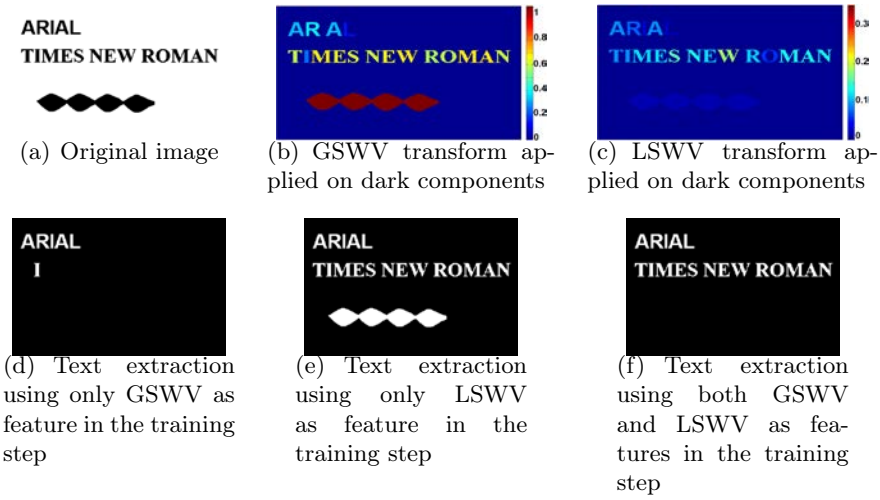
2.2.6.1 Complementarity between the GSWV and LSWV features

Before introducing our classification process, let's discuss firstly the complementarity between the features GSWV and LSWV. The following example (See Figure 2.10) demonstrates that. The original image in Figure 2.10 contains three different regions :
- Text in Arial font.
- Text in Times New Roman font.
- Another curved shape.

Therefore, we can distinguish three different cases. In fact, the first region presents small values in both GSWV and LSWV transforms. The second region presents slightly high values in the GSWV but small values in the LSWV transform. While the third presents high values either in their corresponding LSWV or/and in GSWV transforms. Notice that text components could not present high values in their corresponding LSWV and GSWV transforms. However, under some fonts, text characters could present a slightly high GSWV value but not a high LSWV one. The experiment consists in evaluating the impact of each of these two features. For this reason, we perform three different training and classification processes. In the first training process we include only the GSWV feature. In the second training process we include only the LSWV. Finally, the third training process includes both features. All the three training processes are performed by using the linear SVM classifier trained on seperately (1) GSWV feature (2) LSWV feature (3) both GSWV and LSWV features. The ground truth is composed of several labeled text and non-text components. Therefore, we can conclude that the LSWV and the GSWV are complementary. In fact, according to the text font, each character stroke branch could present a high global stroke width variation while the local variation is small, and vice versa. For this reason we propose to include these two descriptors to the text line feature vector for the training step.

2.2.6.2 Training step

Previously, we have introduced our approach to the classification of text components into "text" and "non-text" components. In this subsection, we propose a classification approach to separate text line candidates into "text" and "non-text" lines. The goal of this step is to increase the precision rates while maintaining a high recall rate by eliminating false positives. In order to detect the final text, we propose a feature vector $F(TL)$ that characterizes each text line TL. This vector includes the four following features :



(a) Original image

(b) GSWV transform applied on dark components

(c) LSWV transform applied on dark components

(d) Text extraction using only GSWV as feature in the training step

(e) Text extraction using only LSWV as feature in the training step

(f) Text extraction using both GSWV and LSWV as features in the training step

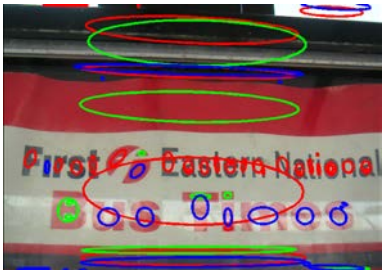
FIGURE 2.10 – The complementarity between the LSWV and the GSWV transforms applied on different text shape candidates.

1) The mean value of the height/width ratio of the text components that belong to TL.
2) The mean value of the LSWV (see Section §2.2.3.2) applied on the text components that belong to TL.
3) The mean value of the GSWV (see Section §2.2.3.3) applied on the text components that belong to TL.
4) The mean solidity value.
5) The mean of the quotient between the lengths of the character's skeleton and their corresponding mean stroke widths.
We trained a Gaussian Kernel SVM on a set of text line candidates in order to separate text from non-text lines by using their corresponding feature vectors F.
A morphological opening is applied to the extracted text components of every text line separately according to its orientation to highlight the location of text regions. Figure 2.11 shows an example that summarizes the main steps of the proposed algorithm.

Figure 2.12 shows a comparison between our proposed method and two existing methods [58, 50]. The method [58] is a connected components-based method where the authors propose the use of a global stroke width feature to describe text. The second method [50] is a region-based method that describes the texture of the text and deal with the multi-oriented text line detection problem. None of them propose any specific process to exploit the color information of the image. We highlighted in green the extracted text regions using [50], [58] and our proposed method. Figure 2.12 demonstrates the different advantages of our proposed method :
1) The robustness to the font changing : If we compare the detection rate of our method to the rate of the method [58] in the first example, we notice that [58] fails to extract some text regions because of the high global stroke width variance. However, our method maintains these

(a) Original image

(b) Connected components (text candidates) after the preprocessing step : each connected components C is represented as an ellipse with the parameters a_c and b_c defined in the preprocessing paragraph. Red for L channel, green for C channel and blue for H channel.



(c) Connected components (text candidates) after the selection step

(d) Graph construction : Neighboring and similar components are connected

(e) Text lines after the graph cuts, DFS and classification steps

(f) Text segmentation after the classification of the text line candidates

FIGURE 2.11 – An illustration of the different steps of the proposed method

(a) Original images

(b) Text extraction using Chen method [58]

(c) Text extraction using Shivakumara method [50]

(d) Text extraction using our proposed method

FIGURE 2.12 – This figure illustrates a comparison between two existing methods and our proposed method. Text regions are highlighted in green. For illustration purposes, we applied in the last example several morphological opening operations to the extracted text components in different angles in order to clearly show the different curved text lines.

Method	Proposed approach	Proposed approach (L channel)	Shivakumara [50]	Chen [58]	Mariano [88]
Precision	79%	82%	73%	53%	54%
Recall	82%	70%	65%	39%	42%

TABLE 2.2 – Experimental results on the multi-oriented text line dataset

Method	Proposed approach	Epshtein	Chen	Minetto	Fabrizio
Precision	74%	73%	73%	63%	46%
Recall	64%	60%	60%	61%	39%

TABLE 2.3 – Experimental results on ICDAR 2003 dataset [89]

components due to the fact that the text line feature vector contains both global and local stroke width variance features.
2) The use of a multi-channel approach allows our method to perform better results than the existing methods (see the third and the fourth examples of the Figure 2.12).
3) The last example in this figure shows that our method is able to detect curved text lines due to the DFS step while both [50] and [58] fail.

2.3 Experimental results

We determined empirically several parameters of our algorithm, namely $\theta_0 = 40°, \alpha = 5$. Our approach was tested on the public datasets ICDAR 2003 [89] and ICDAR 2011 [90] as many text detection studies have been previously evaluated on them. These two datasets were designed for text locating competitions. Each dataset consists of a set of images devoted for training and another set devoted for testing. All the files of the ICDAR datasets are natural scene images that were captured using a digital camera under different luminosity conditions. Mainly, the two classical measures of precision and recall rates are used to present the results. The mathematical formulas of these measures are explicitly defined in [89] for ICDAR 2003 dataset evaluation. The evaluation method proposed in [90] was employed to evaluate our scene text detection approach on the ICDAR 2011 dataset. Note that [90] was designed to overcome the problems of over- , under- and missed segmentation. We compared the performances of our work with several existing text detection methods. For the ICDAR 2003 dataset, we compared our method against the methods proposed in [57, 58, 91, 92] , and for the ICDAR 2011 dataset, we compared our results against the results obtained by the methods that participated in the ICDAR 2011 competition.

Tables 2.3 and 2.4 show that our method performs the best results in terms of recall rates. This can be explained by the fact that we use a multi-channel color selection approach to detect

Method	Proposed approach	Kim	Yi	TH-TextLoc System
Precision	78%	83%	67%	67%
Recall	69%	62%	58%	58%

TABLE 2.4 – Experimental results on ICDAR 2011 dataset [90]

text candidates which allows detecting more text regions. These experiments show also that our proposed method performs promising results comparing to those of existing methods in terms of precision rates.

On the other hand, our method can handle multi-oriented and curved text. Since most of the text lines in the ICDAR dataset are oriented horizontally, we have collected fifty real scene images and posters that contain text regions (where the text lines are multi-oriented, slightly curved or horizontal) in order to construct our own dataset. This dataset contains also photos that present distinct illumination effects. We manually generated the ground-truth of this dataset and we used the same heuristics, i.e. the precision and the recall rates defined in ICDAR competition for evaluations. Figure 2.13 shows a subset of images from this collected dataset. In order to evaluate the performances of our method on this dataset (called from now on multi-oriented text line dataset), we compared it with the methods presented in [50, 58, 88]. Table 2.2 shows the comparison between all these methods in terms of precision and recall rates. It shows that our method performs the best results in terms of precision and recall rates. In fact, our method is designed to be invariant to the scale and the orientation of the text and it is also more robust to the illumination conditions comparing to the other methods. Table 2.2 shows also the detection rates of our proposed method using only the grayscale level (L channel) images. The results demonstrate that the use of the LCH channels increases the recall rate while maintaining a good precision rate. Moreover, the results in Table 2.2 show that our proposed method performs competitive detection rates comparing to the existing methods.

Some detection results using the proposed method are shown in Figure 2.14. This figure shows examples of horizontal, vertical and curved text lines extracted using our method. However, some misdetections are present due to the lack of resolution or the failure of the classification part.

2.4 Dependency to the resolution

As a connected component based method, our approach naturally depends on the resolution of the image. Especially, on the resolution of the text components in the image. In fact, the detection of an extremal region by the MSER depends on its relative area variation when the intensity is increasing. This ratio is affected when the resolution is low and may cause the misdetection of it. In this section, we evaluate the dependency of our text detection approach to

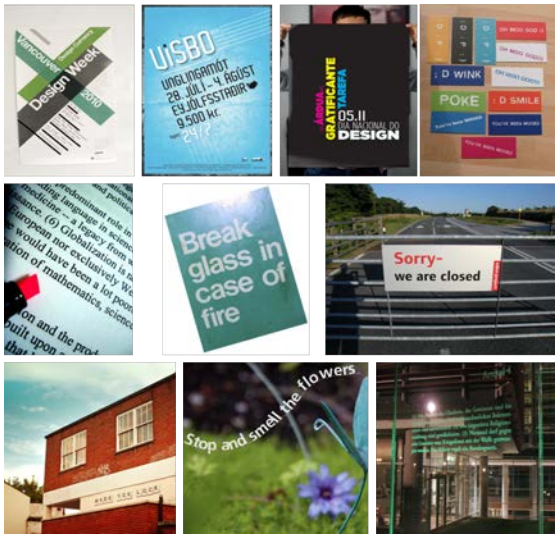


FIGURE 2.13 – A sample set of images from the multi-oriented dataset

(a) Original images (b) Binary images showing the extracted text

FIGURE 2.14 – An illustration of some segmentation results using the proposed method on the multi-oriented dataset

the resolution of the image. For this purpose, we propose to test the robustness to the resolution by creating an artificial dataset composed of 10 pdf images. Each image contains text regions in

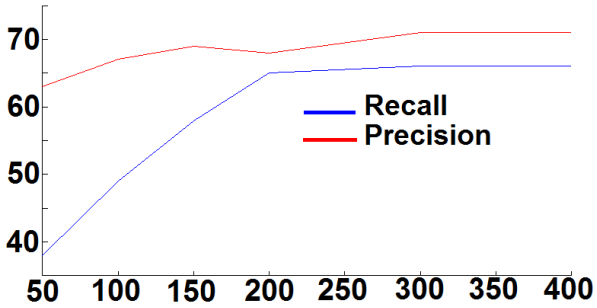


FIGURE 2.15 – A plot that measures the recall and precision rates as a function of the resolution. The x-axis represents the resolution variation while the y-axis represents the rates

Method	Proposed approach	Proposed approach (L channel)	Shivakumara [50]	Chen [58]	Mariano [88]
Time (s)	45.4	23.2	16.2	8.9	16.9

TABLE 2.5 – Computational evaluation

their vector format. Then, we rasterize the images in different resolutions (from 50 dpi to 400 dpi) while evaluating the detection rates by calculating the recall and precision rates. The plot in Figure 2.15 shows the relation between the detection rates and the resolution of the image. This plot demonstrates that our method is robust in terms of precision as the precision rate shows a small variation compared with the resolution variation. However, the relative variation of the recall rates makes evident the fact that the low resolutions affect the detection results. Notice that, in the present evaluation, the detection rates are almost stable from a resolution equal to 200 dpi.

2.5 Computational experiment

We evaluated our approach in terms of processing time by comparing the average execution time of the proposed method (using LCH channels and using L channel only) and those of the methods [50, 58, 88] using the multi-oriented dataset. Note that only [50] is designed to detect non-horizontal text lines. Matlab R2010b has been used to write the scripts on an Intel®Core i5 2.40 GHz machine. Table 2.5 shows the evaluation results. The table shows that our method is slower than existing method. In fact, our method selects text candidates from the L, C and H channels. This implies a high processing time if the image is composed of a lot of connected components. Furthermore, the DFS process that we introduce to detect multi-oriented and curved text line candidates increase the processing time. Our proposed method should be faster if the orientation and the contrast of the image are known, which would reduce the number of text line candidates and therefore decrease the computational time.

2.6 Stable text line regions for multi-oriented text detection

Despite the fact that the proposed DFS approach allows detecting multi-oriented and curved text lines, we can remark that it is not appropriate for cursive text. Furthermore, we noticed that the use of the threshold θ_0 to refine connections between the nodes of the graph causes some mis-detections when the spacing between text characters is small.
In order to overcome this problem, we proposed during this thesis a second solution to deal with the multi-oriented text lines. We called this method "stable text line detection method".
In this section we discuss how this method helps to well detect text of arbitrary orientations by using it instead of the graph refinement step described in Subsection §2.2.4. First, we discuss the new approach of detecting stable text line candidates in Subsection §2.6.1. Finally, we evaluate in Subsection 2.6.2 the performances of our system by comparing it to existing methods in two cases : horizontal and multi-oriented text types.

2.6.1 Stable text line detection

2.6.1.1 Definition

In order to detect multi-oriented text lines, we assume that text characters belonging to the same word are aligned to the same straight segment line. We also assume that the spacing between the characters belonging to one word is constant and different otherwise. In terms of distances between text character candidates, these assumptions imply the following criterion : (C) the Euclidian distance map D that calculates the Euclidian distance between each pixel to the nearest object of interest (text candidates) presents an elongated stable line encompassing the text line regardless of its orientation. The term "stable" is inspired from [71] in detecting extremal regions based on their grayscale level. In fact , if we apply a sequence of thresholding $D_t = (D < t)$ on a distance map D containing text characters ($t = 0, 1, 2, N + 1$, where N presents the maximum distance) we will realize that the thresholded binary image corresponds first to a white image (D_0). Then gradually black regions that correspond to the local minima will appear and grow until we obtain a black image (D_{N+1}). During this process we can see a set of connected components (in black) that are merging together. Obviously, when the threshold is close to the spacing distance between the text characters of the same word, the word component (or the text line) is formed progressively until all the text line components are merged. Then, the component grows constantly. We consider the stability of a text line candidate being the inverse of the relative area variation of the region R (text line candidate) when the thresholding level is increased by δ. With regard to these notations, we admit that the text lines correspond to a set of stable regions.
Let's consider the example of the Figure 2.16. After the "text descriptors" step described in

Subsection§2.2.3 we obtain text character candidates in Figure 2.16(a). Figure 2.16(b) shows the Euclidian distance transform applied on the previous figure. Figure 2.16(c) shows a sequence of thresholding applied to the resulted distance map.
To formally present our stable text line detection approach, let's introduce the following equations :

$$\begin{cases} R = \arg\min_t \frac{\partial |R_t|}{\partial t} \\ \text{with, } \frac{\partial |R_t|}{\partial t} = \lim_{\delta \to 0} \frac{|R_{t+\delta}| - |R_t|}{\delta} \end{cases} \tag{2.16}$$

Where R_t is the t^{th} thresholding level. The resulted R is then the stable text line candidate region detected by our method. We can remark in Figure 2.16(d) that the evolution of the area curve relative to the thresholding variation changes for a threshold t equal to 23. In fact the increasing coefficient of the curve is more important for values less than t. This value corresponds to the threshold that makes all text line components are well merged. It also corresponds to the global minimum of the gradient (See Figure 2.16(e)).

Algorithm 3 Pseudo-algorithm for extracting stable text lines

Data: D,c_1,c_2,...,c_n,%D is the distance map, c_i are the text component candidates.
Result: T_1,T_2,...,T_p, %T_i are the extracted stable text line regions.
R_1,...,R_q=MSER(D),%We extract the maximally extremal regions using [71] from the Euclidian distance map instead of the grayscale level image
R_{i_1},...,R_{i_q}=sort(R_1,...,R_q/μ_{R_1},μ_{R_2},...,μ_{R_q}),%We sort in descending order the stable regions R_i according to their corresponding eccentricities μ_{R_i}
$l = 1$,% l define the indices of the stable text lines
C_{old}={c_1,c_2,...,c_n}

for $j = i_1, ..., i_q$ **do**
  C_t=find_components(R_j, C_{old}),%C_t are the set of the text components that belong to the region R_j and are not yet affected
  if $C_t \neq \{\}$ **then**
    T_l={C_t},% We create a new text line T_l and we affect to it the components of C_t $l = l+1$
    $C_{old} = C_{old} - \{C_t\}$
  end
  if $C_{old} = \{\}$ **then**
    Break,% All the text components are affected to the set of the stable text lines
  end
end

2.6.1.2 Redundancy and refinements

As it was defined in the previous subsection, one text region could belong to more than one stable region (text line candidate). For this matter, we introduce two additional rules :

— If two or more stable regions are too similar (have a high intersection ratio), then, we

Method	Proposed approach	Kim	Yi	TH-TextLoc System
Precision	84%	83%	67%	67%
Recall	66%	62%	58%	58%

Table 2.6 – Experimental results on ICDAR 2011 dataset [90]

preserve the most stable region.

— If a text component belongs to two or more different stable region, we attribute it to the stable region that corresponds to the maximum eccentricity value. The eccentricity of the region R is the ratio between the major- and the minor-axis of the ellipse that corresponds to R's second moment.

Furthermore, text component candidates that are not connected in the graph described in Subsection§2.2.2 could not belong to the same text-line. In fact, this graph describes the similarity and the neighborhood of all the detected text candidates. Hence, non-similar components must be clustered in different text lines. Finally, we eliminate isolated components and very low stable regions before applying the graph cuts step described in Subsection§2.2.5.

Figure 2.17 shows the results of text line detection using the stability criteria on an image containing text under arbitrary orientations. Algorithm 3 shows the pseudo-algorithm that describes briefly the main steps of our proposed method.

2.6.2 Experimental results

2.6.3 Horizontal case

We first evaluated our method using the public dataset ICDAR 2011 [90] that presents a benchmark of text detection in real scene images. Although it is a multi-oriented approach, our proposed method demonstrates competitive recognition rates comparing to the state-of-the art methods (See Table 2.6).

2.6.3.1 Arbitrary orientation case

In order to evaluate our method in detecting text in arbitrary orientation, we used our multi-oriented text line dataset (See Section§2.3 and Figure 2.13). Then we compared our method with the methods [50, 58]. Table 2.7 shows that our method outperforms existing methods since it allows detecting multi-oriented text lines while maintaining a good precision rate.

Method	Proposed approach	Shivakumara [50]	Chen [58]	Mariano [88]
Precision	85%	73%	53%	54%
Recall	83%	65%	39%	42%

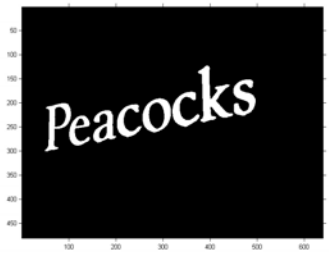
TABLE 2.7 – Experimental results on the multi-oriented text line dataset

2.7 Conclusion

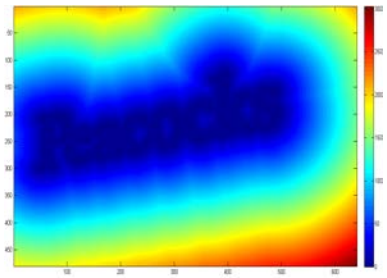
In the first part of this chapter, I introduced our stroke-based approach for text detection in real scenes. Firstly, the method detects the stable regions over the LCH color space using the MSER. A selection criterion has been introduced in order to select the best candidates (in case of redundancy). We also proposed to cluster similar and valid neighboring components in a graph. The connections between the nodes (components) are refined by a DFS approach applied on the edges in order to maintain only the connections between characters that belong to the same text line. A new text descriptor was introduced as an attribute for each node. This descriptor is a function of the percentage of uniform stroke width regions included in the text character. The uniform stroke width regions are detected based on skeleton and distance transforms. Contrarily to related works, the proposed method is able to detect multi-oriented text lines. Furthermore, this method is less sensitive to the different character font types and to the changes in contrast.

In Section §, I presented an improvement to this method by proposing a robust method to detect multi-oriented text lines instead of the DFS approach. We first suppose that we have detected text candidate components. Then, we apply our stable text line detection method to localize multi-oriented text which consists in a rotation invariant method. Furthermore, our proposed method is scale independent which allows detecting text under different sizes. The experimental results on horizontal and multi-oriented text line datasets demonstrate the robustness of this system.

Unlike the DFS approach discussed in Section§2.2, this method does not require any threshold and is able to detect regular spacings between text characters. Furthermore, the detection of the multi-oriented text lines does not depend on the number of text candidates (which is the case of the DFS approach). Hence, this method is much faster when the image presents a lot of text candidates. As a future work, we intend to evaluate the contribution of this method on some other existing methods that detect only horizontal text lines.

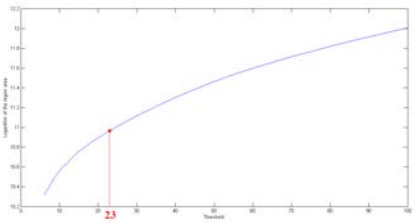


(a) Text candidate components

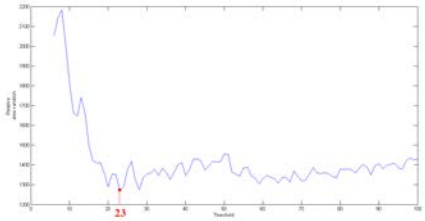


(b) Distance map

(c) Thresholding sequence

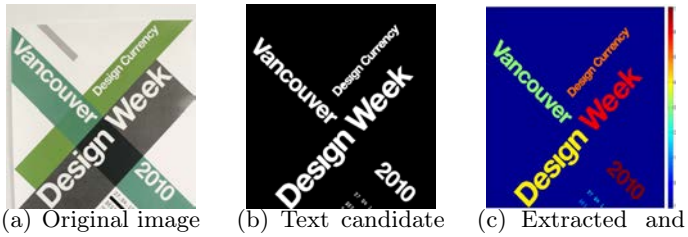


(d) Region area variation of the text line component function of the threshold : for illustration matter we plot the logarithm of the area.



(e) Associated gradient

FIGURE 2.16 – Stable text line detection process



(a) Original image (b) Text candidate components (c) Extracted and isolated text lines

FIGURE 2.17 – An example of the text line detection and separation using the proposed method : different multi-oriented text lines are detected and labeled (each color corresponds to one single text line)

3

Conclusion

3.1 Discussions

In this thesis we dealt with two major domains ; namely text extraction in real scene images and document image segmentation. We presented and discussed our contributions in these two fields in order to insure the segmentation function in most of the image types (for example : photos, posters, magazines and newspapers). In the document image segmentation area we proposed two main contributions. We first have suggested a skew detection method for detecting the orientation of the scanned documents based on the R-signature and the MGD transform. Then we improved the detection rates by means of the discrete ridgelet transform instead of the R-signature. The R-signature allows calculating the orientation of the text line candidates obtained by means of the MGD step. However, the number of these candidates increases significantly if the document image contains mostly non-text regions. For that, we proposed the use of the ridgelets to preselect and enhance the contribution of the aligned structures. As discussion, we noticed that the main advantage of the ridgelets comparing to the R-signature in this application concerns the selection of the lines and the elimination of contribution of the non-line shapes. This implies more angular precision at the cost of the computation time. Thus we advise the use of the first skew detection version if the document contains mainly text lines and, conversely, we advise the use of the second version if the document image contains other regions, more precisely, photo and textured regions. After detecting the skew of the document images we proposed to segment and categorize their contents. We were interested mainly in detecting the following regions of interest : text regions, photo regions, lines (separators) and background(s). We assume that the skew have been already detected and then we were interested only in detecting horizontal and vertical text lines. It is important to note that the final text line detection step could be applied on different orientations. However, we choose to extract only horizontal and vertical text lines. We justify this choice by the fact that the majority of document images contain text in these

two main directions. Furthermore, this choice allows decreasing significantly false positives. We demonstrated in this work that our stroke descriptor designed basically for detecting and extracting text components is also able to detect and extract lines. The separation between text and lines has been insured by a SVM classification step including a feature vector able to distinguish these two regions. We also incorporated the Chen and Vese active contours model in order to detect photo regions. In the end of this work we proposed an adaptive text clustering step based on the Mean-Shift algorithm. This step aims to cluster text components into groups (titles, paragraphs, page numbers etc.). On the other hand, we proposed a new method designed to extract multi-scale and multi-oriented text lines in real scene images. In order to take into account the constraints of the varying size and the orientation of the text we proposed a set of descriptors invariant to the scale and to the orientation of the shapes. For instance we proposed the GSWV descriptor ; a new text descriptor able to describe the global variance of the stroke width. We also proposed a second descriptor that we called LSWV. This descriptor is designed to detect the high local variations inside the shape strokes. This characteristic allows eliminating non-text regions. Both the GSWV and the LSWV descriptors are normalized and then do not depend to the scale of the processed patterns. Furthermore, as most of the stroke based features, our descriptors do not depend to the orientation of text characters. The method ends with a text line clustering step. This step aims to group all the text components that belong to the same text line or to the same word. For that, we proposed a DFS process that consists in searching and selecting the potential text character cluster that constitutes an aligned group of similar components. The advantage of the DFS stage lies in its ability to group curved text lines since it tolerates small angular derivative values between three successive components. However, in some cases we noticed that this step is not able to perform fair clustering. This is due to the eccentricity of the respective component centers that belong to the same word comparing to the principal text line median. In fact, the criterion that we used in order to validate a text line candidate of arbitrary orientation can fail and imply some misdetection cases. For this reason, we proposed a new alternative to cluster components of the same text line independently from its orientation. We called this method stable text line region detection approach. It allows a parallel detection which is applied on a binarized image containing text candidates. This method detects textures presenting regular Euclidian spacing between successive and similar components. Our stability criterion is calculated using the ratio between region area variation and the distance variation between the components inside the same stable region (by applying successive thresholds on the distance map). Thus, all the text components of the same word are grouped and compose a same stable region. Unlike the DFS process, the stable text line detection method is not able to well detect curved text line. However, it is more precise and less vulnerable to the eccentricity of some text characters. The experiments demonstrated that the performances and the detection rates using this method do not depend to the orientation of the text inside the photos. The second

advantage of this method lies in the fact that the computational complexity do not depend on the number of the text candidates but on the size of the image. Finally, this method is able to detect cursive text lines which is not the case of the DFS based approach.
As a conclusion, we proposed two novel segmentation methods of automatically selecting our regions of interest. These regions consist of the content of an input image. The image consists in either a photo where the text is the region of interest or a document image where the text, lines, photo frames and background(s) compose our regions of interest. These two approaches can be used for designing a more general method.

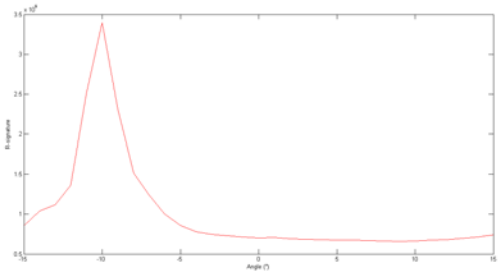
3.2 Future work

As a future work, we plan to generalize our approaches to allow segmenting images without any a priori knowledge about their types. For that, a categorisation step is necessary as a preprocessing to classify images in two groups : photos and document images. After that we apply the suitable method that corresponds to the image type. In the same context, another segmentation application is planned. It consists in segmenting complicated images that we named complex document images. We define these complex document images as classical documents in raster format containing mainly : text, lines and photo frames which contain also text regions. In fact, combining all our proposed methods may insure the segmentation task. The application is mainly made of three steps :

1. Skew estimation : we use our skew detection method in order to correct a potential angular transformation due to the scan (see Figure 3.1).
2. Page segmentation : our proposed page segmentation method is able to detect not only text but also photo frames and lines (see Figure 3.2).
3. Text detection inside photo frames : we apply our multi-oriented text line detection method inside the detected photo frames (see Figure 3.3).

A specific database will be collected and ground-truthed in order to validate this system.
On the other hand, we intend to improve the precisions of our text extraction system. In fact, most of the false alarms that are detected by means of our system consist of regular textured regions. To overcome this problem, we plan to use a sparse representation with discriminative dictionaries. For instance a pre-learned over-complete dictionary could significantly decrease the false alarms. Similarily, a post-processing based on an OCR verification step should improve the segmentation quality.

(a) Skewed image



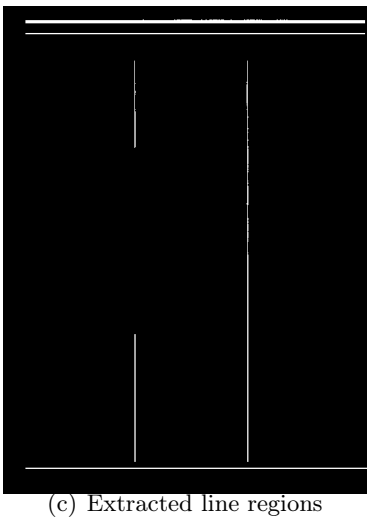
(b) R-signature profile using method 1

(c) Deskewed and cropped image using method 1

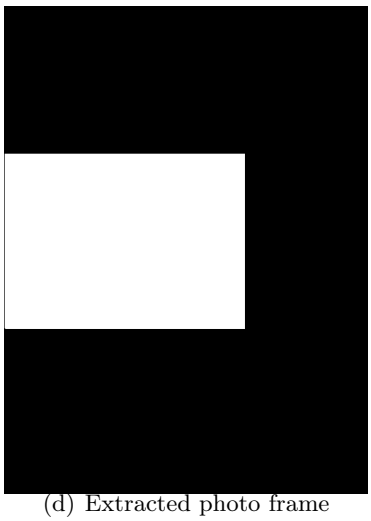
FIGURE 3.1 – Illustration of our R-signature based skew detection method (method 1) applied on a rotated document image

(a) Deskewed document image : text lines are either horizontally or vertically oriented

(b) Extracted text regions



(c) Extracted line regions



(d) Extracted photo frame

FIGURE 3.2 – Segmentation and categorization of a deskewed document image contents

(a) Extracted photo frame

(b) Extracted text regions from the photo frame

FIGURE 3.3 – Our multi-oriented text extraction method applied on the extracted photo frame

Part IV

Thesis Summary in French

1

Introduction

L'objectif principal du domaine de la segmentation des images de documents est de séparer les régions texte des autres régions (photos, graphiques, lignes et bruit). Cela implique le fractionnement et la classification des parties de l'image du document en des blocs isolés. Fondamentalement, le résultat d'une telle segmentation est une image binaire qui indique l'identité de chaque pixel/région. Plusieurs méthodes ont été proposées pour assurer la segmentation des images de documents. Typiquement, un système de segmentation d'image de document comprend classiquement deux étapes : la définition des caractéristiques et la proposition d'un classificateur robuste qui sépare différentes régions et identifie à quelle classe elles appartiennent (texte, image, lignes etc.). De nombreux travaux ont été proposés pour trouver des descripteurs appropriés. Certaines de ces approches, extraient les caractéristiques basiques de l'image, tels que sa fréquence, les variations du gradient, les caractéristiques de la texture ou les informations sur les traits.

Pendant cette thèse, nous avons cherché à assurer l'étape de la segmentation indépendamment de la structure de l'image du document. En particulier, nous avons conçu un système de segmentation qui traite non seulement les images de documents non-réguliers, mais aussi les documents contenant des régions de texte présentant différentes tailles. Dans ce résumé, je décrit notre système de segmentation des images de documents proposé.

Dans les deux prochains chapitres, je décrit nos deux principales contributions dans le domaine de segmentation d'images de document. Concrétement, nous avons proposé une méthode de détection d'inclinaison et une approche de catégorisation d'image de documents. Dans le chapitre 2 je détaille notre nouvelle méthode de détection d'inclinaison qui permet d'ajuster l'image de document avant l'étape de la segmentation. Ensuite, je présente dans le chapitre 3 un système robuste de segmentation à base de trait qui vise à détecter et séparer le texte, les lignes, les photos et l'arriére-plan. Tout d'abord, je présente le système proposé dans la Section §3. Ensuite, je décrit notre descripteur à base de trait conçu pour détecter les candidats du texte et

des lignes dans la Section §3.1. La Section §3.2 est consacrée à la présentation de la technique de la segmentation arriére-plan/image. Dans la suite, les étapes de détection de lignes et de la classification du texte sont décrites dans la Section §3.3. Les résultats expérimentaux de la segmentation conjointement à une comparaison avec d'autres méthodes de segmentation de pages existantes sont présentés dans la Section §3.4.

2

Détection d'inclinaison

Nous présentons dans ce chapitre notre nouvelle méthode de détection d'inclinaison qui permet de détecter et d'ajuster l'inclinaison des documents numérisés. Cette approche à pour objectif d'extraire les régions de texte sous différents contrastes indépendamment de la mise en page et du type du document. Cette méthode est constituée de deux étapes principales : l'étape de segmentation et l'étape d'estimation de l'angle d'inclinaison.

2.1 Maximum Gradient Difference

Comme nous l'avons mentionné plus tôt, notre méthode commence par une étape de segmentation afin d'extraire les candidats de texte. Le processus de segmentation est basé sur la technique de Maximum Gradient Difference (MGD), qui est utilisée dans plusieurs méthodes d'extraction de texte dans les vidéos [53, 63]. Conformément à la méthode Wong [53], la premiére étape consiste à calculer le gradient horizontal G de l'image I :

$$G = I \star g, \tag{2.1}$$

où, $g = [-1 \quad 1]$.

Cette étape est suivie par la sélection des valeurs minimales et maximales du gradient dans une fenêtre locale centrée sur chaque pixel p de taille $w \times 1$.

$$\begin{aligned} \forall i, j, \mathrm{MGD}(i,j) &= \max(G(i,p), j-n<p<j+n) \\ &\quad - \min(G(i,p), j-n<p<j+n), \end{aligned}$$

où, $n = \frac{w}{2} - 1$.

Notez que, même si le texte est sur un fond complexe ou sur des zones d'image, les valeurs MGD correspondants à des régions de texte sont souvent supérieures aux valeurs MGD de l'arriére-plan ou des images. La taille de la fenêtre idéale dépend de la taille des caractéres. En fait, les auteurs de [53] prétendent que l'un des meilleurs choix de la valeur w est une valeur qui se rapproche de la taille des caractéres de la ligne de texte. De maniére particuliére, la fenêtre 1-D devrait être légérement plus grande que la longueur des caractéres.

2.2 La transformée Ridgelets

Beaucoup de publications relatives à la détection de ligne, la détection de texture, débruitage et la représentation de symboles ont été établis à l'aide de la transformée Ridgelets. En effet, cette transformée décrit les singularités de lignes incluses dans les images et en particulier l'orientation des objets dans l'image du document. La transformée continue de Ridgelets a été décrite avec plus de détails par Candès dans [65] et est définie comme suit :

$$R_I(a,b,\theta) = \int\int \psi_{a,b,\theta}\,(x,y)\, I(x,y)dxdy, \tag{2.2}$$

où $\psi_{a,b,\theta}\,(x,y)$ and I représentent respectivement les ridgelets et l'image. Les Ridgelets sont définies à partir à d'une fonction 1-D du type ondelettes $\psi(u)$:

$$\psi_{a,b,\theta}(x,y) = \frac{\psi(x\cos(\theta) + y\sin(\theta))}{a^{1/2}}, \tag{2.3}$$

Compte tenu de cette définition, la transformée continue en Ridgelets pourrait être considérée comme une transformée continu 2-D en ondelettes appliquée à l'ensemble des paramètres (ρ_i, θ_i) à la place des paramètres (x_i, y_i). A noter que le premier ensemble de paramètres décrit ici la position des lignes tandis que le second exprime celle des points.

De même pour le document présenté par Do et al. [66], dans notre implémentation de ridgelets, nous avons appliqué une transformée 1-D en ondelettes sur les colonnes de la transformée de Radon en vue d'obtenir le coefficients de ridgelets de notre image segmentée. Ainsi la transformée ridgelet est liée à une transformée en ondelettes par la transformée de Radon.

La Figure 2.1 montre la relation entre la transformée de Radon et transformée Ridgelet. Dans notre travail, nous appliquons cette transformation à l'image segmentée contenant des lignes de texte et du fond d'image afin d'estimer l'orientation du texte.

Nous décrivons dans la sous-section suivante la méthode que nous avons adoptée dans le but de trouver l'angle d'inclinaison du document.

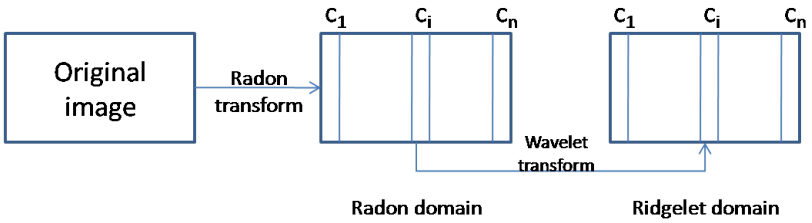


FIGURE 2.1 – La relation entre la transformée de radon et la transformée ridgelets.

2.3 Estimation de l'angle avec la transformée de Ridgelets

Comme suggéré dans le paragraphe précédent, une méthode basée sur la transformée de Ridgelets est appliquée afin de trouver l'angle d'inclinaison de l'image du document. Cependant, la plupart des papiers qui se sont intéressés à la transformée Ridgelets ont appliqué aux ridgelets un seuil fixe de manière à extraire des coefficients pertinents qui décrivent l'image. Dans notre cas, nous nous intéressons uniquement à la détermination d'un seul angle qui est l'angle d'inclinaison pour chaque document. Pour cela, nous proposons dans cette section une méthode robuste pour détecter l'angle en question. Les critères que nous avons adoptés consistent à localiser les pics qui représentent les singularités linéaires typiques de l'image binarisée obtenue à partir de l'étape de segmentation. Nous avons utilisé le déterminant $Det_H(\rho_p, \theta_p)$ de la matrice hessienne $H(f)$ à chaque point $p = (\rho_p, \theta_p)$ (voir équation 2.4) pour mesurer la netteté des pics dans la transformée de Ridgelets f. En fait, ce déterminant décrit à chaque point une courbure locale. Enfin, nous avons fait la somme des valeurs absolues sous chaque colonne d'angle pour obtenir le profil des pics de Ridgelets (quelques exemples sont présentés dans la Figure 2.2(f) et Figure 2.2(g)) L'angle qui maximise le profil des pics sera considéré comme l'angle d'inclinaison θ_0 (voir équation 2.5).

$$H(f)(\rho_p, \theta_p) = \begin{pmatrix} \frac{\partial^2 f}{\partial^2 \rho}(\rho_p, \theta_p) & \frac{\partial^2 f}{\partial \rho \partial \theta}(\rho_p, \theta_p) \\ \frac{\partial^2 f}{\partial \theta \partial \rho}(\rho_p, \theta_p) & \frac{\partial^2 f}{\partial^2 \theta}(\rho_p, \theta_p) \end{pmatrix} \tag{2.4}$$

$$\theta_0 = \text{argmax}_\theta \left(\sum_\rho |Det_H(\rho, \theta)| \right). \tag{2.5}$$

2.3.1 Algorithme

L'idée principale de l'approche proposée consiste à détecter les lignes de texte dans une représentation compacte et ensuite estimer l'angle d'inclinaison (qui est l'orientation des lignes de texte) en utilisant la transformée en Ridgelets. Afin d'améliorer le taux de précision de notre méthode, nous avons adopté une approche par raffinement successifs. Nous décrivons dans ce paragraphe l'algorithme de l'approche proposée. La Figure 2.2 illustre l'algorithme de la méthode de détection d'inclinaison proposée.

1. La transformée MGD_θ de l'image d'origine I est calculée dans 3 directions $\theta \in \{\alpha_1, \alpha_2, \alpha_3\}$. Voir équation 2.6.

$$\mathrm{MGD}_\theta = \mathit{Rotation}_{(-\theta)} \left(\mathrm{MGD} \left(\mathit{Rotation}_\theta \left(I\right)\right)\right), \tag{2.6}$$

   où, $\mathit{Rotate}_\theta(I)$ représente l'image I pivotée par un angle θ.

2. Pour tout pixel (i, j) la valeur qui minimise MGD est calculé comme défini dans l'équation 2.7.

$$\forall i, j, \mathrm{MGD}_1(i, j) = \mathrm{argmin}_\theta \left(\mathrm{MGD}_\theta(i, j)\right), \tag{2.7}$$

   La transformée MGD résultante (MGD_1) est seuillée avec un premier seuil T. Ainsi, on obtient une image binaire B (voir Figure 2.2(b) et Figure 2.2(c)).

3. On estime l'angle d'inclinaison θ_0 à l'aide du profil des pics Ridgelet décrit précédemment : la première taille de recherche est fixée à 1° (voir Figure 2.2(f)) ;
4. Nous recherchons l'angle final d'inclinaison $\theta_f \in [\theta_0 - 2, \theta_0 + 2]$ en appliquant notre profil de pics des Ridgelets à B : la taille finale de recherche est fixée à 0.1° (voir Figure 2.2(g)).

2.4 Résultats expérimentaux

Nous avons évalué notre méthode proposée sur la base de données publique disponibles dans la littérature [4] fournie par Chou et al. [67].

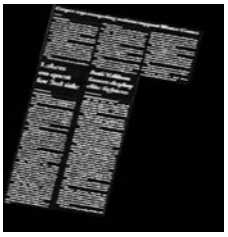
Cette base de données est composée de 500 images de documents générés par numérisation d'une collection de documents différents. Ces images sont sélectionnées pour représenter différents types de documents (journaux, livres, magazines et revues) et ont été produits par la numérisation de documents à une résolution de 300 dpi. Les concepteurs de cette base de données limitent l'angle maximum à ±15°.

Chou et al. [67] ont répartis les données en 5 catégories en fonction de la langue et le type de documents :

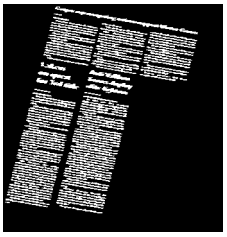
1. documents en anglais ;
2. documents japonais et chinois ;
3. documents contenant des figures grande échelle ;
4. documents contenant des formulaires ou des tableaux ;
5. documents multilingues

4. http ://ocrwks11.iis.sinica.edu.tw/ dar/Download/WebPages/Skew.htm

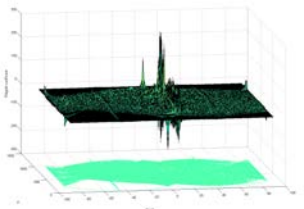
(a) Une image de document incliné de 12°



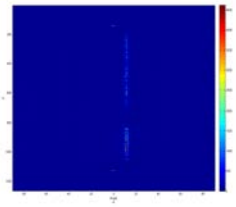
(b) MGD_1



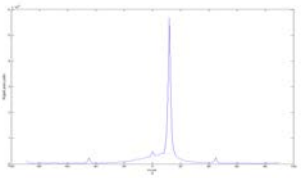
(c) B : MGD_1 image binarisée



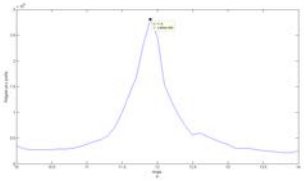
(d) La transformée de Ridgelets de l'image binarisée B



(e) Le déterminant de la matrice Hessienne



(f) Le profil des Ridgelets de B avec une taille de recherche = 1°



(g) Le profil des Ridgelets de B avec une taille de recherche = 0.1° : ici $\theta_f = 11.9°$

FIGURE 2.2 – Un exemple d'illustration de l'algorithme proposé.

Afin d'évaluer notre méthode, nous avons sélectionné cinq autres méthodes existantes : Ondelette [27], PCP [67], PJ [68], TC [69] et de CC [70]. Les résultats expérimentaux de ces méthodes sur l'ensemble de données sont disponibles dans [27] et [67].
Dans notre évaluation, nous avons décidé de diminuer la résolution des images à 150 dpi afin d'accélérer le temps de calcul. Les valeurs des paramètres de l'algorithme sont déterminées de manière empirique : $w = 10$ et $T = 100$. Nous avons aussi fixé les valeurs $(\alpha_1, \alpha_2, \alpha_3) = (-10°, 0°, 10°)$ pour bien couvrir l'intervalle $[-15°\ 15°]$.

Les mesures de performance sont la moyenne de l'erreur et la variance de l'erreur comparée à la vérité terrain mis en place par les concepteurs de l'ensemble de données. Les tableaux 2.1, 2.2, 2.3, 2.4 et 2.5 montrent les performances des 5 méthodes existantes et de notre méthode proposée pour chaque catégorie de la base de données. Ces tableaux montrent également les performances des meilleurs 80% en taux d'erreur. En comparaison avec les méthodes [27] et [67] qui fournissent les meilleurs résultats dans la littérature pour l'ensemble des données fournies par Chou et al. [67], nous remarquons que notre méthode donne des taux de performance compétitifs notamment en termes de variance d'erreur. D'autre part, contrairement aux méthodes existantes, la méthode proposée est capable d'identifier plusieurs angles de rotations de pages dans une seule image de document.

Méthode	Moyenne		Variance	
	Toutes les images	Top 80%	Toute les images	Top 80%
Méthode proposée	0.247	0.187	**0.030**	**0.014**
Ondelette	0.256	0.208	0.088	0.015
PCP	**0.149**	**0.102**	0.129	0.096
PJ	0.230	0.153	0.206	0.140
TC	0.185	0.148	0.180	0.131
CC	0.166	0.115	0.144	0.109

TABLE 2.1 – Performances sur la première catégorie.

Méthode	Moyenne		Variance	
	Toutes les images	Top 80%	Toute les images	Top 80%
Méthode proposée	0.132	**0.060**	**0.032**	**0.005**
Ondelette	**0.126**	0.068	0.035	**0.005**
PCP	0.139	0.088	0.143	0.070
PJ	0.496	0.254	0.591	0.263
TC	0.171	0.108	0.155	0.091
CC	0.180	0.132	0.192	0.096

TABLE 2.2 – Performances sur la deuxième catégorie.

A la fin de cette étape on peut ajuster l'image du document incliné et ainsi faciliter la tâche de la segmentation. La section prochaine sera dédiée à la description de notre méthode de

Méthode	Moyenne		Variance	
	Toutes les images	Top 80%	Toute les images	Top 80%
Méthode proposée	0.485	0.429	0.041	0.034
Ondelette	0.499	0.450	**0.019**	**0.011**
PCP	**0.231**	**0.178**	0.135	0.011
PJ	7.787	3.419	9.049	4.934
TC	0.249	0.183	0.223	0.144
CC	0.345	0.223	0.325	0.186

TABLE 2.3 – Performances sur la troisième catégorie.

Méthode	Moyenne		Variance	
	Toutes les images	Top 80%	Toute les images	Top 80%
Méthode proposée	**0.111**	**0.059**	**0.015**	**0.005**
Ondelette	0.125	0.071	0.021	0.008
PCP	**0.111**	0.062	0.127	0.073
PJ	0.160	0.096	0.163	0.105
TC	0.150	0.078	0.180	0.084
CC	0.139	0.075	0.146	0.078

TABLE 2.4 – Performances sur la quatrième catégorie.

Méthode	Moyenne		Variance	
	Toutes les images	Top 80%	Toute les images	Top 80%
Méthode proposée	0.115	0.062	0.020	0.004
Ondelette	**0.071**	**0.040**	**0.006**	**0.002**
PCP	0.077	0.051	0.075	0.050
PJ	2.050	0.208	5.816	0.264
TC	0.176	0.105	0.240	0.072
CC	0.197	0.129	0.230	0.125

TABLE 2.5 – Performances sur la cinquième catégorie.

segmentation et de catégorisation du contenu des images de documents.

3

Présentation générale du système de segmentation

Notre système se compose de quatre étapes principales (voir Figure 3.1) :

1. Descripteur de la Variation Globale de la Largeur du Trait (VGLT) : cette étape vise à extraire les candidats de texte et de lignes.
2. Modèle du contour actif et l'étude de la variation : à ce stade l'arrière-plan et les régions photos sont identifiés.
3. Classification SVM : ici, on introduit un nouveau vecteur de caractéristiques qui permet la séparation texte/lignes. A la fin de cette étape, les lignes et les séparateurs sont extraits et les composantes texte sont identifiées.
4. Profil de projection adaptatif pour la classification du texte : les composantes texte sont regroupées selon leurs intensités et leurs tailles. Un profil de projection adaptatif est appliqué afin de structurer les paragraphes et les lignes de textes.

3.1 Descripteur de la Variation Globale de la Largeur du Trait (VGLT)

Comme mentionné dans l'introduction, les largeurs de traits pour le texte comme pour les lignes sont généralement uniformes. Dans cette section nous décrivons notre descripteur appelé Variation Globale de la Largeur du Trait (VGLT) qui permet l'identification de ces régions puis qui élimine les régions photos et arrière-plan en estimant les variations de largeur de trait de chaque composante. Tout d'abord, l'image est binarisée en utilisant la méthode de détection de blobs *"Maximally Stable Extremal Regions (MSER)"* [71]. Une région MSER est une région qui est soit plus foncée, soit plus claire que l'arrière-plan qui l'entoure. Afin de détecter ces régions, un

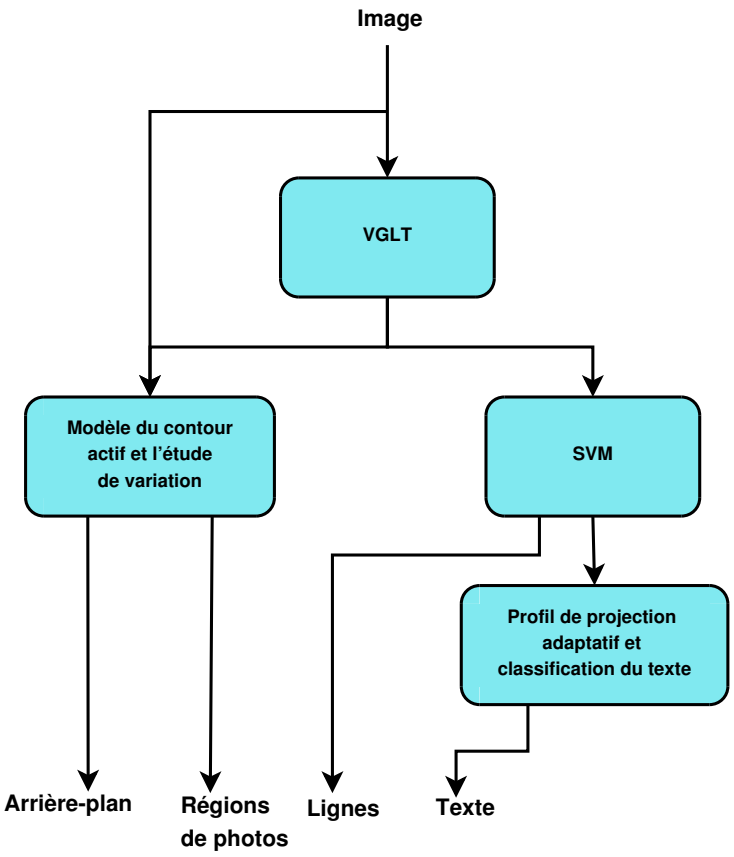


FIGURE 3.1 – Schéma d'ensemble du système de segmentation proposé

ensemble de seuils successifs est appliqué sur l'image en niveau de gris ce qui implique un ensemble d'images binaires. Ces images sont constituées par des composantes connexes. La stabilité d'une région donnée est calculée en déterminant l'inverse de sa variation relative de l'aire lorsque le seuil en question augmente. Par conséquent, un texte lisible ou une ligne contrastée constituent une région stable. Le VGLT est ensuite appliqué sur chaque composante (claire ou foncé) séparément. Avant d'introduire notre descripteur, nous commençons par introduire la fonction suivante :

$$D(C) = dist_C(sk_C) \tag{3.1}$$

Où sk_C, est le squelette ébarbulé correspondant à la composante C obtenu par la méthode [72] qui élimine les branches indésirables. $dist_C$ est le résultat de la fonction distance qui calcule la distance euclidienne séparant chaque pixel de la composante C du pixel le plus proche de l'arrière-plan. Pour chaque branche de trait, nous associons une branche du squelette ébarbulé. Puis, la fonction D permet d'obtenir pour chaque composante C le vecteur $D(C)$ permettant d'estimer la moitié de la projection perpendiculaire de la largeur de trait pour chaque pixel du squelette. $D(C)$ possède alors les mêmes variations que la largeur du trait de la composante C.

Le descripteur VGLT se calcule comme suit :

$$VGLT(C) = std(D(C))/mean(D(C)) \tag{3.2}$$

Où $std(D(C))$ est l'écart-type correspondant au vecteur $D(C)$. Soit la composante C. Si sa largeur de trait varie beaucoup par rapport à sa largeur de trait moyenne alors $VGLT(C)$ est élevé. Inversement, $VGLT(C)$ présente une valeur faible si la largeur de trait de C est uniforme ce qui est le cas pour le texte et pour les lignes.
A la fin de cette étape, toutes les composantes ayant un $VGLT$ au dessous d'un certain seuil t_1 sont considérées comme des candidats de texte ou de lignes. La valeur de t_1 est déterminée de manière empirique.

3.2 Modèle du contour actif et étude de la variation

Une fois les candidats de texte et de lignes extraits, nous envisagons de détecter les régions de photos et d'arrière-plan. Cette partie est composée de trois principales étapes : l'*inpainting* du texte et des lignes, le contour actif pour la segmentation de l'images et l'identification des photos et arrière-plan.

1. L'*inpainting* du texte et des lignes : A cette étape de l'approche, les régions du texte et des lignes sont détectées. Ainsi, avant de détecter les régions homogènes de l'image, il serait très utile d'éliminer les régions déjà détectées. Une façon intuitive de les éliminer est de mettre la valeur 255 (couleur blanche du fond) à la place des régions texte/lignes. Toutefois, ces régions ne reposent pas toujours sur des arrières plans de couleurs blanches. De plus, les arrières plans des documents scannés présentent généralement du bruit. Ainsi, les transitions entre les parties déjà éliminées et l'arrières plan ne seront pas des transitions lisses. Pour ces raisons, nous avons choisi d'effectuer une étape d'*inpainting* afin d'éliminer proprement les régions détectées et de préserver les transitions lisses des zones homogènes et de l'arrière plan avant de les détecter.
   Dans la littérature, l'*inpainting* est utilisé pour reconstruire les éléments/zones perdus de l'image. De nombreux travaux ont été consacrés à la restauration naturelle ou artificielle des parties endommagées ou manquantes de l'image [73, 74]. La plupart des ouvrages existants sont basés sur les équations aux dérivées partielles (PDE) où les parties manquantes sont remplis à travers des techniques basées sur la diffusion [74]. D'autres travaux sont basés sur l'analyse de la texture et des opérations morphologiques pour reconstruire les images [74]. Cependant, ces procédés sont à la base adaptés pour restaurer des images naturelles et dépendent de plusieurs paramètres. Dans ce paragraphe, nous introduisons une méthode efficace basée sur l'analyse morphologique et l'*inpainting* pour éliminer du

texte et les lignes candidates des images du document. A cette fin, nous identifions chaque candidat de texte/ligne par deux heuristiques :

$$maxStrokeWidth(C) = max(D(C))$$

$$r(C) = \begin{cases} 1 & \text{si } C \text{ représente une MSER foncée (minimum local)} \\ -1 & \text{si } C \text{ représente une MSER claire (maximum local)} \end{cases}$$

Pour chaque composante C, nous proposons l'algorithme 4.

Algorithm 4 Processus d'*inpainting*

Data: I; % I est l'image originale
Result: Ip; % I_p l'image finale
Pour toute composante C
if $(r(C) > 0)$ **then**
  $R_c(Ip) = R_c(open(I,'disk', maxStrokeWidth(C)))$; % R_c est la surface qu'occupe la composante connexe C. La fonction open(I, 'disk',maxStrokeWidth(C)) représente l'ouverture morphologique où l'élément structurant est un disque dont le rayon est égale à maxstrokewidth(C)+1 (on rajoute 1 à la valeur de maxstrokewidth(C) pour couvrir toute la surface).
else
  $R_c(Ip) = R_c(close(I,'disk', maxStrokeWidth(C)))$; idem, avec close qui représente la fermeture morphologique.
end

A noter que la complexité de cet algorithme ne dépend pas du nombre de composants puisque nous échantillonnons l'ensemble des composants en nous basant sur l'histogramme de maxstrokewidth et nous appliquons le même élément structurant d'ouverture / fermeture sur tous les composants appartenant à un même intervalle.

2. Le modèle contour actif pour la segmentation de l'image : jusqu'ici, nous avons construit l'image retouchées Ip. Les parties restantes de l'image du document contiennent soit du fond de l'image ou des régions de l'image. Ces deux classes ont des caractéristiques différentes comme l'intensité et les variations de couleur. Le but de cette étape est de partitionner Ip en deux régions différentes et d'en identifier chacune. Pour cette raison, nous proposons l'utilisation du modèle de Chan et Vese pour le contour actif [75] afin de séparer les deux régions qui ont deux différentes distributions de couleurs (moyennes différentes). Mathématiquement, étant donnée la courbe $Cv = \partial\omega$ avec $\omega \subset \Omega$ un sous-ensemble ouvert , et deux constantes inconnues cv_1 et cv_2, ce qui signifie $\Omega_1 = \omega$ et $\Omega_2 = \Omega - \Omega_1$, Chan et Vese ont proposé de segmenter une image J en minimisant l'énergie suivante en considérant cv_1, cv_2 et Cv.

$$F\left(cv_1, cv_2, Cv\right) = \nu\left|Cv\right| + \lambda_1 \int_{Cv} \left|u(x,y) - cv_1\right|^2 \mathrm{d}x\mathrm{d}y + \lambda_2 \int_{Cv} \left|u(x,y) - cv_2\right|^2 \mathrm{d}x\mathrm{d}y$$

où ν définit le lissage de la courbe Cv. Ce paramètre contrÃ´le le résultat de la segmentation. La variation de ν conduit soit à une sur-segmentation, soit à une sous-segmentation. En fait, si le paramètre de lissage est très faible, alors nous négligeons les variations à l'intérieur de chaque région et cela pourrait conduire à une sur-segmentation. Au contraire, une valeur élevée de celui-ci conduit généralement à une sous-segmentation en diminuant l'effet du deuxième et troisième termes de l'équation ci-dessus.

3. L'identification de l'image et du fond de l'image : après le processus de segmentation, nous obtenons un ensemble de régions ayant différentes couleurs. Nous proposons d'évaluer la variation de couleur à l'intérieur de chaque région, en utilisant l'expression suivante :

$$V(A) = std(I(A))/mean(I(A)) \tag{3.3}$$

où $V(A)$ représente la variation de la région A, $I(A)$ est le vecteur qui inclut les valeurs des pixels de la région A. Intuitivement, la valeur de V est peu élevée lorsque la région est homogène. En fait, la répartition d'intensité est concentrée sur la valeur moyenne. Cette caractéristique permet de distinguer le fond de l'image des régions de l'image qui présentent des variations généralement plus élevées. Pour cette raison, un seuil t_2 est déterminé de manière empirique pour séparer les deux régions. Enfin, nous construisons les cadres de sélection qui délimitent les régions de photos.

3.3 Le classificateur SVM

Dans cet approche, nous supposons que les zones de texte et de lignes n'appartiennent pas aux photos, nous éliminons alors le texte / lignes candidates qui sont inclus dans les cadres de la photo extraits au cours de l'étape précédente. Cette hypothèse réduit les faux positifs puisque les photos pourraient contenir des textures et des formes qui ressemblent à du texte ou à des lignes. En outre, la plupart des systèmes de segmentation de pages ignorent le fait que les photos contiennent du texte ou des lignes. Le rÃ´le de cette étape de classification consiste à séparer du texte et les lignes. Premièrement, nous définissons les caractéristiques suivantes qui sont utilisés pour cette séparation :

— Epaisseur relative (RT) :

$$RT(C) = Ar(sk_C)/mean(D(C)) \tag{3.4}$$

Où $Ar(sk_C)$ représente l'aire (le nombre de pixels) de la région sk_C. Nous remarquons que cette variable présente des valeurs élevées pour les lignes puisque leurs largeurs de trait sont très petits comparé à la longueur des composants de la ligne.

— Allongement (El) :

$$El(C) = majoraxis(C)/minoraxis(C) \tag{3.5}$$

Cette variable calcule le rapport entre le grand axe ($majoraxis$) et le petit axe ($minoraxis$) de l'ellipse qui correspond aux second moments centraux normalisés du composant C.

— Compacité ($S(C)$) : Ce scalaire varie entre 0 et 1 et calcule la densité de la zone du composant comparé à sa surface convexe correspondante de l'enveloppe. Notez que les lignes droites ont des valeurs élevées de compacité. Au contraire, des lignes courbes présentent de faibles valeurs de compacité. Par conséquent, nous définissons S_1 comme suit :

$$S_1(C) = 2 * |S(C) - 0.5| \tag{3.6}$$

Ce paramètre est défini pour séparer le texte et les lignes basées sur les informations de compacité. En fait, dans quelques polices particulières, le texte présente des valeurs de compacité moyennes comparé aux lignes. Par conséquent, les lignes présentent normalement des valeurs S_1 supérieures à celles du texte. Tel que discuté précédemment, tous ces paramètres peuvent aider à séparer le texte de régions des lignes. Par conséquent, nous définissons la fonction vecteur (FT) suivante :

$$\forall C, \quad FT(C) = \begin{pmatrix} RT(C) \\ El(C) \\ S_1(C) \end{pmatrix}$$

Ensuite, nous apprenons un classificateur SVM linéaire sur un ensemble de données d'apprentissage contenant un ensemble de composants marqués (1 pour le texte et -1 pour les lignes). Ainsi, ce classificateur permet de séparer du texte et des lignes. Dans cet approche, nous ne considérons que des lignes fines et longues afin d'identifier les séparateurs et nous n'avons pas testé le processus de segmentation sur d'autres types de lignes.

3.4 Résultats expérimentaux

Nous déterminons empiriquement les paramètres de l'algorithme proposé, à savoir $t_1 = 0.5, t_2 = 0.2, a = 1.22, b = 40$. Afin d'évaluer notre méthode, nous avons procédé à une comparai-

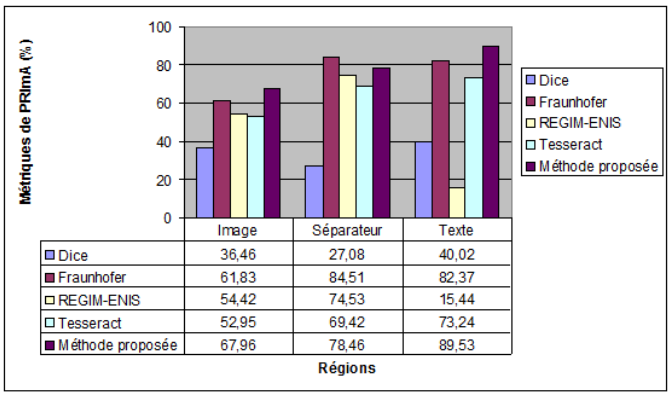


	Image	Séparateur	Texte
Dice	36,46	27,08	40,02
Fraunhofer	61,83	84,51	82,37
REGIM-ENIS	54,42	74,53	15,44
Tesseract	52,95	69,42	73,24
Méthode proposée	67,96	78,46	89,53

FIGURE 3.2 – Résultats expérimentaux en utilisant les métriques de PRImA pour les différentes régions.

son des taux de segmentation de notre méthode à ceux des méthodes de segmentation des pages existantes qui ont participé à la compétition *"ICDAR page segmentation competition"* [42]. Un système d'évaluation (*benchmark*) composé d'une base de donnée appelée PRImA, une vérité terrain et un ensemble de résultats est publiquement disponible. Il est à noter que la base de donnée PRImA se compose de plusieurs images de documents réelles annotées dans des fichiers XML. Chaque image est composée de paragraphes, lignes, séparateurs et régions de photos. La Figure 3.2 montre une comparaison entre les taux de segmentation de reconnaissance relative à quatre méthodes existantes avec ceux de notre méthode en utilisant les métriques PRImA (résultats et métriques sont disponible dans [42, 79]). Ce scénario permet d'évaluer les performances de reconnaissance de toutes les méthodes de segmentation de pages pour différentes régions (image, séparateurs et texte) et ceci d'une manière indépendante.

Les résultats présentés dans la Figure 3.2 montre un avantage global de notre méthode, en particulier, dans la détection des régions de texte et des images. Cet avantage est dÃ» au fait que nous utilisons une approche adaptative afin de définir et regrouper les régions de texte. En outre, l'*inpainting* et les étapes de segmentation de contour actif impliquent de bons taux de reconnaissance pour les différentes régions.

3.5 Conclusion

Dans ce résumé, nous avons décrit notre solution pour segmenter et catégoriser le contenu des images de documents numérisés.

Dans un premier lieu, nous avons proposé une nouvelle méthode pour la détection d'inclinaison des documents numérisés. Tout d'abord, la différence des gradients nous a permis de segmenter l'image en zones de texte et zones de non-texte. Ensuite, et afin d'estimer l'angle d'inclinaison, nous avons utilisé le déterminant de la matrice Hessienne appliqué à la transfor-

mée ridgelets. Les résultats expérimentaux ont montré que cette méthode présente de bonnes précisions surtout en termes de variance de l'erreur. Ce qui signifie que la méthode proposée est robuste. En outre, nous avons démontré que cette méthode est capable de détecter plusieurs angles d'inclinaison dominants différents dans la même image de document.

Une fois le document ajusté, nous avons décrit notre nouveau système de segmentation de pages à base de descripteur de trait. Pour cela, nous avons commencé par extraire les candidats de texte et de lignes en utilisant notre descripteur VGLT qui estime la variation de la largeur du trait pour chaque composante connexe dans l'image. Ensuite, nous avons décrit notre processus d'*inpainting* qui permet l'élimination de ces candidats de texte et de lignes afin de traiter les régions de photos et d'arrière-plan. Les régions photos sont séparées de l'arrière-plan à la suite d'une segmentation basée sur le modèle de contour actif. Les lignes sont séparées du texte par l'utilisation d'un classificateur linéaire SVM appliqué sur un ensemble de caractéristiques discriminantes. Enfin, nous avons proposé d'utiliser un processus de profil de projection adaptatif dans le but de regrouper les régions de texte à l'aide du modèle *mean-shift*. Notre méthode consiste en une méthode hybride car nous combinons des processus basés sur les composantes connexes et ceux basés sur les régions. Cette méthode est capable de segmenter et d'identifier les lignes, l'arrière-plan, les régions photos et le texte ayant une taille variable (multi-échelle). Les résultats sur la base de donnée publique PRImA montrent la précision et les bons taux de reconnaissance de notre méthode.

Bibliography

[1] Gerard Salton. *Automatic Text Processing : The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley, 1989.

[2] David S. Doermann. The indexing and retrieval of document images : A survey. *Computer Vision and Image Understanding*, 70(3) :287–298, 1998.

[3] Jaety Edwards, Yee W. Teh, Roger Bock, Michael Maire, Grace Vesom, and David A. Forsyth. Making latin manuscripts searchable using ghmm's. In Lawrence K. Saul, Yair Weiss, and LÃ©on Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 385–392. MIT Press, Cambridge, MA, 2004.

[4] Avinash Tiwari and Veena Bansal. Patseek : Content based image retrieval system for patent database. In *ICEB*, pages 1167–1171, 2004.

[5] Ze-nian Li, Osmar R. Zaiane, and Zinovi Tauber. Illumination invariance and object model in content-based image and video retrieval. *Journal of Visual Communication and Image Representation*, 10 :219–244, 1999.

[6] Shazia Akram and Mehraj-Ud-Din Dar. Document image processing - a review. *International Journal of Computer Applications*, 10 :0975 – 8887, 2010.

[7] D. J. Crandall. Extraction of unconstrained caption text from general-purpose video. Department of Computer Science and Engineering. The Pennsylvania State University. Technical report, 2001.

[8] Huiping Li and David Doermann. Text enhancement in digital video using multiple frame integration. In *ACM Multimedia*, pages 385–395, 1999.

[9] K. Falkenstern, A. Lindner, S. Susstrunk, and N. Bonnier. Semantic-driven selection of printer color rendering intents. In *IS&T/SID 20th Color and Imaging Conference*, pages 323–328, 2012.

[10] A. Lindner, A. Shaji, S. Susstrunk, and N. Bonnier. Joint statistical analysis of images and keywords with applications in semantic image enhancement. In *Proceedings of the 20th ACM international conference on Multimedia*, pages 489–498, 2012.

[11] Patrick Haffner, Léon Bottou, Paul G. Howard, and Yann LeCun. Djvu : Analyzing and compressing scanned documents for internet distribution. In *ICDAR*, pages 625–628, 1999.

[12] Ioannis Giannoukos, Christos-Nikolaos Anagnostopoulos, Vassilis Loumos, and Eleftherios Kayafas. Operator context scanning to support high segmentation rates for real time license plate recognition. *Pattern Recognition*, 43(11) :3866–3878, 2010.

[13] Han Wang, Stan Z. Li, and S. Ragupathi. A fast and robust approach for document segmentation and classification. In *MVA*, pages 333–336, 1996.

[14] Nazih Ouwayed and Abdel Belaid. A general approach for multi-oriented text line extraction of handwritten documents. *IJDAR*, 15(4) :297–314, 2012.

[15] A. Zahour, B. Taconet, and S. Ramdane. Contribution à la segmentation de textes manuscrits anciens. In *Confrence Internationale Francophone sur l Ecrit et le Document*, 2004.

[16] Wafa Boussellaa, Abderrazak Zahour, Haikal El Abed, Abdellatif BenAbdelhafid, and Adel M. Alimi. Unsupervised block covering analysis for text-line segmentation of arabic ancient handwritten document images. In *ICPR*, pages 1929–1932, 2010.

[17] Laurence Likforman-Sulem, Anahid Hanimyan, and Claudie Faure. A hough based algorithm for extracting text lines in handwritten documents. In *ICDAR*, pages 774–777, 1995.

[18] A. Bennasri, A. Zahour, and B. Taconet. Extraction des lignes d un texte manuscrit arabe. In *Vision Interface*, pages 42–48, 1999.

[19] E. Öztop, Adem Yasar Mülayim, Volkan Atalay, and Fatos T. Yarman-Vural. Repulsive attractive network for baseline extraction on document images. *Signal Processing*, 75(1) :1–10, 1999.

[20] Fei Yin and Cheng-Lin Liu. Handwritten chinese text line segmentation by clustering with distance metric learning. *Pattern Recognition*, 42(12) :3146–3157, 2009.

[21] S. Audithan and RM. Chandrasekaran. Document text extraction from document images using haar discrete wavelet transform. *European Journal of Scientific Research*, 36(4) :502–512, 2009.

[22] M. Fujii and W. J R Hoefer. Field-singularity correction in 2-d time-domain haar-wavelet modeling of waveguide components. *IEEE Transactions on Microwave Theory and Techniques*, 49(4) :685–691, 2001.

[23] John G. Daugman. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *J. Opt. Soc. Am. A*, 2(7) :1160â1169, 1985.

[24] S. Sabari Raju, Peeta Basa Pati, and A. G. Ramakrishnan. Gabor filter based block energy analysis for text extraction from digital document images. In *DIAL*, pages 233–243, 2004.

[25] S. Mori, C.Y. Suen, and K. Yamamoto. Historical review of ocr research and development. *Proceedings of the IEEE*, 80(7) :1029 –1058, jul 1992.

[26] A. D. Bagdanov and J. Kanai. Projection profile based skew estimation algorithm for jbig compressed images. In *ICDAR*, pages 401–406. IEEE Computer Society, 1997.

[27] S. Li, Q. Shen, and J. Sun. Skew detection using wavelet decomposition and projection profile analysis. *Pattern Recognition Letters*, 28(5) :555–562, 2007.

[28] G. Nicchiotti and C. Scagliola. Generalized projections : A tool for cursive handwriting normalization. *Document Analysis and Recognition, International Conference on*, 0 :729, 1999.

[29] L. O'Gorman. The document spectrum for page layout analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15(11) :1162–1173, 1993.

[30] D. S. Le, G. R. Thoma, and H. Wechsler. Automated page orientation and skew angle detection for binary document images. *Pattern Recognition*, 27(10) :1325–1344, 1994.

[31] L. Najman. Using mathematical morphology for document skew estimation. In Elisa H. Barney Smith, Jianying Hu, and James Allan, editors, *DRR*, volume 5296 of *SPIE Proceedings*, pages 182–191. SPIE, 2004.

[32] A. K. Das and B. Chanda. A fast algorithm for skew detection of document images using morphology. *IJDAR*, 4(2) :109–114, 2001.

[33] Somporn Chuai-Aree, Chidchanok Lursinsap, Peraphon Sophatsathit, and Suchada Siripant. Fuzzy c-mean : A statistical feature classification of text and image segmentation method. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 9(6) :661–671, 2001.

[34] J. Li and R. M. Gray. Text and picture segmentation by the distribution analysis of wavelet coefficients. In *Proceedings of International Conference on Image Processing*, volume 3, pages 790–794, 1998.

[35] S. Marinai, B. Miotti, and G. Soda. Digital libraries and document image retrieval techniques : A survey. *Learning Structure and Schemas from Documents*, pages 181–204, 2011.

[36] Simone Marinai. Text retrieval from early printed books. *IJDAR*, 14(2) :117–129, 2011.

[37] Eugen Barbu, Pierre Héroux, Sébastien Adam, and Éric Trupin. Using bags of symbols for automatic indexing of graphical document image databases. In *Graphics Recognition*, pages 195–205, 2005.

[38] Takashi Saitoh, Michiyoshi Tachikawa, and Toshifumi Yamaai. Document image segmentation and text area ordering. In *ICDAR*, pages 323–329, 1993.

[39] Daniel P. Lopresti and George Nagy. Issues in ground-truthing graphic documents. In *GREC*, pages 46–66, 2001.

[40] Oleg Okun and Matti Pietikäinen. Fast and accurate ground truth generation for skew-tolerance evaluation of page segmentation algorithms. *EURASIP J. Adv. Sig. Proc.*, 2006, 2006.

[41] Prima group website. http ://www.cse.salford.ac.uk/prima/.

[42] A. Antonacopoulos, S. Pletschacher, D. Bridson, and C. Papadopoulos. Icdar 2009 page segmentation competition. In *International Conference on Document Analysis and Recognition*, pages 1370–1374, 2009.

[43] Stefan Pletschacher and Apostolos Antonacopoulos. The page (page analysis and ground-truth elements) format framework. In *ICPR*, pages 257–260, 2010.

[44] Chang Ha Lee and Tapas Kanungo. The architecture of trueviz : a groundtruth/metadata editing and visualizing toolkit. *Pattern Recognition*, 36(3) :811–825, 2003.

[45] Jean-Yves Ramel, S. Busson, and M. L. Demonet. Agora : the interactive document image analysis tool of the bvh project. In *DIAL*, pages 145–155, 2006.

[46] David S. Doermann and David Mihalcik. Tools and techniques for video performance evaluation. In *ICPR*, pages 4167–4170, 2000.

[47] J. Sauvola and H. Kauniskangas. MediaTeam document dataset. http ://www.mediateam.oulou.fi/MTDB, 1998.

[48] I. T. Phillips, S. Chen, J. Ha, and R. M. Haralick. English document database design and implementation methodology. *Symposium on Document Analysis and Information Retrievals*, 1993.

[49] Apostolos Antonacopoulos, David Bridson, Christos Papadopoulos, and Stefan Pletschacher. A realistic dataset for performance evaluation of document layout analysis. In *ICDAR*, pages 296–300, 2009.

[50] P. Shivakumara, Trung Quy Phan, and Chew Lim Tan. A laplacian approach to multi-oriented text detection in video. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(2) :412–419, 2011.

[51] Julinda Gllavata, Ralph Ewerth, Teuta Stefi, and Bernd Freisleben. Unsupervised text segmentation using color and wavelet features. In *CIVR*, pages 216–224, 2004.

[52] N. Otsu. A thresholding selection method from gray-level histogram. *IEEE Transactions Systems Man Cybernet*, 9(1) :62–66, 2000.

[53] E. K. Wong and M. Chen. A new robust algorithm for video text extraction. *Pattern Recognition*, 36(6) :1397–1406, 2003.

[54] Rainer Lienhart and Axel Wernicke. Localizing and segmenting text in images and videos. *IEEE Trans. Circuits Syst. Video Techn.*, 12(4) :256–268, 2002.

[55] Minhua Li, Meng Bai, Chunheng Wang, and Baihua Xiao. Conditional random field for text segmentation from images with complex background. *Pattern Recognition Letters*, 31(14) :2295–2308, 2010.

[56] Eri Haneda and Charles A. Bouman. Text segmentation for mrc document compression. *IEEE Transactions on Image Processing*, 20(6) :1611–1626, 2011.

[57] Boris Epshtein, Eyal Ofek, and Yonatan Wexler. Detecting text in natural scenes with stroke width transform. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2963–2970, 2010.

[58] Huizhong Chen, Sam S. Tsai, Georg Schroth, David M. Chen, Radek Grzeszczuk, and Bernd Girod. Robust text detection in natural images with edge-enhanced maximally stable extremal regions. In *International Conference on Image Processing*, pages 2609–2612, 2011.

[59] Palaiahnakote Shivakumara, Weihua Huang, Trung Quy Phan, and Chew Lim Tan. Accurate video text detection through classification of low and high contrast images. *Pattern Recognition*, 43(6) :2165–2185, 2010.

[60] Chunmei Liu, Chunheng Wang, and Ruwei Dai. Text detection in images based on unsupervised classification of edge-based features. In *ICDAR*, pages 610–614, 2005.

[61] Victor Wu, R. Manmatha, and Edward M. Riseman. Automatic text detection and recognition. In *Proceedings of Image Understanding Workshop*, pages 707–712, 1997.

[62] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields : Probabilistic models for segmenting and labeling sequence data. In *ICML*, pages 282–289, 2001.

[63] T. Q. Phan, P. Shivakumara, and C. L. Tan. A laplacian method for video text detection. In *ICDAR*, pages 66–70. IEEE Computer Society, 2009.

[64] S. Tabbone, L. Wendling, and J.-P. Salmon. A new shape descriptor defined on the radon transform. *Computer Vision and Image Understanding*, 102(1) :42–51, 2006.

[65] E. J. Candes. Ridgelets : theory and applications. PhD thesis. Technical report, Department of Statistics, Stanford University, 2001.

[66] M. N. Do and M. Vetterli. Orthonormal Finite Ridgelet Transform for Image Compression. *ICIP*, pages 367–370, 2000.

[67] C.-H. Chou, S.-Y. Chu, and F. Chang. Estimation of skew angles for scanned documents based on piecewise covering by parallelograms. *Pattern Recognition*, 40(2) :443–455, 2007.

[68] W. Postl. Detection of linear oblique structures and skew scan in digitized documents. In *ICPR*, pages 687–689, 1986.

[69] Y. K. Chen and J. F. Wang. Skew detection and reconstruction based on maximization of variance of transition-counts. *Pattern Recognition*, 33(2) :195–208, February 2000.

[70] A. Chaudhuri and S. Chaudhuri. Robust detection of skew in document images. *IEEE Transactions on Image Processing*, 6(2) :344–349, 1997.

[71] Jiri Matas, Ondrej Chum, Martin Urban, and Tomás Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *Proceedings of the British Machine Vision Conference*, pages 1–10, 2002.

[72] Louisa Lam, Seong-Whan Lee, and Ching Y. Suen. Thinning methodologies - a comprehensive survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(9) :869–885, 1992.

[73] A. Criminisi, P. Pérez, and K. Toyama. Region filling and object removal by exemplar-based image inpainting. *IEEE Transactions on Image Processing*, 13(9) :1200–1212, 2004.

[74] E.A. Pnevmatikakis and P. Maragos. An inpainting system for automatic image structure - texture restoration with text removal. In *IEEE International Conference on Image Processing*, pages 2616–2619, 2008.

[75] T. F. Chan and L. A. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10(2) :266–277, 2001.

[76] M. Benjelil, R. Mullot and A. M. Alimi. Page Segmentation Based on Steerable Pyramid Features. *International Conference on Frontiers in Handwriting Recognition*, pages 262–267, 2012.

[77] Y. Cheng. Mean shift, mode seeking, and clustering. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17(8) :790–799, 1995.

[78] Q. Ye, Q. Huang, W. Gao, and D. Zhao. Fast and robust text detection in images and video frames. *Image Vision Comput.*, 23(6) :565–576, 2005.

[79] A. Antonacopoulos and D. Bridson. Performance analysis framework for layout analysis methods. In *ICDAR 2009 Page Segmentation Competition*, pages 1258–1262, 2007.

[80] Pietro Perona and Jitendra Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(7) :629–639, 1990.

[81] Per-Erik Forssén and David G. Lowe. Shape descriptors for maximally stable extremal regions. In *International Conference on Computer Vision*, pages 1–8, 2007.

[82] Andrea Vedaldi and Brian Fulkerson. Vlfeat : an open and portable library of computer vision algorithms. In *Proceedings of the international conference on Multimedia*, pages 1469–1472, 2010.

[83] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 886–893, 2005.

[84] Rodrigo Minetto, Nicolas Thome, Matthieu Cord, Jorge Stolfi, Frédéric Precioso, Jonathan Guyomard, and Neucimar J. Leite. Text detection and recognition in urban scenes. In *IEEE International Conference on Computer Vision Workshops*, pages 227–234, 2011.

[85] Jing Zhang and Rangachar Kasturi. Text detection using edge gradient and graph spectrum. In *International Conference on Pattern Recognition*, pages 3979–3982, 2010.

[86] Jeong-Hun Jang and Ki-Sang Hong. Detection of curvilinear structures and reconstruction of their regions in gray-scale images. *Pattern Recognition*, 35(4) :807–824, 2002.

[87] F. C. Leone, L. S. Nelson, and R. B. Nottingham. The folded normal distribution. *Technometrics*, 3(4) :543–550, 1961.

[88] Vladimir Y. Mariano and Rangachar Kasturi. Locating uniform-colored text in video frames. In *Int. Conf. Pattern Recognition*, pages 4539–4542, 2000.

[89] Simon M. Lucas, Alex Panaretos, Luis Sosa, Anthony Tang, Shirley Wong, and Robert Young. Icdar 2003 robust reading competitions. In *International Conference on Document Analysis and Recognition*, pages 682–687, 2003.

[90] Asif Shahab, Faisal Shafait, and Andreas Dengel. Icdar 2011 robust reading competition challenge 2 : Reading text in scene images. In *International Conference on Document Analysis and Recognition*, pages 1491–1496, 2011.

[91] Rodrigo Minetto, Nicolas Thome, Matthieu Cord, Jonathan Fabrizio, and Beatriz Marcotegui. Snoopertext : A multiresolution system for text detection in complex visual scenes. In *International Conference on Image Processing*, pages 3861–3864, 2010.

[92] J. Fabrizio, B. Marcotegui, and M. Cord. Text Detection in Street Level Images. In *Pattern Analysis and Applications*, volume 16, pages 519–533, 2013

Abstract

In this thesis I discuss the document image segmentation problem and I describe our new approaches for detecting and classifying document contents.

First, I discuss our skew angle estimation approach. The aim of this approach is to develop an automatic approach able to estimate, with precision, the skew angle of text in document images. Our method is based on Maximum Gradient Difference (MGD) and R-signature. Then, I describe our second method based on Ridgelet transform.

Our second contribution consists in a new hybrid page segmentation approach. I first describe our stroke-based descriptor that allows detecting text and line candidates using the skeleton of the binarized document image. Then, an active contour model is applied to segment the rest of the image into photo and background regions. Finally, text candidates are clustered using mean-shift analysis technique according to their corresponding sizes. The method is applied for segmenting scanned document images (newspapers and magazines) that contain text, lines and photo regions.

Finally, I describe our stroke-based text extraction method. Our approach begins by extracting connected components and selecting text character candidates over the CIE LCH color space using the Histogram of Oriented Gradients (HOG) correlation coefficients in order to detect low contrasted regions. The text region candidates are clustered using two different approaches ; a depth first search approach over a graph, and a stable text line criterion. Finally, the resulted regions are refined by classifying the text line candidates into "text" and "non-text" regions using a Kernel Support Vector Machine K-SVM classifier.

Keywords: document image segmentation, text extraction, stroke descriptors, classification, clustering, stable text lines.

Résumé

Dans cette thèse, nous abordons le problème de la segmentation des images de documents en proposant de nouvelles approches pour la détection et la classification de leurs contenus.
Dans un premier lieu, nous étudions le problème de l'estimation d'inclinaison des documents numérisées. Le but de ce travail étant de développer une approche automatique en mesure d'estimer l'angle d'inclinaison du texte dans les images de document. Notre méthode est basée sur la méthode Maximum Gradient Difference (MGD), la R-signature et la transformée de Ridgelets. Nous proposons ensuite une approche hybride pour la segmentation des documents. Nous décrivons notre descripteur de trait qui permet de détecter les composantes de texte en se basant sur la squeletisation. La méthode est appliquée pour la segmentation des images de documents numérisés (journaux et magazines) qui contiennent du texte, des lignes et des régions de photos. Le dernier volet de la thèse est consacré à la détection du texte dans les photos et posters. Pour cela, nous proposons un ensemble de descripteurs de texte basés sur les caractéristiques du trait. Notre approche commence par l'extraction et la sélection des candidats de caractères de texte. Deux méthodes ont été établies pour regrouper les caractères d'une mÃªme ligne de texte (mot ou phrase) ; l'une consiste à parcourir en profondeur un graphe, l'autre consiste à établir un critère de stabilité d'une région de texte. Enfin, les résultats sont affinés en classant les candidats de texte en régions "texte" et "non-texte" en utilisant une version à noyau du classifieur Support Vector Machine (K-SVM).

Mots-clés: document de segmentation d'image, extraction de texte, descripteurs de texte, classification, clustering.