On the stability of document analysis algorithms : application to hybrid document hashing technologies

Sébastien Eskenazi

UNIVERSITÉ DE LA ROCHELLE

ÉCOLE DOCTORALE S2IM

LABORATOIRE : L3I

THÈSE présentée par :

Sébastien ESKENAZI

soutenue le : **14 décembre 2016**
pour obtenir le grade de : **Docteur de l'Université de La Rochelle**
Discipline : **Informatique et Applications**

On the stability of document analysis algorithms.

Application to hybrid document hashing technologies.

RAPPORTEURS	**Apostolos ANTONACOPOULOS**	Professor, PRImA, University of Salford (United Kingdom)
	Josep LLADÓS	Professor, CVC, Universitat Autònoma de Barcelona (Spain)
EXAMINATEURS	**Jean-Philippe DOMENGER**	Professeur, LaBRI, Université de Bordeaux
	Nicole VINCENT	Professeur, LIPADE, Université Paris-Descartes
	Utpal GARAIN	Professor, CVPR, Indian Statistical Institute Kolkatta (India)
DIRECTEUR	**Jean-Marc OGIER**	Professeur, L3i, Université de La Rochelle
ENCADRANTE	**Petra GOMEZ-KRÄMER**	Maître de conférences, L3i, Université de La Rochelle

Thèse réalisée au Laboratoire L3i
Faculté des Sciences et Technologies
Avenue Michel Crépeau
17042 La Rochelle cedex 01

Tél : +33 5 46 45 82 62
Fax : +33 5 46 45 82 42

Web : l3i.univ-lr.fr

Sous la direction de Jean-Marc Ogier jean-marc.ogier@univ-lr.fr

Co-encadrement Petra Gomez-Krämer petra.gomez@univ-lr.fr
Financement Allocation de recherche de la Communauté d'agglomérations de La Rochelle

Résumé

Un nombre incalculable de documents est imprimé, numérisé, faxé, photographié tous les jours. Les documents d'aujourd'hui sont hybrides : ils existent sous forme papier et numérique. De plus les documents numériques sont ubiquitaires. Ils peuvent être consultés et modifiés simultanément dans de nombreux endroits. Avec la grande disponibilité des logiciels d'édition d'image, il est devenu très facile de modifier ou de falsifier ces documents. Cela crée un besoin croissant pour un système d'authentification capable de traiter ces documents hybrides.

Les solutions actuelles reposent sur des processus d'authentification duals, un pour le papier et l'autre pour les documents numériques. Ils sont complexes et coûteux. D'autres solutions reposent sur une vérification visuelle et offrent seulement une sécurité partielle. Dans d'autres cas elles nécessitent que les documents sensibles soient stockés à l'extérieur des locaux de l'entreprise et un accès au réseau au moment de la vérification.

Afin de surmonter tous ces problèmes, nous proposons de créer un algorithme de hachage sémantique pour les images de documents. Cet algorithme de hachage devrait fournir une signature compacte pour toutes les informations visuellement significatives contenues dans le document. Ce condensé permettra la création de systèmes de sécurité hybrides pour sécuriser tout le document. Ceci peut être réalisé grâce à des algorithmes d'analyse du document. Cependant ceux-ci ont besoin d'être porté à un niveau de performance sans précédent, en particulier leur fiabilité qui dépend de leur stabilité.

Nos contributions sont : la création d'un descripteur de mise en page et d'un algorithme de hachage perceptuel d'image qui satisfont nos contraintes. Nous avons également tenté de faire un algorithme de segmentation d'image de document stable et un algorithme de reconnaissance optique de caractères (OCR) stable. À cet égard, nous avons amélioré l'état de l'art avec un algorithme de pré-segmentation capable d'identifier des zones de couleur visuellement uniformes. Cet algorithme étend le concept de composantes connexes aux images en niveau de gris et en couleur. En outre, il peut être utilisé pour détecter les contours et effectuer une détection précise et continue de l'échelle des objets contenus dans l'image. Nous avons également amélioré la stabilité des algorithmes d'OCR, mais

elle reste insuffisante. En outre, les algorithmes présentés dans cette thèse sont sans paramètres.

Enfin, nous avons produit cinq ensembles de données totalisant plus de quatre-vingt-neuf mille images pour évaluer nos algorithmes et nous avons introduit de nouvelles mesures pour évaluer la stabilité de plusieurs types d'algorithmes d'analyse d'image de document. Ce travail est basé sur plusieurs examens de l'état de l'art, en particulier un examen approfondi des algorithmes de segmentation d'image de document.

Mots clés : stabilité, analyse d'images de document, sécurité, impression et scan, segmentation, hachage perceptuel d'image, superpixels, composantes connexes en couleurs, descripteur de mise en page, OCR.

On the stability of document analysis algorithms Application to hybrid document hashing technologies

Abstract

An innumerable number of documents is being printed, scanned, faxed, photographed every day. Today's documents are hybrid: they exist as both hard copies and digital copies. Moreover their digital copies are ubiquitous. They can be viewed and modified simultaneously in many places. With the wide availability of image modification software, it has become very easy to modify or forge such documents. This creates a rising need for an authentication scheme capable of handling these hybrid documents.

Current solutions rely on dual authentication schemes, one for paper and one for digital documents, that are complex and costly. Other solutions rely on manual visual verification and offer only partial security or require that sensitive documents be stored outside the company's premises and a network access at the verification time.

In order to overcome all these issues we propose to create a semantic hashing algorithm for document images. This hashing algorithm should provide a compact digest for all the visually significant information contained in the document. This digest will allow current hybrid security systems to secure all the meaningful content of the document. This can be achieved thanks to document analysis algorithms. However those need to be brought to an unprecedented level of performance, in particular in terms of their reliability which depends on their stability.

Thus, our contributions are: the creation of a layout descriptor and of a perceptual image hashing algorithm that both satisfy our requirements. We also attempted to make a stable document image segmentation algorithm and a stable optical character recognition (OCR) algorithm. In that regard we improved the state of the art with a presegmentation algorithm capable of identifying patches of visually uniform color. This algorithm extends the concept of connected components to gray level and color images and can be used to detect edges and to perform a continuous and precise scale identification. We also improved the stability of OCR algorithms but it remains insufficient. Furthermore, the algorithms presented in this thesis are parameter free.

At last we produced five datasets totaling more than eighty nine thousand images to evaluate our algorithms and we introduced new performance indicators to

evaluate the stability of several types of document image analysis algorithms. This work is based on several reviews of the state of the art, in particular a thorough review of document image segmentation algorithms.

Keywords: stability, document image analysis, security, print and scan, segmentation, perceptual image hashing, superpixels, color connected components, layout descriptor, OCR.

Remerciements

This thesis was a real opportunity in my career and I sincerely and enormously thank my supervisors Jean-Marc Ogier and Petra Gomez-Krämer for giving me the chance of doing it. Jean-Marc is an extraordinary person with an incredible insight and a lot of human qualities that I appreciate every day. Petra's comments and reviews were essential and very useful. She has relentlessly pushed me to improve my work which lead to the quality of this thesis. I also thank them both for giving me a lot of freedom in the way I conducted this work.

I would like to thank the members of the jury for accepting to participate in this defense, in particular Professors Antonacopoulos and Lladós for reviewing this thesis. Their comments have been most helpful.

I thank my friends and colleagues at L3i and in all the other places I visited during this thesis. They are always a good source of advice and contributed to the very pleasant work atmosphere that I have known there.

I also take this chance to thank Boris Bodin whose work was very helpful to complete the chapter on perceptual image hashing.

Work is not everything and I thank the people at the association of the PhD students of La Rochelle (ADocs) as well as those I met at the sports club of the university (SUAPSE, sailing and dancing in particular). The moments spent with you have been a lot of refreshing fun, especially sailing in the rain.

My family has been very supportive, in particular my mother Stéphanie and her husband Michel. I thank them all for this. It is nice to have people encouraging you and supporting you in your projects.

Finally, I thank my beloved Phương. She has been enlightening these years with happiness, supporting me and helping me with her experience. I cannot thank her enough for being here and luckily I will be able to thank her everyday for many more years.

Contents

List of Figures

List of Tables

Introduction

More and more paper documents are scanned to be transmitted electronically. These documents are processed automatically with a class of algorithms called document image analysis algorithms. As more and more industries digitize their document work-flows, the need for such algorithms rises and new use cases arise. A particular use case is that of extracting the content of the digitized documents for security applications.

For instance, an easy way to obtain a fraudulent identity card is not to forge one but to obtain a real one with fraudulent documents such as a fake electricity bill and a fake birth certificate [Smi02]. Some of these documents are in paper format and some are in digital format. Ensuring the security of these two kinds of documents and of documents that can change format is called hybrid security. So far, there is no other choice but to use two authentication systems: one for the paper documents and one for the digital documents. As shown on Figure 0.0.1, paper documents are usually secured by means of a watermark or a fingerprint which is a watermark issued only once. Digital documents are authenticated against their cryptographic hash (detailed in Chapter 1). We propose to use a semantic hashing to secure hybrid documents.

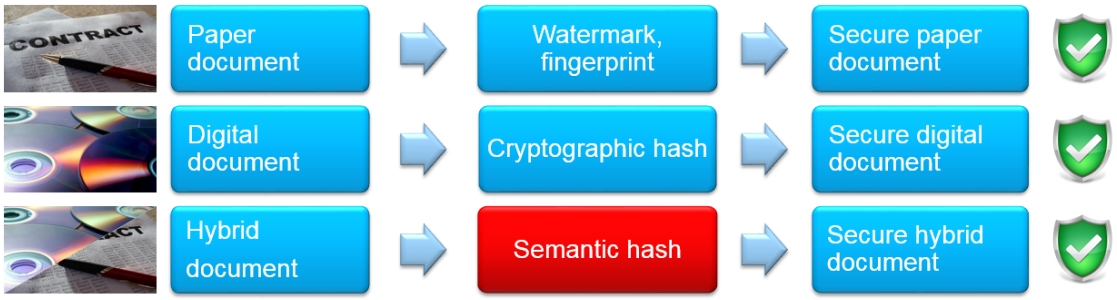


Figure 0.0.1: Authentication technologies for paper, digital and hybrid documents.

Semantic hashing will use document analysis algorithms to make a unique system capable of securing a document no matter its support. Document analysis algorithm will serve to extract the content of the document images, hence the

name "semantic" hashing.

This semantic hashing operation requires the sequencing of many document analysis algorithms, each of which is generally very noise sensitive. This sensitivity generally generates a lot of instabilities in the final results of the processing chains. Hence the key issue is that of the stability of the algorithms involved in the content extraction and hashing process.

Thus we have two main issues. From an industrial point of view, we would like to use document image analysis algorithms to secure the content of a document. From a scientific point of view, we are interested in evaluating and producing stable algorithms.

Context and motivations

With the increase of the industrial document digitization, the rising need for document archiving with destruction of the original copy and the arrival of security technologies based on document processing, document processing algorithms need to be more and more reliable. The key factor in the reliability of these algorithms is their repeatability, e.g. their ability to produce similar or even identical results on several copies of the same document. This is also called "stability" as the results of the algorithms have to be stable with respect to the noise that can occur between two copies of the same document.

Baird and Casey [BC06] already identified this as a key factor of performance for an algorithm when they mention that we should aim at "confidence before accuracy". They mean that when an algorithm gives a result we should be confident that it will produce a similar result with a (not necessarily exactly) similar input. They also mean that this confidence is more important than producing accurate results which we are not sure to be able to reproduce.

The L3i laboratory has several industrial partners involved in the document digitization market. They pointed out one serious issue with digitizing: being able to trust the documents that come in the digitization process with minimal human intervention. By instance, when a bank receives a payslip from a customer who applies for a loan, it wants to be sure that it is authentic. This is where document analysis algorithms come into place. They can extract all the information from the document image thus allowing the use of standard security frameworks. However this will only work if we can guarantee the quality of the extracted information. For instance, companies want to be sure to extract the name and gross salary even though the rest of the extracted content may be completely wrong. This illustrates the "confidence before accuracy" paradigm and more generally, the importance of having stable algorithms.

Objectives and challenges

The objectives of this thesis are two-fold. On the applicative side, we will attempt to create algorithms that can be used for document authentication. On the scientific side we will attempt to create stable algorithms.

In particular we are interested in the following algorithms:

- Layout descriptors: these algorithms describe the relative positions of the elements or regions of a document.
- Segmentation algorithms: these algorithms identify the regions of a document image.
- Optical Character Recognition (OCR) algorithms: these algorithms extract the text from a document image.
- Perceptual image hashing algorithms: these algorithms are very useful to authenticate images and detect possible modifications between two images.

For these algorithms to work in an authentication framework they need to satisfy two constraints:

- Have a high level of detection of non-authentic documents.
- Make little or no false detection of authentic documents.

These two constraints are antagonistic. Hence a given algorithm will either be too "robust" and not detect all the non-authentic documents or it will be too sensitive and falsely detect authentic documents as non authentic ones. In the general case, there is a trade-off between these two situations. Making stable algorithms will allow us to improve this trade-off by keeping the general sensitivity of the algorithms while not falsely detecting copies of the same document as non-authentic.

For these algorithms to work in an industrial context they must satisfy another requirement than plain security performance. They need to have an operational performance measured by the processing time and the size of the digest or descriptions that they provide. Since we work in a security context, we can combine the algorithms with cryptographic techniques to reduce the size of the digests as well as protect the confidentiality of the content being secured. Making stable algorithms that produce the same output on all the copies of a document will make it easier to use these cryptographic techniques.

We see here that the stability of document image analysis algorithms is at the heart of the problem. Unfortunately, it has barely been studied in the literature and there is no generic definition for it. Hence the first challenge will be to define

what is a stable algorithm. Then we will need to define an evaluation framework and adapt it to each kind of algorithm that we study.

The next challenge is that, because stability has not been studied, we do not know the performance of existing algorithms with this criteria. Hence our first task will be to evaluate the stability of existing algorithms and then to improve them or produce new, more stable algorithms.

At last, we focus on stability with respect to print and scan noise. While there are many synthetic noise models, it is actually quite rare to see studies or datasets with real print and scan noise. To our knowledge no study of the representativeness of the synthetic models is available either. Hence we will need to produce the necessary material to ensure that the algorithms perform properly on real print and scan noise.

Contributions

As David Doermann once said during the International Document Image Processing Summer school [KSS+14], a new problem is often more interesting than a new solution. Considering this, our first contribution is to bring into the spotlight the issue of the stability of document image analysis algorithms.

We can summarize the main contributions of this thesis as follows:

- A proper formalization of the study of the stability of an algorithm,
- A new parameter free color connected component (CCC) segmentation algorithm which extends the definition of connected components to gray level and color images and is three to five times more stable than other approaches,
- A new parameter free layout descriptor which is particularly stable,
- A new parameter free image hashing algorithm which is both stable and precise,
- A new parameter free OCR post-processing algorithm that drastically improves OCR stability.

Regarding the parameter free algorithms, the computation of the description that they produce is parameter free but some of them require a matching algorithm that requires parameters to allow the user to choose the performance trade-off that he wants.

In order to achieve these results we produced several other contributions:

- A new generic document typology,

- A new digital model of the human eye including a new spatio-colorimetric color distance,
- An in depth-coverage of the issues related to a print and scan process,
- Several reviews of the state of the art of the algorithms used in this thesis,
- An evaluation of the stability of several document image analysis algorithms which sets a baseline for further analysis of the stability of similar algorithms,
- Several large datasets containing print and scan noise.

The evaluations and the datasets cover the algorithm scope defined in the objectives of the thesis.

Thesis plan

The organization of this thesis starts with two introductory chapters for the context and the definition of stability. The next five chapters follow the chain of algorithms to produce a semantic hash.

Chapter 1: The first chapter of this thesis describes the applicative context and the proposed semantic hash framework. In this chapter we will define the document typology. We will review the noise introduced by a print and scan process as well as current document security systems that attempt to deal with it.

Chapter 2: The second chapter presents a formal definition of the stability of an algorithm and a framework to evaluate it. This is our main performance criteria for the algorithms that we study.

Chapter 3: The third chapter focuses on the description of the layout of a document. It presents both an evaluation tool and and algorithm. The descriptor it proposes is based on a set of points (the centroids of the regions of a document). This is useful both to describe the content of a document (the layout is part of it) and to compare two segmentation results. We review the state of the art and present a new algorithm: the Delaunay Layout Descriptor (DLD) which outperforms the state of the art. We also present a dataset of layouts similar to those produced by an ideal segmentation algorithm on several copies of the same documents.

Chapter 4: The fourth chapter deals with document image segmentation algorithms. They are necessary to obtain the document layout and produce the document regions which can be processed by the other algorithms. After a comprehensive review of the state of the art we benchmark four state of the art algorithms and show that they are completely unstable. This chapter includes a new dataset of photocopies of color documents.

Chapter 5: Considering the instability of current segmentation algorithms, the fifth chapter proposes to extend the definition of connected components to gray level and color images. We call these: color connected components (CCC). This may be useful to extend to color images the segmentation algorithms that make use of connected components. It will give them more information which may help make them more stable. We show the existing issues of superpixel approaches and propose a set of new algorithms to solve them. They are based on a new model of human vision which, to our knowledge, is more detailed than existing ones. These new algorithms outperform the state of the art by a vast margin in terms of stability.

Chapter 6: The sixth chapter deals with one of the next document image processing steps after the segmentation: optical character recognition (OCR). After a survey of the state of the art, we propose a simple alphabet reduction to improve the stability of OCR algorithms. This improvement is demonstrated on two state of the art algorithms. This chapter includes a new dataset of photocopies of text only documents with several typographic variations.

Chapter 7: The seventh and last chapter is focused on another document image processing step: perceptual image hashing. This is necessary to describe the content of the graphical parts of a document such as logos and handwritten signatures. After a survey of the state of the art we study the usefulness of approaches based on key-points. These do not have a sufficient performance but we demonstrate that the positional information of key-points is more useful than the information contained in their associated descriptors. Finally we propose a new perceptual image hashing algorithm that outperforms the state of art. This chapter also presents two new datasets of photocopies of handwritten signatures (including several trials by the same person and fraudulent copies) and of photocopies of logos.

Conclusion: Finally the conclusion will summarize the contributions of this thesis and discuss future improvements and the new problems that arose during this work.

Chapter 1

Context of the study

This chapter presents the background information related to this thesis. Notably, we define the types of documents that we work on. We study the different sources of noise in a print and scan process. Because of the complex nature of this process, existing models are either partial or very simplified. We explain the basics of digital hashing and electronic signatures. Then we outline the different available hybrid security technologies and show their current limitations. At last we present the proposed hybrid security framework in which the work of this thesis is set.

In the introduction we highlighted the fact that our work is set in the context of securing hybrid documents. These are documents that can exist on multiple supports. Here we focus on paper and scanned documents produced by printing and scanning. We also need to know how the authentication algorithms work and what are their requirements. Hence, before going through the in-depth study of the different document image processing algorithms, let us first define the three main context elements in which this thesis is set. Thus this chapter is organized as follows:

- Section 1.1 presents the types of document images that we focus on.
- Section 1.2 presents the noise introduced by the print and scan process. This is the main kind of noise that we will deal with.
- Section 1.3 presents the state of the art of hybrid document security technologies. In this section we propose a hybrid document image hashing framework to solve the challenges of the state of the art. This thesis presents several algorithms that could be used in this framework.

The contributions of this chapter other than the state of the art surveys are:

- A proper document typology in Section 1.1,
- A thorough list of print and scan sources of noise in Section 1.2.1,
- A new hybrid document image hashing framework in Section 1.3.4.

We will now present the typology of the documents that we will study.

1.1 Typology of documents

Since the main processing step of document image analysis relies on extracting or indexing the text that they contain, we have chosen to sort document images according to the amount of textual content they have. Figure 1.1.1 summarizes the main types of images from the most textual on the left to the least textual on the right. It is difficult to make a generic and clear classification so the boundaries between the different categories should be considered as fuzzy. For instance, some comics do not contain any text and some magazine pages are only textual.

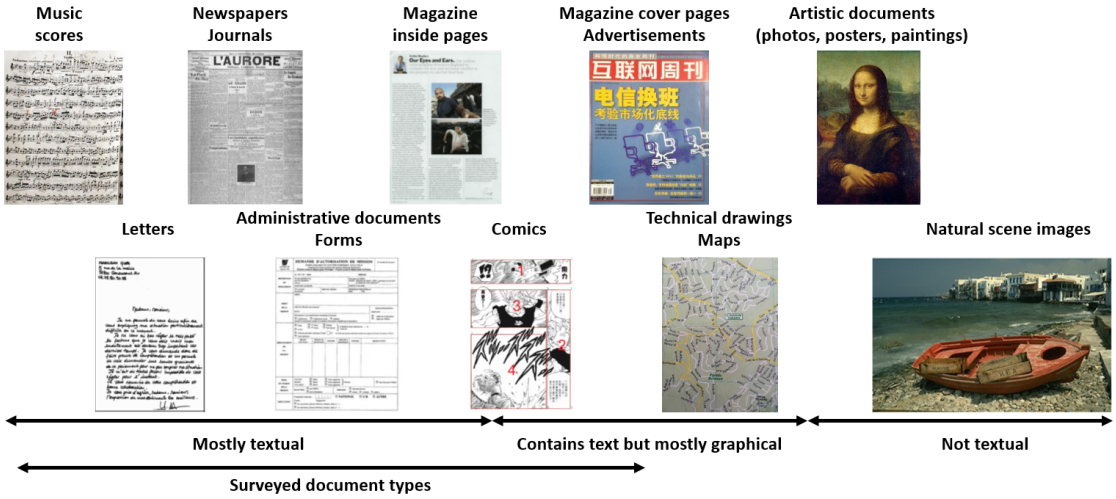


Figure 1.1.1: Document typology by decreasing amount of text from left to right.

Music scores are classified as one of the most textual documents because we can consider music writing as a language and thus as textual content. This is reflected by the fact that most related publications deal with music symbol segmentation, staff removal and direct recognition of music scores without any layout analysis [RFP+12].

There is no specific category for historical documents. This is because all the categories on the figure contain both contemporary and historical documents. Moreover, defining a specific category for historical documents would require defining a boundary between historical and contemporary documents which would most likely not be feasible. The scientific community separates historical from contemporary documents mostly because of the degradations and the variability they have and the specific algorithms they need to deal with them. Hence this separation should not be related to the time but to the degradations of the document. Furthermore, it is not impossible that algorithms made for historical documents perform well on contemporary documents. Actually, many algorithms [DPB08, JRME08, LCC08b, LGPH09, LLG+11, FT12, Kon12, DKS13, LLS14a] have been evaluated on both modern and historical documents.

Similarly to historical documents, handwriting can appear in all categories and as such does not have a category of its own. Moreover, documents that were written before the invention of printing were obviously handwritten but the writing style can be closer to machine print in some cases.

Color-wise, there are three main types of color depth: black and white (BW), gray level (GL) and color (C).

The document analysis community frequently adds another separation documents between off-line (static images) and on-line (with writing strokes/history) documents.

Another way of classifying document images is to focus on how structured they are. However, because we will have to deal with documents having varying degrees of structure, this classification does not seem relevant for our purpose.

We focus on all off-line mostly textual documents and magazine cover pages whose main degradation is made by a print and scan process. Comics and advertisements are less prone to needing security features as very little gain can be made from fraudulent comics and adding security features to advertisements would alter their design. Thus, they are not present in our datasets. We do not deal with technical drawings or maps because they usually require very specific tools to be analyzed. Now that we have defined the documents on which we focus, let us study the main challenge when dealing with these documents: the noise introduced in the print and scan process.

1.2 Noise introduced by a print and scan process

Printing and scanning is a very common process. It has been mostly studied in two fields of research: image watermarking and perceptual image hashing. Several models have been proposed since the 90s for the noise it introduces. One of the earliest works for binary images is that of Kanungo et al. [KHP93]. The most

thorough overview of them is Chapter 5 of Smoaca's thesis [Smo12]. In Chapter 6 she proposes a new model for print and scan noise. A detailed physical model of the print and scan process can be found in [YNS05]. A theoretical model of the print and scan process is proposed in [MF06] with some approximations. However, it should be noted that the study of print and scan noise is frequently taken from a different perspective than that of the signal analysis community. Most models come from a idealized, and frequently incomplete, model of the print and scan hardware and chemicals and they are usually not validated experimentally other than by producing visually not abnormal results. The only results having a strong experimental validation are in the field of CCD models and empirical noise models for binary images.

There are two approaches to describe the noise introduced by a print and scan process. One consists in describing the noise produced by each potential source of noise. The other one consists in empirically modeling the noise present in the images.

1.2.1 Study of the sources of noise in a print and scan process

There are several sources of noise during the print and scan process:

- Editing noise: related to the printing software used by the user.
- Image conversion for printing: related to printer conversion of RGB (red, green, blue) or gray level images into CMYK (cyan, magenta, yellow and black) halftoned images,
- Physical printing noise: related to the electro-mechanical mechanisms of the printer introducing undesired motion, the ink properties and the wear of the system,
- Paper noise: related to paper properties (ink absorption, reflectance, bending) and paper wear (stains, scratches, torn pieces...),
- Sensor scanning noise: related to the sensor electrical noise, its spectral sensitivity, the optical imperfections, the resolution adequacy, the scanner illuminant and the wear of the sensor and the illuminant light source,
- Mechanical scanning noise: related to the electro-mechanical mechanisms of the scanner, the position of the document on the scanner and the wear of the system,
- Scanner post processing: related to specific algorithms embedded in some scanners and that process images in order to "enhance" the raw images produced by the scanner.

We will now study each of them.

Editing noise

When printing a document a user may be offered the possibility of selecting which part of a document to print. He may use this to remove margin, foot notes, etc. Then if the final document or image format does not fit the paper format, he has to choose how to make them correspond. Figure 1.2.1 shows the Microsoft Windows interface which proposes this functionality.

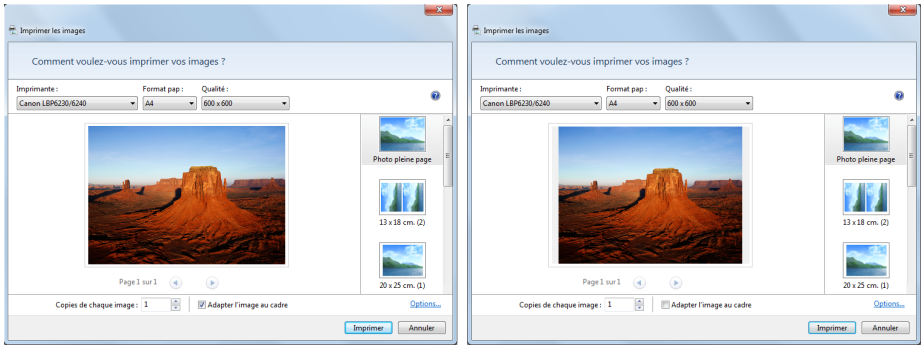


Figure 1.2.1: Two ways of adjusting an image format to a paper format.

The result of this processing is cropping and non isotropic scale variations. The impact of cropping on the discrete Fourier transform has been studied in [LC99] which also deals with generic scale variations.

Image conversion for printing

Electronic displays emit light to display colors and images. Thus they use additive color combinations: the more colors we add, the brighter the image. On the contrary printers use inks which - like paint - use subtractive color combinations. The primary colors for each type of color and their combinations are shown on Figure 1.2.2.

Thus the RGB (red green blue) colors of the initial document image need to be converted to CMYK (cyan, magenta, yellow and black) colors for the printer. This conversion introduces a colorimetric noise because CMYK colors are not the same as RGB colors. Such noise is shown on Figure 1.2.3.

The next issue is related to the fact that the four colors of the printer have a specific fixed intensity/brightness. In order to produce several levels of intensity they are printed as micro-dots of varying number and spacing within the area corresponding to a given pixel. This is called halftoning. Figure 1.2.4 shows examples of halftoning for three given colors. This process works because the

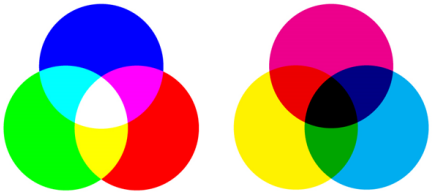


Figure 1.2.2: Two color combination systems. Left: additive color combination with RGB colors. Right: subtractive color combination with CMYK colors.

Figure 1.2.3: Colorimetric noise introduced by the conversion from RGB to CMYK colors.[1]

human eye spatial resolution is lower than the printing resolution and thus it cannot distinguish the printing dots. However scanners may be sensitive to this quantization noise.

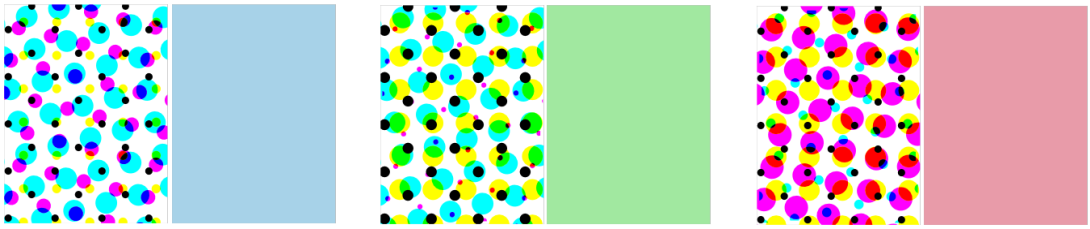


Figure 1.2.4: Halftoned images and the corresponding perceived color.

[Smo12] highlights the fact that depending on the number of possible micro-dots per pixel the number of shades of color can be limited. This phenomenon is not perceptible as long as the halftoning noise is not perceptible either which is the case in our scenario.

[1]Image courtesy of `www.tvoru.com.ua`

Only the halftoning has been studied in [LC99, BMI07, PN95]. They present the mathematical functions related to halftoning. Halftoning noise only appears if the scanning resolution is higher than the printing resolution which is not the case in most document digitizing processes. These processes deal with millions of documents and have storage constraints such as a limited scanning resolution in order to save memory space and improve scanning speed. Thus we will not detail halftone noise here. For illustration purposes Figure 1.2.5 shows an example of halftoning noise.

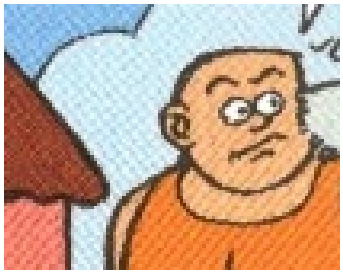


Figure 1.2.5: Halftoning noise when a document is scanned at a too high resolution.[2]

Physical printing noise

All the electromechanical parts of the printer can introduce noise. In a laser printer, the laser, the laser actuators, the laser lens, the drum and its actuators all have defects. These result in blur, local image warping and/or shearing. The paper feeding system may also put marks on the paper or may not have a constant speed which could introduce warping. The ink can also add some noise. In particular the size of dots increases when the ink dries. The ink noise has been studied in [BMI07, LC99, MF06, Smo12, YNS05, PN95] and is commonly included in what is called the point spread function of the printer which is assimilated to a Gaussian blur. This point spread function basically reflects all the blur produced by the complete print and scan process.

The impact of the other noises and the specific impact of the ink on the printed image have not been shown. The presence of dust in the printer has not been envisioned by existing studies.

Paper noise

The paper on which the image is printed can introduce noise because of its coating and reflectance (glossy paper) or its color. Its roughness influences the way the ink

[2]Image taken from [Deb15]

is absorbed and may bleed and may create texture noise. Its thickness influences the possible local warping/bending. During its life cycle a paper may also be stained, folded, scratched, its edges can be torn, etc. [MF06] implicitly includes the influence of paper reflectance in a generic print and scan model. We have not found models related specifically to degradations occurring during the document life cycle. [LCOB15] and [SCE+15] include it in a more general empirical noise model which will be described in Section 1.2.2.

Sensor scanning noise

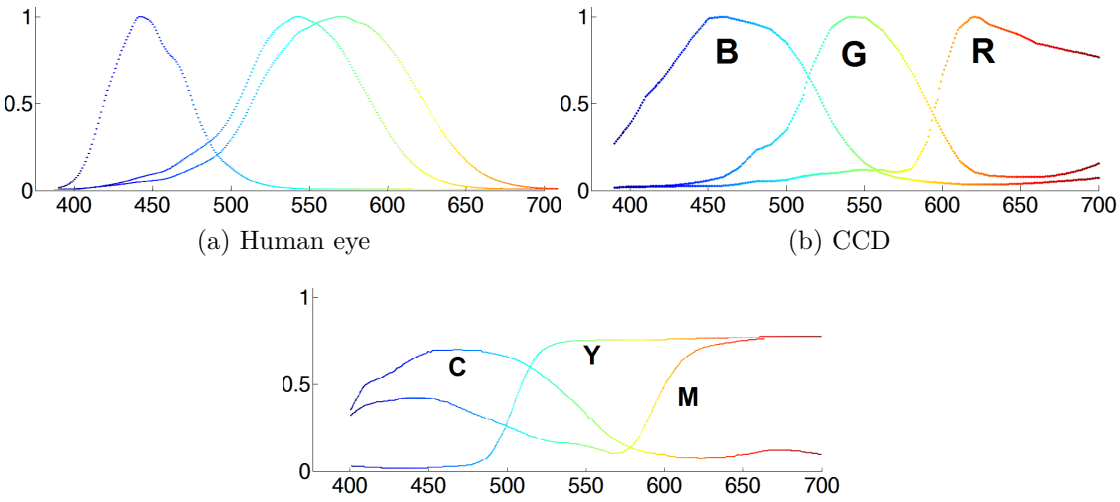
Scanner sensors are arrays of CCD sensors. An inherent noise to imaging devices is called "shot noise". This noise is due to the amount of light that is received by the sensor. Basically because of the sensor size, its illumination time and its sensitivity, fluctuations in the flux of incoming photons may be reflected in the image produced by the sensor. These fluctuations are inevitable because of the particle nature of light. This noise can be easily seen when taking a photograph with a high ISO or when zooming on regions of uniform color in an image. With the paper texture this is the main reason for which regions of uniform colors do not have exactly a uniform color once scanned. This noise is classically modeled by a Poisson law [Smo12]. [BMI07, MF06] include its effect in a more generic polynomial model.

Another noise is related to the inaccuracies of the optical system which creates a blur included in the point spread function of the overall system.

If the scanning resolution is close to or higher than the printing resolution the halftoning pattern becomes visible [LC99, BMI07, PN95] but as we mentioned, this is not the case in our scenario.

There is also a significant range of noise that has not been studied. The printer uses CMYK inks/colors and the sensors use RGB colors. Thus the light spectrum reflected by the paper does not match the light spectrum to which the sensor is sensitive which creates color inaccuracies. The same phenomenon occurs with a computer screen and the human eye. They use different spectra. This is (partially) the reason why when someone takes a picture of a scene and compares the colors of the scene to those of the picture, they are different. Figure 1.2.6 shows the spectral reflectivity of cyan, magenta and yellow inks taken from [Sey13], the sensitivity of a Kodak KAF5101-CE CCD sensor [Kod03] and of the three types of cones/sensors of a human eye [SS00]. As we can see none of the spectra match.

The scanning sensors receive the light that is reflected by the paper. This light does not come from the sun but from a light source called an illuminant. Depending on this light source and its spectrum, the colors that are sensed can also be different from the initial colors. This can be compensated by a functionality called the "white balance" and explains why some scanned images look more blueish than



(a) Human eye (b) CCD

(c) CMY inks

Figure 1.2.6: Spectral sensitivities of a CCD sensor and of the three color sensors of a human eye and spectral reflectivity of CMY inks.

others. Figure 1.2.7 shows this kind of noise.

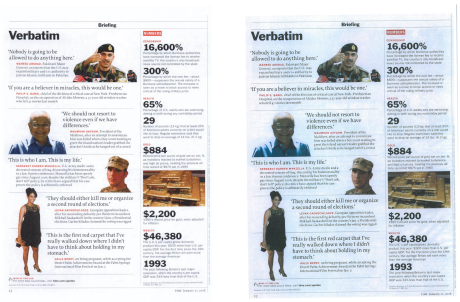


Figure 1.2.7: Two images with different illuminant/while balance.

At last, the sensor and light source do not last for ever and undergo some wearing process. To our knowledge this has not been studied either.

Mechanical scanning noise

Similarly to the mechanical printing noise, the scanner elements are not perfect and introduce a certain amount of noise. In [BMI07, MF06, YNS05] it is modeled by a Gaussian noise. The width and intensity of the blur may be different in the direction of the relative motion between the paper and the scanner and in the perpendicular direction.

The scanner array of sensor can be mounted on a tray which move to scan the image. [Smo12] found that the motion of this tray can introduce some spatial noise between two copies of the same image. This is due to the motion inaccuracies of the tray which does not always move by an exactly constant distance or whose starting position is not always exactly the same. However this noise is only noticeable at very high scanning resolutions (1200 dpi in her experiments).

If the scanner uses a feeder tray, it may introduce similar distortions as the paper feeder tray of the printer. The rotation of the document (also called the skew) may however be more important because the position of the document to scan is less constrained than that of the paper for the printer which is in a tray. Thus the human operator may introduce a significant skew. Our experiments have shown that this skew is usually lower than 15°. Scanners also frequently produce cropped images if part of the document is outside the scanning window or they may add some black or white borders if the document is smaller than the scanning window. These noises can now easily be removed [BCC+15] and many commercial applications embed algorithms that do this such as CamScanner [3], Office Lens [4], Genius Scan [5], Doc Scanner [6] or Evernote [7].

The wear of the scanner or the presence of dust on its glass or the sensor has not been studied.

Scanner post processing noise

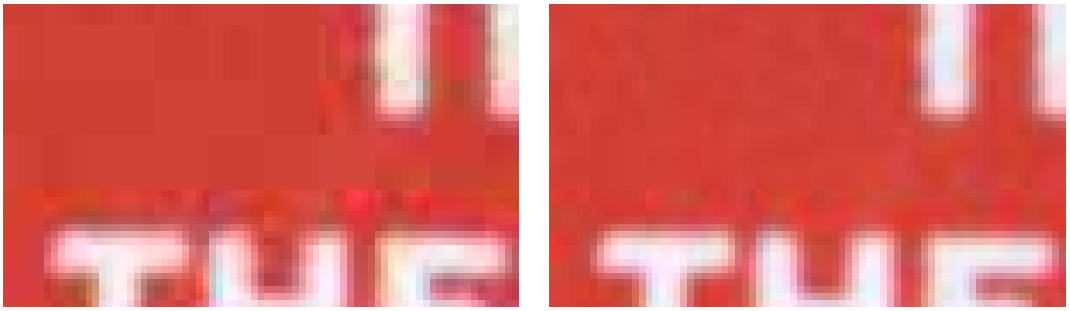
The last step for the production of an image is the scanner post-processing that is embedded in the scanner driver. Most frequently this is a JPEG compression [ITU92]. While describing the whole JPEG compression process would not be of use here, we can point out a few important steps of it. The first step in JPEG compression is the conversion of the classical RGB colors to YCbCr colors. This conversion is linear as shown in Equation (1.2.1). The interest of this color space is that the human eye is less sensitive to Cb and Cr channels. Hence they can be quantized to save memory space without a significant visual loss. This quantization is usually done on a spatial basis e.g. storing only one Cr or Cb channel information for two or four pixels.

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.1687 & -0.3313 & 0.5 \\ 0.5 & -0.4187 & -0.0813 \end{bmatrix} \times \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{1.2.1}$$

[3] www.camscanner.com/
[4] www.microsoft.com/fr-fr/store/p/office-lens/9wzdncrfj3t8
[5] thegrizzlylabs.com/
[6] www.docscannerapp.com/
[7] evernote.com/intl/fr/

This step creates a loss of color information.

Another step divides the images into blocks of 8 by 8 pixels. This does not create any information loss per se but the next compression steps can make these blocks appear and create what is commonly called blocking artifacts. They are particularly visible in textured areas where the smooth color variations are disrupted by the edges of the blocks. Figure 1.2.8 shows two images whose sole difference is the JPEG compression quality factor. This factor allows the user to decide how much information loss he accepts.



(a) Q=25% (b) Q=75%

Figure 1.2.8: Two images with different JPEG quality factors (best seen in color).

1.2.2 Empirical models of print and scan noise

As we can see, there are many sources of noise that are not modeled. Thus it may be more convenient to empirically model the noise that is produced by the print and scan process. This kind of approach is usually application driven in order to avoid producing overly complex models.

In the case of printing a binary pattern such as a barcode the intensity of black and white pixels can be modeled by a log-normal distribution [BC13]:

$$p(i) = \frac{1}{\sqrt{2\pi\sigma i}} \exp\left(-1/2 \frac{(\ln i - \mu)^2}{\sigma^2}\right) \tag{1.2.2}$$

where μ is the mean intensity, σ is the intensity variance, i is the intensity and $p(i)$ is density probability at intensity i. This work has been corroborated in [TPSD16]. They also find out that print and scan noise is not additive or ergodic (space invariant) or of a white noise type. This invalidates most existing noise models as they were based on the assumption that the noise is additive. However, they do conclude that it is stationary e.g. time invariant.

[HS05] studied the impact of the print and scan process on the coefficients of a discrete Fourier transform (DFT). They find that there is significant noise on all bands except for low frequency bands with high magnitude. The textures and the relationships between DFT coefficients are preserved. They also find that the [0;255] intensity range is transformed into a [70;220] range.

Two document degradation models have been proposed in 2015. In [LCOB15] they propose to model three types of noise: random noise, edge deforming noise and connected component (character) breaking noise. This is a probabilistic model applied on synthetic documents. In [SCE+15], they use real degradations taken from ancient documents and apply them on other ancient documents. They work in the gradient domain which allows them to avoid any binarization and to obtain seamless results. However, the random nature of the location of the added degradations makes the document look unrealistic.

Source of noise	Modeled	References
Cropping	Yes	[LC99]
Non isotropic rescaling	No	
RGB to CMYK conversion	No	
Halftoning	Yes	[LC99, BMI07, PN95, Smo12]
Electromechanical printing parts	in PSF	[BMI07, LC99, MF06, Smo12, YNS05, PN95]
Ink	in PSF	[BMI07, LC99, MF06, Smo12, YNS05, PN95]
Paper coating	in generic model	[MF06]
Paper texture	No	
Paper thickness	No	
Document lifecycle degradations	in generic model	[LCOB15, SCE+15]
Scanner shot noise	Yes	[Smo12]
Scanner optical system	in PSF	[BMI07, LC99, MF06, Smo12, YNS05, PN95]
Scanning resolution	Yes	[LC99, BMI07, PN95]
Scanner spectral sensitivity	No	
Scanner light source	No	
Electromechanical scanning parts	Yes	[BMI07, MF06, YNS05]
Document handling when scanning	No	
Size of document and of scanning window	No	
Dust in printer and/or scanner	No	
Wear of printer and/or scanner	No	
JPEG compression	Yes	JPEG algorithm [ITU92]

Table 1.1: Availability of models for sources of noise in a print and scan process. PSF stands for point spread function.

1.2.3 Summary of the print and scan noise models

We have seen that most of the work on the noise introduced in the print and scan process has focused on modeling the sources of this noise. However many sources of noise have not been taken into account. Table 1.1 summarizes these sources and if they have been studied. We distinguish three kinds of models for a source of noise: a specific model ("yes" in the table) or it is included in the PSF or it is included in another generic model.

As we can see several sources of noise have not been modeled and modeling them all could be extremely difficult. This explains the empirical approaches which describe the noise present in the image of a printed and scanned document. Let us now focus on the security systems that can be applied to documents undergoing this kind of noise.

1.3 Document security systems

Before hybrid documents existed, one had to secure paper only documents - and this was done with seals, watermarks and other physical technologies - or digital only documents - which was done with digital technologies. Unfortunately, physical technologies do not work with digital documents and vice versa. Thus, providing a security technology for hybrid documents is a critical challenge to maintain a sufficient level of trust in these documents.

Most security technologies designed for paper documents are not maintained through a print and scan process. Hence we will not present them except for watermarking techniques that resist this kind of degradation. This section will present hashing algorithms, digital content security algorithms, hybrid document security algorithms (including watermarking algorithms) and finally the proposed semantic hashing framework.

1.3.1 Hashing algorithms

A digital hash algorithm computes a digest for a message. Basically, this is a mathematical algorithm that scrambles and mixes the bits of input data (called a message) in order to produce a much shorter data (called a digest) that is representative of the message. Since such an algorithm works on the bits of the message, the message can be of any kind such as text, image, sound, video. Any digital data can be hashed. A very good overview of digital hash algorithms can be found in [Zau10].

Figure 1.3.1 shows the general process when using a hashing algorithm. Assuming a man wants to secure a message with a hashing algorithm. He first computes

the digest of his message. Then he sends his message to a woman. The digest is transmitted to her either along with the message or through some other way. Then she computes the digest of the message she received and compares it with the digest of the original message. If they are identical, then the message she received is the same as the original one. Otherwise a transmission error has occurred or the message has been tampered with.

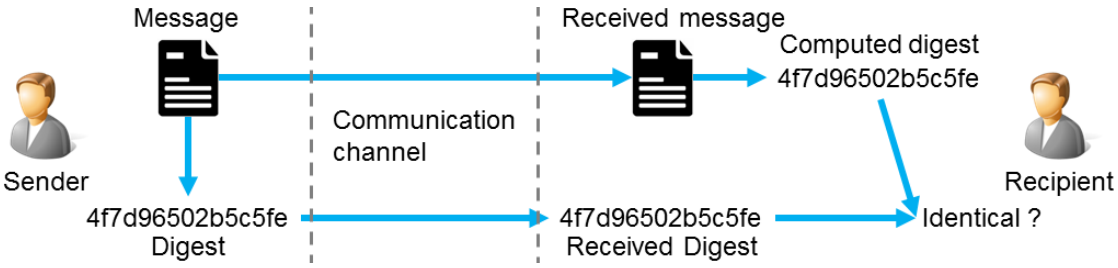


Figure 1.3.1: Process to use a hash algorithm.

The first kind of hash algorithm is cryptographic hashing. It was made in order to be able to control the integrity of the message content without having to read the entirety of the message. This was useful at the beginning of the Internet because of network transmission errors. Cryptographic hash algorithms are now mostly used for security applications and content retrieval. They have several features. The first one is that any small change in the message will change the digest with a very high probability. This is reflected in the collision probability, which is the probability of two different messages having the same digest. Another important feature of hash algorithms is the inability to recover the original message from its digest (hence the name "cryptographic"). The main consequence of this requirement is that any smallest change will completely change the digest. This allows the authentication of a confidential message without having to compromise the confidentiality of the message. The current standard cryptographic hash algorithm is SHA-256 as defined in the Federal Information Processing Standard FIPS PUB 180-4 [BG12] and FIPS PUB 186-4 [KG13]. A security analysis of SHA-256 can be found in [GH04]. A notable, non secure, popular hashing algorithm is MD5 [Riv92]. It is still frequently used to check the integrity of a file downloaded from a website or in peer to peer networks because of its low computational cost.

The second kind of digital algorithm is fuzzy hash algorithms. Contrarily to cryptographic hashes, a small change in the message will only change a portion of the digest. This allows the retrieval of different messages that have similar message parts, but that are not completely identical (hence the name "fuzzy"). For this reason, the content of the message is not as protected as in a cryptographic hash algorithm. The most common fuzzy hash algorithm is ssdeep [Kor06]. Fuzzy hash algorithms are also called perceptual hash algorithms especially in the image

processing community as the fuzzy hash is related to the content of the message and to the way a human perceives it. They are widely used for message retrieval, where the message can be of any kind such as an image [HPSO12], a text [BNV13] or even raw bits of data on a hard drive [WSY13]. A security analysis of perceptual hash algorithms can be found in [KVBP08].

The main issue with this system is that anyone can modify the message and compute the digest for the new message. If that person can replace the digest of the original message by the fraudulent one, then the recipient has no mean of knowing that the message has been tampered with. A classical answer to this issue is to make the hashing algorithm a shared secret between sender and recipient. However, this is difficult to obtain in reality to the point that in 1883 Kerchoffs [Ker83] laid down a principle which says that when studying attacks on a security system, one should assume that the algorithms it uses are known by the attackers. This principle is now a standard for most security analysis.

1.3.2 Digital content security algorithms

Basically, the issue with hashing algorithms is that anyone can replace both the original message and its digest without leaving any trace of it. Thus the efforts of the security community have been done towards preventing such an invisible modification.

Another important aspect of these security technologies is their legal value. No matter what technology one uses, if it does not allow you to sue in court the person who defrauded you, its use is greatly diminished. Thus security technologies evolve with the legislation which, in turn, also evolves with the available technologies.

At the European level, the first legal solution was brought by a directive of the European Parliament in 1999 [Eur99]. It has now been updated with the European regulation called eIDAS [Cou14] whose effect has started on the first of July 2016. A notable difference between these two documents is that the directive had to be implemented by each member state of the European Union. Thus it led to several different security referentials. The regulation is directly applicable as is so that now, all the member states have the same security referential.

The eIDAS regulation defines three levels of security. The first one is called an “electronic signature” and, from a technological point of view, is very much equivalent to the hash security system described previously. The second level is called an “advanced electronic signature”. In this case, the digest is linked to the data but also to the emitter. Figure 1.3.2 illustrates a technological solution to produce such a signature.

The first step is the same as before: we compute a digest of the message. Then this digest is encrypted with a public key infrastructure. This is a system that gives two keys to the message emitter. One key is private and used to encrypt the

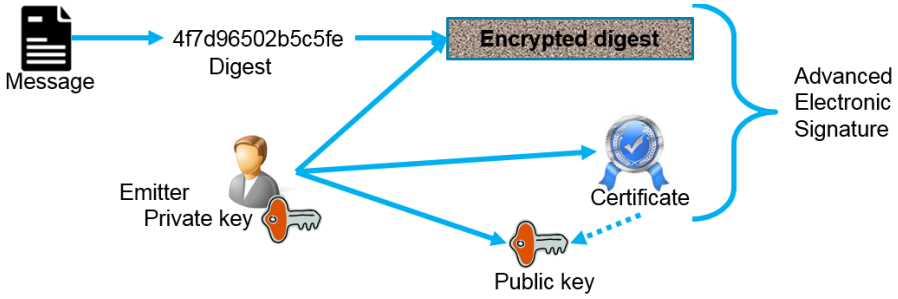


Figure 1.3.2: Process to generate and advanced electronic signature.

digest. This key should remain private and not be given to anyone. The second key is a public key and serves to decrypt the encrypted digest. Everyone has access to it. With this system, only the emitter can encrypt the digest while everyone can read it. This prevents any possibility of modifying the digest without leaving any trace of it. The remaining issue is that in order to read the digest, one needs to know which public key to use. The emitter and the public key are linked through the certificate. This certificate is sent along with the encrypted digest. Thus when someone receives the encrypted digest and the certificate, he can first verify the identity of the person to whom the certificate is related. This ensures that he receives the message from the right person and not a fraudster. Then he uses the public key to decrypt the digest. Finally, the verification process is the same as for an "electronic signature": he computes the digest of the received message and compares it with the decrypted digest.

There is still at least one weakness: the infrastructure and the people who generate the signature and who manage the certificates could be compromised (willingly or not). This is why there is a third level of security called a "qualified electronic signature" which on top of all the above requires that the infrastructures and companies producing and managing the signatures and the certificates be qualified with some certification process.

1.3.3 Hybrid document security algorithms

If the message to secure is a document that can be printed and scanned several times, it is possible to print the certificate and the encrypted digest with the document in the margin in the form of a barcode, a web link or anything similar. The issue rises with maintaining security when scanning the document. A scanner produces an image of the document from which it is possible to obtain the certificate and the decrypted digest.

However, if the digest was computed on the document file in its native format such as PDF or Word, the original file structure and meta-data are lost after

printing and it is not possible to compute a similar digest for the scanned document image. If the digest was computed on a rendered image of the document, the printing and the scanning introduces some noise. This noise will modify the bits of the image and thus the encrypted digest and the digest of the image of the scanned document are bound to be different. Hence the document security is lost.

The current digital security framework does not allow to secure a hybrid document. The main issue lies in computing the digest: the hashing algorithm. This issue is not new and several attempts have been made to solve it.

Shimizu and Kim [SK07] had an idea similar to ours but they project never came to reality[8]. They wanted to identify the components of a document (text and graphics), use an OCR software to extract the text and another algorithm to describe the images. They only went as far as locating the components of the document. The text/graphic identification was not done.

Villán et al. [VVK+07] elaborated two hashing algorithms. The first one is based on the combination of an OCR software (Abbyy Finereader) and a cryptographic hash (SHA-180). The second one is based on a random tiling hash, which computes the average luminance value on 1024 random rectangles for each word. They used an Arial font with a font size of 10 and no emphasis. The text was simple English text. The first algorithm performs rather well with only two errors out of 64 but on a very small dataset. The second algorithm cannot differentiate one character from a similar one such as "o" and "e", thus it is not precise enough.

Tan et al. [TSZZ11] developed a perceptual text hashing algorithm, which is based on the skeleton features of each character. It has the drawback of removing all punctuation from the text which is not sufficiently precise. Furthermore, this algorithm and the ones above can only secure the textual content of a document. Its layout or graphics are not secured.

The Estampille project [BC12, BC13] aimed at developing a 2D-barcode that could be used as a fingerprint. The barcode is printed with extreme precision in a specific printing process. This fingerprint is supposed to be impossible to be reproduced or to be copied without detection. The latest results actually prove that having a dozen copies of an authentic fingerprint is enough to forge one. This does not include the fact that the document content is not included in the fingerprint. Thus once a forged fingerprint is made, it is possible to render any document authentic-like.

The SIGNED project [Mal13] was more ambitious and corresponds to what we can call a "hybrid signature". The goal of the project was to produce a digest of a document that allows the detection of any modification. It is based on a fuzzy hash algorithm, which has the disadvantage of breaching the possible confidentiality of

[8] We had confirmation by the second author that the project is aborted and never went further than this publication.

the document, but this also allows for the localization of the modification. The document is analyzed at the signal level. They cut the document in tiles and use a Discrete Haar Wavelet Transform on each of these tiles. For a document scanned at 600 dpi, the tiles have a size of 64 by 64 pixels. Then a cryptographic hash is applied on each tile and all the digests are concatenated to create a fuzzy digest. The fuzzy digest is then printed on the document. During the verification process the digest of the scan is computed and compared with the one that is printed on it. A distance is computed between these two digests and if it is too large the document is considered to be fraudulent.

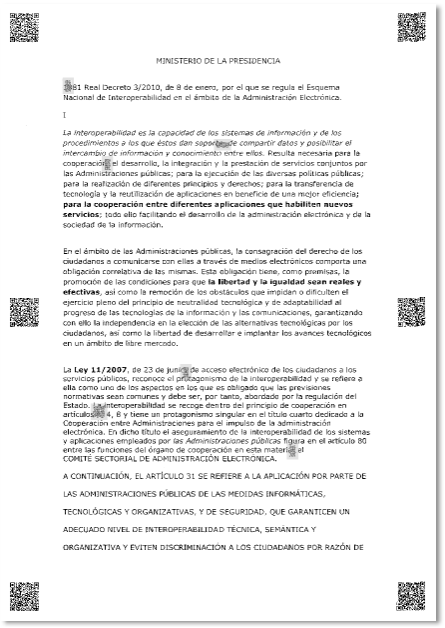
The results of the SIGNED project had to meet six performance indicators given by the industrial partners of the project. As such, they represent a reasonable goal to reach:

1. Probability of missed detection and false alarm
   a) Probability of false alarm (PFA) below 0.001
   b) Probability of missed detection (PMD) below 0.001 with $PFA < 0.001$
      - for the replacement of digits in Arial 8, 10 and 12
      - for the replacement of dots by commas in Arial 10 and 12
2. Collision probability below 0.001
3. Minimum area size to detect a manipulation: 42 by 42 pixels at 600 dpi
4. Throughput below 5 seconds per page
5. Size of the document digest below 4 kiloBytes (kB, 1kB = 1ko)
6. Compatibility with current scanners and printers

A false alarm occurs when a document is detected as fraudulent while it is not and a missed detection is the contrary. The minimum area size means that a manipulation has to be bigger than a square of 42 by 42 pixels to be detected. The project met all these requirements except for:

- The PMD for the replacement of dots by commas that required a bigger font than Arial such as Verdana.
- The minimum area size that is of 64 by 64 pixels at 600 dpi.
- The throughput that was not achieved for the verification phase (no figure was given).
- The size of the document digest that was between 4.8 and 170 kB depending on the required precision.

The solution of the SIGNED project [Mal13] is great except for the size of the digest which can require 6 2D barcodes as shown on Figure 1.3.3. Such a size is unusable in practice.



MINISTERIO DE LA PRESIDENCIA

El Real Decreto 3/2010, de 8 de enero, por el que se regula el Esquema Nacional de Interoperabilidad en el ámbito de la Administración Electrónica.

I

La interoperabilidad es la capacidad de los sistemas de información y de los procedimientos a los que éstos dan soporte de compartir datos y posibilitar el intercambio de información y conocimiento entre ellos. Resulta necesaria para la cooperación, el desarrollo, la integración y la prestación de servicios conjuntos por las Administraciones públicas; para la ejecución de las diversas políticas públicas; para la realización de diferentes principios y derechos; para la transferencia de tecnología y la reutilización de aplicaciones en beneficio de una mejor eficiencia; **para la cooperación entre diferentes aplicaciones que habiliten nuevos servicios**; todo ello facilitando el desarrollo de la administración electrónica y de la sociedad de la información.

En el ámbito de las Administraciones públicas, la consagración del derecho de los ciudadanos a comunicarse con ellas a través de medios electrónicos comporta una obligación correlativa de las mismas. Esta obligación tiene, como premisas, la promoción de las condiciones para que **la libertad y la igualdad sean reales y efectivas**, así como la remoción de los obstáculos que impidan o dificulten el ejercicio pleno del principio de neutralidad tecnológica y de adaptabilidad al progreso de las tecnologías de la información y las comunicaciones, garantizando con ello la independencia en la elección de las alternativas tecnológicas por los ciudadanos, así como la libertad de desarrollar e implantar los avances tecnológicos en un ámbito de libre mercado.

La **Ley 11/2007**, de 23 de junio, de acceso electrónico de los ciudadanos a los servicios públicos, reconoce el protagonismo de la interoperabilidad y se refiere a ella como uno de los aspectos en los que es obligado que las previsiones normativas sean comunes y debe ser, por tanto, abordado por la regulación del Estado. La interoperabilidad se recoge dentro del principio de cooperación en artículos 4, 8 y tiene un protagonismo singular en el título cuarto dedicado a la Cooperación entre Administraciones para el impulso de la administración electrónica. En dicho título el aseguramiento de la interoperabilidad de los sistemas y aplicaciones empleados por las *Administraciones públicas* figura en el artículo 80 entre las funciones del órgano de cooperación en esta materia: el COMITE SECTORIAL DE ADMINISTRACIÓN ELECTRÓNICA.

A CONTINUACIÓN, EL ARTÍCULO 31 SE REFIERE A LA APLICACIÓN POR PARTE DE LAS ADMINISTRACIONES PÚBLICAS DE LAS MEDIDAS INFORMÁTICAS, TECNOLÓGICAS Y ORGANIZATIVAS, Y DE SEGURIDAD, QUE GARANTICEN UN ADECUADO NIVEL DE INTEROPERABILIDAD TÉCNICA, SEMÁNTICA Y ORGANIZATIVA Y EVITEN DISCRIMINACIÓN A LOS CIUDADANOS POR RAZÓN DE

Figure 1.3.3: Example of a document secured with the technology developed in the SIGNED project. Image reproduced from [Mal13].

There are also several hybrid security technologies that try to securely embed data in a paper document in such a manner that it is resistant to print and scan.

The first type of such technology uses visible markings protected with cryptography such as the 2D-Doc technology. It is an official standard of the Agence Nationale des Titres Sécurisés (ANTS), the french government body in charge of producing secure documents [ANT13]. It is developed by AriadNext[9]. In this technology the information on which the digest is computed, is produced by the information system or manually entered by the emitter of the document. In the case of a birth certificate, it can be the name of the child, its place and date of birth. The advanced or qualified signature is then printed on the document with a 2D barcode. This 2D barcode also contains the digital information which is secured (name, date of birth, etc). This allows it to be easily read once the document is scanned. The information which is verified is the one contained in the barcode. The printed text contained in the document image needs to be manually compared with the one extracted from the barcode. The 2D-Doc is designed so

[9] `www.ariadnext.com`

that big billing companies can use it. As such, it is possible to "secure" up to one million documents per hour with this technology[10]. This allows the French phone company SFR[11] to secure all the bills sent to their customers every month. Other companies provide a similar technology such as Alphacode[12] and 2D Origin[13]. A notable fact is that the solution proposed by 2D Origin allows to secure the entire document provided that its owner accepts to store it on a third party server. This drawback is quite frequent and not acceptable for sensitive documents. Furthermore, this security system requires a network access to view the secure version of the document which may not be feasible at all times. The main drawback of this kind of technologies is that since there is no automatic comparison of the embedded information with that contained in the paper document, the paper is not secured (unless a human does the comparison). If the 2D-Doc of a multimillion contract is applied on a cooking recipe, the recipe could be considered secure. Hence the paper (document) should rather be considered like a support of the real information which is the one contained in the barcode/datamatrix.

The second type of technology uses invisible markings and relies on its stealthiness for its security. This is commonly called watermarking. These technologies all have the same issue related to the fact that they, at most, only secure a small part of the information contained in the paper document. Such technology is available with companies like SOOD [14].

Finally, the project SHADES aims at overcoming these drawbacks [EGKO15b]. Since embedding technologies are already well studied, it focuses on the hashing algorithm. It plans to extract the content from a document image in a reliable manner to compute a hash on this information. Hence, when creating a document it is possible to compute a hash on the information extracted from a rendered image of the document and when verifying the printed document, it is possible to do the same thing on the scanned document image. The hashes can then be easily compared. If the document image content extraction algorithms are stable enough, this should allow one to secure all the document. Since only the extracted information would need to be hashed, a compact digest would result from this process. Thus the hash could be embedded by any means that suits the user.

Table 1.2 shows a summary of the main hybrid document security technologies and projects. Since the Project SHADES is not finished, its performance for text verification is not known yet. However we can see that it is the most ambitious project in terms of functionality. The algorithms whose security performance on text is not available (NA) are those which do not allow an automatic text content

[10]We learnt this from a personal discussion with the company team.
[11]`www.sfr.fr`
[12]`www.alphacode.eu`
[13]`www.2dorigin.com`
[14]`www.sood.fr`

verification.

Algorithm	Privacy protection	Automatic verification	Security tied to the content	Entire content secured	Graphics	Layout	Performance on text	Hash size
[VVK+07]	X	X	X				+ -	+
[TSZZ11]	X	X	X				+ -	- - -
Estampille	X	X					NA	+
SIGNED	X	X	X	X	X	X	++	- -
2D-Doc							NA	+
Sood	X	X					NA	++
SHADES	X	X	X	X	X	X	?	++

Table 1.2: State of the art of hybrid security technologies and projects.

1.3.4 Proposed semantic hashing framework

This thesis gave the original idea for the project SHADES (Semantic Hash for Advanced Document Electronic Signature) [15] and is part of it. This project is funded by the ANR, the French National Research Agency. It is an interdisciplinary project which should allow us to create the proposed hashing algorithm and to study the legal consequences of this new technology.

The two main features of the hashing algorithm are the type of content that it can process and the type of modifications that are allowed. We will present them first after which we will present the process to compute the digest.

Type of content that can be secured

The proposed system should extract all and only the information contained in the document in order to make a digest with it. This digest should be such that the document privacy is protected as much as possible. Depending on the industry requirements, it may be necessary to identify the location of the discrepancies between the document being verified and the original document (which is not available, we only have its digest).

Figure 1.3.4 shows the percentage of the different types of content found in all the document types that were studied (bills, payslips, etc.) in a study done by ITESOFT in the project SHADES. Typewritten text and logos are present in all documents and thus are not depicted.

Since this is the first time that one attempts to build a hashing algorithm based on document image analysis algorithms, our objective is to demonstrate a proof of

[15] `http://shades.univ-lr.fr/`

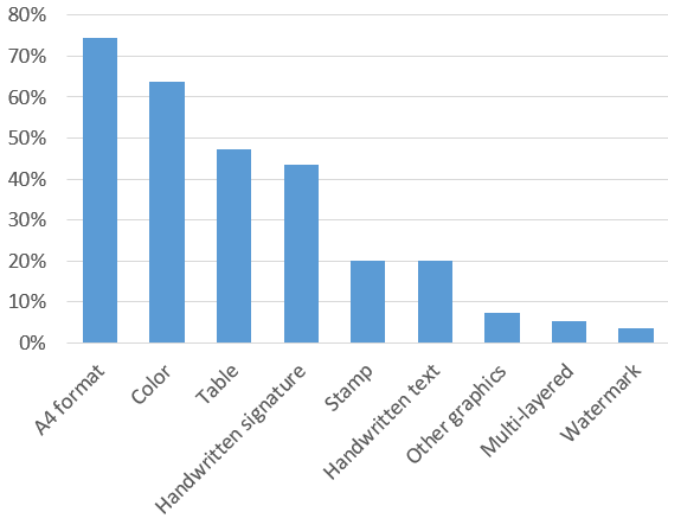


Figure 1.3.4: Percentage of each content type in the document types of the study.

concept that works on usable but simple cases. Hence, the types of content that we will focus on are:

- The typewritten text
- The logos and handwritten signatures
- The layout

We will only extract these components from documents with non overlapping components (single layer documents) in color with a potential extension to gray level images. It would be too difficult to try to reliably separate overlapping components and color images guarantee that we have all the available information.

We will not attempt to secure handwritten text or text font and emphasis because the corresponding analysis algorithms are not reliable enough [STRV15, BTGK+15]. Table and stamps combine text and graphics which is too complicated for a first approach. Finally watermarks are made not to be detected so we will not secure them either.

The functionality presented here should allow us to deal with 27% of the document types. The number of occurrences of each document type is dependent on the company and the business case so we prefer to focus on the variety of the documents.

Now that the scope of the information to be secured is fairly well defined, we need to look at which modifications of the document image are allowed and which are not.

Type of modifications that are allowed

The hashing algorithm is called a semantic hashing because the underlying base principle is that it allows any modification that does not change the meaning of the document. Thus it allows any print and capture (scan, fax, copy, photograph) noise unless it severely degrades the document to the point that part of it could become unintelligible. In this study we focus on the print and scan noise which has been described in Section 1.2. Traces of folding and soft stains are allowed as long as the stains are not opaque enough to hide anything that could have been under them.

Some documents can be created in color but copied in gray levels. This color removal may change the meaning of the content of the document. However, considering its commonality we consider that we should be able to tell whether the only difference between the documents is the lack of colors for one.

The modifications that are clearly not allowed are any visually significant modification of the text, the images or the layout such as a character change or the removal of part of the document.

Concept of "semantic" hashing

Regarding the definition of semantic, contrarily to what is usually understood, here we only focus on very low level semantics, visual semantics. We are not trying to understand the meaning of the text contained in the document. Neither do we try to understand its layout (header, caption, footnote, etc.). We simply make sure that this information is there and not changed. We focus on the physical layout (place of the document parts) and not the logical layout (function of the document parts). Our level of semantics is equivalent to saying "there is this series of characters at this place in the document" or "the top right image is made of three vertical rectangles, with one color each, blue, white and red". We will not say "this text talks about buying a car" or "the French flag is depicted in the top right corner".

Process description

Figure 1.3.5 shows the process of the hashing algorithm of the SHADES project. The steps with thick borders are the ones dealt with in this thesis. The hash computation is only studied for each analysis step and not on a global level.

The geometric correction aims at obtaining an image of the flat document without any borders around the document. This is particularly useful if the document has been photographed with a mobile camera or a webcam. It can also remove the black borders produced by a scanner.

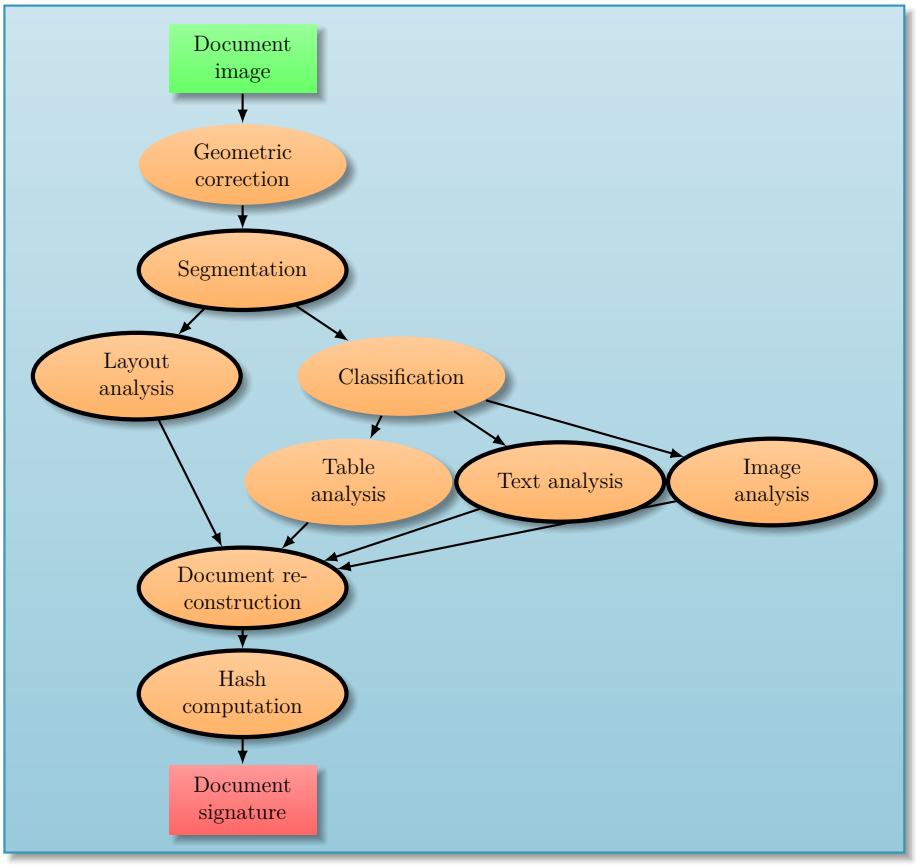


Figure 1.3.5: Algorithm for semantic hash generation

The segmentation locates the components of the document and separates them. It will be detailed in Chapter 4.

The layout analysis provides a unique description for the layout of the components of the document as well as a unique order of the components. This step is closely related to the document reconstruction step. It will be described in Chapter 3.

The classification identifies the type of content contained in the components of the document.

The table, text and image analysis respectively aim at extracting and describing the content of the components containing tables, texts and images. Those components have been identified thanks to the classification step that precedes. The text analysis will be described in Chapter 6 and the image analysis will be described in Chapter 7.

The document reconstruction step gathers all the information extracted by the

previous steps and formats it into a stable document representation. This is merely a concatenation operation based on the components ordering.

The hash computation step computes the hash of the reconstructed document. The matching algorithm to compare two signatures will depend on this step once it is completed.

Considering the dependency between all the processing algorithms, one understands the importance that each algorithm be as stable as possible in order not to jeopardize the performance of the overall system. In particular the segmentation and classification tasks are critical and the output of the classification should be completely stable. otherwise, the following content extraction steps will not be able to produce stable results since they will have the wrong inputs. The same goes between the layout analysis and the document reconstruction. If the former is unstable, so will be the latter.

1.4 Conclusion

In this chapter we have proposed a document typology and used it to identify the types of document on which we will focus in this thesis. These are modern mostly textual (administrative) documents and magazine cover pages whose main degradation is made by a print and scan process. We have also detailed the different sources of noise in a print and scan process more exhaustively than had been done before.

We surveyed existing technologies to secure a hybrid document and highlighted their main weakness: there is no hashing algorithm capable of producing the same output on the images of all copies of a same document. Such an algorithm should focus on the visually meaningful content of the document images and be insensitive to the non meaningful modifications. This is why we call this a semantic hashing algorithm. This thesis proposed the semantic hashing framework of the project SHADES to solve this issue for some specific cases with the help of document analysis algorithms. Since it is the first attempt at creating such a hashing algorithm we favor feasibility over functionality.

We intend to secure the typewritten text characters, the graphics, in particular the logos and handwritten signatures and the layout of single layer documents. The main type of noise/modification that we will focus on allowing is the one generated by printing and scanning a document.

Creating a reliable, compact semantic hashing algorithm for document images has many implications. It means bringing state of the art technologies to a level of performance that has never been reached before. These technologies can then be used for many other applications. Also, hashing algorithms have many other uses than just security. By instance, they are particularly useful to index data and to

detect duplicates. Thus this thesis should benefit significantly both the scientific community of document analysis and the document processing industry.

Now that we have described the types of documents and the types of modifications that we focus on as well as the general framework to process them, we will present in the next chapter the main performance criteria: the stability of the algorithms that we will use.

Chapter 2

Definition and analysis of the notion of stability

In this chapter, we start by presenting a general understanding of what is a stable algorithm. Then we move on to an example of a specific definition from the literature and finally we extend this definition to a generic class of algorithms. The last section of this chapter is dedicated to the evaluation of the stability of an algorithm. We introduce a selection of performance indicators and a diagram to analyze them for algorithms that have at least one parameter.

In order to authenticate two copies of the same document, one needs to produce identical or at least very similar signatures for both documents. This ability to produce similar outputs if the inputs are similar is what we call stability. It should not be mistaken with numerical or dynamic stability which is related to the convergence of an algorithm or of a mathematical series.

The usual way of evaluating document analysis algorithms is with "instantaneous/single result" performance indicators. These are performance indicators that only require one result from the algorithm in order to evaluate its quality. This usually comes with the requirement for a ground truth. These are well known performance indicators such as accuracy, precision, recall, etc.

We could only find three works related to the evaluation of the stability of an algorithm. Peña et al. [PLL99] evaluate the variation of the stability of the results of the k-means clustering algorithm with respect to the initialization method. Guo et al. [GKNK14] design a line extraction algorithm and they evaluate its ability to extract the same lines over a range of photographs of the same objects under slightly different poses. Agrawal et al. [AKB08] study the repeatability of their key-point detector and descriptor. These last two works are done on less than 10

images and only reach a repeatability below 80%. This highlights the novelty of the proposed evaluation criteria.

This chapter is organized as follows:

- Section 2.1 presents some examples to gain a general understanding of the notion of stability.
- Section 2.2 formalizes the definition of stability.
- Section 2.3 proposes a framework for the evaluation of the stability of an algorithm.

These sections are completed by a conclusion. The contributions of this chapter are:

- A proper formalism for the definition of the stability of an algorithm in Section 2.2,
- An evaluation framework/methodology for the evaluation of the stability of an algorithm in Section 2.3.

We will now present a general description of our notion of stability.

2.1 General understanding of the notion of stability

In order to avoid any confusion we will clarify the differences between: accuracy, robustness and stability.

- Accuracy requires a ground truth to evaluate how close a result is to this ground truth. Accuracy can be evaluated with only one result as long as there is also a ground truth.
- Robustness is not a performance indicator itself but rather highlights the ability of an algorithm to maintain a good accuracy even with degraded conditions. Hence, robustness relates to the ability of an algorithm to produce meaningful results when the input is noisy.
- Stability does not require any ground truth. Stability requires at least two results with similar inputs to see how close these results are together compared to how close the inputs were.

In our case, similar inputs are photocopies of the same document. A consequence of this is that an algorithm can be very stable and yet not be accurate. It just needs to always make the same mistakes. Here are two examples of algorithms with an absolute stability and zero accuracy:

- A layout descriptor that always describes only one region at the same position.
- A segmentation algorithm that always produces one region covering the whole image.

The contrary is not true. An algorithm with an absolute accuracy will always produce results that are identical to the ground truth and hence identical between each other. An algorithm that is perfectly accurate is also perfectly stable.

Figure 2.1.1 shows a visual comparison of accurate (first row), robust (second row) and stable (third row) person detection algorithms on clean (first column), dark (middle row) and blurry (last row) images. We can notice the following differences:

- The accurate algorithm can produce very good results provided that there is no noise. This is why it has a poor performance on dark or blurry images (Figures 2.1.1a, 2.1.1b).
- The robust algorithm provides a meaningful output on all images but randomly fails to find part of a person (Figure 2.1.1e) or a person (Figure 2.1.1f).
- The stable algorithm only finds the head and torso of the people, but it always finds them all.

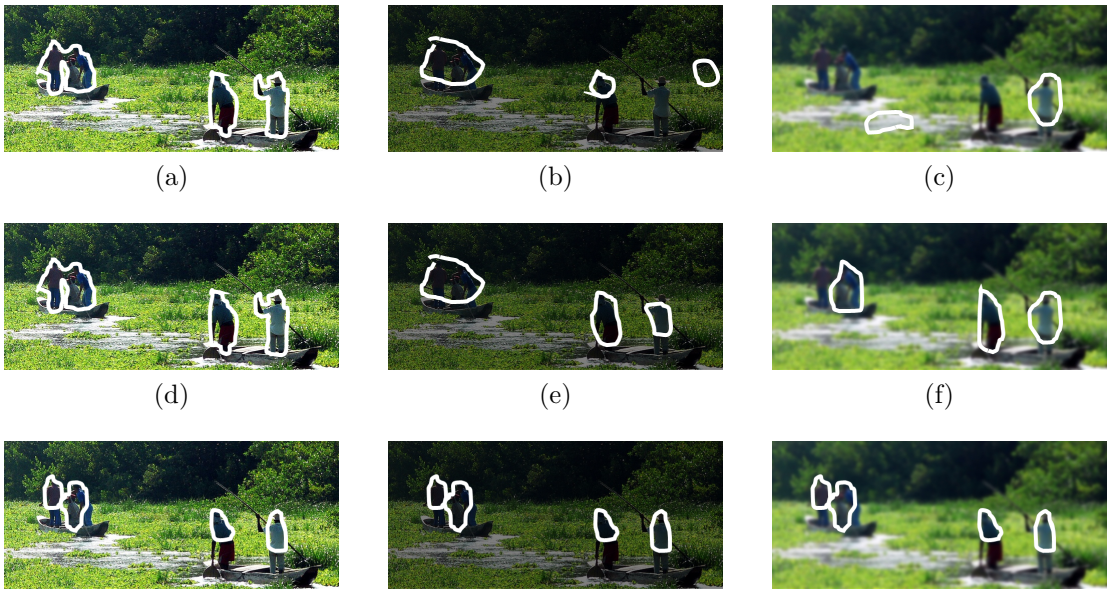
A consequence of this is that if an algorithm performs well on one image, in order to be stable it will have to have the same performance on all other images. Thus the stability requirement is very difficult to achieve and when achieved, frequently leads to an algorithm with high performance or well identified faults. Because they are well identified, these faults are easier to handle and such an algorithm is generally easier to use in an industrial environment. We will now attempt to make a formal definition of the stability of an algorithm.

2.2 Formal definition of the notion of stability

In this section, we will first present an example of a definition of stability for a specific use case and then we will generalize it.

2.2.1 The example of learning algorithms

Bousquet and Elisseeff [BE02] propose a definition of the stability for learning algorithms. Such algorithms are trained on a dataset (the input) and when evaluated have a learning error (the output). They focus on bounding the variation of



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 2.1.1: Comparison of accurate (top row), robust (middle row) and stable (bottom row) person detection algorithms on normal (first column), dark (second column) and blurry (third column) images.

the learning error when a single element of the input dataset is modified as shown in Equation (2.2.1).

$$\forall S \in \mathbb{D}^m, \forall i \in [\![1; m]\!], \|l(A_S), l(A_{S^{\setminus i}})\| \leq \beta \tag{2.2.1}$$

where

- $\mathbb{D}$ is the space of the elements of the dataset
- m is the number of elements in the dataset
- $l(.)$ is the learning error
- A is the algorithm with the learning dataset as index
- $S^{\setminus i}$ is the dataset without its i^{th} element
- $\|.\|$ is a norm (uniform, L_1, point wise L_1)
- β is the upper bound on the error variation

An algorithm is considered stable if Equation (2.2.1) is true for any m and β varies with the inverse of m. This is normal as we expect one item to have less influence on the learning error as the size of the dataset grows. This definition is tailored to their need but some important characteristics start to emerge: the inputs do not vary much and the variation of the output is limited in relation with that of the input.

2.2.2 Generalizing the example

Before defining stability we need to define "similar input" and "similar output". To this intent we introduce similarity functions:

Definition 2.2.1 : **Similarity function.** *A similarity function on the space A is a symmetric binary function of two variables:*

$$c : \begin{cases} A \times A \longmapsto \{0,1\} \\ (x,y) \longmapsto c(x,y) = \begin{cases} 1 \text{ if } x \text{ and } y \text{ are similar} \\ 0 \text{ otherwise} \end{cases} \end{cases} \tag{2.2.2}$$

We can now define what are stable algorithms. We consider that an algorithm is a specific kind of function.

Definition 2.2.2 : **Stable function.** *Let us have*

- *A function f (the algorithm): $f : I \longmapsto O$.*
- *A similarity function s_1 for its input space I and a similarity function s_2 for its output space O.*

f is stable with respect to s_1 and s_2 if and only if

$$\forall \{a,b\} \in I^2, s_2(f(a), f(b)) = s_1(a,b) \tag{2.2.3}$$

Definition 2.2.3 : **Unilaterally stable function.** *Let us have*

- *A function f (the algorithm): $f : I \longmapsto O$.*
- *A similarity function s_1 for its input space and a similarity function s_2 for its output space.*
- *A stability choice $sc \in \{0,1\}$*

f is unilaterally stable with respect to s_1 and s_2 for the choice sc if and only if

$$\forall \{a,b\} \in I^2, s_1(a,b) = sc \Rightarrow s_2(f(a), f(b)) = sc \tag{2.2.4}$$

Basically a unilaterally stable function is function that is stable either for similar input or for dissimilar inputs but not for both.

The first definition can be rephrased by saying that if the inputs are similar, so should be the outputs *and* also if they are different. The second definition only enforces one requirement e.g. we only ask that the output matches similar *or* different inputs, not both cases.

We can see that the definition of Bousquet and Elisseeff [BE02] is that of a unilaterally stable function because they only consider similar inputs. In our case and since we target security applications, we will require a completely stable function (not unilateral). Our input similarity function is the indicator of whether or not the input images are copies of the same document. Our output similarity function will depend on the algorithm.

The definition of a stable function can also be represented with the commutative diagram of Figure 2.2.1. Basically, s_1 should be equal to the composition of f and s_2.

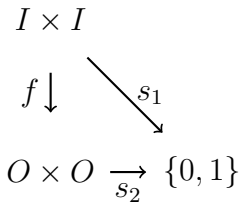


Figure 2.2.1: Commutative diagram of the definition of a stable function.

Before defining new performance indicators, let us summarize what we need to verify the definition of a stable function f:

- A similarity function for the input space: s_1
- A similarity function for the output space: s_2
- A set of similar and different inputs

Obviously $s1$ and $s2$ depend on the space in which they are defined but to the extent possible they should be made independent of f in order to keep a generic definition of stability. Thus, for a proper methodology, before each evaluation of a function we will need to define these items on a case by case basis.

2.3 Evaluation of the stability of an algorithm

Now that we have properly defined what is a stable function, we can measure how much stable is a function e.g. we want to measure how much Definition 2.2.2 is true. We will first define a set of performance indicators to evaluate it and then we will present a diagram to study its variations with one parameter.

2.3.1 Stability evaluation performance indicators

Since this definition can only take a Boolean value (true or false), we will instead measure how frequently it is true. More precisely, given two inputs a and b what is the probability that $s2(f(a), f(b)) = s1(a, b)$?

Since usually it is impossible to have all the possible inputs (no dataset could contain all the document images in the world), this probability needs to be estimated with a dataset of reasonable size. This is the field of descriptive and inferential statistics.

The first step for this is to define a positive and a negative condition. This definition comes from inferential statistics. A positive condition occurs when the inputs are equal. This is the null hypothesis: there is a relationship between the inputs; their distance is 0 and their similarity is 1. A negative condition occurs when they are not. This is the alternative hypothesis: there is no relationship between them, their distance is not zero. This should not be confused with many medical or security related conventions where a test is said to be positive when the outcome is not equal/not normal. If such case is encountered, one should be careful and look at the mathematical definitions behind the terms employed.

The similarity of the algorithm's output can be considered as a prediction. It is a true prediction if the output similarity/positiveness is the same as that of the inputs e.g. if the two sides of Equation (2.2.3) are equal, and false otherwise. The question becomes: what is the probability that the prediction matches the condition e.g. that it is true? A set of classical performance indicators has already been defined to estimate this probability on a given dataset. They are shown in Table 2.1.

Two other performance indicators are frequently used. The F-Measure also called F1-score:

$$F_1 = 2 \times \frac{precision \times recall}{precision + recall} \tag{2.3.1}$$

And the Matthews Correlation Coefficient:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{2.3.2}$$

Total population	**Prediction positive**	**Prediction negative**		
Condition positive	**True positive**	**False negative (Type II error)**	True positive rate, sensitivity, recall $\frac{\sum \text{True positive}}{\sum \text{Condition positive}}$	False negative rate, miss rate $\frac{\sum \text{False negative}}{\sum \text{Condition positive}}$
Condition negative	**False positive (Type I error)**	**True negative**	False positive rate, fall-out $\frac{\sum \text{False positive}}{\sum \text{Condition negative}}$	True negative rate, specificity $\frac{\sum \text{True negative}}{\sum \text{Condition negative}}$
Prevalence $\frac{\sum \text{Condition positive}}{\sum \text{Total population}}$	Positive predictive value, precision $\frac{\sum \text{True positive}}{\sum \text{Prediction positive}}$	False omission rate $\frac{\sum \text{False negative}}{\sum \text{Prediction negative}}$	Accuracy $\frac{\sum \text{True positive} + \sum \text{True negative}}{\sum \text{Total population}}$	
	False discovery rate $\frac{\sum \text{False positive}}{\sum \text{Prediction positive}}$	Negative predictive value $\frac{\sum \text{True negative}}{\sum \text{Prediction negative}}$		

Table 2.1: List of usual performance indicators[1]. The grayed performance indicators are the ones used in this thesis.

Where T, F, P, N stand respectively for true, false, positive and negative. This performance indicator is the correlation coefficient between the results of the algorithm and those of a perfect algorithm. It gives a value between -1 and 1. It is considered to be one of the best single valued performance indicators for evaluating the quality of the results [Pow07].

When analyzing the performance indicators of Table 2.1 we can notice that their denominator can be related to the positive or the negative matches. Thus if the number of positive and negative matches is unbalanced, some statistics can make an algorithm appear better or worse than it really is. We can also notice that horizontally aligned performance indicators on the right such as true positive rate and false negative rate convey the same information since their sum is equal to one. The same relationship can be said of vertically aligned performance indicators on the bottom such as true predictive value and false discovery rate. Thus we only need to choose one performance indicator per pair to have all the information they convey. The performance indicators in gray in the table tend to zero with better algorithms. This allows us to use a logarithmic scale to represent them and

[1]Table reproduced from `https://en.wikipedia.org/wiki/Precision_and_recall`

compare algorithms performance.

Other than just telling us which algorithm is the most stable we may want to have more information from the value of a performance indicator. The F-measure and Matthews Correlation Coefficient do not provide much information on the characteristics of the algorithm. Neither do they answer practical questions such as "what proportion of the authentic documents can I expect to detect as fraudulent?". In contrast to this, we have the following information from the selected performance indicators from Table 2.1:

- The false negative rate (FNR) is the probability that an event is predicted negative when it is positive e.g. that an authentic document is wrongly detected as modified.

- The false positive rate (FPR) is the probability that an event is predicted positive when it is negative e.g. that a modified document is wrongly detected as authentic.

- The false omission rate (FOR) is the probability that an event is positive when it is predicted negative e.g. that a document detected as modified is actually authentic.

- The false discovery rate (FDR) is the probability that an event is negative when it is predicted positive e.g. that a document detected as authentic is actually modified.

They all require the ground-truth of similar/dissimilar inputs to be computed during testing. However, once they are computed for a given algorithm, they can be used in the following cases. FOR and FDR provide information about the veracity of the prediction for a given prediction result (without knowing the ground truth) and thus are widely used in commercial applications. FNR and FPR estimate the veracity of the prediction for a given condition (with knowledge of the ground truth) and are thus used to evaluate an algorithm on a given dataset.

Thus these are the four evaluation performance indicators that we choose to compare the stability of the algorithms we present in this thesis.

2.3.2 NPOD diagram

Algorithms with one parameter are very frequent and when they have more than one parameter, it is possible to fix all the parameters but one. This allows one to study the variation of the stability of a given algorithm with the value of the non-fixed parameter. Figure 2.3.1 shows the NPOD diagram of a typical algorithm on a dataset with a majority of negative conditions (CN). NPOD stands for FNR,

FPR, FOR and FDR. These four performance indicators are plotted for every value of the parameter.

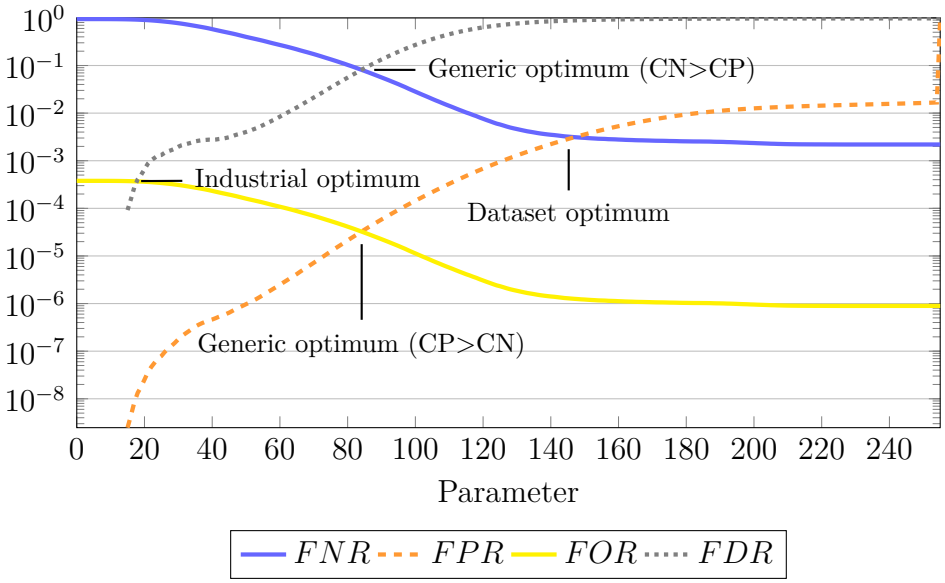


Figure 2.3.1: Typical NPOD diagram of an algorithm

If the FNR and FOR (respectively FPR and FDR) decrease (respectively increase) with the parameter, then large values of this parameter correspond to accepting more false negatives. Hence this parameter is similar to a distance threshold whose larger values lead to matching pairs of inputs with a larger distance between them. In the opposite case, the parameter will be similar to a similarity threshold.

On this diagram, we can see that the FNR and FDR are higher than the FOR and FPR respectively. This is because there is a large majority of negative conditions in the dataset. There are three use cases:

- From a generic perspective, the optimal performance is obtained with the threshold corresponding to the highest intersection point. Thus it is the intersection between the FNR and FDR. Were the dataset to have a majority of positive conditions (CP), it would have been the intersection point between the FOR and the FPR.
- From an industrial perspective, the optimal performance is reached for the threshold corresponding to the intersection point of the FOR and FDR that is interesting.
- In order to optimize the algorithm based on the knowledge of the dataset we can look at the intersection point between the FPR and the FNR.

We can note that, in our datasets, there is always a large majority of negative conditions (pairs of copies of different documents) and the generic optimum will always be the intersection between the FNR and the FDR.

A consequence of the above is that, in order to obtain the same performance in all cases, it would be better to have the smallest quadrangle between the four intersection points. We will call this the zone of interest. In particular the FOR/FDR and FNR/FPR intersections should be as close to each other as possible.

We will also see in the rest of this thesis that some unstable algorithms have degenerated forms of this diagram where not all four intersection points exists. Hence, the NPOD diagram is a useful visual tool to estimate how stable is an algorithm.

2.4 Conclusion

In this chapter we have given a general understanding, a proper definition of the stability of an algorithm. The definition is accompanied by the definition of a similarity function. These definitions have been completed by an evaluation framework based on adequate performance indicators that provide useful insights for both the scientific and industrial community. Finally we have provided a visual evaluation tool to study the impact of the parameters of an algorithm on its stability.

We will now study each type of document image analysis algorithm in details.

Chapter 3

Layout description

This chapter deals with the issue of finding a stable description for the layout of a document. We start by defining what is the layout that we need to describe and the corresponding issues. In particular we present the trade-off between providing a precise description and providing a stable description. A precise description will use numeric values which can easily change and make an unstable description. After reviewing the state of the art, we propose a new descriptor based on the Delaunay triangulation of the centroids of the document regions. This descriptor is paired with a matching algorithm that detects the possible sources of instability. It allows the proposed descriptor to be combined with cryptographic hashing techniques to significantly outperform the state of the art on more than nine hundred layouts. It actually achieves performances that hardly need improvement. It should be noted however, that this descriptor does not perform any layout extraction task and requires a segmentation algorithm to produce the layout. It only describes the layout extracted by the segmentation.

The layout of a document is an integral part of its content. Aesthetically it can have an impact on the impression given to the reader as well as on the reading order. More importantly the text may use some locutions to refer to other parts of the document such as “on the left” or “above”. Changing the layout could then change the meaning of these references. This is why we need to secure the layout of a document. This layout will be secured by the means of a layout descriptor. It will serve for the layout analysis step of Figure 1.3.5. On top of providing a description of the layout it should also define a unique ordering of the layout regions. This ordering will be necessary to produce a stable reconstruction of the document being processed.

This chapter is organized as follows:

- Section 3.1 presents the problem of describing a layout.
- Section 3.2 surveys the state of the art and its issues.
- Section 3.3 present the Delaunay Layout Descriptor (DLD) which is the proposed layout descriptor.
- Section 3.4 evaluates the DLD and compares it to the state of the art.

These sections will be completed by a conclusion.

The contributions of this chapter are :

- The Delaunay Layout Descriptor (DLD), a layout descriptor which represents a breakthrough when compared to the state of the art, in particular for its stability. It is presented in Section 3.3,
- A study of the stability of two state of the art layout descriptors and of the DLD in Section 3.4,
- The L3iLayoutCopies dataset, a dataset of layouts that contains photocopy and print and scan noise without segmentation noise. It is presented in Section 3.4.1.

We will now properly define the problem at hand.

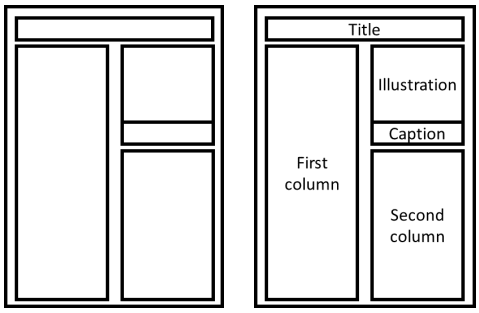
3.1 Problem statement

Before defining the objectives of the layout descriptor, let us properly define a layout.

3.1.1 Definition of the notion of layout

There are two kinds of layouts of a document: the logical layout and the physical layout. Both are shown on Figure 3.1.1. According to [Cha07], the physical layout of a document refers to the physical location and boundaries of various regions in the document image. On the opposite, the logical layout refers to the function of the regions (title, paragraph, caption, etc.). Here we focus on the physical layout.

The physical layout extraction typically relies on a page segmentation algorithm. The page segmentation computes the boundaries of the various regions in the document image. Then, based on these boundaries, the layout extraction determines the spatial relationships between these regions.



(a) Physical layout (b) Logical layout

Figure 3.1.1: The possible types of layouts.

One issue with this definition is that region borders can easily move by a few pixels between two copies of the same document. This is likely to lead to create instabilities.

Therefore, we simplify the above definition by only keeping the physical locations of the regions for the layout. We consider that the position of the region boundaries is the page segmentation. The region boundaries of page segmentation vary in case of noise and are thus very unstable. Hence, we consider that the three segmentation results shown in Figure 3.1.2 have different segmentation results but identical layouts: two regions side by side at the top and two regions on top of each other at the bottom.

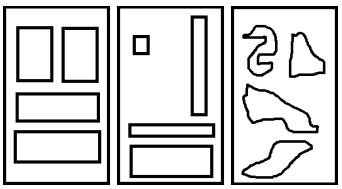


Figure 3.1.2: Three layouts that we consider to be identical

3.1.2 Objectives and challenges

The layout descriptor has two main objectives: being very stable and providing a unique ordering of the layout regions for the document reconstruction step. Two more requirements can be added to this: being precise in order to separate different layouts and using a small amount of memory.

The most challenging objective is the stability since very little work has been done on it. Ideally *similar* layouts should have *identical* descriptors. This would

allow us to combine this descriptor with cryptographic hashing techniques making it extremely compact and allowing for a very fast layout comparison.

The instability of an algorithm will occur with every possibility of changing the output. For every threshold in an algorithm, it is possible that the value on which this threshold is applied is close to the threshold. Hence for several copies of the same document this value could oscillate around the threshold and lead to unstable results. As a result, one should ban the use of "continuous" values such as distances or areas as they contain a threshold for every possible value. Similarly, having parameters may lead to unstable results between two values of the same parameter. Hence parameters should be avoided too. This brings us to the main trade-off when designing any stable descriptor. On the one hand, the more precise the information and the more information is included in the descriptor, the more unstable the descriptor will be. On the other hand, the less precise and the less information is in the descriptor, the less useful the descriptor will be.

When computing the descriptor, we consider that a segmentation algorithm has already been run on the document and the document regions are already identified. The task of the descriptor is to describe the spatial relationships between all the regions of the document.

We will now review the state of the art of layout descriptors.

3.2 State of the art

As far as we know, only few works have been presented on layout descriptors. They can be classified into four categories:

- Matrix descriptors that store the layout information in a matrix,
- Rule based descriptors that store the layout with a set of rules,
- Graph based descriptors that represent the layout with a graph,
- Local hashing descriptors that compute local properties of the layout to describe it.

Since they all have the stability drawback, we will detail it at the end.

3.2.1 Matrix layout descriptors

Álvaro and Zanibbi [ÁZ13] propose a layout descriptor for handwritten math expressions. They use a polar histogram to describe the positions between two symbols and a support vector machine to classify it. There are five classes: horizontal, superscript, subscript, below, and inside (e.g. in a square root). Figure 3.2.1

shows a pair of symbols representing x^2 and the corresponding polar histogram. This algorithm is tailored for mathematical expressions and would be difficult to adapt to a more varied set of spatial relationships.

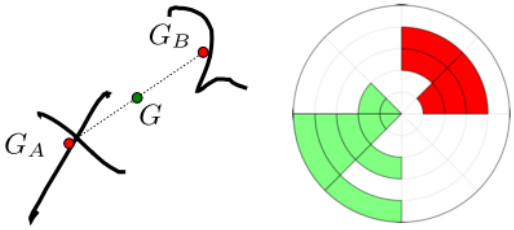


Figure 3.2.1: A pair of symbols (left) and the corresponding polar histogram (right) centered on their center of gravity G. Image reproduced from [ÁZ13].

Another matrix layout descriptor is the MPEG-7 standard [KY01]. It divides an image into an 8 by 8 grid, takes the mean color value of each grid cell, converts it into the YCbCr color space and runs a discrete cosine transform on it. This descriptor can then be used for video retrieval. This algorithm uses a fixed grid layout and cannot represent a generic segmentation result.

3.2.2 Rule based layout descriptors

One of the earliest work on rule based layout description is the one of Esposito et al. [EMS94]. They classify the first page of scientific journal papers according to the journal they belong to. Each region is described with four attributes:

- The width can take 8 qualitative values,
- The height can take 11 qualitative values,
- The type can take 6 qualitative values,
- The position in the document can take 9 qualitative values.

This makes 4752 possible combinations to describe each region. Three types of rules are then added to describe the relative position between two regions. Two binary relations are “on top” and “on the right of”. The third rule can take 9 values and is “aligned”. The values depend on the nature of the alignment (first line, on the left, etc.). The region attributes and the three rules are stored as an unordered list. They use a rule based classifier with heuristics tailored for their dataset. The descriptor performs well on 161 document images. Since they use tailored heuristics this algorithm cannot be generalized to our scenario.

The Description and modification of Segmentation DMOS with the EPF grammatical language [Coü06] can be used to provide an original layout description. EPF allows a user to describe a layout that he seeks and DMOS uses this description to find this layout in images and to identify the image regions corresponding to the description. This is very efficient for segmenting images whose layout is fixed and known but it cannot adapt to the specific layout of the document at hand hence it cannot be used in our scenario.

3.2.3 Graph layout descriptors

The majority of layout descriptors uses graphs. We have found three types of graphs that were used: fully connected attributed relational graphs (FCCARG), trees and generic graphs.

Descriptors using FCCARG

A classical approach is that of [LDMG02]. Similarly to [EMS94], they use their descriptor to classify the first page of scientific journal papers according to the journal they belong to. Each vertex of the FCCARG represents a region and has three attributes:

- The coordinates of the top left corner of the region (in pixels),
- The dimensions of the region (in pixels),
- The font size of the region which can take 3 qualitative values.

The edges represent the relative positions of the regions they link and can take 9 qualitative values. They use a custom graph matching algorithm which performs successfully on 140 text only document images.

A probabilistic variation of this approach has been proposed by [BW03]. Each attribute is considered to be the observation of an independent random variable that follows a Gaussian distribution. They call such a graph a first order Gaussian graph (FOGG). The vertices/regions have four attributes:

- The coordinates of the region center (in pixels),
- The dimensions of the region (in pixels),
- The font size (in pixels),
- The number of text lines in the region.

The edges do not have any attributes other than their existence and represent the neighborhood as defined by the Delaunay triangulation. Two FOGGs with a different number of nodes are compared by adding null vertices and edges. They use the entropy to compute a graph distance. They learn FOGG models from a dataset with hierarchical clustering. The classification is then done by finding the graph model (and its class) with the highest probability of producing the observed graph. Once again they use this algorithm to classify scientific papers according to the journal to which they belong. Their algorithm works successfully on 658 text only documents.

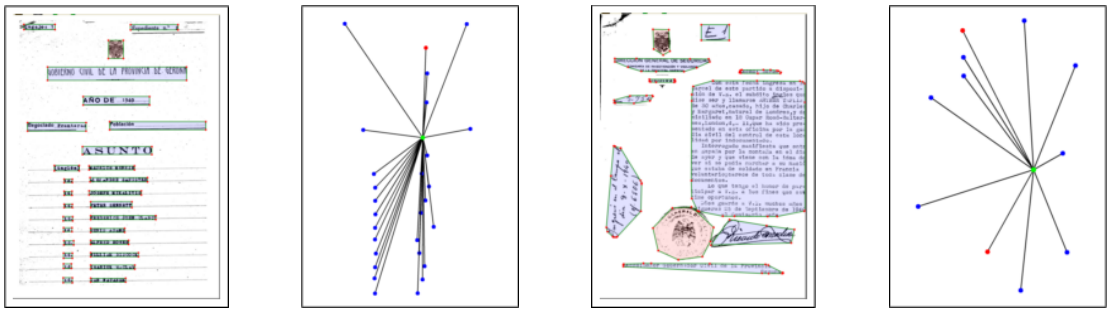
These two algorithms use graph matching algorithms that are computationally expensive and are restricted to text only documents. Thus they are not suitable either for our problem.

Descriptors using trees

Cesarini et al. [CLMS01] modify the X-Y cut algorithm by allowing cuts along black line separators. The cut sequence is encoded in a tree structure. Each intermediate node of the tree represents a cut and each leaf represents a region. The intermediate nodes have one attribute which can take four values depending on the cut direction (horizontal or vertical) and type (white or black). The leaves have one attribute which can take 4 qualitative values depending on the region type. Each tree is then described by the distribution of the subtrees of three nodes that it contains. Each possible subtree is considered as a feature. Since there are 384 possible subtrees there are 384 features to describe each tree. Four extra features are added based on distribution of the size of the regions contained in the document. The document are classified with a multi-layer perceptron with a rejection option. Their algorithm performs well to classify 305 pages of newspaper into five classes. This algorithm is tied to the X-Y cut segmentation algorithm which is too restrictive for our needs.

While the above presented descriptors do not consider robustness to noise, rotation and scale, Gordo and Valveny [GV09] present one which is meant to be invariant to scale and rotation. Figure 3.2.2 shows the graphs built by the descriptor for two documents. This is a star graph/tree centered on the center of gravity of the centroids of each region. Each edge links the center of gravity to the centroid of one region and has four attributes:

- The angle horizontal axis (in radians) θ,
- The edge length (in pixels) L,
- The region area (in square pixels) A,
- The region type in case that the region has a type (text, graphics, etc.) T.



(a) Document a (b) Graph of document a (c) Document b (d) Graph of document b

Figure 3.2.2: The graphs produced by the descriptor of Gordo and Valveny. Image reproduced from [GV09].

The edges are ordered by increasing angle. This allows the creation of a vector/list representation of the graph. This graph can be made invariant to rotation by replacing the absolute angle attribute by the angle difference with the next edge. It can also be made invariant to scale by normalizing the lengths of the edges and the areas of the regions. The distance between two nodes x and y is computed with the following cost function:

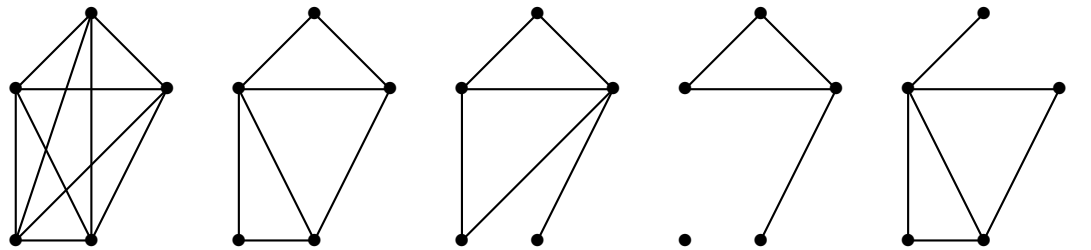
$$\gamma(x,y) = k_1|\theta_x - \theta_y| + k_2\left(1 - \frac{L_x+L_y}{2\max L-x,L_y}\right) + k_3\left(1 - \frac{A_x+A_y}{2\max A-x,A_y}\right) + k_4(T_x \wedge T_y) \tag{3.2.1}$$

The parameters k_1 to k_4 are obtained with a training set. The matching between two graphs is performed with a dynamic time warping. In order to handle rotations which would make a circular permutation of the edges, the vector representation of the graph is concatenated with itself. This algorithm performs well to classify 658 varied documents into 8 categories.

Descriptors using generic graphs

A very interesting approach appeared recently. De Sousa and Kropatsch [dSK15] try to find a canonical (e.g. unique) representation for any set of N points. For this they build a connected graph whose nodes are the N points and require that it has a maximal entropy and a minimal edge weight. This graph will be the descriptor of the N points.

They demonstrate that the nodes of a connected graph with maximal entropy have all possible degrees from 1 to N-1 and two nodes have the same degree. The degree of a node is the number of other nodes to which it is connected. This proves



(a) Fully connected graph (b) Delaunay triangulation (c) Connected graph with maximum entropy (d) Non-connected graph with maximum entropy (e) Connected graph with maximum entropy and minimal edge weight

Figure 3.2.3: Comparison of several graphs linking the centroids of a layout with five regions.

that there are not many possible variants of the proposed graph and adding the constraint of minimal edge weight is likely to make it unique in most cases. Figure 3.2.3 illustrates a comparison of this kind of graph with other possible graphs to describe a five-points layout.

Such a representation would remove any ambiguity. The graph matching procedure is very fast since it relies on the node degree and computing a geometrical deformation between the graphs once the nodes are matched. However the computation of the descriptor requires $\mathcal{O}(N^3E)$ complexity where N is the number of nodes and E is the number of edges. The algorithm has been tested on variation of 256 images of a single object and used the object contours. They applied a combination of rotation, scaling and translation to the images to create several copies of them. Unfortunately no evaluation results were reported so it is difficult to know how well this algorithm performs.

3.2.4 Local hashing layout descriptors

All the previous approaches classify a given layout by comparing it with all the possible layouts in the database. This is very costly as this requires at least as many match computations as there are different layouts. A solution to this is to use a hashing scheme.

This is the approach of [NKI06] who propose a locally likely arrangement hashing (LLAH) algorithm. Originally it is designed to retrieve text documents in a database with an image captured by a camera. It is based on the description of the local spatial organization of the centroids of the connected components (CC). This

spatial organization is reflected by the computation of an affine invariant between four points:

$$inv(A, B, C, D) = \frac{area(ACD)}{area(ABC)} \tag{3.2.2}$$

where $area(.)$ is the area of the polygon. Given the centroid of a connected component C_0, the algorithm takes its 7 nearest neighbors. For each combination of 6 of these 7 neighbors, it computes the above affine invariant for every combination of 4 of the 6 points. The combinations are ordered clockwise. Once all the affine invariants are computed for a given set of 6 points, they are hashed with the following formula:

$$h = \sum_{i=1}^{{}_6C_4} (r_i \times k^i) \tag{3.2.3}$$

where r_i are the invariants, k is a parameter controlling the amount of quantization and ${}_6C_4$ is the binomial coefficient of choosing 4 unordered elements among 6. This hash is used to index a vector composed of the document id, the id of C_0 and the values of the r_i. Each document is stored in a hash database and indexed by all its hashes. The matching is straightforward. Given a document, the algorithm computes the hashes. Then it verifies the consistency of the r_i based on some generic heuristics. Each hash casts a vote for all the documents that are indexed with it. Finally, the document with the highest number of votes is returned. This technique allows constant time retrieval which in practice leads to real time retrieval. However, it only retrieves one document per query.

The main drawback of state-of-the art layout descriptors is the fact that by "invariant" many authors mean reasonably invariant. For instance, if we consider two copies of the same document, a small difference is accepted as long as the descriptors of the copies can be matched. This "weak" invariance requires computing a costly distance for every possible match and, from a security point of view, should be avoided whenever possible. We should favor an exact invariance e.g. the descriptors of similar layouts should be identical. This allows the use of cryptographic hashing which protects the confidentiality of the content and has a faster constant time comparison.

This main issue is related to the use of "continuous" values that we highlighted in section 3.1.2. If a length can vary between 1 and 100, then it has 99 thresholds (if it is an integer). If there were no continuous values, the comparison would be a straightforward test of equality.

3.3 The Delaunay Layout Descriptor (DLD)

We have highlighted the two weaknesses of existing layout descriptors: continuous values and pairwise matching. In order to avoid using continuous values, we propose to store only the layout/graph structure without any attribute. If the descriptor of this layout structure is stable enough, we will be able to use a cryptographic hashing algorithm. Similarly to the state of the art, we consider that the region locations are those of their centroids. Hence we need to describe the layout of this set of points. This fairly strong hypothesis is supported by the good results of our algorithm and is discussed in more detail in the conclusion.

A common algorithm to triangulate a set of points is the Delaunay triangulation. There are three properties of a Delaunay triangulation that influence its practical use and stability and make it more interesting than other triangulation algorithms.

Property 3.3.1. *Given a set of points, there always exists a Delaunay triangulation except when all the points are aligned.*

This property is of the highest interest as it proves that we will always be able to compute a Delaunay triangulation. Yet, it also highlights one case of instability: aligned points. While this will never occur for the whole page as we add a set of three non aligned points outside the borders of the document (see Section 3.3.1); it can occur locally and create local instabilities.

Property 3.3.2. *The Delaunay triangulation tries to maximize the minimum value of the angles inside each triangle.*

This leads to only few near flat triangles whose degenerated form could lead to an instability.

Property 3.3.3. *When a subset of four or more points can be placed on the same circle, the Delaunay triangulation of the points is not unique.*

This means that in this case, the Delaunay triangulation is not stable.

Apart from these properties, the use of the triangulation graph has many advantages when dealing with the print and scan noise described in Section 1.2. It is not influenced by rotation or scaling and cropping has no influence either as long as the regions are not cropped significantly. If the regions are cropped significantly, then the document has been modified and should not be considered as authentic. The local warping introduced by the print and scan is very moderate and hence easily handled since it will not change the triangulation. The other sources of noise are not present after the segmentation step.

This is why, our descriptor is based on the Delaunay triangulation of the centroids of the regions of the layout. We will now see the Delaunay triangulation

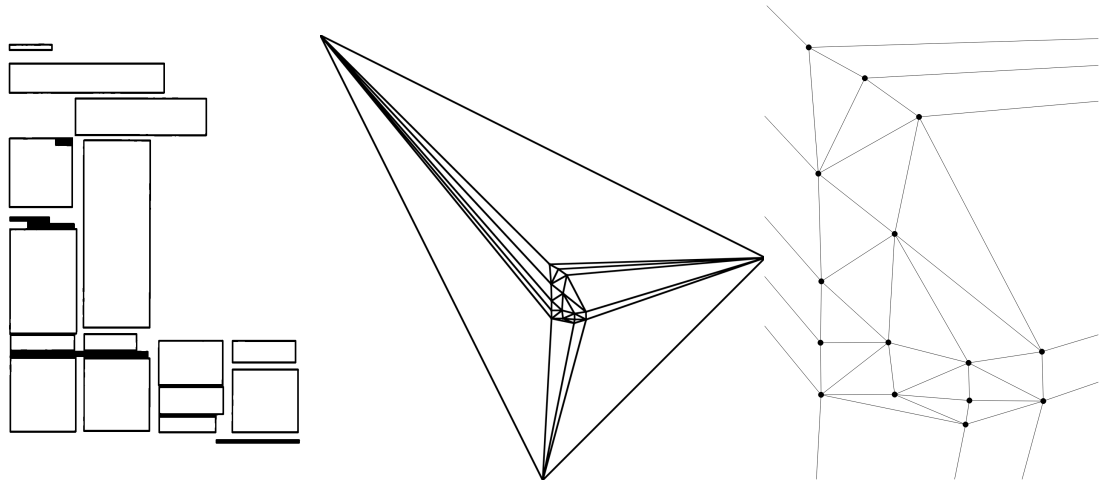


Figure 3.3.1: From left to right: initial layout, example of a Delaunay triangulation produced by OpenCV and zoom in on the triangulation graph. The vertices are the centroids of the regions of the layout.

algorithm in order to understand how it works. Then we will present the algorithm to compute the Delaunay Layout Descriptor and finally we will describe its matching algorithm.

3.3.1 Computation of a Delaunay triangulation

The Delaunay triangulation algorithm adds three points outside of the document image in order to compute the triangulation. One is far in the top left direction and will be the root of the graph. The two other points are the farthest points from the center of Figure 3.3.1. They are at the bottom and to the right of the other points.

The pseudo code of the triangulation algorithm is shown in algorithm 3.3.1. The update method updates the removed triangles to include the new vertex a (the last vertex in Tv). This is done by adding the vertex a to these triangles. This makes quadrangles that are split along one diagonal to make new triangles that will be added to Tl. The condition to choose along which diagonal to split the quadrangle is detailed in Section 3.3.3.

3.3.2 Computation of the Delaunay Layout Descriptor

Our goal is to describe as uniquely as possible the graph of the Delaunay triangulation of the centroids of the regions that have been segmented. For this we

Algorithm 3.3.1 Delaunay triangulation algorithm

Input: a set of vertices V.
Output: the list of Delaunay triangles.

create empty vertex list Vl
create empty triangle list Tl
create three outer vertices $v1$, $v2$, $v3$
add $v1$, $v2$, $v3$ to Vl
add the triangle $[v1, v2, v3]$ to Tl
for all vertice a in V **do**
  add a to Vl
  create empty triangle list $RemovedTri$
  for all triangle t in Tl **do**
    if a is inside the circum cirle of t **then**
      remove t from Tl
      add t to $RemovedTri$
    end if
  end for
  update(Tv, Tt, $RemovedTri$)
  delete $RemovedTri$
end for
return Tl

use the adjacency matrix which contains all the organizational information of the graph.

Obviously the edge distances and the point positions are missing from this description. However, the fact that this description is the one of a Delaunay triangulation adds many constraints which limit the confusion possibilities. This also gets us back to the trade-off between precision and stability. Storing continuous values in our descriptor will inevitably make it unstable.

One last issue is the ordering of the vertices. For a given graph only one adjacency matrix exists modulo the ordering of the vertices. To this intent we use a variant of the breadth-first search (BFS) algorithm starting from the top left corner.

The top left corner is a good starting point because the Delaunay triangulation algorithm always generates it. Our algorithm differs from the BFS as it adds a specific ordering of the children of a given node. Let us consider the situation of Figure 3.3.2 where A is a parent node and B and C are its children. x is the horizontal axis. B and C are ordered by increasing value of the angles $\overrightarrow{Ax}, \overrightarrow{AB}$ and $\overrightarrow{Ax}, \overrightarrow{AC}$ in $[-180°; 180°]$. Here, both angles are negative and C comes before B.

The complete ordering algorithm is described in Algorithm 3.3.2.

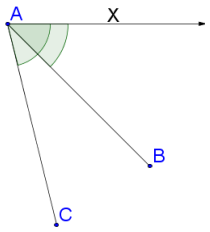


Figure 3.3.2: One node with two children

Algorithm 3.3.2 Graph ordering algorithm

Input: a Delaunay graph G, a starting vertex V.
Output: the ordered list of vertices.

create empty vertex list Vl
add v to Vl
mark v as done
for $i = 0$; $i \leq G.nbVertices - 1$; $i++$ **do**
    create empty vertex list $Children$
    for all vertices a adjacent to $Vl(i)$) **do**
        if a is not done **then**
            add a to $Children$
            mark a as done
        end if
    end for
    order($Children$)
    for all vertices a in $Children$ **do**
        add a to Vl
    end for
    delete $Children$
end for
return Vl

Once the graph is ordered we compute its adjacency matrix. The adjacency matrix is the descriptor which we call the Delaunay Layout Descriptor (DLD). It has no parameter and virtually no threshold. This adjacency matrix can be hashed in order to save memory space. Figure 3.3.3 shows the entire descriptor computation process.

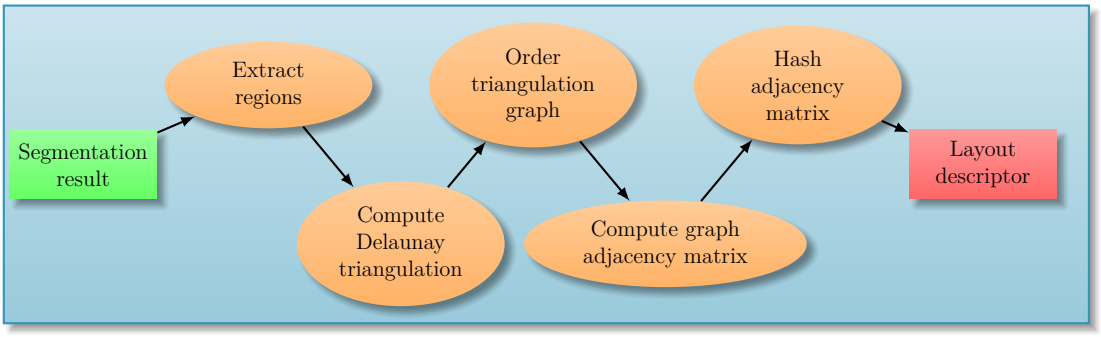


Figure 3.3.3: Process to compute the Delaunay Layout Descriptor.

3.3.3 Sources of instability of the DLD

We have created a descriptor with no threshold or parameter. However this is not sufficient for an absolute stability. As explained before, there are three cases of instability: aligned points, flat triangles and co-cyclic points. Flat triangles and aligned points cover the same geometric situation since the vertices of a flat triangle are aligned. There is also a fourth one related to the ordering of the graph. Thus, we propose an appropriate matching algorithm absorbing these possible sources of instability of the descriptor. It creates the possible variations of the descriptor and finds the exact matches with the descriptor(s) that we want to match. The way the matching algorithm handles the flipping of edges inside a quadrangle, the instability due to near aligned points and the instability due to the implicit threshold at $-180°/180°$ in the ordering of the graph is explained in the following.

Edge flipping

The Property 3.3.2 can be achieved by appropriately choosing the diagonal to split the quadrangles in the update function of the triangulation algorithm. If we consider the quadrangle ABCD of Figure 3.3.4, it can be split along $[AC]$ or along $[BD]$ to make two triangles. To choose which edge must be created, the algorithm computes the sum of the opposing angles: $\widehat{ABC} + \widehat{CDA}$ and $\widehat{BCD} + \widehat{DAB}$. One of them is bigger than 180°(this is a trivial mathematical property since their sum is equal to 360°). To satisfy Property 3.3.2, the quadrangle must be split along the segment that joins the opposing angles whose sum is the biggest. Here it is $\widehat{ABC} + \widehat{CDA} = 200°$ and the quadrangle will be split along $[BD]$.

One immediate source of instability comes when the sums are both equal to 180°. This means that all four vertices are on the same (circum) circle and there are two possible triangulations of the quadrangle. This explains Property 3.3.3. The situation worsens in the digital world due to the quantization of the coordinates

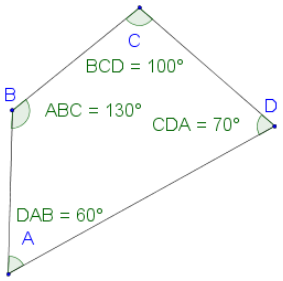


Figure 3.3.4: Example of a quadrangle that needs to be triangulated

to integer pixel values. This introduces an error on the angle measurements which needs to be taken into account. There exists an instability area when a couple of opposing angles forming a quadrangle have a sum within $[180° - \epsilon; 180° + \epsilon]$. If this is the case, we will flip $[AD]$ to $[BC]$ and try to match both possibilities of splitting the quadrangle. Figure 3.3.5 shows such a case of instability.

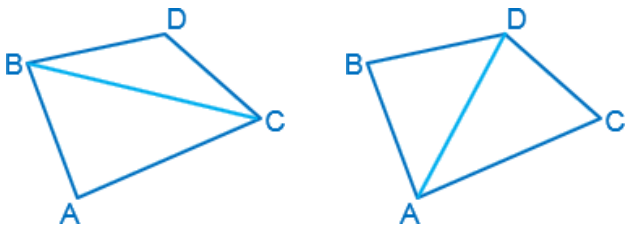


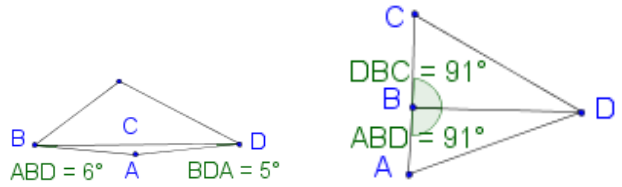
Figure 3.3.5: Two unstable ways of splitting a quadrangle into triangles when the sum of opposite angles is equal to 180°.

The parameter ϵ remains to be found and can be defined by the user. The value of ϵ will be discussed in the evaluation.

Aligned points

As stated in Property 3.3.1, for any number of aligned points there exists no triangulation. If a subset of our centroids is aligned, this will create a zone of instability as this subset will be difficult to triangulate. Furthermore, we just stated that there is an error margin on angle measurements. This increases the zone of instability due to aligned points.

Let us consider the two situations of Figure 3.3.6, and the point of view of the edge $[BD]$. Situation 3.3.6a occurs when $\widehat{ABD} + \widehat{ADB} < \epsilon$. In this case, the edge makes a flat triangle with A. This should be triangulated the other way (with $[AC]$ splitting the quadrangle). The situation 3.3.6b is a proper triangulation. However, it could very well have been triangulated the other way around with $[AC]$ splitting



(a) $[BC]$ makes a flat triangle

(b) $[AC]$ could make a flat triangle

Figure 3.3.6: Two situations with nearly aligned points.

the quadrangle. This situation occurs when $180° + \epsilon > \widehat{ABD} + \widehat{DBC} > 180° - \epsilon$. Figure 3.3.7 shows such an instability.

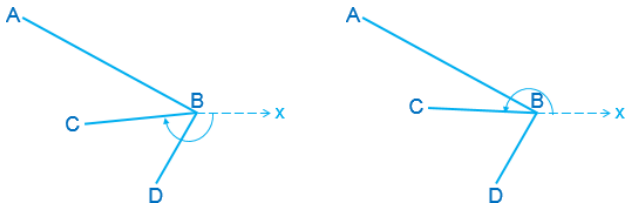
Figure 3.3.7: Two unstable ways of splitting a quadrangle into triangles when three vertices are nearly aligned.

We perform similar tests on both sides of the edge and with both vertices of the edge. This defines the zone of instability related to the alignment of points. If this situation occurs for an edge, we flip the edge/diagonal inside the quadrangle and try to match both splitting configurations.

To prevent this situation from occurring too often, one should try to segment text columns rather than text paragraphs or text lines. Text lines and paragraphs are usually aligned in the same column. While this is preferable, this is not compulsory. Our tests include layouts with paragraphs, aligned regions and we performed a specific test with a table layout (e.g. regions aligned horizontally and vertically) and the algorithm still performs perfectly.

Ordering implicit threshold

The algorithm to transform the Delaunay triangulation into a graph contains a step when the children of a node are ordered. This step contains an implicit threshold. With the notations of Figure 3.3.8, let us consider that $\widehat{\overrightarrow{Bx}, \overrightarrow{BC}} < -180° + \delta$. If we make an angle measurement error of $\epsilon > \delta$ then we can have $\widehat{\overrightarrow{Bx}, \overrightarrow{BC}} < -180°$. This angle will be congrued back inside $[-180°; 180°]$ to become $\widehat{\overrightarrow{Bx}, \overrightarrow{BC}} + 360° < 180°$. This changes the ordering of C from being the first to being the last child.



(a) Order: ABCD (b) Order: ABDC

Figure 3.3.8: Instability in the node ordering when the first child has an angle near 180°.

To deal with this, we define an instability zone equal to $[-180°; -180° + \epsilon] \cup [180° - \epsilon; 180°]$. If one or more children are within this zone of instability, we change the ordering of the children of the current node by performing a circular permutation. We then try to match all the possible orderings of the graph with all the order instabilities and their combinations. Because of the exponential number of combinations, this is the most costly part but since the graph is traversed from the leftmost node and the instabilities occur for the children on the left of their parent, their are not so frequent.

3.3.4 Matching of the DLD

Now that we have identified all the sources of instability, we can test all their combinations to match the layout L. While there are no parameters to compute the DLD, the matching algorithm has two parameters related to the performance requirements that we have. The first one is ϵ for the angle error and the second one is n for the number of simultaneous instabilities.

When combining several instabilities we recompute them. For instance, we apply one instability. Then we recompute the instabilities and only after that do we apply a second instability. This is motivated by the fact that applying one instability in a quadrangle can create new instabilities for the edges of the quadrangle.

The number of instabilities due to the Delaunay triangulation can potentially be rather large. As the combinatorial of their combinations will grow exponentially, we limit the number of simultaneous instabilities to a value n. For instance, if $n = 2$, we will only consider the cases when a maximum of two instabilities occur in the whole layout.

The ordering instabilities are quite rare as we start from the top left corner and they occur for the children on the left of the current node. Hence we will test all the ordering instabilities. The matching procedure is detailed in Algorithm 3.3.3 where the function “delaunayInstabilities” finds all the instabilities related to edge flipping and aligned points and the function “orderingInstabilities” finds all the

ordering instabilities.

Algorithm 3.3.3 Matching algorithm for the DLD

Input: a layout L and a database of layout descriptors S.
Parameters: angle error ϵ, maximum number of simultaneous instabilities n.
Output: the list of matches in the database.

```
1: create empty list of matches Matches
2: add (find L in S) to Matches
3: for all delaunayInstabilities(ε, n) do
4:    update L
5:    for all orderingInstabilities(ε) do
6:       update L
7:       add (find L in S) to Matches
8:    end for
9: end for
10: return Matches
```

3.4 Evaluation of the DLD

In order to evaluate the DLD we created a layout dataset that contains copies of layouts. We will compare the DLD with baseline algorithms which will be presented next. Finally, we will analyze the evaluation results.

3.4.1 Testing dataset: L3iLayoutCopies

To test our algorithm we created a database of 15 layouts similar to the ones in Figure 3.4.2. The main challenge in creating this dataset is to produce the print and scan noise that remains after the segmentation task. Using copies of the document and applying a document image segmentation algorithm would not work because the segmentation algorithm would make errors. These errors would result in different layouts for several copies of the same document. Instead, this dataset should contain similar layouts for the copies of the same document.

Hence we devised the creation process depicted on Figure 3.4.1. We started with a digital layout (the output of a segmentation algorithm similar to those of Figure 3.4.2) which we printed twice on one printer (arrow number 1, p1 and p2). Then we photocopied the prints (arrow 2) making 4 pages (2 prints, p1-2 and 2 copies, p1-2c1). We photocopied these four pages again (arrow 3) making 8 pages (2 prints p1-2, 4 copies p1-2c1-2, 2 double copies p1-2c1c1). We then scanned in black and white these pages twice on two scanners (arrows 4 for first scanner and 5

for the second) making 32 layout images. We repeated this process with an other printer making a total of 64 images of the same layout. This whole process was done for 15 layout images. The total size of the dataset is then $15 \times 64 = 960$ images. The scanners added salt and pepper noise which created many regions made of one or two pixels. Such noise would not be produced by a segmentation algorithm and we removed it manually from the dataset images.

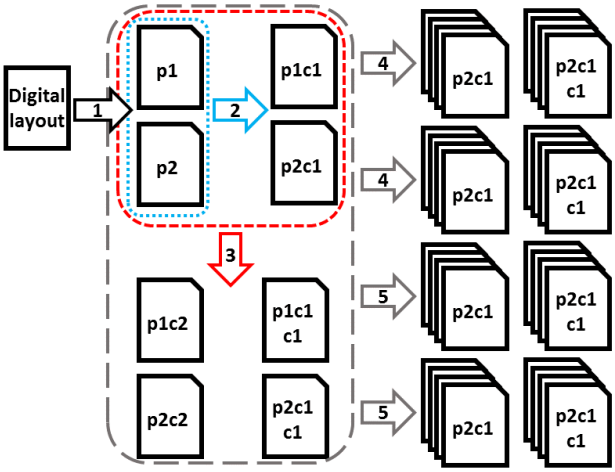


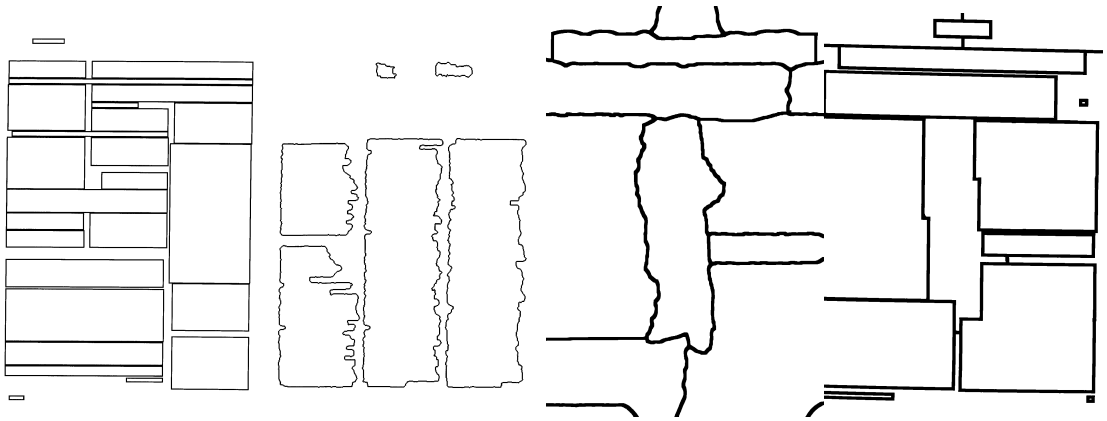
Figure 3.4.1: The creation process of our dataset.

The dataset contains scale variations as the printers add margins around the layout images and hence change their scale. We also used batch scanners that have introduced a surprisingly significant amount of skew (about $5 - 10°$). Hence we can say that this dataset is representative of the print and scan noise described in Section 1.2 and that remains after the segmentation step.

This method has the advantage of reflecting the impact of print and scan degradation on the layout of the document image while ensuring the perfect stability of the layout of the segmentation results. Thus, this dataset will allow us to test the robustness and stability of our descriptor to real print and scan noise.

The layout images are the results obtained by three segmentation algorithms, PAL [CYL13], JSEG [DM01] and Voronoi [KSI98] on 14 random documents of the PRiMA dataset used for the Page Segmentation Competition of ICDAR 2009 [ABPP09]. We chose this dataset and those layouts as they are varied and contain both Manhattan (Figure 3.4.2a and 3.4.2b) and non Manhattan layouts (Figure 3.4.2c and 3.4.2d). A Manhattan layout is a layout similar to the layout of the streets of Manhattan, with well divided square regions. The layouts contain between 6 and 28 regions. Among these 15 layouts two of them are identical but obtained with different segmentation algorithms: they have the same number of regions with approximately the same size and the same positions. Regarding the

similarity function on this dataset (which is the input of the layout descriptor algorithm, see function $s1$ defined in Section 2.2), these two layouts are considered identical. Thus there are actually 14 different layouts, one having twice more copies than the others. This will allow us to study the stability with respect to the segmentation algorithm. It is available on `http://shades.univ-lr.fr/datasets/`.



(a) Layout 1 (PAL) (b) Layout 2 (JSEG) (c) Layout 3 (Voronoi) (d) Layout 4 (PAL)

Figure 3.4.2: Four of the 15 layouts we used to test the descriptor, the algorithm used to make them is in parentheses. Layout 3 and 4 are the two identical layouts produced by different algorithms. Layout 4 has been modified to be similar to layout 3. Layout 1 and 2 are Manhattan layouts while the others are not.

3.4.2 Baseline algorithms

We compare the results of our descriptor with two other methods that we presented in Section 3.2: the one of Gordo and Valveny (G & V) [GV09] which claims to be robust to scale and rotation and the one of Nakai et al. (LLAH) [NKI06] which makes use of hashing techniques and is very robust too.

The descriptor of Gordo and Valveny contains four pieces of information:

- The angle between the edge joining the centroid of a region and the centroid of all the regions and the next edge,
- The length of this edge,
- The area of the region,
- The type of the region (text or non text).

We only use the first three and set the fourth one to text all the time as it is not available in our case. This will only increase the stability of the descriptor as there is one less variable per region. We use the matching algorithm and metric that they defined. Considering the reported results and the small influence of the distance parameters on the error rate and the average precision, we take $k_1 = k_2 = k_3 = k_4 = 1$. If the distance between two layouts is below a given threshold then they are considered identical.

We use the original implementation of LLAH and replace the word centroids by the region centroids. The feature descriptors are stored in a hash table along with the number of the document to which they belong. When matching a document each feature is searched in the hash table and the associated documents get a vote per match. The quantization parameter k is set to 14. Thus we can obtain a matrix V_{ij} which contains the number of votes of document j when searching the features of document i. This number of votes is divided by the minimal number of features of the two documents making a score ratio. This score ratio can exceed 1 because several variations of the features are looked for in the hash table. Then we consider that the score ratio above a given threshold are proper matches and those below are not. It should be noted that LLAH usually works best with at least a hundred points so its performance may not be as good as it is when used to retrieve documents.

3.4.3 Evaluation results

According to our stability evaluation framework defined in Section 2.3, we still need to define the similarity function for the input and the output spaces. The similarity function for the input space is the indicator of whether the images are photocopies of the same layout. There are two variations of this similarity function. One considers that the two segmentation results of the same layout are different layouts (s_{11}) and the second one considers that they are the same (s_{12}).

Regarding the similarity function for the output space, the algorithms of Gordo and Valveny and Nakai et al. are completely unstable if one seeks an exact stability (exact matches). Their performance for such a criteria is shown on Figure 3.4.3 with a threshold equal to 0. The false negative rate (FNR) is equal to 98% (it is supposed to be close to 0%). Thus in the following we will evaluate the weak stability of these algorithms; hence the use of their original matching algorithms as a similarity function for the output space. The requirement for an exact stability will only be maintained for the DLD. Its output similarity function is the equality of the descriptors.

Table 3.1 summarizes all the results. All the values should be as low as possible. For the false negative, positive, omission and discovery rates (FNR, FPR, FOR, FDR) and the matching time, the first value is obtained for s_{11} and the second

one for s_{12}. n is the number of regions in a document and m is the number of layouts/unique documents to match. We have chosen the best values of the DLD that have a matching time lower than that of the other descriptors. s_{11} uses a 5°angle error and 3 instabilities while s_{12} uses a 15°angle error and 2 instabilities. We have chosen the distance and score ratio thresholds that give the best trade-off for the algorithms of Gordo and Valveny and LLAH.

Perf. Ind.	DLD	G & V	LLAH
FNR (%)	**0.0/0.8**	26.7/35.0	46.1/45.0
FPR (%)	5.2/**0.0**	**1.9**/2.8	3.3/3.7
FOR (%)	**0.0/0.1**	1.9/2.9	3.3/3.7
FDR (%)	**5.0/0.0**	26.7/34.9	46.3/44.9
Computational cost of descriptor	$\mathcal{O}(n \log \log n)$	$\mathcal{O}(n \log n)$	$\boldsymbol{\mathcal{O}(n)}$
Size of the descriptor	$\boldsymbol{\mathcal{O}(1)}$	$\mathcal{O}(n)$	$\mathcal{O}(n)$
Computational cost of matching	$\boldsymbol{\mathcal{O}(1)}$	$\mathcal{O}(m)$	$\boldsymbol{\mathcal{O}(1)}$
Descriptor computation time	**0.01 s**	0.05 s	0.06 s
Matching time	**0.04/0.01 s**	0.06 s	0.07 s
Memory use	**284(96) MB**	4.7(3.9) GB	(114) GB

Table 3.1: Summary of the results. The best results are in bold.

We have also added some other performance indicators: the computational cost and the memory usage. We first look at them from a theoretical perspective and confirm our findings with an experimental analysis.

Analysis of the stability of the algorithm of Gordo and Valveny (G&V)

Figure 3.4.3 shows the NPOD diagram (introduced in Section 2.3) for the algorithm of Gordo and Valveny. The blue curves are for s_{11} and the yellow ones are for s_{12}. The FDR and the FPR curves overlap for both cases.

The algorithm performs significantly worse for s_{12} than it does for s_{11}. This is an expected challenge and we shall see how the other algorithms perform. Also, its stability is insufficient as it should be in the vicinity or below the 10^{-2} mark.

Analysis of the stability of the algorithm of Nakai et al. (LLAH)

Figure 3.4.4 shows the results for LLAH. We can see that the case with s_{12} is also more difficult. The increasing and decreasing nature of the curves are reversed which means that the threshold for LLAH has the inverse role than that of G&V. This is true: the distance for G&V is smallest for identical layouts while the vote ratio of LLAH is largest for identical layouts.

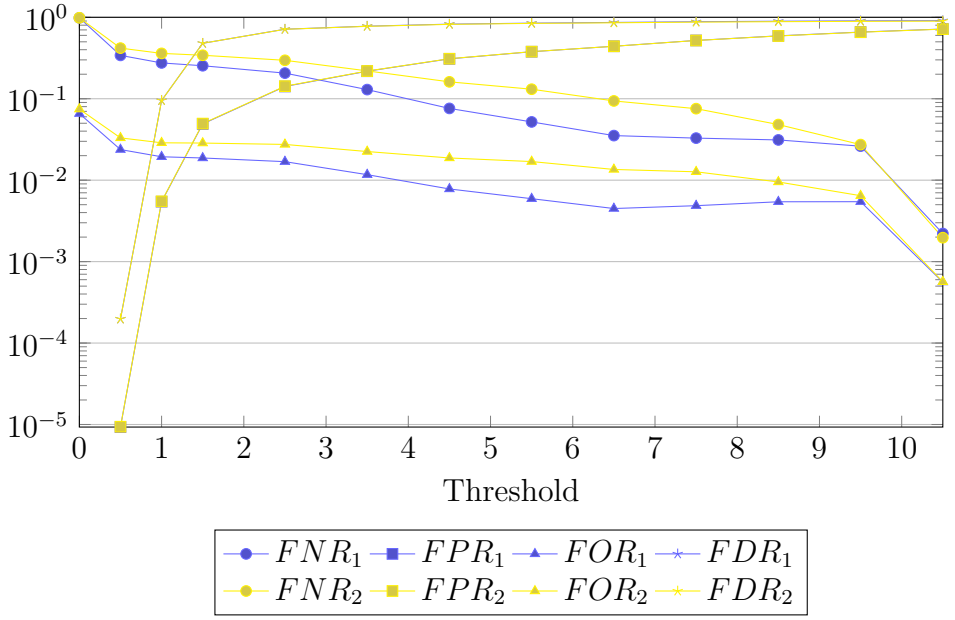


Figure 3.4.3: Performance of the algorithm of Gordo and Valveny. The performance indicator index indicates which similarity function is used to compute it. The FPR and FDR curves overlap for both cases.

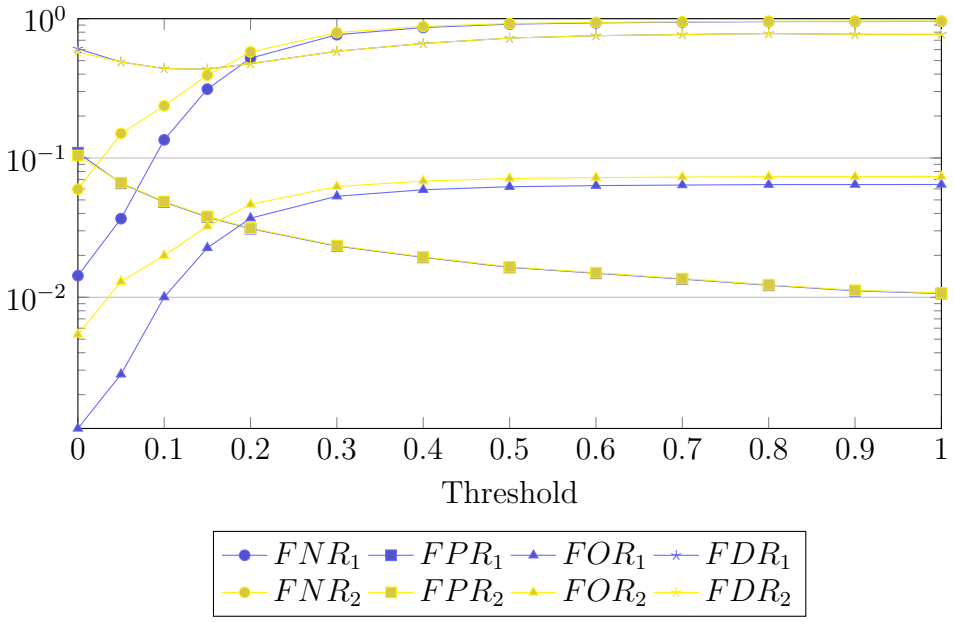


Figure 3.4.4: Performance of the algorithm of Nakai et al. The performance indicator index indicates which similarity function is used to compute it. The FPR and FDR curves overlap for both cases.

There is no intersection point between the FOR and the FDR. This is because the FDR does not go down to 0 when the threshold increases. Thus the optimal point from an industrial point of view is the one that minimizes the FDR.

We can also see that the gap between the curves for s_{11} and s_{12} in the intersection zone is larger than the one of G&V. This highlights the fact that LLAH is more influenced by the segmentation algorithm than G&V. At last, its performance is worse.

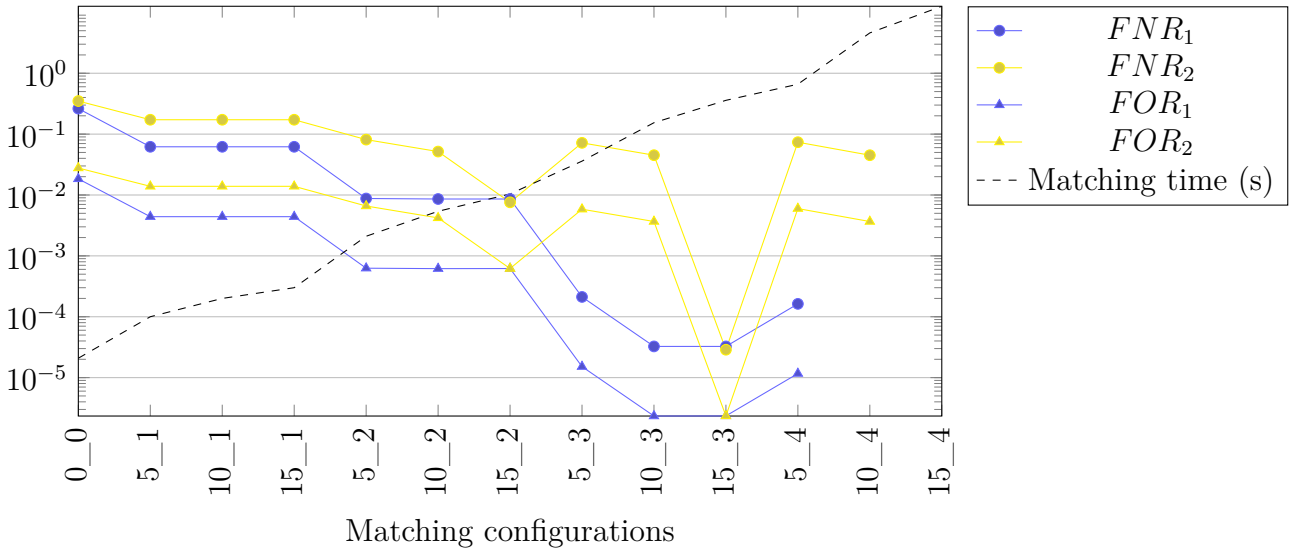


Figure 3.4.5: Performance of our algorithm depending on the angle error and number of simultaneous instabilities. They are indicated on the x axis in the format "*angle_instabilities*". The non plotted values for 10_4 and 15_4 are equal to 0. The performance indicator index indicates which similarity function is used to compute it.

Analysis of the stability of our algorithm

Figure 3.4.5 shows the results for our algorithm. It never makes any false positives except for s_{11} where the two segmentation results of the same layout create some "false" positives. Thus we removed the curves for the FPR and the FDR as they are always equal to 0 in the relevant cases.

The graph in figure 3.4.5 is different from those of G&V and LLAH because there is no distance or similarity threshold to choose. The only trade-off is between the performance and the computational time. This is why the different configurations of angle error ϵ and simultaneous instabilities have been ordered by increasing matching time.

We can see that checking for only one instability already gives a performance comparable with the one of the other algorithms with matching times of less than a millisecond. We can also notice that the importance of the angle error grows with the number of simultaneous instabilities. It does not have much impact with one or two instabilities but it severely impacts the performance from three instabilities onwards. Considering the very low number of errors, it is possible that this last conclusion is related to our dataset.

Descriptor computation

The worst case computational cost of the Delaunay triangulation is $\mathcal{O}(n^2)$ as implemented in OpenCV 2.4.9 but it can be brought down to $\mathcal{O}(n \log n)$ with the sweep line algorithm [LL92]. For most cases, it will actually be $\mathcal{O}(n \log \log n)$. The transformation of the Delaunay triangulation into an ordered graph costs $\mathcal{O}(n)$ as each centroid is processed once. The total computational cost of the descriptor is then $\mathcal{O}(n^2)$ in its current implementation and it can be optimized to $\mathcal{O}(n \log n)$ and to $\mathcal{O}(n \log \log n)$ in the general case.

Gordo and Valveny's descriptor can be computed in $\mathcal{O}(n \log n)$ because of the sorting algorithm required to sort the features by ascending order of their first value. LLAH can be computed in $\mathcal{O}(n)$ as each region is only processed once independently from the other regions. These results and the following ones are reported in Table 3.1.

Size of the descriptor

Once hashed, the DLD has a constant size, hence its memory size is $\mathcal{O}(1)$. The other descriptors are computed region wise and hence use a memory size of $\mathcal{O}(n)$.

Matching computation

Regarding the matching of one layout with the database, for the DLD it can be achieved in $\mathcal{O}(1)$ with respect to m. This is due to the use of cryptographic hashing (we are looking for an exact match). Similarly LLAH uses hashes and requires $\mathcal{O}(1)$. Gordo and Valveny's descriptor requires $\mathcal{O}(m)$ computations as the query needs to be matched on a one by one basis with all the layouts in the database.

Descriptor computation and matching time

The computational complexity of the descriptor and matching algorithms may be similar but this only shows the dependency of the computation time with given parameters. Two descriptors requiring n operations and $1000 \times n$ operations both

have a complexity of $\mathcal{O}(n)$. Yet one is a thousand times slower than the other. Thus it is useful to measure how much time they actually take to perform their task.

The measurements were made on an Intel Core i7 3740QM with 8 cores at 2.7 GHz and 8 GB of RAM. The algorithms are multi-threaded to the extent possible. Computing the DLD takes on average 12 ms. Matching a layout with the whole database with an angle error of 5° and a maximum of 2 simultaneous errors takes on average 2.1 ms and 3.6 ms with 3 simultaneous instabilities. If we use an angle error of 15°, these values go up to 10.4 and 359 ms respectively. It should be noted that this matching time is independent from the size of the database and could be improved as the computation of the instabilities and of their combinations has not been optimized.

Gordo and Valveny's algorithm takes 49 ms to compute a descriptor and 61 ms to match a query. LLAH takes 56 ms to compute a descriptor and 69 ms to match a query.

Memory usage

While the memory usage should be directly related to the size of the descriptor, its practical implementation and implementation constraints can change it significantly. Similarly to the computation time, two descriptors using respectively n bits and $1000 \times n$ bits scale up with $\mathcal{O}(n)$. Yet one uses a thousand times more memory than the other.

To compute this performance indicator for all descriptors, we assume a database of one million layouts and 20 documents/images per layout. This makes a database of 20 million documents. We also consider that each document contains 12 regions on average and that an integer is stored on 32 bits (or 4 Bytes) of memory. The values in brackets in Table 3.1 indicate theoretical values while the other values are obtained experimentally.

The memory space taken by the DLD is 256 bits for secure applications using SHA256. In the case of a non secure application using other hashing algorithms such as MD5 [Riv92], it can be reduced to 128 bits. Each layout will then require $(128 + 20 \times 32)/8 = 96$ Bytes. 128 bits for the descriptors and 20×32 bits for the 20 numbers of the associated documents. The whole database will then require 96 MB. We created an index for a virtual database and it uses 284 MB of memory.

This size difference is explained by the fact that the theoretical value does not take into account implementation constraints such as memory alignment and storage structure (unordered_map in our case). We can see that this has a significant impact on the real memory usage.

Gordo and Valveny's descriptor contains 4 values per region plus one value to identify the document. Hence the theoretical size of the database would be

$(4 \times 12 + 1) \times 4 \times 20e6 = 3.9$ GB. Experimentally we obtained a size of 4.7 GB.

LLAH stores one hash (8 Bytes or 2 integers) per word/region in the document. In Takeda et al.'s paper [TKI11] they consider a normal number of 200 words. Each word/region is associated to a document which means that there are two integer values per word/region: the hash and the number of the document. Actually, the algorithm stores the hashes of the descriptors of all the permutations of 6 out 7 neighbors. There are 7 such permutations. This hash is associated to the document id, the point id and the ${}_6C_4 = 15$ affine invariants. Hence the theoretical size of the index is $7 \times 17 \times 4 \times 12 \times 20e6 = 114$ go. This value neglects the possible collisions e.g. the words that have the same hash. For each collision, there is one hash less to store. Experimentally and with significant memory reduction techniques, Takeda et al. obtained a size of 120 GB (in RAM memory) for 200 words which corresponds to a theoretical size of 1.9 TB. In any cases, our algorithm uses far less memory.

3.5 Discussion and conclusion

In this chapter we present a stable document layout descriptor (the Delaunay Layout Descriptor) based on a Delaunay triangulation of the centroids of the regions of the document. It comes with a matching algorithm to obtain outstanding performances. Currently it is the only available algorithm that can be used in a security application thanks to the possibility of applying a cryptographic hashing to its entirety. We have also created a useful dataset for the evaluation of layout descriptors.

We will discuss three topics in this section. First, we will look at how representative of a given layout is the DLD. Then we will conclude on its performance and finally we will explain its use for the evaluation of segmentation algorithms that will be done in the next chapter.

Discussion

The DLD contains very little information about the layout (no angle, distance, area). Hence one could question how much it is representative of a given layout. The performance of the DLD proves this representativeness. The DLD does not use the size of the regions, neither does it use the distance between the centroids of these regions. But, for a mostly convex region, the bigger the region, the further its centroid will be from the other centroids. Hence the area of a region is reflected by the distance between the centroids. The Delaunay triangulation is directly dependent on these distances between the centroids and their relative positions. Hence, the DLD indirectly contains both pieces of information. It is invariant to

scale and to a certain extent to rotation. This is expected as rotating a layout too much can change this layout.

Conclusion on the performance of the DLD

We have shown that the DLD improves the state of the art in every aspect. Its FNR, FPR, FOR and FDR can be as low as 0% in our experiments. It also reduces by a factor 17 the memory required to index a document database and can match a document against a database of any size in less than a second up to 13 seconds depending on the required level of performance.

When using the Delaunay Layout Descriptor, one should have in mind one tip to leverage all its power. The segmentation algorithm should not produce too many aligned regions. Hence, it should not segment text lines but rather text paragraphs or even better: text columns. However, we tested the DLD with several copies of a 7 by 4 grid layout and it performed perfectly again. This means that while the above advice remains valid, the DLD still performs adequately if it is not respected. Our test dataset, L3iDocCopies, contains layouts with as little as 6 regions and as much as 28 regions which proves that the descriptor should work in most cases with a similar number of regions. All the documents are multi-columns but since we add three extra points during the computation of the triangulation, single column documents should not be an issue either.

Finally, the Delaunay Layout Descriptor is fast, stable, robust, precise and concise beyond all expectations. From our point of view, it solves the issue of describing a layout. Its implementation could probably be improved as it has not been extensively optimized especially regarding the combination of the instabilities. The next challenge related to this descriptor is to have a segmentation algorithm with the same level of performance as our descriptor will not work if the layout is wrong or unstable.

Use of the DLD to evaluate segmentation algorithms

Since the DLD requires a stable document image segmentation algorithm to produce a stable layout description, it may have made more sense to create a stable segmentation algorithm first. However in order to produce a stable segmentation algorithm, we need to evaluate its stability. This requires the definition of a similarity function for its output space e.g. between two segmentation results. A segmentation result describes the regions of the document which also represent its layout. Hence we first needed to build a function capable of comparing the layouts produced by a segmentation algorithm e.g. a layout descriptor.

This explains the critical nature of the stability requirement as it bounds the stability that we can measure on the segmentation algorithm. If the segmentation

algorithm produces two similar layouts but the layout descriptor is not stable enough to recognize these layouts as similar, this could lead to a wrong result. Hence we cannot expect to measure a stability for the segmentation algorithm that will be higher than that of the layout descriptor. We are limited by the "precision" or "sensitivity" of our measuring tool.

Chapter 4

Document image segmentation

The document image segmentation step is a crucial part of the proposed semantic hashing framework. This chapter presents the issue of stability for such algorithms and gives a thorough overview of current techniques. Then we benchmark the stability of three state of the art document image segmentation algorithms and of one natural scene image algorithm on almost one thousand images. It turns out that all algorithms are completely unstable and that the natural scene image segmentation algorithm provides interesting results on document images. Using color images, multi-scale analysis and texture cues may help improve this situation.

A typical paper document content extraction process contains many steps among which is a segmentation step. This is shown in Figure 4.0.1. Document segmentation aims at dividing the document image into meaningful parts. These parts can be glyphs, words, text lines, paragraphs, regions (usually with one type of content such as text or graphic). A common issue with segmentation algorithms is that in order to split the document image properly we need to understand its content and vice versa. This paradigm is related to the associationist and Gestalt theories of vision [TG80]. This is why document segmentation algorithms can include and work in symbiosis with a classification algorithm that will identify the content of the document.

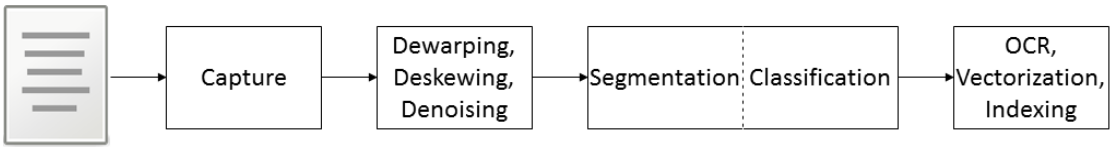


Figure 4.0.1: A classical document content extraction process

Segmentation algorithms can be applied to document images but also to a set of document images (in order to segment a book into its chapters for instance), to natural images [VS12], to medical images [PWK12] and even to 3D meshes [KLT05]. More generally a segmentation algorithm can be viewed as a specific kind of clustering or partitioning algorithm or of classification algorithm when it labels the parts that are segmented. More formally for the value of each input element $I(x)$ a segmentation algorithm associates a region number or a label $J(x)$ where x is the element index. x can be the page number, the node index, or even the pixel coordinates in an image.

$$\forall x, \quad I(x) \overset{Segmentation}{\rightarrow} J(x) \tag{4.0.1}$$

This creates a set of arc-connected regions with a uniform label. Several regions can have the same label.

Some segmentation algorithms are capable of identifying the type of content contained in each area. In section 3.1.1 we defined two kinds of identification: the physical layout and the logical layout. The physical layout relates to the location of the regions of the document. Sometimes it also includes the nature of the content such as typewritten text, handwritten text, graphics, diagram, picture, decoration, etc. The logical layout relates to the function of the content such as header, footnote, main body, etc. Considering the framework presented in Section 1.3.4, it is not necessary that the segmentation algorithm produces any labeling. The classification step in Figure 1.3.5 can do it if necessary.

The goal of this chapter is to present an overview of the current segmentation technologies and their stability. Hence it is organized as follows:

- Section 4.1 presents the challenges of segmentation algorithms.
- Section 4.2 surveys the state of the art.
- Section 4.4 evaluates the stability of four algorithms from the state of the art.

These sections will be completed by a conclusion.

The contributions of this chapter are:

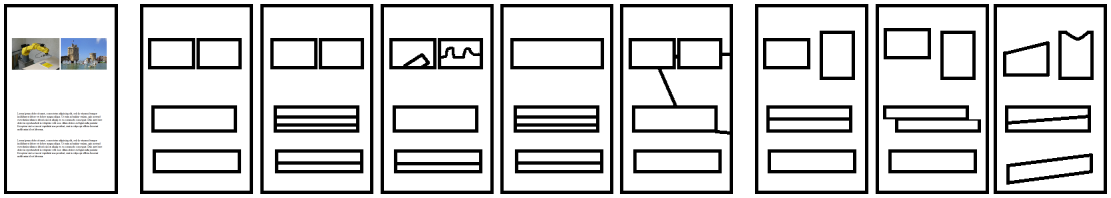
- A thorough survey of the state of the art and of the limitations of segmentation algorithms since 2008 in Sections 4.2 and 4.3,
- A typology of document segmentation algorithms that contains more information than previous typologies in Section 4.2.1,
- A thorough study of the stability of four state of the art segmentation algorithms in Section 4.4,

- The L3iDocCopies dataset, a dataset of photocopies of documents (magazine cover and inside pages, technical documents, scientific papers). It is presented in Section 4.4.2,

- A set of recommendations and comments to improve the stability of segmentation algorithms in Section 4.4.5.

4.1 Problem statement

The main challenge will be to produce stable segmentation results over several copies of the same document. Figure 4.1.1 shows the difference between precise and stable segmentation algorithms. The precise segmentation algorithm produces results that contain the boundaries of the document regions, but it randomly adds some of other boundaries that create new regions. The stable algorithm fails to segment correctly the first paragraph and may have some location and boundary noise, but it never adds any region or changes the document layout.



(a) Input image (b) Precise segmentations (c) Stable segmentations

Figure 4.1.1: The difference between precise and stable segmentation algorithms.

So far segmentation algorithms have only been evaluated based on how close they are to the ground truth segmentation. No one has paid attention to the repeatability of their results such as producing always the same number of regions or the same layout. Hence, it is likely that segmentation algorithms are unstable. This is why we do a thorough survey of them and study their stability in order to choose the best possible algorithms. Since no one has attempted making a stable segmentation algorithm before we will focus on the simplest task: identifying the physical layout. The algorithms that are used for the benchmark will be presented in detail in Section 4.4.1.

4.2 State of the art of segmentation algorithms

One of the first document image segmentation algorithms is the Run-Length Smoothing Algorithm (RLSA) [WCW82]. This classical algorithm was followed in 1992 by the X-Y cut algorithm [NSV92]. Many other algorithms have been presented and surveyed since then [Nag00, MRK03]. Two recent surveys propose formal definitions and detail the main trends of document segmentation algorithms [NJ07, Kis14], but they do not include algorithms after 2007. There has also been a significant number of competitions and benchmarks providing an overview and a comparison of state of the art techniques [SKB08, APBP09, ACPP11, LLS11, ACPP13b, ACPP13a, ACPP15, MRHR15], but they are far from exhaustive. Thus we will survey the algorithms that have been published since 2008 included.

Line segmentation algorithms can serve for document image segmentation as long as they clearly detect the beginning, end, top and bottom of the lines.

While natural image segmentation algorithms are not made for document images, that does not mean that they cannot perform this task as shown in a recent study [EGKO16]. Nevertheless we do not survey them as they are usually not tested on document images. The interested reader can refer to [VS12] for a thorough overview of them. Neither do we consider the basics of segmentation algorithms. If needed [NJ07, Kis14] provide a very good introduction to them.

4.2.1 Typology of segmentation algorithms

Before going through the in-depth survey of segmentation algorithms, it is wise to define a typology to organize them. Document image segmentation algorithms are typically classified into three groups [MRK03, NJ07]: top-down, bottom-up and hybrid algorithms. Top-down algorithms start from the whole page and try to partition it. Bottom up algorithms start from a small scale and try to agglomerate the elements at this scale into bigger elements up to the scale of the whole document. There are three main scales from which they start: pixels, connected components and "patches" which is a user-defined scale. This classification is very objective but does not reflect the capabilities and limitations of each algorithm. It only reflects the order of information processing.

Kise [Kis14] classifies first the algorithms according to their capability of segmenting documents with overlapping layouts such as a stamp on top of some text. This allows one to select a suitable algorithm based on the segmentation task at hand. However this typology only considers classification algorithms for segmenting documents with overlapping layouts which is too restrictive.

A given segmentation algorithm may not be able to segment any layout. This

is the main limitation of such an algorithm and we use it to classify the surveyed algorithms into three groups. The layout segmentation limitation can come either from the way the algorithm itself works (group 1) such as X-Y cut that has been written to segment a specific kind of layout with only square horizontal and vertical regions (called Manhattan layout, see Section 3.4.1). It can also come from the parameters given to the algorithm (group 2) such as Voronoi, which is versatile, but requires different parameters depending on the document style (font size, noise characteristics, connected component size distribution, etc.). A third group of algorithms attempts to overcome these limitations and could potentially not have any, such as neural networks. The overall algorithm type classification is shown in figure 4.2.1. Thanks to the groups we defined, it allows us to represent both the techniques and the limitations of the algorithms. Most algorithms rely on one main technique but also make use of other secondary techniques to obtain intermediate data. As such, the classification of an algorithm is necessarily fuzzy and we classified each algorithm based on its core technique.

The algorithms in group 1

They usually aim at segmenting a specific, predefined kind of layout such as a Manhattan layout. Hence they can be used without any training. There are three subcategories in this group:

- The algorithms that make clear assumptions about the document layout either define this layout with a grammar, a set of rules or they assume that it is a Manhattan layout and use projection profiles.
- The algorithms that use filtering techniques to make the document regions appear usually rely on RLSA, mathematical morphology or other filters. The filters characteristics reflect the assumptions made on the layout geometry.
- The algorithms that try to identify straight lines use a Hough transform to identify straight lines or square borders, or identify white space alignments in which case the “lines” may be invisible.

The algorithms in group 2

Their difference with the algorithms in group 1 is that they try to adapt to local variations in the document in order to be able to segment a broader range of layouts with the same algorithm. The counter part of this is usually a higher number of parameters which are difficult to tune and may require training. These algorithms are usually only limited by the values assigned to their parameters. There are three subcategories in this group:

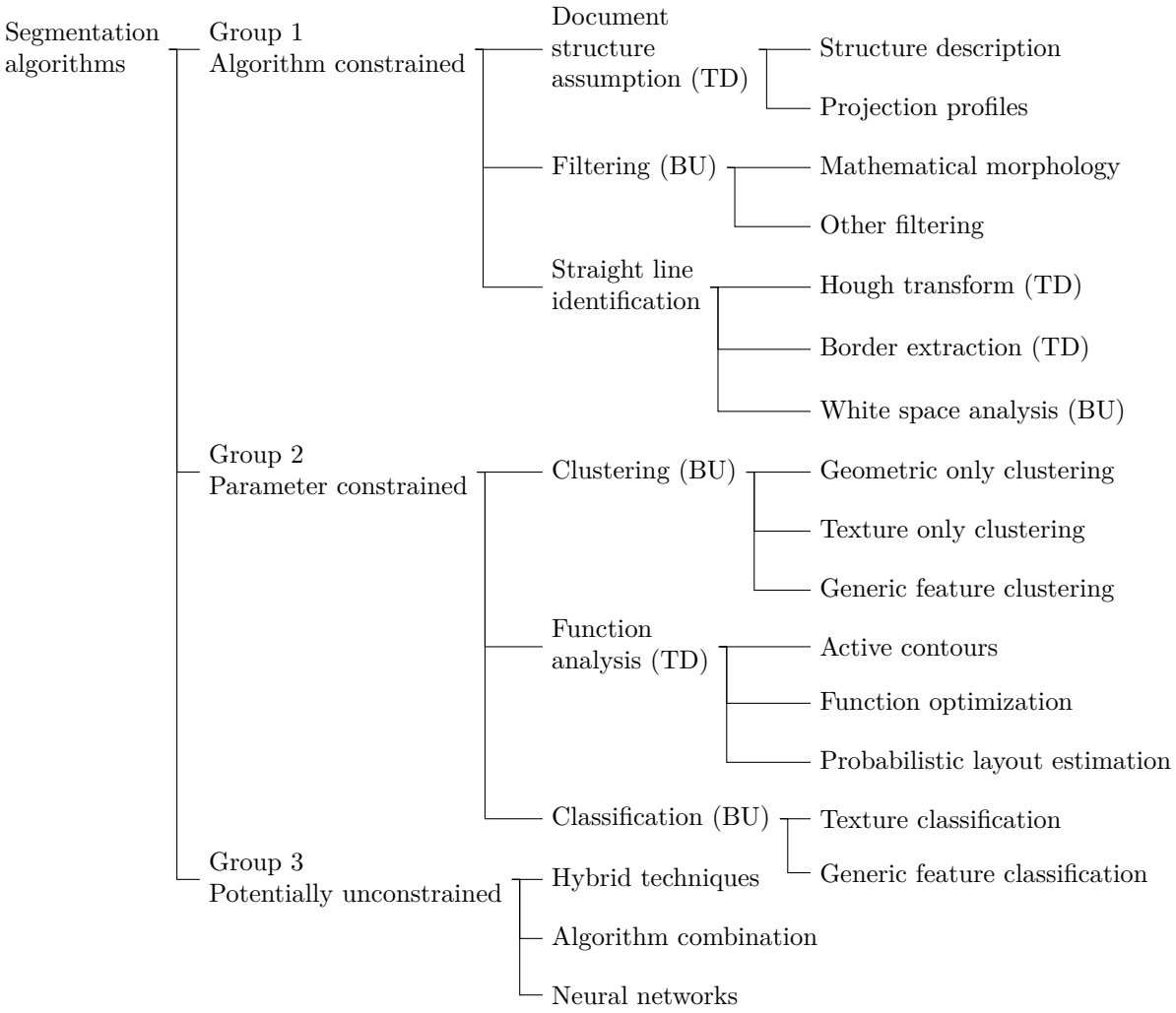


Figure 4.2.1: Typology of document segmentation algorithms. We also specify top-down (TD) and bottom-up (BU) algorithms.

- Clustering algorithms try to cluster elements based on geometric or texture or a more general set of features.

- The algorithms based on function analysis rely on function optimization e.g. trying to bring a function as close as possible to an objective value. Some specific cases are active contours, energy minimization and probabilistic layout estimation. An interesting fact about these algorithms is that while most algorithms work with the region areas, those based on function analysis usually work with the region boundaries.

- Classification algorithms are trained to recognize the different types of elements based on a given set of features (purely texture or more generic features). Thus they need training and produce labeled elements.

The algorithms in group 3

They try to overcome the limitations of the other algorithms by combining them or by using artificial intelligence. There are three subcategories in this group:

- Hybrid algorithms combine several other algorithms in symbiosis. While they could potentially accumulate the strength of several other algorithms, some of them tend to be very complex without significant performance or versatility improvement.
- The combination algorithm (we have only found one) combines the results of several algorithms to effectively improve them.
- Neural network algorithms make use of artificial intelligence to automatically learn significant features and perform the required task. They need careful design and are subject to over training. They also tend to be a "black box" whose functioning is not easily explained.

The limits and capabilities of these algorithms will be detailed in Section 4.3.

4.2.2 Layout constrained by the algorithm

The algorithms in this group are such that their internal mechanisms limit the variety of layouts that they can segment.

4.2.2.1 Segmentation based on document structure assumption

These algorithms are the most limited ones. They are made for a very specific type of layout hence they are only applicable to documents with a structured layout. This drawback is counterbalanced by their success rate in segmenting this specific layout in comparison with other more flexible techniques. They are also extremely fast.

Grammar

There are five algorithms of this type [LCC08a, LCC08b, SBKB08, SvBKB08, CLC15]. They were first published in 2006 by Coüasnon [Coü06] and tested on a dataset of 88745 documents which is an unrivaled dataset size. He designed a layout grammar language called DMOS. This language can describe any layout and

the associated parser recognizes this layout in an image. The grammar also allows the association of a label to each region of the layout thus producing a labeled segmentation. Lemaitre et al. improved it in 2008 [LCC08b] by adding a multi-resolution approach which made it flexible enough to segment handwritten letters (provided that their layout still obeyed certain rules) and to identify text lines in administrative documents in French and Bangla. Carton et al. [CLC15] continued this work with an interactive training step capable of creating automatically an exhaustive set of models for a large dataset.

Shafait et al. [SBKB08], proposed another grammar algorithm based on a probabilistic layout formulation. The user defines a set of cuts whose position is defined approximately. Then for each image a probabilistic fitting is performed to obtain the appropriate regions. This algorithm is capable of segmenting tight layouts with very small margins. Its results are not compared with those of DMOS. The paper only considers horizontal or vertical cuts which limits it to Manhattan layouts. More generally, the grammar flexibility can be a limitation to a grammar-based segmentation algorithm.

Projection profiles

These algorithms stem from the original X-Y cut algorithm of Nagy et al. [NSV92]. There are 7 algorithms of this type [OB08, LZY10, OBA10, PSKC10, OB12] and the two CASIA algorithms published in [MRHR15]. Ouwayed and Belaïd [OB12] use projection profiles to segment multi-oriented, text only documents. They make a paving of the document with rectangles. Then they compute the projection profile of each rectangle along several directions. The direction with the highest maximum of Wigner-Ville distribution is that of the text. Then, they use heuristics combined with local projection profiles to detect regions with non homogenous text orientation and text lines. They are also capable of segmenting curved text lines. Another feature of the algorithm is the capability to separate intricate text lines, but this costs the generality of the algorithm as it requires typographic heuristics specific to Arabic handwriting.

Liu and al. [LZY10] use [OB12] to segment Manhattan layouts. After a binarization that replaces text lines by black regions, they use projection profiles to remove border noise and detect text columns. The interest of the algorithm lies in the noise removal process. It first classifies the text lines into two levels of confidence. Then based on the most confident ones, it computes its internal feature parameters on the fly. These features help discriminate the lines as noise or text lines. The algorithm improves significantly the state of the art and is evaluated on 1922 documents containing four representative languages: Arabic, English, Chinese and Yiddish.

4.2.2.2 Segmentation based on filtering algorithms

These algorithms usually use specific predefined filters to segment a given type of content. They frequently rely on assumptions that the text lines are straight and/or horizontal.

Mathematical morphology

Six algorithms using mathematical morphology have been published [BCM11, BSB11a, BSB11b, LCC11, TWB14, FBEB09]. Bockholt et al. [BCM11] use several combinations of erosion and dilation to efficiently identify successively the pictures, the graphics and the text. While being a basic type of processing it proves very efficient for the task of document retrieval.

Ferili et al. [FBEB09] replace the logical AND of RLSA by an OR. This makes the algorithm more computationally efficient as only one run length is performed. They make the assumption that the text is horizontal. An interesting addition is the extension of the algorithm to natively digital documents based on their basic blocks.

Buckhari et al. [BSB11b] use Bloomberg's segmentation algorithm [Blo91] which is based on mathematical morphology. They add a first step to merge broken horizontal and vertical lines with a hit-miss morphological transform and a second step to fill holes.

Other filtering

The only contribution for this kind of algorithm was done by Shi et al. [SSG09]. It was reused by A2iA in a competition [MRHR15] where it was ranked third out of 7 participants. The method is based on steerable filters (filters that can be rotated) to detect text lines along five orientations. A heuristic post processing is used to solve the issue of connected components spanning several lines.

4.2.2.3 Segmentation based on straight line identification algorithms

Three contributions are based on straight line identification [LGPH09, CYL13, WZT15]. Louloudis et al. [LGPH09] split the connected components horizontally into blocks based on the average character height. This allows the algorithm to work on handwritten text where several characters are merged into one connected component. However the horizontal partitioning assumes that the text is also horizontal. Once this partitioning is done, they apply a Hough transform on the centers of gravity of each block to detect text lines.

Wang et al. [WZT15] attempt to reconstruct the border of the frames in comic books in order to segment them. Their algorithm is able to segment frames with

only two apparent borders but is limited to quadrangle regions. They separate the background, then they use the Douglas-Peucker algorithm [DP73] to fit quadrangles onto the candidate frames. This is followed by a classification of the frame complexity and specific heuristics are used to complete the frame border. This algorithm only improves the state of the art for difficult to very difficult layouts which is its original goal.

Chen et al. [CYL13] analyze the white spaces to segment the document into text columns. The connected components are grouped into horizontal chains to create white spaces between these chains. Then the white spaces are grouped vertically to make white lines/column separators. The algorithm works well to detect text but not for graphical parts. It won the two segmentation competitions of ICDAR 2013 [ACPP13b, ACPP13a].

4.2.3 Layout constrained by the parameters

This group contains the majority of algorithms that have been published. They remain fairly simple while being flexible enough to address a wide range of problems.

4.2.3.1 Segmentation based on clustering

This is clearly the most popular type of algorithm with 40 algorithms.

Geometric only clustering

The vast majority (31 publications) of clustering algorithms uses only geometric features. Before further classifying them based on the color information level that they can process, we can highlight the contribution of the Fraunhofer Institute and the team of Konya et al. who made several contributions to the field and participated in every document segmentation competition and won two of them [APBP09, ACPP11, Kon12, ACPP13b, ACPP13a, ACPP15].

Black and white geometric clustering algorithms

Most of them only process black and white images [APBP09, ACPP11, ACPP13b, CAP12, DKS13, FV09, GELE08, Kon12, KKDAA11, LLS11, LLG+11, LFJ08, LLS14b, LLS14a, MEE+09, OLT10, OLT+13, RTBO13, RPL12, YL09b, WAS11]. Liu et al. [LFJ08] use a Gaussian Mixture model to classify connected component triplets as text or non text. They use three geometric features (distance, area, density) and thus have trivariate Gaussian distributions. The first order neighborhood of a connected component is computed with the Delaunay triangulation and they use the second order neighbors to obtain all the possible triplets. They

also use a specific training called MMS which maximizes the class separability. Although the algorithm is not made for color images, it is tested on binarized color advertisements and magazine cover pages. It performs well with a precision and recall over 90%.

Agrawal and Doermann improved the original Voronoi algorithm [KSI98] with Voronoi++ which adapts the Voronoi parameters to the local spatial context [AD09]. Then they made a fuzzy version of it (with fuzzy edges) called CVS [AD10]. It formulates hypothetical regions and then validates them. The validation phase is done based on a distance and similarity (texture) contexts. This was evaluated on 350 documents with 5 evaluation schemes and consistently outperforms Voronoi and Voronoi++.

Gaceb et al. [GELE08] use a custom binarization optimized for fast processing. They take a very novel stand in trying not to group dissimilar connected components. They do this with a graph coloring technique where the dissimilarity constraint is reflected by that of two adjacent (dissimilar) nodes (connected components) having a different color. The connected components having the same color are the text lines of the documents. They benchmark their algorithm on 10000 envelopes and it outperforms RLSA and X-Y cut while providing a significant speedup.

Yin and Liu [YL09b] use metric learning based on geometric features to compute the minimum spanning tree between the connected components of the binary image. A post processing is then applied to obtain the final text-lines.

Faure and Vincent [FV09] use geometric clustering to segment horizontal and vertical text plus technical drawings in historical documents. The interesting contribution they have is the use of a confidence value for each alignment (text line) and a conflict resolution post processing when there is an inconsistency between two text lines.

Olivera et al. [OLT+13] improve their parallel line regression algorithm [OLT10] by creating queues of horizontal and vertical neighbors of every connected component. They are processed by decreasing order of queue length and the parallel line regression clustering is applied on each queue. The parallel regression is based on geometric heuristics deduced from the typographic rules of six different fonts. This improvement allows them to significantly reduce the over segmentations of their previous algorithms and actually improves the state of the art.

Liu et al. made two contributions in the perspective of near-duplicate document image matching. In the first one [LLS14b] they index a document with a set of features among which is the distribution of distances between the segmented components of the document. This segmentation is performed by grouping the connected components based on a distance threshold. This was successfully tested on more than 24000 images. In [LLS14a] they build a component hierarchical tree

from a basic segmentation and then build every possible segmentation across all the tree levels. They consistently outperform the state of the art on 1425 modern and historical documents except for 20 text only documents.

Other geometric clustering algorithms

Out of the five algorithms that make use of more information [ACPP13b, OLL13, ACPP15, ZENM13, CK09], two stand out. Ouji et al. [OLL13] have a versatile algorithm based on a deep understanding of how a document is created. They identify color and non color regions in a document image. Then they separate the non color regions into binary (text) and gray-level (illustration) regions. The color regions are similarly separated into monochromatic and polychromatic ones based on a multi-level analysis. They outperform the state of the art on 448 challenging documents from magazine inside pages to advertisements with text overlapping natural images.

Clavelli and Karatzas [CK09] segment propaganda posters that have nearly uniform colors. They make use of this property to segment the image components with a pixel clustering based on pixel neighborhood and RGB color distance. Then they define a search region around each component in order to group them into text lines. This allows them to identify very curved text lines with letters of different colors with a varying background.

Texture only clustering

Three algorithms use only texture features [JRME08, MGKH$^+$13, MHGK$^+$13]. Journet et al. [JRME08] use features at pixel level and tested them on both modern and historical documents. They highlight the importance of a multi-resolution approach to reduce the noise in pixel clustering techniques. Working at pixel level allows the clustering of many different types of objects such as drop caps, a specific kind of graphic, text, text fonts, etc.

Mehri et al [MGKH$^+$13] demonstrate that Gabor texture features outperform auto-correlation and co-occurrence texture features for historical documents.

Generic feature clustering

Six algorithms use several types of features [CBC$^+$15, KAA10, ZF10, CW09, AD09, AD10]. The most outstanding one is that of Chen and Wu [CW09]. Roughly, they cut the document into blocks which are then multi-thresholded to create several layers. The connected components of each layer are identified and grouped across blocks based on a predefined set of features. The evaluation dataset is small (65 documents) but very challenging as it contains only magazine covers and advertisements that are multi-layered color documents with uneven background.

They outperform significantly the state of the art and achieve both precision and recall above 99% for text extraction.

Carel et al. [CBC+15] uses a multi-resolution color and spatial clustering to identify the color layers contained in a document and the connected components in each layer.

4.2.3.2 Segmentation based on function analysis

Sixteen algorithms rely on function analysis techniques. They have the advantage that, based on the "flexibility" of the functions, one can select how much they will follow the contours of the elements to segment. This can be helpful if we want to have a rough outline of the document regions or if we want to segment precise elements such as warped text lines.

Active contours

All these techniques were proposed by Bukhari et al. [BSB08, BSB09a, BSB11c, BSB13b, BSB13a, BSB09c, BSB09b]. Their work can be considered as the state of the art for text line extraction based on active contours. It works by adding coupled snakelets (a kind of non closed active contour) on the top and bottom of a connected component and by deforming them based on the vertical component of the gradient vector flow. The snakelets are then extended laterally in order to include neighboring connected components. This algorithm has been evaluated on 10 different scripts in [BSB13b].

Function optimization

Seven algorithms use function optimization [RKC14, KO11, SLDZ09, DPB08, KC10, ACPP15, MRHR15]. They usually define a cost or energy function which needs to be minimized. So far, they work best for text line segmentation although the ISPL method was second in the last ICDAR document segmentation competition [ACPP15]. The state of the art in this field is the algorithm of Ryu et al. [RKC14] which also won the ICDAR 2013 Competition for handwriting segmentation [SGL+13] and 2015 Competition on text line detection [MRHR15]. Their contribution resides in over-segmenting connected components that do not fit a normalization criterion. From this they obtain a better estimation of the belonging of each connected component to a given text line which in turn allows them to build a better cost function. The optimization of this function is improved with dynamic programming. The over segmented components are then merged into proper components.

Shen et al. [SLDZ09] use both intra- and interline metrics to build a segmentation cost function. After an initial geometric clustering they perform a simulated

annealing optimization. This means that the probability of accepting a new segmentation that increases the cost function is not null but decreases with each iteration. Combined with a custom binarization, this allows them to extract text line from challenging color documents with non uniform background (CD covers) or unusual layout (business cards).

Kim and Oh [KO11] highlight the interest of using interline information over intra-line information for Asian scripts.

Probabilistic layout estimation

Two algorithms make a probabilistic estimation of the layout [YL09a, CT14] but do not bring a significant improvement. Yin and Liu [YL09a] perform an estimation of the number of text lines with a blur filter and then use a variational Bayes approach to segment the image rescaled at 75 dpi. This improves slightly the state of the art on a large but not very challenging dataset.

Cruz and Terrades [CT14] proposed a method based on Conditional Random Field (CRF) and location features similar to [FT12] (see section 4.2.3.3) but without any improvement over the state of the art. This work is at the crossing between optimizing a probabilistic layout estimation and a classification.

4.2.3.3 Segmentation based on classification

This is the second most popular type of algorithms with 30 algorithms. A noticeable difference in the scientific work when compared with the clustering is the fact that classification algorithms all require training.

Texture classification

Three algorithms use only texture features [APBP09, BKMA10, BI11]. Baechler and Ingold [BI11] string together three Dynamic Multi-Layer Perceptrons (DMLP) at three resolutions to segment historical documents. Each DMLP uses the label output of the DMLP at a lower resolution plus texture features at its resolution. Each level processes only part of the labels produced by the lower level in order to refine these specific labels.

Generic feature classification

Most classification algorithms (24 out of 27) use a generic set of features.

Black and white generic feature classification algorithms

Among them, 12 use only binary images [PSG+09, SSL09, BASB10, PB11, PSGR12, BMA14, LNV+15, YM08, APBP09, HPN11, FT12, PSGS13]. Peng [PSGS13] and Pinson [PB11] focus on extracting overlapping handwritten and typewritten text. Pinson and Barret's algorithm [PB11] automatically selects the appropriate features based on the desired typewritten text classification accuracy. It selects the first 100 feature vectors of a Principal Component Analysis (PCA) of the character images. Then any new text is projected into this new space. If it is close enough to the typewritten or the handwritten templates then it is classified appropriately. Otherwise it is considered as made of several touching characters and split with a graph-cut thus making two new connected components to classify. They achieve 98% precision for typewritten text and 71% precision for handwritten text on 500 forms each coming from a different writer. The low handwritten precision is related to the training set that did not contain typewritten text of a small size. As a result small typewritten fonts were classified as handwritten. Replacing the small font size in the test set brings the precisions to 94% and 89% respectively. This highlights the limits of the training set and the versatility of the algorithm with respect to the writing style.

Peng et al. [PSGS13] work at connected component and patch level. Patches are found with a morphological closing. They use a first Markov Random Field (MRF) to classify the patches into typewritten, handwritten or overlapped text. Then, they use another MRF to reclassify them based on their context. The overlapped text is separated at a pixel level with a third MRF and by using Shape Context Features (SCF) [BMP02]. It performs slightly worse than Pinson, but the dataset size (28 documents) hinders the significance of this performance.

Bukhari [BASB10] focus on extracting text from documents that contain graphic illustrations such as circuit drawings. The challenge here is to identify correctly text and graphics. While the challenge seems easier than separating overlapped typewritten and handwritten text, many segmentation algorithms do not handle graphics well. Thus targeting this specific issue is a good contribution to the state of the art. To do this, they rescale every connected component to a predefined size and do the same with a wider image centered on the connected component in order to capture its context. Then they use a Multi-Layer Perceptron (MLP) to do the classification. The recall for text and non-text (graphic) is consistently above 93 and 96% respectively on 100 documents.

Fern and Terrades [FT12] focus on extracting the regions of structured documents. They use Gabor (texture) features with a CRF. Their contribution lies in the addition of relative location features which are the probability of a region being of a certain class given its position relatively to the regions of the other classes. These features (one per class and per region) have a significant impact for

segmenting structured documents. The improvement remains less significantly on non structured documents.

Gray-level generic feature classification algorithms

Four algorithms use gray level information [ZC15, MGHN09, DKS11, GSD11]. Diem [DKS11] tackles the challenge of segmenting document fragments. After extracting candidate word blobs with projection profiles, they introduce Gradient Shape Features (GSF) to refine the segmentation and classify the text as handwritten or typewritten with a support vector machine (SVM). GSF are computed on a sliding window scaled to the size of the word blob. For a given window they are similar to shape context features applied on the inverted gradient image instead of the original image. Then they perform a geometric clustering of the word blobs into lines. A final global voting with another SVM classifies the candidate lines into typewritten or handwritten again. They include an error back-propagation to relabel the word blobs that were mistakenly labeled. They improve the state of the art for graphic classification while maintaining a similar performance on the text in ICDAR 2009 segmentation competition [APBP09].

Zhong and Cheriet [ZC15] devise a new tensor-based learning algorithm and apply it successfully to classify text and non text (borders, noise, background, etc.) on text only ancient manuscripts.

Color generic feature classification algorithms

Eight feature classification algorithms use color information [WBA09, BLI13, WBSI13, CWL$^+$14, FBG$^+$14, CSL$^+$15, MRHR15, GHCG11]. Garg et al. [GHCG11] separate text and graphics in challenging magazine covers. They use an SVM to classify Gabor and edge features followed by a CRF to include the local spatial context. The CRF improves the performance by 2%.

Wei et al. [WBSI13] compare the performance of SVM, MLP and GMM (Gaussian mixture model) classifiers. They find that SVM and MLP outperform GMM but cannot conclude which one is best. This depends on the data. The dependency on the features is not studied.

Wang et al. [WBA09] make a very interesting contribution by finding a way to automatically discover features to improve the performance of a 2-nearest neighbor classifier. Given a large set of features, they sample the feature space. From this sample they find a cluster of errors and project it into the quotient space with the already discovered features. Then they compute a new feature in this hyperspace with a linear combination of the already existing features.

We finish this section with an algorithm on the border between classification and neural networks (which belongs to the third group). Chen et al. [CSL$^+$15] use

convolutional autoencoders (a kind of neural network) to automatically discover distinctive features on image patches at three scales and train an SVM with these features. In their evaluation they demonstrate the usefulness of combining these features and their superiority to handcrafted features.

4.2.4 Layout potentially unconstrained

This last group contains eleven algorithms with a majority of them published since 2014. These algorithms try to overcome the shortcomings of the others by hybridizing them, combining them or with advanced neural networks.

4.2.4.1 Segmentation based on hybrid techniques

A large majority of them make several techniques work in symbiosis to obtain better results [APBP09, Smi09, BACP14, ACPP15, ACKEs15, MRHR15, WFSN15]. The most recent significant work was the MHS method developed by Tran et al. which won the last ICDAR complex document segmentation competition [ACPP15]. It works by iteratively classifying connected components based on multi-level homogenous regions and white space analysis.

Another significant contribution is the one by Barlas et al. [BACP14] which can segment an extremely diverse and complex range of documents in several languages. They generate a feature codebook with self-organizing maps which serves to describe the training set and then train an MLP. Once they have obtained the class layers, they create regions by combining RLSA and white space analysis. They tested it on 1000 documents from the Maurdor dataset with three classes (graphics, typewritten and handwritten text) and won the evaluation campaign.

Wang et al. [WFSN15] propose another algorithm for extracting text lines from complex documents with multi-oriented text in several languages. They use MSER to extract candidate characters which are then classified as text or non text with a fast Adaboost classifier. The low confidence text is further evaluated with a Convolutional Neural Network (CNN). The next step is a coarse line extraction with geometrical grouping based on a linearity constraint. This graph is refined by a minimum spanning tree. The last step is an energy minimization refinement of the lines.

4.2.4.2 Segmentation based on algorithm combination

The only contribution to this type of algorithm was done by Stamatopoulos et al. [SGP09]. They devise a procedure to significantly improve the performance of individual segmentation algorithms by combining their results. The procedure is based on the overlap of the regions produced by the algorithms. They are

considered as good above 90% overlap. They use the regions that have more than 70% to compute the values of a task based set of features. All regions below 90% undergo a splitting based on their intersection followed by a merging starting from the regions with the highest overlap. This is successfully applied to text line segmentation and improve single algorithm results by 15 to 25% depending on the metric.

4.2.4.3 Segmentation based on neural networks

Two algorithms use neural networks [MKW15, MRHR15]. The A2iA1 in the text line segmentation competition [MRHR15] did not win it and it is similar to that of Moysset et al. [MKW15] which seems to have improved it. They normalize the width of a text column (or of a text only single column document) and use a bidirectional long-short term memory neural network (BLSTM) to detect text lines and paragraphs. They tackle the issue of modeling gaps and interlines and conclude that each should have its own class. They also show that using specialized neural networks trained on a specific set is better than using a single system trained on a more varied dataset.

4.3 Limits and capabilities of document segmentation algorithms

After this review of document image segmentation algorithms it may be of use to summarize their drawbacks and capabilities. We will first make some generic remarks and then summarize the capabilities and limitations of the main algorithms that were surveyed. A study of current evaluation practices and of the trends of the community can be found in Appendix A.

4.3.1 Remarks on the applicability of the surveyed algorithms

When choosing a segmentation algorithm from the above survey, there are a few important remarks to have in mind as they may impact its suitability for a given task. Here is a disparate list of them.

Text related remarks

One may think that being able to segment curved text is better than only straight lines. While this is generally true, it can also lead to merging straight lines that are close to each other and only separated by their orientation.

Many text line segmentation algorithms assume that the document contains only text and should be complemented with another algorithm capable of dealing with non textual content. Classification techniques provide labeled regions but often cannot create a text line level segmentation. If one needs this level of segmentation, they should either use another algorithm or add a text-line segmentation algorithm.

As a general rule, all segmentation algorithms are script independent e.g. their performance does not depend on the language of the document. Yet they frequently make the assumptions that a language is made of disjoint characters and that each character is made of one connected component. These assumptions are not true for handwritten text and for several languages. Thus the script independence of an algorithm may be limited to scripts that satisfy one or both of the above assumptions.

General remarks

Many algorithms working on binary images assume that the image is the result of a thresholding process similar to Otsu binarization. They usually do not work on binarization techniques that use error diffusion (dithering). Admittedly very few document analysis works deal with such images. Yet, in this case, the image should first be converted to gray level with inverse dithering and then appropriately binarized. Figure 4.3.1 shows the difference between these two types of binarization. Also, most analyzes based on connected components require binary images.

(a) Binarization with thresholding (b) Binarization with error diffusion

Figure 4.3.1: Comparison of thresholding and error diffusion binarization techniques.

Regarding the processing speed and independently from the implementations that can drastically impact it, bottom-up algorithms are usually slower than top down algorithms and the smaller their processing level, the slower they are. This is illustrated by the fact that there are many more pixels than connected components in an image. The more items there are to process the longer the processing will be. Of course, this also depends on the computational cost of generating the high level processing view such as connected component identification.

Although most papers specify parameters in pixels it is better to use the resolution of the test dataset to convert these parameters in absolute units such as

millimeters. This will make the algorithm resolution independent and more versatile.

Not all algorithms absolutely require a training dataset. However, they can have many parameters which may require training.

Finally, one should keep in mind that an algorithm that has been trained or tuned on a given dataset is only supposed to segment correctly similar documents. While it may work for other kinds of documents the training dataset is usually a limitation to the segmenting capability of the algorithms. This fact is clearly shown in the scientific papers by the difference of size between the training and testing sets. The training set is often several times larger than the testing set.

4.3.2 Functional point of view

This survey reviewed the algorithms from a scientific stand point. We would now like to summarize them from a user/functionality perspective. This would be the place for a qualitative analysis of the algorithms in order to identify which ones may be the most suitable for a given task. However the lack of cross-technique comparison and of diversity in many datasets prevents us from doing such an analysis. Furthermore, the performance differences of the algorithms between experimental and real data would make such an analysis irrelevant.

Instead, we try to answer two questions related to this: what do they do (functionality) and what do they require to do it (requirements)? Table 4.1 summarizes this for the main algorithms of this survey. The kind of documents that an algorithm can take as input (layout, multi-layered, color depth, text orientation and alignment) is both a functionality and a requirement. A constraining requirement is the need for training. The output (type of output and labels) produced by the algorithm is a functionality. We ordered them from what seemed to be the most to the least critical from a user/industrial point of view. We added the last three columns (dataset size, number of languages and of document types) as a guidance to estimate how extensively the algorithms have been tested and thus how reliable they are. The algorithm in bold is one of the algorithms that we will use in our benchmark.

Table 4.1: Summary of the characteristics of the main document segmentation algorithms. The type of input is the kind of document that can be processed. Multi-layered indicates whether it can process documents with overlapping contents.

Algorithm	Input layout	Multi-layered	Color depth	Labels	Training	Type of output	Text orientation	Text alignment	Dataset test size	Nb of languages	Nb of doc types
[OLL13]	Any	Yes	Color	Yes	No	Text lines	Horizontal	Straight	448	1	2
[WBA09]	Any	Yes	Color	Yes	Yes	Regions	Any	Curved	87	1	2
[GHCG11]	Any	Yes	Color	Yes	Yes	Regions	Any	Curved	16	2	1
[CBC+15]	Any	Yes	Color	No	No	Regions	Any	Curved	2000	2	2
[CK09]	Any	Yes	Color	No	No	Text lines	Any	Curved	50	1	1
[SLDZ09]	Any	Yes	Color	No	Yes	Text lines	Any	Straight	21	2	1
[CW09]	Any	Yes	Gray	Yes	No	Regions	Horizontal	Straight	65	2	1
[BACP14]	Any	Yes	BW	Yes	Yes	Regions	Horizontal	Straight	1000	3	6
[BI11]	Any	No	Color	Yes	Yes	Regions	Any	Curved	100	1	1
[CSL+15]	Any	No	Color	Yes	Yes	Regions	Any	Curved	49	3	2
[WBSI13]	Any	No	Color	Yes	Yes	Regions	Any	Curved	49	3	2
[WFSN15]	Any	No	Color	No	Yes	Text lines	Any	Curved	214	3	3
[MKW15]	Any	No	Color	No	Yes	Text lines	Horizontal	Straight	1072	3	6
[ZC15]	Any	No	Gray	Yes	Yes	Regions	Any	Curved	86	3	1
[DKS11]	Any	No	Gray	Yes	Yes	Regions + text lines	Any	Curved	501	1	2
[JRME08]	Any	No	Gray	No	No	Regions	Any	Curved	400	1	2
[MGKH+13]	Any	No	Gray	No	No	Regions	Any	Curved	25	1	1
[ACPP15] (MHS)	Any	No	BW	Yes	No	Regions	Horizontal	Straight	70	1	1
[BCM11]	Any	No	BW	Yes	No	Regions	Horizontal	Straight	3742	2	1
[Kon12]	Any	No	BW	Yes	No	Regions + text lines	Horizontal	Straight	185	2	2
[FT12]	Any	No	BW	Yes	Yes	Regions	Any	Curved	75	2	2
[AD10]	Any	No	BW	No	No	Regions	Any	Curved	350	2	3
[LLS14b]	Any	No	BW	No	No	Regions	Any	Curved	24339	2	2
[FBEB09]	Any	No	BW	No	No	Regions	Any	Straight	100	1	1
[CYL13]	Any	No	BW	Yes	No	Regions + text lines	Horizontal	Straight	55	1	1
[FV09]	Any	No	BW	No	No	Text lines	Any	Straight	52	1	1
[KO11]	Any	No	BW	No	No	Text lines	Any	Straight	95	4	1
[BSB13b]	Any	No	BW	No	No	Text lines	Horizontal	Curved	649	10	2

Continued on next page

Table 4.1 – continued from previous page

Algorithm	Input layout	Multi-layered	Color depth	Labels	Training	Type of output	Text orientation	Text alignment	Dataset test size	Nb of languages	Nb of doc types
[LLS14a]	Any	No	BW	No	Yes	Regions	Any	Curved	1425	2	3
[LCC08b]	Structured	No	BW	Yes	No	Regions	Any	Curved	100	2	2
[GELE08]	Structured	No	BW	No	No	Regions	Horizontal	Straight	10000	1	1
[SBKB08]	Structured	No	BW	No	Yes	Regions	Any	Curved	260	1	1
[PB11]	Text only	Yes	BW	Yes	Yes	Regions	Any	Curved	500	1	1
[SSG09]	Text only	No	Gray	No	No	Text lines	Horizontal	Straight	45	1	1
[PSGS13]	Text only	No	BW	Yes	Yes	Regions	Any	Curved	28	1	1
[OB12]	Text only	No	BW	No	No	Text lines	Any	Curved	100	1	1
[YL09a]	Text only	No	BW	No	No	Text lines	Any	Straight	853	1	1
[SGP09]	Text only	No	BW	No	No	Text lines	Horizontal	Straight	50	2	1
[OLT+13]	Text only	No	BW	No	No	Text lines	Skewed	Curved	202	1	2
[LGPH09]	Text only	No	BW	No	No	Text lines	Skewed	Straight	120	2	1
[RKC14]	Text only	No	BW	No	Yes	Text lines	Horizontal	Straight	150	3	1

4.4 Evaluation of the stability of the state of the art

Evaluations of document segmentation algorithms are regularly published either as dedicated papers [SKB08, SLG15] or as competition reports [ACPP13b, ACPP13a, ACPP15, APBP09, AGB07]. These papers have investigated how to evaluate a single segmentation result based on a given ground truth. A corollary is the question of the quality, relevance and bias of the ground truth [SLG15]. For example, a line-wise ground truth will favor line segmentation algorithms over paragraphs segmentation algorithms. Yet, both kinds of algorithms could actually be of nearly equal use and of equal quality.

Regarding the comparison of several results with each other, i.e. stability, to our knowledge, no work has studied it for image segmentation algorithms. This is the topic of this section.

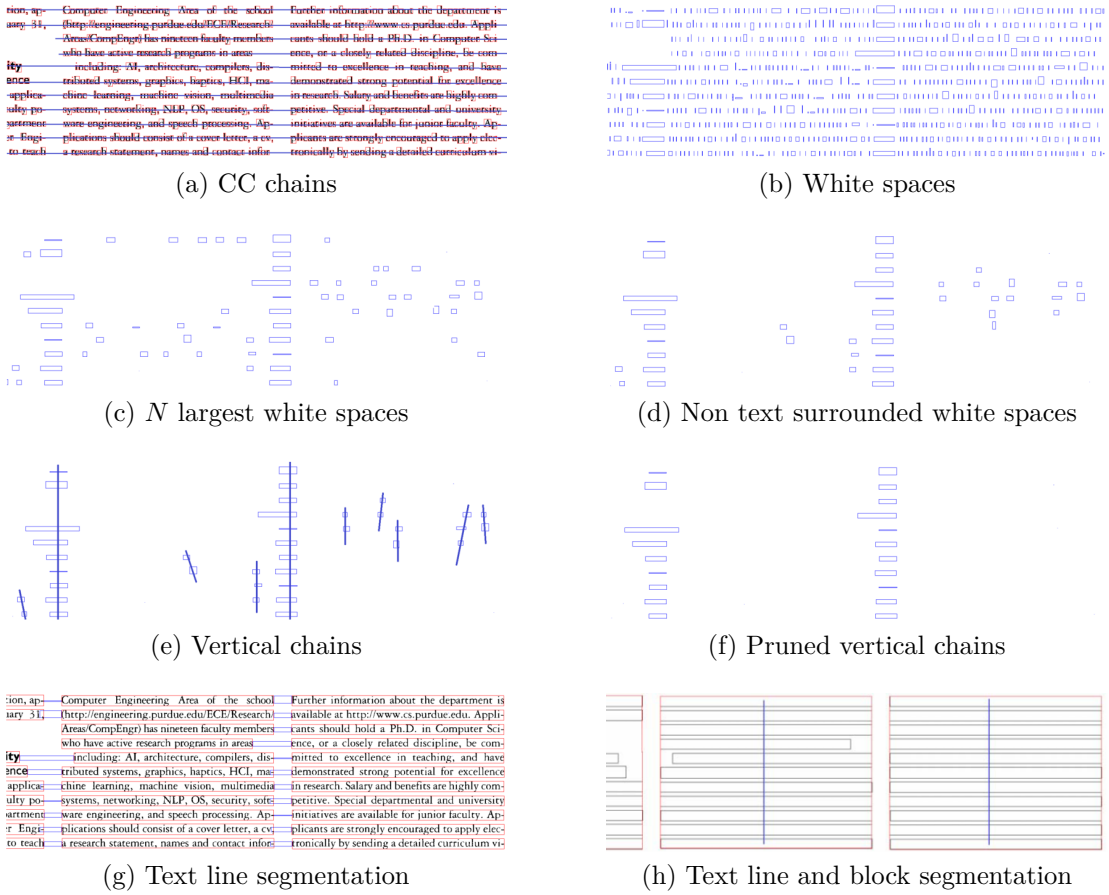
We will first detail the panel of algorithms that are evaluated and then the dataset used to evaluate them. The next subsections deal with the performance indicators and the process used to evaluate the algorithms. Finally, Section 4.4.5 presents the results of the benchmark.

4.4.1 Algorithm panel

As we have seen lots of document image segmentation algorithms have been published. Unfortunately many target specific documents or use cases, others are compared only with similar algorithms and a lot of them are tested on specific datasets. Hence, for lack of a better selection method, we decided to choose the algorithms that performed best either in competitions or in generic benchmarks with existing performance indicators: PAL and Voronoi. We also added a natural scene image segmentation algorithm, JSEG, in order to study the suitability of these algorithms for document image segmentation. To our knowledge no one has used natural scene image segmentation algorithms on document images and no one has proved that they do not work on these images. This is the occasion of having a first idea of their suitability for document image segmentation.

PAL segmentation algorithm

We selected the PAL [CYL13] algorithm which won the two ICDAR 2013 competitions on historical document layout analysis [ACPP13a, ACPP13b]. Considering our typology, it is a "Group 1/Straight line identification/White space analysis" algorithm. The algorithm first extracts the connected components (CC) from a binary image. The CCs larger than one tenth of the page width are kept aside for later processing. Then it processes the CCs from left to right and searches the nearest right neighbor for each of them. This makes line chains (Figure 4.4.1a).



(a) CC chains (b) White spaces

(c) N largest white spaces (d) Non text surrounded white spaces

(e) Vertical chains (f) Pruned vertical chains

(g) Text line segmentation (h) Text line and block segmentation

Figure 4.4.1: Segmentation process of PALL and PALB algorithms. Images reproduced from [CYL13].

The spaces between the chain CCs are white spaces (Figure 4.4.1b). Then for each chain, only the N largest white spaces are kept (Figure 4.4.1c). N is computed with the following formula:

$$N = \max(1, 8 \times L/page\_width) \tag{4.4.1}$$

where L is the chain length. The removal of a white space means that its adjacent CCs are merged into one line. The white spaces are further pruned by eliminating all the white spaces that are entirely surrounded (left, right, above and below) by text lines (Figure 4.4.1d). The remaining white space are grouped into vertical chains based on their nearest below neighbor (Figure 4.4.1e). The white spaces that do not have the largest width of their horizontal line chain are tagged "candidate within line spaces". The vertical chains that are made only of candidate within line

spaces are removed (Figure 4.4.1f). The large CCs that were initially put aside are now integrated with the text lines if they do not surround one (frame-like CC) or if they are not too elongated (vertical or horizontal separators). The integration is done with a set of heuristics to produce the final line level segmentation (Figure 4.4.1g). Up to this result, we call the algorithm PALL.

The algorithm further groups the lines to make blocks. For a given text line the algorithm searches its direct below neighbors e.g. the neighbors directly below the text line as shown on Figure 4.4.2.

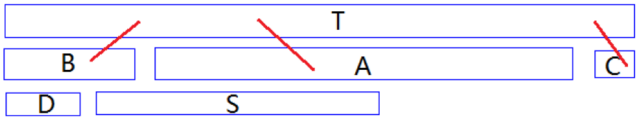


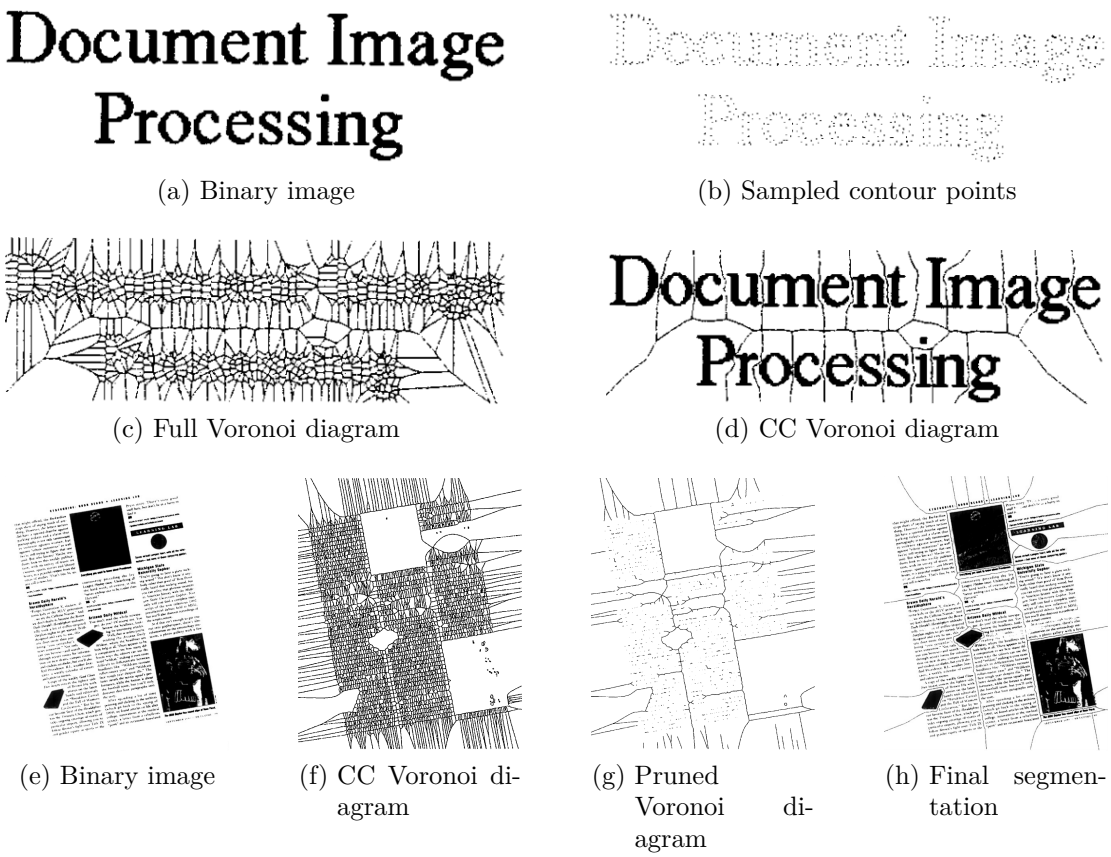
Figure 4.4.2: Below nearest neighbors of line T. A, B and C are direct below neighbors but not D and S. Image reproduced from [CYL13].

These direct below neighbors make text line chains that stop when they are the below neighbor of several text lines, or when they have several below neighbors or when they have no below neighbor (Figure 4.4.1h). We call this block level segmentation algorithm PALB.

Voronoi segmentation algorithm

Shafait et al. [SKB08] did an analysis of document segmentation algorithms and found that Voronoi segmentation [KSI98] has the best accuracy. This algorithm is in the second group of our typology under "Clustering/Geometric only clustering". The algorithm works by pruning the superfluous edges of the Voronoi diagram of a binary image. Given the binary image (Figure 4.4.3a and 4.4.3e), the contour of each CC is sampled at a rate $1/s$ where s is the first parameter of the algorithm (Figure 4.4.3b). The CCs whose border length is lower than the second parameter of the algorithm are considered as noise and removed. Then the Voronoi diagram of the sampled points is computed (Figure 4.4.3c). This diagram is the dual of the Delaunay triangulation e.g. its edges are perpendicular to those of the Delaunay triangulation. Finally the Voronoi edges that overlap the CCs are removed to make Voronoi cells surrounding each CC (Figure 4.4.3d and 4.4.3f).

The pruning of the Voronoi edges is based on two features: the minimal distance d between the CCs that they separate and the area ratio a_r between the Voronoi cells that they make. For a given edge, it is deleted if one of the following conditions



(a) Binary image

(b) Sampled contour points

(c) Full Voronoi diagram

(d) CC Voronoi diagram

(e) Binary image

(f) CC Voronoi diagram

(g) Pruned Voronoi diagram

(h) Final segmentation

Figure 4.4.3: Segmentation process of the Voronoi segmentation algorithm. Images reproduced from [KSI98].

is met:

$$d/T_{d1} < 1 \tag{4.4.2}$$

$$d/T_{d2} + a_r/T_a < 1 \tag{4.4.3}$$

The first condition removes edges between CCs that are too close such as characters belonging to the same word. The second condition removes edges between Voronoi cells whose area is too different considering the distance between their CCs. T_{d1}, T_{d2} and T_a are parameters. T_{d1} and T_{d2} depend on the resolution and font size of the document image. Thus they need to be set adaptively. Considering the distribution of the distances f, it generally has two main peaks. The first peak f_1 is the main distance between characters and the second peak f_2 is the main distance between lines. T_{d1} is the fiber of f_1 ($f(T_{d1}) = f_1$). Since some lines are separated by a distance higher than the fiber of f_2, the algorithm adds a margin

to it. T_{d2} is chosen such that:

$$f(T_{d2}) = t \times f_2 \tag{4.4.4}$$

$$T_{d2} > t_2 \text{ where } t_2 \text{ is the fiber of } f_2 \tag{4.4.5}$$

$$\forall x \in [t_2; T_{d2}[, f(x) > f(T_{d2}) \tag{4.4.6}$$

Basically T_{d2} is the first fiber of $t \times f_2$ greater than t_2 where $t \in [0; 1]$ is the third parameter of the algorithm. The fourth parameter is T_a. Because the distribution of f may contain some noise it can be smoothed by a windows whose width is controlled by the user with the fifth and last parameter of the algorithm. This pruning produces the result shown on Figure 4.4.3g.

Among the remaining edges some do not create new regions. These edges are removed to make the final segmentation (Figure 4.4.3h).

JSEG algorithm

To complete our panel we added JSEG [DM01]. While it is not a document image segmentation algorithm, it still fits our typology and is in the second group under "Clustering/Generic feature clustering". We used the implementation available on the project website[1]. JSEG is originally made for segmenting natural images and is a reference algorithm. It has the particularities of being quite robust and of using color textures as well as a multiscale analysis. All this makes it an interesting natural image segmentation algorithm. It will allow us to study the impact of color textures and multi-scale analysis on stability and, more importantly, the applicability of natural image segmentation algorithms to document image segmentation.

JSEG first quantizes the image colors. The image is smoothed then converted to CIE LUV color space (a color space similar to Lab color space). After this, the image colors are quantized into n_{col} colors. n_{col} is estimated from the image data. The color clusters closer than a distance d_{col} can be further merged if the parameter d_{col} is given to the algorithm. Otherwise the color quantization stops at n_{col} colors (Figure 4.4.4b).

For each quantized color C_n it is possible to compute the spatial standard deviation of the pixels that have this color:

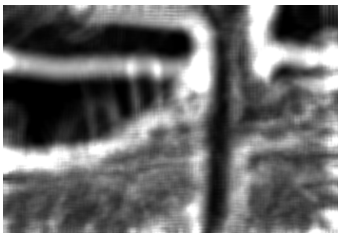
$$S_n = \sum_{p \in C_n} ||p - \overline{p_n}||_2 \tag{4.4.7}$$

where $p = (x, y)$ is the coordinate vector of a pixel having color C_n and $\overline{p_n}$ is the center of gravity of all these pixels. Similarly it is possible to compute the spatial

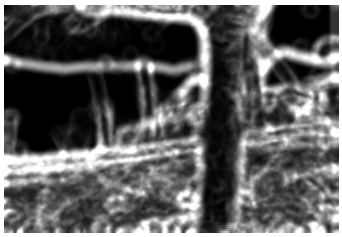
[1] `http://vision.ece.ucsb.edu/segmentation/jseg/software/`

(a) Original image $\overline{J} = 0.435$

(b) Quantized colors



(c) J-image at scale 3



(d) J-image at scale 2

(e) Segmentation at scale 3 $\overline{J} = 0.103$

(f) Segmentation at scale 2 $\overline{J} = 0.125$

(g) Final segmentation $\overline{J} = 0.088$

Figure 4.4.4: Segmentation process of the JSEG segmentation algorithm. Images reproduced from [DM01].

standard deviation for all the pixels of the image S_w. From this the J value is defined by

$$J = \frac{S_w - \sum_n S_n}{\sum_n S_n} \tag{4.4.8}$$

Both terms S_w and $\sum_n S_n$ sum the same number of elements. However the standard deviation on a subset of the image is necessarily smaller than on the whole image, hence J is always positive. The only case where $J = 0$ is if all colors have the same center of gravity which is also the one of the whole image. Then the image has a uniform spatial color distribution. The less uniform is the color spatial distribution, the higher J will be. This criteria works independently from the number of colors and thus allows uniformly textured region to have a low J value

Scanner	300 dpi	600 dpi
Konica Minolta Bizhub 223	X	XX
Konica Minolta Bizhub C364e		X
Fujitsu fi 6800	X	
Lexmark x543 PS	X	

Table 4.2: Scanning resolution for each scanner

as well as regions with a uniform color. Thus an image is considered to be properly segmented when the weighted average J value ($\overline{J}$) of its regions is minimal. The region weight is the number of pixels in the region.

The algorithm works by computing the J value on a local circular window at exponentially larger scales (9x9, 17x17, 33x33, 65x65) to make J-images (Figure 4.4.4c and 4.4.4d). Starting from the J-image at the largest scale, the local mean μ and standard deviation σ of the J value. The J-image is then thresholded with the threshold $\mu + a\sigma$ where a is preset in the algorithm. The regions that are lower than this threshold and whose size is larger than a predefined value dependent on the scale are considered as seed regions. Then the algorithm uses a region growing process using increasingly smaller scales to obtain a detailed segmentation (Figure 4.4.4e and 4.4.4f). The number of scales is a parameter of the algorithm. It also contains an undisclosed automatic scale selection. Usually choosing a scale of 3 provides good results.

The last step of the algorithm merges the previous regions based on their color histogram. The distance between two adjacent regions is the euclidean distance between their CIE LUV vector color histograms. Then it uses an agglomerative clustering algorithm similar to the one used for the color quantization. The clustering stops when no distance is lower than a threshold defined by the user (Figure 4.4.4g).

4.4.2 Testing dataset: L3iDocCopies

Our dataset is based on the PRImA dataset [APBP09] which has the advantage of having documents with a very varied content. We printed its 55 pages with 3 printers: a Lexmark x543 PS, a Canon iR Advance C9060 Pro and a Konica Minolta C5501. We then scanned each sheet of paper three times at 300 dpi and three times at 600 dpi on several scanners as shown in Table 4.2. We used the defaults settings for all the printers. Notably, the images use JPEG compression with a quality factor above 75%. Thus this dataset contains real print and scan noise as described in Section 1.2.

For each document there are 3 printers $\times$ 3 scanners $\times$ 2 resolutions = 18 copies.

Figure 4.4.5: Sample images from the L3iDocCopies dataset.

The total dataset contains $18 \times 55 = 990$ document images. Figure 4.4.5 shows some images from the dataset.

4.4.3 Performance indicators

The use of our stability performance indicators requires the definition of a similarity function in the input space (the images, s_1) and in the output space (the segmentation results, s_2). The former one (s_1) is given by the indicator of whether two images are copies of the same document. The latter (s_2) is given by the DLD computed on the image representing the boundaries of the segmented regions. We accepted an angle error of 5° and a maximum number of 2 simultaneous instabilities. This should be sufficient for a first evaluation.

We will use the performance indicators defined in Section 2.3: the false negative, positive, omission and discovery rates (FNR, FPR, FOR and FDR). Since it is likely that the algorithms will have an extremely poor performance we complement these performance indicators with the normalized standard deviation of the number of segmented regions which we define below.

Two layouts will never be identical if they do not have the same number of regions. Hence having a stable number of regions is a necessary condition to having stable results and the standard deviation of this number of regions should be as low as possible. However a standard deviation of 5 is high if there are 10

regions but low if there are 50 regions. Hence we normalize it by the number of regions produced for each document by each algorithm.

Let us consider a given algorithm and a given image I of a document from an input dataset. We list the number of regions of each segmentation result of the copies of the same document. Let n_i be the number of segmented regions in the i^{th} copy. The normalized standard deviation of the number of regions S_R for image I is computed as

$$S_R(I) = \frac{\sigma(n_i)}{\overline{n_i}} \tag{4.4.9}$$

where $\sigma(n_i)$ is the standard deviation of n_i and $\overline{n_i}$ is the average of n_i.

The S_R for the algorithm is the average of all the S_R for the images of the input dataset. In the case of cross-validation, we ignore the single copies of a document for the computation of the S_R. To be able to analyze the values of S_R in more details, we also compute the average number of regions produced by each algorithm, here denoted as $\overline{n_R}$.

4.4.4 Evaluation process

Considering the requirements of each algorithm, we resized the original 300 dpi and 600 dpi dataset (denoted HR) to 60 dpi (denoted LR) with a bicubic interpolation. This preserves the global appearance of the document images and their readibility from a distance but the text is not readable from up close. Figure 4.4.6 shows an example image at 60 dpi.

(a) Complete page

(b) Zoom on top left region

Figure 4.4.6: A document image resized at 60 dpi.

We tested all algorithms separately with high-resolution images and with low-resolutions images except for JSEG which cannot use HR images. This allows us to evaluate the influence of the resolution on the stability of the results. In order to have enough images for testing, we decided to perform a 5-fold cross-validation. The dataset is first split randomly into 5 datasets. The first dataset is used for

testing in the first fold and the rest for training. The second dataset is used for testing in the second fold and so on.

The algorithms have different parameters which call for specific training strategies in order to evaluate each algorithm in the best conditions. As a rule of thumb we decided to study the parameter values around their default value.

PAL

Both versions of PAL take a color image as input and do not use any parameter so we ran them on the complete dataset and evaluated them without training.

Voronoi

Voronoi takes binary images as input and has five parameters. We binarized the original images with Otsu's method [Ots79]. We chose the parameter sampling rates in order to achieve the best trade-off between an exhaustive sampling of the parameters and a small number of combinations to provide the best possible training within a feasible time. In particular we focused on parameter values around those used in the original paper and in [SKB08] and tried to limit ourselves to less than ten possible values per parameter.

The first parameter (the contour sampling rate s) can take an integer value above 1. The default value is 7 which means that 1 out of 7 pixels on the contours of the connected components will be selected. We consider the values between 5 and 10 included. A smaller value would result in a clear oversampling and bigger value could prevent a proper sampling of the contours of small connected components. The second parameter (contour length of a noise CC) has a default value of 20 pixels. Thus we consider values between 10 and 50 pixels by increments of 5 pixels. Because these two parameters are pixel values, they need to be adapted to the image resolution. They are related to a length at a resolution of 300 dpi so we scale them linearly with the image resolution.

The third parameter(ratio of the value of the second highest peak t) has a default value of 0.5. We make it vary between 0.2 and 0.8 by increment of 0.1. A lower value would not be selective enough and a higher value would be too selective. The fourth parameter (area ratio T_a) has a default value of 40 and we make it vary between 20 and 60 by increments of 5. The last parameter (size of the smoothing window) has a default value of 2. After performing exhaustive testing, the only relevant values are between 1 and 3.

This makes $6 \times 9 \times 7 \times 9 \times 3 = 10206$ parameter combinations. Considering the size of the dataset and the processing time, a brute force optimization would be inefficient. This is why we settled for the use of a genetic algorithm with a

population of 20 individuals, 20 generations, a mutation probability of 0.05 and a mating probability of 0.6.

JSEG

JSEG requires a small resolution color image as input (60 dpi maximum) and has three parameters. The requirement for a small resolution image is related to the convergence of the algorithm and its implementation. The color quantization threshold and the number of scales used for multiscale processing have an automatic mode which chooses the best parameter on an image per image basis. Because this is a finer grained tuning than fixing these parameters for the whole dataset, we used the automatic mode for them. The last parameter is the threshold to merge regions which can take a value between 0 (all regions are merged) and 1 (no regions are merged). Its default value is 0.4. We chose to study the values from 0.1 to 0.9 with a step of 0.1. Our preliminary tests have shown that a finer step does not result in a significant improvement. That makes 9 values to test so that a brute force training can easily be performed and ensures the best training.

4.4.5 Results

Before comparing the algorithms we will first analyze each of them. All algorithms never produce false positives (the same layout for images of different documents) thus the FPR and FDR are always equal to 0. This is expected considering the high variation of the results so we will not discuss these performance indicators further. Similarly the FOR is always equal to 1.7% which is the maximal possible value given the balance of the dataset between positive and negative conditions. Thus it is as bad as possible, not discriminative and will not be discussed in the following. For comparison purposes with the results, the maximal possible value of the FNR on our dataset is 94.4% and its ideal value is 0%.

PAL

Table 4.3 summarizes the different results of both versions of PAL. As we can see it is not stable. The use of low-resolution images improves it a bit. This is mostly due to the fact that PAL is designed for text line extraction and over-segments the graphical areas. We also noticed that PALL and PALB tend to produce similar segmentation results on graphical areas. Thus, an improvement could be to remove the regions that are common to PALB and PALL results.

Version	Resolution	FNR (%)	S_R	$\overline{n_R}$
PALB	LR	**94.3**	0.32	26
	HR	94.4	0.9	72
PALL	LR	94.4	**0.20**	89
	HR	94.4	0.34	149

Table 4.3: PAL testing results. All values should be as low as possible.

	Fold	Parameter number					Perf. Ind.		
		1	2	3	4	5	FNR (%)	S_R	$\overline{n_R}$
Low resolution	1	9	15	0.2	25	3	**93.9**	0.37	18
	2	7	10	0.2	40	3	94.0	0.38	18
	3	9	15	0.2	35	2	94.4	0.41	19
	4	6	10	0.2	30	3	94.4	**0.32**	16
	5	7	10	0.8	25	3	94.4	0.33	72
High resolution	1	7	25	0.7	25	1	94.4	0.61	41
	2	7	25	0.4	30	1	94.4	0.63	31
	3	6	25	0.4	20	1	94.4	0.63	37
	4	9	30	0.4	30	3	94.4	0.65	35
	5	5	25	0.4	30	2	94.4	0.63	27

Table 4.4: Voronoi testing results. "Perf. Ind." stands for "Performance indicators". All values should be as low as possible.

Voronoi

Table 4.4 summarizes the results of the Voronoi segmentation algorithm for each fold of the cross-validation. The numbering of the parameters is the same as that of section 4.4.4. From a global point of view, the algorithm performs better with low-resolution images although it is nearly as unstable as PAL. It has a slightly lower FNR, a lower S_R and it produces a lower number of regions. The fifth fold of the low-resolution testing is a clear outlier.

Regarding the values of each parameter, we can make the following analysis:

- The contour sampling rate (parameter $n°1$) has roughly the same value for HR and LR images. This means that our scaling of this parameter with the image size is adequate. Its best value seems to be 7.

- The maximum size of a noise contour (parameter $n°2$) has an optimal value in HR that is double of that in LR. Its best value is 10 in LR and 25 in HR.

Parameter	LR		HR	
	FNR ($^0/_{000}$)	S_R	FNR ($^0/_{000}$)	S_R
1	3.7	0.023	1.7	0.034
2	6.4	0.030	6.6	0.097
3	12.6	0.026	1.8	0.055
4	3.1	0.019	1.8	0.025
5	5.2	0.033	0.2	0.026

Table 4.5: Influence of each parameter of the Voronoi segmentation algorithm on the evaluation performance indicators

- The ratio of the value of the second highest peak (parameter $n°3$) has different values between HR and LR. This can be explained by the fact that because the LR images have a smaller size, the peaks of the distribution of the distances of the Voronoi edges will be steeper. In order to keep enough edges, we need to have a lower threshold in LR than in HR. Its best value is 0.2 in LR and 0.4 in HR.

- The maximum area ratio between two regions (parameter $n°4$) is approximately the same in LR and HR, around 30.

- The optimal size of the smoothing window for the distribution of the distances of the Voronoi edges (parameter $n°5$) is bigger in LR than in HR. This is coherent with the fact that the peaks are steeper in LR than in HR and thus need more smoothing. Its best value is 3 in LR and 1 in HR.

In order to study the influence of each parameter. Table 4.5 presents the maximum variation of the performance indicators (in per ten thousand) when only one parameter is changed. This reflects the influence of each parameter on the performance of the algorithm.

We can make the following analysis:

- The contour sampling rate (parameter $n°1$) has an average influence on the algorithm performance.

- The maximum size of a noise contour (parameter $n°2$) has a strong influence on the algorithm performance.

- The ratio of the value of the second highest peak (parameter $n°3$) has a significant influence on the algorithm performance in LR and should be carefully tuned. This explains its stability across the folds.

Fold	RMT	FNR (%)	S_R	$\overline{n_R}$
1	0.1	94.4	**0.18**	22
2	0.1	94.4	**0.18**	21
3	0.1	94.4	0.19	22
4	0.1	94.4	0.19	21
5	0.1	94.4	0.19	21
Best trade-off	0.3	**93.4**	0.27	12

Table 4.6: JSEG testing results (on low-resolution images). All values should be as low as possible.

- The maximum area ratio between two regions (parameter $n°4$) has a low influence on the algorithm performance.

- The optimal size of the smoothing window for the distribution of the distances of the Voronoi edges (parameter $n°5$) parameter has an average influence in LR and a very low influence in HR on the algorithm performance.

To summarize, parameter 2 (respectively 3) has the highest impact in HR (respectively LR). Parameters 4 and 5 have the lowest impact. Anyhow all parameters have a negligible impact in comparison to the improvement required to make the algorithm stable.

JSEG

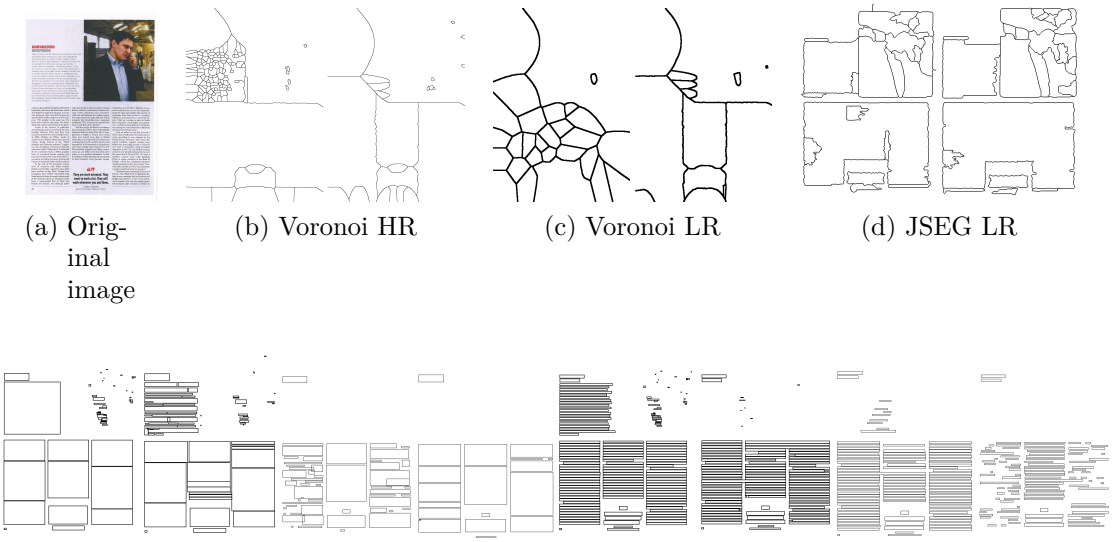
Table 4.6 summarizes the testing results of JSEG for each fold. The RMT is the region merging threshold. The results do not vary much and are surprising because JSEG has a very good S_R and a very bad FNR. This is due to the fact that the layout is not just related to the number of regions.

When looking deeper in the training data, we see that the FNR (respectively S_R) decreases (respectively increases) when the region merging threshold increases. JSEG actually produces a stable number of regions when there are many regions and this number becomes more unstable when their number decreases. This suggests that the merging process of JSEG is unstable. When visually examining the quality of the segmentation results we notice that an RMT of 0.1 over-segments the document and an RMT of 0.9 under-segments it very clearly. In our opinion, the value of 0.3 achieves the best trade-off with a FNR of 93.4%, an S_R of 0.270 and proper segmentation results.

The FNR on the training data goes from 68.9% to 94.4% with an average of 82.6%, S_R goes from 0.2 to 0.61 with an average of 0.44 and $\overline{n_R}$ goes from 3 to 22 with an average of 10.

Comparison of the algorithms

Figure 4.4.7 shows two results of each algorithm for each tested resolution on different copies of the same document with the same testing parameters. We can see the influence of the resolution on the algorithm results and their instability. Except for JSEG, they all tend to create artificial regions in place of the picture.



(a) Original image (b) Voronoi HR (c) Voronoi LR (d) JSEG LR

(e) PALB HR (f) PALB LR (g) PALL HR (h) PALL LR

Figure 4.4.7: Segmentation results of the algorithms

Considering all the above, JSEG has a better performance in terms of the stability of the number of regions it produces. It is followed by PALL. Voronoi and PALB both perform honorably on low-resolution images but are heavily unstable on high resolution images. Regarding the FNR, JSEG is also the only one that has one below 0.94. Texture features are known for being robust to noise and adding a multi-scale analysis probably allows a global view of the layout which helps producing more stable results.

Unfortunately, all the algorithms studied have a poor stability. The slightly best one, JSEG, achieves only a FNR of 93% while the other algorithms are at 94%. Even if an outstanding algorithm were twice as good as these, its FNR would still be at 46% which is far from the objective of being below 5%. Thus we need a radically new approach.

4.5 Conclusion

In this chapter we have presented the challenges related to document image segmentation and their use in industrial applications. We have extensively surveyed the state of the art and proposed a typology of document image segmentation algorithms that incorporates both their limitations and their functioning. The evaluation of three of these algorithms and of a natural scene image segmentation algorithm shows that these algorithms are highly unstable. It also shows that using natural scene image segmentation for document image segmentation yields interesting results.

It is likely that the use of color information and of algorithms capable of handling both textured and non-textured regions are two prerequisite to a good segmentation algorithm. PALL, PALB and Voronoi all use binary images as input which results in their poor handling of graphics and many false regions. Furthermore, the need to handle several font sizes calls for a multi-scale approach. This is one of the issues of the Voronoi segmentation algorithm which considers only one font size.

Most algorithms use binary images as input and connected components. Thus being able to extend the definition of connected components to color images would probably allow these algorithm to process color information with little modifications. This is the topic of the next chapter.

Chapter 5

Color connected components

The previous chapter highlighted the utter lack of stability of existing document image segmentation algorithms. Many of them use the connected components extracted from a binary image. Unfortunately, the binarization introduces a significant loss of information. Hence, in this chapter, we propose to extend the concept of connected components to color images. We first formalize the problem at hand and compare it with the superpixel formalism. Then we present a new model of human vision for computer vision. Notably it represents the gradient sensitivity of the human eye and its spatio-colorimetric sensitivity. Based on this model, we have developed two color connected component (CCC) segmentation algorithms. One that produces precise contours and one that produces less superfluous CCCs. We complete this set of algorithms with a post-processing based on our spatio-colorimetric color distance. All algorithms perform similarly to the state of the art on the Berkeley segmentation benchmark but our algorithms are three to five times more stable on almost one thousand document images. They also have a very high versatility and provide parameter-free edge and scale detection.

A fairly unambiguous way of describing an image is with the use of regions of uniform color or with a smooth color gradient. The detection of such regions is the topic of this chapter.

As we could see on the results of the different algorithms that we surveyed, they tend to create many noisy regions and some fail to identify the contours of the regions. Furthermore most of them use connected components which require binarizing the input image. This binarization implies a severe loss of information which may contribute to the poor stability of segmentation algorithms. Thus being able to produce "color connected components" (CCC) extending the definition of connected components to gray level and color images without losing any significant

information would be very useful.

This chapter presents two variants of such a color connected components segmentation algorithm and a post processing algorithm. It is organized as follows:

- Section 5.1 formalizes the problem of CCC segmentation.
- Section 5.2 presents the problems related to producing a stable CCC segmentation algorithm.
- Section 5.3 presents the state of the art of superpixel segmentation algorithms which are the closest algorithms to ours.
- Section 5.4 presents the model of human vision on which is based our algorithm.
- Section 5.5 presents our CCC segmentation algorithms
- Section 5.6 compares our algorithms with state of the art superpixel algorithms.
- Section 5.7 analyzes in depth the results of our algorithms and presents some direct applications.

The last section will conclude this chapter.

There are several significant contributions in this chapter:

- A formal analysis of the problem of CCC segmentation and of the related issues in superpixel segmentation in Sections 5.1 and 5.3,
- A detailed digital model of human vision for computer vision in Section 5.4 and summarized in Section 5.4.5,
- A new spatio-colorimetric distance to compute the perceptual color difference between two regions of uniform color in Section 5.4.3,
- A comparison of the state of the art of edge-preserving filters and of their suitability to model the contrast sensitivity of the human eye in Section 5.4.4,
- Two parameter free CCC segmentation algorithms and a post-processing algorithm for them. All represent a breakthrough compared to superpixel segmentation algorithms. They are presented in Section 5.5,
- A study of the stability of two state of the art superpixel segmentation algorithms and of the proposed CCC segmentation algorithms in Section 5.6.2,

- Some direct applications of the CCC segmentation algorithms and of their post processing including parameter free edge and scale detection and document layer/level separation in Section 5.7.

We will now start by formalizing the problem at hand.

5.1 Preliminary definitions

In a binary image a connected component is a connected area with a uniform color (black or white depending on the convention). When considering color or gray level images, the definition of a connected region remains the same. However, because of the noise of the image, it would not be relevant to consider the regions with a strictly uniform color such as the exact same RGB values for color images. Hence one needs to consider as uniform colors what a human observer sees as uniform colors. Furthermore, with the possibility of having a whole range of values comes the possibility of having color or intensity gradients. Such gradients can mark boundaries but some do not. For instance, the sky is a blue gradient and a flower petal can also be a color gradient. Figure 5.1.1 shows both boundary and non boundary gradients. It would not make sense to divide such regions into smaller regions with uniform colors. Trying to do this would actually raise the issue of positioning the boundaries of such regions. Hence we can consider that a smooth color gradient is also a connected component.

Figure 5.1.1: Gradients showing a boundary (rectangle) and not (ellipse).

Finally, the goal of our algorithm is to produce a set of connected regions where each region has a perceptually uniform color or is a continuous color gradient. Said otherwise, we want to produce the largest regions that do not contain any visually significant boundaries inside them. This can be defined as follows:

Definition 5.1.1 : **Color Connected Component.** *Color Connected Components (CCC) are regions of visually uniform color and/or with a visually smooth color gradient.*

From a mathematical point of view, we want to create a partition of the image space I into a set of regions $\{S_i\}$. Topologically, we consider each S_i to be open and to be a partition of i, $\{S_i\}$ must satisfy the following conditions:

$$\bar{I} = \bigcup_i \bar{S_i} \tag{5.1.1}$$

$$\forall i, j, i \neq j, S_i \cap S_j = \emptyset \tag{5.1.2}$$

where $\bar{}$ denotes the topological closure operator. This is a bit different from the common definition of a partition because we enforce the fact that the S_i are open. From an image point of view, this means that the regions contain the pixel areas but not the pixel borders. This will be useful to define neighbor regions. This is a similar definition to the one of the previous chapter in Equation 4.0.1.

Our uniformity/lack of boundary condition can be formulated with mathematical topology. Let us first define two neighbor regions.

Definition 5.1.2 : **Neighbor regions.** *Let u and v be two arc connected regions. They are neighbor regions if and only if*

$$u^\circ \cap v^\circ = \emptyset \tag{5.1.3}$$

$$\bar{u} \cap \bar{v} \neq \emptyset \tag{5.1.4}$$

where $^\circ$ denotes the interior operator.

We can now define a CCC partition of the image space:

Definition 5.1.3 : **CCC partition.** *Let I be an image space and $\{S_i\}$ a partition of it. $\{S_i\}$ is a CCC partition of I if and only if*

$$\forall S_i, \forall \{u, v\} \subset S_i^2, \phi(u, v) \leq t \tag{5.1.5}$$

$$\forall i \neq j, \phi(S_i, S_j) > t \tag{5.1.6}$$

where u, v are two neighbor regions. ϕ is a perceptual color distance function and t is a threshold above which a human observer would consider that there is a significant color difference.

There are a few things to notice from this definition. First the size and shape of the neighbor regions is not defined, neither is the perceptual color distance function. This is due to the fact that the current knowledge of the human visual sensory system does not allow us to define those properly for a computer vision application. Similarly, the threshold t depends on ϕ and cannot be defined unless ϕ is. We will attempt a first definition of these elements in Section 5.4.

Figure 5.3.1: From left to right: original image and two examples of superpixel segmentation produced by [VS08] and by [FH04].

5.2 Problem statement

Apart from the issue of finding the above mentioned perceptual color distance ϕ and the associated threshold t, we need to produce an algorithm that is stable, accurate and produces relevant results.

We highlighted the issue of stability in the previous chapter, in Section 4.1. The issue of accuracy relates to the fact that the boundaries of the regions produced by the algorithm should correspond to the ones of the objects/CCCs in the image.

Finally, our algorithm should not add superfluous regions that do not correspond to a CCC in the image.

5.3 State of the art

The nearest processing algorithms to what we are trying to produce are superpixel algorithms. These algorithms try to partition the image into regions called superpixels as shown on Figure 5.3.1. They generally try to optimize an "energy/objective" function of the form:

$$E(S) = H(S) + C(S) \tag{5.3.1}$$

where S is a partition of the image space, H is a function estimating the homogeneity of the regions of the partition and C is a function estimating their compactness (e.g. disk like or convex). Furthermore they frequently add another constraint which is the number of superpixels to produce.

5.3.1 Analysis of the issues related to superpixel segmentation

The objective function of superpixel segmentation has two main internal contradictions. Together, they explain the issues with superpixel algorithms. The first contradiction is that, on the one hand each superpixel should be homogenous and

on the other hand it should be compact. However, many homogenous regions are not compact and thus will be improperly partitioned. The second issue is related to predefining the number of superpixels. While this may be useful for applications whose computing resources are constrained, this limits the versatility of the algorithms. The number of homogenous regions in a given image is unknown a priori and may vary from one image to another. Hence defining the number of superpixels beforehand will most likely lead to a wrong number of superpixels.

These issues are even more acute when considering document images. We would expect to have one superpixel per character, but we do not know the number of characters in the image. Regarding the compactness constraint, characters are far from having a compact or disk-like shape. Hence we will keep these issues in mind while reviewing the state of the art of superpixel algorithms.

5.3.2 Overview of the state of the art

There has been a significant amount of work in the last years on the topic of superpixels. They have been surveyed in [ZMCL16] although the two most prominent works covering the state of the art are those of [ASS+12] and [Stu14]. Achanta et al. provides a concise overview of the main techniques of the field and benchmarks them. Stutz does an extremely thorough review and benchmark work and concludes that superpixel technologies are completely mature for 2D still images.

Considering the drawbacks of superpixel technologies that we mentioned in Section 5.3.1 and the fact that excellent overviews of the field already exist, we will start by summarizing the main algorithms in Table 5.1. While the main algorithms have many different names and functioning, roughly, they all tend to ascend or descend a gradient. They can also all be seen as optimization algorithms. The only exception is the case of SLIC superpixels [ASS+12] and the algorithms that use it, as SLIC is based on k-means.

The last two columns of the table are the results of each algorithm on the Berkeley segmentation benchmark 500 [AMFM11]. The first of the two columns corresponds to the image boundary recall (BR, how much the superpixels cover the image boundaries) and the second column to the undersegmentation error (UE, how much the superpixels cross the image boundaries). The boundary recall should be as close to 1 as possible and the undersegmentation error should be close to 0. This benchmark will be detailed in Section 5.6.1. These results depend on the values of the parameters of the algorithms as well as on a distance tolerance parameter for the computation of the boundary recall. Because of this they may be different from one evaluation to another. We used the values that made consensus across the different results available.

The algorithms are first sorted according to the number of parameters that they have, then according to whether or not the user needs to specify the number of

superpixels (SP) and finally according to the data structure which is the most common classification criteria. Considering our previous comment about specifying the number of superpixels we can already discard the algorithms that require it. Among the remaining algorithms, the algorithm of [PZZL14] has seven parameters which will make it very difficult to train. Finally, the only remaining algorithms that are relevant to our problem are those of [CM02, FH04, VS08].

Table 5.1: Summary of the characteristics of the main superpixel (SP) algorithms.

Algorithm	Number of parameters						Data structure	Features	Objective function	Main algorithm	BR	UE
	Total	Nb of SP	Homo-geneity	Shape	Scale	Other						
[CM02]	1	0	0	0	1	0	Matrix	Labxy	Distance	Mean shift	0.83	NA
[LSK+09]	1	1	0	0	0	0	Matrix	Vxy	Gradient	Active contours	0.61	0.24
[ASS+12] SLIC	2	1	0	1	0	0	Matrix	Labxy	Distance	K-means	0.82	0.07
[NP14](1)	2	1	0	1	0	0	Matrix	Labxy	Distance	SLIC	0.82	0.07
[NP14](2)	2	1	0	1	0	0	Matrix	Vxy	Distance	Watershed	0.8	0.07
[VS08]	3	0	0	1	1	1^{t}	Graph	Labxy	Distance	Quick-Shift	0.79	0.2
[FH04]	3	0	0	0	2	1^{s}	Graph	RGB	Distance	Custom MST building	0.84	0.23
[MPW+08]	3	1	0	1	1	0	Graph	Contour gradient	Contour gradient	Min-cut	0.68	NA
[ZHMB11]	3	1	2	0	0	0	Graph	Lab	Energy	Elimination algorithm	0.82	NA
[SFS12]	3	1	0	2	0	0	Matrix	Labxy	Distance	SLIC	0.8	0.2
[VBM10]	4	1	0	2	1	0	Graph	Vxy	Energy	α-expansion	0.7	0.22
[LTRC11]	4	1	0	0	2	1^{t}	Graph	Vxy	Entropy	Greedy optimization	0.92	0.06
[VBRV12]	4	1	0	0	1	2^{tq}	Matrix	Labxy	Energy	Hill-climbing	0.9	0.05
[CMM13]	4	1	2	1	0	0	Matrix/ graph	Labxy	Energy	Greedy optimization	0.98	0.03
[LFB16]	5	1	1	1	1	1^{q}	Matrix/ graph	Lab+bag of words	MAP	MRF solver	0.98	0.05
[PZZL14]	7	0	1	3	2	1^{i}	Matrix	RGBxy	Distance	Watershed	0.95	0.14
[PSYL15]	7	1	2	2	2	0	Graph	Labxy	Gradient +Energy	α-expansion	0.9	0.4

^{t} parameter used for color/geometry distance trade-off. ^{q} parameter used for quantization. ^{s} parameter used for image smoothing. ^{i} parameter used for number of iterations. MST stands for minimal spanning tree. MAP stands for maximum a posteriori. V in features stands for Value from HSV

5.3.3 Detailed review of relevant algorithms

The Mean-Shift algorithm [CM02] is a well-known superpixel algorithm based on estimating the normalized density gradient. Given a set of points $x_{i=1..n}$ in a d-dimensional space and a point x in this space, the normalized local density gradient is estimated by the mean-shift vector:

$$m_{h,G}(x) = \frac{\sum_{i=1}^{n} x_i g\left(||\frac{x-x_i}{h}||^2\right)}{\sum_{i=1}^{n} g\left(||\frac{x-x_i}{h}||^2\right)} - x \tag{5.3.2}$$

where h is a bandwidth/scale parameter and g is the normal kernel function (e.g. a Gaussian window). The algorithm starts by using a set of seed points sampled on a grid of the image. Then it computes the mean-shift of each point and move them accordingly to the mean-shift vector. Computing the mean-shift and moving the points continues until all points have converged to an attraction point (the paper proves the convergence of the algorithm). The attraction basin of each convergence points makes a superpixel. Thus this algorithm can be seen as a type of gradient ascent algorithm. The grid sampling of the image is only used for optimization purposes. The algorithm could use every pixel (for an increased computational cost). Thus we can consider that the sampling step is not a parameter of the algorithm. The strength of the mean shift algorithm lies in the fact that it does not need to estimate the data density to compute the density gradient. Furthermore the use of a normalized gradient (normalized by the local density) allows the algorithm to converge quickly in areas of low density and slowly in dense areas. Hence it does not waste iterations in areas where the density peak is unlikely to be and it refines its search where it is likely to be. Assuming that the density is known it should produce similar results to a watershed algorithm. Thus it has the same drawback as the watershed: it tends to produce too many small regions. Furthermore the algorithm is also particularly slow to converge.

The Mean-Shift algorithm has been improved with the Quick-Shift (QS) algorithm [VS08]. The algorithm first estimates the local density at the point x_i with

$$P(x_i) = \frac{1}{n} \sum_{j=1}^{n} \frac{g\left(\frac{D_{ij}}{h}\right)}{h} \tag{5.3.3}$$

where D_{ij} is the distance between the points x_i and x_j. Then each point is linked to its nearest neighbor y_i that increases the local density:

$$y_i = \underset{j:P_j>P_i}{\arg\min}\, D_{ij} \tag{5.3.4}$$

This operation is iterated for every x_i until $y_i = x_i$. Since there is no upper limit on D_{ij} the whole image ends up being merged into one cluster. The clustering order can be used to describe this cluster as a tree whose branch lengths are the distances between the points. To obtain a specific superpixel clustering, the user needs to specify a maximum value for D_{ij} and the tree can be cut at this level. The last parameter of the algorithm relates to the scaling of the color space compared to the geometric space. This parameter could actually be included in most algorithms including the Mean-Shift.

[FH04] devise a superpixel segmentation algorithm that uses the graph structure to represent the image geometry. Their algorithm produces a segmentation for each image channel and then intersects all segmentation results. The first step of the algorithm is a Gaussian blur to remove small invisible artifacts. The size of the blur window is the first parameter. Then the algorithm builds a graph where each pixel is a node and the edges represent the pixel neighborhood. The edge weight $w(e)$ is the intensity difference between the pixels. At the beginning each pixel is its own connected component (CC). The edges are processed by increasing weight. If the current edge links two different CCs CC_1 and CC_2 and the edge satisfies the following condition

$$w(e) < \min(\max_{e \in MST(CC_1)}(w(e)) + \frac{k}{|CC_1|}, \max_{e \in MST(CC_2)}(w(e)) + \frac{k}{|CC_2|}) \qquad (5.3.5)$$

where MST stands for minimal spanning tree, $|.|$ denotes the cardinality operator and k is a parameter controlling the size of observation then the two CCs are merged. At last a post processing step merges regions whose size is below the third parameter of the algorithm. Once again, this third parameter could be used by all other algorithms. However, since it can remove small significant details, it should be used with caution.

As we have seen some issues remain about the size of objects to identify and the trade-off between geometric and color distance. This is why we base our algorithm on a model of the human eye which we will describe now.

5.4 Human vision model

It is very difficult to identify the meaningful content of an image and the perceptual difference between different CCCs. One step towards solving this issue is through a better understanding of the human vision. This would allow the removal of the insignificant elements and the design of a virtual eye capable of extracting the information as the human eye sees it.

Several studies have been done to understand and model the inner workings of

the human eye. [Buc80, VT86, UB87, SRCEB05, LPN08, vdBSK08] have focused on the perception of colors. [NSB85, PGCW01] have focused on the physical geometry of the human eye which could help model its geometric aberrations. [BC69, Bar92] study the contrast sensitivity. Finally a thorough analysis of the brain processing involved in visual perception is done in [LH88]. To our knowledge, the main model of human vision that has been proposed in computer vision is that of [IKN98], but it is limited to visual saliency. It only provides a saliency map of the image without any information on the perceived colors, regions or gradients. Multi-layer perceptrons (MLP) are also sometimes mistaken for models of the human eye. However their modern implementation with a repetitive and identical topology is in contradiction with the neural organization described in [LH88]. This kind of implementation also forgets the ideas described in the original MLP paper [Ros61] which planned for far more complex MLP architectures to model human vision. Hence it seems that the model proposed here would be the first of its kind.

Our guiding principle is not to reproduce the physical mechanisms of the human eye and the human brain but rather to find image processing algorithms with similar sensitivity. Our model encompasses four characteristics of the human eye:

- Its spatial sensitivity,
- Its colorimetric sensitivity,
- Its spatio-colorimetric sensitivity,
- Its contrast sensitivity.

These will be described in the next subsections.

5.4.1 Spatial sensitivity

The human eye is a nearly spherical optical system composed primarily of a lens located behind the pupil and of photoreceptors located on the opposite side on the cornea.

The spatial acuity of the human eye has been studied by optometrists [MSM09]. There are two main measures related to it. The minimum visibile is the ability to see a black line on a white background. It is approximately equal to one arc-second e.g. one 3600^{th} of an angular degree. The minimum separabile is the separation power of the human eye e.g. the minimum distance between two points that we can distinguish. Its value is usually between 25 and 30 arc-seconds. The Ophthalmology Congress of Naples in 1909 set the normal minimum separabile to one arc-minute. This measure is done in optimal experimental conditions so it seems reasonable to test for this value and twice this value in our experiments.

In order to translate this angular distance to the perceived resolution of a document, we need to define a reading distance. A basic experiment with several people show that this distance is usually between 30 and 40 cm. This corresponds to the Harmon distance [Har51] which is the comfortable reading distance. Equation (5.4.1) shows how to convert the reading distance d in centimeters and the angular minimum separabile a into a dot per inch (dpi) resolution.

$$dpi = \frac{2.54}{d \times \tan(a)} \tag{5.4.1}$$

This corresponds to the situation depicted in Figure 5.4.1.

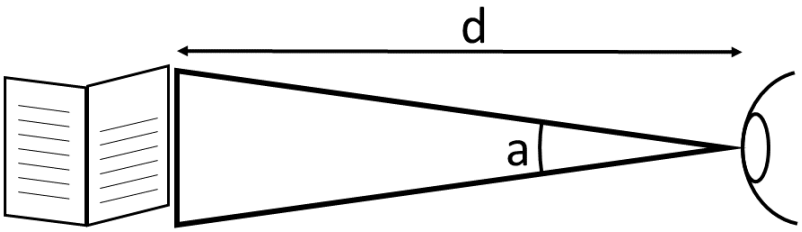


Figure 5.4.1: Situation to convert a reading distance and a minimum separabile into a resolution.

Table 5.2 summarizes the results obtained with it. Considering usual resolution values the possibilities are 150, 200 and 300 dpi.

Reading distance (cm)	Minimum separabile (arc-minutes)	
	$a = 1$ arc-min	$a = 2$ arc-min
$d = 30$	291 dpi	146 dpi
$d = 35$	249 dpi	125 dpi
$d = 40$	218 dpi	109 dpi

Table 5.2: Resolutions in dpi corresponding to minimum separabile and reading distances.

The resolution value range is between 109 dpi and 291 dpi. Considering that 200 dpi is in the middle of this range and a standard resolution value we can use it for the practical and perceived resolution for document image processing. It should be noted that aliasing issues may require a resolution of 300 dpi for improved OCR processing.

For an image of dimensions x, y taken with a camera, its also possible to obtain

its angular resolution r in arc-minutes per pixel in with the following equation:

$$r = 60 \times \frac{\text{lens\_aperture\_angle}}{\sqrt{x^2 + y^2}} \tag{5.4.2}$$

Once this angular resolution is obtained one simply needs to scale it to one or two arc-minutes per pixel to obtain an image at a proper resolution.

5.4.2 Colorimetric sensitivity

As we mentioned in Section 1.2.1, the human eye spectral sensitivity does not match the emission spectrum of an RGB channel. The human eye has three types of photoreceptors (also called cones because of their shape) dedicated to color. Figure 5.4.2 shows the variation of their sensitivity with light wavelength under a viewing angle of 2°[SS00]. These curves are very different from the ones of digital sensors. This is why the RGB color space is not perceptually uniform. Hence we need to find a color representation of an image that is more in accordance with how it is perceived.

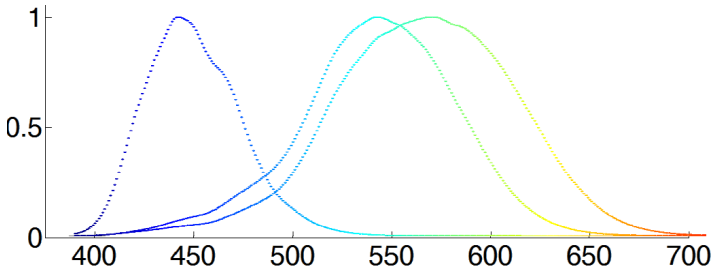


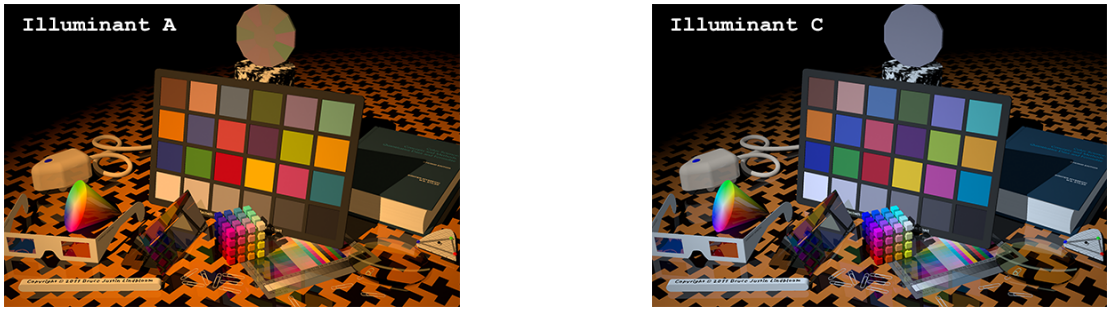
Figure 5.4.2: Variation of the sensitivity of the three human color photoreceptors with light wavelength.

Choice of the RGB color space

An image can be represented with three color channels: red (R), green (G) and blue (B). This defines the RGB color space. To be precise, there are actually several RGB color spaces depending on the hardware used to display RGB images. Each hardware has its own sensors or emitting devices which do not produce exactly the same colors. The hardware properties and its ability to display all visible colors is called the gamut or an ICC profile (ICC stands for International Color Consortium). This serves to define the RGB color space of a given hardware.

Another point to take into account is the light source. This is called the "illuminant" e.g. the white light source that is used. A common illuminant is called D65.

It corresponds to an average outdoor natural light and is the one used by the International Telecommunication Union [ITU15] and the International Commission on Illumination (CIE, Commission Internationale de l'Eclairage). It is defined by the ISO/CIE norm 11664-2:2007 (CIE S 014-2/E:2006). Another similar illuminant is D50. It is the standard illuminant of the ICC. Figure 5.4.3 shows the impact of changing the illuminant on an image.



(a) Illuminant A (b) Illuminant C

Figure 5.4.3: The impact of changing illuminants.[1]

Here we will consider that all RGB images use the sRGB color space. This particular color space is made to be independent from the gamut and uses the D65 illuminant. If one wants to have the exact RGB values displayed by a screen, he can use the screen ICC profile to compute the RGB values corresponding to his screen from the sRGB values and vice versa.

Since we try to make an algorithm that works for every hardware, one easy approximation consists in neglecting the ICC/gamut correction. This is motivated by the fact that recent screen technologies, in particular the screens of the Microsoft Surface brand are calibrated for the sRGB color space with negligible error [Son15, How15]. Hence no correction is needed for them and the input images.

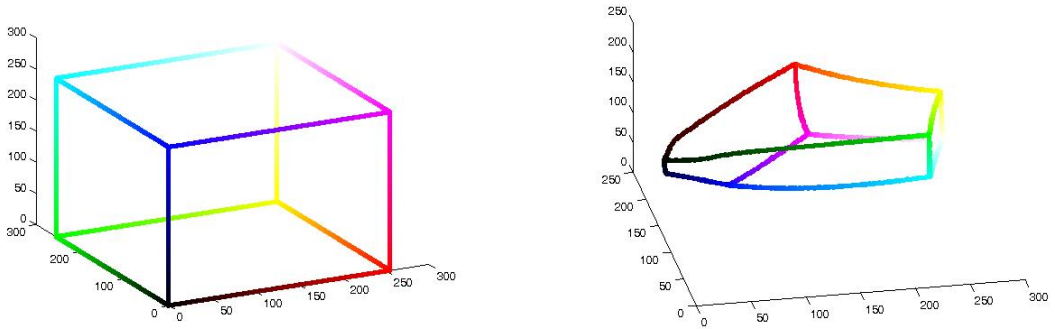
Now that the initial color space and illuminant of the input image are properly defined, the next task is to find out to which color space we will convert the input image. This color space should be representative of how the image colors are perceived by the human eye.

Choice of a perceptually uniform color representation

The most perceptually uniform color space to date is the Lab color space. It is a non linear color space which was created in 1976. It is currently defined in the ISO/CIE norm 11664-4:2008 (CIE S 014-4/E:2007). The details of the conversion

[1] Image reproduced from `http://www.brucelindbloom.com/`

from the sRGB color space to the Lab color space is available in Annex B. Figure 5.4.4 shows a comparison of the geometry of the sRGB and the Lab color spaces.



(a) sRGB (b) CIE Lab

Figure 5.4.4: The geometry of sRGB and Lab color spaces.

Originally, the Lab color space was thought to be sufficiently precise to use the Euclidean distance to estimate color differences. This distance is also referred to as CIE76. Since then, more precise measurements have shown that this is not the case and a new color distance has been defined: CIEDE 2000. The interested reader can find the details of the computation of this distance in the ISO/CIE norm 11664-6:2014 (CIE S 014-6/E:2013). There is a strong restriction for the use of this distance. It is made to measure color differences between colors with a Euclidean distance (in the Lab color space) lower than 5 [SWD05] and with a lightness L around 50 [Kue02]. Thus one should be careful when using it and the Euclidean distance may be preferable in generic cases. It also has the advantage of being easy to compute.

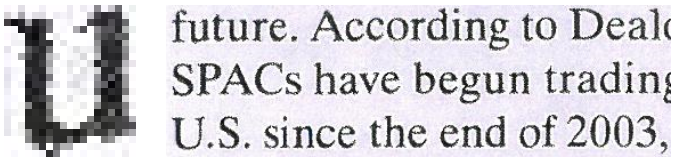
According to Sharma [Sha02] the just noticeable color difference in Lab space with the Euclidean distance is at approximately 2.3. In the Lab space with the CIEDE 2000 metric, it is of 0.6. Based on this, Linhares et al. [LPN08] studied the number of discernible colors from real images. It is of approximately 2 million colors. This figure is corroborated by theoretical findings. Thus it seems that a 32 bits color representation is sufficient to represent accurately visible colors since it contains over 16 million colors.

Considering all this, we will convert sRGB images to the Lab color space and use the Euclidean distance to measure color differences.

5.4.3 Spatio-colorimetric sensitivity

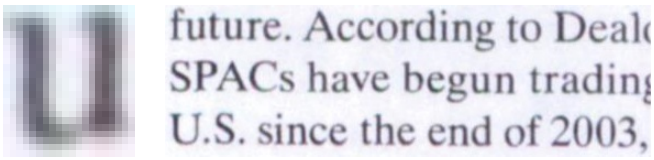
It is easier to differentiate large patches of colors rather than small color dots. Hence the size of the CCCs and the scale of color variations influence our perception of color differences. This is what we call the spatio-colorimetric sensitivity of the human eye.

A good study of the general optical processing functions of the human brain has been done in [LH88], but it does not address the specific question of having a spatio-colorimetric distance between two colored regions of a different size. Such a distance would allow the creation of color cleaning algorithms that would remove the color print and scan noise as shown on Figure 5.4.5 without removing the colors of significant regions.



(a) Impulsion noise

(b) Zoomed out impulsion noise



(c) Blur noise

(d) Zoomed out blur noise

Figure 5.4.5: Color print and scan noise.

Quick review of the state of the art

There are not many works that study the influence of the region size on the perception of color differences. Instinctively, it is more difficult to distinguish colors if they are on a small region than if they are on a large region. This is why all color experiment tests are done on precise region sizes [SRCEB05].

This is also why Stockman and Sharpe [SS00] studied the variation of the eye sensitivity with two viewing angles: 2°and 10°. The variation of color perception with the size of the stimuli (the color region) has been already studied [CIE14, WSH15, XLL+11] but only for regions with a viewing angle of at least 1°. This

is at least 30 times larger than the minimum separabile and too large for the use case of color denoising where color regions have the size of one or a few pixels.

Proposed spatio-colorimetric distance

Getting down to the physical principles behind the influence of region size, we can expect that the ability to distinguish colors is relative to the amount of light from that specific colored region that is received by the eye sensors. Hence it should be a combination of light intensity and the area of the light emitting region. Assuming that two regions are illuminated by the same illuminant under identical conditions, the light dependency is already accounted for in the Lab color space. Thus only the dependency with the region area remains.

We can also assume that above a certain region area the maximal eye color discrimination is reached and thus the color distance should not change.

Thus we propose the model of a spatio-colorimetric distance of Equation (5.4.3) with ΔL, Δa and Δb being the differences between the values of the corresponding Lab channels. It varies linearly with the region area S. It is an adaptation of the Euclidean distance with three parameters S_L, S_a and S_b which correspond to the region size above which one can clearly distinguish colors.

$$\Delta_{Lab} = \Bigg[min\left(\left(\frac{S}{S_L}\right)^2, 1\right) \times \Delta L^2 + min\left(\left(\frac{S}{S_a}\right)^2, 1\right) \times \Delta a^2 + min\left(\left(\frac{S}{S_b}\right)^2, 1\right) \times \Delta b^2 \Bigg]^{1/2} \tag{5.4.3}$$

Experimental validation of the proposed distance

We conducted very rough real world experiments that need to be reproduced in a more controlled environment with a more precise setup. However, they allow us to highlight two facts:

- The distance model is consistent with the experiments modulo the approximations related to the experimental conditions.
- It is necessary to use such a spatio-colorimetric distance.

The experiments were as follows: a human observer sits in front of an uncalibrated computer display. This screen displays four aligned color dots at a minimal size and separated by white space. The four colors are spaced equally along one axis in the Lab color space. The user is then asked to increase the size of the dots until he can differentiate the colors. Annex C gives the Lab coordinates of each row of dots. We are aware that the experimental conditions are not ideal at all compared to those of the state of the art. This is mostly due to hardware availability and time

Color variation	Minimal surface (px)	S_L (px)	S_a (px)	S_b (px)
$\Delta L = 10$ (a=20,b=0)	4	17.4		
$\Delta L = 10$ (a=b=0)	14	60.9		
$\Delta a = 10$ (L=50,b=0)	5		21.7	
$\Delta b = 10$ (L=70,a=0)	20			87.0
$\Delta L = 15$ (a=b=0)	NA			
$\Delta a = 15$ (L=50,b=0)	5		32.6	
$\Delta b = 15$ (L=70,a=0)	14-15			91.3

Table 5.3: Experimental results on the human spatio-colorimetric sensitivity

constraints. As we mentioned, before, these experiments need to be performed with a more reproducible setup and in a better environment, in particular with a calibrated display.

The average minimal sizes given by the users are summarized in Table 5.3. They are expressed in pixels at 100 dpi with a viewing distance of approximately 30-35cm. The experiment with a distance of $\Delta L = 15$ did not work because the colors appeared indiscernible, probably because of gamut limitations.

If we neglect gamut issues and since the Lab just noticeable Euclidean distance is 2.3, the values of S_L, S_a and S_b can be found by solving Equation (5.4.4) which only has one unknown for each series of four dots. The results of this equation are given in Table 5.3.

$$\Delta_{Lab} = 2.3 \tag{5.4.4}$$

We can notice the significant influence of the presence of color when L varies. The fact that the screen was not calibrated and thus the luminance was far from ideal probably explains this. The variation of S_a with Δa is probably due to the measurement error on the minimal surface. S_b does not vary much which means that our model is not too far from reality. We also notice that S_b is the largest which matches the fact that the human eye is less sensitive to this channel. In the following we will use $S_L = 17.5$, $S_a = 27.2$ and $S_b = 89.1$ at 100 dpi and scale them with the processing resolution. Once again, considering the experimental conditions to obtain these values, they should definitely be confirmed by more thorough experiments.

Assuming that these surfaces are disks, it is possible to determine their radius and the corresponding viewing angle with Equation (5.4.7). S_{cm^2} is the surface in square centimeters, *dpi* is the resolution corresponding to the pixel surfaces, r is

Color variation	Minimal radius (cm)	Viewing distance		
		25 cm	30 cm	35 cm
$\Delta L = 10$ (a=20,b=0)	0,06	16	14	12
$\Delta L = 10$ (a=b=0)	0,11	31	26	22
$\Delta a = 10$ (L=50,b=0)	0,07	18	15	13
$\Delta b = 10$ (L=70,a=0)	0,13	37	31	26
$\Delta a = 15$ (L=50,b=0)	0,08	23	19	16
$\Delta b = 15$ (L=70,a=0)	0,14	38	31	27

Table 5.4: Minimal viewing angle in arc-minutes for each experiment and different viewing distances.

the corresponding radius, d is the viewing distance and α is the viewing angle.

$$S_{cm^2} = \frac{S \times 2.54^2}{dpi^2} \tag{5.4.5}$$

$$r = \sqrt{\frac{S_{cm^2}}{\pi}} \tag{5.4.6}$$

$$\alpha = 2 \quad arctan\left(\frac{r}{d}\right) \tag{5.4.7}$$

Table 5.4 gives the corresponding minimal viewing angles corresponding to our experiments. We can see that most values are below half a degree (30 arc-minutes) which confirms the necessity of establishing such a spatio-colorimetric distance.

5.4.4 Contrast sensitivity

The human eye does more processing than what has been described so far in this Section [LH88]. In particular the human eye incorporates an edge detector/filter and thus is very sensitive to edges. On the opposite its acuity is reduced for uniform regions. A model was proposed for the contrast sensitivity in [Bar92]. However it involves 9 parameters and only deals with one dimensional stripe patterns. Adapting it to a 2-dimensional set of regions of arbitrary shape would be extremely difficult without a lot more experimental data. However, the field of image processing has recently seen the apparition of filters with properties similar to those of the human eye: edge-preserving filters. There exist several of them and we shall review them to find which one is the most suitable for our use.

Nagao and Matsuyama [NM79] proposed a simple parameter-free edge-preserving filter based on local windowed statistics filtering around each pixel. However their algorithm does not scale up: it took more than 10 minutes to pro-

cess one image.

Perona and Malik [PM90] designed an edge preserving filter based on the heat I diffusion equation for a medium with constant conductivity c:

$$I = c\Delta I \tag{5.4.8}$$

Here I is the image intensity and c is related to the intensity gradient. It has four parameters: the number of iterations of the filter, the conduction coefficient κ, a continuous Laplacian approximation parameter λ and the choice of the conduction function. κ serves to define the influence of small gradients much like a scale factor. λ influences the speed of the diffusion and should be set to 0.25. The first diffusion function favors high contrast edges while the second one favors large regions. According to the authors, the function type does not have much influence on the result. Our tests give the best results for three iterations, $\kappa = 50$ and the second function.

Farbman et al. [FFLS08] designed a filter which can identify the different levels of detail in an image and show its use for edge-preserving smoothing. However their algorithm requires more than 4.5 GB of memory which is unpractical.

A filter was proposed based on wavelet decomposition [Fat09]. It produces a wavelet decomposition of the image and then linearly combines the wavelet layers to keep only the desired level of detail. This filter is very fast, but, once set, it can only process details at a given scale. The best results are obtained with the default parameters and simply removing the highest level of detail.

Another simple filter based on fixed windowed filtering was proposed in 2009 [NP09]. Similarly to the one of [NM79], it is very slow (more than 4 minutes for three iterations). Thus we used three iterations in our experiments.

The fast guided filter [HST13] uses the intensity of an image to guide its filtering. It uses a box window ω_k centered on pixel k and of radius r. The filtering output q is computed with Equation (5.4.9) where i is the output pixel index, p is the image input and I is its intensity map.

$$q_i = \overline{a_i} I_i + \overline{b_i} \tag{5.4.9}$$

$$a_k = \frac{\frac{1}{(2r+1)^2} \sum_{i \in \omega_k} I_i p_i - \mu_k \overline{p_k}}{\sigma_k^2 + \epsilon} \tag{5.4.10}$$

$$b_k = \overline{p_k} - a_k \mu_k \tag{5.4.11}$$

μ_k and σ_k are the mean an variance of I in ω_k. $\overline{p_k}$ is the mean of p in ω_k and $\overline{a_k}$ and $\overline{b_k}$ are the averages of a_k and b_k across all windows ω_k overlapping the pixel i. ϵ is a regularization parameter set to determine the relevant edges. Our experiments show that the best results are obtained for $r = 5$, $\epsilon = 0.01$ and a sampling rate of

1 which means that the image is processed at its full resolution.

Chaudhury et al. [CSU11] improved the original bilateral filter [TM98] by using trigonometric functions to approximate the desired kernel. This reduces the computational cost of the filter. The base equation of the bilateral filter is:

$$J_k(x) = \frac{\int_{y\in\omega_x} \psi(y)\phi(I_k(y) - I_k(x))I_k(y)dy}{\int_{y\in\omega_x} \psi(y)\phi(I_k(y) - I_k(x))dy} \tag{5.4.12}$$

where $I_k(x)$ and $J_k(x)$ are respectively the input and output intensities at pixel x of channel k. ψ is called the spatial kernel and ϕ is called the range kernel. Both are Gaussian kernels of respective variances σ_s and σ_r approximated with trigonometric function up to a precision of ϵ. ω_x is a box window centered around the pixel x and of width w. We obtained the best results for $\sigma_s = 1.5$, $\sigma_r = 50$, $\epsilon = 0.01$ and $w = 6\sigma_s$.

The last edge preserving filter that we studied is the Domain Transform filter [GO11]. Their idea is to change the space of the image instead of changing the filter. The domain transform Equation (5.4.13) changes an n-dimensional signal I_k (k is the dimension index) into a one dimensional function ct on which the chosen filter can be applied.

$$ct(u) = \int_0^u 1 + \frac{\sigma_s}{\sigma_r} \sum_{k=1}^{c} \left| \frac{dI_k(x)}{dx} \right| dx \tag{5.4.13}$$

The parameter σ_r is related to the value range standard deviation of the filter and σ_s is related its spatial range standard deviation. We use a recursive filter to produce a filtered signal J (Equation (5.4.14), n is the iteration number). Its feedback coefficient a depends directly from σ_r and σ_s. Since it works only along one dimension, it is applied successively along x and y directions. This allows turning around corners and can be iterated for more complex geometries.

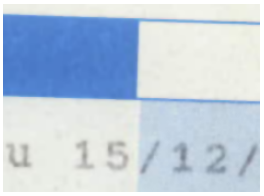
$$J[n] = (1 - a^d)I[n] + a^d \times J[n-1] \tag{5.4.14}$$

The authors recommend using three iterations of the filter and our tests show the best performance for $\sigma_r = 0.2$ and $\sigma_s \approx 1\ cm$ (to be converted in pixels at the image resolution). The fact that the spatial range actually has a physical significance is of interest as it allows the algorithm to be resolution independent. [Bar92] mentions the fact the human eye can integrate intensity over a spatial range of 12°which corresponds to approximately 5-7 cm depending on the reading distance. For a Gaussian window with a standard deviation of 1 cm, 95% of the integral value is made from pixels over a span of approximately 4 cm. We found that increasing the standard deviation to extend this to 7 cm does not improve

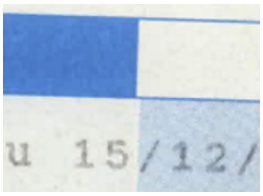
the results but increases the processing costs thus 1 cm seems to be an optimal choice.

We used the filter implementations provided by the authors when available. All the experiments were run on Matlab with A4 color images at 300 dpi. Figure 5.4.6 shows the results produced by these filters. Most filters either maintain good edge sharpness and text quality but fail to reduce the noise such as [Fat09] or they reduce the noise properly but blur the text and edges such as [HST13]. The domain transform filter [GO11] provides the best trade-off with completely uniform regions except for the lower right quadrant and very sharp text and edges as well. Thus this is the filter that we will use.

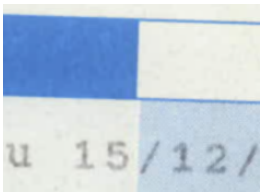
(a) Original image



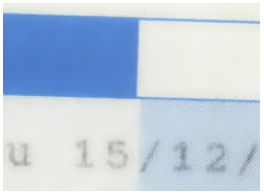
(b) [PM90]



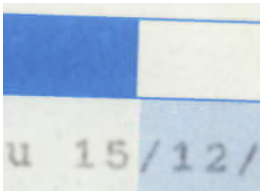
(c) [Fat09]



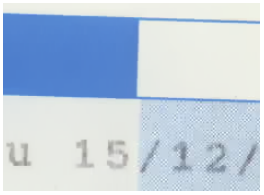
(d) [NP09]



(e) [HST13]



(f) [CSU11]



(g) [GO11]

Figure 5.4.6: Filtering results of the algorithms (best viewed in color)

5.4.5 Final eye model

To summarize the contents of this section our eye model is made of the following pre-processing steps:

- Resize the image to 200 dpi (or maybe 300 dpi for OCR processing) to account for the spatial sensitivity.

- Filter the image with the domain transform filter [GO11] to account for the contrast sensitivity.

- Convert the filtered image from sRGB to Lab color space to account for the colorimetric sensitivity.

The resizing function will be detailed in Section 5.5.1.

This model also includes our proposed spatio-colorimetric color distance to measure the color difference between two regions. This is the function ϕ that we proposed in the problem statement in section 5.2. Considering how we built the function, the threshold t should be equal to 2.3.

While this model is quite simple, to our knowledge, this is the most comprehensive one available in computer vision.

It is noticeable that the input image is filtered before being converted to Lab color space. This is motivated by a very important difference between the RGB color space and the Lab color space: RGB and L channels represent an intensity value while a and b channels represent a color value. Hence algorithms that are made for image representations with intensity channels should not be applied to other types of channels without validating that this is a relevant approach. In the case of Lab color space they may introduce colorimetric aberrations. This is the case for filters (although some sort of color bleeding may be understandable in the case of a blur filter) and for hill-climbing/gradient algorithms such as watershed.

We will now present our CCC segmentation algorithms.

5.5 Watercolor CCC segmentation algorithm

The goal of a Color Connected Component segmentation algorithm is to extend the definition of connected components to color images.

We developed a first algorithm capable of identifying CCCs. It is called Watercolor (WC) because it is based on the watershed algorithm. This algorithm is based on a gradient representation which produces precise region contours but too many regions in particular for the gradients on the border of characters. This is why we also propose another version of this algorithm called Smooth Watercolor (SWC). It is based on a smoother gradient representation which handles better the character border gradients, but produces contours that are less precise than those of WC. Finally, we also provide a post-processing algorithm to merge regions of similar colors. It is based on our spatio-colorimetric distance.

5.5.1 Watercolor (WC)

Both WC and SWC algorithms work on the same principle depicted in figure 5.5.1. The general idea is to apply the watershed algorithm on the inter-pixel color distances in order to identify regions with small local color variation e.g. regions of uniform colors or of small color gradients. We can also notice that the first steps are the ones advised in Section 5.4.5.

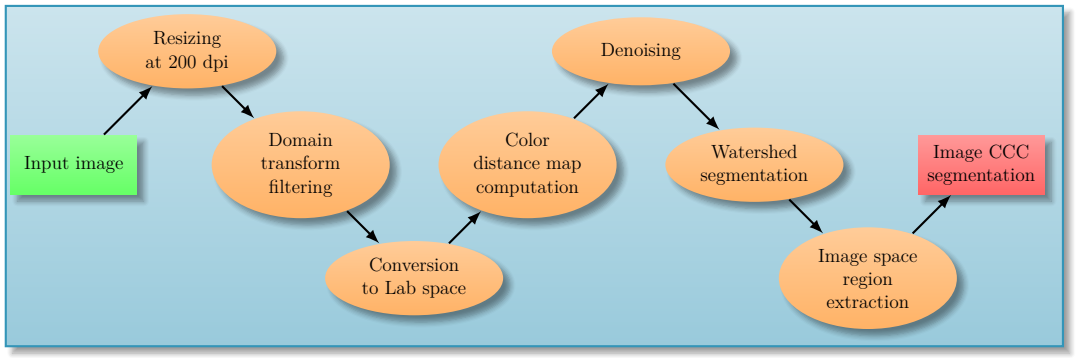


Figure 5.5.1: Watercolor and Smooth Watercolor algorithms

Resizing at 200 dpi: This is a simple resizing with bi-cubic interpolation [Key81] and anti-aliasing by modifying the interpolation kernel h_a as follows:

$$h_a(x) = sf \times h_o(sf \times x) \tag{5.5.1}$$

$$kernel\_width = \left\lceil \frac{4}{sf} \right\rceil \tag{5.5.2}$$

$$sf = \frac{200}{\text{input image resolution}} \tag{5.5.3}$$

where $\lceil . \rceil$ denotes the ceiling function, h_o is the original bicubic interpolation kernel and h_a is the kernel with anti-aliasing. This avoids producing the artifacts that would be present with a bilinear or nearest neighbor interpolation. The anti-aliasing also preserves the contours.

Domain transform filtering: This is the one described in Section 5.4.4.

Conversion to Lab space: This is a classical conversion performed in floating point arithmetic in order to avoid adding noise.

Color distance map computation: The color distance map with the denoising step are the two critical steps of the algorithm. The goal of the color distance map is to represent the color distances/gradients between the image pixels. Gradient maps and other similar metrics do not provide precise information with each

neighbor of a given pixel. Using a graph representation would involve costly processing so we use a matrix representation of color distances. This implies the use of a four-connected pixel neighborhood since we cannot represent the two diagonal distances with one diagonal pixel. We use a plain color distance with each four-connected neighbor. The matrix representation doubles the image dimensions in order to interpose distance values between the pixels. Figure 5.5.2 represents such a color distance map.

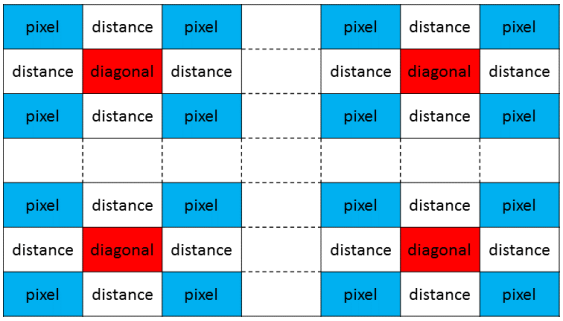


Figure 5.5.2: Organization of the color distance map

The color distance map M is computed with the following equations (in the same order):

Compute the color distances between horizontal neighbor pixels:

$$\forall 1 \leq i \leq n, \quad 1 \leq j \leq m-1, \quad M(2(i-1)+1, 2j) = d(I(i,j), I(i,j+1)) \tag{5.5.4}$$

Compute the color distances between vertical neighbor pixels:

$$\forall 1 \leq i \leq n-1, \quad 1 \leq j \leq m, \quad M(2i, 2(j-1)+1) = d(I(i,j), I(i+1,j)) \tag{5.5.5}$$

Compute the color distances for the pixels:

$$\forall 1 \leq i \leq n, \quad 1 \leq j \leq m, \tag{5.5.6}$$
$$M(2(i-1)+1, 2(j-1)+1) = min_4(2(i-1)+1, 2(j-1)+1, M)$$

Compute the color distances for the diagonals:

$$\forall 1 \leq i \leq n-1, \quad 1 \leq j \leq m-1, \quad M(2i, 2j) = max_4(2i, 2j, M) \tag{5.5.7}$$

where n and m are the dimensions of the input image, $max_4(x, y, M)$ and $min_4(x, y, M)$ are the functions which return respectively the maximal and minimal value of M in the four-neighborhood around the coordinates (x, y) and d is the Euclidean distance in Lab color space. Since the pixel values computed in Equation 5.5.6 are local minima, no pixel will be on the border of a region produced by the watershed algorithm. Similarly, since the diagonal values computed

in Equation 5.5.7 are equal to the local maximum, they will not create new regions with the watershed algorithm.

Denoising: In spite of the filtering there still exists some colorimetric noise. This noise creates local minima which in turn create superfluous regions. The goal of the denoising is to remove these minima. Thus all values below a distance threshold t are set equal to 0. The threshold is computed with

$$t = \frac{\sigma(M)}{c_2} + c_1 \tag{5.5.8}$$

where $\sigma(M)$ is the standard deviation of the distance map values. The constant $c_1 = 0.2$ is equivalent to 10% of the just noticeable color difference hence we consider it to be the minimum value below which color differences are not meaningful anymore. The constant $c_2 = 7.5$ is estimated empirically and serves to estimate all the color differences that are clearly not meaningful based on the color distances contained in the document. Since we only need to remove the local minima, this threshold is very conservative and only removes the smallest non meaningful color distances. The other non meaningful color distances will be removed by the watershed.

This threshold makes the algorithm adaptable to both the document content and the colorimetric print and scan noise. It may be argued that 7.5 is dependent on the image content and thus a parameter. However, we make the hypothesis that the image content dependency is contained in $\sigma(M)$ and that the 7.5 factor reflects how much the human visual system can adapt to this content variation and more precisely the color variations. The idea behind this is that the human eye and the way we analyze an image adapt to how much color variations are present in the image. This color variation is measured by $\sigma(M)$. For images with low color variations we will try to make sense of smaller color differences than for an image with high color variations. Our tests to find this value on several kinds of images support our hypothesis that the 7.5 factor is only dependent on the observer and should not be modified as long as the observer has a normal eye sight. As we will show in Section 5.7, this value yields plausible and good results on a very wide range of document and natural scene images. Similarly to the experiment for the spatio colorimetric model of Section 5.4.3, one could try to corroborate this hypothesis with specially designed experiments.

Increasing the image resolution will span the same color difference over a larger number of pixels hence the values 0.2 and 7.5 are resolution dependent. Assuming that color gradients are spread linearly with the resolution (which may not be the case), these two values can also be scaled linearly.

Watershed: Once all the previous steps are done, we perform a watershed transform. This will segment the color distance map in as many regions as there

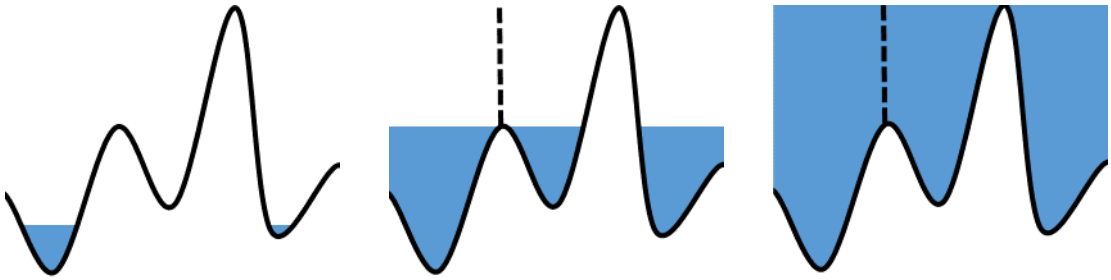


Figure 5.5.3: Process of the watershed transform.

are local minima. It produces a region index map N of the same size as the color distance map. This region index map puts in correspondence each value of the color distance map with the index of the region to which it belongs.

The watershed transform creates one water basin per local minima of the image. Each basin is filled from the bottom up (Figure 5.5.3a) and when two basins meet each other at a local maximum, it creates a "wall" to separate them (Figure 5.5.3b). The process ends when the water level reaches the global maximum of the image (Figure 5.5.3c).

Image space region extraction: The goal of this step is to convert the region index map N of the color distance map into one for the input image O. This is a straightforward sampling of the index region map on the locations of the pixels in the color distance map shown in Figure 5.5.2.

$$\forall 1 \leq i \leq n, \quad 1 \leq j \leq m, \quad O(i,j) = N(2(i-1)+1, 2(j-1)+1) \tag{5.5.9}$$

The regions produced for the input image do not have any border between them thus removing the ambiguity related to border pixels.

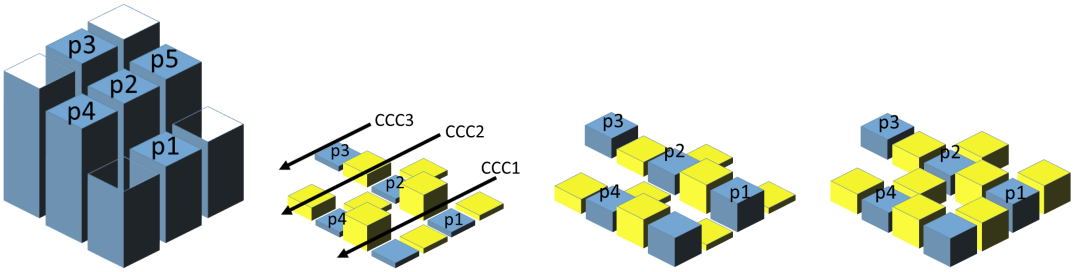
5.5.2 Smooth Watercolor (SWC)

The Watercolor algorithm produces very precise contours but separates character edge gradients along the direction of the edge as shown in Figure 5.5.4. This is because the pixel values in the color distance map are computed as the minimal color distance on the four neighborhood. An example of this is shown with the three ridges and CCCs in Figure 5.5.5b.

Instead the pixel values should depend on the maximum color gradient across the current pixel. Thus we propose a second version of this algorithm which solves this problem at the cost of the precision of the region contours: Smooth Watercolor.

Smooth Watercolor is the same as Watercolor except for the color distance map which is computed with the distance across the pixel rather than between the pixel

Figure 5.5.4: CCC segmentation and color distance map produced by Watercolor. Left: each region has a uniform color. Right: the pixel intensity represents the distance value. Notice the vertical rectangles in the "l" characters. The white borders will produce several regions.



(a) Image gradient (b) Color map of WC (c) Color map of SWC after Equation (5.5.10) (d) Color map of SWC after Equations (5.5.16) and (5.5.17)

Figure 5.5.5: Computation of pixel color distance/gradient with SWC. Pixels are in blue and distances in yellow. For an improved readability,the diagonal values are not represented.

and its neighbors. The first two steps of the color distance map computation are the same as those of Equations 5.5.4 and 5.5.5. Then we compute the pixel values to take into account the color distance (gradient) between their neighboring pixels in both directions (horizontal and vertical) with Equation (5.5.10).

$$\begin{aligned} &\forall 1 \leq i \leq n, \quad 1 \leq j \leq m, \\ &M(2(i-1)+1, 2(j-1)+1) = \max(d(I(i,j-1), I(i,j+1))/2, \\ &\qquad\qquad d(I(i-1,j), I(i+1,j))/2) \end{aligned} \tag{5.5.10}$$

Figure 5.5.5c shows the result of these equations. Notice how the CCCs of WC (Figure 5.5.5b) are not possible anymore because of the new pixel values. The valleys where were CCC1, CCC2 and CCC3 have been filled.

We can prove that this formula will not create new local minima or maxima

thus avoiding the creation of superfluous regions.

Proof. Without loss of generality we can assume that for a given couple (i, j)

$$M(2(i-1)+1, 2(j-1)+1) = d(I(i, j-1), I(i, j+1))/2 \tag{5.5.11}$$

The triangle inequality yields

$$M(2(i-1)+1, 2(j-1)+1) \leq d(I(i, j-1), I(i, j)) + d(I(i, j+1), I(i, j)) \tag{5.5.12}$$

$$\leq \max(d(I(i, j-1), I(i, j)), d(I(i, j+1), I(i, j))) \tag{5.5.13}$$

$$\leq max_4(2(i-1)+1, 2(j-1)+1, M) \tag{5.5.14}$$

Similarly we can prove

$$M(2(i-1)+1, 2(j-1)+1) \geq min_4(2(i-1)+1, 2(j-1)+1, M) \tag{5.5.15}$$

□

Because the pixel values are now equal to the maximum distance between two opposite neighbors, it is possible that the distance values have become local minima. For instance, this is the case of the distance between p2 and p4 in Figure 5.5.5c. This could create superfluous CCCs. Hence, the next two equations update the distance values in order to obtain a smooth distance variation.

Smooth horizontal distances:
$\forall 1 \leq i \leq n, \quad 1 \leq j \leq m-1,$

$$M(2(i-1)+1, 2j) = \max(M(2(i-1)+1, 2j-1), M(2(i-1)+1, 2j), M(2(i-1)+1, 2j+1)) \tag{5.5.16}$$

Smooth vertical distances:
$\forall 1 \leq i \leq n-1, \quad 1 \leq j \leq m,$

$$M(2i, 2(j-1)+1) = \max(M(2i-1, 2(j-1)+1), M(2i, 2(j-1)+1), M(2i+1, 2(j-1)+1)) \tag{5.5.17}$$

The result of these equations is illustrated in Figure 5.5.5d where the color map has now become a smooth distance variation which will not create new local minima or maxima.

Finally, the diagonal distance values are computed as in Watercolor with Equation (5.5.7). Figure 5.5.6 shows the new color distance map. Notice how all the small regions of Figure 5.5.4 have disappeared.

Figure 5.5.6: Color distance map produced by Smooth Watercolor. The pixel intensity represents the distance value.

Since the pixel values are not necessarily local minima those that are on the border between two regions after the watershed algorithm are merged with the neighbor region with the closest color.

5.5.3 Post processing

Both watercolor algorithms oversegment the CCCs. The issue is that they are not capable of handling the high color variations that occur on a small scale (such as the size of a character) and are imperceptible for a human observer. In order to solve this issue we use our spatio-colorimetric color distance model presented in Section 5.4.3 to merge these regions. Since the distance parameters S_L, S_a, S_b were obtained in very approximate conditions, this post processing should be seen as a proof of concept. The post processing algorithm is described in Algorithm 5.5.1.

The principle of the algorithm is simply to merge neighboring regions with a spatio-colorimetric distance below the color threshold t. Considering how this distance was defined, we recommend $t = 2.3$. The color of a region is computed for each channel as the median of the channel values of the pixels of the region. The median is used to reduce any sensitivity to noise. This color computation is summarized on line 4.

In order to be independent from the processing order of the regions, the merging process has two steps. The first step (lines 11 to 16) lists the pairs of region to merge together *without* actually merging them. The second step (lines 17 and 18) performs the merge operations found by the first step. This independence is important as it makes the algorithm more stable. It will produce similar results even if the image is rotated or reversed.

Another issue that can happen, is when two large regions are linked by a chain of small regions. Because of the size of the small regions, all the spatio-colorimetric distances will be small. Hence all the regions will be merged together even if the large regions have very different colors. Figure 5.5.7 shows such a situation. In order to avoid this, the merging is performed by increasing region size. This explains the two for loop on lines 9 and 11.

Algorithm 5.5.1 Post-processing algorithm for Watercolor

Input: the list of region indexes *regidxlist* and the input color image I.
Parameters: color threshold t.
Output: the list of region indexes *regidxlist*

1: create empty list of colors *regcolors*
2: create list of merging rights *canbemerged* =**false**
3: **for** $i = 0$; $i \leq regidxlist.size()$; $i++$ **do**
4: $regcolors(i) = median(I(regidxlist(i)))$
5: **if** $d(regcolors(i), I(regidxlist(i))) \leq 4t$ **then**
6: $canbemerged(i)$ =**true**
7: **end if**
8: **end for**
9: **for** $i = 1$; $i \leq$ maximum region size; $i++$ **do**
10: create null merging list *merge*
11: **for all** region j with a size i **do**
12: find neighbor region of region j which can be merged and with closest color n
13: **if** $d(regcolors(j), regcolors(n)) \leq t$ **then**
14: $merge(j) = n$
15: **end if**
16: **end for**
17: merge all regions j with $merge(j)$ when it is not null
18: update *regidxlist* and *regcolors*
19: **end for**
20: **return** *regidxlist*

Finally, some regions that are produced are color gradients. In this case, the color of the region can differ significantly from the color of some pixels of the region and it would be unsafe to merge this region based on its color. This is why we use the list *canbemerged* to identify such regions. Its computation occurs at the same time as that of the region colors between lines 2 and 8.

We call the watercolor algorithms followed by the post processing: WCP and SWCP. We will now compare the results of these algorithms with the state of the art.

5.6 Comparison with the state of the art

Considering the state of the art, we will benchmark our algorithms against the Quick-Shift (QS) algorithm [VS08] and the algorithm of [FH04] denoted FH. As

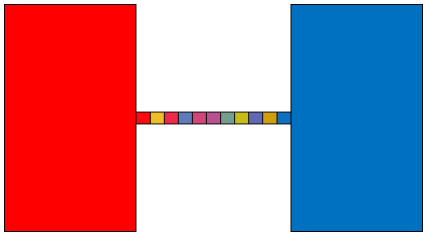


Figure 5.5.7: Chain of small regions between two large regions.

mentioned in Section 5.3, with the Mean-Shift algorithm that is improved by QS, they are the main algorithms that do not produce a fixed number of regions. The algorithms that need a specific number of regions are likely to be very difficult to adapt to document images with varying contents. Hence, they are discarded.

We will use two benchmarks: the Berkeley segmentation benchmark 500 [AMFM11] and the benchmark of Chapter 4 with the L3iDocCopies dataset presented in Section 4.4.2. For this last benchmark we will only use the normalized standard deviation of the number of regions S_R and the mean number of regions $\overline{n_R}$, defined in Section 4.4.3, because superpixel algorithms produce too many regions for the other stability performance indicators to be meaningful.

5.6.1 Berkeley segmentation benchmark 500

The Berkeley segmentation benchmark 500 is a natural scene image segmentation benchmark. It is composed of 500 natural scene images in landscape and portrait mode with a resolution of 321 by 481 pixels. No information is available on the device with which the images were taken or on its lens aperture angle. There are 200 images for training, 100 images for validation and 200 images for testing. Figure 5.6.1 shows some of these images.

Each image has several man made segmentation ground truths. These ground truths try to segment different objects of the image. Since the interpretation and segmentation of the image objects is subject to the observer's opinion, this explains the presence of several ground truths. This allows the benchmark to choose the segmentation ground truth which corresponds most to the result of a natural scene image segmentation algorithm.

This benchmark has been adapted to the evaluation of superpixel algorithms with the following performance indicators: boundary precision, recall and F-measure [MFM04], undersegmentation error [LSK+09, NP14] and achievable segmentation accuracy [LTRC11]. We use the evaluation code provided by Arbelaez and Stutz[2].

[2]available on `https://github.com/davidstutz/extended-berkeley-segmentation-benchmark`

The computation of the boundary precision (BP), recall (BR) and F-measure (BF) relies on the definition of the positiveness of conditions and predictions as explained in Section 2.3. The conditions are given by the ground truth, and the predictions by the segmentation results. Positive results are related to the presence of a boundary and negative results to the absence of a boundary. Since the algorithms may not have an absolute spatial accuracy, there is a distance tolerance to consider that a boundary pixel in the ground truth matches one in the result. We used the default distance tolerance equal to 0.75 % of the image diagonal which corresponds to approximately 4.3 pixels on this dataset.

The undersegmentation error (UE) was first proposed in [LSK+09], but it would penalize large regions. Hence [NP14] proposed the following formulation which does not have this drawback:

$$\mathrm{UE}(S,G) = \frac{1}{A}\sum_{G_i \in G}\sum_{S_j \cap G_i \neq \emptyset} \min(|S_j \cap G_i|, |S_j - G_i|) \tag{5.6.1}$$

where A is the image area, G is the ground truth segmentation and S is the proposed segmentation and $|.|$ denotes the cardinal/area function. This performance indicator evaluates the bleeding of the superpixels/CCCs across edges.

The achievable segmentation accuracy (ASA) is defined by:

$$\mathrm{ASA}(S,G) = \frac{1}{A}\sum_{S_j \in S}\max_{G_i}(|S_j \cap G_i|) \tag{5.6.2}$$

This performance indicator provides an upper bound on the accuracy that could be reached by a segmentation algorithm using the proposed partition of the image [LTRC11].

5.6.2 Results

We will now present the results of all algorithms on the Berkeley benchmark first and then on the L3iDocCopies dataset.

Berkeley benchmark

For this benchmark we used the parameters for QS and FH that seemed best according to [Stu14]. For QS we used a ratio of 0.75 between the color and geometric distances, $h = 3$ and a maximal distance of 7 pixels. For FH we chose a standard deviation of 0.8 for the Gaussian blur, $k = 100$ and a minimal region size of 25 pixels. There is no parameter to tune for our algorithm. However it does need the resolution or the angular resolution of the image to process. As we can see from the images of Figure 5.6.1, the resolution of the images of the database varies from

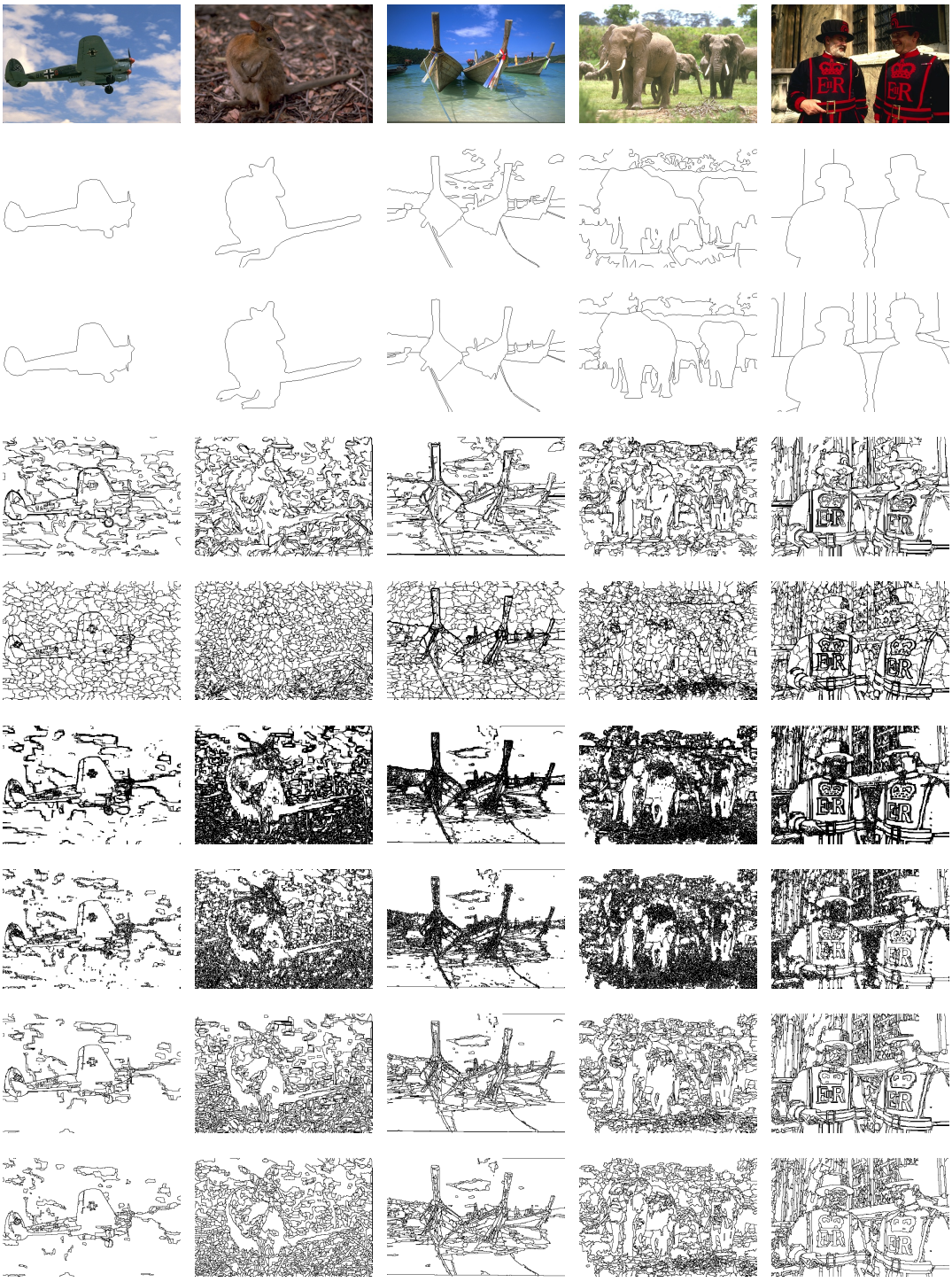


Figure 5.6.1: Segmentation results, from top to bottom: original image, ground truth 1, ground truth 2, results of FH, QS, WC, SWC, WCP, SWCP.

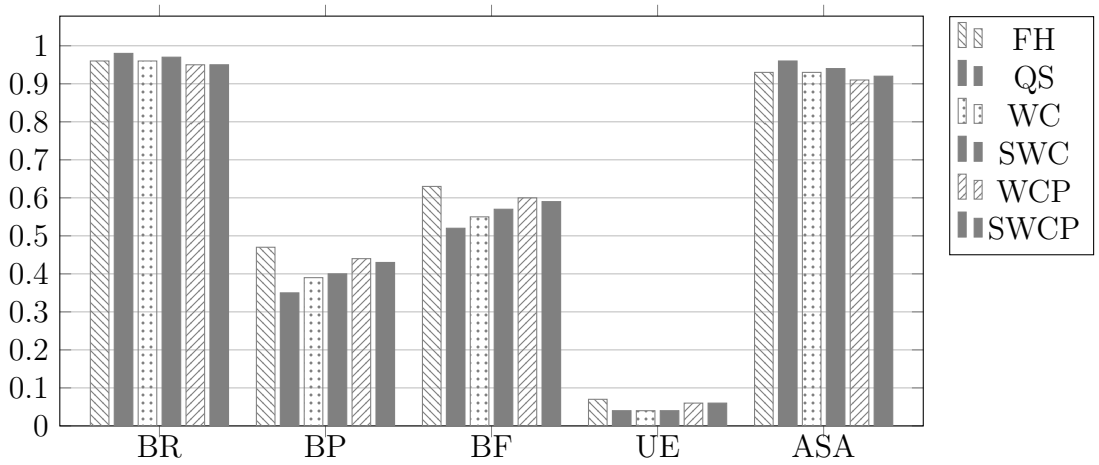


Figure 5.6.2: Performance of the algorithms on the Berkeley segmentation benchmark.

one image to the other. Since we had no way of estimating it, we chose to process the images at their full resolution and hence assigned them a resolution of 200 dpi.

The performance results of the algorithms are presented in Figure 5.6.2. We can notice that all algorithms have a very good boundary recall, undersegmentation error and achievable segmentation accuracy. The boundary precision and F-measure are much worse which is expected since, by design, the algorithms produce too many regions and too many boundaries. While not being the best, the proposed algorithms have a better boundary precision than Quick-Shift which means that they produce less superfluous edges. They also have a better undersegmentation error than the algorithm of [FH04].

The very big difference lies in the kind of segmentation produced by each algorithm. Figure 5.6.1 shows the segmentation results of the algorithms. We can already notice that Quick-Shift produces many superpixels over the entire image. This is because of its maximal distance parameter. However, increasing this parameter will lead to creating superpixels that do not respect the image boundaries anymore. When considering the images of the plane and the boats (first and third columns), all our algorithms produce cleaner segmentations than QS and FH. The pictures of the kangaroo and the elephants (second and fourth columns) are both very textured and result in many regions. The last image with the two people contains both uniform and textured areas and all algorithms (except QS) handle them properly producing many regions in the textured areas and uniform regions otherwise. We can also notice that the use of the post processing improves the segmentation results by producing cleaner segmentations.

Table 5.5 summarizes the number of regions and the processing time of the

Algorithms	FH	QS	WC	SWC	WCP	SWCP
Mean number of regions	**518**	2342	14612	12917	726	717
Approximate processing time	**0.05-1 s**	4-6 s	1-1.5 s	1-1.5 s	7-60 s	7-60 s

Table 5.5: Number of regions and processing time of the algorithms on the Berkeley segmentation benchmark. The best results are in bold.

algorithms. In terms of number of regions, SWC produces 10% less regions than WC, and the addition of the post processing reduces it by a factor 18 to 20. WC and SWC produce many regions because most of them are made of a few pixels. The post processing proves to be an intelligent way of removing all the superfluous regions (small and large). QS produces far more regions than FH, WCP and SWCP with the same performance. This confirms the fact that it produces too many regions.

Time-wise it is difficult to compare the algorithms because the one of FH is implemented in C++ while the others are implemented in Matlab. In particular the post processing is quite slow because of its loops which are not well handled by Matlab. We can notice that FH is particularly fast and that the addition of the post processing increases a lot the processing time. The large time variation of the post processing is due the strong dependence of the algorithm computational complexity on the number of regions.

Hence on this benchmark the proposed algorithms match the state of the art without any parameter tuning (actually with the handicap of not knowing the image resolutions) and produce cleaner segmentation results. Depending on the need, one may use WC or SWC for a fast processing or WCP or SWCP for cleaner results.

L3iDocCopies benchmark

No evaluation had been done for QS and FH on document images, hence we had to devise a proper set of parameters for them. For FH we kept the standard deviation of 0.8 for the Gaussian blur, $k = 100$ and the minimal region size was set to 10 pixels, once again to keep the details of the characters. These values are for processing images at 300 dpi. k and the region size units are in pixels hence we scale them with the square of the resolution ratio. When processing images at 600 dpi, QS required more memory than the one available on our computer so these images were rescaled at 300 dpi. We kept the ratio of 0.75 between the color and geometric distances and the maximal distance of 7 pixels. However, a scale of $h = 3$ was too large to keep the character details hence we set $h = 2$.

Perf. Ind.	FH	QS	WC	SWC	WCP	SWCP
S_R	0.41	0.50	**0.11**	0.13	0.15	0.15
$\overline{n_R}$	38 170	155 246	142 077	74 582	11 303	**8 783**
Approximate processing time (300 dpi / 600 dpi)	10/40 s	150 s	**15/15 s**	15/15 s	600/600 s	500/500 s

Table 5.6: Performance of the algorithms on the L3iDocCopies benchmark. The best results are in bold. QS can only process images at 300 dpi

Table 5.6 summarizes the results of the algorithms. Our algorithms clearly outperform the state of the art and are three to four times more stable. While the good normalized standard deviation of the number of regions S_R for WC and SWC could be attributed to the large number of regions which would mechanically produce a small value for S_R, this performance is maintained for WCP and SWCP which produce four times less regions than FH and QS. The proposed algorithms are also 1.5 to 2 times more stable than JSEG and we can expect that a segmentation algorithm based on the CCCs produced by one of our algorithm will be even more stable.

These results also show the impact of the improved computation of the distance map which halves the number of regions. The addition of the post processing drastically reduces the number of regions down to a number which can be considered close to ideal: one region per character plus several regions for the graphical regions. Figure 5.6.4 shows the results of the algorithms on two typical images of the dataset. Once again, FH and QS have difficulties producing relevant regions because of their inability to deal with scale variations, e.g. they either produce many small regions everywhere (small scale processing) or they loose the details of the characters (large scale processing). We also noticed that FH suffers from the same artifacts as WC on the character gradients. Those artifacts are displayed in Figure 5.6.3. This highlight the importance of properly computing the distance map.

Figure 5.6.3: Artifacts of FH algorithm. Each region has a uniform color.

Time-wise, the images of the L3iDocCopies dataset are much larger than those of the Berkeley Segmentation benchmark (up to 5000x7000 pixels at 600 dpi). Hence the algorithms take a much longer time to process them. FH takes approximately 10 s on a 300 dpi image and 40 s on a 600 dpi image which proves its linear computational complexity with the number of image pixels. WC and SWC take roughly the same time to process any image because all input images are rescaled at 200 dpi. Once again, the addition of the post processing increases these values significantly. The variation of the post-processing time is due to the number of regions produced by each algorithm. Since WC produces more regions than SWC, its processing time is longer. It should also be noted that the processing contains many loops that are not well handled by Matlab. This contributes significantly to the very long processing time.

We can now study in depth the capabilities and differences of WC, SWC, WCP and SWCP.

5.7 Analysis and applications of the proposed algorithms

The proposed algorithms have already found a few applications. We will first analyze and compare their results on some challenging images. Then we will show four application cases in pairs: firstly, edge and scale detection and secondly image level and background separation.

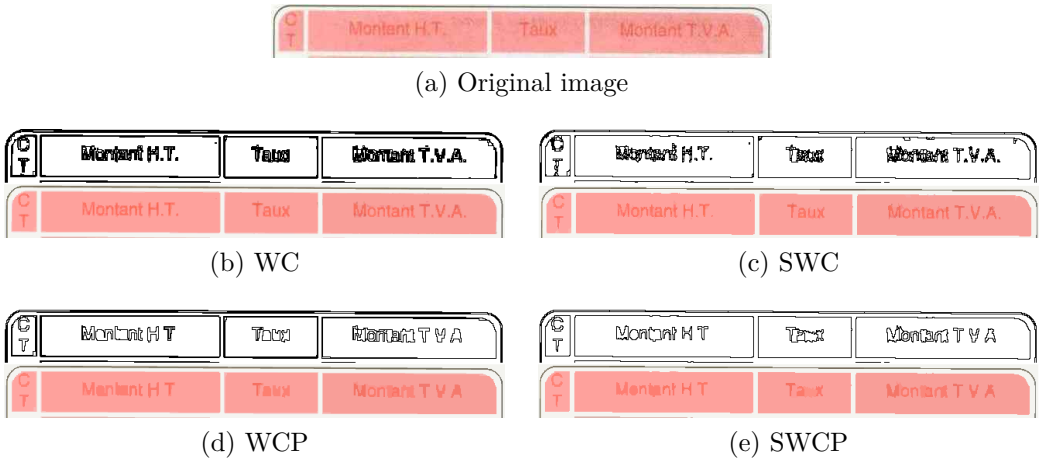
5.7.1 Analysis and comparison of WC, SWC, WCP and SWCP

These algorithms have been evaluated on more difficult cases than could be displayed here. To name a few, they perform successfully on images with textured background (with one CCC per texture element as could be expected), on gray level images, on text with any shape, direction and size. However, they do have issues with high compression JPEG noise on textured areas as it creates false edges.

Figure 5.7.1 presents the results of Watercolor on a modern administrative document, on a historical manuscript from the Saint Gall dataset [Str90, FIB+10] and on a natural scene image with a higher resolution than the ones of the Berkeley segmentation benchmark (although it was still considered to have a resolution of 200 dpi). Since the algorithms do not have any parameter, no tuning was required. The first image shows how our algorithm handles properly zones that have a color gradient and does not over-segment them. Although it is a bit difficult to see, the blue handwritten signature over blue background in the bottom left corner is preserved. More generally, all the overlapping components are well separated. The

Figure 5.6.4: Results of the algorithms on two document images. Order for each series, line-wise from the top left image: FH, QS, WC, SWC, WCP, SWCP. The images produced by QS look gray because of the many region boundaries.

Figure 5.7.1: Results of Watercolor. First column: original image. Second column: CCC boundaries. Third column: CCCs with uniform colors. Number of CCCs per image from top to bottom: 101802, 86006 and 37988. Second line original image reproduced from [Str90].



(a) Original image

(b) WC

(c) SWC

(d) WCP

(e) SWCP

Figure 5.7.2: Results of Watercolor algorithms for red text on red background.

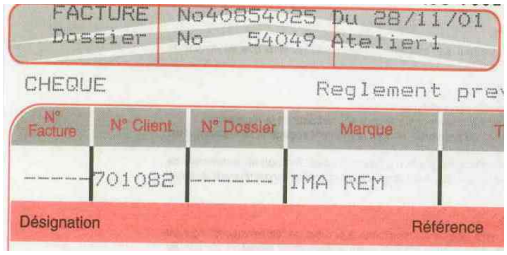
second document image shows the algorithm performance on degraded documents. It has no issue handling the paper noise and the bleed through in the top right corner. At last the natural scene image shows that the details are well preserved independently from their scale (lighthouse and rock details). Once again the sky gradient is properly handled. This also highlights the versatility of the algorithm and its ability to identify CCCs as the human eye would do it. SWC, WCP and SWCP produce similar results at this scale of viewing.

Since the blue gradient at the bottom of the first image becomes a white background after Watercolor, the white text that is overlaid on it could be merged in the post-processing. This explains the need for the *canbemerged* list of the post-processing algorithm.

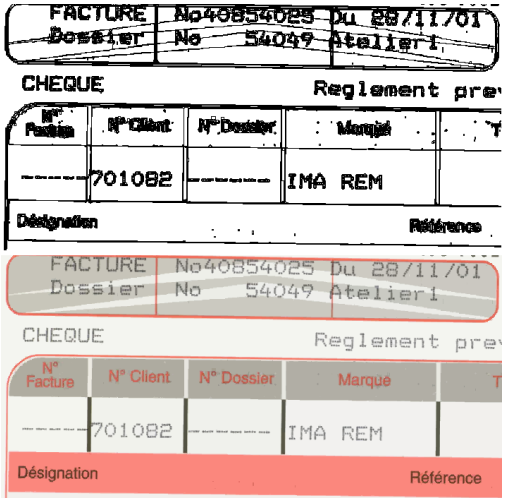
Among the challenging cases we tackled, Figure 5.7.2 shows the detail of the CCCs segmentation for red text on red background for all algorithms. They are perfectly capable of handling this kind of situation. Smooth Watercolor produces less regions and contours that are less precise than Watercolor. The addition of the post-processing also reduces the number of regions and the precision of the region contours hence it is best to apply it after WC rather than SWC which already degrades the contours. We can notice the removal of the dots by the post processing because of their small size and the color closeness.

We can see the ability of Watercolor to produce precise contours and to handle matrix/dot text printing in Figure 5.7.3. All algorithms preserve the sharp edges of the chevron and the gray matrix printed text. The post processing removes any noise regions that could remain after WC and SWC. Notice how the algorithm handles perfectly the overlapping text.

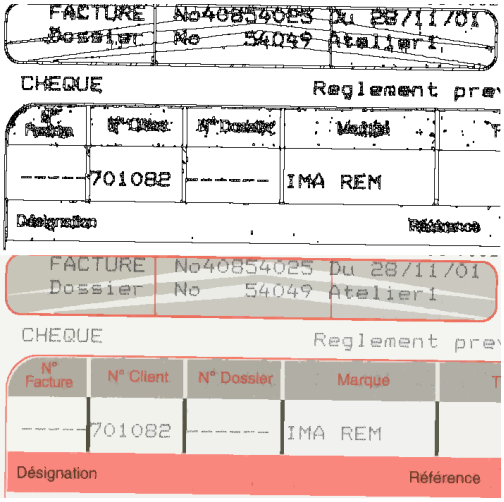
Finally Figure 5.7.4 compares text CCC segmentation for standard text. We can see that Watercolor produces better edges than Smooth Watercolor and the



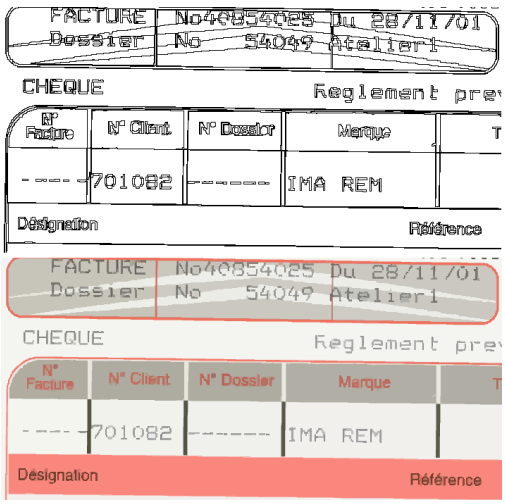
(a) Original image



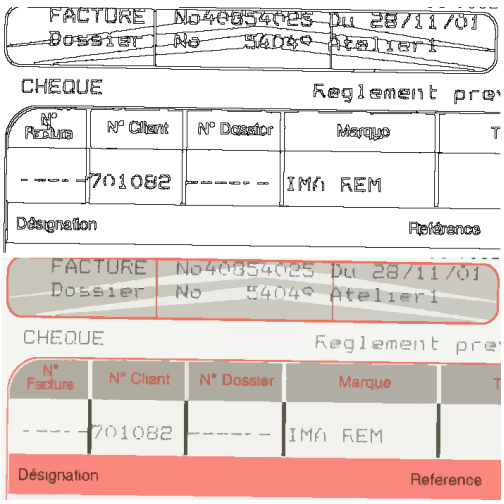
(b) WC



(c) SWC



(d) WCP



(e) SWCP

Figure 5.7.3: Results of Watercolor algorithms for matrix text and precise contours.

post processing may chip some character parts. Thus Watercolor (without post processing) is probably the best algorithm if one wants to use it for character recognition. Another solution could be to increase the resolution up to 300 dpi as suggested in Section 5.4.1.

Thus, the main issue of the algorithms is their remaining inability to reproduce the way the human eye deals with small regions as they currently split text characters into many smaller regions because of the colorimetric noise. WC is the best choice for character recognition and contour detection while SWC provides less regions and is more suitable for scale detection. For an extra cost, the post processing reduces drastically the number of regions, producing regions that are more meaningful and easier to process (because of their smaller number). On the one hand, since SWC and the post processing both degrade the quality of the contours, it is best to apply the post processing after WC. On the other hand, its use after SWC requires less computing and produces even less regions.

5.7.2 Edge and scale detection

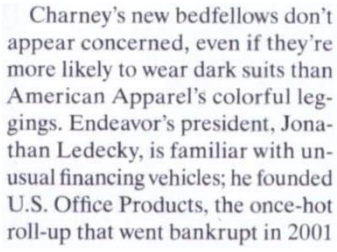
Since Watercolor algorithms identify CCCs, they identify their edges and work as a great parameter-free edge detector. This is shown in Figures 5.6.1, 5.6.4, 5.7.1, 5.7.2, 5.7.3 and 5.7.4. Another direct result is the identification of the scale of the CCCs. This scale can be determined from the size of the CCC bounding box, best fitting rectangle (including rotated rectangles) or even scale interpolation by considering that the CCCs is a disk. In this last case Equation (5.7.1) gives the formula to compute the scale form the CCC area.

$$scale = 2\sqrt{\frac{area}{\pi}} \tag{5.7.1}$$

Figure 5.7.5 shows the scale map produced by SWCP on the three images of Figure 5.7.1. The values used for the gray levels are the logarithmic value of the scale divided by 15 in order to make all values fit within the image brightness range.

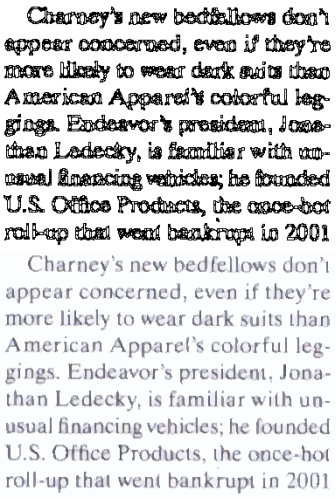
5.7.3 Level and background segmentation

Since the image CCCs are identified it is possible to number them from the outermost CCCs. This is what we call image level segmentation. Figure 5.7.7a shows the expected result. Once this is done, the regions can easily be renumbered so that level numbers only change when a region has a hole as shown on Figure 5.7.7b. Such algorithms are easy to implement and do not require explicitly detecting the holes of a region thus saving the corresponding computational cost. On the contrary they allow very fast identification of holes. The simple condition to detect a

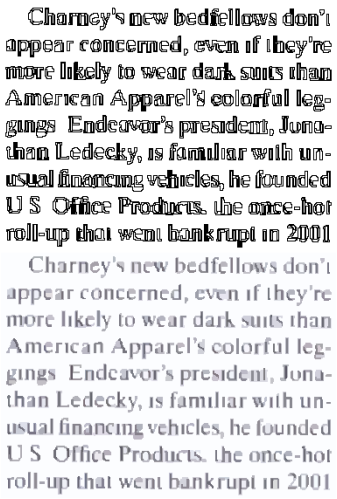


(a) Original image

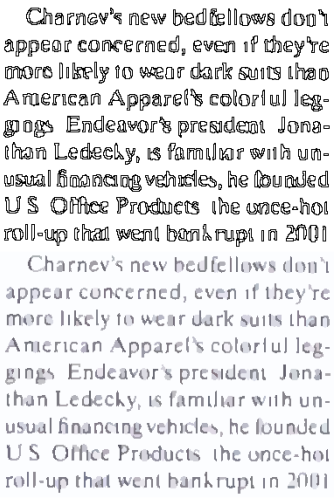
(b) WC



(c) SWC



(d) WCP



(e) SWCP

Figure 5.7.4: Results of Watercolor algorithms for standard text.

Figure 5.7.5: Scale maps produced by Smooth Watercolor with post processing. The brighter the larger the scale.

hole is that a segmented region at a given level n has only one neighbor CCC at level $n-1$.

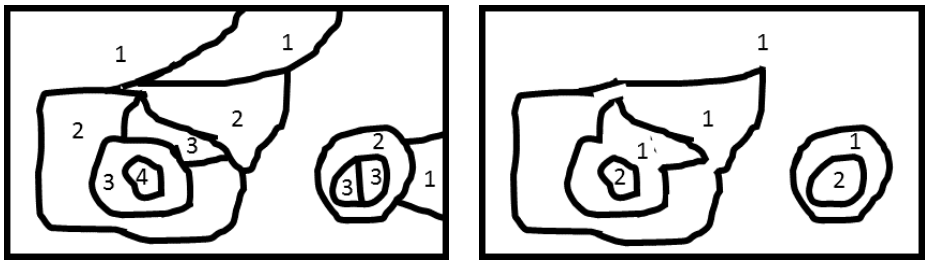
This hole level segmentation can become very convenient for binarizing/segmenting simple documents. Figure 5.7.6 shows three possible applications. The first one is the binarization of a document image with uniform text backgrounds. The black regions are the ones with a pair level number. The second one is the extraction of comics panels [Deb15]. The panels are the regions with a level strictly greater than 1. At last, Watercolor makes it very easy to isolate image components. Here several logo images have been printed with a black border around them. The logos have a level number strictly greater than 3 and their background is on level 3. This allows extracting each logo and the background of each logo.

Other applications could include vectorizing raster graphics, text and line detection and segmentation free OCR processing.

5.8 Conclusion

In answer to the issue of the stability of document image segmentation algorithms highlighted in the previous chapter, we proposed a parameter free color connected component (CCC) segmentation algorithm named Watercolor. This algorithm is based on a model of human vision which includes several characteristics previously unused in the computer vision community. We improved Watercolor's ability to handle color gradients in small complex regions. At last, we added a post processing to include the variation of the eye perception with the spatio-colorimetric distance that we proposed. This class of algorithms extends the definition of con-

Figure 5.7.6: Three application examples for the Watercolor algorithm. Results produced by Watercolor. Original image of second column reproduced from [Deb15].



(a) Standard level segmentation (b) Hole level segmentation

Figure 5.7.7: The two principles of level segmentation.

nected components from binary images to gray level and color images.

We compared the proposed algorithms with superpixel segmentation algorithms since they are the closest similar algorithms. They match the performance of the state of the art on natural scene images. When looking at stability performance indicators on document images, our algorithms clearly outperform the state of the art by a factor three to four. They are also more stable than the most stable document image segmentation algorithm that we found in Chapter 4: JSEG.

We have highlighted the large versatility of these algorithms as they are capable of dealing with natural scene images, historical documents and modern documents of nearly any kind. This feat is achieved thanks to a well posed problem (contrarily to the one of superpixels) and adopting an approach based on the observer rather than on the observed data. This makes our algorithms content agnostic.

Watercolor and its derivatives solve the problems of detecting edges and producing a scale map without any parameter. Watercolor is recommended for contour detection while Smooth Watercolor (with post-processing) is recommended for computing scales. Other direct uses include image level segmentation, image binarization, image background and layer segmentation. With further processing we foresee that they could be used for OCR processing, text and text line detection and image vectorization. OCR processing however will require very precise contours and aliasing at 200 dpi may be problematic. Thus it may be necessary to process images at 300 dpi.

The algorithms are not flawless. They have issues with high compression JPEG noise on textures. This noise could be alleviated with specific JPEG denoising algorithms. The spatio-colorimetric distance should also probably be improved and in particular the constants S_L, S_a and S_b should be measured more appropriately. Some other improvements may also be necessary in particular the use of region noise statistics in order to decide with which region should we merge a small region.

Chapter 6

Optical character recognition

Once the document regions are obtained, one of the main remaining tasks is optical character recognition (OCR) to extract the text from the textual regions. Hence this chapter focuses on the stability of OCR algorithms. We start by pointing out the extremely high level of performance that is required to have a sufficiently stable algorithm. After reviewing the state of the art we propose a simple disambiguation technique called "alphabet reduction". It is based on the principle that characters that are visually similar should be the same character. It significantly improves the stability of two state of the art OCR algorithms on almost forty three thousand images. Yet the obtained stability is still insufficient for our use case.

An optical character recognition (OCR) algorithm serves to identify the characters contained in an image. In our scenario, they are useful to extract the textual content from a scanned document image. This is a key element to the semantic signature and it is the topic of this chapter. We will focus on evaluating the influence of disambiguating characters on the stability of OCR algorithms. This chapter is organized as follows:

- Section 6.1 presents the issues to solve.
- Section 6.2 surveys the state of the art.
- Section 6.3 describes our text disambiguation technique called alphabet reduction.
- Section 6.4 evaluates the alphabet reduction.

It is completed by a conclusion.

The main contributions of this chapter are:

- The alphabet reduction, a disambiguation algorithm for OCR output which significantly improves the stability of OCR algorithms. It is presented in Section 6.3,
- A thorough study of the stability of two state of the art OCR algorithms in Section 6.4,
- The L3iTextCopies dataset, a dataset of plain text documents. It is presented in Section 6.4.1,
- The demonstration that when all other variations are taken into account (image resolution, font, font emphasis) the font size does not influence the stability of an OCR algorithm. It is presented in Section 6.4.3.

We will now properly define the problem at hand.

6.1 Problem statement

We mentioned in Section 1.3.4 that we will focus on printed text because the technology is not mature enough for handwritten text [STRV15].

Printed Latin modern text recognition is considered by many as a solved problem and most software frequently reach character accuracies above 90-95%. This corresponds to a character error rate below 5-10%. However our main performance criteria is the stability of the algorithm which, once again, has not been studied. The stability for an OCR algorithm means that the sequence of characters produced by the OCR algorithm should always be the same. Hence, either it does not make errors on copies of the same document or it makes the same errors on all copies.

A page contains approximately 2000 characters. A 5% false negative rate means that one page can be different out of 20 pages. If we assume that the OCR does not make any mistake on the other 19 pages, this implies a character error rate below 1 out of 38 000 or 0.0026%.

Besides this extremely low character error rate, the OCR algorithm should be able to use either text lines or unsegmented document images as inputs. This is due to the fact that document image segmentation algorithms can only extract text lines or text blocks. Furthermore the OCR algorithm should not rely too heavily on a word dictionary since many administrative documents contain names and item references. The other issue with dictionaries, is that the algorithm output should remain true to the original text including its errors. A dictionary based

approach brings the risk of correcting the errors contained in the document. This issue is even more severe when dealing with fraud detection. Any kind of correction could actually hide the fraudulent modification and this is not acceptable.

The available inputs are color images, gray level images, edges of image components and scale of image components. Some algorithms also use a binary input but this implies a significant loss of information which may be problematic for documents containing significant color information. The noise present in the input image is the one described in Section 1.2.

The expected outputs are a digital transcription of the text and a line or character-wise probability of the OCR being right which is commonly called a confidence measure.

6.2 State of the art

The state of the art of OCR algorithms for type written text can be divided in two main trends: the first one is that of actually creating a new OCR algorithm and the second one is that of improving an existing OCR algorithm without modifying it, for instance by adding some pre- or post-processing.

6.2.1 New OCR algorithms

There are three main classes of OCR. The first class works at the character level and thus requires a segmentation of each character. The second and current best class works at the line level without a character segmentation (but with a line segmentation). Some algorithms of this class are wrongly called segmentation free and should not be mistaken for an algorithm of the third class. The algorithms of the third class do not require any segmentation at all. Since no segmentation is done, they are more difficult to create and they have not been studied very much. However, no segmentation also means no segmentation error and potentially a more stable algorithm. Furthermore such algorithms could be capable of identifying isolated characters with varying sizes and orientations which would make them extremely versatile.

Before going through an in-depth description of the state of the art we can highlight previous comparison works. Casey and Lecolinet [CL96] do a thorough survey of OCR algorithms and highlight the current issues at their time. The basis for most modern techniques is already described in this survey except for neural networks and new segmentation-free OCR algorithms. A reference benchmark was also done the same year [RJN96] with more than 2000 pages and four million characters. The best results for binary and gray level images are reported at a resolution of 300 dpi which corresponds to our analysis in Section 5.4.1 when

taking into account image aliasing. We can also notice that accuracy increases with character frequency which may reflect a training bias. [HKP12] does a thorough comparison of two state of the art OCRs: Tesseract and FineReader on Latin and Fraktur scripts. FineReader is better at handling original Fraktur images and Latin scripts while Tesseract is better on preprocessed images. Finally, [Lop09] propose an evaluation of OCR algorithm based on the usability of their results for further natural language processing. He proposes three performance indicators based on line boundary detection, word/token recognition and tagging. The tagging is done at the page level. These performance indicators are studied with the evaluation of Tesseract.

Character based OCR

Most character based OCRs use either the raster image of the character or its contours or a set of key-points. Likforman-Sulem and Sigelle [LSS08] recognize characters with two hidden Markov models (HMM): one processing each horizontal line and one processing the vertical lines. Both HMMs hidden states are coupled by adding a dependency of the horizontal HMM on the the vertical HMM. A second dependency is added between the observed state at time t and at time t-1 to make an auto-regressive coupled model. This model outperforms the state of the art in particular for degraded and historical characters.

Another approach is proposed by [CSN05] for character recognition from PDF documents where the character fonts are embedded, but not their character code such as UNICODE or ASCII. They use a hash table built from a set of fonts. If the font is a raster font, then the raster image is hashed with a cryptographic hash. If the font describes the character contours with Bézier curves then the position of the control points of the Bézier curves are hashed. This scheme has been successfully tested on more than one billion characters.

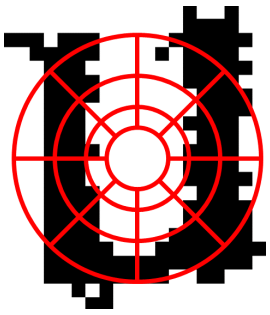


Figure 6.2.1: Typical disk quadrants used to compute a shape context. The histogram counts the number of black pixels in each quadrant.

Several algorithms make use of shape context descriptors. These are log polar

histograms as shown in Figure 6.2.1. [AS02] compute them on the raster image of Indian Kannada scripts and use one SVM per character class to classify them. Since some character classes are difficult to differentiate, these are grouped in one class group. In this case, a second level of SVM is used to differentiate between the similar classes in one class group. This algorithm relies on over-segmenting the characters because many Kannada glyphs are composed of several sub-glyphs (vowels and consonants).

Shape contexts can also be computed on character contours as done in Nabocr [SS13] which develops an OCR for Arabic and Urdu scripts. They extract the intersections of the character contour with a grid. The character image is split into four quarters and the shape context is computed for each contour intersection in each quarter. Then the shape contexts of each quarter are summed to make four shape contexts which are finally concatenated. They use a k-nearest neighbor to match characters.

Another well-known use of character contours is Tesseract [Smi07]. The contours are approximated with polygons. Then, tangents are extracted from these polygons. The characters are then classified with a two step process. A coarse step finds candidate character classes by counting the number of tangent matches in a look-up table. The second step refines this by computing a proper similarity with each candidate class prototype.

The previous OCR algorithms all use binary or contour image. Some other algorithms have been developed with key-points detectors/descriptors and thus use gray level images. [Liu12] compares several key-points (Harris corners, histograms of gradients, SIFT and SURF) and combines them with four classifiers: 5-nearest neighbor, 20-nearest neighbor, Naive Bayes and SVM. He finds that SURF with SVM is the best combination. SIFT with SVM comes second but is four times slower. Since their goal is to have a low power and fast OCR, they implement it on an FPGA and achieve 70% character performance and a processing time of 5 ms per character.

The other interest of the sparse key-point description is the ability to recognize broken characters as in [SCK15]. The SIFT descriptor is used to recognize Oriya Indian script characters. They use a brute force matching with two criterion on the descriptor distance and the key-points spatial distance. However the scale parameters of SIFT pose a problem and make the algorithm resolution and font size dependent.

SIFT is a local descriptor, thus [JQZ+09] add to it a shape context computed for each detected key-point. The shape context is computed on rings and does not contain any directional information. The ring radii are computed based on the scale of the SIFT descriptor. This allows them to keep the scale and rotation invariance of the SIFT descriptor. Their performance is slightly better than SIFT. They

highlight the fact that key-point-based approaches are not suitable to describe simple characters such as "-" or "l".

A pseudo-OCR with custom graph/key-point descriptors for Chinese text is described in [BRY+15]. Its goal is to filter spam images. It is a pseudo-OCR because they describe characters but do not recognize them. They skeletonize Chinese characters and represent them as graphs with corner key-points. Then they compute an 8-directional histogram of the graph vertices and a 12-directional histogram of intersection angles. These key-point descriptors are classified as spam or non spam with a brute-force nearest-neighbor algorithm. If more than 25% of the image descriptors are spam, then the image is considered to be spam.

Line segmentation based OCR

There are two main approaches for OCRs working at the line level: holistic approaches try to match each word as a whole and other approaches use features based on a sliding window.

Finereader OCR [Abb13] is a leading (closed source) commercial OCR with a holistic approach. Holistic approaches have also been used for Arabic script [ABH98]. In this algorithm each word is split into a set of 12 geometrical primitives. These primitives are matched with indexed primitives by a control algorithm which also produces a probability estimation of the confidence in the word recognition. This requires one threshold per primitive which is computed during the training stage. They reach a word accuracy of 94% on scanned documents.

One of the first algorithms using a sliding window was published in 1992 [CD92, CD93]. They use masks that slide over the line image to detect potential occurrences of a specific character. The masks contain a neutral zone and the foreground is manually divided in several zones which must all have minimum number of matches. A second set of features is contour Bézier curves. As the window slides, votes are cast for these features. Each feature has a weight and the corresponding character with the highest number of votes is chosen.

The majority of approaches with sliding windows use them with hidden Markov models (HMM). [LBK+99, SZGH09] use intensity based features while [JHM+10] uses discrete cosine transform. The open source OCR OCRopus [Bre08] also contains an HMM.

One of the latest OCRs uses bidirectional long-short term memory neural networks [BUhAS13] which is also included in OCRopus. It is able to recognize slightly curved text thanks to a probabilistic baseline and height estimator.

[TAA05] uses the generalized Hough transform to recognize and locate characters in a text line. The generalized Hough transform [Bal81] allows one to detect any shape such as the one of a character. Similarly to the Hough transform it builds an accumulator array with one dimension for each parameter of the shape (x

position, y position). For a given shape we can compute a reference point. The vector from each edge point to the reference point is described by the orientation of the edge gradient at this point. Thus a shape is described by the list of gradient orientations and corresponding vector coordinates. This description is indexed by the orientation value in what is called an R-table. When an edge point is found in the image, its gradient orientation can cast a vote for all the vectors (or positions of the reference point) that have the same gradient orientation. There are two dimensions for the location/vector of the character. Two extra dimensions are added for the horizontal and vertical scales. While the paper only uses it for text lines, it could easily be used for a whole page image provided that there is enough memory to store the accumulator array. This would make a fully segmentation free OCR.

Segmentation free OCR

Fully segmentation free OCR is actually some sort of character spotting similar to word spotting. The difference is that because of the size and smaller number of details of a character the same features and techniques cannot be used. The general task of character spotting should be able to find characters of any size and orientation and in any document with or without non-text elements. However this task is very complex and most algorithms focus on text only documents and sometimes limit the orientations and scales that they detect.

[Kim99] uses a set of windowed operator/filter corresponding to each character. The filter responses are then met with prototype responses with a relaxed nearest neighbor algorithm which can efficiently find a near neighbor with a modified kd-tree.

In the case of Greek characters it is possible to use character open and closed cavities to detect and recognize them [GNP+04, GNP+06, NGP+07]. The cavities are detected with a horizontal and vertical waterfall algorithm. They use fifteen features from the length and slopes of the protrusions spanning from the cavities. The characters are classified with a decision tree.

Another approach based on waterfalls is used in [PPTL10]. The reservoirs are used to segment Bangla and Devnagari text into characters. The characters are described with a directional histogram computed on circular rings and concentric convex hulls. The features are matched with an SVM. They are capable of recognizing words on a whole image no matter their orientation and with scale variations.

Another work uses key-points: [KZK13] uses a SIFT descriptor with only half the gradient directions on grid spaced key-points and an SVM classifier to recognize characters on ancient coins. The use of only half the gradient directions allows a better handling of coin shadows. They significantly outperform FineReader.

6.2.2 Improvements of existing algorithms

Creating an OCR algorithm is a very complex task. Thus it is sometimes more convenient to add some processing to improve an existing algorithm. This can also be useful for closed source OCR algorithms or to adapt a generic algorithm to a specific use case. The interested reader can find a thorough review of the state of the art for post-processing techniques in [Nik10].

Some recent pre-processing works include the improvement of training [AGC13] and improving the binarization [CRG13]. [AGC13] considers a semi-supervised training where the OCR is initially trained and then a crowd-sourcing platform improves this training. They choose the samples to be presented to the users based on their interest computed with the confusion matrix of the OCR on a validation set. Then, each newly labeled sample is only accepted if the OCR performance is improved on the validation set. In [CRG13], the authors propose an algorithm to split a text image into segments and then choose the most appropriate binarization algorithm for each segment in order to maximize OCR accuracy. They use 265 features: the mean, standard deviation and distribution skew of each color channel as well as a 256 bin color histogram from the HSV color space. Then an SVM classifier selects the most appropriate binarization algorithm.

An interesting post-processing approach is that of Kae et al. [KHDLM10]. They run Tesseract on a document to detect a set of reliable characters. Then they use SIFT as a character descriptor and an SVM classifier to OCRize the document. Unfortunately this method requires training the classifier for each document and prevents the algorithm from performing in real time. They reduce Tesseract's error rate by 20% on 10 documents.

Most generic approaches use either a lexicon, a confusion matrix or reduce the character space to remove some confusions. An early work is that of [RH75] which does a thorough study of OCR post-processing issues. Their work is based on an OCR that produces two outputs for a given input image: a text only and a digit only output. Then, they first devise an algorithm to separate textual and numerical content. The separation is done with Bayesian inference. It uses bigram (pairs of adjacent characters) contextual information for the text and the digit error probability is estimated based on the corresponding textual output. Then they use a dictionary based approach to verify and correct misrecognized words.

Reynaert [Rey08] first reduces the character space of documents and ignores digits. Then he creates a set of word variants (including anagrams) within a given edit distance of a lexicon (2 in the paper). Then each anagram is paired with its potential variants from the most frequent to the least frequent. They differentiate these pairs based on whether the anagrams were part of the original lexicon or whether there is a clear recognition error. For a given query word, the possible anagrams are first sorted by the number of possible character confusions

for each anagram, then by their edit distance with the query word and finally by the frequency of the lexicon word from which the anagram has been created. The algorithm called TICCL can detect between 55% and 89% of OCR errors on a corpus of historical Dutch newspapers.

Niklas [Nik10] combines the work of Reynaert with a new word hashing algorithm called OCR-Key. It replaces the extension process (trying all possible anagrams of a word) by a word similarity process which is more computationally efficient and the hash is based on predefined character classes. Several heuristics are then used to compute OCR corrections. They are based on occurrence frequency, dictionary words and other specifically tailored rules in particular for word hyphenation. He achieves an error reduction rate between 39% and 75% on a corpus made of several issues of The Times newspaper between 1835 and 1985.

Although this is a very specific application, a significant challenge for OCR algorithms is the recognition of mathematical expressions. The work presented in [STF+03] identifies mathematical expressions based on two methods. The first one is the detection of meaningless sequences text in the OCR output. The second one is based on recognition inconsistencies between the recognized characters and the corresponding images such as character location and size. Then, they use a specific OCR called INFTY with more than 500 character classes. It builds a graph representation of the possible transcriptions of the expression being processed. Then these possibilities are pruned based on digram position rules whose parameters are computed from a training dataset.

Another algorithm for identification of mathematical expressions has been proposed in [FYM+13] with a word bigram language model. The words whose probability of being part of normal text is too low are considered as formula candidates. Then they use the geometric properties of the character bounding boxes and their standard deviations to make a final classification with an SVM. However the performance of the algorithm is lower than that of INFTY.

Considering all the above a lot of work has been done to improve OCR performance and several interesting approaches have been proposed. Our case deals with clean printed modern documents. This is considered by everyone as a solved problem with no challenge. Thus our work focuses on studying the stability of the results of OCR algorithms and solving some ambiguities.

6.3 Alphabet reduction

We propose a simple post processing algorithm whose goal is not to correct OCR errors but rather to disambiguate its results. This leads to a problem that is better posed and is easier to solve. Contrarily to most works who focus on the algorithm and the text, we focus on the observer and what is meaningful for him. Hence our

Character	Replacement
Empty line	Removed
Tabulation and space	Removed
— (long hyphen)	- (short hyphen)
’, ‘ (left and right apostrophes)	’ (centered apostrophe)
”, “, ” (left and right quotes, double apostrophe)	" (centered quote)
I, l, 1 (capital i, 12^{th} letter of the alphabet, number 1)	|(vertical bar)
O (capital o)	0 (zero)
fi (ligature)	fi (two letters f and i)
fl (ligature)	fl (two letters f and l)

Table 6.1: Alphabet reduction

approach is unsupervised and based on human vision.

When reading a password displayed on a screen it is frequently difficult to differentiate an O from a 0 or an I from an l. Furthermore, replacing one of these character by the other does not change the readability of the text but for reference numbers and other codes in which case there will also be an ambiguity for a human reader. Thus it seems pointless to ask an OCR algorithm to differentiate these characters. Furthermore, they introduce a visual ambiguity which will significantly reduce the stability of the output of OCR algorithms.

To remove the ambiguities contained in OCR algorithms we decided to apply what we call an alphabet reduction. Table 6.1 shows the character classes that are projected onto the same character. Because it is difficult to identify the number of empty lines and sometimes the spacing between lines, we decided to remove them. Once again this does not change the readability of the text. Similarly, differentiating between tabulation and a certain number of space is difficult thus they are removed. A similar principle is applied for the other characters that are visually difficult to distinguish.

This alphabet reduction could lead to projecting two different words onto the same word which could introduce a possibility for undetected modifications of the textual content of a document. To study this we applied the alphabet reduction to the Aspell English dictionary. The only collisions where those due the confusion between an I (a capital i) and an l (a lower case L). In such cases, the confusion is possible but would not make a meaningful sentence. If we take the font case into account we obtain a collision probability of 0.0002. This guarantees that the level of security of the system is preserved.

6.4 Evaluation of the alphabet reduction

We compare the performance of two state of the art OCR: Tesseract and Finereader Engine 11 without and with alphabet reduction. For both algorithm we use the initial English training provided with them. For this we will first present the test dataset, then an additional performance indicator and finally the evaluation results.

6.4.1 Testing dataset: L3iTextCopies

Considering the quite stringent requirement for the OCR accuracy, we chose to use clean, text-only, printed documents. Both OCRs can analyze documents with a single or double column layout, so we tested it with a combination of these. We used only and all the characters that they both can recognize (the limitation comes from Tesseract).

The dataset is made of 22 pages of text with the following characteristics:

- 1 page of a scientific article with a single column header and a double column body
- 3 pages of scientific articles with a double column layout
- 2 pages of programming code with a single column layout
- 4 pages of a novel with a single column layout
- 2 pages of legal texts with a single column layout
- 4 pages of invoices with a single column layout
- 4 pages of payslips with a single column layout
- 2 pages of birth extract with a single column layout

We created several variants of these 22 text pages by combining:

- 6 fonts : Arial, Calibri, Courier, Times New Roman, Trebuchet and Verdana
- 3 font sizes : 8, 10 and 12 points
- 4 emphases : normal, bold, italic and the combination of bold and italic

This makes 1584 documents. We printed these documents with three printers (a Konica Minolta Bizhub 223, a Sharp MX M904 and a Sharp MX M850) and scanned them with three scanners and at different resolutions between 150 dpi and 600 dpi as shown in Table 6.2. This makes a dataset of 42768 document images. Figure 6.4.1 shows an example of the images contained in the dataset.

Scanner	150dpi	300dpi	600dpi
Konica Minolta Bizhub 223		X	XX
Fujitsu fi-6800	XXX	X	
Konica Minolta Bizhub C364e		X	X

Table 6.2: Scanning resolution for each scanner, one "X" per scan

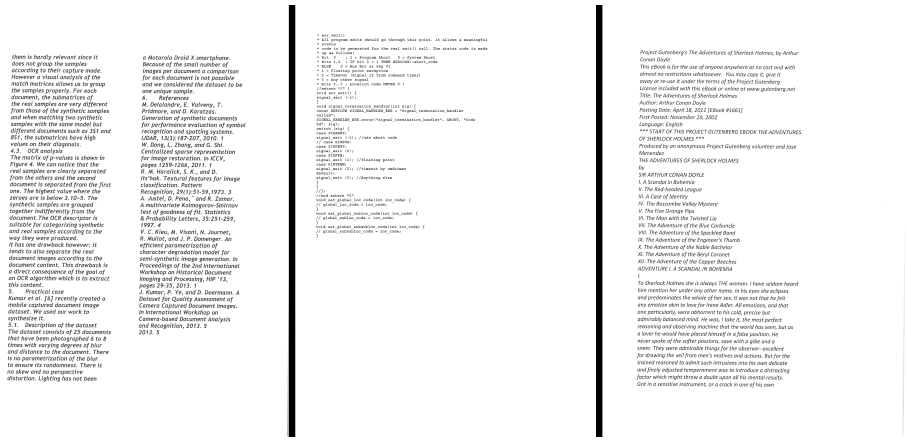


Figure 6.4.1: An example of three document images of the dataset

6.4.2 Performance indicators

In order to use the stability performance indicators (FNR, FPR, FOR and FDR) presented in Section 2.3 we need to define the input and output similarity functions. The input similarity function, s_1, is the indicator of whether the images are copies of the same document at the same resolution. Thus to be considered identical images will need to have the same text, font, font size, font emphasis and resolution. Such a precise criteria will allow us to study the impact of each parameter (font, font size, font emphasis and resolution). The output similarity function, s_2, is the binary comparison of the OCR outputs.

A common performance indicator used for the evaluation of OCR performance is the character accuracy which is defined by:

$$acc = \frac{nc_{cr}}{nc_t} \tag{6.4.1}$$

where nc_{cr} is the number of correctly recognized characters and nc_t is the total number of characters. In order to use objective values that tend towards 0 we will use the error rate which is defined by

$$err = 1 - acc \tag{6.4.2}$$

		Original algorithm			With alphabet reduction			
Perf. Ind.	Algorithms	150 dpi	300 dpi	600 dpi	150 dpi	300 dpi	600 dpi	BCS
FNR (%)	Tesseract	89	87	85	88	70	65	50
	Finereader	**76**	**70**	**68**	**62**	**50**	**48**	**31**
FOR (%)	Tesseract	0.019	0.018	0.018	0.019	0.015	0.014	0.010
	Finereader	**0.016**	**0.015**	**0.014**	**0.013**	**0.011**	**0.010**	**0.007**
err (%)	Tesseract	8.7	1.6	1.5	8.1	0.7	0.6	**0.2**
	Finereader	**1.5**	**1.1**	**1.2**	**0.8**	**0.5**	**0.6**	0.3

Table 6.3: Performance of the OCR algorithms and of the alphabet reduction. "BCS" stands for "best case scenario". The figures in bold are the best results between the two algorithms.

To evaluate the OCR error rate we need a ground truth for the original text. This is easily obtained from the original documents. In the case where we use the alphabet reduction as a post-processing step for the OCR output, we will also apply it on the ground truth so that the evaluation is consistent.

6.4.3 Results

Table 6.3 shows the results for both OCR algorithms without and with the alphabet reduction post processing. The text variations are such that there are no false positives and the false positive and discovery rates (FPR and FDR) are both always equal to 0.

Clearly, Finereader is better than Tesseract at all resolutions. Tesseract has definitely more difficulties dealing with images at 150 dpi as its error rate raises up above 8% while it remains below 2% at the other resolutions. Finereader on the other hand has a very slightly higher error rate at 600 dpi than at 300 dpi. Both facts are likely related to the default algorithm training being performed on images at 300 dpi for both algorithms. In the case of Tesseract, this may also come from a technical limitation. Tesseract uses a contour approximation algorithm to recognize characters. It is possible that at 150 dpi this algorithm is not able to approximate the contours appropriately because of the low resolution.

The alphabet reduction reduces the FNR by approximately 20 points, e.g. from 84% to 65% for Tesseract at 600dpi. The improvement is lower at 150 dpi in particular for Tesseract. This is because the error rate is higher at this resolution and thus there are more sources of instability that are not related to recognition ambiguity. The alphabet reduction also reduces the error rate by 50%, once again except for Tesseract at 150 dpi.

The FOR is very low and does not vary much because of the overwhelming bias

of the dataset towards negative matches. This is normal since for each document there are 9 copies/positive conditions and $4751 \times 9 = 42759$ different images/negative conditions. This bias will increase with the dataset size.

We can also notice that while increasing the resolution from 300 to 600 dpi does not change the error rate much, it reduces the FNR for both algorithms. This shows that both algorithms don't make less mistakes but make mistakes in a more repetitive manner.

A more detailed analysis of the errors shows the following facts:

- The font size (in pts) has a similar effect as the resolution.
- The font size (from the font design) can also reduce the performance, in particular Times New Roman and Courier.
- Italic poses problems. This is due to the ambiguity between "/" and the italic "I" (capital i).
- The pages of code are badly recognized. This is because they have more out of dictionary words and an unusual syntax.
- The dotted lines in forms are badly recognized for the same reason.

Thus we devise a more reasonable best case scenario without italic emphasis, with a resolution of at least 300 dpi, with a font size of at least 10 points, with all fonts but Times New Roman and Courier and with no pages of code. In this case we reach an FNR of 30% with Finereader. While this is far from the goal of being below 5%, this is also far better from the initial 75-90%.

Since the OCR algorithms focus on the text content without taking into account the font type, size or emphasis or the image resolution it should be possible to have an input similarity function that does not take these into account either. In order to study the influence of taking into account each parameter we have incrementally relaxed the input similarity function. Table 6.4 shows the FNR variation with the similarity function and criteria for both algorithms with the alphabet reduction.

As we can see, the more we relax the similarity criteria, the more unstable the algorithms are. This is because there are more errors for each set of images of a same document. Thus there are more combinations and more instability possibilities. However, it seems that relaxing the font size does not increase the instability by more than 1%. This is probably linked to the low error rate, but nevertheless it shows that evaluating the stability of an OCR on one font size could be sufficient provided that the other parameters are varied enough. Considering our best case scenario and usual character sizes, a size of 10 points could be a good choice.

	Tesseract		Finereader	
Stability with respect to	All cases	BCS	All cases	BCS
Printer and scanner	74	50	53	31
+Resolution	90	72	62	34
+Emphasis	96	77	69	40
+Font	98	83	75	46
+Font size	98	84	76	46

Table 6.4: Influence of input similarity criteria on the FNR performance of the OCR algorithms with alphabet reduction. Values are in percentage. "BCS" stands for "best case scenario".

When looking at the FNR variations, we can also notice that it varies more in the general case than in the best case scenario and Tesseract's performance degrades twice more easily than the one of Finereader: Tesseract's FNR increases by 34 points in the best case scenario while the increase is only of 15 points for Finereader. This contributes to showing that Finereader has a superior stability.

6.5 Conclusion

In this chapter we have studied the stability of OCR algorithms with respect to several parameters: the font, the font size, the font emphasis, and the image resolution. To this intent we have produced a dataset with 42 768 images of English texts.

Producing a stable algorithm requires removing as many ambiguities as possible. Thus we have proposed an alphabet reduction post processing on the output of any OCR algorithm. The main idea behind this is to have a well posed problem and a reasonable challenge. In the case of OCR processing, we base our approach on a study of the observer and of human character recognition. This allows us to have an algorithm whose performance is not influenced by the content being processed. We propose that similarly looking characters should be considered as the same character. This simple post-processing halves the character error rate and reduces the FNR by 20 points. To our knowledge no other post processing achieves these results while being content agnostic. This post processing is currently limited to Latin characters but we expect that it can be extended in the future.

We have benchmarked two state of the art algorithms: Tesseract and Finereader. Both are limited by the FNR performance. Finereader clearly stands out as the most accurate, versatile and stable algorithm. If the only image variation allowed is the printer and scanner hardware, it reaches an FNR of 31%. This goes up to

46% when every variation is allowed.

The performances presented here are obtained on a fairly clean dataset as shown by the very low error rate (below 1%). Thus the algorithms may be less stable on noisier images.

The last consequence of this study is that the issue of having a high quality stable OCR for printed English text is far from being solved, despite the common belief that this issue does not present a challenge anymore. The current best way to extract text in a stable manner is to use FineReader and apply the proposed alphabet reduction on it. It works best if the text is not in italic and for character sizes above the one of the Arial font at 10 points. Text should also be scanned at a resolution of at least 300 dpi. Also text with unusual syntax such as programming code will yield worse results.

Chapter 7

Perceptual image hashing

Apart from the text, the other significant contents of a document are the logos and the handwritten signatures. These graphic elements are usually described with the means of what is called perceptual hashing. This kind of algorithm produces digests between which a distance can be computed. This is necessary because of the undefined nature of what is significant in a graphic element. We start by introducing the desired properties of such a hashing algorithm and then we review the state of the art. A naive approach could be based on detecting key points in the image. However we show that such a description is not precise enough. We also show that the spatial relationship between the key points provides more information than their descriptors. Next, we introduce a new perceptual image hashing algorithm which relies on simple image transformations. Its subtlety lies in how these transformations are applied to handle print and scan noise as well as to provide a maximum stability. When compared with the state of the art on nearly forty five thousand images, the proposed algorithm performs much better in particular to detect image differences.

Perceptual image hashing extracts the robust features from the image to generate a compact representation, the so-called digest/hash. This hash can then be encrypted to make a signature. One can compute a similarity measure between two image digests or signatures to verify if these images are similar or not.

Robust or perceptual or content based image hashing was introduced by Schneider and Chang [SC96]. Figure 7.0.1 shows the general process to perform a perceptual image hashing.

In our case perceptual image hashing is useful for the graphical parts of the document such as the handwritten signatures, the logos, the schematics and the diagrams. It allows us to compute a compact digest which can then be used to

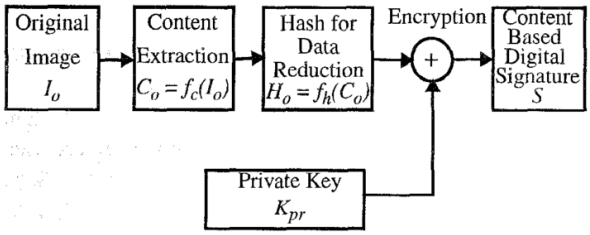


Figure 7.0.1: The process to compute a content based hash. Image reproduced from [SC96].

authenticate these elements. We will only focus on handwritten signatures and logos as we established in Section 1.3.4 as they are the most frequent types of graphics in administrative documents. Furthermore, if we can make an algorithm whose functioning is generic and not based on the specificities of the content being processed, we can expect it to work on other graphics within the algorithm's operational range.

This chapter is organized as follows:

- Section 7.1 presents the challenges and requirements of perceptual image hashing algorithms.
- Section 7.2 presents the state of the art of perceptual image hashing algorithms.
- Section 7.3 studies the interest of key-point-based approaches for perceptual image hashing.
- Section 7.4 presents the proposed hashing algorithm, its matching algorithm and the decision making process to compare an image and a digest.
- Section 7.5 compares the proposed algorithm with two state of the art hashing algorithms.

These sections will be completed by a conclusion.

The contributions of this chapter are:

- A study of the interest of key-point-based approaches and the demonstration that key points relative locations are more discriminative than their descriptors in Section 7.3,
- The ASYCHA perceptual image hashing algorithm and its matching algorithm which outperform significantly the state of the art. They are presented in Section 7.4,

- A study of the stability of two state of the art perceptual image hashing algorithms and of the proposed one in Section 7.5,
- The L3iSignCopies and L3iLogoCopies datasets of photocopies of handwritten signatures and logos in Section 7.5.1.

We will now present the issues related to perceptual image hashing.

7.1 Problem statement

Some important requirements are expected from a perceptual image hashing function [SHKY04, SZ06, LWH11, ASU10, ZHKY10, Smo12] :

Robustness: The digest should be (nearly) invariant to accidental changes, such as image compression, geometric distortions (rotation, translation, zoom), adding unintentional noise (photocopying process), modification of the brightness or contrast.

Fragility: The digest should allow the separation of visually different images.

Security: The digest should resist to attacks seeking to pass authentication with a falsified image. It should also allow to locate the tampering.

Confidentiality: It should be impossible to recover the content of the original image from the digest.

Compactness: As a digest does not convey any information other than the integrity of the image, but it needs to be transmitted with it, it should not be too large. Otherwise, it may be impossible to embed or to transmit it and the storage cost may be too high for a large number of images.

In our case, the goal is to properly identify the legit copies of the same images and the modified copies of these images. A copy is considered modified if it differs significantly from the original image. We have already stated that the main issue is the print and scan process which introduces some noise. Smoaca's thesis [Smo12] is dedicated to this issue for the case of identity photographs and print and scan copies. It provides a thorough analysis of the problem to solve and of the state of the art.

An image is a very complex set of information and it is difficult to pinpoint what is really meaningful in it and what is not. What level of color modification, of intensity modification is significant? What is the minimal size of a significant modification? Considering the state of the art of Section 1.3.3, we can use the minimal size sensitivity that is defined by the SIGNED project: a square of 42 by 42 pixels at a resolution of 600 dpi. For the rest we will resort to making sure that our algorithm performs well on a very challenging dataset. This dataset will be described in Section 7.5.1. Its main characteristic will be to contain print and scan noise as described in Section 1.2.

Performance-wise, we would like it to have all false negative, positive, omission and discovery rates (FNR, FPR, FOR and FDR, defined in Section 2.3) below 5% or reasonably close to it with a digest size below or around 500 bytes. We mentioned in the introduction that a typical maximal digest size is of 1.6kB. This is not an issue for the layout and the text because they make use of cryptographic hashing. However, this is not the case for perceptual image hashing. Obviously the smaller the digest the better, but considering the uncertainty about the meaningful content of an image we will have to use a fuzzy hash algorithm which can be quite large. A typical document has some text, a logo and a handwritten signature. If we use 500 bytes for the logo and 500 bytes for the signature, that leaves 600 bytes for the text, the layout and the signature data (certificate, etc.). This seems to be a reasonable size distribution.

Finally our problem should not be confused with near duplicate image detection or other perceptual hashing schemes designed to retrieve similar looking images even though they may not be identical. Here we only want to find the exact same images modulo the print and scan noise. This is a difference of similarity function for the input space, s_1.

7.2 State of the art

Since the work of Schneider and Chang [SC96] much research has been done. It mostly focuses on the robustness of the hashing e.g. its ability to recognize identical or similar images. This means that only few works study the ability of the hashing algorithm to detect small modifications of the image [SC96, LL03, LC01, LC98, AAK08, ASU10, LU08, ME06, Que98, LH05, MM07]. Furthermore, most of them do it for large modifications or on a small number of images.

Another remark is that the large majority of the works only deals with images of a small predefined size. Thus they are not capable of capturing all the details of a large image. This is an issue similar to the one we encountered in Section 5.6.2 where the images of the Berkeley dataset were five to ten times smaller than the document images that we have to deal with. It was not an issue then because the superpixel/CCC segmentation was no used directly for authentication purposes. However, this is a serious issue with perceptual image hashing.

The different perceptual image hashing techniques can be roughly classified into the following categories based on their approach [LWH11, ME06, LL03, Smo12]:

- Coarse representation-based approaches
- Statistical approaches
- Relationship-based approaches

- Sparse feature-based approaches
- Matrix factorization-based approaches

There are also some generic works that we will review at the end.

7.2.1 Coarse representation-based approaches

In these approaches the hashes are extracted using the raw information of the image. They show a good robustness to imperceptible changes, but can be vulnerable to geometric distortions and are likely not to be robust to print and scan noise.

Wavelet decomposition is a very popular content extraction method [CWLW98, ASU10, LU08, YC05, MV02]. The algorithm of [LU08] is based on the JPEG2000 compression algorithm which encodes the coefficients of the wavelet decomposition into code blocks. The first code blocks are concatenated until the required hash size is reached. They further improve this algorithm by using a private key to compute the filters of the wavelet decomposition. This algorithm detects successfully digital tampering on digital images but has a reduced robustness.

[ASU10] also makes heavy use of cryptographic techniques. They first divide the image into blocks of 16 by 16 pixels. Then they permute the image pixels inside each block with a randomizing algorithm whose seed is the user private key. The resulting image is then transformed into the wavelet domain. After a linear combination of the wavelet sub-bands they are randomly permuted again to make the final hash. They introduce a very interesting quantization scheme. They quantize the linear combination of the wavelet sub-bands and hash them with SHA1. Then, they add to it a perturbation vector which allows the receiver of the image to identify acceptable quantization variations. With this vector, it is possible to match the hash of the image after a print and scan process with the original one even though its is a cryptographic hash.

Other frequency based approaches use the low frequencies of the discrete cosine transform (DCT) [FG00]. [STC05] uses all the quantized DCT coefficients with a quantization correction code similar to that of [ASU10]. One can also use a quantized polar histogram of the Fourier transform [SMW06].

A few approaches use the Radon transform (e.g. the Hough transform for a line) [WGFJ98, LML02, LCM03] which makes it easy to handle image rotation and scaling. [LCM03] uses the two main eigen vectors of the covariance matrix of the columns of the Radon transform. Two signatures are matched by computing their cross correlation and comparing its maximum with a threshold. Unfortunately this method and the other methods using the Radon transform do not handle image translations.

Fridrich [Fri99] uses an interesting approach based on generating random matrix zero-mean filters and applying them to blocks of the image to hash. The values of

the filtered blocks are then binarized based on the positiveness to make the final hash. It is robust to a wide range of distortions, but its fragility is not studied.

7.2.2 Statistical approaches

They are similar to coarse representation-based approaches, but they rely less on local information in order to increase their robustness. The hashes are extracted by calculating statistics such as mean, variance, higher moments on blocks of the image or histograms. These techniques are made to be robust but usually lack the necessary fragility to detect tampering.

Most methods compute moments. [KN01] computes the variance of image block for several levels of JPEG compression. Then it finds the largest distance between these variances and the one of the original image to obtain the distance threshold used to verify the image. They also extend this algorithm to other tilling approaches similar to superpixel segmentation. [YGN06] compute the mean of image blocks for the original image and several rotated versions of it. They also add several randomized permutation steps.

[VKJM00] propose a method that is less computationally expensive. They first extract the Haar 3-level wavelet sub-bands of the image. Then they make a random tiling of each sub-band as shown on Figure 7.2.1. The coarse sub-band is used to compute mean values and the other sub-bands to compute standard deviations. These values are then randomly quantized which allows the use of a private key. The random quantization simply works by drawing the quantization threshold from a random number generator whose seed is the private key. Finally they use two stages of error-correcting codes to make the hash more robust and more random. They use the Hamming distance to compare two hashes.

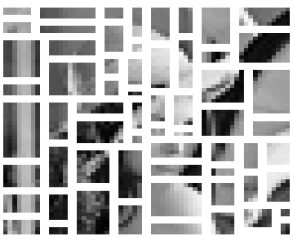


Figure 7.2.1: A random tiling of the coarse sub-band of Lena. Image reproduced from [VKJM00].

Other methods use cumulants (a kind of statistics similar to moments) combined with DCT and quantization [YSBS05]. [DDLM05] also computes a DCT but on the variance of the radial histograms of the input image. The early work of [SC96] was also using plain histograms of blocks of the input image. With a small block

size, this algorithm detects successfully digital tampering on digital images, but it is not robust.

7.2.3 Relationship-based approaches

These techniques try to establish relationships or relative relations between the data obtained with the previous techniques. There are two main trends to do this. One is with pairing images features and the other one is with a multiple stage processing usually with a Radon transform followed by a frequency based transform such as DCT or FFT. The second and further processing stages are used to compute relationships between the features produced by the first stage (Radon transform).

Pairing based algorithms usually rely on a random number generator with a private key to generate the sequence of blocks to pair and the other algorithm parameters. In [LC98, LC01] the image DCT is divided in blocks which are paired and then a subset of the DCT coefficients in each block are compared to make a final binary value. [AMVK99] uses a similar approach but performs the blocking in the spatial domain. [AAK08] extends this with the pixel intensity values, a random pairing and a more generic signal based approach.

[LL03] use a wavelet decomposition and pairs the coefficients at a given scale with those at the scale directly below if their difference is larger than a user-defined value. Histogram bins have also been paired in [XKH07].

Regarding the works that use a multiple stage processing, [SHKY03, SHKY04] use the auto-correlation of the Radon transform to make it translation invariant. Then they use a log-mapping (e.g. a logarithmic coordinate scale) and a Fourier transform to make it scale invariant. The DCT transform has also been used [OR09]. They apply a 1D-DCT transform on the Radon transform along several directions. Only the second coefficient of each DCT is kept after which they are quantized.

A very interesting work is the one of [WZN09]. It is one of the very few to test a hashing scheme with real print and scan noise. They first compute the Radon transform and resize it to 40 by 20. Then they compute a Haar wavelet decomposition and the Fourier transform of its high frequency coefficients. The real part of the Fourier transform is finally compared to its mean to make binary values. They use the Hamming distance to compare two hashes. Their hashing scheme is very robust and is also capable of discriminating different images including images of faces. They make no authentication error on their dataset. No tampering detection was tested.

7.2.4 Sparse feature-based approaches

Instead of trying to hash all the data contained in the image, some algorithms focus on a reduced set of points or edges. These are either taken from standard key-points or edge detectors such as Harris [HS88] or custom made.

Among the algorithms based on edge detection [Que98] uses a Sobel or a Canny edge detector whose result is then thresholded to make a binary edge map. This map is then sub-sampled and compressed to make the final signature. Two signatures can be compared with the Hamming distance. This is similar to the approach proposed in [DSS99].

The first works based on key-points used custom made key-point detectors. [BK98] use a Mexican hat wavelet. Then they compare its response between two consecutive scales of the input image and select the points whose response difference is above a given threshold. These points are further pruned based on the local variance. The signature is the set of positions of the key points. Two signatures are compared point-wise. This mode of comparison is the same for all other key-point-based algorithms unless precised otherwise.

[ZKPF99] use the local maxima of the image gradient to detect points that lie at the intersection of two lines. This intersection is verified by the number of sign changes on a circle around each point.

In, [KSNO03] the DCT is used to make a low pass filter by only keeping its low frequency coefficients. The pixels whose difference between their low pass filtered value and their original value is above a given threshold are the key points. They perform this key-point detection at several JPEG compression levels and keep only the points that appear for all levels. Based on this, they compute an error margin on the total number of detected key-points. During the matching, as long as the number of matched key-points is within this margin, the images are a match.

A similar filtering technique is used in [LHSC04, LH05], but they replace the DCT by a wavelet transform. This is similar to the edge preserving filter proposed by [Fat09]. After this step they use the Harris detector [HS88] to detect key-points. Next, they compute the Delaunay triangulation of these key-points to separate the image into triangles as shown on Figure 7.2.2. Then they normalize each image triangle with an affine transformation that increases the largest angle to become a right angle and that scales its edges to make an isosceles triangle. The hashing continues by making a square composed of the normalized triangle and its transpose. This square is further divided into blocks. They extract the first non constant coefficient of the DCT of each block and binarize this sequence to make the final hash. The hash of an image is composed of the hashes of all its triangles. Two images are compared by comparing the hashes of their triangles. They also devised a coarse matching procedure to increase the processing speed.

Monga et al. [MVE05] also use wavelets to detect key-points. They use a special

Figure 7.2.2: Triangle tessellation of Lena image obtained by [LHSC04, LH05]. Image reproduced from [LHSC04].

wavelet called end-stopped wavelet which is a 2D wavelet defined by the mother function:

$$\psi(X) = \exp^{-\frac{1}{2}|X|^2} \left(\exp^{jK.X} - \exp^{-\frac{1}{2}|K|^2}\right) \tag{7.2.1}$$

where $X = (x, y)$ is the local coordinate vector and $K = (k_0, k_1)$ is the wave-vector which determines the scale and direction of the wavelet. They compute the third level wavelet transform of the image for several resolutions and directions. The key-points are local maxima for all directions that are also above a given threshold. Two images are matched by finding the perspective distortion that minimizes a modified Hausdorf distance between the images. The Hausdorff distance is modified to reduce its sensitivity to outliers such as points that could not be matched because they have no counter-part in the other image.

They improve this algorithm in [ME06] by adding a random tiling of the input image similar to the one of Venkatesan [VKJM00]. They also add a random quantization of the output. The matching is the same, but they use a Hamming distance since their output is binary. They evaluate their work with synthetic noise and digital modifications. This hashing algorithm performs well on a database of 1000 images.

[YS07] combine the detector of [ZKPF99] and the matching algorithm of [MVE05]. They benchmark this on a dataset of 1002 images that were printed and scanned and show that the proposed algorithm is robust to the noise introduced by the print and scan process.

7.2.5 Matrix algebra-based approaches

This is a completely separate field of perceptual hashing. The hashes are computed with matrix operations.

[KVM04] uses singular value decomposition (SVD) on tiles of an image computed similarly to [VKJM00]. The SVD vectors are then concatenated to make a new image on which the whole process is repeated again. They also replace the first SVD by a DCT. Both versions of the algorithm are evaluated on 5000 images with

synthetic noise. They properly identify the image copies while separating them from the other images.

Monga and Mihçak [MM07] show that non negative matrix factorization (NMF) is more precise than SVD while maintaining a good if not better level of robustness. Thus they devise the same algorithm as [KVM04] but replace the SVD/DCT by NMFs. This algorithm is completed by a final projection step where the hash obtained so far is projected onto a set of Gaussian random vectors to reduce the size of the hash. This algorithm performs much better than the one of [KVM04] on a dataset of 10 000 images with synthetic noise. They also provide a theoretical analysis of the performance of their algorithm.

Smoaca [Smo12] designs an interesting hashing scheme based on independent component analysis (an improved kind of principal component analysis). Basically, they use it to detect template features from a training set of face images. Then an incoming image is projected onto these features and the resulting vector is binarized. This performs very well on printed and scanned images, but it can only process the kind of images for which it has been trained which makes it not versatile enough in our case.

7.2.6 Generic works

Apart from these approaches we can note the works of [MBE03, CLW+99, CZ01] who focus on the quantization scheme in order to reduce, make more robust and keep the precision of the digest. They devise new clustering algorithms that adapt to the distribution of the vectors or graphs produced by the techniques above. The signatures values can then be replaced by shorter ones associated to the clustering scheme.

From a more generic point of view, Zhu et al. [ZHKY10] made an interesting theoretical study of quantization based image hashing techniques and [VKBP09] made a theoretical study of the limits of general perceptual hashing (not just image) in the context of data transmission.

Out of all these algorithms only Yu [YS07], Wu [WZN09] and Smoaca [Smo12] seriously tested their algorithms with print and scan noise. Monga et al. [MM07] also tested it on one image. The most thorough study is that of Smoaca who used 30 images and copied them 31 times making 930 copies. Unfortunately, his algorithm is not versatile enough for our scenario. Yu's method is too sparse as will be shown in the next section and thus only Wu's method is really applicable to our scenario. For comparison purposes we will also use Venkatesan's method [VKJM00] whose statistical nature and popular tiling approach should yield interesting results.

At last we can note that no algorithm studies the issues brought by the discrete nature of images. For instance a round neighborhood cannot be round on a

pixel matrix. This is the aliasing phenomenon and it may have an impact on an algorithm's performance.

7.3 Study of key-point-based approaches

Sparse feature-based approaches are likely to lack precision, but their sparse nature may produce a digest whose size is small enough. Considering the stringent digest size requirement, this is an important advantage. All the recent sparse feature-based algorithms use key-point detectors, so this seems to be a reasonable solution. Furthermore Yu's algorithm [YS07] is based on key-points and is robust to print and scan noise which is another requirement that we have. Thus, in this section we study the informativeness of key-points and show that key-point based image representations are not precise enough.

Key-points use three algorithms: a detector that finds the locations of the key-points, a descriptor that describes the neighborhood of the key-points and a matching algorithm that pairs the key-points between two images. These approaches usually do not make use of color information and only use the image intensity. Thus our analysis will be focused on handwritten signatures as they do not use color information either. Since key-point based approaches do not work for them it is unlikely that they will work on more complex images and studying this more complex use case does not seem necessary.

In order to keep a minimal digest size and similarly to existing approaches we started our experiments with only the position information of the key-points and did not use a descriptor.

7.3.1 Feasibility of using only the key-points positions

The approaches that we surveyed do not make use of the latest key-point algorithms. We tried a set of eight classical detectors. Considering the negative conclusion of this section we simply review their main characteristics here:

- Harris corner detector [HS88] uses the eigenvalues of the local auto-correlation to detect edges and corners.
- GFTT [ST94] detects points based on the presence of texture, which is determined by the eigen values of a local window.
- SIFT [Low04] detects the maxima and minima of the difference of Gaussians of the Laplacian of the image.
- SURF [BTG06] detects the maxima of the determinant of the Hessian matrix (matrix of second order derivatives) computed on the box-filtered image.

- FAST [RD06] detects points that are darker or brighter than their circular neighborhood. This allows a very fast computation.
- CenSurE (also called STAR in OpenCV) [AKB08] uses center-surround filters similar to FAST with improved processing.
- ORB [RRKB11] improves FAST with a multi-scale approach and a focus on orientation detection.
- BRISK [LCS11] improve FAST with a multi-scale approach capable of dealing with continuous scales.

One important notion of key-point detectors that was popularized in SIFT and reused by nearly all other detectors is the difference of Gaussians. The idea is to iteratively divide the image resolution by 2 to produce octaves. Then several Gaussian filters of increasing width are applied on the image of each octave to make a scale-space representation of the image. Computing the difference between two consecutive images allows one to discover features that are present at a specific scale. Figure 7.3.1 shows this scale-space representation and how the difference of Gaussian is computed. However this approach is limited to only the computed scales which can be very constraining. This is why [LCS11] interpolate it to continuous scales.

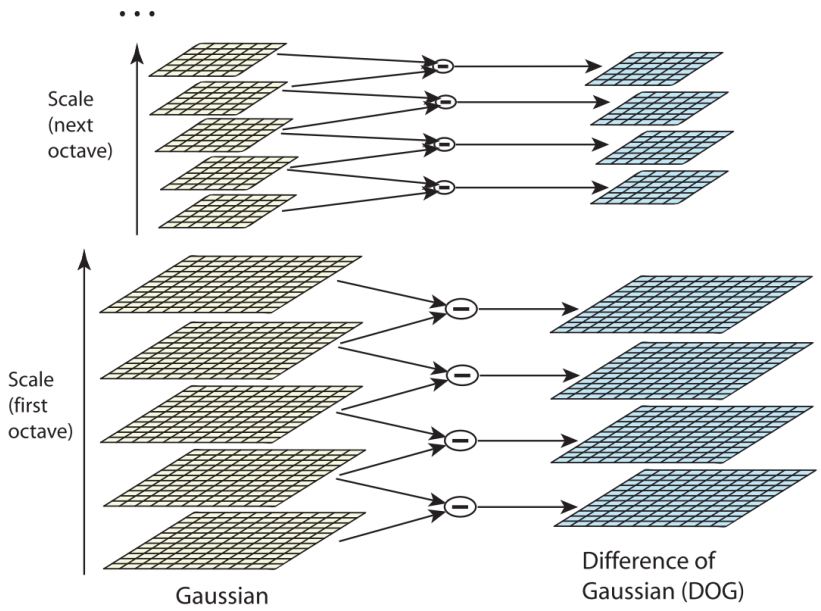


Figure 7.3.1: Computation of a space-scale image representation (left) and its difference of Gaussians (right). Image reproduced from [Low04].

(a) Harris (b) GFTT

(c) SIFT (d) SURF

(e) FAST (f) CenSurE

(g) ORB (h) BRISK

Figure 7.3.2: The points detected by the key-point detectors (best seen in color).

Figure 7.3.2 shows the results of these detectors on a handwritten signature (different from the ones of the dataset whose disclaimer forbids reproducing them) with their default parameters. Of course these results depend on the detector parameters but they give us a visual idea of the kind of results that are produced. Harris and GFTT, which both use eigenvalues, have similar distributions of points. SIFT produces less points that are more evenly distributed. It also produces points at the curvature centers (minima) between the loops. SURF produces a very noisy output. Many points do not lie on the signature. FAST covers the signature with points. CenSurE, ORB and BRISK definitely improve FAST and try to produce only relevant points without redundancy. CenSurE's output is a bit noisy too. Both CenSurE and ORB have difficulty finding points on the large curve of the P which may lead to a lack of precision. This may be related to a curvature radius that is too big, or it could be an issue with the scale detection.

These key-points detectors have several parameters related to the neighborhood size, the range of scales to process and other parameters for selecting the best key-points. The detected key-points between two images are matched with LLAH [NKI06]. LLAH uses the relative positions of key-points to compare images and, considering the number of key-points that are used (more or about 200), is suitable for the task.

Two experiments were made: one with two copies of two images of handwritten signatures making a total of 4 images. We cannot show the four images that were used because of their license. It served to use a brute force testing of all possible parameters of the key-points detectors. The best trade-off of parameters yielded an FNR of 50%, an FPR of 0%, an FOR of 33% and an FDR of 0% which is far from being sufficient. The results where nearly the same for all detectors.

The second experiment used a larger set of four copies of four different signatures making 16 images. It used the most promising sets of parameters from the first experiment. Considering that each position is made of two 8-bit integers (the positions can be quantized if necessary), 250 positions can fit in 500 bytes. Thus, for the second experiment, when the detector gave a point quality value, we took the best 250 points, otherwise we took the first 250 points returned by the detector. When the detectors produced less than 250 points, all the points were kept. This experiment was done to ensure that the poor performance of the first experiment was not related to the dataset. The results of this experiment are shown in Figure 7.3.3. We can see clearly that SURF and ORB are the best key-point detectors although their performance is still insufficient.

In order to have a thorough study of the possibilities of key-points based approaches we can use the descriptors associated to these key-points. This will not fit the digest size constraint, but maybe we can find a compression technique that will compensate this.

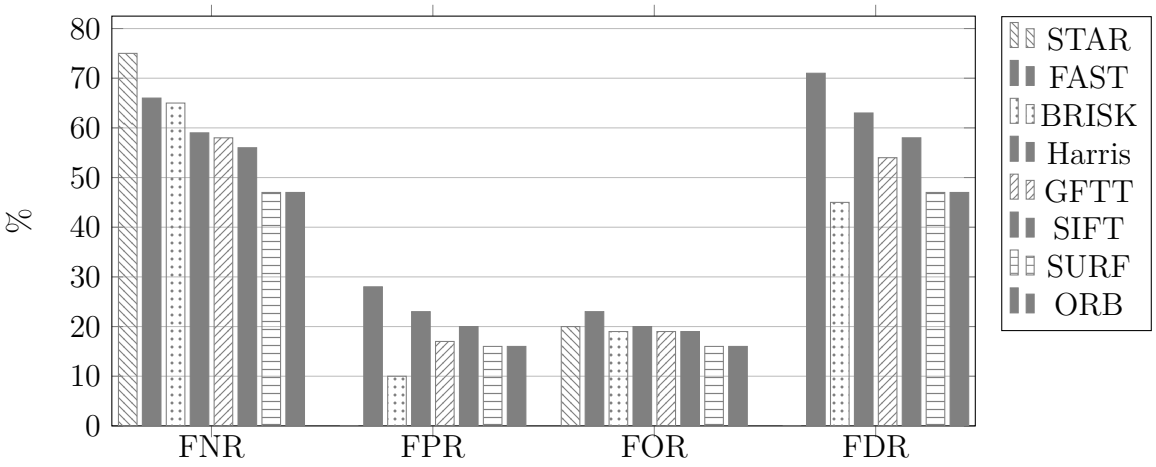


Figure 7.3.3: Performance of the key-point detectors matched with LLAH on 16 images of handwritten signatures. The values are in percentage.

7.3.2 Feasibility of using the key-points descriptors

This time we use the descriptors associated with the three best detectors: SIFT, SURF and ORB. Because BRISK is similar to ORB it is also included in the benchmark to see if its descriptor could improve its results. The dataset is the same as before with 16 handwritten signature images.

We can quickly review the descriptors:

- SIFT's descriptor uses the distribution of gradients around the key-points.
- SURF uses local wavelet responses.
- ORB's descriptor is based on another one called BRIEF [CLSF10]. Both use a set of local binary sets (pairs of pixels) and ORB improves on BRIEF by making it deterministic and more robust to rotations.
- BRISK's descriptor uses a fixed oriented pattern to compute local binary tests and computes them in a more efficient way than BRIEF.

Size-wise, SIFT, SURF, ORB and BRISK's descriptors require 128, 256, 32 and 64 bytes per key-point respectively. When compared to the two bytes required for the position one understands the size cost and compression requirement of using descriptors.

The descriptors are matched with FLANN [ML09] which is a state of the art nearest neighbor finding algorithm. Figure 7.3.4 shows the results of this experiment.

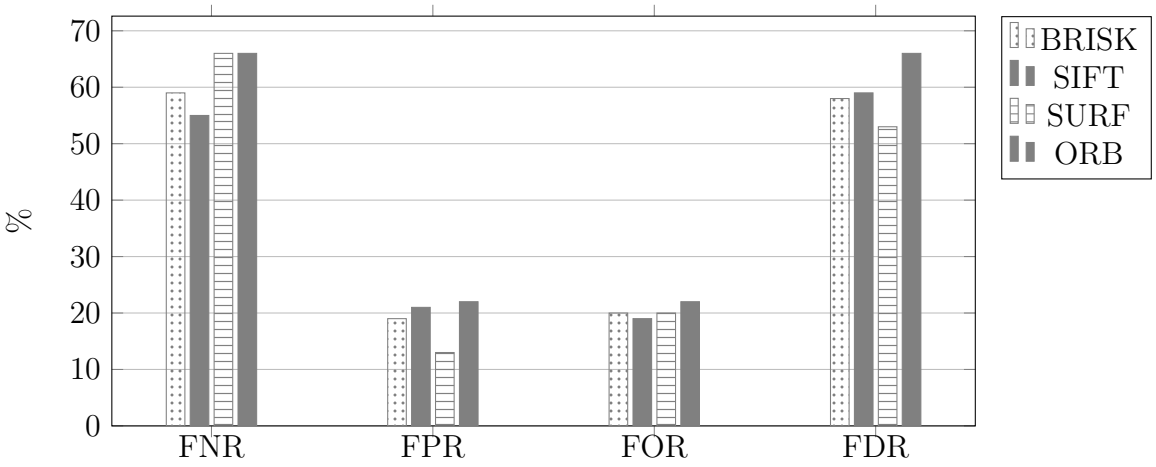


Figure 7.3.4: Performance of the key-point descriptors matched with FLANN. The values are in percentage.

It appears that ORB is the worst descriptor. SIFT and SURF get the minimum FNR and FDR respectively. When considering the maximum performance indicator value for each algorithm, BRISK and SIFT have the minimal one. SIFT has a significantly lower FNR than BRISK and thus may be the best descriptor (in this scenario).

We can also notice that, on our sample images, the performance is generally worse when using the descriptors than when using the key-point positions. This means that the position of the key-points is more representative of the image than the descriptors. This could be expected since the key-points are localized on edge corners and thus represent the geometry of the objects in the image. The descriptors are more likely to represent the local texture which is less discriminative. This result should be confirmed by more thorough experiments since our testing dataset is very small. yet, it probably explains why several sparse feature-based perceptual hashing algorithms use only the key-point positions.

This approach could be conducted further with a combination of position and descriptor information, but, considering the clearly insufficient performance reached so far, key points do not seem to be a promising approach in our case.

7.4 A Simple Yet Complex Hashing Algorithm (ASYCHA)

Similarly to most perceptual hashing approaches we plan to compute a compact description of the image and then an adequate matching algorithm. Our approach

is based on a coarse representation of the image.

There are two main challenges. The first one is to describe the image with enough precision in a size that is small enough. The second one is to be able to match two images in spite of the print and scan noise.

The general idea is that the digest should be an extremely lossy and high ratio compression of the image. Then we use image registration techniques to compare two digests. Similarly to the state of the art, none of these techniques are particularly complex or new, but combining them to achieve the required level of performance is quite a challenge. Hence the name of the algorithm. However, we differ from classical techniques by working in the spatial domain. This reduces the noise related to the discrete nature of the image space. This noise would be more significant with frequency based approaches especially if the frequency values have to be quantized to fit in one byte like an image pixel. Our registration technique is also more advanced than the one of [MVE05] and allows a better handling of geometric distortions.

This pair of algorithms (hashing and matching) is made to work on both logos and handwritten signatures. It is actually content agnostic apart from the fact that we suppose that the background is white. If not, a background segmentation algorithm such as the one presented in Section 5.7.3 can identify the background and replace it with white. As such, they should work on other more generic sets of images although we did not test this. The general process of image hashing and authentication is displayed on Figure 7.4.1

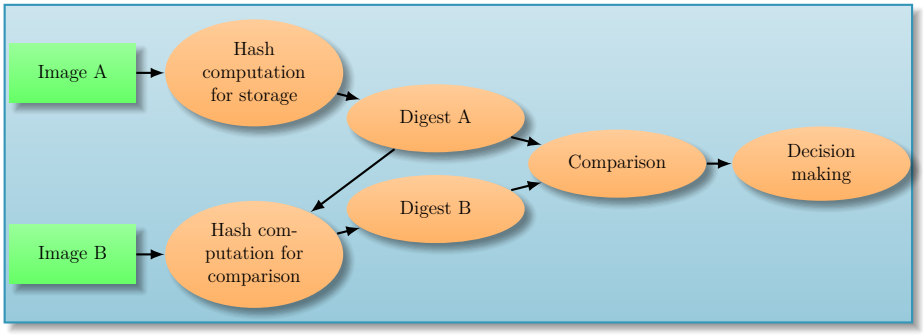


Figure 7.4.1: Overview of the image hashing and authentication process.

7.4.1 Hashing algorithm

The algorithm for the generation of the digest is illustrated in Figure 7.4.2 and the impact of the different steps on the image are shown in Figure 7.4.3. The input is the original color (RGB) image, I, accompanied by its resolution information ρ (dpi). The dpi is often stored in the image meta-data produced by the scanner.

Its output is a specific indexed image and the second order moments of the input image.

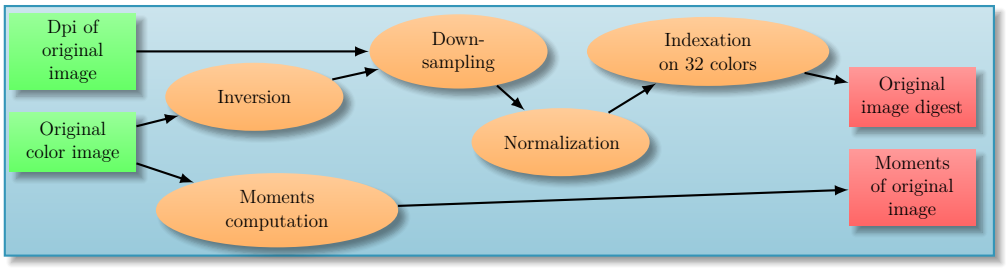
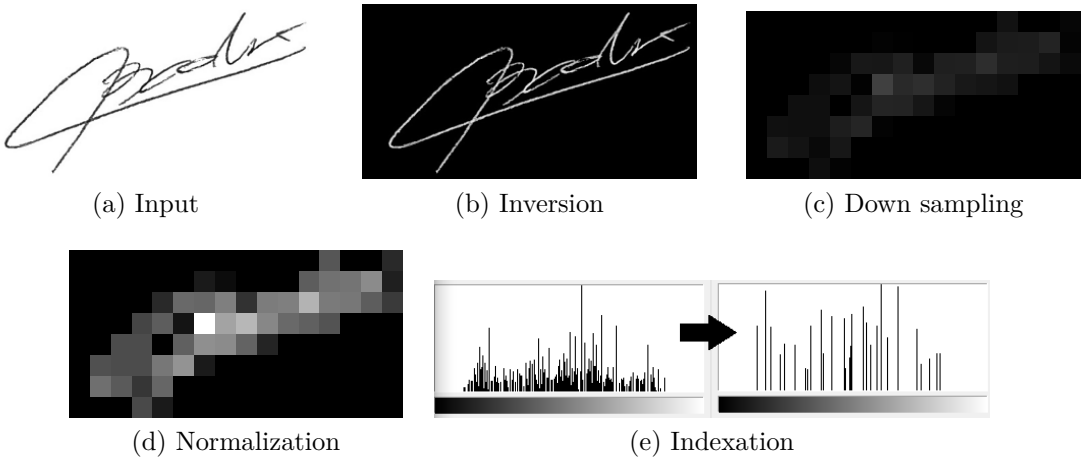


Figure 7.4.2: Algorithm for digest generation.



(a) Input (b) Inversion (c) Down sampling

(d) Normalization (e) Indexation

Figure 7.4.3: The resulting image after each step of image hashing.The indexation does not produce any visible change on the image hence we show its impact on the image histogram.[1]

Moments computation: The function A extracts, the central second order moments, $A(I) = \{\mu_{11}, \mu_{02}, \mu_{20}\}$, which are computed as:

$$\forall\{p,q\} \in [\![0;2]\!], \quad p+q=2, \quad \mu_{pq} = \sum_x \sum_y (x-\bar{x})^p (y-\bar{y})^q I(x,y) \tag{7.4.1}$$

It will be stored with the digest of the image and used in the matching for image registration. Since they are only used to compute the direction of the inertia axes of the image, we do not need the moments of higher order.

[1]Handwritten signature image courtesy of Boris Bodin

Inversion: This step relies on our assumption that the background is white. It is accomplished because we want to correlate the signatures and logos and not their background. The inversion intends to increase the brightness of the shape of the handwritten signature or the logo and to decrease the brightness of the background. To perform this inversion, the RGB image is converted into the YCbCr color space which separates the intensity information in the Y channel from the color information in the Cb and Cr channels. The YCbCr color space is chosen because it is the one of the JPEG compression algorithm. The inversion function B is defined as:

$$I_B = 255 - I_Y(x, y) \tag{7.4.2}$$

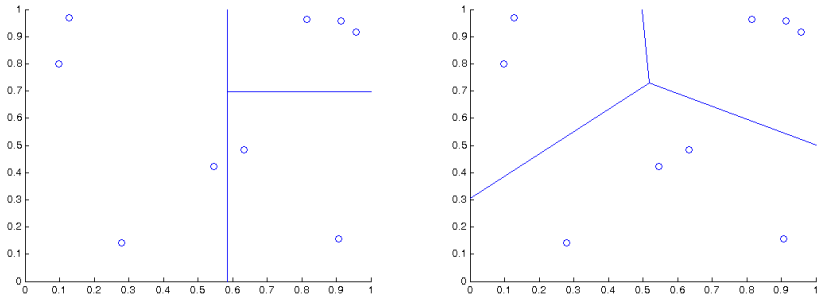
where I_Y is the Y channel of image I, and I_B is the inverted image. Then the image is converted back to the RGB color space. This step is not necessary if the image background is segmented and can be ignored during the image correlation.

Down-sampling: The down-sampling compresses the image and reduces salt and pepper noise as well as small color variations introduced by the print and scan process. All images are resized to a resolution of $\frac{600}{42}$ dpi(≈ 14.3 dpi) similar to that of the project SIGNED [Mal13].This means that we consider that modifications with a smaller size than a square of 1.8 mm are insignificant. The down-sampling uses a bilinear interpolation with a reduction factor of $\frac{600}{42\times\rho}$. We have tested a bicubic and nearest neighbor interpolation as well, but they create artifacts that reduce the performance of the overall system.

Normalization: The normalization compensates for brightness variations introduced by the print and scan process. The normalization is performed on the Y channel, and consists of stretching its histogram to use all the range of values. This normalization, will increase the space between the colors and allow a better selection of different colors in the quantization of the indexing step. This is not a histogram equalization which would smooth the histogram.

Indexing: The indexing intends to remove the colorimetric noise introduced by the print and scan while retaining the significant color information. This step also helps reducing drastically the size of the digest. Thus the image is indexed on a maximum of 32 colors i.e. on 5 bits. The final image is composed of an index matrix H and its color mapping table T.

The indexing is done by a k-means clustering of the image on the 3-dimensional RGB cube with the Euclidean distance. For a good initialization of the centroids of the k-means, we use a variance minimization quantization [Wu91] which makes the indexing deterministic. Having a deterministic color quantization algorithm is paramount as we need to have the highest stability and any randomness will reduce it. The variance minimization quantization is deterministic but not satisfactory as it is a hierarchical partitioning algorithm which may not produce balanced clusters as shown on Figure 7.4.4a. Unbalanced clusters mean that some of the content of



(a) Non optimal/unbalanced clustering result.

(b) Balanced results.

Figure 7.4.4: Balanced and unbalanced clustering results to produce three clusters.

the image which has a a specific color may be lost and we cannot afford this. This is why the variance minimization quantization is followed by a k-means clustering which produces a balanced result as shown on Figure 7.4.4b.

Finally, the digest of the image contains the moments $\{\mu_{11}, \mu_{02}, \mu_{20}\}$, the resolution information of the image ρ, the index matrix H and its color mapping table T.

7.4.2 Matching algorithm

The matching algorithm takes as input the digest of the original image (indexed image, color table, dpi and second order moments) and the color test image (denoted with a quote I'). The matching process has two steps: the generation of the digest of the test image and its comparison with the digest of the original image.

7.4.2.1 Generation of the test image digest

The generation of the test image digest depicted in Figure 7.4.5 is similar to the process of generating the digest of the original image (Section 7.4.1) except for the linear image registration. We add an image registration step in order to handle the rotation and scale variations between the two images which might be introduced by the print and scan process.

The linear image registration has two components: the linear matrix and the linear transformation based on this matrix. The base idea is to compute the equivalent ellipse orientations and sizes based on the second order moments of the images to be compared. After this the linear transformation rotates and resizes the test image so that its corresponding ellipse is the same as that of the original image. The linear transformation matrix T is computed with the second order moments

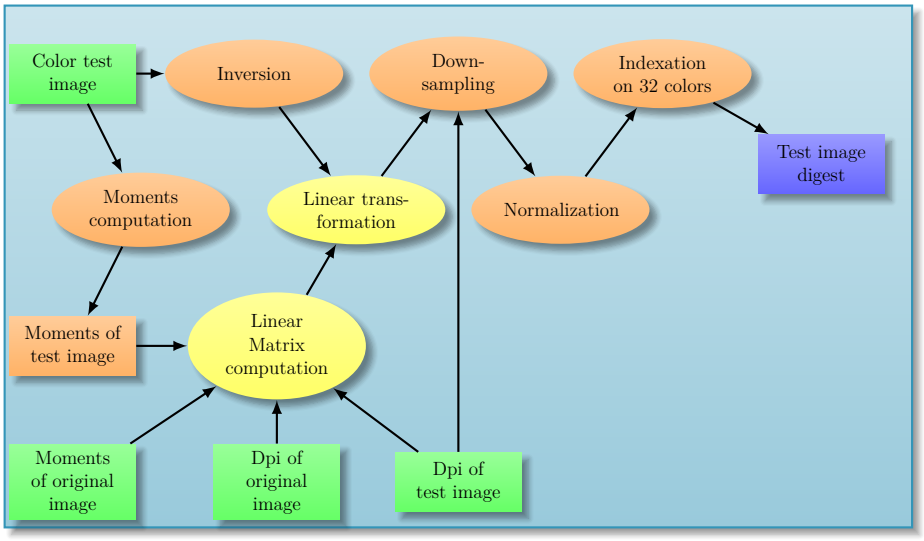


Figure 7.4.5: Generation of the test image digest

and the resolution information of both images. It is based on a scale factor and a rotation angle. The rotation angle called θ is determined by Equation (7.4.3), and the scale factor called Δ by Equation (7.4.4).The matrix, T, is computed as shown in Equation (7.4.5).

$$\theta = \arctan\left(\sqrt{\frac{\mu'_{20}}{\mu'_{02}}}\right) - \arctan\left(\sqrt{\frac{\mu_{20}}{\mu_{02}}}\right) \tag{7.4.3}$$

$$\delta' = \sqrt{\frac{(\mu'_{20}+\mu'_{02})+\sqrt{(4*(\mu'_{11})^2)+(\mu'_{20}-\mu'_{02})^2}}{2}} \qquad \Delta = \frac{\rho'}{\rho} \cdot \sqrt{\frac{\delta}{\delta'}} \tag{7.4.4}$$

$$\delta = \sqrt{\frac{(\mu_{20}+\mu_{02})+\sqrt{(4*(\mu_{11})^2)+(\mu_{20}-\mu_{02})^2}}{2}}$$

$$T = \begin{bmatrix} \Delta \cdot \cos(\theta) & \Delta \cdot \sin(\theta) \\ -\Delta \cdot \sin(\theta) & \Delta \cdot \cos(\theta) \end{bmatrix} \tag{7.4.5}$$

When applying the rotation with the linear transformation G, the unknown pixels are filled with black (background) color. The background is black because the image has been inversed. The resulting image is shown in Figure 7.4.6.

The rotation and scaling may introduce some significant artifacts, in particular for rotations of small angles. To deal with this issue is we simply test the image with and without the image registration as explained in Section 7.5.2.

Figure 7.4.6: Impact of the registration step. From left to right: original image, test image, test image after registration.

7.4.2.2 Digest comparison

The comparison algorithm (Figure 7.4.7) takes as input the digests of the original image and of the test image. We will now detail each step of this comparison.

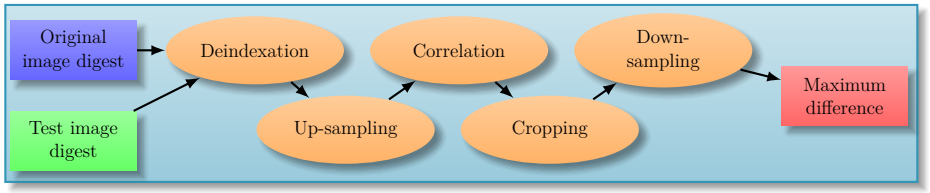


Figure 7.4.7: Algorithm for digest comparison

Deindexation: The deindexation will convert back the digests to RGB images for further processing.

Up-sampling: Because of the very small size of the digests, it can happen that the optimal correlation position occurs between two pixels. To account for this, each digest is resized by a factor 12. This factor allows an exact positioning on every half, third, quarter, sixth and twelfth of a pixel. We use a bicubic interpolation. This may create artifacts but because they are created on both images, we expect that if the images are similar, so will be the artifacts and they should not increase significantly the difference between the images.

Correlation: The correlation of the digests, allows us to get the best overlay between the two digests and to compensate for any translation. We perform one correlation per color channel as:

$$\forall i \in \{1,2,3\}, \quad R_i(x,y) = \frac{\sum_{x',y'}(J_i(x',y') \cdot J'_i(x+x',y+y'))}{\sqrt{\sum_{x',y'} J_i(x',y')^2 \cdot \sum_{x',y'} J'_i(x+x',y+y')^2}} \tag{7.4.6}$$

where i indicates the color channel, x, y are the image coordinates, and J and J' are the up-sampled original and test images respectively. Then, the resulting matrices R_i are summed to obtain the coordinates of their maximum, (x_m, y_m):

$$(x_m, y_m) = \underset{x,y}{\operatorname{argmax}}(\sum_i R_i(x, y)) \tag{7.4.7}$$

This maximum defines the translation which gives the optimal overlay of both digests. To perform the correlation for most image sizes, the test digest is padded on each side by half the size of the original digest.

Cropping: The cropping keeps only the overlapping area of the digests after the translation by the correlation. At the end of this step, the digests have the same size. Figure 7.4.8 shows the result of the correlation followed by the cropping.

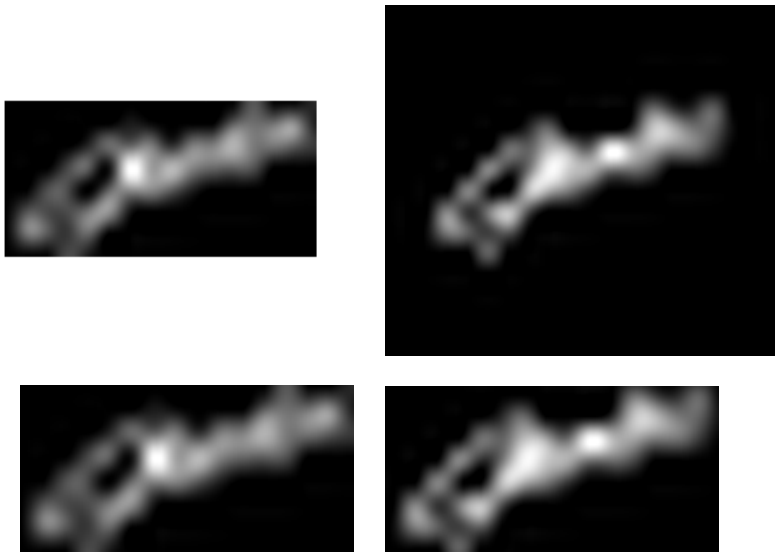


Figure 7.4.8: Impact of the correlation followed by the cropping. Top row: up-sampled images, bottom row: correlated and cropped images, left column: original images, right column: test images.

Down-sampling: The image down-sampling resizes the digests back to their original scale with a factor 1/12 and a bilinear interpolation.

Maximum difference: The maximum difference υ is the Hausdorff distance between the two digests, e.g. the maximum absolute value of the digest differences along each color channel.

$$\upsilon = \max_i(\text{abs}(K'_i - K_i)). \tag{7.4.8}$$

where i is the color channel and K and K' are the cropped original and test image respectively. In [MVE05] they mention that the Hausdorff distance is sensitive to outliers, but this is not the case here because they have been removed during the down-sampling steps.

The above steps form the matching process which provides the distance between both images: υ. It is between 0 and 255 for 8-bit integer images.

7.4.3 Decision making

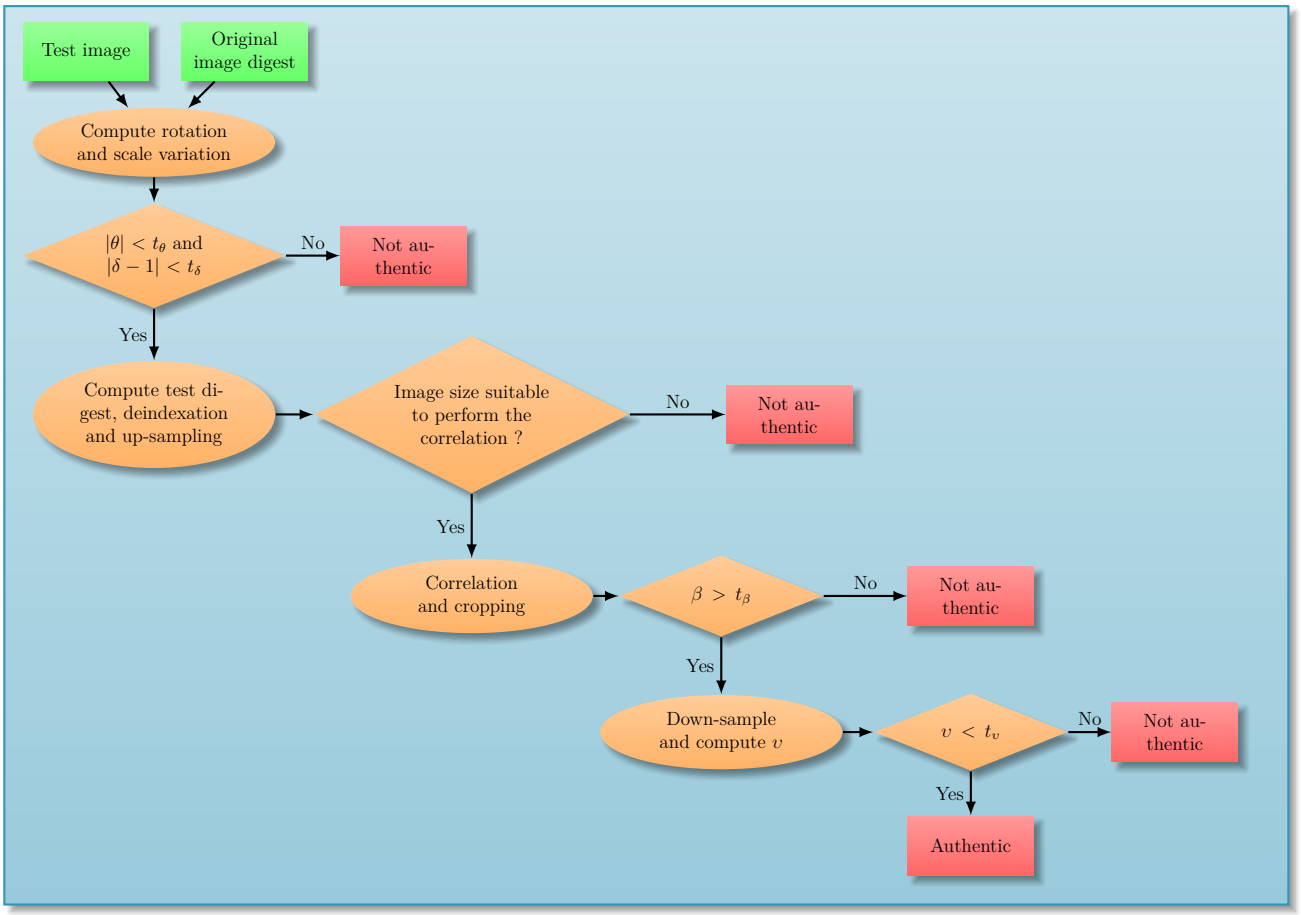


Figure 7.4.9: Digest comparison and decision tree.

The decision making is done through a decision tree depicted in Figure 7.4.9 along with its integration with the comparison process. The goal of this decision tree is not to make an elaborated decision scheme but rather to discard obvious cases as soon as possible in order to save computation power. The critical decision is really only made at the last stage.

If the rotation angle θ (7.4.3) is too large or if the scale factor Δ (7.4.4) is too far from the identity value 1, then the images are considered as different. Similarly, if their size does not allow to make the correlation, they are considered as different. From the cropping step we extract another feature, the coverage β, defined by:

$$\beta = \frac{\text{area}(K')}{\max(\text{area}(J'), \text{area}(J))} \tag{7.4.9}$$

If this coverage β falls below a threshold, the images are again considered as different. Finally, if the distance between the images υ is too large, the images are considered as different. This is this last decision criteria that significantly impacts the performance of the algorithm.

This successive case pruning allows the algorithm to eliminate obvious different images without having to perform too much computation. Furthermore, as we evaluate weak constraints first, we safely reduce the potential noise for the last constraint (the distance) which becomes a better classifier.

7.5 Evaluation of ASYCHA

In order to evaluate ASYCHA we created a dataset of photocopies of logos and handwritten signatures.

Considering our stability evaluation framework defined in Section 2.3 we need to define the similarity functions for the input and for the output spaces. The similarity function of the input space is the indicator of whether the two images being compared are photocopies of the same image. The similarity function of the output space is given by the matching algorithm and the decision process. We first present the testing dataset and then the evaluation results.

7.5.1 Testing datasets: L3iSignCopies and L3iLogoCopies

We have compiled two very challenging datasets to evaluate our algorithm. L3iSignCopies requires that the algorithm should be capable of detecting the differences between two signatures made by the same person while being able to identify photocopies of the same signature as identical.

L3iLogoCopies has similar requirements, but this time the differences to detect can involve color as well as localized modifications in contrast to signatures where slight differences occur on the whole signature.

Both datasets include JPEG compression with quality factors of 75, 82 and 94 as produced by the scanners. The printers have also produced different levels and kinds of noise.

Signature dataset (L3iSignCopies)

L3iSignCopies is a dataset of photocopied handwritten signatures. The original images are from the training dataset of SigComp2009 [BVFV09] containing 1898 images of handwritten signatures. They were printed by three printers: Sharp MX 904, Lexmark x543 PS and Konica Minolta Bizhub 223. Then, they were scanned by four scanners at two different resolutions making $3 \times 6 = 18$ copies of each image (see Table 7.1). This makes a total of $18 \times 1898 = 34164$ signatures.

	SignCopies		LogoCopies	
Resolution	300 dpi	600 dpi	300 dpi	600 dpi
Fujitsu fi 6800	1	0	0	0
Konica Minolta Bizhub 223	1	2	1	2
Konica Minolta Bizhub C364e	0	1	1	1
Lexmark x543 PS	1	0	1	0

Table 7.1: Number of copies for each scanner and each resolution.

It should be noted that this dataset contains several signatures made by the same person as well as forged signatures. In our case we will consider that only the photocopies of the same signature are identical. Thus several signatures from the same author are considered different as well as their forged versions. Our algorithm is not made for handwritten signature authentication. We cannot display any image from this dataset because of the SigComp2009 dataset disclaimer.

Logo dataset (L3iLogoCopies)

L3iLogoCopies is a dataset of photocopied logos. The original images taken from the site of the logo library[2] are composed of 200 logos of beer brands. They were scaled at three different sizes: 20 mm, 25 mm, 30 mm and they were printed by three printers: Sharp MX 36N, Lexmark x544 and Ricoh pro c7100x. Then they were scanned by three scanners at two different resolutions as shown in Table 7.1.

It should also be noted that some logos are from the same brand but at different times and thus have only small differences. These logos should be considered different unless the difference is smaller than the spatial resolution of the digest which is a square of 42 by 42 pixels. Figure 7.5.1 shows some of the logos.

This makes a total of $18 \times 3 \times 200 = 10800$ logos and a total dataset size (SignCopies and LogoCopies) of 44964 images.

[2] `www.lalogotheque.com`

Figure 7.5.1: An example of different images of logos of the L3iLogoCopies dataset.

7.5.2 Results

The results are organized as follows. After the results summary we will study the stability of the baseline algorithms and of ASYCHA. Then we will study the digest sizes and the time required to compute and match each digest.

Results summary

We compare our results with the method of Venkatesan et al. [VKJM00] and that of Wu et al. [WZN09]. We used 250 blocks instead of the original 150 blocks for the method of Venkatesan in order to have roughly the same spatial resolution than that of our method. Keeping a constant number of blocks, independent from the image size (in cm) would result in a significant loss of performance for large images. This conclusion was verified by our experiments and we only present the best results obtained with 250 blocks. Wu's method works best as described in the original paper so no adaptation was made.

The thresholds for Venkatesan's and Wu's methods are chosen to optimize both robustness and fragility on the whole dataset. Their values are shown in Table 7.2.

We study here three versions of our algorithm. The first named Algo1 does not use the linear registration to compute the test digest. The second Algo1R, uses it. When the rotation is very small, the linear transformation adds more noise than it removes, thus increasing the distance between the images to be compared. So, we made a third version of our algorithm, Algo2, that takes the minimal distance provided by Algo1 and Algo1R. The thresholds for the decision making are shown in Table 7.2. The optimal decision parameters are a maximum angle of rotation

Perf. Ind. (%)	Venkatesan ($t_{\text{Ven}} = 0.009$)	Wu ($t_{\text{Wu}} = 0.12$)	Algo1 ($t_v = 94$)	Algo1R ($t_v = 93$)	Algo2 ($t_v = 83$)
FNR	**0,3**	5.2	14.0	12.0	8.2
FPR	8.9	39.3	5.2×10^{-3}	5.0×10^{-3}	$\mathbf{3.2 \times 10^{-3}}$
FOR	2.7×10^{-2}	3.4×10^{-3}	5.7×10^{-3}	4.9×10^{-3}	$\mathbf{3.3 \times 10^{-3}}$
FDR	49.9	99.9	13.1	12.3	**8.0**
Digest size	500 bytes	**50 bits**	186 to 1174 bytes median 427 bytes		
Digest computation time (ms)	**53.5**	152.0	64.0		
Digest matching time (ms)	**0.1**	196	25.5	27.8	53.0

Table 7.2: Best results for the different methods : $t_{\text{Ven}}, t_{\text{Wu}}, t_v$ are the decision thresholds of the different methods. All the values should be as small as possible.

of $t_\theta = 2°$, a maximum scale difference of $t_\delta = 8\%$ and a minimum coverage of $t_\beta = 85\%$. They produce the best performance trade-off one the whole dataset.

We can see from Table 7.2, that Venkatesan's and Wu's methods are very robust (low FNR and FOR) but not very fragile (or precise) in our context (high FPR and FDR). Our method is far more fragile than the state of art (FPR, FDR) and thus more able to distinguish different and potentially fraudulent images. It also maintains a similar robustness (better FOR than Venkatesan's method, but not on FNR). Globally our method achieves a better trade-off than the state of the art.

Regarding the size and computation times of the digests, ASYCHA produces the biggest digest size but achieves a reasonable trade-off between the digest size and computation time. It also has a reasonable matching time.

Analysis of the stability of the algorithm of Venkatesan et al.

Figure 7.5.2 shows the performance variation with the distance threshold. This is the NPOD diagram already introduced in Section 2.3.2. The threshold can vary

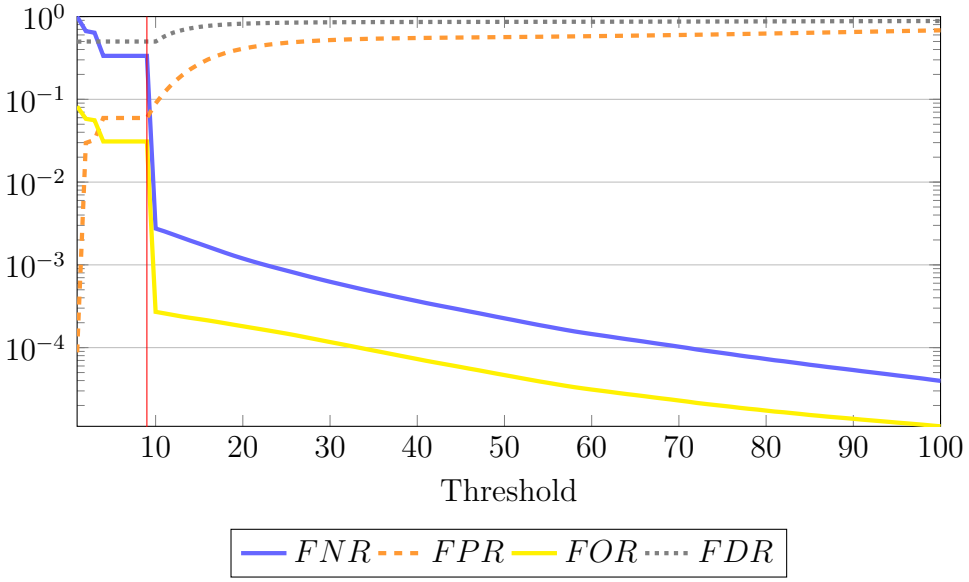


Figure 7.5.2: Performance of the algorithm of Venkatesan et al. The threshold values are multiplied by 1000. The red vertical bar shows the optimal threshold value.

between 0 and 1. The figure only presents the range between 0 and 0.1 as it contains the optimal values. There is no intersection between the FOR and the FDR for a threshold above 0.1. The performance is stable between 0.003 and 0.008. Then the FOR and FNR decrease sharply for the optimal threshold of 0.009.

Analysis of the stability of the algorithm of Wu et al.

Figure 7.5.3 shows the performance variation with the distance threshold. The threshold can vary between 0 and 1. The figure only presents the range between 0 and 0.55 as it contains the optimal values. This is an even more degenerate situation than the one of Venkatesan as there is no intersection. The FDR is near 100% all the time and the FPR also has a very high value. This highlights the lack of sensitivity of the algorithm. The performance does not vary much between 0 and 0.15. The optimal performance is achieved for a threshold of 0.12.

Analysis of the stability of ASYCHA

Figure 7.5.4 shows the performance variation with the distance threshold. The threshold can vary between 0 and 255. The performance is far better than that of the other algorithms. The four usual intersections are present and they occur at fairly low values. This illustrates clearly the difference between a stable algorithm

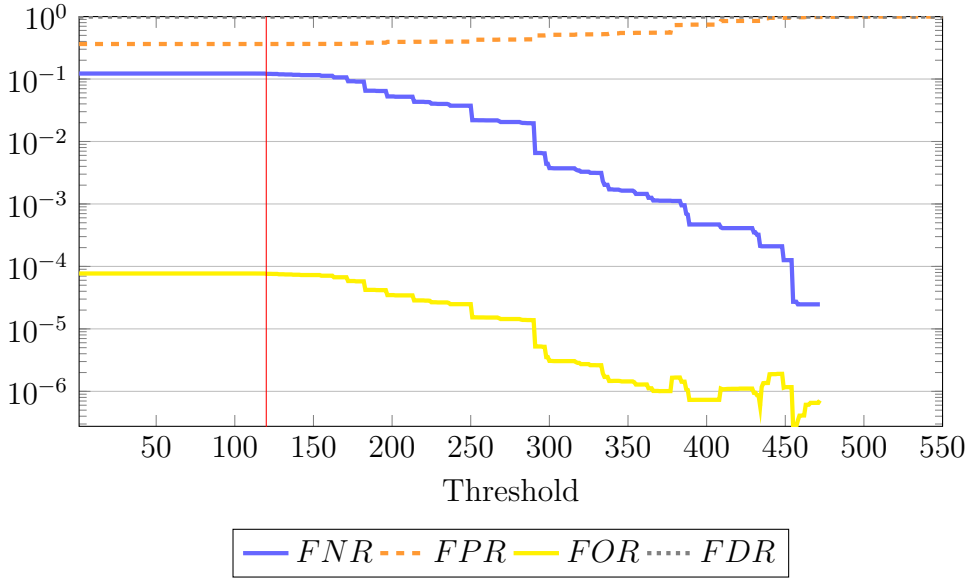


Figure 7.5.3: Performance of the algorithm of Wu et al. The threshold values are multiplied by 1000. The red vertical bar shows the optimal threshold value.

and the algorithms that are commonly produced. It also illustrates the importance of using the four proposed performance indicators in order to evaluate the performance of an algorithm.

Digest size

The digest size for Venkatesan is 500 bytes and that of Wu is 50 bits. ASYCHA does not produce a fixed size digest. This is done in order to maintain a fixed spatial precision for the image. The digest size s for an image of size (m, n) pixels is given by Equation (7.5.1) in bits for a color map of 32 colors. The first $2 \times 10 + 10 + 3 \times 16 = 78$ bits are used to store the image size, resolution and moments. Then $3 \times 8 \times 32 = 768$ bits are used for the color map and the rest is the color index (5 bits) for each pixel of the digest.

$$s = 78 + 768 + 5 \times (m \times n) \times \frac{600 \times 600}{(42 \times 42 \times \rho^2)} = 846 + (m \times n) \times \frac{50\ 000}{(49 \times \rho^2)} \quad (7.5.1)$$

For a dynamic color map of k colors, it becomes:

$$s = 78 + 24 \times k + (m \times n) \times \frac{ceil(log_2(k)) \times 10\ 000}{(49 \times \rho^2)} \quad (7.5.2)$$

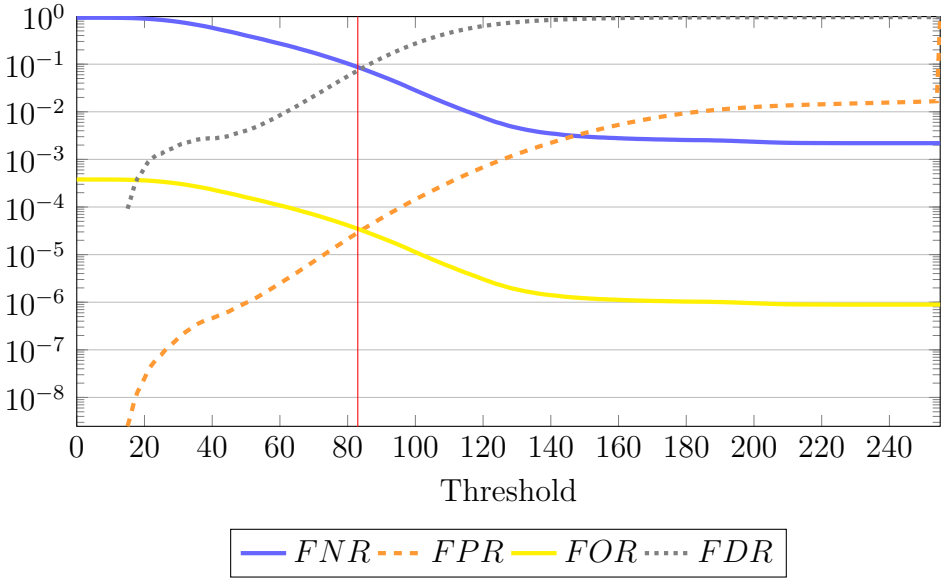


Figure 7.5.4: Performance of ASYCHA. The red vertical bar shows the optimal threshold value.

On the test dataset the minimal digest size is 186 bytes, the maximal size is 1174 bytes, the average is 443 bytes and the median value is 427 bytes. While this is more than expected for some images, the majority of the digest fits within the allotted space. Furthermore, only the digests of a handwritten signature are too large. This is in relation to the sizes (in cm) of the handwritten signatures that can also be quite big. Thus the combination of the digests of a logo and of a handwritten signature should generally fit within the allotted space.

Computation times

Venkatesan's algorithm is extremely fast in particular for the matching. On the opposite Wu's algorithm is very slow. ASYCHA has digest computation and matching times around 50 ms each which is quite sufficient for our needs.

It should be noted that with our dataset, there is more than two billion matches to compute for each algorithm. This requires more than three years of computational time on a single thread at 50 ms per match. We used a computing cluster to perform all the experiments.

7.6 Conclusion

In this chapter we have shown the limits of current approaches for perceptual image hashing. Key-point based approaches as well as two state of the art methods were evaluated and proven not to be precise. Robustness has been widely studied in the state of the art but not precision.

The proposed algorithm, ASYCHA, has been tested on an extensive dataset and it outperforms the state of the art in terms of raw performance in particular for the detection of different or fraudulent images. It achieves a reasonable digest size and computation times. It is able to detect any modification bigger than a square of 42 by 42 pixels at 600 dpi as specified by the industrial partners of the project SIGNED [MF06]. We have shown that using the raw image data and compressing it with down-sampling and color indexing is a good way to capture the stable information contained in an image while not requiring much computation.

The decision process to compare two images takes four thresholds to make a decision: angle, scale, coverage and intensity differences. They also have the advantage of relating to physical properties of the image which makes them easier to tune. The scale and coverage should not need to be changed unless for some very specific cases. The rotation is dependent on the print and scan process but is easy to estimate. The intensity difference has been evaluated on two billion matches so it should only be changed to have a different balance between false positives and false negatives. It is not influenced by the brightness of the images nor by their colorimetric noise as these have been taken into account in the digest computation and comparison.

We acknowledge that the basic blocks involved in this algorithm are very simple. However, we would like to point out that it is the careful choice of these algorithms and their precise combination that allows us to deal with all the sources of noise, artifacts and instability and to achieve our level of performance. Considering the performance improvement that is almost as desired, this work is a significant step in the right direction which could be improved further. In particular, we plan to alleviate the algorithm drawbacks with the corresponding measures:

- Better handling of scale variation with other moment formulas and allowing different horizontal and vertical scale variations.
- Better handling of the colorimetric noise with a relative normalization between the two images being compared.
- Better handling of the background by not including it during the registration process.
- Faster registration with a pyramidal correlation.

- Improved security with pixel order randomization of the digest with a private key.
- Improved security and compactness with the pseudo-cryptographic hashing scheme described in [ASU10] which would only require four bits per pixel instead of five for 32 colors and 128 or 256 bits for the color map.

The last two improvements aim at making the hash more secure with the use of cryptographic techniques and can be taken off-the-shelf. The current hashing algorithm does not completely hide the information being secured since a degraded version of it is still available.

Thanks to keeping the color map it is also very easy to compare the intensity of an image with the one contained in the digest of another image. This allows image authentication in the classical case, where an image has been photocopied in gray levels. If this is the case, the image can still be authenticated with a warning that it has lost its color information.

Conclusion and perspectives

This thesis has tackled a wide range of algorithms under the angle of stability. These algorithms should help producing a hybrid hashing algorithm e.g. an algorithm capable of authenticating documents even if they are printed, photocopied, scanned, faxed, etc.

We will recall here the objectives of this thesis, what contributions were made to reach them and finally we will present the new perspectives opened by this work.

Objectives of the thesis

Our main objective was to create stable document image analysis (DIA) algorithms that could be used in a document authentication framework. This required to achieve good robustness as well as a good precision. The algorithms also had to be fast and to produce a compact description of the document.

The use of cryptographic techniques was interesting to protect the security and confidentiality of the content being secured but this was dependent on the stability of the algorithms involved.

Since the issue of the stability of DIA algorithms has barely been studied before, another objective was to evaluate the current state of the art in this field with respect to this criterion.

The algorithms that were inside the scope of this thesis perform the following tasks: layout description, document image segmentation, superpixel/connected color component (CCC) segmentation, optical character recognition (OCR) and perceptual image hashing.

Review of the progress done on the proposed semantic hashing framework

In Section 1.3.4 we proposed a new semantic hashing framework to solve the issues of current hybrid security technologies. We recall the general process of this framework in Figure 7.0.1.

Looking back at the work done, we have found that the task of producing a stable document image segmentation is extremely difficult and current algorithms are not stable (Chapter 4). In order to help make progress on this topic we have proposed

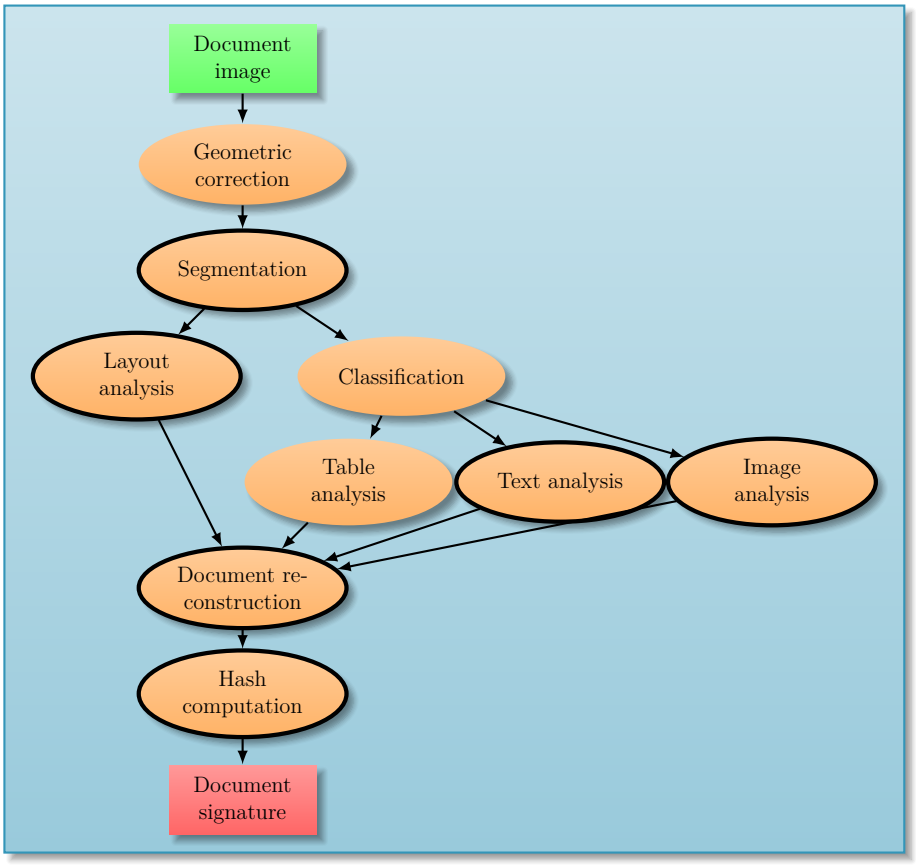


Figure 7.0.1: Algorithm for semantic hash generation

a way to extend the definition of connected components to color and gray level images with an outstanding stability (Chapter 5). This extension may also serve to detect edges and scales as well as to provide a notion of inclusion levels in an image. All these could come in support of creating a stable segmentation algorithm. We managed to solve the question of describing the layout of the document in a very stable and compact manner (Chapter 3). The text analysis is the second most difficult challenge after the segmentation. We managed to improve drastically the stability of existing algorithms but we have not yet reached a sufficient level of performance (Chapter 6). Finally we have produced an image analysis/hashing algorithm that achieves reasonably good results although it should be improved further (Chapter 7).

As mentioned in Section 1.3.4, the document reconstruction is a mere concatenation of the output of the other algorithms. The hashing is not yet finalized. Cryptographic hashing can be applied on the layout description and we have pro-

posed a hashing solution for the image description. It is possible that the hashing algorithm allows us to handle the instability of the other algorithms. It may also allow us to build a digest that offers a better privacy protection. Both topics remain to be studied.

From a more generic perspective, Chapters 1 and 2 have laid out some foundations for the other partners of the project SHADES. They now have a good overview of the issues that they have to deal with as well as a good definition of stability and means to study it.

Detailed summary of the contributions

Because of the groundbreaking nature of this thesis, it includes many contributions.

As we mentioned in the introduction, the main contribution is to bring into light the critical issue of the stability of document image analysis algorithms. In particular we formalized the definition of a stable algorithm in Section 2.2 and we provided a generic framework to evaluate it in Section 2.3.

On the topic of authenticating hybrid documents, we proposed a new hybrid document image hashing framework in Section 1.3.4. This framework should enable us to make a hybrid hashing algorithm once all its components are done.

Both the study of stability and the hybrid hashing algorithm rely on two elements: a proper document typology which was presented in Section 1.1 and an analysis of print and scan noise which was done in Section 1.2.

The chapters of this thesis grouped the contributions by topic so we will now group them by kind of contribution. In order to solve the problems raised in this thesis, we proposed several stable algorithms:

- The Delaunay Layout Descriptor (DLD, Section 3.3)
- Two CCC segmentation algorithms (Watercolor and Smooth Watercolor) and a post-processing algorithm for them (Section 5.5)
- The alphabet reduction, a disambiguation algorithm for OCR output (Section 6.3)
- ASYCHA, a perceptual image hashing algorithm (Section 7.4)

These algorithms are all parameter free although the DLD and ASYCHA require a matching algorithm that has some parameters. Those allow the user to choose the performance trade-off that best suits his needs. The last three algorithms are user centric rather than content centric. This helps having better posed problems and more stable and versatile algorithms.

The Delaunay Layout Descriptor leverages the stability of the Delaunay triangulation algorithm and combines it with a specific ordering algorithm to make a

description of a set of points that is stable enough to be hashed with a cryptographic hashing algorithm. This results in a very compact description. Two such digests can be matched with an algorithm that analyzes the potential sources of instabilities and test them. The user can choose the maximum number of simultaneous instabilities and the maximum rotation angle that are allowed. Setting low values for these makes the algorithm extremely fast while being more tolerant allows one to reach a higher degree of stability. In any case, the DLD performs better than state of the art and, depending on the trade-off, far better in terms of stability.

The CCC segmentation algorithms extend the notion of connected components to color and gray level images. This is a significant advance as it will allow the use of algorithms that use connected components with gray level and color images. So far, they were limited to binary images. They are based on a detailed model of human vision that takes into account its spatial resolution, its colorimetric sensitivity and its gradient sensitivity. Once this model is applied on the image, a color distance map is computed and processed by a watershed algorithm. Two versions of the color distance map have been proposed. Watercolor uses one that makes precise contours and Smooth Watercolor uses one that is better at dealing with gradients but produces less precise contours. Both algorithms produce many superfluous regions mostly because of the fact that the size of color regions influences the perception of their color differences. Hence we also proposed a post-processing algorithm that merges the regions based on a spatio-colorimetric distance. All these algorithms perform similarly to the corresponding state of the art superpixel algorithms in terms of segmentation quality, but they outperform it by a vast margin in terms of stability.

The alphabet reduction simply aims at removing unreasonable ambiguities from the expected results of an OCR algorithm. The idea is to replace visually similar looking characters such as a capital “i” and a lower case “L”. by a single character. This leads to a problem that is better posed and drastically improves the stability of OCR algorithms. The alphabet reduction could been seen as a loss of precision but actually, it does not change the content of the document for a human reader. Hence there is no loss of information. From a computer point of view, the loss of information only occurs for the collisions which are extremely rare. Once OCR algorithms will have reached a sufficient stability it will be possible to apply cryptographic hashing on them to produce extremely compact descriptions of the text contained in a document. In the meantime, fuzzy hashing will allow one to have a compact description and detect the mistakes of the OCR algorithm and the modifications of the document.

Finally, ASYCHA intelligently combines several simple image processing algorithms to produce a compact and yet precise description of any color image.

Roughly the image is resized to a much smaller size and its colors are quantized. The matching algorithm registers a test image with the digest of an original image. Then it produces a digest for the test image and compares both digests with a Hausdorff distance. While the algorithms used are simple, their precise combination allows us to handle all kinds of print and scan noise properly and to produce stable results. This is notably the case with a new color quantization scheme which combines two algorithms to obtain a deterministic and balanced clustering of the image colors. Among the four thresholds used in the matching algorithm, only the angle variation may need to be adapted to the printing and scanning process. The final decision threshold (intensity difference) serves to change the balance between false negatives and false positives. Once again ASYCHA outperforms the state of the art in term of stability and precision while keeping a similar performance on other criteria. The current version of ASYCHA does not make use of cryptographic techniques but we have identified off-the-shelf technologies to improve the security of the content being hashed.

The above algorithms were evaluated in several benchmarks. Since the stability of DIA algorithms had not properly been studied before we created stability benchmarks for the following types of algorithms:

- Layout descriptors (Section 3.4)
- Document image segmentation algorithms (Section 4.4)
- Superpixel segmentation algorithms (Section 5.6.2)
- OCR algorithms (Section 6.4)
- Perceptual image hashing algorithms (Section 7.5)

These benchmarks were accompanied by the corresponding datasets:

- L3iLayoutCopies: photocopies of perfectly stable segmentation layouts (960 images of 15 layouts, 64 copies per layout, Section 3.4.1)
- L3iDocCopies: photocopies of documents from the PRImA dataset (990 images of 55 documents, 18 copies per document, Section 4.4.2)
- L3iTextCopies: photocopies of text only documents with 216 typographical variations per text (42 768 images, 22 texts, 9 copies per variation, Section 6.4.1)
- L3iLogoCopies: photocopies of logos printed in different sizes and with small variations (10 800 images of 200 logos, 18 copies per logo of each size, Section 7.5.1)

- L3iSignCopies: photocopies of handwritten signatures from SigComp2009 including several trials by the same author and forgeries (34 164 images of 1898 signatures, 18 copies per signature, Section 7.5.1)

Apart from these main contributions we also obtained a few significant corollary results. We made a quick formal analysis of the problem of superpixel segmentation and highlighted its internal contradictions. This led use to define a proper corresponding problem which is the one of CCC segmentation (Sections 5.2 and 5.3). Having a well posed problem was one of the keys to making a stable algorithm. The other key was to focus on the observer and not on the observed data. This led to two contributions. Section 5.4.5 describes a new spatio-colorimetric distance capable of taking into account the size of color regions to compute their color difference. This was included in a detailed model of human vision which we believe contains more features than other existing models. It is summarized in Section 5.4.5. Among the new features included in this model, we looked at modeling the contrast sensitivity of the human eye. For this we benchmarked several edge-preserving filters in Section 5.4.4 and concluded that the domain transform filter [GO11] is the most suitable.

Because our CCC segmentation algorithms provide meaningful CCCs they also provide parameter free edge and scale detection algorithms. Another application is the separation of the image into layers from the outermost regions to the innermost regions. This allows a parameter free binarization of color images based on these layers. It is also very easy to select some parts of an image based on the layer number.

During our study of OCR algorithms in Section 6.4.3, we showed that, when all other variations are taken into account (image resolution, font, font emphasis), the font size does not influence the stability of an OCR algorithm. This may help reducing the size of the datasets when evaluating such algorithms.

In Section 7.3 we have shown that key points' relative locations are more discriminative than their descriptors to identify images of handwritten signatures.

Finally, we performed several reviews of the state of the art with varying degrees of thoroughness on the following topics:

- Print and scan noise/models (Section 1.2)
- Digital and hybrid security algorithms (Sections 1.3.1 to 1.3.3)
- Layout descriptors (Section 3.2)
- Document image segmentation algorithms (Section 4.2)
- Superpixel segmentation algorithms (Section 5.3)

- Edge preserving filters (Section 5.4.4)
- OCR algorithms (Section 6.2)
- Perceptual image hashing algorithms (Section 7.2)

It may be of interest to study how the work presented in this thesis could be applied to other types of documents or noise. Generally speaking because we mostly took an observer based approach, the algorithms that we designed should be fairly versatile and work on a wide range of documents and noise as long as their underlying hypotheses are maintained. Let us review this for each algorithm that we propose. The Delaunay Layout Descriptor describes the relative positions of a given set of points. As such it is independent of the document and of the noise it contains. As we said its limitations are related to the angle error and the number of simultaneous instabilities. The matching algorithm currently limits its practical use to less than 100 regions/points. The CCC segmentation algorithms are all based on the observer and their versatility has been demonstrated as long as the resolution of the input document is known. The results they produce are not fully satisfying and increasing the noise level may produce worse results. This may be compensated with an improved post-processing. The alphabet reduction is another content agnostic algorithm and hence can be applied on any document. The improvement it brings on the character error rate is independent of the noise (roughly one point) but its impact on the stability of the OCR output may be masked at the page level if there are too many OCR errors. Finally ASYCHA is also content agnostic but tailored for a certain level of print and scan noise. Increasing this noise is likely to degrade the algorithm performance.

Future works and perspectives

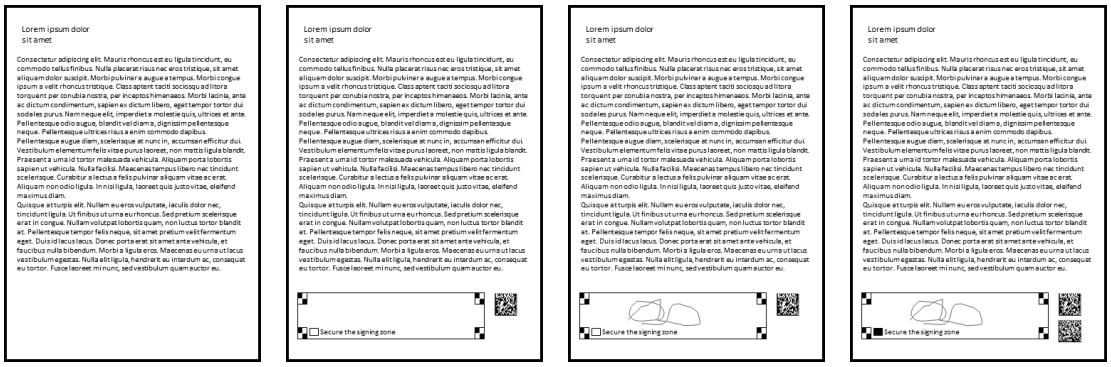
This thesis opened a whole new field of research. We hope that the document analysis community will continue pushing further the boundaries of stable algorithms and to study the stability of a larger variety of algorithms. This has proven to be an efficient way to make significant progress. The rationale behind this is that making a stable algorithm requires to extend its performance to other similar documents. This increases its reliability but also its raw performance since it works better on more documents.

The main missing part from this thesis is a complete document image segmentation algorithm. We expect that our CCC segmentation algorithms combined with a segmentation free OCR should allow one to make a very stable segmentation algorithm. The CCC segmentation algorithms that we proposed rely significantly on a model of human vision. Considering the results we obtained, our model can

probably be improved further. The spatio-colorimetric distance that we proposed also needs to be validated with more specialized hardware and experiments.

Overall, there is still a lot of work to do to complete the hybrid hashing algorithm that we propose. Classification algorithms still need to be studied, OCR algorithms need to be improved and some improvements remain to be implemented for ASYCHA.

Some may point out the fact that many documents are not static. They undergo a document life cycle during which they may be modified, annotated and/or signed/stamped. We have not provided for such kind of modification in this thesis. However, it seems possible to allow for a specific zone of the document not be secured until specified by the user. This would allow for a document to be authenticated before and after being signed. A fairly simple way to do so, would be to add some markers in the corners of the dedicated zone and call the signing process after the document has been signed. Figure 7.0.2 shows such a process.



(a) Unsigned original document.

(b) Secured unsigned original document.

(c) Secured original document with a non secured signature.

(d) Secured signed original document.

Figure 7.0.2: A solution to allow for a document to be modified while maintaining the security of its content.

Stability has also led us to focus on many interesting questions that, so far, have been neglected by the community. We solved two of them: producing a parameter free edge detection algorithm and another one for scales. We have not been able to solve several others and we hope that some people will do so in the future. We describe these issues in the following.

The k-means clustering [Llo82] is a very good clustering algorithm that produces a fixed number of clusters. However it is non deterministic and, since this problem is NP hard [ADHP09], it produces an approximate solution of the ideal clustering. Its main flaw is that we do not produce any internal non supervised estimate of

the quality of the results it produces e.g. how close they are to the ideal clustering. To our knowledge none of the studies so far have tackled this issue or even tried to study a relation between running several times a k-means and getting close to the optimal result. It is also very frequent to see studies that focus only on an ideal number of clusters which is not a real case scenario. Among the few very interesting works on the topic we would like to refer [PLL99] and [BMvL12]. [PLL99] does a thorough empirical study of the initialization of the algorithm and [BMvL12] does a theoretical one and proposes a new initialization scheme. Generally, it seems that the k-means is taken as an ideal algorithm (which it is not) and this has led to it being strongly understudied. This makes it difficult to use it if one wants to make a stable algorithm. Hopefully it can gain some new interest.

We have insisted on producing threshold and parameter free algorithms whenever possible [EGKO15a]. Considering the significant use of clustering algorithms we would like to point out a few that are parameter free [KLR04, CMO07, Git72, CLQL11, IPM09, BGO$^+$10, MNM13]. There is also the very classic watershed transform [BL79] and its hierarchical improvement [Beu91] which seems to have been forgotten. When using those algorithms, one should keep in mind that the algorithm performance is dependent on the paradigm underlying the algorithm itself. For instance, APSCAN [CLQL11] uses a density criterion, thus it is not capable of clustering correctly data that would not obey this criterion.

During our study of print and scan noise, of the human eye and of CCC segmentation we have been faced with aliasing. It is quite significant at 200 dpi and should be alleviated at 300 dpi. However the underlying issue is that our eye does not perceive the world with squares. Hence it may be of interest to find a more isotropic image sensing and representation system. This will also make it easier to apply mathematical formulas to the world of discrete images. The errors introduced by this continuous to discrete conversion did not have much impact so far and thus they have been completely neglected. However, they have a significant impact on the stability of algorithms and they cannot be neglected anymore.

Finally, we have pointed out the fact that the human eye spectral response is not the same as the one of a CCD sensor. This explains why it is currently impossible to take a digital picture that actually has the same colors as the scene that was photographed. A solution to this would be to increase the spectral accuracy of sensors and displays together by adding more color channels. Some studies have already been done in Japan on this topic [MIO$^+$04, HIH16, YTO$^+$02] and some vendors such as LG and BenQ have already started producing displays with four and six color channels. We hope that this kind of technology will soon become more popular in research and in the industry. Having higher quality images will inevitably lead to a better performance for image analysis algorithms.

Annexes

Appendix A

Discussion on segmentation algorithms

In this section we discuss the applicability scope of the surveyed algorithms, the evaluation practices of the community, its general trends. Finally we quickly review the surveyed algorithms from a user's point of view.

A.1 Evaluation of segmentation algorithms

Evaluating and comparing the performance of an algorithm is a difficult task. We summarize here the current state of the art for the evaluation of document segmentation algorithms. A first remark that can be made in the light of all the surveyed papers is that many authors tend to compare their algorithms with other algorithms based on the same technique. This prevents any cross-technique comparison to estimate which techniques are the most promising. Competitions and contest try to remedy this but it would be good if the community could base its future evaluation on the comparison with the best algorithms having the same functionality instead of the same technique.

A.1.1 Existing benchmarks

There has been three independent industry lead benchmarks: the MADCAT program[1] by DARPA, RIMES[2] by A2iA which is a document analysis company and MAURDOR[3] led by the French equivalent of DARPA, the DGA. MADCAT aims at creating a system to automatically categorize and translate any document into

[1] http://opencatalog.darpa.mil/MADCAT.html
[2] http://www.a2ialab.com/doku.php?id=rimes_database:start
[3] http://www.maurdor-campaign.org/

English. RIMES is a dataset of handwritten letters that has been used for several competitions [GA09, GEA11]. The most comprehensive one is clearly the MAURDOR campaign and it is publicly available[4]. It contains more than 8000 color documents with a complete ground truth (regions, region type, contained text, text type, text language and other meta data) as well as ready to use evaluation tools. It contains all types of document except for comics and in both French and Arabic typewritten and handwritten text.

ICDAR also organizes recurrent document segmentation competitions [APBP09, ACPP11, LLS11, ACPP13b, ACPP13a, ACPP15, MRHR15]. Unfortunately they frequently use a dataset that is too small (below 100 documents) and the evaluation tool is closed source, does not allow processing documents in batch mode and contains numerous parameters which may make the evaluation less objective [APBP09, ACPP11, ACPP13b, ACPP13a, ACPP15]. The competition by Murdock et al. [MRHR15] is a very good addition as it comes from the industry and thus aims at addressing real case scenario issues. The competition that was organized by Lamiroy et al. [LLS11] adopted an interesting point of view in that it divided an end to end system in several steps for which submissions could be made. Then the contribution of each algorithm was evaluated with the improvement it brought to the overall system.

A.1.2 Datasets

The evaluations in the different surveyed papers raise the question of the exhaustivity of the datasets. Clearly, the state of the art is the MAURDOR dataset. Yet it does not contain Asian or Cyrillic scripts. Historical documents are also missing. For these, the St gall, Parzival and Washington triptych [BLI13, WBSI13, CWL+14, CSL+15][5] is recommended as it covers a wide time span. Adding copies of the same documents could allow the evaluation of the stability of segmentation algorithms as done in [EGKO16]. A last addition could be documents similar to comics such as those in [RTBO13, WZT15]

In section 1.1 we highlighted the difficulty of defining what is a historical document. A 16$^{\text{th}}$ century manuscript is not the same as a decree from the 19$^{\text{th}}$ century, yet both are historical documents. The exhaustivity in that regard should be considered with the sampling of documents in time not just sampling two classes of documents.

The testing datasets do not all have the same exhaustivity. Clustering algorithms are evaluated on datasets that are, on average, six times bigger than those used for classification algorithms. This is explained by the fact that classification

[4] `http://catalog.elra.info/product_info.php?products_id=1242`

[5] available at `http://www.fki.inf.unibe.ch/databases/iam-historical-document-database`

algorithms require large training datasets in particular because of the curse of dimensionality. Thus for the same total dataset size (training + testing) a clustering algorithm will usually be tested on a larger dataset. However, this does not change the fact that, because of the testing dataset size, the results obtained for clustering algorithms are more reliable than those for classification algorithms.

Moreover, Baird and Casey [BC06] advocate for versatile algorithms. These are algorithms capable of dealing with a wide range of documents. Thus, they will have to handle documents that they have not encountered before. Hence the training set should definitely not be bigger than the testing set and the classical k-fold evaluation and datasets that specify training and testing sets need to be updated accordingly.

A.1.3 Performance indicators

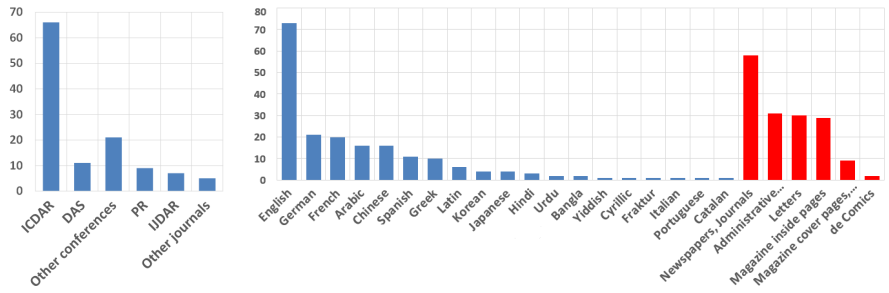
Most evaluations are based on the same principles of counting: false alarms (adding a region), misses (removing a region), merges (two or more regions in one), splits (one region in two or more) and matches (properly segmented region). Algorithms that tend to merge (respectively split) regions are said to under-segment (respectively over-segment) the documents. This is the most reasonable way to evaluate the performance of an algorithm on a single document. However this kind of performance indicator does not evaluate the repeatability and performance stability of the algorithm over a range of documents. Baird and Casey [BC06] denote this as evaluation based on "confidence before accuracy".

We actually analyzed the repeatability of four state of the art segmentation algorithms [EGKO16] and found that they all have a very poor stability. This is to be expected since they where never evaluated with this criteria but we hope that it will be used in future evaluations.

A.2 Trends and statistics

This survey gave us a very good insight on the trends, strengths and weaknesses of current algorithms. Figure A.2.1a shows the main publishing venues.ICDAR stands as the flagship conference of the community. DAS is the second biggest venue of the community and its main journals are PR and IJDAR.

Figure A.2.1b shows the detailed number of algorithms that studied each language and document type. The algorithms have been applied to 19 languages and scripts. This supports the worldwide applicability of the findings of the document analysis community. The top 5 most studied languages are in decreasing order: English, German, French, Arabic and Chinese. Regarding the document types,



(a) Number of publication per venue

(b) Number of algorithms that studied each language and document type.

Figure A.2.1: Venue, language and document type publication trends.

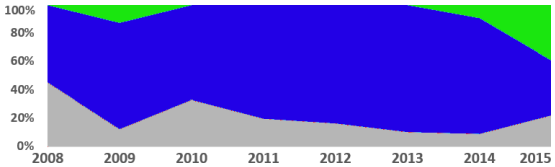
newspapers and journals have been studied twice as much as any other type. The least studied types are magazine cover pages and comics.

Figure A.2.2a summarizes the number of publications based on each technique group. The techniques in groups 1 and 2 are disappearing to the profit of the third group. The main techniques used for document segmentation are bottom up techniques, in particular geometric clustering and feature classification. They cover 24% and 19% of all surveyed algorithms respectively. All geometric clustering algorithms work at the connected component level.

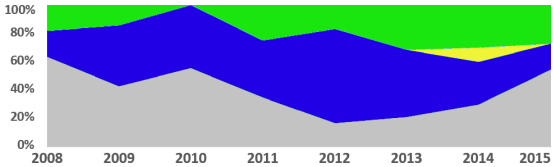
For the first two groups, we can analyze the shares of top-down and bottom-up algorithms and among the bottom-up algorithms the processing scale. Figure A.2.2b shows that there is majority of bottom-up algorithms but top-down algorithms are gaining a new interest. Overall, most of the bottom-up algorithms work at the connected component level but pixel level analysis is becoming more popular.

Figures A.2.2c and A.2.2d show the proportion of algorithms that use a specific color depth and the proportion of algorithms that are evaluated on historical, modern or both types of documents. More and more algorithms make use of color information and are evaluated on historical documents. Currently there is a tie between testing on modern or historical documents. Testing on both types of documents (historical and modern) has a significant share although it seems to become less active.

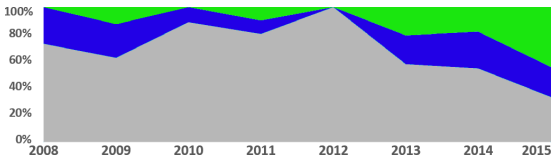
Figure A.2.3 summarizes the number of publications that have been tested on a given number of different document types (according to the typology of section 1.1) and of different languages for a given dataset size. We can see with the big blue and red bubbles that most algorithms that are tested on a dataset below 1000 images are only tested on one language and one document type. The algorithms tested on more than 1000 images are mostly tested on one type of document and



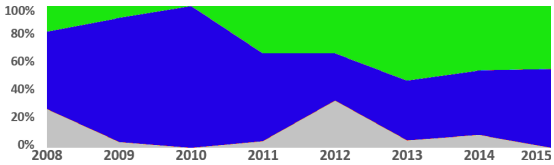
(a) Shares of the algorithms groups. From top to bottom: group three, 2 and 1.



(b) Shares of processing level for the algorithms in the first two groups. From top to bottom: pixel, patch and connected component level for the bottom-up algorithms. In gray at the bottom, top-down algorithms.



(c) Shares of the color depths that are used by the surveyed algorithms. From top to bottom: color, gray level, black and white.



(d) Shares of the algorithm that are evaluated on historical documents. From top to bottom: historical, modern, both.

Figure A.2.2: Trends of algorithm techniques and evaluations between 2008 and 2015

two languages. The most extensive testings are the blue bubble on the left (2 document types, 10 languages, dataset between 100 and 1000 images) and the green bubble on the right (6 document types, 3 languages, dataset bigger than 1000 images).

Time wise, Figure A.2.4 shows the tendency for the extensive nature of the evaluation of segmentation algorithms. The datasets on which they are tested tend to include more document types, more languages and tend to be bigger. Finally, nearly all papers are now compared to the state of the art.

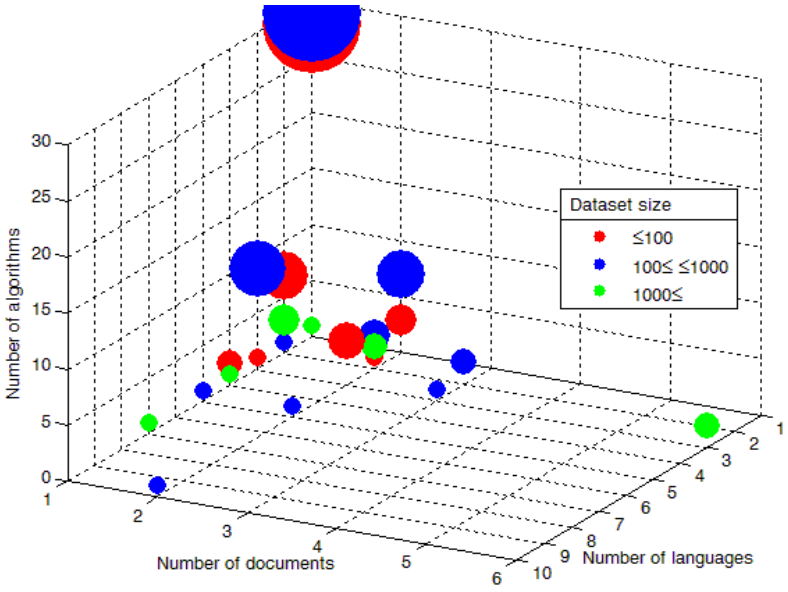


Figure A.2.3: Breakdown of the number of algorithms based on the number of languages and document types in the evaluation corpus. The radius of each bubble is proportional to the number of algorithms which is also the vertical coordinate. The color of the bubbles relates to the size of the corpus.

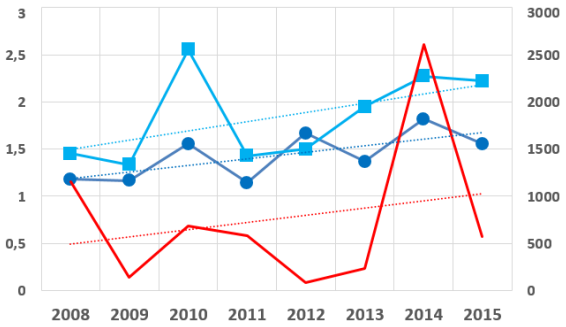


Figure A.2.4: From top to bottom: evolution of the average number of different languages, the average number of different documents and the average dataset size (scale on the right). The dotted lines show the linear tendency of these values.

Appendix B

Conversion from sRGB to Lab color space

The conversion of image sRGB color values to Lab values requires the use of an intermediate color space: the XYZ color space.

It was in 1931 by the CIE based on experiments on the human eye. It only depends on the illuminant. Y represents the sensitivity to light intensity and X and Z relate to the color. After correcting the RGB values with their ICC profile, it is easy to compute their XYZ counterparts with a simple linear transformation as shown in Equation B.0.1. All RGB values are normalized between 0 and 1. The conversion matrix is dependent on the RGB color space here is the one for the sRGB color space. Figure B.0.1 show the geometry of this color space.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.4124564 & 0.3575761 & 0.1804375 \\ 0.2126729 & 0.7151522 & 0.0721750 \\ 0.0193339 & 0.1191920 & 0.9503041 \end{bmatrix} \times \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{B.0.1}$$

The human perception is not linear thus the XYZ color space is imperfect and an improved, a non linear color space was created in 1976: the Lab color space. It is currently defined in the ISO/CIE norm 11664-4:2008 (CIE S 014-4/E:2007). Equations B.0.2 and B.0.3 detail the conversion from XYZ color space to Lab color space.

$$f : \begin{cases} [0,1] & \longmapsto & [0,1] \\ t & \longmapsto & f(t) = \begin{cases} t^{1/3} & \text{if } t > \left(\frac{6}{29}\right)^2 \\ \frac{1}{3}\left(\frac{29}{6}\right)^2 t + \frac{4}{29} & \text{otherwise} \end{cases} \end{cases} \tag{B.0.2}$$

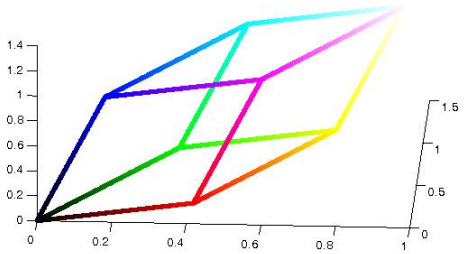


Figure B.0.1: Geometry of the XYZ color space.

$$\begin{aligned} L &= 116 \times f(Y/Y_n) - 16 \\ a &= 500[f(X/X_n) - f(Y/Y_n)] \\ b &= 200[f(Y/Y_n) - f(Z/Z_n)] \end{aligned} \tag{B.0.3}$$

X_n, Y_n and Z_n are related to the illuminant. For D65, we have $X_n = 95.047$, $Y_n = 100.00$ and $Z_n = 108.883$. This can be found in the norm ISO 11664-2:2007 (CIE S 014-2/E:2006).

Appendix C

Lab values used in the spatio-colorimetric experiment

Color variation	Color 1 (Lab)	Color 2 (Lab)	Color 3 (Lab)	Color 4 (Lab)
DL=10	40,20,0	50,20,0	60,20,0	70,20,0
Da=10	50,60,0	50,50,0	50,40,0	50,30,0
Db=10	70,0,10	70,0,20	70,0,30	70,0,40
DL=10	40,0,0	50,0,0	60,0,0	70,0,0
DL=15	20,0,0	35,0,0	50,0,0	65,0,0
Da=15	50,30,0	50,45,0	50,60,0	50,75,0
Db=15	70,0,10	70,0,25	70,0,40	70,0,55

Appendix D

List of publications

D.1 Publications based on the work presented in this thesis

Journal papers

Eskenazi, S., Gomez-Krämer, P., and Ogier, J.-M. (2016). An comprehensive survey of mostly textual document segmentation algorithms since 2008. *Pattern Recognition* (accepted for publication)

Conference papers

Eskenazi, S., Gomez-Krämer, P., and Ogier, J.-M. (2015). Let 's be done with thresholds ! In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 851–855. IEEE.

Eskenazi, S., Gomez-Krämer, P., and Ogier, J.-M. (2015). The Delaunay document layout descriptor. In *Symposium on Document Engineering (DocEng)*, pages 167–175. ACM

Other contributions

Eskenazi, S., Gomez-Krämer, P., and Ogier, J.-M. (2016). Evaluation of the stability of four document segmentation algorithms. In *Document Analysis Systems (DAS)*, pages 1–6. IEEE.

Eskenazi, S., Gomez-Krämer, P., and Ogier, J.-M. (2015). When document security brings new challenges to document analysis. In *International Workshop on Computational Forensics (IWCF)*, pages 104–116. SPIE.

Eskenazi, S., Gomez-Krämer, P., and Ogier, J.-M. (2014). Document semantic hashing for hybrid security. In *International Document Image Processing Summer*

School (IDIPS), Best poster award

D.2 Other publications by the same author

Conference papers

Dao N.B., Eskenazi S., Bertet K., and Revel A. (2015). A fuzzy precedence graph definition for algebra based dimension reduction. In *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE.

Burie J.C., Chazalon J., Coustaty M., Eskenazi S., et al. (2015). ICDAR2015 Competition on smartphone document capture and OCR (SmartDoc). In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 1161-1165. IEEE.

Nayef N., Luqman M.M., Prum S., Eskenazi S., et al. (2015). SmartDoc-QA: A dataset for quality assessment of smartphone captured document images - single and multiple distortions, In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 1231-1235. IEEE.

Eskenazi S. (2014). Quels pôles de compétitivité pour l'avenir ? In *Rencontres économiques d'Aix en Provence*, page 71. Le Cercle des Economistes.

Annexe E

Résumé étendu

Avec la numérisation en masse des documents, il convient de trouver des moyens de garantir l'authenticité de documents qui sont capables de changer de support (papier, numérique). Actuellement, les documents papier sont sécurisés avec des filigranes et les documents numériques avec des systèmes de cryptographie basés sur ce qu'on appelle des algorithmes de hachage. Malheureusement, les filigranes ne résistent pas toujours à la numérisation et ceux qui y résistent ne sont pas en mesure de sécuriser tout le contenu du document. Les systèmes cryptographiques quant à eux, ne sont pas capable de gérer le bruit introduit par le processus d'impression et de numérisation. Il convient donc de trouver un système de sécurité capable de fonctionner avec des documents hybrides. C'est à dire des documents qui existent à la fois au format numérique et papier. Nous proposons de créer un algorithme de hachage sémantique.

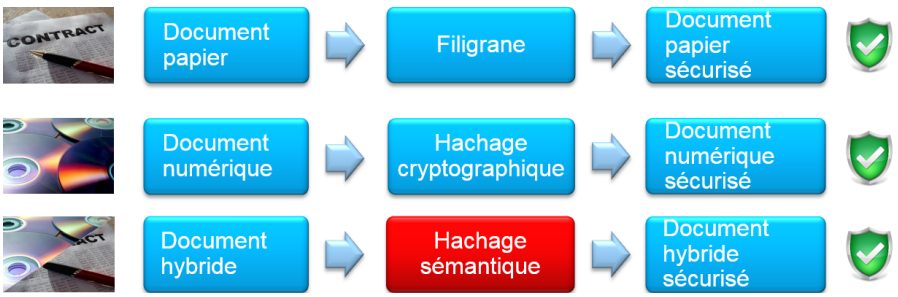


FIGURE E.0.1 : Technologies de sécurisation de documents papiers, numériques et hybrides.

Cet algorithme va reposer sur des algorithmes d'analyse d'image de document qui vont extraire le contenu du document pour qu'il puisse être sécurisé. Il y a deux principaux critères de performance pour évaluer la qualité de ces algorithmes :

- Ils doivent détecter toutes les modifications.

- Ils ne doivent pas détecter des modifications à tort.

Comme on le voit ces deux objectifs sont contradictoires et il y a un compromis à faire entre les deux. Les algorithmes stables ont un avantage dans cette situation car s'ils ont une bonne performance sur certains documents, ils auront la même performance sur toutes les autres copies de ces documents et probablement aussi sur les documents similaires. En outre, s'ils font une erreur sur un type de document, résoudre cette erreur améliorera les résultats de l'algorithme sur tous les documents similaires. L'effort nécessaire pour améliorer un algorithme stable est donc beaucoup plus rentable que pour un algorithme non stable.

Les contributions de cette thèse portent sur :

- Une formalisation de la définition et de l'étude de la stabilité d'un algorithme,
- Un nouvel algorithme sans paramètre qui segmente une image en composantes connexes couleurs (CCC). Il étend la définition des composantes connexes aux images en niveau de gris et en couleur,
- Un nouveau descripteur de mise en page sans paramètre et qui est particulièrement stable,
- Un nouvel algorithme de hachage d'image perceptuel qui est à la fois précis et stable,
- Un nouvel algorithme de post-traitement de reconnaissance optique de caractères (OCR) qui améliore drastiquement la stabilité de l'état de l'art.

En ce qui concerne les algorithmes sans paramètres, la description qu'ils produisent ne nécessite pas de paramètre mais certains d'entre eux utilisent un algorithme de comparaison qui a des paramètres afin de permettre à l'utilisateur de choisir le compromis de performance qui correspond à ses besoins.

Afin de parvenir à ces résultat nous avons obtenu plusieurs autres contributions :

- Une typologie générique pour les images de document,
- Un nouveau modèle de la vision humaine, y compris une nouvelle distance spatio-colorimétrique,
- Une revue approfondie des problèmes liés au processus d'impression et de numérisation,
- Plusieurs états de l'art des algorithmes utilisés dans cette thèse,
- Une évaluation de la stabilité de plusieurs algorithmes d'analyse de documents qui établit une référence pour l'évaluation d'autres algorithmes similaires,

- Plusieurs jeux de données contenant du bruit d'impression et de numérisation pour un total de 89 682 images.

Nous allons maintenant nous intéresser au contexte de notre étude.

E.1 Contexte de l'étude

La Figure E.1.1 montre la typologie des documents que nous proposons. Nous allons nous intéresser aux documents principalement textuels y compris les couvertures de magazines.

FIGURE E.1.1 : Typologie des documents triés par quantité de texte décroissante de gauche à droite.

L'objectif de cette thèse est d'étudier la stabilité des algorithmes par rapport à l'impression et à la numérisation. Il peut donc être intéressant d'identifier les sources et types de bruits introduits par ce processus. Le Tableau E.1 fournit une liste des différentes sources de bruit et des modèles existants pour les représenter. La fonction de flou ponctuel (point spread function, PSF) est une fonction qui permet de représenter l'étalement et le flou introduit par l'impression et la numérisation.

Un certain nombre de systèmes de sécurité existent pour les documents soumis à ce type de bruit. Ils sont tous basés sur un code visuel de type datamatrix. Ce code peut être fait de manière à ne pas être reproductible comme dans le projet Estampille [BC12, BC13] ou alors il peut contenir de l'information relative au contenu du document comme dans le projet SIGNED [Mal13]. Ce dernier projet est actuellement celui qui a obtenu les meilleurs résultats en terme de sécurité du contenu. Cependant, il a un défaut important : la taille de la signature qu'il

Sources de bruit	Modélisée	Références
Rognage	Oui	[LC99]
Changement d'échelle anisotropic	Non	
Conversion RGB vers CMYK	Non	
Conversion en demi-ton	Oui	[LC99, BMI07, PN95, Smo12]
Pièces électromécaniques de l'imprimante	dans la PSF	[BMI07, LC99, MF06, Smo12, YNS05, PN95]
Encre	dans la PSF	[BMI07, LC99, MF06, Smo12, YNS05, PN95]
Revêtement du papier	dans un modèle générique	[MF06]
Texture du papier	Non	
Épaisseur du papier	Non	
Dégradations liées au cycle de vie du document	dans un modèle générique	[LCOB15, SCE+15]
Bruit d'impulsion du scanner	Oui	[Smo12]
Système optique du scanner	dans la PSF	[BMI07, LC99, MF06, Smo12, YNS05, PN95]
Résolution de numérisation	Oui	[LC99, BMI07, PN95]
Sensitivité spectrale du scanner	Non	
Source lumineuse du scanner	Non	
Pièces électromécaniques du scanner	Oui	[BMI07, MF06, YNS05]
Manutention du document lors de la numérisation	Non	
Taille du document et de la vitre de numérisation	Non	
Poussière dans l'imprimante et/ou le scanner	Non	
Usure de l'imprimante et/ou du scanner	Non	
Compression JPEG	Oui	Algorithme JPEG [ITU92]

TABLE E.1 : Modèles des sources de bruit existant dans un processus d'impression et de numérisation. PSF veut dire point spread function.

génère est très grosse et nécessite plusieurs datamatrix pour être imprimée sur le document. Cela pose des problèmes pour trouver suffisamment de place et des problèmes esthétiques. La raison de cette grosse signature vient du fait que SIGNED utilise une approche de type analyse du signal de l'image. Dès lors qu'il y a beaucoup de pixels, la signature devient grande.

Une autre technologie qui émerge en France est le 2D-Doc. C'est un code barre 2D qui est un standard de l'agence nationale des titres sécurisés (ANTS). Il contient

un certain nombre de champs prédéfinis relatifs à l'information clé du document que l'ont veut sécuriser. Ces champs sont ensuite sécurisés avec des techniques cryptographiques similaires à celle pour les documents purement numériques. Cette technologie a plusieurs défauts. Le premier est que les champs doivent être renseignés à la main ou extraits d'un système d'information. La vérification de leur cohérence avec le contenu du document est manuelle. Cela empêche toute vérification automatique du contenu réel du document. Tout comme un filigrane, il est possible d'appliquer le 2D-Doc d'un contrat sur une recette de cuisine. La recette de cuisine sera alors sécurisée avec l'information du contrat. L'autre inconvénient du 2D-Doc c'est que puisqu'il est basé sur des champs prédéfinis, toute fraude en dehors de ces champs passera inaperçue. Le 2D-Doc ne sécurise pas tout le contenu du document. Enfin, les champs sont prédéfinis pour certains types de documents. Le 2D-doc ne peut donc pas sécuriser tout type de document administratif.

C'est pourquoi nous proposons d'utiliser des algorithmes d'analyse d'image de documents pour sécuriser tout le contenu d'un large panel de documents et pour produire une signature compacte puisqu'elle sera basée uniquement sur le contenu extrait du document. La Figure E.1.2 montre le processus général de notre algorithme de hachage. Les étapes avec une bordure épaisse sont celles qui seront présentées dans cette thèse.

La correction géométrique sert à corriger les éventuelles distorsions géométriques introduites lors de la numérisation. La segmentation sépare les régions du document et la classification identifie la nature de leur contenu. Les algorithmes d'analyse visent à extraire l'information correspondante. Cette information est ensuite regroupée dans un document virtuel lors de la reconstruction du document. Enfin, le hachage final est calculé avec ce document virtuel.

Ce processus nous permettra de sécuriser la majorité des éléments d'un document. Par rapport aux élément graphiques (analyse d'image), nous nous focaliserons sur les logos et les images de signatures manuscrites.

Comme on le voit, les algorithmes sont dépendants les uns des autres. Il est donc important que chaque étape soit aussi stable que possible pour toujours fournir le même résultat sur toutes les copies d'un même document. Il est temps de définir la notion de stabilité d'un algorithme et comment l'évaluer.

E.2 Définition et analyse de la notion de stabilité

Conceptuellement, un algorithme est stable s'il produit des sorties similaires quand ses entrées sont aussi similaires et inversement quand elles ne sont pas similaires. Il faut donc définir la notion de similarité.

Definition E.2.1 : **Fonction de similarité.** *Une fonction de similarité sur un*

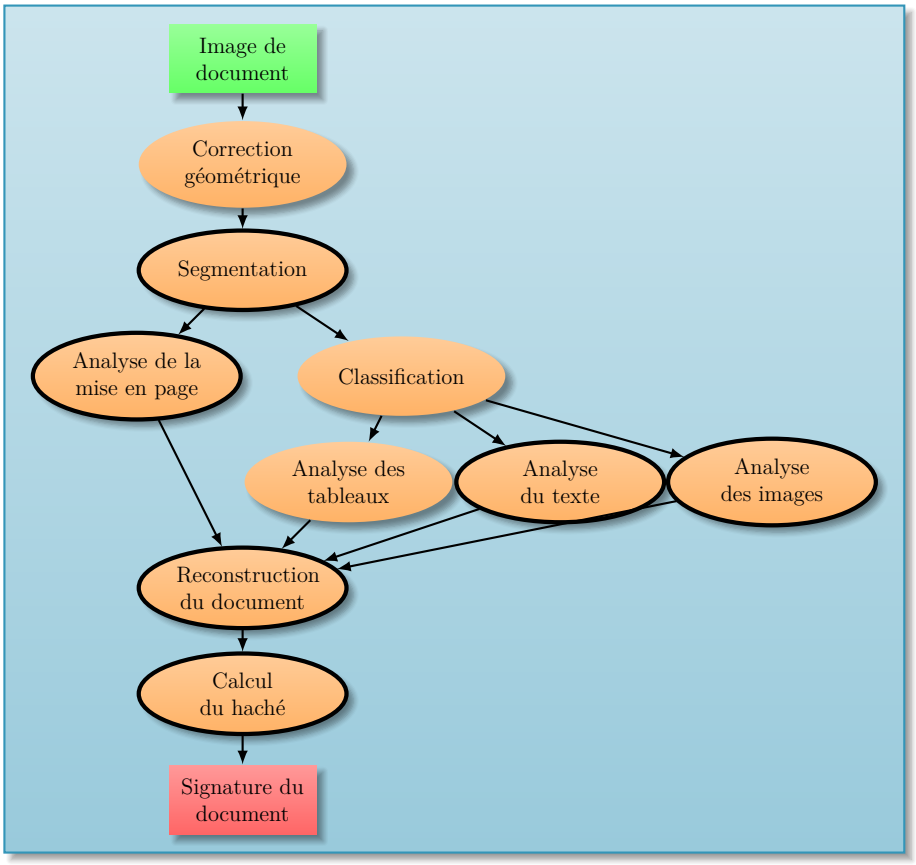


FIGURE E.1.2 : Algorithme de hachage sémantique de document

espace A est une fonction de deux variables, binaire et symétrique :

$$c : \begin{cases} A \times A \longmapsto \{0,1\} \\ (x,y) \longmapsto c(x,y) = \begin{cases} 1 \text{ } si \text{ } x \text{ } et \text{ } y \text{ } sont \text{ } similaires \\ 0 \text{ } sinon \end{cases} \end{cases} \tag{E.2.1}$$

Nous pouvons ensuite définir ce qu'est une fonction ou un algorithme stable :

Definition E.2.2 : **Fonction stable.** *Soit :*

- *Une fonction f (l'algorithme) : $f : I \longmapsto O$.*
- *Une fonction de similarité s_1 pour le domaine de définition I une fonction de similarité s_2 pour son ensemble image O.*

f est stable pour les fonctions s_1 et s_2 si et seulement si

$$\forall\{a,b\} \in I^2, s_2(f(a), f(b)) = s_1(a,b) \tag{E.2.2}$$

Afin d'évaluer la stabilité d'un algorithme, il faut définir les fonctions s_1 et s_2 et avoir un ensemble de données contenant des entrées similaires et dissimilaires. Pour mesurer la stabilité nous allons mesurer combien de fois l'équation E.2.2 est vraie. Quand les entrées sont similaires, cela fait une condition positive et sinon, c'est une condition négative. Le résultat fourni par l'algorithme est appelé une prédiction. Si l'équation E.2.2 est vraie pour un cas positif ($s_1(a,b) = 1$) alors c'est un vrai positif et si elle est fausse, c'est un faux négatif. Il en va inversement avec les cas négatifs. On peut alors définir quatre mesures de performance :

- Le taux de faux négatifs (FNR) est la probabilité qu'un événement soit prédit négatif quand il est positif.
- Le taux de faux positifs (FPR) est la probabilité qu'un événement soit prédit positif quand il est négatif.
- Le taux de fausse omission (FOR) est la probabilité qu'un événement soit positif quand il est prédit négatif.
- Le taux de fausse découverte (FDR) est la probabilité qu'un événement soit négatif quand il est prédit positif.

Elles doivent toutes être aussi proche de 0 que possible.

Maintenant que nous avons défini la stabilité d'un algorithme et comment l'évaluer nous allons pouvoir étudier la stabilité des algorithmes de description de la mise en page.

E.3 Description de la mise en page

Il y a deux types de mise en page : la mise en page physique et la mise en page logique. La mise en page physique contient uniquement les frontières des régions du document alors que la mise en page logique contient aussi le type de contenu et éventuellement la fonction de ces régions. Nous nous attacherons ici uniquement à la mise en page physique. La position des frontières des régions peut varier de quelques pixels sans pour autant changer la mise en page. Nous considérons donc que c'est avant tout la position des régions qui importe.

De même que la position des frontières des régions, une distance ou une aire peut changer un peu sans pour autant que la mise ne page soit différente. Ainsi, il convient d'éviter l'utilisation de ces valeurs. Nous proposons donc l'utilisation d'un

graphe non attribué calculé à partir de la triangulation de Delaunay des centres de gravité des régions. Le descripteur sera la matrice d'adjacence de ce graphe. Nous l'appelons le DLD (Delaunay Layout Descriptor). Le processus complet de calcul du descripteur de mise en page est représenté dans la Figure E.3.1.

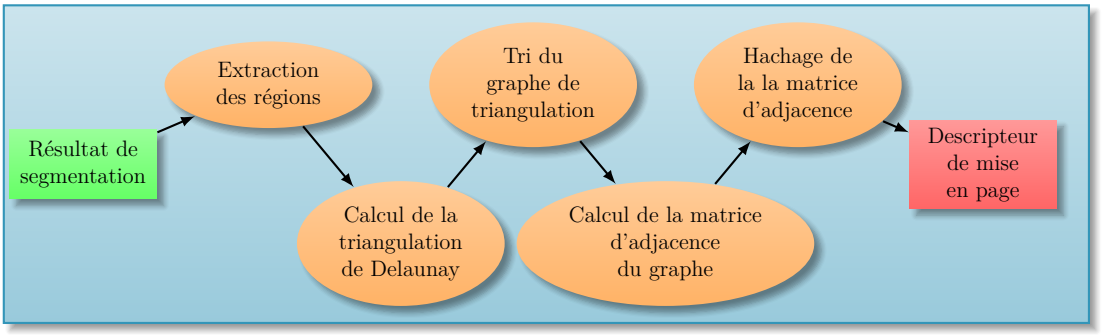


FIGURE E.3.1 : Processus de calcul du DLD.

Le descripteur a besoin d'un résultat de segmentation et décrit la mise en page de ce résultat. Le tri du graphe de la triangulation sert à obtenir un ordre des régions qui soit unique et déterministe. Cela permet de garantir la stabilité du descripteur.

Il subsiste quand même trois sources d'instabilité qui sont compensées par l'utilisation d'un algorithme de comparaison lors de la comparaison d'un descripteur avec une liste de descripteurs. Il a deux paramètres qui permettent de choisir entre la stabilité et le temps de calcul : l'erreur angulaire qui est acceptée et le nombre d'instabilités qui peuvent survenir en même temps sur un document.

Nous avons comparé ce descripteur (DLD) avec deux autres descripteurs de l'état de l'art. Celui de Gordo et Valveny [GV09] (G & V) et celui de Nakai et al. [NKI06] (LLAH). Ces deux algorithmes sont complètement instables si on considère qu'ils doivent produire des descripteurs identiques pour les copies d'un même document. Nous utilisons donc le calcul de distance qu'ils proposent pour la fonction de similarité s_2. En revanche le DLD produit des descripteurs identiques via son algorithme de comparaison et s_2 est donc l'égalité des descripteurs.

Le tableau E.2 résume les principaux résultats de l'évaluation de ces algorithmes sur un jeu de données de 990 images de 14 mises en pages dont une fournie par deux algorithmes de segmentation. La fonction de similarité s_1 pour ce jeu de données est l'indicatrice du fait que deux images contiennent la même mise en page (pas forcément produite par le même algorithme de segmentation). Le DLD dépasse très significativement l'état de l'art sur tous les critères.

Nous pouvons donc dire que le descripteur de mise en page de Delaunay (DLD) dépasse largement l'état de l'art en stabilité, en rapidité et en utilisation mémoire.

Indicateurs de performance	DLD	G & V	LLAH
FNR (%)	**0.8**	35.0	45.0
FPR (%)	**0.0**	2.8	3.7
FOR (%)	**0.1**	2.9	3.7
FDR (%)	**0.0**	34.9	44.9
Temps de calcul du descripteur	**0.01 s**	0.05 s	0.06 s
Temps de comparaison de deux descripteurs	**0.01 s**	0.06 s	0.07 s
Utilisation de la mémoire	**96 Mo**	3.9 Go	114 Go

TABLE E.2 : Résumé des résultats. Les meilleurs résultats sont en gras.

En outre, compte tenu de ses performances, il résout le problème de la description de la mise en page.

Enfin, puisque le DLD permet de comparer des mises en page, il peut comparer les résultats produits par un algorithme de segmentation. Cela va nous permettre d'étudier la stabilité des algorithmes de segmentation d'images de documents.

E.4 Segmentation d'images de documents

Afin d'avoir une vue d'ensemble des technologies disponibles pour la segmentation d'images de documents nous pouvons commencer par les passer en revue. Pour ce faire nous avons défini une typologie des algorithmes de segmentation représentée sur la Figure E.4.1. Elle permet de voir non seulement quels algorithmes traitent l'information de l'échelle globale à l'échelle locale (TD) et inversement (BU) mais aussi d'où viennent les limitations principales des algorithmes suivant la technique employée.

Les algorithmes du groupe 1 ne nécessitent pas d'entraînement, sont rapides et demandent peu de ressources de calcul. En revanche ils ne sont pas très flexibles. Il y a trois types d'algorithmes. Certains sont spécifiquement faits pour certains types de mise en page et utilisent des ensembles de règles ou des profils de projection. Le deuxième type d'algorithme utilise des filtres. Les caractéristiques des filtres reflètent les hypothèses faites sur la structure du document. Le dernier type d'algorithme se base sur la détection de lignes droites, de frontières carrées ou d'alignements d'espaces blancs.

Les algorithmes du groupe 2 sont plus flexibles. Ils essaient de s'adapter aux variations locales du document afin de pouvoir segmenter des documents plus variés avec le même algorithme. En contrepartie ces algorithmes ont plus de paramètres à choisir et peuvent nécessiter un entraînement. Les valeurs des paramètres sont bien souvent la principale source des limites de ces algorithmes. Les algorithmes

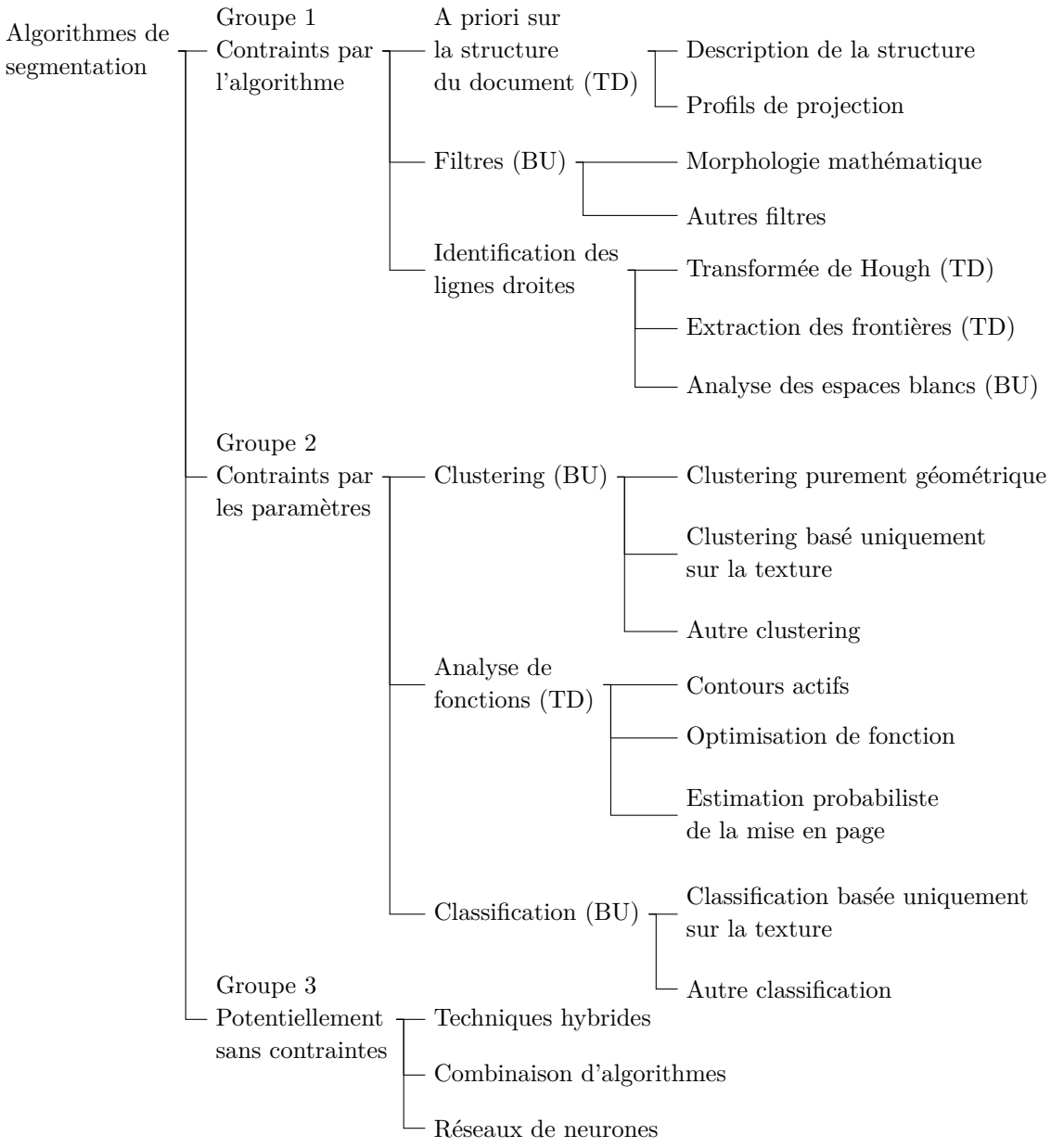


FIGURE E.4.1 : Typologie des algorithmes de segmentation d'images de documents. Nous précisons aussi les algorithmes descendants (TD) et montants (BU).

utilisent des techniques de clustering pour regrouper les éléments du document en régions. D'autres algorithmes utilisent des techniques d'analyse de fonction pour décrire les contours des régions. On compte parmi ces techniques : les contours

actifs, l'optimisation d'une fonction objectif ou encore l'estimation probabiliste de la mise en page du document. Enfin, il est possible de classer les éléments du document par type en utilisant des caractéristiques purement géométriques, des texture ou un ensemble de caractéristiques. Les techniques de classification ont l'avantage de fournir la nature des régions en plus de leurs frontières.

Enfin, les algorithmes du groupe 3 essaient d'être le plus flexible possible. Une solution consiste à créer des algorithmes hybrides qui utilisent plusieurs algorithmes en symbiose. Ces algorithmes sont malheureusement souvent complexes à mettre en œuvre. Il est aussi possible de combiner les résultats de plusieurs algorithmes parallèles. Enfin, les récents développements dans le domaine des réseaux de neurones ont trouvé des applications pour la segmentation d'images de documents. Ils nécessitent toutefois beaucoup de données d'apprentissage.

Compte tenu de la pléthore d'algorithmes existants et des difficultés à comparer leurs performances, nous avons choisi de comparer les performances de quatre algorithmes. PAL [CYL13] a gagné les deux compétitions de segmentation d'ICDAR en 2013. C'est un algorithme du premier groupe qui analyse les espaces blancs. Il en existe deux versions. Une qui fournit une segmentation au niveau des blocs (PALB) et une autre qui fournit une segmentation au niveau des lignes (PALL). Le troisième algorithme est celui de Kise et al. [KSI98] qui est considéré comme le meilleur algorithme dans [SKB08]. C'est un algorithme de type clustering purement géométrique. Il produit une segmentation de documents en noir et blanc avec des tesselations de Voronoï. Enfin, nous avons inclus un algorithme de segmentation d'images naturelles : JSEG [DM01]. Il utilise un clustering avec des caractéristiques génériques, notamment l'information de texture combinée avec des informations spatiales et une analyse multi-échelle.

Nous avons fait une validation croisée de la stabilité de ces algorithmes sur un jeu de données de 990 images contenant 18 copies de 55 documents. La fonction de similarité s_1 est l'indicatrice du fait que deux documents soient des copies du même document et la fonction s_2 est l'algorithme de comparaison du DLD. Il s'avère que les quatre algorithmes sont complètement instables. JSEG a un léger avantage, en particulier, il produit un nombre de régions qui varie moins qu'avec les autres algorithmes.

La première étape de PALB, PALL et de l'algorithme de Kise et al. est une binarization de l'image de document afin d'obtenir des composantes connexes. Cette binarization entraîne une perte d'information significative. Une avancée serait donc d'étendre la notion de composante connexe aux images en niveau de gris et en couleur.

E.5 Composantes connexes en couleurs

Il existe déjà des algorithmes qui essaient d'identifier des régions de base qui pourraient être assimilées à des composantes connexes. Ce sont les algorithmes de segmentation en superpixels. Cependant ils ont deux défauts. Tout d'abord, la plupart de ces algorithmes fournit un nombre de superpixels prédéfini par l'utilisateur. Or il est impossible de prédire le nombre de composantes connexes qu'il y aura dans une image avant de l'avoir analysée. Ensuite, de nombreux algorithmes ont une fonction de régularisation afin de faire des régions avec des frontières relativement simples. Cela implique qu'il peut leur être difficile de suivre des contours précis pour le texte. Une conséquence de ces deux contraintes est l'inaptitude des algorithmes de segmentation en superpixels à gérer des composantes connexes de tailles très différentes (changement d'échelle) au sein d'une même image.

Afin de pallier ces problèmes nous proposons un algorithme dont le seul objectif est de segmenter des régions de couleur uniforme ou avec un dégradé de couleurs.

Alors que la plupart des algorithmes ont une approche basée uniquement sur le contenu de l'image, nous adoptons une approche basée sur la perception de l'image par un observateur humain. A ce titre nous avons développé un modèle de vision humaine qui en reproduit quatre caractéristiques. La sensibilité spatiale de l'œil humain se situe entre 200 et 300 dpi. Sa sensibilité colorimétrique correspond à une distance Euclidienne de 2.3 environ dans l'espace colorimétrique Lab. La sensibilité colorimétrique de l'oeil humain varie avec la taille des régions qu'il voit. Par conséquent nous avons développé une distance spatio-colorimétrique :

$$\Delta_{Lab} = \Big[min\left(\left(\tfrac{S}{S_L}\right)^2, 1\right) \times \Delta L^2 + min\left(\left(\tfrac{S}{S_a}\right)^2, 1\right) \times \Delta a^2 + min\left(\left(\tfrac{S}{S_b}\right)^2, 1\right) \times \Delta b^2 \Big]^{1/2} \tag{E.5.1}$$

où ΔL, Δa et Δb sont les différences absolues entre les coordonnées Lab de deux régions. S est la surface minimale des deux régions et S_L, S_a et S_b sont des surfaces en dessous desquelles on ne distingue plus de différence de couleur pour chaque canal Lab. Enfin, la sensibilité de notre vision varie suivant le contraste local. Notre acuité s'accroît dans les zones de fort contraste et elle décroit dans les zones uniformes. Nous utilisons une transformée de domaine [GO11] pour représenter cet effet.

C'est ainsi que nous avons développé l'algorithme de segmentation en composantes connexes couleurs (CCC) : Watercolor et sa variante Smooth Watercolor. Leur processus est représenté sur la Figure E.5.1. La carte de distances colorimétriques est un élément critique de ces algorithmes car elle représente les variations locales de couleurs. C'est sur cette carte qu'est faite la segmentation en CCC. Les

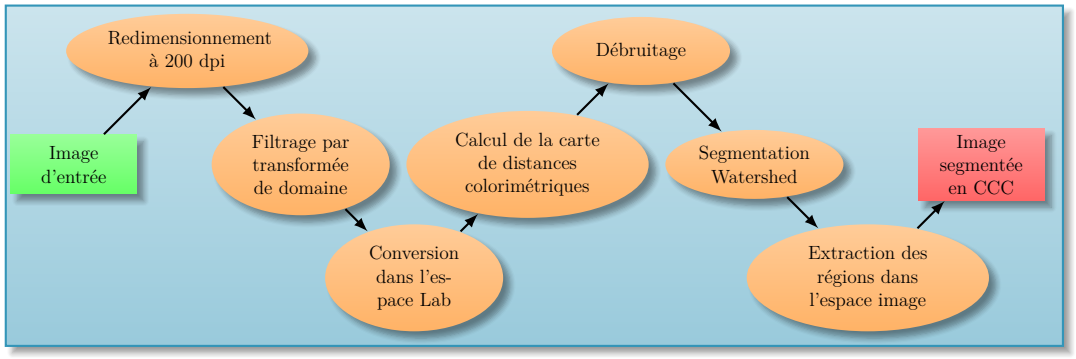


FIGURE E.5.1 : Algorithmes Watercolor et Smooth Watercolor

régions obtenues sur cette carte sont ensuite transposées dans l'espace image afin de fournir la segmentation finale. Watercolor et Smooth Watercolor diffèrent par le mode de calcul de leur carte de distances colorimétriques. Celle de Watercolor permet d'avoir des contours très précis mais produit beaucoup de petites régions superflues en particulier pour les gradients à de petites échelles. Smooth Watercolor produit moins de régions mais avec des frontières moins précises.

Nous avons ajouté un algorithme de post-traitement afin de tenir compte de notre distance spatio-colorimétrique et pour fusionner les régions dont la petite taille rend les couleurs indiscernables.

Lors d'une comparaison avec des algorithmes de l'état de l'art sur le benchmark de Berkeley [AMFM11], nos algorithmes ont une performance similaire à l'état de l'art. En revanche, lorsque nous les comparons sur le jeu de données de copies de documents que nous avons utilisé pour la segmentation, nos algorithmes produisent un nombre de régions trois à cinq fois plus stable que l'état de l'art.

En terme d'utilisation, Watercolor est utile pour ce qui est relatif à la détection de contours et à la reconnaissance de caractères. Pour ce dernier cas, il sera peut-être nécessaire d'utiliser une résolution de 300 dpi afin de tenir compte des problèmes de crénelage des petits caractères. Smooth Watercolor est plutôt à utiliser pour calculer les échelles des éléments présents dans une image et pour obtenir des régions globales. Le post-traitement est actuellement coûteux en temps de calcul. Si cela n'est pas un problème, il recommandé de l'utiliser sauf pour la reconnaissance de caractères.

Ces algorithmes résolvent deux problèmes importants dans la communauté de la vision par ordinateur : la détection de bordures et la détection d'échelle et ce sans paramètres. Il est aussi possible d'utiliser ces algorithmes pour fournir une décomposition de l'image en niveaux d'inclusion des régions les plus extérieures vers les régions les plus intérieures. Cela permet de binariser une image quelle que soit la couleur du texte et sa luminosité par rapport à l'arrière plan. Cela permet

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

(e) (f) (g) (h)

FIGURE E.5.2 : Résultats de Watercolor. (a)-(f) : image originale puis frontières des CCC. (g) et (h) : sélection des logos sans leur arrière plan avec leur niveau d'inclusion. (e) reproduite depuis [Str90].

aussi d'isoler facilement certains éléments du texte se trouvant à un niveau d'inclusion particulier. De plus, puisque ces algorithmes sont basés sur l'observateur et non sur le contenu, ils s'appliquent à tout type d'image (à condition d'avoir la résolution spatiale ou angulaire de l'image). La Figure E.5.2 montre les résultats de segmentation en CCC sur des images de document moderne, ancien et sur une image de scène naturelle ainsi qu'un exemple de sélection d'élément avec son niveau d'inclusion. L'arrière-plan de l'image est au niveau 1. Les bordures noires sont au niveau 2 et l'arrière plan des logos au niveau 3. En sélectionnant les niveau supérieurs ou égaux à 4, on garde uniquement les logos sans leur arrière-plan.

Enfin le but premier de ces algorithmes est d'étendre la définition de composantes connexes aux images en couleur et en niveau de gris ce qu'ils font très bien. Idéalement il faudrait encore améliorer la distance spatio-colorimétrique que nous avons définie et le modèle de décision de fusion des régions dans le post-traitement

mais notre travail montre que c'est une voie prometteuse.

Nous allons maintenant nous intéresser à la descriptions des différents éléments d'un document à commencer par son contenu textuel.

E.6 Reconnaissance optique de caractères

Le texte est un élément primordial et pour lequel quasiment aucune modification n'est acceptée. Il faut être capable d'extraire le texte avec une stabilité telle que, sur vingt copies d'un même texte, le texte extrait soit identique pour au moins dix-neuf copies.

Pour ce faire, notre approche consiste à réduire les ambiguïtés possibles par le biais d'un post-traitement que nous appelons une "réduction d'alphabet". Celui-ci consiste à projeter les caractères différents ayant la même apparence visuelle sur un seul et unique caractère. Par exemple, le "i" majuscule, le "L" minuscule et la barre verticale "|" sont tous remplacés par une barre verticale car c'est leur apparence visuelle. Cela ne changera quasiment rien pour un humain lisant le texte car il saura de quel caractère il s'agit. Nous avons étudié la possibilité de créer une confusion avec le dictionnaire Aspell. Les seules confusions possibles sont entre le "i" majuscule et le "L" minuscule. Dans un tel cas, le contexte permettrait encore de savoir de quel mot il s'agit. Ces confusions ont une probabilité d'apparaître de 0.0002 ce qui est acceptable.

Nous avons comparé la performance de deux logiciels de reconnaissance de caractères : Tesseract et FineReader avec et sans réduction d'alphabet sur un jeu de données de plus de 42 000 images de texte contenant de nombreuses variations de police, taille de texte, de typographie (gras, italique, les deux), de résolution et de matériel d'impression et de numérisation. Il y a 22 textes. Chacun a 1 584 variantes et chaque variante a 9 copies exactes (3 à chaque résolution). La fonction de similarité s_1 est l'indicatrice du fait que deux images sont les copies d'une même variante d'un même texte. La fonction de similarité s_2 est l'égalité des textes extraits des images.

Il s'avère que l'ajout de la réduction d'alphabet permet de diviser par deux le taux d'erreur par caractère des algorithmes et réduit leur taux de faux positif de 20 points. C'est une amélioration très significative. La meilleure performance est obtenue avec FineReader dans un cas idéal. Nous avons alors un taux de faux positifs (FNR) de 31%. C'est beaucoup mieux que Tesseract à 150 dpi qui a un FNR de 88% mais c'est encore insuffisant par rapport à notre objectif de moins de 5%.

Nous avons aussi mis en évidence le fait que quand toutes les autres variations sont présentes, il est possible de n'utiliser qu'une seule taille de police pour le jeu de données.

E.7 Hachage perceptuel d'image

Après le texte, les parties graphiques telles que les logos et les signatures manuscrites sont des éléments importants à sécuriser. Il existe déjà de nombreux algorithmes pour faire du hachage perceptuel d'image. On appelle ce hachage "perceptuel" car il vise à être uniquement sensible aux variations du contenu de l'image qui sont perceptibles par un humain. La plupart des algorithmes existants se focalisent sur la robustesse de l'algorithme de hachage, autrement dit, sa capacité à fournir des signatures proches y compris quand l'image subit des dégradations (bien souvent de synthèse). Il n'y a que très peu d'articles qui s'intéressent à la robustesse au bruit d'impression et de numérisation ainsi qu'à la détection de modifications. Notre problème ne devrait pas non plus être confondu avec celui qui consiste à trouver des images presque similaires. Les algorithmes pour cette tâche sont très peu sensibles aux petites modifications et donc ne conviennent pas du tout à nos besoins.

En dehors des contraintes liées à la stabilité et à la sensibilité (aussi appelée fragilité) de l'algorithme, la signature produite doit avoir une taille réduite. Elle a vocation à être insérée dans le document qui est sécurisé par exemple par le biais d'un 2D-Doc. Les descriptions de la mise en page et du texte avaient l'obligation d'être parfaitement exactes entre deux copies d'un même document. Cela permet d'utiliser un hachage cryptographique qui est très compact. Ce n'est pas possible avec le hachage perceptuel à cause de la difficulté d'isoler précisément ce qui est significatif dans une image. Il va falloir produire des signatures plus grandes que l'on comparera ensuite. Compte tenu de la capacité du 2D-Doc et des signatures de tous les éléments d'un document, nous avons fixé une taille maximale de 500 octets.

Nous avons étudié la faisabilité de l'utilisation de points d'intérêts pour décrire l'image. Nous nous sommes intéressés à Harris [HS88], GFTT [ST94], SIFT [Low04], SURF [BTG06], FAST [RD06], CenSurE [AKB08], ORB [RRKB11] et BRISK [LCS11]. Leur nature parcimonieuse pourrait permettre une description compacte de l'image. Nos résultats prouvent que l'utilisation de la position des points d'intérêts combinée avec un descripteur de leur agencement local tel que LLAH permet de mieux décrire l'image que les descripteurs associés aux détecteurs des points d'intérêts. Quoi qu'il en soit, leur stabilité est insuffisante.

Nous proposons donc un algorithme de hachage perceptuel basé sur une représentation grossière de l'image : ASYCHA. Le fonctionnement global de la comparaison de deux images est montré sur la Figure E.7.1. Il repose sur le hachage d'une image originale. Lorsqu'on souhaite comparer une image de test avec l'image originale, on calcule une signature spécifique que l'on peut ensuite comparer pour décider de la similarité ou non des images.

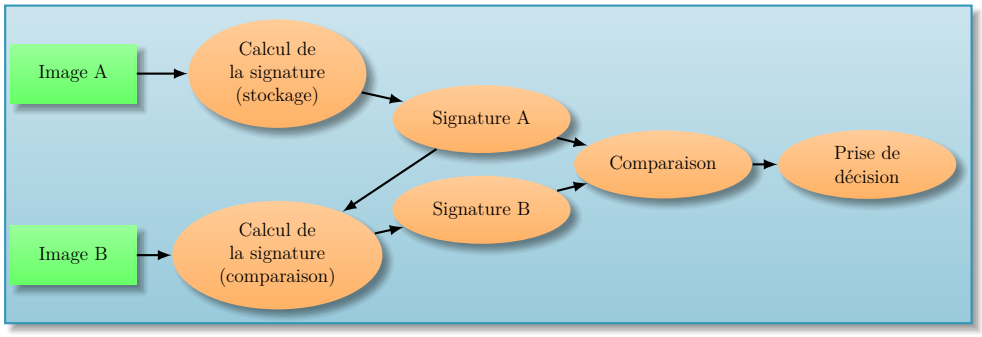


FIGURE E.7.1 : Processus de hachage et de comparaison d'images ASYCHA.

Le processus de hachage est représenté sur la Figure E.7.2. Le choix des méthodes pour effectuer chaque étape est important pour la stabilité et la sensibilité de l'algorithme. En particulier nous avons combiné deux algorithmes de clustering pour indexer l'image sur 32 couleurs sans perdre les couleurs peu représentées tout en ayant un algorithme déterministe. L'étape de normalisation vise à compenser les variations de luminosité lors du processus d'impression et de numérisation.

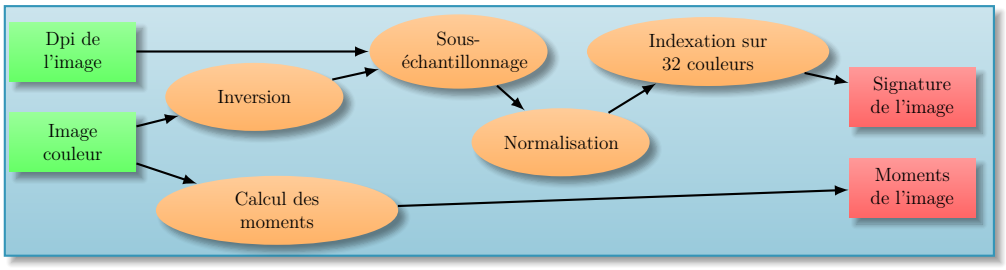


FIGURE E.7.2 : Algorithme de hachage de l'image à but de stockage.

Le calcul de la signature pour comparaison est similaire mais inclut un processus de recalage d'image pour compenser la rotation et les changements d'échelle. La comparaison des deux signatures permet de gérer les translations pour ensuite calculer la distance de Hausdorff entre les deux signatures. La prise de décision pour la comparaison se base sur cinq critères qui doivent être satisfaits : la différence d'orientation, le ratio d'échelle, les tailles des images dont le ratio de leur intersection avec leur surface obtenu après correction de la translation et enfin, leur distance de Hausdorff.

Nous avons comparé ASYCHA avec deux autres algorithmes de l'état de l'art. L'algorithme de Wu et al. [WZN09] est censé être robuste au bruit d'impression et de numérisation. L'algorithme de Venkatesan et al. [VKJM00] quant à lui utilise une technique de décomposition de l'image en rectangles qui donne de bons résultats et qui a été reprise par plusieurs auteurs. Pour les comparer nous avons créé

FIGURE E.7.3 : Différentes images de logos utilisées.

Mes. de Perf.	Venkatesan	Wu	ASYCHA
FNR (%)	**0,3**	5.2	8.2
FPR (%)	8.9	39.3	$\mathbf{3.2 \times 10^{-3}}$
FOR (%)	2.7×10^{-2}	3.4×10^{-3}	$\mathbf{3.3 \times 10^{-3}}$
FDR (%)	49.9	99.9	**8.0**
Taille de la signature	500 octets	**50 bits**	186 à 1174 octets médiane 427 octets

TABLE E.3 : Résultats des différentes méthodes. Toutes les valeurs doivent être aussi basses que possible.

deux jeux de données. Un jeu de presque 2 000 images de signatures manuscrites avec 18 copies de chaque signature soit plus de 34 000 images et un jeu de 18 copies de 200 logos en trois tailles différentes soit plus de 10 000 images. Les signatures ont la particularité de contenir des imitations d'une même signature ainsi que plusieurs exemplaires d'une signature par le même auteur. Les logos contiennent aussi plusieurs variations d'un même logo. Nous ne pouvons pas montrer les images de signatures mais la Figure E.7.3 montre quelques images de logos. La fonction de similarité s_1 est l'indicatrice du fait que deux images sont les copies d'une même image d'origine (logo ou exemplaire de signature manuscrite). La fonction de similarité s_2 est fournie par l'algorithme de comparaison pour ASYCHA et par la comparaison de la distance entre les deux images et un seuil pour les autres algorithmes.

Le tableau E.3 compare les résultats des différents algorithmes. L'algorithme de Venkatesan a le meilleur taux de faux négatifs (FNR) et celui de Wu a la plus petite signature. Néanmoins ASYCHA a de très loin le meilleur taux de faux positifs (FPR) et de fausses découvertes (FDR) tout en gardant un FNR relativement bas. En outre la taille de sa signature, est majoritairement sous la limite des 500 octets celle-ci étant surtout dépassée pour des signatures manuscrites de grandes

dimensions.

L'algorithme que nous avons proposé est donc meilleur que l'état de l'art notamment en produisant beaucoup moins de faux positifs et donc en étant plus à même de détecter les modifications dans les parties graphiques d'un document. Il faudra toutefois encore l'améliorer pour réduire la taille de la signature produite.

E.8 Conclusion et perspectives

Le travail présenté ici fournit des avancées significatives dans le domaine de l'analyse d'images de documents. Nous avons proposé un formalisme générique pour la stabilité d'un algorithme et une méthodologie d'évaluation correspondante. Celle-ci a été appliquée à la description de l'agencement d'un ensemble de points qui peut être étendu à la description de mise en page, à la segmentation d'images de document, à la segmentation d'images en composantes connexes en couleur et en superpixels, à la reconnaissance optique de caractères, aux détecteurs et descripteurs de points d'intérêt et enfin au hachage d'image perceptuel. Cela a conduit au développement de nouveaux algorithmes qui dépassent l'état de l'art et apportent des contributions importantes. Ces algorithmes ont aussi résolu un certain nombre de problématiques de la communauté telles que la description de la mise en page, l'extension de la définition de composante connexe au images en couleur et en niveau de gris, ou encore la détection de contours et d'échelle sans paramètres.

Il reste toutefois encore beaucoup de travail et nous avons ouvert un nouveau champ de recherche : celui d'algorithmes d'analyse d'images de documents stables. Nous espérons que les algorithmes que nous avons développés pourront encore être améliorés et que la stabilité d'autres types d'algorithmes sera étudiée. Certains points se sont avérés problématiques et nous souhaiterions attirer l'attention dessus. En particulier l'algorithme de k-moyennes n'a pas d'indicateur de qualité intrinsèque et l'estimation de la qualité de ses résultats quand on l'exécute plusieurs fois n'a pas été étudiée non plus. Le crénelage est un vrai problème, surtout pour la reconnaissance de caractères, et l'étude de représentations d'image plus isotropes qu'une matrice carrée pourrait être intéressante. Enfin, il serait possible d'avoir une meilleur fidélité colorimétrique (et donc moins de bruit) en utilisant plus de canaux de couleurs. Des avancées dans ces domaines permettraient de fournir des algorithmes plus performant et nous espérons donc qu'elles auront lieu.

Bibliography

[AAK08] Sufyan Ababneh, Rashid Ansari, and Ashfaq Khokhar. Scalable multimedia-content integrity verification with robust hashing. In *IEEE International Conference on Electro/Information Technology*, pages 263–266. IEEE, 2008.

[Abb13] Abbyy. Finereader, 2013.

[ABH98] Badr Al-Badr and Robert M. Haralick. A segmentation-free approach to text recognition with application to Arabic text. *International Journal on Document Analysis and Recognition (IJDAR)*, 1(3):147–166, 1998.

[ABPP09] Apostolos Antonacopoulos, David Bridson, Christos Papadopoulos, and Stefan Pletschacher. A realistic dataset for performance evaluation of document layout analysis. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 296–300. IEEE, 2009.

[ACKEs15] Abedelkadir Asi, Rafi Cohen, Klara Kedem, and Jihad El-sana. Simplifying the reading of historical manuscripts. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 826–830. IEEE, 2015.

[ACPP11] Apostolos Antonacopoulos, Christian Clausner, Christos Papadopoulos, and Stefan Pletschacher. Historical document layout analysis competition. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 1516–1520. IEEE, 2011.

[ACPP13a] Apostolos Antonacopoulos, Christian Clausner, Christos Papadopoulos, and Stefan Pletschacher. Competition on Historical Book Recognition (HBR 2013). In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 1459–1463. IEEE, aug 2013.

[ACPP13b] Apostolos Antonacopoulos, Christian Clausner, Christos Papadopoulos, and Stefan Pletschacher. Competition on Historical Newspaper Layout Analysis (HNLA 2013). In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 1454–1458. IEEE, aug 2013.

[ACPP15] Apostolos Antonacopoulos, Christian Clausner, Christos Papadopoulos, and Stefan Pletschacher. ICDAR2015 Competition on recognition of documents with complex layouts. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 1151–1155. IEEE, 2015.

[AD09] Mudit Agrawal and David Doermann. Voronoi++: a dynamic page segmentation approach based on Voronoi and Docstrum features. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 1011–1015. IEEE, 2009.

[AD10] Mudit Agrawal and David Doermann. Context-aware and content-based dynamic Voronoi page segmentation. In *Document Analysis Systems (DAS)*, pages 73–80. IEEE, 2010.

[ADHP09] Daniel Aloise, Amit Deshpande, Pierre Hansen, and Preyas Popat. NP-hardness of Euclidean sum-of-squares clustering. *Machne Learning*, 75(2):245–248, 2009.

[AGB07] Apostolos Antonacopoulos, Basilis Gatos, and David Bridson. ICDAR2007 Page segmentation competition. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 1279–1283. IEEE, 2007.

[AGC13] Arpit Agarwal, Ritu Garg, and Santanu Chaudhury. Greedy search for active learning of OCR. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 837–841. IEEE, 2013.

[AKB08] Motilal Agrawal, Kurt Konolige, and Morten Rufus Blas. CenSurE : Center Surround Extremas for realtime feature detection and matching. In *European Conference on Computer Vision (ECCV)*, pages 102–115. Springer, 2008.

[AMFM11] Pablo Arbelaez, Michael Maire, Charless C. Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 33(5):898–916, 2011.

[AMVK99] Mohamed Abdel-Mottaeb, Gandhi Vaithilingam, and Santhana Krishnamachari. Signature-based image identification. In *Multimedia Systems and Applications*, pages 22–28. Springer, 1999.

[ANT13] ANTS. Spécifications techniques des codes à barres 2D-Doc. Technical report, ANTS, 2013.

[APBP09] Apostolos Antonacopoulos, Stefan Pletschacher, David Bridson, and Christos Papadopoulos. ICDAR2009 Page segmentation competition. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 1370–1374. IEEE, 2009.

[AS02] T. V. Ashwin and P. S. Sastry. A font and size-independent OCR system for printed Kannada documents using support vector machines. *Sadhana*, 27(1):35–58, 2002.

[ASS+12] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. SLIC superpixels compared to state-of-the-art superpixel methods. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 34(11):2274–2282, nov 2012.

[ASU10] Fawad Ahmed, M. Y. Siyal, and Vali Uddin Abbas. A secure and robust hash-based scheme for image authentication. *Signal Processing*, 90(5):1456–1470, 2010.

[ÁZ13] Francisco Álvaro and Richard Zanibbi. A shape-based layout descriptor for classifying spatial relationships in handwritten math. In *Symposium on Document Engineering (DocEng)*, pages 123–126. ACM, 2013.

[BACP14] P. Barlas, S. Adam, C. Chatelain, and Thierry Paquet. A typed and handwritten text block segmentation system for heterogeneous and complex documents. In *Document Analysis Systems (DAS)*, pages 46–50. IEEE, apr 2014.

[Bal81] D. H. Ballard. Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition (PR)*, 11(2):111–122, 1981.

[Bar92] Peter G. J. Barten. Physical model for the contrast sensitivity of the human eye. In *Visual Processing, and Digital Display*, pages 57–72. SPIE, 1992.

[BASB10] Syed Saqib Bukhari, Mayce Ibrahim Ali Al Azawi, Faisal Shafait, and Thomas M. Breuel. Document image segmentation using discriminative learning over connected components. In *Document Analysis Systems (DAS)*, pages 183–190. IEEE, 2010.

[BC69] C. Blakemore and F. W. Campbell. On the existence of neurones in the human visual system selectively sensitive to the orientation and size of retinal images. *Journal of Physiology*, 203(1):237–260, 1969.

[BC06] Henry Baird and Matthew R. Casey. Towards versatile document analysis systems. In Horst Bunke and A. Lawrence Spitz, editors, *Document Analysis Systems*

(DAS), Lecture Notes in Computer Science, pages 280–290, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[BC12] Cléo Baras and François Cayre. 2D bar-codes for authentication: a security approach. In *European Signal Processing Conference (EUSIPCO)*, pages 1760–1766. IEEE Sign. Proc. Soc. Press, 2012.

[BC13] Cléo Baras and François Cayre. Vers un modèle de canal réaliste pour l'analyse de la sécurité du processus d'authentification par code matriciel 2D. In *Colloque GRETSI*, pages 2–5. GRETSI, 2013.

[BCC+15] Jean-christophe Burie, J. Chazalon, M. Coustaty, Sébastien Eskenazi, and M. M. Luqman. ICDAR2015 Competition on Smartphone Document Capture and OCR (SmartDoc). In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 1161–1165. IEEE, 2015.

[BCM11] Tiago C. Bockholt, George D. C. Cavalcanti, and Carlos A. B. Mello. Document image retrieval with morphology-based segmentation and features combination. In Gady Agam and Christian Viard-Gaudin, editors, *Document Recognition and Retrieval (DRR)*, pages 787415–787415–12. SPIE, jan 2011.

[BE02] Olivier Bousquet and Andre Elisseeff. Stability and generalization. *Journal of Machine Learning Research*, 2:499–526, 2002.

[Beu91] S. Beucher. The watershed transformation applied to image segmentation. In *Scanning Microscopy International*, pages 299—-314. Scanning Electron Microscopy, Inc., 1991.

[BG12] John Bryson and Patrick Gallagher. Secure Hash Standard (SHS), 2012.

[BGO+10] Christian Bhöm, Sebastian Goebl, Annahita Oswald, Claudia Plant, Michael Plavinski, and Bianca Wackersreuther. Integrative parameter-free clustering of data with mixed type attributes. In Mohammed J. Zaki, Jeffrey Xu Yu, B. Ravindran, and Vikram Pudi, editors, *Advances in Knowledge Discovery and Data Mining*, pages 38–47. Springer, 2010.

[BI11] Micheal Baechler and Rolf Ingold. Multi resolution layout analysis of medieval manuscripts using dynamic MLP. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 1185–1189. IEEE, 2011.

[BK98] Sushil Bhattacharjee and Martin Kutter. Compression tolerant image authentication. In *International Conference on Image Processing (ICIP)*, pages 435 – 439. IEEE, 1998.

[BKMA10] Mohamed Benjelil, Slim Kanoun, Rémy Mullot, and Adel M. Alimi. Complex documents images segmentation based on steerable pyramid features. *International Journal on Document Analysis and Recognition (IJDAR)*, 13(3):209–228, 2010.

[BL79] S. Beucher and C. Lantuejoul. Use of watersheds in contour detection. In *International workshop on image processing: real time edge and motion detection/estimation*, pages 2.1–2.12, 1979.

[BLI13] Micheal Baechler, Marcus Liwicki, and Rolf Ingold. Text line extraction using DMLP classifiers for historical manuscripts. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 1029–1033. IEEE, aug 2013.

[Blo91] Dan S. Bloomberg. Multiresolution morphological approach to document image analysis. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 963–971. IEEE, 1991.

[BMA14] Mohamed Benjlaiel, Remy Mullot, and Adel M. Alimi. Multi-oriented handwritten annotations extraction from scanned documents. In *Document Analysis Systems (DAS)*, pages 126–130. IEEE, 2014.

[BMI07] Paulo V. K. Borges, Joceli Mayer, and Ebroul Izquierdo. A practical protocol for digital and printed document authentication. In *European Signal Processing Conference (EUSIPCO)*, pages 2529–2533. IEEE Sign. Proc. Soc. Press, 2007.

[BMP02] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape matching and object recognition using shape contexts. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 24(24):509–522, 2002.

[BMvL12] Sébastien Bubeck, Marina Meila, and Ulrike von Luxburg. How the initialization affects the stability of the k-means algorithm. *ESAIM: Probability and Statistics*, 16:436–452, 2012.

[BNV13] Djamal Belazzougui, Gonzalo Navarro, and Daniel Valenzuela. Improved compressed indexes for full-text document retrieval. *Journal of Discrete Algorithms*, 18:3–13, jan 2013.

[Bre08] Thomas M. Breuel. The OCRopus open source OCR system. In *Document Recognition and Retrieval (DRR)*, page 68150F. SPIE, 2008.

[BRY+15] Xu Bin, Li Ruiguang, Liu Yashu, Yan Hanbing, Li Siyuan, and Zhang Honggang. Filtering Chinese image spam using pseudo-OCR. *Chinese Journal of Electronics*, 24(1):134–139, 2015.

[BSB08] Syed Saqib Bukhari, Faisal Shafait, and Thomas M. Breuel. Segmentation of curled textlines using active contours. In *Document Analysis Systems (DAS)*, pages 270–277. Springer, 2008.

[BSB09a] Syed Saqib Bukhari, Faisal Shafait, and Thomas M. Breuel. Coupled snakelet model for curled textline segmentation of camera-captured document images. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 61–65. IEEE, 2009.

[BSB09b] Syed Saqib Bukhari, Faisal Shafait, and Thomas M. Breuel. Ridges based curled textline region detection from grayscale camera-captured document. In *Computer Analysis of Images and Patterns (CAIP)*, pages 173–180. Springer-Verlag, 2009.

[BSB09c] Syed Saqib Bukhari, Faisal Shafait, and Thomas M. Breuel. Script-independent handwritten textlines segmentation using active contours. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 446–450. IEEE, 2009.

[BSB11a] Syed Saqib Bukhari, Faisal Shafait, and Thomas M. Breuel. High performance layout analysis of Arabic and Urdu document images. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 1275–1279. IEEE, 2011.

[BSB11b] Syed Saqib Bukhari, Faisal Shafait, and Thomas M. Breuel. Improved document image segmentation algorithm using multiresolution morphology. In Gady Agam and Christian Viard-Gaudin, editors, *Document Recognition and Retrieval (DRR)*, pages 78740D–78740D–8. SPIE, jan 2011.

[BSB11c] Syed Saqib Bukhari, Faisal Shafait, and Thomas M. Breuel. Text-line extraction using a convolution of isotropic Gaussian filter with a set of line filters. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 579–583. IEEE, 2011.

[BSB13a] Syed Saqib Bukhari, Faisal Shafait, and Thomas M. Breuel. Coupled snakelets for curled text-line segmentation from warped document images. *International Journal on Document Analysis and Recognition (IJDAR)*, 16(1):33–53, mar 2013.

[BSB13b] Syed Saqib Bukhari, Faisal Shafait, and Thomas M. Breuel. Towards generic text-line extraction. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 748–752. IEEE, aug 2013.

[BTG06] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF : Speeded Up Robust Features. In *European Conference on Computer Vision (ECCV)*, pages 404–417. Springer, 2006.

[BTGK+15] Romain Bertrand, Oriol Ramos Terrades, Petra Gomez-Krämer, Patrick Franco, and Jean-Marc Ogier. A conditional random field model for font forgery detection. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 576–580. IEEE, 2015.

[Buc80] G. Buchsbaum. A spatial processor model for object colour perception. *Journal of the Franklin Institute*, 310(1):1–26, 1980.

[BUhAS13] Thomas M. Breuel, Adnan Ul-hasan, Mayce Al Azawi, and Faisal Shafait. High-performance OCR for printed English and Fraktur using LSTM networks. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 683–687. IEEE, 2013.

[BVFV09] V. L. Blankers, C. E. Van Den Heuvel, K. Y. Franke, and L. G. Vuurpijl. The ICDAR 2009 signature verification competition. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 1403–1407. IEEE, 2009.

[BW03] Andrew D. Bagdanov and Marcel Worring. First order Gaussian graphs for efficient structure classification. *Pattern Recognition (PR)*, 36:1311–1324, 2003.

[CAP12] Christian Clausner, Apostolos Antonacopoulos, and Stefan Pletschacher. A robust hybrid approach for text line segmentation. In *International Conference on Pattern Recognition (ICPR)*, pages 335–338. IEEE, 2012.

[CBC+15] Elodie Carel, Jean-christophe Burie, Vincent Courboulay, Jean-Marc Ogier, and Vincent Poulain d'Andecy. Multiresolution approach based on adaptive superpixels for administrative documents segmentation into color layers. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 566–570. IEEE, 2015.

[CD92] C. H. Chen and J. L. Decurtins. A segmentation-free approach to OCR. In *Workshop on Applications of Computer Vision*, pages 190 – 196. IEEE, 1992.

[CD93] C. H. Chen and J. L. DeCurtins. Word recognition in a segmentation-free approach to OCR. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 573–576. IEEE, 1993.

[Cha07] Biduyt B. Chaudhuri. *Digital document processing. major directions and recent advances.* Springer, 2007.

[CIE14] CIE. CIE 208:2014 effect of stimulus size on colour appearance. *Color Research & Application*, 39(5):518, 2014.

[CK09] Antonio Clavelli and Dimosthenis Karatzas. Text segmentation in colour posters from the spanish civil war era. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 181–185. IEEE, 2009.

[CL96] Richard G. Casey and Eric Lecolinet. A survey of methods and strategies in character segmentation. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 18(7):690–706, 1996.

[CLC15] Cérès Carton, Aurélie Lemaitre, and Bertrand Coüasnon. Automatic and interactive rule inference without ground truth. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 696–700. IEEE, 2015.

[CLMS01] F. Cesarini, M. Lastri, S. Marinai, and G. Soda. Encoding of modified X-Y trees for document classification. *International Conference on Document Analysis and Recognition (ICDAR)*, 2001.

[CLQL11] Xiaoming Chen, Wanquan Liu, Huining Qiu, and Jianhuang Lai. APSCAN : A parameter free algorithm for clustering. *Pattern Recognition Letters (PRL)*, 32(7):973–986, 2011.

[CLSF10] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. BRIEF : Binary Robust Independent Elementary Features. In *European Conference on Computer Vision (ECCV)*, pages 778–792. Springer Berlin Heidelberg, 2010.

[CLW+99] Edward Chang, Chen Li, James Wang, Peter Mork, and Gio Wiederhold. Searching near-replicas of images via clustering. In *Multimedia Storage and Archiving Systems*, pages 281–295. IEEE, 1999.

[CM02] Dorin Comaniciu and Peter Meer. Mean Shift : a robust approach toward feature space analysis. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 24(5):603–619, 2002.

[CMM13] Christian Conrad, Matthias Mertz, and Rudolf Mester. Contour-relaxed superpixels. In *International Conference on Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 280–293. Springer, 2013.

[CMO07] Eugenio Cesario, Giuseppe Manco, and Riccardo Ortale. Top-down parameter-free clustering of high-dimensional categorical data. *Transactions on knowledge and data engineering*, 19(12):1607–1624, 2007.

[Coü06] Bertrand Coüasnon. DMOS, a generic document recognition method: application to table structure analysis in a general and in a specific way. *International Journal on Document Analysis and Recognition (IJDAR)*, 8(2-3):111–122, 2006.

[Cou14] Council of the European Union. Regulation (EU) No 910/2014 of the European Parliament and of the Council of 23 July 2014 on electronic identification and trust services for electronic transactions in the internal market and repealing Directive 1999/93/EC. *Official Journal of the European Union*, 57(L 257):73–114, 2014.

[CRG13] T. Chattopadhyay, V. Ramu Reddy, and Utpal Garain. Automatic selection of binarization method for robust OCR. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 1170–1174. IEEE, aug 2013.

[CSL+15] Kai Chen, Mathias Seuret, Marcus Liwicki, Jean Hennebert, and Rolf Ingold. Page segmentation of historical document images with convolutional autoencoders. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 1011–1015. IEEE, 2015.

[CSN05] K. Chellapilla, P. Simard, and R. Nickolov. Fast optical character recognition through glyph hashing for document conversion. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 829–833. IEEE, 2005.

[CSU11] Kunal Narayan Chaudhury, Daniel Sage, and Michael Unser. Fast O(1) bilateral filtering using trigonometric range kernels. *Transactions on Image Processing (TIP)*, 20(12):3376–3382, 2011.

[CT14] Francisco Cruz and Oriol Ramos Terrades. EM-based layout analysis method for structured documents. In *International Conference on Pattern Recognition (ICPR)*, pages 315–320. IEEE, aug 2014.

[CW09] Yen Lin Chen and Bing Fei Wu. A multi-plane approach for text segmentation of complex document images. *Pattern Recognition (PR)*, 42(7):1419–1444, 2009.

[CWL+14] Kai Chen, Hao Wei, Marcus Liwicki, Jean Hennebert, and Rolf Ingold. Robust text line segmentation for historical manuscript images using color and texture. In *International Conference on Pattern Recognition (ICPR)*, pages 2978–2983. IEEE, aug 2014.

[CWLW98] Edward Chang, James Ze Wang, Chen Li, and Gio Wiederhold. RIME : A Replicated Image detector for the world-wide web. Technical report, Department of Computer Science, Stanford University, 1998.

[CYL13] Kai Chen, Fei Yin, and Cheng-Lin Liu. Hybrid page segmentation with efficient whitespace rectangles extraction and grouping. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 958–962. IEEE, aug 2013.

[CZ01] Sen-Ching S; Cheung and Avideh Zakhor. Video similarity detection with video signature clustering. In *International Conference on Image Processing (ICIP)*, pages 649–652. IEEE, 2001.

[DDLM05] C De Roover, C De Vleeschouwer, Frédéric Lefèbvre, and Benoit Macq. Robust image hashing based on radial variance of pixels. In *International Conference on Image Processing (ICIP)*, pages III–77–80. IEEE, 2005.

[Deb15] Narayan Debnath. *Batul the great.* Deb Sahitya Kutir Pvt. Ltd., 2015.

[DKS11] Markus Diem, Florian Kleber, and Robert Sablatnig. Text classification and document layout analysis of paper Fragments. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 854–858. IEEE, 2011.

[DKS13] Markus Diem, Florian Kleber, and Robert Sablatnig. Text line detection for heterogeneous documents. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 743–747. IEEE, aug 2013.

[DM01] Yining Deng and B. S. Manjunath. Unsupervised segmentation of color-texture regions in images and video. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 23(8):800–810, 2001.

[DP73] David H. Douglas and Thomas K. Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 10(2):112–122, 1973.

[DPB08] Xiaojun Du, Wumo Pan, and Tien D. Bui. Text line segmentation in handwritten documents using Mumford-Shah model. *Pattern Recognition (PR)*, 42(12):3136–3145, 2008.

[dSK15] Samuel de Sousa and Walter G Kropatsch. Graph formulation as the minimum-weight maximum-entropy problem. In Cheng-Lin Liu, Bin Luo, Walter G. Kropatsch, and Jian Cheng, editors, *Graph-Based Representations in Pattern Recognition*, pages 13–22. Springer, 2015.

[DSS99] Jana Dittmann, Arnd Steinmetz, and Ralf Steinmetz. Content-based digital signature for motion pictures authentication and content-fragile watermarking. In *International Conference on Multimedia Computing and Systems*, pages 209–213. IEEE, 1999.

[EGKO15a] Sébastien Eskenazi, Petra Gomez-Krämer, and Jean-Marc Ogier. Let 's be done with thresholds ! In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 851–855. IEEE, 2015.

[EGKO15b] Sébastien Eskenazi, Petra Gomez-Krämer, and Jean-Marc Ogier. When document security brings new challenges to document analysis. In *International Workshop on Computational Forensics (IWCF)*, pages 104–116. SPIE, 2015.

[EGKO16] Sébastien Eskenazi, Petra Gomez-Krämer, and Jean-Marc Ogier. Evaluation of the stability of four document segmentation algorithms. In *Document Analysis Systems (DAS)*, pages 1–6. IEEE, 2016.

[EMS94] Floriana Esposito, Donato Malerba, and Giovanni Semeraro. Multistrategy learning for document recognition. *Applied Artificial Intelligence an International Journal*, 8(1):33–84, 1994.

[Eur99] European Parliament and Council. Directive 1999/93/EC of the European Parliament and of the Council of 13 December 1999 on a community framework for electronic signatures. *Official Journal of the European Communities*, 95(837):1–9, 1999.

[Fat09] Raanan Fattal. Edge-avoiding wavelets and their applications. *ACM Transactions on Graphics*, 28(3):1, 2009.

[FBEB09] Stefano Ferilli, Marenglen Biba, Floriana Esposito, and Teresa M.A. Basile. A distance-based technique for non-Manhattan layout analysis. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 231–235. IEEE, 2009.

[FBG+14] Andreas Fischer, Micheal Baechler, Angelika Garz, Marcus Liwicki, and Rolf Ingold. A combined system for text line extraction and handwriting recognition in historical documents. In *Document Analysis Systems (DAS)*, pages 71–75. IEEE, apr 2014.

[FFLS08] Zeev Farbman, Raanan Fattal, Dani Lischinski, and Richard Szeliski. Edge-preserving decompositions for multi-scale tone and detail manipulation. *ACM Transactions on Graphics*, 27(3):1, 2008.

[FG00] Jiri Fridrich and M. Goljan. Robust hash functions for digital watermarking. *International Conference on Information Technology: Coding and Computing*, pages 178–183, 2000.

[FH04] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision (IJCV)*, 59(2):167—-181, 2004.

[FIB+10] Andreas Fischer, Emanuel Indermühle, Horst Bunke, Gabriel Viehhauser, and Michael Stolz. Ground truth creation for handwriting recognition in historical documents. In *Document Analysis Systems (DAS)*, pages 3–10. Springer, 2010.

[Fri99] Jiri Fridrich. Robust bit extraction from images. In *International Conference on Multimedia Computing and Systems*, pages 536–540. IEEE, 1999.

[FT12] Francisco Cruz Fern and Oriol Ramos Terrades. Document segmentation using relative location features. In *International Conference on Pattern Recognition (ICPR)*, pages 1562–1565. IEEE, 2012.

[FV09] Claudie Faure and Nicole Vincent. Simultaneous detection of vertical and horizontal text lines based on perceptual organisation. In Kathrin Berkner and Laurence Likforman-Sulem, editors, *Document Recognition and Retrieval (DRR)*, pages 72470M–72470M–8. SPIE, jan 2009.

[FYM+13] F. Furukori, S. Yamazaki, T. Miyagishi, K. Shirai, and M. Okamoto. An OCR System with OCRopus for scientific documents containing mathematical formulas. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 1207–1211. IEEE, 2013.

[GA09] Emmanuèle Grosicki and Haikal El Abed. ICDAR 2009 Handwriting recognition competition. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 1398–1402. IEEE, 2009.

[GEA11] Emmanuele Grosicki and Haikal El-Abed. ICDAR 2011 French handwriting recognition competition. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 1459–1463. IEEE, 2011.

[GELE08] Djamel Gaceb, Véronique Eglin, Frank Lebourgeois, and Hubert Emptoz. Application of graph coloring in physical layout segmentation. In *International Conference on Pattern Recognition (ICPR)*, pages 1–4. IEEE, dec 2008.

[GH04] Henri Gilbert and Helena Handschuh. Security analysis of SHA-256 and sisters. In Springer Berlin Heidelberg, editor, *Selected Areas in Cryptography*, pages 175–193. Springer Berlin Heidelberg, 2004.

[GHCG11] Ritu Garg, Ehtesham Hassan, Santanu Chaudhury, and M. Gopal. A CRF based scheme for overlapping multi-colored text graphics separation. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 1215–1219. IEEE, 2011.

[Git72] Israel Gitman. A Parameter-free clustering model. *Pattern Recognition (PR)*, 4(3):307–315, 1972.

[GKNK14] Yuliang Guo, Naman Kumar, Maruthi Narayanan, and Benjamin Kimia. A multi-stage approach to curve extraction. *European Conference on Computer Vision (ECCV)*, pages 663–678, 2014.

[GNP+04] Basilis Gatos, K. Ntzios, I. Pratikakis, S. Petridis, T. Konidaris, and Stavros J. Perantonis. A segmentation-free recognition technique to assist old Greek handwritten manuscript OCR. In *Document Analysis Systems (DAS)*, pages 63–74. Springer-Verlag, 2004.

[GNP+06] Basilis Gatos, K. Ntzios, I. Pratikakis, S. Petridis, T. Konidaris, and Stavros J. Perantonis. An efficient segmentation-free approach to assist old Greek handwritten manuscript OCR. *Pattern Analysis and Applications (PAA)*, 8(4):305–320, 2006.

[GO11] Eduardo S. L. Gastal and Manuel M. Oliveira. Domain transform for edge-aware image and video processing. *ACM Transactions on Graphics*, 30(4):1, 2011.

[GSD11] Angelika Garz, Robert Sablatnig, and Markus Diem. Layout analysis for historical manuscripts using SIFT features. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 508–512. IEEE, 2011.

[GV09] Albert Gordo and Ernest Valveny. A rotation invariant page layout descriptor for document classification and retrieval. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 481–485. IEEE, 2009.

[Har51] Darell Boyd Harmon. The coordinated classroom. Technical report, American Seating Company, Santa Ana, California, 1951.

[HIH16] Keita Hirai, Daisuke Irie, and Takahiko Horiuchi. Multi-primary image projector using programmable spectral light source. *Journal of the Society of Information Display*, 24(3):144–153, 2016.

[HKP12] Marcin Heliński, Miłosz Kmieciak, and Tomasz Parkoła. Report on the comparison of Tesseract and ABBYY FineReader OCR engines. Technical report, Poznań Supercomputing and Networking Center, 2012.

[How15] Brett Howse. The Microsoft Surface Book review, 2015.

[HPN11] David Hebert, Thierry Paquet, and Stéphane Nicolas. Continuous CRF with multi-scale quantization feature functions application to structure extraction in old newspaper. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 493–497. IEEE, 2011.

[HPSO12] Azhar Hadmi, William Puech, Brahim Ait Es Said, and Abdellah Ait Ouahman. Perceptual image hashing. In Mithun Das Gupta, editor, *Watermarking - Volume 2*, chapter Perceptual, pages 17–42. InTech, 2012.

[HS88] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Alvey vision conference*, pages 147–151. Alvey, 1988.

[HS05] Dajun He and Qibin Sun. A practical print-scan resilient watermarking scheme. In *International Conference on Image Processing (ICIP)*, volume 1, pages 257–260. IEEE, 2005.

[HST13] Kaiming He, Jian Sun, and Xiaoou Tang. Guided image filtering. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 6311(6):1397 – 1409, 2013.

[IKN98] Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 20(11):1254–1259, 1998.

[IPM09] Dino Ienco, Ruggero G. Pensa, and Rosa Meo. Parameter-free hierarchical co-clustering by n -ary splits. In Wray Buntine, Marko Grobelnik, Dunja Mladenić, and John Shawe-Taylo, editors, *Machine Learning and Knowledge Discovery in Databases*, pages 580–595. Springer, 2009.

[ITU92] ITU. Recommendation T.81. Technical report, International Telecommunication Union, 1992.

[ITU15] ITU. Parameter values for the HDTV standards for production and international programme exchange. Technical report, International Telecommunication Union, 2015.

[JHM+10] Sobia T. Javed, Sarmad Hussain, Ameera Maqbool, Samia Asloob, Sehrish Jamil, and Huma Moin. Segmentation free nastalique Urdu OCR. *World Academy of Science, Engineering and Technology*, 46:456–461, 2010.

[JQZ+09] Zhen Jin, Kaiyue Qi, Yi Zhou, Kai Chen, Jianbo Chen, and Haibing Guan. SSIFT: an improved SIFT descriptor for Chinese character recognition in complex images. In *International Symposium on Computer Network and Multimedia Technology (CNMT)*, pages 1–5. IEEE, 2009.

[JRME08] Nicholas Journet, Jean-Yves Ramel, Rémy Mullot, and Véronique Eglin. Document image characterization using a multiresolution analysis of the texture: application to old documents. *International Journal on Document Analysis and Recognition (IJDAR)*, 11(1):9–18, 2008.

[KAA10] Jayant Kumar and Wael Abd-Almageed. Handwritten arabic text line segmentation using affinity propagation. In *Document Analysis Systems (DAS)*, pages 135–142. IEEE, 2010.

[KC10] Hyung Il Koo and Nam Ik Cho. State estimation in a document image and its application in text block identification and text line extraction. In *European Conference on Computer Vision (ECCV)*, volume 6312, pages 421–434. Springer, 2010.

[Ker83] Auguste Kerckhoffs. La cryptographie militaire. *Journal des sciences militaires*, IX(Janvier):5–38, 1883.

[Key81] Robert G. Keys. Cubic convolution interpolation for digital image processing. *Transactions on Acoustics, Speech and Signal Processing*, 29(6):1153–1160, 1981.

[KG13] Cameron F. Kerry and Patrick Gallagher. Digital Signature Standard (DSS), 2013.

[KHDLM10] Andrew Kae, Gary Huang, Carl Doersch, and Erik Learned-Miller. Improving state-of-the-art OCR through high-precision document specific modeling. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1935–1942. IEEE Comput. Soc. Press, 2010.

[KHP93] Tapas Kanungo, Robert M. Haralick, and Ihsin Phillips. Global and local document degradation models. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 730–734. IEEE, 1993.

[Kim99] Hae Yong Kim. Segmentation-free printed character recognition by relaxed neighbor learning of windowed operator. In *Brazilian Symposium on Computer Graphics and Image Processing*, pages 195–204. IEEE, 1999.

[Kis14] Koichi Kise. Page segmentation techniques in document analysis. In *Handbook of Document Image Processing and Recognition*, pages 135–175. Springer London, London, 2014.

[KKDAA11] Jayant Kumar, Le Kang, David Doermann, and Wael Abd-Almageed. Segmentation of handwritten textlines in presence of touching components. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 109–113. IEEE, 2011.

[KLR04] Eamonn Keogh, Stefano Lonardi, and Chotirat Ann Ratanamahatana. Towards parameter-free data mining. In *International conference on Knowledge discovery and data mining*, pages 206–215. ACM, 2004.

[KLT05] Sagi Katz, George Leifman, and Ayellet Tal. Mesh segmentation using feature point and core extraction. *The Visual Computer*, 21(8-10):649–658, sep 2005.

[KN01] C. Kailasanathan and R. Safavi Naini. Image authentication surviving acceptable modifications using statistical measures and k-mean segmentation. In *IEEE-EURASIP Work. Nonlinear Sig. and Image Processing*, pages 1–13. IEEE, 2001.

[KO11] Minwoo Kim and Il-Seok Oh. Script-free text line segmentation using interline space model for printed document images. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 1354–1358. IEEE, 2011.

[Kod03] Kodak. KODAK KAF-5101CE Image Sensor. Technical report, Kodak Image Sensor Solutions, 2003.

[Kon12] Iuliu Konya. *Adaptive methods for robust document image understanding.* PhD thesis, Rheinischen Friedrich-Wilhelms-Universität, 2012.

[Kor06] Jesse Kornblum. Identifying almost identical files using context triggered piecewise hashing. *Digital Investigation*, 3:91–97, sep 2006.

[KSI98] Koichi Kise, Akinori Sato, and Motoi Iwata. Segmentation of page images using the area voronoi diagram. *Computer Vision and Image Understanding*, 70(3):370–382, jun 1998.

[KSNO03] C. Kailasanathan, R. Safavi-Naini, and P. Ogunbona. Compression tolerant DCT based image hash. In *International Conference on Distributed Computing Systems Workshops*, pages 562–567. IEEE, 2003.

[KSS+14] Ergina Kavallieratou, Efstathios Stamatatos, Irini Stathi, Apostolos Antonacopoulos, and Josep Llados. International Document Image Processing Summer School. Summer school, 2014.

[Kue02] Rolf G. Kuehni. CIEDE2000 , milestone or final answer ? *Color Research & Application*, 27(2):126–127, 2002.

[KVBP08] Oleksiy Koval, Sviatoslav Voloshynovskiy, Fokko Beekhof, and Thierry Pun. Security analysis of robust perceptual hashing. In Edward J. Delp III, Ping Wah Wong, Jana Dittmann, and Nasir D. Memon, editors, *Security, Forensics, Steganography, and Watermarking of Multimedia Contents*, pages 1–10. SPIE, feb 2008.

[KVM04] S.S. Kozat, Ramarathnam Venkatesan, and M. Kıvanç Mıhçak. Robust perceptual image hashing via matrix invariants. *International Conference on Image Processing (ICIP)*, 5:0–3, 2004.

[KY01] E. Kasutani and A. Yamada. The MPEG-7 color layout descriptor: a compact image feature description for high-speed image/video segment retrieval. In *International Conference on Image Processing (ICIP)*, volume 1, pages 674–677. IEEE, 2001.

[KZK13] Albert Kavelar, Sebastian Zambanini, and Martin Kampel. Reading ancient coin legends: object recognition vs. OCR. In *Annual Workshop of the Austrian Association for Pattern Recognition*, pages 1–8. Austrian Association for Pattern Recognition, 2013.

[LBK+99] Zhidong Lu, Issam Bazzi, Andras András Kornai, John Makhoul, Premkumar Natarajan, and Richard Schwartz. A robust, language-independent OCR system.

In *AIPR Workshop: Advances in Computer-Assisted Recognition*, pages 96–104. SPIE, 1999.

[LC98] Ching-Yung Lin and Shih-Fu Chang. Generating robust digital signature for image / video authentication. In *Multimedia and Security Workshop at ACM Multimedia*, pages 1–6. ACM, 1998.

[LC99] Ching-yung Lin and Shih-fu Chang. Distortion modeling and invariant extraction for digital image print-and-scan process. In *International Symposium on Multimedia Information Processing*, pages 1–10. IEEE, 1999.

[LC01] Ching-Yung Lin and Shih-Fu Chang. A robust image authentication method distinguishing JPEG compression from malicious manipulation. *IEEE Transactions on Circuits and Systems For Video Technology*, 11(2):153–168, 2001.

[LCC08a] Aurélie Lemaitre, Jean Camillerapp, and Bertrand Coüasnon. A generic method for structure recognition of handwritten mail documents. In *Document Recognition and Retrieval (DRR)*, page 68150W. SPIE, 2008.

[LCC08b] Aurélie Lemaitre, Jean Camillerapp, and Bertrand Coüasnon. Multiresolution cooperation makes easier document structure recognition. *International Journal on Document Analysis and Recognition (IJDAR)*, 11(2):97–109, 2008.

[LCC11] Aurélie Lemaitre, Jean Camillerapp, and Bertrand Coüasnon. A perceptive method for handwritten text segmentation. In *Document Recognition and Retrieval (DRR)*, pages 78740C–78740C–9. SPIE, 2011.

[LCM03] Frédéric Lefèbvre, Jacek Czyz, and Benoit Macq. A robust soft hash algorithm for digital image signature. In *International Conference on Image Processing (ICIP)*, pages 495–498. IEEE, 2003.

[LCOB15] Do Thi Luyen, Elodie Carel, Jean-Marc Ogier, and Jean-Christophe Burie. A character degradation model for color document images. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 806–810. IEEE, 2015.

[LCS11] Stefan Leutenegger, Margarita Chli, and Roland Y. Siegwart. BRISK : Binary Robust Invariant Scalable Keypoints. In *International Conference on Computer Vision (ICCV)*, pages 2548–2555. IEEE, 2011.

[LDMG02] Jian Liang, David Doermann, Matthew Ma, and Jinhong K. Guo. Page classification through logical labelling. In *International Conference on Pattern Recognition (ICPR)*, volume 3, pages 477–480. IEEE, 2002.

[LFB16] Imanol Luengo, Andrew P. French, and Mark Basham. Hierarchical piecewise-constant super-regions. *Computing Research Repository (CoRR)*, abs/1605.0:1–9, 2016.

[LFJ08] Xiabi Liu, Hui Fu, and Yunde Jia. Gaussian mixture modeling and learning of neighboring characters for multilingual text extraction in images. *Pattern Recognition (PR)*, 41(2):484–493, 2008.

[LGPH09] G. Louloudis, Basilis Gatos, I. Pratikakis, and C. Halatsis. Text line and word segmentation of handwritten documents. *Pattern Recognition (PR)*, 42(12):3169–3183, 2009.

[LH88] Margaret Livingstone and David Hubel. Segregation of form, color, movement, and depth : anatomy, physiology, and perception. *Science*, 240(May):740–750, 1988.

[LH05] Chun-Shien Lu and Chao-Yong Hsu. Geometric distortion-resilient image hashing scheme and its applications on copy detection and authentication. *Multimedia Systems*, 11(2):159–173, 2005.

[LHSC04] Chun-Shien Lu, Chao Yong Hsu, Shih-Wei Sun, and Pao-chi Chang. Robust mesh-based hashing for copy detection and tracing of images. In *International Conference on Multimedia and Expo*, pages 713–734. IEEE, 2004.

[Liu12] Frank Liu. Fast and low-power OCR for the blind. Technical report, Stanford University, 2012.

[LL92] G. Leach and G. Leach. Improving worst-case optimal Delaunay triangulation algorithms. In *Canadian Conference on Computational Geometry*, pages 340–346, 1992.

[LL03] Chun-shien Lu and Hong-Yuan Mark Liao. Structural digital signature for image authentication : an incidental distortion resistant scheme. *IEEE Transactions on Multimedia*, 5(2):161–173, 2003.

[LLG+11] Guillaume Lazzara, Roland Levillain, Thierry Géraud, Yann Jacquelet, Julien Marquegnies, and Arthur Crépin-Leblond. The SCRIBO module of the Olena platform : a free software framework for document image analysis. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 252–258. IEEE, 2011.

[Llo82] Stuart P. Lloyd. Least squares quantization in PCM. *Transactions on Information Theory*, 28(2):129–137, 1982.

[LLS11] Bart Lamiroy, Daniel Lopresti, and Tao Sun. Document analysis algorithm contributions in end-to-end applications: report on the ICDAR 2011 contest. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 1521–1525. IEEE, 2011.

[LLS14a] Li Liu, Yue Lu, and Ching Y Suen. Near-duplicate document image matching: A graphical perspective. *Pattern Recognition (PR)*, 47(4):1653–1663, 2014.

[LLS14b] Li Liu, Yue Lu, and Ching Y Suen. Novel global and local features for near-duplicate document image matching. In *International Conference on Pattern Recognition (ICPR)*, pages 4624–4629. IEEE, aug 2014.

[LML02] Frédéric Lefèbvre, Benoit Macq, and Jean-Didier Legat. RASH : RAdon Soft Hash algorithm. In *European Signal Processing Conference (EUSIPCO)*, pages 1–4. IEEE Sign. Proc. Soc. Press, 2002.

[LNV+15] Viet Phuong Le, Nibal Nayef, Muriel Visani, Jean-Marc Ogier, and Cao De Tran. Text and non-text segmentation based on connected component features. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 1096–1100. IEEE, 2015.

[Lop09] Daniel Lopresti. Optical character recognition errors and their effects on natural language processing. *International Journal on Document Analysis and Recognition (IJDAR)*, 12(3):141–151, 2009.

[Low04] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (IJCV)*, 60(2):91–110, nov 2004.

[LPN08] Joao Manuel Maciel Linhares, Paulo Daniel Pinto, and Sergio Miguel Cardoso Nascimento. The number of discernible colors in natural scenes. *Journal of the Optical Society of America A*, 25(12):2918–2924, 2008.

[LSK+09] Alex Levinshtein, Adrian Stere, Kiriakos N. Kutulakos, David J. Fleet, and Sven J. Dickinson. TurboPixels : fast superpixels using geometric flows. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 31(12):2290 – 2297, 2009.

[LSS08] Laurence Likforman-Sulem and Marc Sigelle. Recognition of degraded characters using dynamic Bayesian networks. *Pattern Recognition (PR)*, 41(10):3092–3103, 2008.

[LTRC11] M.-Y. Liu, O. Tuzel, S. Ramalingam, and R. Chellappa. Entropy rate superpixel segmentation. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2097 – 2104. IEEE, 2011.

[LU08] Gerold Laimer and Andreas Uhl. Key-dependent JPEG2000-based robust hashing for secure image authentication. *EURASIP Journal on Information Security*, 2008:1–19, 2008.

[LWH11] Yanqiang Lei, Yuangen Wang, and Jiwu Huang. Robust image hash in Radon transform domain for authentication. *Signal Processing: Image Communication*, 26(6):280–288, 2011.

[LZY10] Zongyi Liu, Hanning Zhou, and Ning Yang. Semi-supervised learning for text-line detection. *Pattern Recognition Letters (PRL)*, 31(11):1260–1273, 2010.

[Mal13] Alberto Malvido Garcià. Secure Imprint Generated for Paper Documents (SIGNED). Technical Report December 2010, Bit Oceans, 2013.

[MBE03] Vishal Monga, Arindam Banerjee, and Brian L. Evans. A clustering based approach to perceptual image hashing. *IEEE Transactions on Information Forensics and Security*, 1(1):68–79, 2003.

[ME06] Vishal Monga and Brian L. Evans. Perceptual image hashing via feature points : performance evaluation and trade-offs. *Transactions on Image Processing (TIP)*, 15(11):3452–3465, 2006.

[MEE+09] Vincent Malleron, Véronique Eglin, Hubert Emptoz, Stéphanie Dord-Crouslé, and Philippe Régnier. Text lines and snippets extraction for 19th century handwriting documents layout analysis. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 1001–1005. IEEE, 2009.

[MF06] Alberto Malvido and P. Fernando. A novel model for the print-and-capture channel in 2D bar codes. In *International Workshop on Multimedia Content Representation, Classification and Security*, pages 627–634. Springer, 2006.

[MFM04] David R. Martin, Charless C. Fowlkes, and Jitendra Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 26(5):530–549, 2004.

[MGHN09] Florent Montreuil, Emmanuèle Grosicki, Laurent Heutte, and Stéphane Nicolas. Unconstrained handwritten document layout extraction using 2D conditional random fields. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 853–857. IEEE, 2009.

[MGKH+13] Maroua Mehri, Petra Gomez-Krämer, Pierre Héroux, Alain Boucher, and Rémy Mullot. Texture feature evaluation for segmentation of historical document images. In *International Workshop on Historical Document Imaging and Processing (HIP)*, page 102, New York, New York, USA, 2013. ACM Press.

[MHGK+13] Maroua Mehri, Pierre Heroux, Petra Gomez-Krämer, Alain Boucher, and Remy Mullot. A pixel labeling approach for historical digitized books. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 817–821. IEEE, aug 2013.

[MIO+04] Yuri Murakami, Jun-Ichiro Ishii, Takashi Obi, Masahiro Yamaguchi, and Nagaaki Ohyama. Color conversion method for multi-primary display for spectral color reproduction. *Journal of Electronic Imaging*, 13(4):701–708, 2004.

[MKW15] Bastien Moysset, Christopher Kermorvant, and Christian Wolf. Paragraph text segmentation into lines with recurrent neural networks. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 456–460. IEEE, 2015.

[ML09] Marius Muja and David G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *International Conference on Computer Vision Theory and Applications*, pages 331–340. Science and Technology Publications, Lda, 2009.

[MM07] Vishal Monga and M. Kivanç Mihçak. Robust and secure image hashing via non-negative matrix factorizations. *IEEE Transactions on Information Forensics and Security*, 2(3):376–390, 2007.

[MNM13] Lazaros Mavridis, Neetika Nath, and John B. O. Mitchell. PFClust : a novel parameter free clustering algorithm. *BMC Bioinformatics*, 14(213):1–20, 2013.

[MPW+08] Alastair P. Moore, Simon J. D. Prince, Jonathan Warrell, Umar Mohammed, and Graham Jones. Superpixel lattices. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2008.

[MRHR15] Michael Murdock, Shawn Reid, Blaine Hamilton, and Jackson Reese. ICDAR 2015 Competition on text line detection in historical documents. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 1171–1175. IEEE, 2015.

[MRK03] Song Mao, Azriel Rosenfeld, and Tapas Kanungo. Document structure analysis algorithms: a literature survey. In Tapas Kanungo, Elisa H. Barney Smith, Jianying Hu, and Paul B. Kantor, editors, *Document Recognition and Retrieval (DRR)*, pages 197–207. Elsevier, jan 2003.

[MSM09] David Miller, Paulo Schor, and Peter Magnante. Optics of the normal eye. In *Ophthalmology*, chapter 2.7, pages 52–60. Elsevier, 3rd edition, 2009.

[MV02] M. Kıvanç Mıhçak and Ramarathnam Venkatesan. New iterative geometric methods for robust perceptual image hashing. In *Security and Privacy in Digital Rights Management*, pages 13–21. Springer Berlin Heidelberg, 2002.

[MVE05] Vishal Monga, Divyanshu Vats, and Brian L. EvanS. Image authentication under geometric attacks via structure matching. In *International Conference on Multimedia and Expo*, pages 229–232. IEEE, 2005.

[Nag00] George Nagy. Twenty years of document image analysis in PAMI. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 22(1):38–62, 2000.

[NGP+07] K. Ntzios, Basilis Gatos, I. Pratikakis, T. Konidaris, and Stavros J. Perantonis. An old greek handwritten OCR system based on an efficient segmentation-free approach. *International Journal on Document Analysis and Recognition (IJDAR)*, 9(2):179–192, 2007.

[Nik10] Kai Niklas. *Unsupervised post-correction of OCR Errors*. PhD thesis, Leibniz Universität Hannover, 2010.

[NJ07] Anoop M. Namboodiri and Anil K. Jain. Document structure and layout analysis. In *Digital Document Processing*, pages 29–48. Springer London, 2007.

[NKI06] Tomohiro Nakai, Koichi Kise, and Masakazu Iwamura. Use of affine invariants in locally likely arrangement hashing for camera-based document image retrieval. *Lecture Notes in Computer Science (LNCS)*, 3872:541–552, 2006.

[NM79] Makoto Nagao and Takashi Matsuyama. Edge preserving smoothing. *Computer Graphics and Image Processing*, 9(4):394–407, 1979.

[NP09] Nikos Nikolaou and Nikos Papamarkos. Color reduction for complex document images. *International Journal of Imaging Systems and Technology*, 19(1):14–26, 2009.

[NP14] Peer Neubert and Peter Protzel. Compact Watershed and preemptive SLIC : on improving trade-offs of superpixel segmentation algorithms. In *International Conference on Pattern Recognition (ICPR)*, pages 996–1001. IEEE, 2014.

[NSB85] R. Navarro, J. Santamaria, and J. Bescos. Accommodation-dependent model of the human eye with aspherics. *Journal of the Optical Society of America A*, 2(8):1273–1281, 1985.

[NSV92] George Nagy, Sharad Seth, and Mahesh Viswanathan. A prototype document image analysis system for technical journals. *Computer*, 25(7):10–22, jul 1992.

[OB08] Nazih Ouwayed and Abdel Belaïd. Multi-oriented text line extraction from handwritten arabic documents. In *Document Analysis Systems (DAS)*, pages 339–346. IEEE, sep 2008.

[OB12] Nazih Ouwayed and Abdel Belaïd. A general approach for multi-oriented text line extraction of handwritten documents. *International Journal on Document Analysis and Recognition (IJDAR)*, 15(4):297–314, 2012.

[OBA10] Nazih Ouwayed, Abdel Belaïd, and François Auger. General text line extraction approach based on locally orientation estimation. In *Document Recognition and Retrieval (DRR)*, pages 75340B–75340B–8. SPIE, 2010.

[OLL13] Asma Ouji, Yann Leydier, and Frank LeBourgeois. A hierarchical and scalable model for contemporary document image segmentation. *Pattern Analysis and Applications (PAA)*, 16(4):679–693, nov 2013.

[OLT10] Daniel M. Oliveira, Rafael D. Lins, and Gabriel Torreão. A new method for text-line segmentation for warped documents. In Aurélio Campilho and Mohamed Kamel, editors, *International Conference on Image Analysis and Recognition (ICIAR)*, Lecture Notes in Computer Science, pages 398–408, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[OLT+13] Daniel M. Oliveira, Rafael D. Lins, Gabriel Torreão, Jian Fan, and Marcelo Thielo. An efficient algorithm for segmenting warped text-lines in document images. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 250–254. IEEE, aug 2013.

[OR09] Yang Ou and Kyung Hyune Rhee. A key-dependent secure image hashing scheme by using radon transform. In IEEE, editor, *International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*, pages 595–598, 2009.

[Ots79] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *Systems, Man, and Cybernetics*, 9(1):62–66, 1979.

[PB11] Samuel J. Pinson and William a. Barrett. Connected component level discrimination of handwritten and machine-printed text using eigenfaces. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 1394–1398. IEEE, 2011.

[PGCW01] Jason Porter, Antonio Guirao, Ian G Cox, and David R Williams. Monochromatic aberrations of the human eye in a large population. *Journal of the Optical Society of America A*, 18(8):1793–1803, 2001.

[PLL99] J.M. Peña, J.A. Lozano, and P. Larrañaga. An empirical comparison of four initialization methods for the K-Means algorithm. *Pattern Recognition Letters (PRL)*, 20(10):1027–1040, 1999.

[PM90] Pietro Perona and Jitendra Malik. Scale-space and edge detection using anisotropic diffusion. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 12(7):629–639, 1990.

[PN95] Thrasyvoulos N. Pappas and David L. Neuhoff. Printer models and error diffusion. *IEEE Transactions on Image Processing*, 4(1):66–80, 1995.

[Pow07] David M. W. Powers. Evaluation: from precision, recall and F-factor to ROC, informedness, markedness & correlation. Technical Report December, School of Informatics and Engineering, Flinders University, Adelaide, Australia, 2007.

[PPTL10] Umapada Pal, Partha Pratim Roy, Nilamadhaba Tripathy, and Josep Lladós. Multi-oriented Bangla and Devnagari text recognition. *Pattern Recognition (PR)*, 43(12):4124–4136, 2010.

[PSG+09] Xujun Peng, Srirangaraj Setlur, Venu Govindaraju, Ramachandrula Sitaram, and Kiran Bhuvanagiri. Markov random field based text identification from annotated machine printed documents. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 431–435. IEEE, 2009.

[PSGR12] Xujun Peng, Srirangaraj Setlur, Venu Govindaraju, and Sitaram Ramachandrula. Using a boosted tree classifier for text segmentation in hand-annotated documents. *Pattern Recognition Letters (PRL)*, 33(7):943–950, may 2012.

[PSGS13] Xujun Peng, Srirangaraj Setlur, Venu Govindaraju, and Ramachandrula Sitaram. Handwritten text separation from annotated machine printed documents using Markov random fields. *International Journal on Document Analysis and Recognition (IJDAR)*, 16(1):1–16, mar 2013.

[PSKC10] Vassilis Papavassiliou, Themos Stafylakis, Vassilis Katsouros, and George Carayannis. Handwritten document image segmentation into text lines and words. *Pattern Recognition (PR)*, 43(1):369–377, 2010.

[PSYL15] Jianteng Peng, Jianbing Shen, Angela Yao, and Xuelong Li. Superpixel optimization using higher-order energy. *IEEE Transactions on Circuits and Systems for Video Technology*, 26(5):917 – 927, 2015.

[PWK12] Jialin Peng, Jinwei Wang, and Dexing Kong. A new convex variational model for liver segmentation. In *International Conference on Pattern Recognition (ICPR)*, pages 3754–3757. IEEE, 2012.

[PZZL14] Xiao Pan, Yuanfeng Zhou, Caiming Zhang, and Qian Liu. Flooding based superpixel generation with color, compactness and smoothness constraints. In *International Conference on Image Processing (ICIP)*, pages 4432–4436. IEEE, 2014.

[Que98] Maria Paula Queluz. Towards robust, content based techniques. In *Workshop on Multimedia Signal Processing*, pages 297–302. IEEE, 1998.

[RD06] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *European Conference on Computer Vision (ECCV)*, pages 430–443. Springer, 2006.

[Rey08] Martin Reynaert. Non-interactive OCR post-correction for giga-scale digitization projects. *Computational Linguistics and Intelligent Text Processing*, 4919:617–630, 2008.

[RFP+12] Ana Rebelo, Ichiro Fujinaga, Filipe Paszkiewicz, Andre R. S. Marcal, Carlos Guedes, and Jaime S. Cardoso. Optical music recognition: state-of-the-art and open issues. *International Journal of Multimedia Information Retrieval*, 1(3):173–190, 2012.

[RH75] W. S. Rosenbaum and J. J. Hilliard. Multifont OCR postprocessing system. *IBM Journal of Research and Development*, 19(4):398–421, jul 1975.

[Riv92] R Rivest. The MD5 message-digest algorithm. Technical report, Internet activities board, 1992.

[RJN96] Stephen Rice, Frank Jenkins, and Thomas Nartker. The fifth annual test of OCR accuracy. Technical Report April, Information Science Research Institute, 1996.

[RKC14] Jewoong Ryu, Hyung Il Koo, and Nam Ik Cho. Language-independent text-line extraction algorithm for handwritten documents. *Signal Processing Letters*, 21(9):1115–1119, 2014.

[Ros61] Frank Rosenblatt. Perceptrons and the theory of brain mechanisms. Technical report, Cornell Aeronautical Laboratory Inc, Buffalo N.Y., 1961.

[RPL12] Partha Pratim Roy, Umapada Pal, and Josep Lladós. Text line extraction in graphical documents using background and foreground information. *International Journal on Document Analysis and Recognition (IJDAR)*, 15(3):227–241, 2012.

[RRKB11] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB : an efficient alternative to SIFT or SURF. In *International Conference on Computer Vision (ICCV)*, pages 2564–2571. IEEE, 2011.

[RTBO13] Christophe Rigaud, Norbert Tsopze, Jean-Christophe Burie, and Jean-Marc Ogier. Robust frame and text extraction from comic books. In *Graphics Recognition. New Trends and Challenges*, volume 7423 LNCS, pages 129–138. Springer, 2013.

[SBKB08] Faisal Shafait, J. Beusekom, Daniel Keysers, and Thomas M. Breuel. Structural mixtures for statistical layout analysis. In *Document Analysis Systems (DAS)*, pages 415–422. IEEE, 2008.

[SC96] Marc Schneider and Shih-Fu Chang. A robust content based digital signature for image authentication. *International Conference on Image Processing (ICIP)*, 3:227–230, 1996.

[SCE+15] Mathias Seuret, Kai Chen, Nicole Eichenberger, Marcus Liwicki, and Rolf Ingold. Gradient-domain degradations for improving historical documents images layout analysis. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 1006–1010. IEEE, 2015.

[SCK15] Sandeepika Sharma, Sneha Choudhray, and Bhupendra Kumar. Recognition of machine printed broken Oriya characters using SIFT features. In *International Conference on Computer and Communication Technology*, pages 106–109. ACM, 2015.

[Sey13] John Seymour. Why does my cyan have the blues? *FlexoGlobal eZine*, pages 1–5, 2013.

[SFS12] Alexander Schick, Mika Fischer, and Rainer Stiefelhagen. Measuring and evaluating the compactness of superpixels. In *International Conference on Pattern Recognition (ICPR)*, pages 930–934. IEEE, 2012.

[SGL+13] Nikolaos Stamatopoulos, Basilis Gatos, Georgios Louloudis, Umapada Pal, and Alireza Alaei. ICDAR 2013 Handwriting segmentation contest. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 1402–1406. IEEE, aug 2013.

[SGP09] Nikolaos Stamatopoulos, Basilis Gatos, and Stavros J. Perantonis. A method for combining complementary techniques for document image segmentation. *Pattern Recognition (PR)*, 42(12):3158–3168, 2009.

[Sha02] Gaurav Sharma. Color fundamentals for digital imaging. In *Digital Color Imaging Handbook*, chapter 1, page 816. CRC Press, 2002.

[SHKY03] Jin S. Seo, Jaap Haitsma, Ton Kalker, and Chang D. Yoo. Affine transform resilient image fingerprinting. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 61–64. IEEE, 2003.

[SHKY04] Jin S. Seo, Jaap Haitsma, Ton Kalker, and Chang D. Yoo. A robust image fingerprinting system using the Radon transform. *Signal Processing: Image Communication*, 19(4):325–339, 2004.

[SK07] Diego Massola Shimizu and Hae Yong Kim. Perceptual hashing for hardcopy document authentication using morphological segmentation. In *International Symposium on Mathematical Morphology*, pages 77–78. IEEE, 2007.

[SKB08] Faisal Shafait, Daniel Keysers, and Thomas M. Breuel. Performance evaluation and benchmarking of six-page segmentation algorithms. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 30(6):941–54, jun 2008.

[SLDZ09] Xiaolu Shen, Changsong Liu, Xiaoqing Ding, and Yanming Zou. Text line extraction in free-style document. In *Document Recognition and Retrieval (DRR)*, pages 72470L–72470L–12. SPIE, 2009.

[SLG15] Nikolaos Stamatopoulos, Georgios Louloudis, and Basilis Gatos. Goal-oriented performance evaluation methodology for page segmentation techniques. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 281–285. IEEE, 2015.

[Smi02] Andrew Smith. Identity fraud: a study. Technical Report July, Economic and Domestic Secretariat Cabinet Office, 2002.

[Smi07] Raymond W. Smith. An overview of the Tesseract OCR engine. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 629–633. IEEE, 2007.

[Smi09] Raymond W. Smith. Hybrid page layout analysis via tab-stop detection. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 241–245. IEEE, 2009.

[Smo12] Andreea Smoaca. *ID photograph hashing: a global approach*. PhD thesis, Université Jean Monnet, 2012.

[SMW06] Ashwin Swaminathan, Yinian Mao, and Min Wu. Robust and secure image hashing. *IEEE Transactions on Information Forensics and Security*, 1(2):215–230, 2006.

[Son15] Raymond M. Soneira. Surface Pro display technology shoot-out: Microsoft Surface Pro 4, 2015.

[SRCEB05] Theresa J. Squire, Marisa Rodriguez-Carmona, Anthony D. B. Evans, and John L. Barbur. Color vision tests for aviation: comparison of the anomaloscope and three lantern types. *Aviation Space and Environmental Medicine*, 76(5):421–429, 2005.

[SS00] Andrew Stockman and Lindsay T. Sharpe. The spectral sensitivities of the middle- and long-wavelength-sensitive cones derived from measurements in observers of known genotype. *Vision Research*, 40(1):1711–1737, 2000.

[SS13] Nazly Sabbour and Faisal Shafait. A segmentation-free approach to Arabic and Urdu OCR. In *Document Recognition and Retrieval (DRR)*, page 86580N. SPIE, 2013.

[SSG09] Zhixin Shi, Srirangaraj Setlur, and Venu Govindaraju. A steerable directional local profile technique for extraction of handwritten arabic text lines. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 176–180. IEEE, 2009.

[SSL09] Prateek Sarkar, Eric Saund, and Jing Lin. Classifying foreground pixels in document images. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 641–645. IEEE, 2009.

[ST94] Jianbo Shi and Carlo Tomasi. Good features to track. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 593–600. IEEE, 1994.

[STC05] Qibin Sun, Qi Tian, and Shih-Fu Chang. A robust and secure media signature scheme for JPEG images. *Journal of VLSI signal processing systems for signal, image and video technology*, 41(3):305–317, 2005.

[STF+03] Masakazu Suzuki, Fumikazu Tamari, Ryoji Fukuda, Seiichi Uchida, and Toshihiro Kanahori. INFTY — an integrated OCR system for mathematical documents. In *Symposium on Document engineering (DocEng)*, pages 95–104. ACM Press, 2003.

[Str90] Walahfrid Strabo. *Cod. Sang. 562: Vitae sancti Galli et Otmari*. St. Gallen Stiftsbibliothek, St. Gallen, 890.

[STRV15] Joan Andreu Sanchez, Alejandro H. Toselli, Veronica Romero, and Enrique Vidal. ICDAR 2015 Competition HTRtS : Handwritten Text Recognition on the tranScriptorium dataset. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 1166–1170. IEEE, 2015.

[Stu14] David Stutz. Superpixel segmentation using depth information. Technical report, RWTH Aachen University, Aachen, 2014.

[SvBKB08] Faisal Shafait, Joost van Beusekom, Daniel Keysers, and Thomas M. Breuel. Background variability modeling for statistical layout analysis. In *International Conference on Pattern Recognition (ICPR)*, pages 1–4. IEEE, 2008.

[SWD05] Gaurav Sharma, Wencheng Wu, and Edul N. Dalal. The CIEDE2000 color-difference formula: Implementation notes, supplementary test data, and mathematical observations. *Color Research & Application*, 30(1):21–30, feb 2005.

[SZ06] Martin Schmucker and Hui Zhang. Benchmarking metrics and concepts for perceptual hashing. Technical report, Fraunhofer Gesellschaft e.V., 2006.

[SZGH09] Tong Hua Su, Tian Wen Zhang, De Jun Guan, and Hu Jie Huang. Off-line recognition of realistic Chinese handwriting using segmentation-free strategy. *Pattern Recognition (PR)*, 42(1):167–182, 2009.

[TAA05] Sofien Touj, Najoua Ben Amara, and Hamid Amiri. Generalized Hough transform for Arabic printed optical character recognition. *The International Arab Journal of Information Technology*, 2(4):326–333, 2005.

[TG80] Anne M. Treisman and Garry Gelade. A feature-integration theory of attention. *Cognitive Psychology*, 12(1):97–136, jan 1980.

[TKI11] Kazutaka Takeda, Koichi Kise, and Masakazu Iwamura. Real-time document image retrieval for a 10 million pages database with a memory efficient and stability improved LLAH. In *International Workshop on Camera-Based Document Analysis and Recognition (CBDAR)*, pages 59–64. IEEE, 2011.

[TM98] Carlo Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *International Conference on Computer Vision (ICCV)*, pages 839–846. IEEE, 1998.

[TPSD16] Iuliia Tkachenko, William Puech, Olivier Strauss, and Christophe Destruel. Experimental study of print-and-scan impact as random process. *Electronic Imaging*, 2016(2):1–11, 2016.

[TSZZ11] Lina Tan, Xingming Sun, Zhili Zhou, and Wei Zhang. Perceptual text image hashing based on shape recognition. *Advances in Information Sciences and Service Sciences (AISS)*, 3(8):1–7, 2011.

[TWB14] Youbao Tang, Xiangqian Wu, and Wei Bu. Text line segmentation based on matched filtering and top-down grouping for handwritten documents. In *Document Analysis Systems (DAS)*, pages 365–369. IEEE, apr 2014.

[UB87] Keiji Uchikawa and Robert M. Boynton. Categorical color perception of Japanese observers: comparison with that of Americans. *Vision Research*, 27(10):1825–1833, 1987.

[VBM10] Olga Veksler, Yuri Boykov, and Paria Mehrani. Superpixels and supervoxels in an energy optimization framework. In *European Conference on Computer Vision (ECCV)*, pages 211–224. Springer, 2010.

[VBRV12] Michael Van, Xavier Boix, Gemma Roig, and Luc Van Gool. SEEDS : Superpixels Extracted via Energy-Driven Sampling. In *European Conference on Computer Vision (ECCV)*, pages 13–26. Springer, 2012.

[vdBSK08] E.L. van den Broek, Th.E. Schouten, and P.M.F. Kisters. Modeling human color categorization. *Pattern Recognition Letters (PRL)*, 29(8):1136–1144, 2008.

[VKBP09] Sviatoslav Voloshynovskiy, Oleksiy Koval, Fokko Beekhof, and Thierry Pun. Conception and limits of robust perceptual hashing: towards side information assisted hash functions. In Edward J. Delp III, Jana Dittmann, Nasir D. Memon, and Ping Wah Wong, editors, *Media Forensics and Security*, page 72540D. SPIE, feb 2009.

[VKJM00] Ramarathnam Venkatesan, S.-M. Koon, M.H. Jakubowski, and P. Moulin. Robust image hashing. In *International Conference on Image Processing (ICIP)*, pages 664–666. IEEE, 2000.

[VS08] Andrea Vedaldi and Stefano Soatto. Quick Shift and kernel methods for mode seeking. In *European Conference on Computer Vision (ECCV)*, pages 705–718. Springer, 2008.

[VS12] Sreenath Rao Vantaram and Eli Saber. Survey of contemporary trends in color image segmentation. *Journal of Electronic Imaging*, 21(4):040901–1, oct 2012.

[VT86] D. Van Norren and L. F. Tiemeijer. Spectral reflectance of the human eye. *Vision Research*, 26(2):313–320, 1986.

[VVK+07] R. Villán, Sviatoslav Voloshynovskiy, Oleksiy Koval, F. Deguillaume, and T. Pun. Tamper-proofing of electronic and printed text documents via robust hashing and data-hiding. In Edward J. Delp III and Ping Wah Wong, editors, *Security, Steganography, and Watermarking of Multimedia Contents*, pages 65051T–65051T–12. SPIE, feb 2007.

[WAS11] Amy Winder, Tim Andersen, and Elisa H. Barney Smith. Extending page segmentation algorithms for mixed-layout document processing. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 1245–1249. IEEE, sep 2011.

[WBA09] Sui-Yu Wang, Henry Baird, and Chang An. Document content extraction using automatically discovered features. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 1076–1080. IEEE, 2009.

[WBSI13] Hao Wei, Micheal Baechler, Fouad Slimane, and Rolf Ingold. Evaluation of SVM, MLP and GMM classifiers for layout analysis of historical documents. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 1220–1224. IEEE, aug 2013.

[WCW82] Kwan Y. Wong, Richard G. Casey, and Friedrich M. Wahl. Document analysis system. *IBM Journal of Research and Development*, 26(6):647–656, 1982.

[WFSN15] Liuan Wang, Wei Fan, Jun Sun, and Satshi Naoi. Text line extraction in document images. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 191–195. IEEE, 2015.

[WGFJ98] H. Wang, F. Guo, D. D. Feng, and J. S. Jin. A signature for content-based image retrieval using a geometrical transform. In *International Conference on Multimedia*, pages 229–234. ACM, 1998.

[WSH15] Martijn Withouck, Kevin A. G. Smet, and Peter Hanselaer. Brightness prediction of different sized unrelated self-luminous stimuli. *Optics Express*, 23(10):13455–13466, 2015.

[WSY13] Christian Winter, Markus Schneider, and York Yannikos. F2S2: Fast forensic similarity search through indexing piecewise hash signatures. *Digital Investigation*, XXX:1–11, sep 2013.

[Wu91] Xiaolin Wu. Efficient statistical computation for optimal color quantization. In Andrew S. Glassner and James Arvo, editors, *Graphics Gems II*, pages 126–134. Academic Press Inc., 1991.

[WZN09] Di Wu, Xuebing Zhou, and Xiamu Niu. A novel image hash algorithm resistant to print-scan. *Signal Processing*, 89(12):2415–2424, 2009.

[WZT15] Yongtao Wang, Yafeng Zhou, and Zhi Tang. Comic frame extraction via line segments combination. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 856–860. IEEE, 2015.

[XKH07] Shijun Xiang, Hyoung-Joong Kim, and Jiwu Huang. Histogram-based image hashing scheme robust against geometric deformations. *Workshop on Multimedia & security*, page 121, 2007.

[XLL+11] Kaida Xiao, M. Ronnier Luo, Changjun Li, Guihua Cui, and Dusik Park. Investigation of colour size effect for colour appearance assessment. *Color Research & Application*, 36(3):201–209, 2011.

[YC05] Shih-Hsuan Yang and Chin-Feng Chen. Robust image hashing based on SPIHT. In *International Conference on Information Technology: Research and Education*, pages 110–114. IEEE, 2005.

[YGN06] Bian Yang, Fan Gu, and Xiamu Niu. Block mean value based image perceptual hashing for content identification. In *International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pages 167–172. IEEE, 2006.

[YL09a] Fei Yin and Cheng-Lin Liu. A variational bayes method for handwritten text line segmentation. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 436–440. IEEE, 2009.

[YL09b] Fei Yin and Cheng-Lin Liu. Handwritten Chinese text line segmentation by clustering with distance metric learning. *Pattern Recognition (PR)*, 42(12):3146–3157, 2009.

[YM08] Takuma Yamaguchi and Minoru Maruyama. Feature extraction for document image segmentation by pLSA model. In *Document Analysis Systems (DAS)*, pages 53–60. IEEE, 2008.

[YNS05] Longjiang Yu, Xiamu Niu, and Shenghe Sun. Print-and-scan model and the watermarking countermeasure. *Image and Vision Computing*, 23(9):807–814, 2005.

[YS07] Longjiang Yu and Shenghe Sun. Image authentication in print-and-scan scenario. In *International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pages 295–298. IEEE, 2007.

[YSBS05] Longjiang Yu, Martin Schmucker, Christoph Busch, and Shenghe Sun. Cumulant-based image fingerprints. In *Security, Steganography, and Watermarking of Multimedia Contents*, pages 68–75. SPIE, 2005.

[YTO+02] Masahiro Yamaguchi, Taishi Teraji, Kenro Ohsawa, Toshio Uchiyama, Hideto Motomura, Yuri Murakami, and Nagaaki Ohyama. Color image reproduction based on the multispectral and multiprimary imaging : Experimental evaluation. In *Color Imaging: Device Independent Color, Color Hardcopy and Applications*, pages 15–26. SPIE, 2002.

[Zau10] Christoph Zauner. *Implementation and benchmarking of perceptual image hash functions*. PhD thesis, University of Applied Sciences Hagenberg, 2010.

[ZC15] Guoqiang Zhong and Mohamed Cheriet. Tensor representation learning based image patch analysis for text identification and recognition. *Pattern Recognition (PR)*, 48(4):1211–1224, apr 2015.

[ZENM13] F. Zirari, A. Ennaji, Stéphane Nicolas, and D. Mammass. A document image segmentation system using analysis of connected components. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 753–757. IEEE, aug 2013.

[ZF10] Majid Ziaratban and Karim Faez. An adaptive script-independent block-based text line extraction. In *International Conference on Pattern Recognition (ICPR)*, pages 249–252. IEEE, 2010.

[ZHKY10] Guopu Zhu, Jiwu Huang, Sam Kwong, and Jianquan Yang. Fragility analysis of adaptive quantization based image hashing. *IEEE Transactions on Information Forensics and Security*, 5(1):133–147, 2010.

[ZHMB11] Yuhang Zhang, Richard Hartley, John Mashford, and Stewart Burn. Superpixels via pseudo-boolean optimization. In *International Conference on Computer Vision (ICCV)*, pages 1387 – 1394. IEEE, 2011.

[ZKPF99] Barbara Zitová, Jaroslav Kautsky, Gabriele Peters, and Jan Flusser. Robust detection of significant points in multiframe images. *Pattern Recognition Letters (PRL)*, 20:199–206, 1999.

[ZMCL16] Hongyuan Zhu, Fanman Meng, Jianfei Cai, and Shijian Lu. Beyond pixels : a comprehensive survey from bottom-up to semantic image segmentation and cosegmentation. *Journal of Visual Communication and Image Representation*, 34(1):12–27, 2016.

De la stabilité des algorithmes d'analyse de documents. Application aux technologies de hachage de documents hybrides.

Résumé : Un nombre incalculable de documents est imprimé, numérisé, faxé, photographié chaque jour. Ces documents sont hybrides : ils existent sous forme papier et numérique. De plus les documents numériques peuvent être consultés et modifiés simultanément dans de nombreux endroits. Avec la disponibilité des logiciels d'édition d'image, il est devenu très facile de modifier ou de falsifier un document. Cela crée un besoin croissant pour un système d'authentification capable de traiter ces documents hybrides.
Les solutions actuelles reposent sur des processus d'authentification séparés pour les documents papiers et numériques. D'autres solutions reposent sur une vérification visuelle et offrent seulement une sécurité partielle. Dans d'autres cas elles nécessitent que les documents sensibles soient stockés à l'extérieur des locaux de l'entreprise et un accès au réseau au moment de la vérification.
Afin de surmonter tous ces problèmes, nous proposons de créer un algorithme de hachage sémantique pour les images de documents. Cet algorithme de hachage devrait fournir une signature compacte pour toutes les informations visuellement significatives contenues dans le document. Ce condensé permettra la création de systèmes de sécurité hybrides pour sécuriser tout le document. Ceci peut être réalisé grâce à des algorithmes d'analyse du document. Cependant ceux-ci ont besoin d'être porté à un niveau de performance sans précédent, en particulier leur fiabilité qui dépend de leur stabilité.
Après avoir défini le contexte de l'étude et ce qu'est un algorithme stable, nous nous sommes attachés à produire des algorithmes stables pour la description de la mise en page, la segmentation d'un document, la reconnaissance de caractères et la description des zones graphiques.

Mots clés : stabilité, analyse d'images de document, sécurité, impression et scan, segmentation, hachage perceptuel d'image, superpixels, composantes connexes en couleurs, descripteur de mise en page, OCR.

On the stability of document analysis algorithms. Application to hybrid document hashing technologies.

Abstract: An innumerable number of documents is being printed, scanned, faxed, photographed every day. These documents are hybrid : they exist as both hard copies and digital copies. Moreover their digital copies can be viewed and modified simultaneously in many places. With the availability of image modification software, it has become very easy to modify or forge a document. This creates a rising need for an authentication scheme capable of handling these hybrid documents.
Current solutions rely on separate authentication schemes for paper and digital documents. Other solutions rely on manual visual verification and offer only partial security or require that sensitive documents be stored outside the company's premises and a network access at the verification time.
In order to overcome all these issues we propose to create a semantic hashing algorithm for document images. This hashing algorithm should provide a compact digest for all the visually significant information contained in the document. This digest will allow current hybrid security systems to secure all the document. This can be achieved thanks to document analysis algorithms. However those need to be brought to an unprecedented level of performance, in particular for their reliability which depends on their stability.
After defining the context of this study and what is a stable algorithm, we focused on producing stable algorithms for layout description, document segmentation, character recognition and describing the graphical parts of a document.
Keywords: stability, document image analysis, security, print and scan, segmentation, perceptual image hashing, superpixels, color connected components, layout descriptor, OCR.

Informatique Image Interaction
Univ. La Rochelle

Laboratoire Informatique, Image, Interaction
Faculté des Sciences & Technologies
Avenue Michel Crépeau

17042 LA ROCHELLE CEDEX 1

Communauté d'Agglomération de La Rochelle