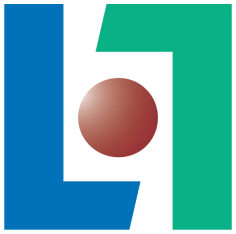


LaBRI
LABORATOIRE BORDELAIS DE RECHERCHE EN INFORMATIQUE
Structuration et analyse d'image

RABEUX VINCENT
Janvier 2013

ÉVALUATION DE LA QUALITÉ DES DOCUMENTS ANCIENS NUMÉRISÉS

THÈSE
par
VINCENT RABEUX

Présentée et soutenue publiquement
pour l'obtention du

LE 6 MARS 2013
DOCTORAT DE L'UNIVERSITÉ DE BORDEAUX 1

Directeur de thèse :	JEAN PHILIPPE DOMENGER	Labri, Professeur à l'Université Bordeaux 1
Encadrant de thèse :	NICHOLAS JOURNET	Labri, Mcf à l'Université Bordeaux 1
Rapporteurs :		
	JEAN-MARC OGIER	L3I, Professeur à l'université de La Rochelle
	JEAN-YVES RAMEL	LI, Professeur à l'université de Tours
Examinateurs :		
	VERONIQUE EGLIN	LIRIS, Mcf à l'université de Lyon

Abstract

This PhD. thesis deals with quality evaluation of digitized document images. In order to measure the quality of a document image, we propose to create new features dedicated to the characterization of most commons degradations. We also propose to use these features to create prediction models able to predict the performances of different types of document analysis algorithms. The features are defined by analyzing the impact of a specific degradation on the results of an algorithm and then used to create statistical regressors.

The relevance of the proposed features and predictions models, is analyzed in several experimentations. The first one aims to predict the performance of different binarization methods. The second experiment aims to create an automatic procedure able to select the best binarization method for each image. At last, the third experiment aims to create a prediction model for two commonly used OCRs.

This work on performance prediction algorithms is also an opportunity to discuss the scientific problems of creating ground-truth for performance evaluation.

Keywords :
Ancient document images ; quality evaluation ; image features ; optical character recognition ; performance evaluation ; synthetic document image generation ; ground-truth creation.

Résumé

Les travaux de recherche présentés dans ce manuscrit décrivent plusieurs apports au thème de l'évaluation de la qualité d'images de documents numérisés. Pour cela nous proposons de nouveaux descripteurs permettant de quantifier les dégradations les plus couramment rencontrées sur les images de documents numérisés. Nous proposons également une méthodologie s'appuyant sur le calcul de ces descripteurs et permettant de prédire les performances d'algorithmes de traitement et d'analyse d'images de documents. Les descripteurs sont définis en analysant l'influence des dégradations sur les performances de différents algorithmes, puis utilisés pour créer des modèles de prédiction à l'aide de régresseurs statistiques.

La pertinence, des descripteurs proposés et de la méthodologie de prédiction, est validée de plusieurs façons. Premièrement, par la prédiction des performances de onze algorithmes de binarisation. Deuxièmement par la création d'un processus automatique de sélection de l'algorithme de binarisation le plus performant pour chaque image. Puis pour finir, par la prédiction des performances de deux OCRs en fonction de l'importance du défaut de transparence (diffusion de l'encre du recto sur le verso d'un document).

Ce travail sur la prédiction des performances d'algorithmes est aussi l'occasion d'aborder les problèmes scientifiques liés à la création de vérités-terrains et d'évaluation de performances.

Mots-clés :
Images de documents anciens ; évaluation de la qualité ; modèles de prédiction ; descripteurs images ; binarisation ; reconnaissance de caractères ; évaluation de performances ; génération de documents synthétiques ; création de vérité-terrains ; régression linéaire.

Table des matières

Introduction

Au cours des siècles, le livre a été associé à des supports radicalement différents : papyrus, bois, de soie, parchemins, etc. Jusqu'aux années 80, le livre reste un objet physique qui par impression peut être copié et diffusé. L'avènement de l'informatique marque un réel changement, les livres produits à partir de ces années sont alors créés dans un format numérique puis imprimé sur un support physique. Cependant, le format numérique original n'a été que très peu souvent conservé. Ainsi, la sauvegarde et l'accès aux ouvrages existants (anciens ou non) se font majoritairement *via* leur forme physique. Les 20 dernières années ont vu se mettre en place de nombreuses campagnes de numérisation. C'est dans ce contexte que de nombreux besoins et enjeux scientifiques ont vu le jour. L'un d'entre eux est celui de l'évaluation de la qualité de l'image numérique produite.

Si la qualité du matériel permettant de numériser des documents s'est significativement améliorée ces dernières années, il se trouve qu'en raison de la qualité même du document (trous, papier, reliure, etc.) ou d'erreurs provenant de la numérisation (mauvais réglages du scanner par exemple) la version numérique produite n'est parfois pas de qualité acceptable. C'est pour cela que de nombreuses sociétés de numérisation effectuent un contrôle qualité manuel, souvent complexe, après cette étape de numérisation.

Un enjeu clair est donc de pouvoir être en mesure d'identifier dès la phase de numérisation la présence de défauts (colorimétrie, transparence, placement de l'ouvrage, etc.). Cet objectif d'évaluation de la qualité entrouvre la voie à d'autres applications. En effet, les documents numérisés, dès lors qu'ils ne sont pas destinés à un simple archivage, peuvent être sujets à divers traitements post numérisation (binarisation, restoration, etc.) et d'analyse de contenu (OCR, indexation, etc.). Cependant, si dans ce domaine de nombreux algorithmes existent, leurs performances baissent face à la quantité et l'importance des dégradations présentes sur les images. Un deuxième enjeu important est donc de pouvoir analyser les dégradations d'une image de document en amont de l'exécution d'une chaîne de traitements et d'analyse d'images de documents.

C'est dans ce contexte d'évaluation de la qualité d'images de documents numérisés que ces travaux de thèse se placent. Le premier apport de ces travaux de recherche est la mise en place de nouveaux descripteurs permettant de caractériser la présence, la quantité et l'importance de certaines dégradations courantes (transparence, problèmes d'illumination, etc.). Le deuxième aspect innovant de ces travaux est la proposition d'une méthodologie qui, sur la base des descripteurs calculés, permet de prédire les performances d'algorithmes d'analyse et de traitement d'images de documents. De cette manière, il est possible de prédire les performances d'une chaîne entière de traitements, mais aussi de sélectionner l'algorithme le plus performant à chaque étape d'une chaîne afin de garantir les meilleures performances possible pour chaque image.

Le premier chapitre de ce manuscrit présente de façon générale le contexte de cette thèse. Nous présenterons les grandes campagnes de numérisation européenne ainsi que leur fonctionnement. Cela permet de préciser les contraintes du contrôle qualité et de lister les dégradations les plus souvent présentes sur les images de documents. Dans un deuxième temps, nous analyserons ,à travers un état de l'art, l'influence des dégradations sur les performances des chaînes de traitements.

Le deuxième chapitre se concentre sur la création d'une méthodologie permettant de concevoir des descripteurs caractérisant la qualité d'une image de document. Nous commencerons par un état de l'art général des descripteurs utilisés pour l'analyse et le traitement d'image, puis nous présenterons les descripteurs existants et dédiés à la caractérisation de la qualité d'une image de document binaire. Nous

étudions également les algorithmes de modélisation et de restauration de dégradations.Cet état de l'art a pour objectif de construire une méthodologie permettant la création d'un ensemble de descripteurs caractérisant des défauts. Cette méthodologie est basée sur l'analyse conjointe d'une dégradation (les différentes formes qu'elle peut prendre), des algorithmes de génération et de restauration de cette dernière et du type d'algorithme à prédire. Elle est illustrée à travers la création de descripteurs permettant de caractériser deux dégradations fréquemment observées dans les documents anciens : les perturbations fond-encre (comme par exemple, les taches ou les défauts sur l'encre) et la transparence (visibilité de l'encre du verso sur le recto).

Le troisième chapitre se focalise quant à lui sur les méthodes et techniques permettant, à partir de descripteurs du chapitre précédent, de prédire les performances d'algorithmes. Dans un premier temps, un état de l'art présente les différentes propositions scientifiques décrivant comment il est possible de prédire ou sélectionner un OCR adéquat en fonction de l'état de dégradation des images. Dans un deuxième temps, nous détaillerons notre méthode permettant de prédire les performances de plusieurs algorithmes de binarisation (séparation fond-encre). Pour cela plusieurs modèles de prédiction seront entrainés puis validés. Ces derniers sont ensuite utilisés conjointement pour sélectionner automatiquement la meilleure méthode de binarisation pour chaque image de document. Dans un troisième temps, le même protocole est utilisé pour prédire les performances de deux OCRs en fonction de l'importance de la dégradation de transparence. Ce chapitre permet de valider, sur deux exemples, notre approche pour l'évaluation de la qualité d'une image de document numérisée et la prédiction des performances d'algorithmes pour sélectionner automatiquement celui qui permettra d'obtenir les meilleurs résultats pour chaque image.

Le chapitre quatre met en avant les difficultés relatives à la création, l'utilisation et la diffusion de corpus d'images de documents annotées de leurs vérité-terrains. En effet, des enjeux scientifiques et techniques liés à l'évaluation de la robustesse des méthodes de segmentation des dégradations ou à l'évaluation des performances des algorithmes à prédire ont permis de soulever un ensemble de problèmes auxquels nous répondons par la conception de logiciels collaboratifs dédiés à la création de vérité-terrains. Pour cela, nous considérons trois méthodes de création de vérité-terrains : la génération semi-synthétique, l'annotation d'un document réel par un expert et l'acquisition d'informations perceptuelles sur les images. Ces logiciels reposent sur une architecture distribuée dont les concepts fondamentaux sont présentés en dernière section du chapitre.

Au travers de deux apports distincts que sont la création de descripteurs pour l'évaluation de la qualité d'images de documents et la proposition d'une méthodologie permettant la prédiction des performances d'algorithmes d'analyse et de traitement d'images de documents, ce mémoire traite d'un sujet encore peu étudié dans la littérature.

Chapitre 1

Contexte et motivations

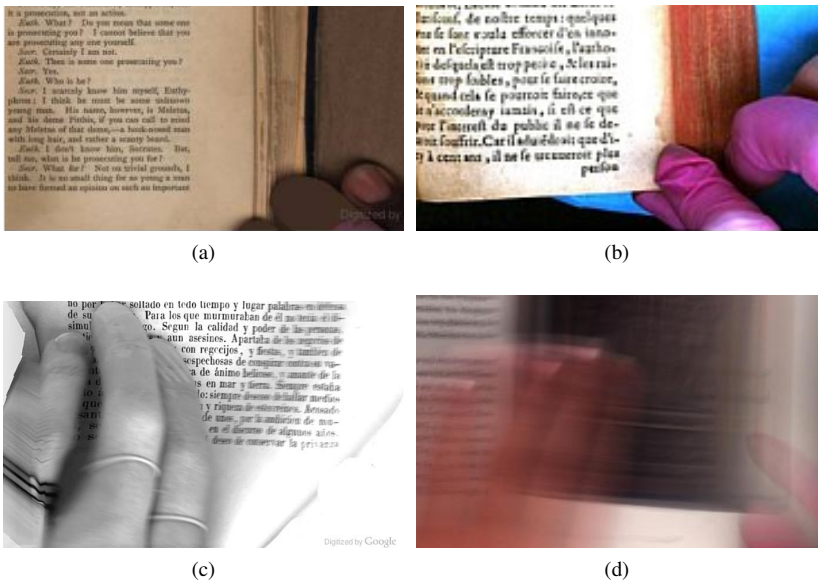
Bien avant l'invention de l'imprimerie par Gutenberg au milieu du XVe siècle, différents supports et écritures sont utilisés pour communiquer. Les Égyptiens écrivaient sur des rouleaux de papyrus, les Chinois sur des livres de bois et de soie et les Romains écrivaient sur des parchemins (appelés codex). Les moines copistes avaient pour tâches de recopier les livres afin de les préserver ou de les diffuser. Ainsi les textes religieux ou philosophiques pouvaient être consultés à travers toute l'Europe. Cependant la reproduction d'un ouvrage pouvait mettre plusieurs mois. Dans le courant du XIVe siècle, l'Occident importe une technique d'origine chinoise appelée "xylographie" (utilisation d'une tablette de bois gravé comme empreinte). Cette technique, ancêtre de l'imprimerie moderne, remplace peu à peu les techniques de copie du Moyen Âge. C'est l'imprimerie moderne donnant naissance à la typographie (règles d'écriture) qui permet finalement de produire des livres en très grande quantité. Six siècles plus tard, la quantité de documents produits est colossale. Ces derniers, principalement stockés dans les bibliothèques constituent une part importante du patrimoine culturel et scientifique [1].

L'informatique marque, dans les années 80, une nouvelle évolution dans la manière de concevoir et de diffuser les documents. L'utilisation de ce nouveau média permet de remplacer en partie le support papier et engendre de très sérieuses modifications dans la façon dont les ouvrages sont produits, préservés, diffusés et valorisés. Dans les années 90, de nombreuses campagnes de numérisation de documents voient le jour. Dans les premières campagnes, l'outil informatique est essentiellement utilisé à des fins d'archivage et de préservation. Les premiers sites web dédiés à la valorisation de l'image d'un document apparaissent et créent un nouveau besoin, celui de l'indexation qui permet d'obtenir un accès simplifié aux informations contenues dans les images (textes, illustrations, ...).

Même dans des conditions de conservation quasi idéales, les documents antérieurs au XXe siècle se trouvent parfois dégradés, par exemple dû à des intempéries, des inondations, des incendies. Ces dégradations rendent les étapes de numérisation et de valorisation complexes. Dans un souci de préservation du document, il est impératif de garantir que l'image résultant de la numérisation soit la plus fidèle possible au document original. Il serait, par exemple, inacceptable de sauvegarder une image floue ou avec une mauvaise colorimétrie. La vérification de la qualité de la numérisation est actuellement réalisée lors du contrôle par un être humain. Malgré ce contrôle, il arrive que des documents mal numérisés soient, par erreur, mis en ligne comme nous pouvons le voir sur la figure 1.1 provenant de *Google Books* [2] et la figure 1.5b présentant une image d'un document numérisé par *Arkhenum* [3].

Dans le contexte de la numérisation de documents anciens, une première problématique scientifique est d'aider ou d'automatiser la phase de contrôle qualité réalisée manuellement par les opérateurs des scanners. Une seconde problématique identifiée est celle liée à la volonté de donner un accès simplifié et rapide aux données contenues dans les documents numérisés. Cet objectif passe par la mise en place d'une chaîne de traitement (workflow) composée d'algorithmes de traitements ou d'analyses d'images permettant d'extraire la structure, transcrire le texte, identifier les scripteurs, analyser les illustrations, etc. Même si des résultats scientifiques ont permis la réalisation de nombreuses avancées, l'expérience

1. http://cerig.efpg.inpg.fr/dossier/impression-numerique/sommaire.htm
2. http://books.google.fr/
3. http://www.arkhenum.fr/



(a) (b)

(c) (d)

FIGURE 1.1 – *Erreurs de numérisation : la présence des doigts de l'opérateur maintenant la page à plat n'est pas signalée ni corrigée lors du contrôle qualité.*

nous montre qu'il n'existe pas à l'heure actuelle d'algorithmes idéals pouvant s'appliquer à tous les types de documents. De fait, la construction d'une chaîne de traitements se fait manuellement pour chaque type de document. Cette étape est complexe, fastidieuse et coûteuse. Les opérateurs doivent s'appuyer sur des critères subjectifs afin de choisir au mieux les algorithmes de la chaîne et peu d'indicateurs objectifs permettent de garantir qu'ils ont fait le meilleur choix.

À ce stade un certain nombre de questions se posent : quelles sont les étapes suivies par un document de sa numérisation jusqu'à sa mise en ligne ? Quels sont les usages ? Quels algorithmes appliquer aux documents ? Comment garantir l'adéquation entre les performances d'un algorithme et les usages donnés ? Quelles sont les dégradations présentes sur les documents anciens ? À quels types de dégradations les algorithmes de traitement sont-ils sensibles ?

Dans ce chapitre, nous tenterons de répondre à ces questions. Nous détaillerons le cycle de vie de l'image de document, de sa numérisation jusqu'à sa diffusion (section 1.1.1). Puis, nous analyserons certaines caractéristiques des documents (section 1.2) et en particulier les catégories de dégradations et les éléments structurels (paragraphes, illustrations, titres, etc.) qui les composent. Pour finir, la section 1.3.2 est quant à elle dédiée à l'influence de ces caractéristiques sur la chaine de traitement appliquée aux documents. Nous positionnerons cette thèse sur la problématique d'évaluation de la qualité afin de prédire les performances d'algorithmes et ainsi d'optimiser les chaînes de traitements appliquées aux images de documents.

1.1 Chaînes de numérisation des documents

1.1.1 Enjeux scientifiques d'une campagne de numérisation de documents

On peut distinguer deux étapes importantes dans le processus suivi par une image de document : sa numérisation et son utilisation. L'utilisation d'un document est fortement liée aux besoins et usages de l'utilisateur. Ceux-ci peuvent être complexes et multiples, par exemple, dans le cas des bibliothèques :

l'archivage, la préservation, et la valorisation.

1.1.1.1 Historique des campagnes de numérisation

Les documents patrimoniaux sont de natures très diverses et s'étalent sur plusieurs siècles. On peut considérer que la production de documents numériques a commencé dans les années 80 [**?**]. De fait, tous les documents antérieurs à cette date ne disposent pas de version numérique et sont conservés pour la plupart dans des bibliothèques soit sur un support papier soit sur micros films. On ne peut que constater que les conditions de conservation des documents sont variables. La nature originelle du document ainsi que les conditions de conservation participent à sa dégradation plus ou moins rapide. La numérisation des documents a eu et a encore comme premier objectif de sauvegarder et de préserver les documents.

Les premières campagnes de numérisation de documents débutent dans les années 60. La quantité de documents (journaux, articles scientifiques, livres anciens, etc.) est alors importante. Dans ce contexte, la préservation de notre patrimoine dans toute sa diversité et sa richesse est devenue un des domaines les plus importants de l'économie numérique. L'évolution des performances techniques liées à la numérisation dans les années 90 a engendré la création de nombreux projets à travers le monde. En 1993, la première bibliothèque numérique française (**A**ssociation des **B**ibliophiles **U**niverselles) est créée. Cette dernière contient 288 textes de 101 auteurs en janvier 2002. En 1997 c'est au tour de la **B**ibliothèque **N**ational de **F**rance de créer le projet *Gallica* (figure 1.3.a) [4]. En 2004, le projet *Google Books* démarre. Disposant de moyens considérables, *Google* entreprend alors une numérisation massive du patrimoine documentaire mondial. En 2008, le succès et l'essor du projet de *Google* incitent *Gallica* à accélérer la numérisation des documents avec pour objectif de numériser près de 100 000 ouvrages par an, 2500 images de documents sont mises en ligne par semaine. En 2010, 400 000 ouvrages ont été numérisés soit environ 60 millions de pages [5]. *Gallica* est à ce jour la bibliothèque numérique française la plus importante. L'initiative de *Google* entraine aussi la commission européenne à s'interroger sur la nécessité de numériser le patrimoine écrit européen. Suite à cette réflexion en 2008 le projet *Europeana* [6] est lancé. Ce projet collaboratif implique un grand nombre de partenaires comme la *BNF*, la *British Library* à Londres, le *Rijksmuseum* à Amsterdam, et le *Louvre* à Paris. En s'inspirant du projet *Gallica,* mais avec une interface plus moderne et une indexation pleine texte plus évoluée, *Europeana* propose un catalogue de recherche multilingue. Ce projet comptait quinze millions d'images en 2011. Néanmoins, *Google Books* reste la bibliothèque numérique la plus importante à travers le monde : elle comptait plus de quinze millions de livres en 2010 dont 3 millions consultables. Nous sommes entrés dans une phase que l'on peut qualifier de *numérisation de masse*.

La quantité d'images de documents mise en ligne pose de gros problèmes d'infrastructure. Les images de documents sont souvent de très grandes tailles, et si l'on prend le cas de la BNF qui numérise 2500 ouvrages par semaine nous arrivons très vite à plusieurs Tera octets de données ajoutées par semaine. Le jeudi 20 novembre 2008, *Europeana* tombe en panne pendant presque 48 heures [7] à cause d'une mauvaise estimation des possibilités de montée en charge du site. Les évolutions techniques liées au cloud computing (informatique dématérialisée) permettront de répondre à ces besoins et de proposer des solutions à très haute disponibilité [8]. Ce type d'infrastructure engendre des coûts financiers et d'intégrations importantes. Certains projets de recherche comme IMPACT (Improving access to text) [9] et SCAPE (SCAlable Preservation Environments) [10] [11] ont pour objectif secondaire de résoudre une partie de ces problématiques.

4. http://gallica.bnf.fr/
5. http://www.bnf.fr/fr/collections_et_services/bibliotheques_numeriques_gallica/a.numerisation_masse_bnf.html
6. http://www.europeana.eu/portal/
7. Les explications du crash du site Europeana : http://www.silicon.fr/les-explications-du-crash-du-site-europeana-32660.html
8. http://www.haute-disponibilite.net/2009/10/26/promesses-haute-disponibilite-cloud-computing/
9. http://www.impact-project.eu/
10. http://www.scape-project.eu/
11. https://github.com/schenck/scape ; https://github.com/shsdev/scape ; https://github.com/yuliaro/scape

(a) (b) (c)

FIGURE 1.2 – *Exemples de scanners produits par la société I2S (http://www.i2s-digibook.com/) : a. un scanner manuel, b. un scanner grand format , c. un scanner automatique avec tourneur de page.*

1.1.1.2 Valorisation des documents numériques

L'évolution des nouvelles technologies de numérisation oblige les bibliothèques à s'interroger sur leurs missions. Par exemple : à quels besoins répondent-elles ? Actuellement, elles proposent principalement un accès au catalogue d'images numérisées. Mais la mise en ligne d'un catalogue aussi large ne permet pas de valoriser pleinement ces documents. En effet, l'utilisateur doit disposer d'un ensemble d'outils permettant une exploitation aisée des documents. Par exemple la recherche par mots-clés (à l'image d'un moteur de recherche) est devenue un outil indispensable. Ou encore la possibilité de naviguer facilement dans des données de taille conséquente est un réel défi.

(a) (b)

FIGURE 1.3 – *Valorisation des documents : a. un document disponible en ligne sur Gallica, b. un document numérique disponible sur une liseuse.*

La recherche par mots-clés est impossible sur l'image brute. Il est donc nécessaire d'extraire et d'indexer son contenu. L'indexation d'un document consiste à repérer une information pertinente dans le contenu de l'image afin de créer des index terminologiques. Ces informations font le lien entre l'image et son contenu. Nous pouvons différencier deux types d'indexations : l'indexation manuelle et l'indexation automatique. L'indexation manuelle est faite par les opérateurs, les experts ou les historiens et consiste à annoter le document en fournissant un ensemble d'informations le plus souvent textuelles. Les informations considérées par l'indexation manuelle ont une sémantique élevée et ne peuvent être renseignées que par un expert. D'autres informations par exemple, l'auteur ou la date d'édition si elles

sont contenues dans l'ouvrage peuvent être renseignés à partir d'une extraction automatique du contenu. L'aide à la navigation quant à elle peut se faire à plusieurs niveaux :
– Au niveau du catalogue : classification par année d'édition, par thème, par type de contenu,.... Ce type de classification utilise essentiellement les métadonnées extraites lors de la phase d'indexation.
– Au niveau de l'ouvrage : navigation interactive en utilisant la table des matières, de chapitre en chapitre, d'illustration en illustration. Ce type de navigation repose aussi sur la phase d'indexation, mais utilise la structure du document. On différencie deux types de structures [MRK03] : la structure physique (blocs de texte, illustrations) et la structure logique (paragraphes, titre de chapitre, titre de section, numérotation des pages, table des matières, table des illustrations, ...).

La valorisation des documents ne s'arrête donc pas à la mise en ligne de leurs images. L'avènement des livres (figure 1.3.b) et des liseuses numériques génère également d'autres besoins par exemple la recomposition du livre. L'objectif est donc de convertir le livre au format image vers un format dit numérique (pdf, Mobipocket, epub, ...) [MMS11, MMS10]. Le contenu du document doit alors être extrait du document afin de reconstruire une version interactive.

1.1.2 Algorithmes associés aux chaînes de traitements

Dans cette sous-section, nous détaillerons les principales étapes d'une campagne de numérisation, ainsi que les algorithmes associés à ces différentes étapes. Nous présenterons sommairement les différents traitements qui sont appliqués aux documents avant leur mise en ligne et leur indexation.

1.1.2.1 Du document physique à sa version numérisée

De plus en plus de campagnes de numérisation suivent le protocole présenté en figure 1.4. On peut distinguer plusieurs étapes importantes : l'établissement du cahier des charges, la phase de numérisation, le contrôle qualité interne et le contrôle qualité externe (tests de recette).

Établissement du cahier des charges

Le client (bibliothèque, mairie, administration, etc.), possédant les documents physiques à numériser, charge un prestataire de numériser un ensemble de documents (appelé *lot*). Le prestataire doit répondre au cahier des charges définissant les spécifications du service de numérisation à réaliser. Ce document contractuel est utilisé comme un référentiel entre le prestataire et son client, entre la maîtrise d'oeuvre, et la maîtrise d'ouvrage. Il peut par exemple contenir la résolution des images finales, leurs formats, ou les traitements qui peuvent leur être appliqués, mais aussi des contraintes liées à la qualité. Par exemple le client peut exiger qu'au maximum 1% des documents soient flous. En général, le client demande à ce que les images de documents livrées soient les plus fidèles possible aux documents originaux. Cette vérification est faite, par échantillonnage du lot, lors du contrôle qualité.

La numérisation

Une fois le lot reçu, le prestataire commence la numérisation en choisissant le type de scanner à utiliser (si celui-ci n'est pas spécifié dans le cahier des charges). En principe, chaque document est numérisé page par page par un opérateur qui doit paramétrer le scanner en fonction de la nature du document. Il est parfois nécessaire de re paramétrer le scanner au cours de la numérisation d'un même document pour s'adapter au contenu du document ou à des changements extérieurs (luminosité ambiante par exemple). La tâche de l'opérateur n'est pas aisée : il doit faire preuve d'une attention constante afin de ne pas faire d'erreur (oubli de pages, flou, mauvaise colorimétrie, balance des blancs, etc.) et ce malgré la grande quantité de documents qu'il doit numériser au cours de la journée.

Le contrôle qualité

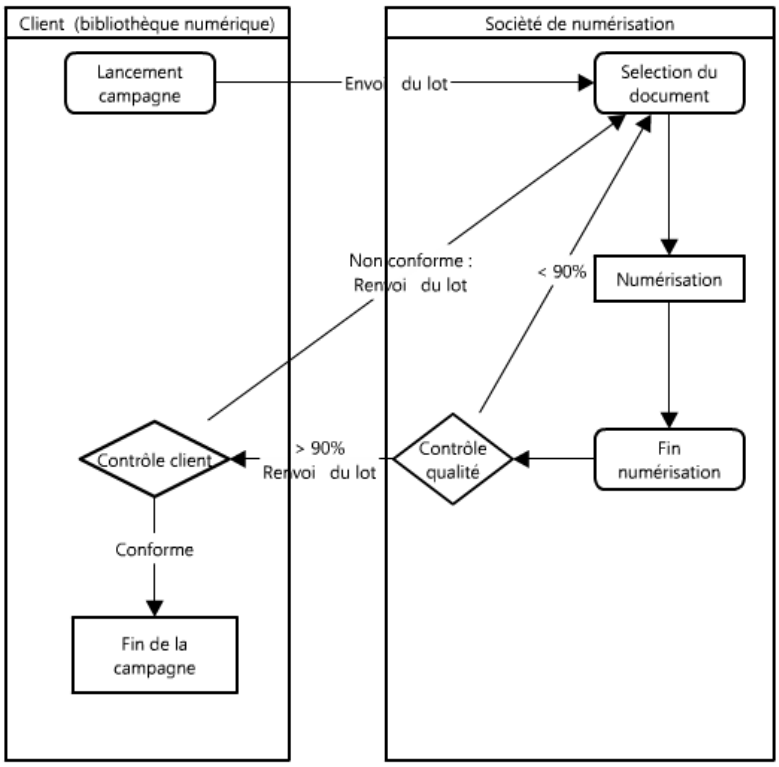


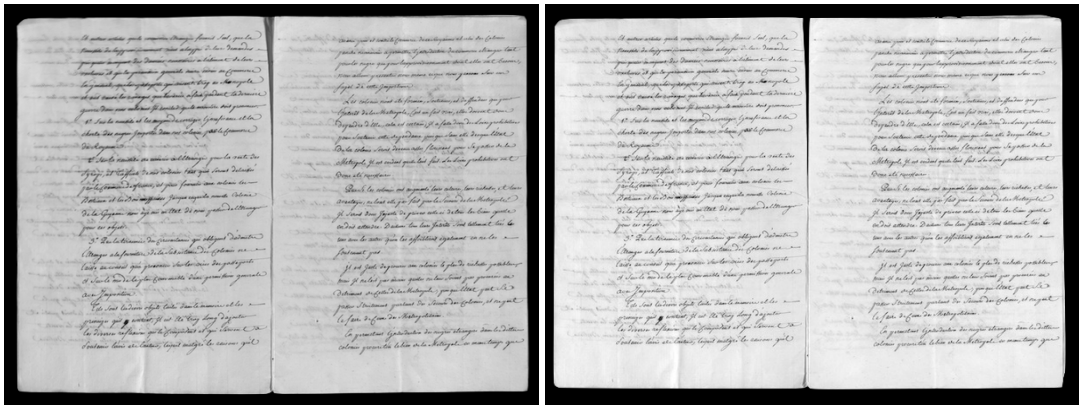
FIGURE 1.4 – *Protocole de numérisation d'un lot de documents entre une bibliothèque (le client) et un prestataire de numérisation. Deux contrôles qualité sont effectués. Le premier est réalisé en interne. Le second est réalisé chez le client lors de la livraison des images. Le refus d'un lot lors de ces contrôles peut engendrer des coûts importants de renumérisation.*

Le contrôle qualité a pour objectif de vérifier que l'ensemble des images numérisées est conforme aux exigences du client (exigences définies dans le cahier des charges). Dans la plupart des campagnes de numérisation, deux contrôles qualité sont effectués : le premier par le prestataire, le second par le client.

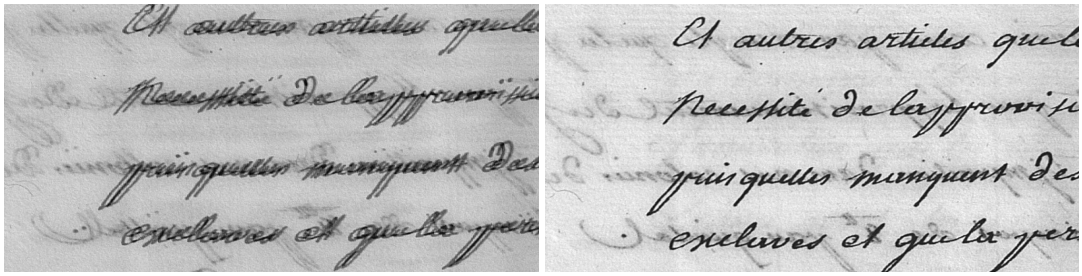
La phase de numérisation étant une phase complexe et sujette à un grand nombre d'erreurs, les prestataires de numérisation sont souvent obligés de faire un contrôle qualité interne avant de livrer le lot d'images au client. Lors de cette étape, les images présentant un défaut de numérisation doivent être détectées et numérisées à nouveau. Cette opération est une étape critique pour deux raisons :

1. Le nombre d'images à vérifier est très important et pour des raisons de productivité les images sont inspectées à faible résolution sous forme de vignettes. Or, les vignettes sont parfois insuffisantes pour détecter certains défauts comme le montre la figure 1.5 où il est difficile de remarquer juste en visualisant la vignette que l'image est floue.

2. Le fait de devoir numériser une nouvelle fois les pages défectueuses engendre des coûts supplémentaires qui peuvent être très importants. En général, le document n'étant plus sur la table du scanner, une opération de numérisation en cours doit être mise en pause pour renumériser les pages du document présentant des défauts. L'opérateur chargé de la numérisation doit aussi rechercher les pages concernées dans le document physique et reparamètrer le scanner juste pour ces quelques pages.



(a) Deux images en miniature



(b) Zoom sur chacune des images

FIGURE *1.5 – Illustration de la difficulté du contrôle qualité manuel : il est très difficile de remarquer que la page de gauche est floue sans zoomer.*

Si le document numérisé passe le contrôle qualité du prestataire, il est envoyé au client. Le client teste statistiquement le lot d'images afin d'en vérifier la conformité. Si le taux d'images non conformes est supérieur à un seuil (précisé dans le cahier des charges), le lot sera de nouveau entièrement renumérisé. Les coûts engendrés par un refus du lot d'images lors de cette étape sont d'autant plus importants étant donné que le document physique doit, dans la plupart des cas, être renvoyé au prestataire.

Afin de minimiser les coûts engendrés par une mauvaise paramétrisation du scanner ou par d'autres erreurs de numérisation, il serait possible d'alerter l'opérateur le plus tôt possible, quand le document est encore en sa possession, sur la table du scanner. Des scanners intelligents basés sur une phase d'apprentissage existent et permettent de classer automatiquement les documents numérisés (factures, tickets, etc.), mais rien n'a été fait concernant la qualité de la numérisation des documents. Dans le cadre du projet DIGIDOC (Document Image diGitisation with Interactive DescriptiOn Capability) [12], des travaux de recherches tentent de concevoir un scanner cognitif capable de se re-paramétrer et de s'adapter aux caractéristiques de la page.

1.1.2.2 Analyse, traitement et diffusion de l'image de document

Une fois l'image de document livrée, cette dernière est mise en ligne sur les plateformes de traitements numériques de la bibliothèque. L'image sera alors téléversée, dans un premier temps de façon brute, puis dans un second temps analysée en fonction de l'usage attendu. L'analyse peut être divisée en une succession d'algorithmes qui constitue la *chaîne de traitement* ou *workflow*. Chacun de ces algorithmes modifie directement l'image en produisant une nouvelle image ou associe un ensemble d'annotations (meta-données) à l'image. Les informations ainsi ajoutées sont utilisées soit par l'indexation soit par les

12. http://digidoc.labri.fr/

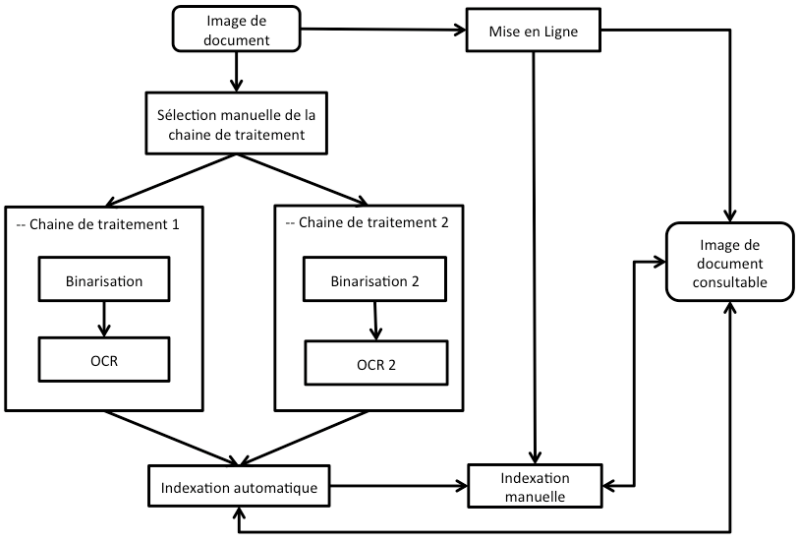


FIGURE 1.6 – *Cycle de vie de l'image de document. L'image brute sera tout d'abord mise en ligne pendant qu'un ensemble de traitements lui sont appliqués. Ces traitements ont pour but d'améliorer la lisibilité de l'image (restauration), mais aussi de l'indexer automatiquement. Le résultat de l'indexation peut être modifié et corrigé de façon manuelle.*

algorithmes de traitement. Les résultats de chaque étape de la chaîne peuvent être corrigés ou enrichis manuellement. L'ensemble de ces étapes est schématisé sur la figure 1.6 qui présente le cycle de vie associé au traitement d'un document.

Il est impossible d'être exhaustif sur les catégories d'algorithmes applicables à l'image de document. Néanmoins, nous considérons plusieurs familles principales qui participent à la chaîne de traitements :
- **Prétraitements, restaurations, filtrages** : ces algorithmes ont pour objectif d'atténuer les dégradations et donc d'améliorer la qualité de l'image. En particulier il s'agit de proposer une version de l'image qui serait la meilleure possible pour les traitements suivants dans la chaine. Cette famille de traitement compte par exemple, les algorithmes capables de débruiter une image, d'enlever sa transparence ou encore de redresser l'image.
- **Binarisation** : ce type d'algorithme a pour objectif de séparer le fond et l'encre.
- **Analyse de structures** : identification et extraction des paragraphes, des illustrations, des tableaux, des formules mathématiques, des tables des matières, des titres, ou encore les bordures du livre.
- **OCR (Optical Character Recognition)** : identification des caractères de l'image. Les OCRs modernes sont souvent complexes et incluent des algorithmes de restauration, binarisation, d'analyse de structure, des moteurs de reconnaissance des caractères aidés par des dictionnaires et des grammaires.
- **Recomposition du livre** : une fois l'OCR réalisé sur l'ouvrage, nous disposons d'informations sur le contenu du livre. Il est alors possible de l'indexer et de le recomposer en générant automatiquement la table des matières par exemple.

À l'heure actuelle, le choix des algorithmes et leurs paramétrages sur une chaîne de traitement particulière sont réalisés manuellement pour un ouvrage donné. Malheureusement, la sélection manuelle n'est pas toujours adaptée. En effet, le type de contenu des pages d'un même ouvrage peut varier. Une sélection et une configuration automatiques de la chaîne de traitement pour chaque image permettraient de garantir les meilleurs résultats possible (avec les algorithmes disponibles), et ce en fonction du cas d'utilisation de l'image (archivage, compression, visualisation, reconnaissance de caractères, indexation, etc.). Une analyse générale de l'image et de ses caractéristiques permettrait d'extraire les éléments pertinents pour une sélection automatique des algorithmes dont les performances sont les plus en adéquation avec l'objectif.

1.2 Analyse des caractéristiques des documents anciens

Un grand nombre de défauts peuvent être évités ou corrigés pendant la phase de numérisation en signalant à l'opérateur que les paramètres ne sont pas optimaux. D'autres sont présents sur le document physique. Certes, certains peuvent être atténués par des réglages spécifiques lors de la numérisation, mais ils ne peuvent être complètement corrigés qu'avec des algorithmes de restauration appliqués sur l'image du document numérisé.

Dans l'objectif d'alerter l'opérateur pour corriger un défaut, ou d'automatiquement sélectionner un algorithme de restauration d'image, nous réaliserons une taxonomie des principales dégradations d'un document ancien en deux catégories : les dégradations inhérentes à l'ouvrage lui-même et celles issues de la numérisation.

1.2.1 Les dégradations de l'ouvrage

La première catégorie de dégradation regroupe tous les défauts qui proviennent essentiellement de l'état document. Dans la plupart des cas, ces dégradations ne peuvent être atténuées. À titre d'exemple voici une illustration des dégradations associées aux ouvrages :

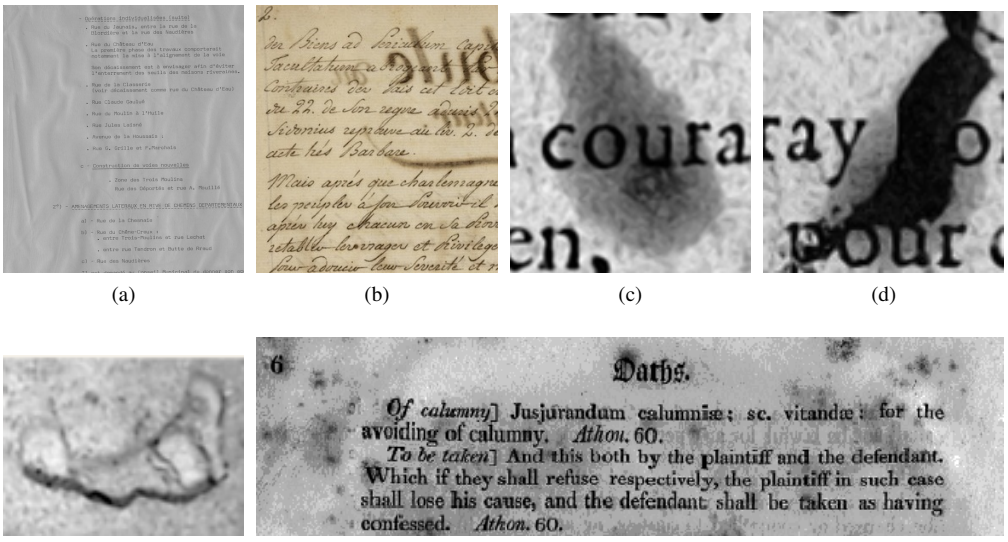


FIGURE 1.7 – *Exemples de défauts présents sur les documents avant même leur numérisation : (a) ondulations de la page due à l'humidité, (b) transparence du verso de la page (grande lettre de titre) (c) (d) et (f) montrent des taches d'usures et de mauvaise conservation. (e) montre un trou dans la page. Les images ont été numérisées par la société Arkhénum.*

- Les **défauts touchant les caractères**. L'encrage faible ou de mauvaises qualités peut dégrader le contraste entre l'écriture et le fond. Dans d'autres cas, l'encre peut se diffuser de façon trop importante dans le papier et créer des caractères épais avec des formes non uniformes et des trous bouchés.
- La **transparence du verso** : certains papiers, trop fins, laissent apparaître l'encre présente au verso du document à travers le recto. Selon la quantité d'encre qui est diffusée, la transparence peut être plus ou moins visible. Un exemple est présent sur la figure 1.7b.
- Les **taches** sont souvent dues à l'humidité, mais peuvent aussi être la cause de mauvais usages. Les figures 1.7c, 1.7d 1.7f montrent des exemples de taches pouvant avoir plusieurs origines. La figure 1.8 montre l'effet d'une tache d'humidité sur plusieurs pages consécutives d'un même ouvrage.
- Les **pliures et ondulations** peuvent avoir plusieurs origines. Par exemple, le coin d'un document

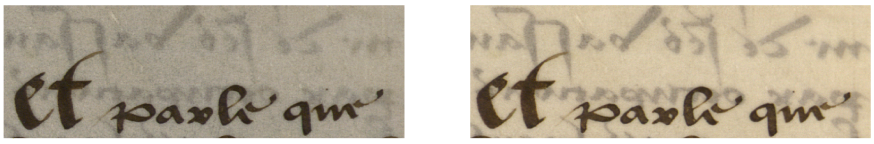
FIGURE 1.8 – *Une tache d'humidité (en gris clair) s'est créée sur le côté du livre. Plusieurs pages sont affectées par cette dégradation.*

peut être plié comme le montre la figure 1.7a, la reliure peut aussi créer une ondulation locale sur une partie du document. Certains documents sont très ondulés comme le montre la figure 1.7a où le document présente des taches d'humidités.

– Les **déchirures et trous** (figure 1.7e) sont des dégradations plus rares qui peuvent avoir des origines diverses (usure, mauvaise utilisation, insectes, etc.).

1.2.2 Les dégradations dues à la numérisation

Dans cette sous-section nous présentons des défauts issus d'une mauvaise numérisation. On parle de mauvaise numérisation quand l'image de document résultante n'est pas suffisamment fidèle à la page originale. Cela peut provenir de mauvais réglages (focale, balance des blancs, réglage de la couleur, ...), de mauvaises manipulations (mouvements de la page pendant sa numérisation, mauvais placement de la vitre, ...), d'une évolution des paramètres extérieurs au scanner (changement de luminosité de la pièce par exemple) ou d'une mauvaise adéquation entre le capacité du scanner et le type du document. Voici une liste non exhaustive de ce type de dégradations :



(a) Une feuille noire est placée entre le verso et la page suivante

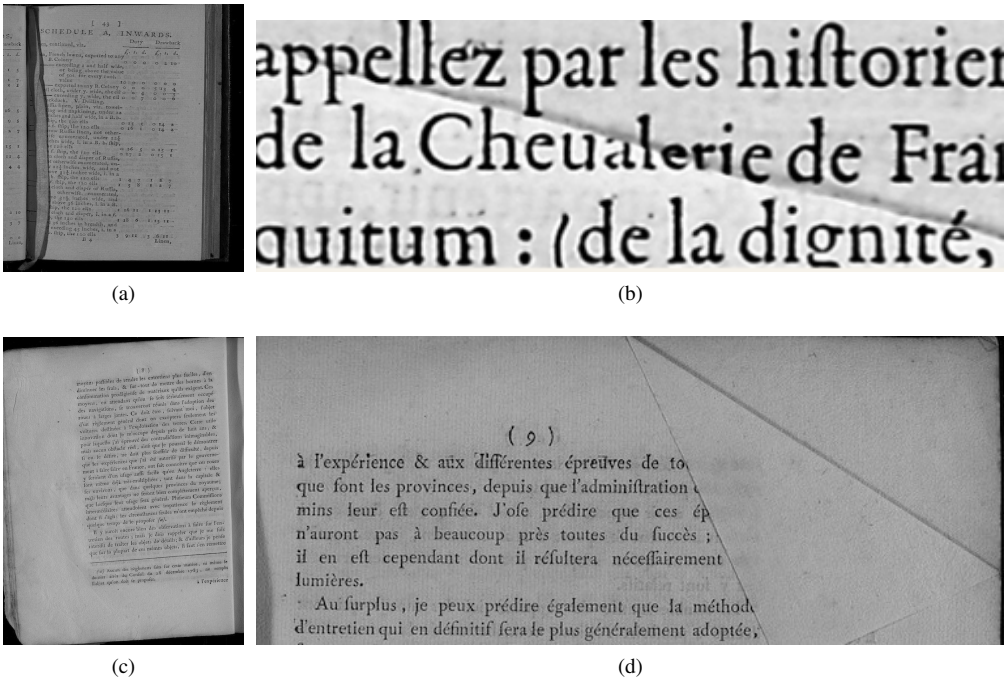
(b) Une feuille blanche est placée entre le verso et la page suivante

FIGURE *1.9 – Certains paramètres ou réglages du scanner peuvent amplifier certaines dégradations. Ici nous remarquons que l'utilisation d'une feuille blanche entre le verso et la page suivante rend la transparence bien plus visible.*

– L'**ouverture du document** est l'angle maximal entre deux pages. Lorsque la reliure est trop serrée, il

est alors difficile de mettre le document à plat. Ceci peut alors poser des problèmes de mise au point et de zone noire dans la reliure. Il existe aussi des scanners capables de descendre dans les marges du document.

– Le **flou** peut être le résultat de plusieurs mauvaises opérations ou réglages. Par exemple, c'est l'opérateur de numérisation qui réalise le réglage du focus sur les scanners manuels. Or, certains livres anciens sont très volumineux et le scanner doit être réajusté au bout d'un certain nombre de pages numérisées. Le défaut de flou est présent sur toute la page. Dans d'autres cas, le flou est local : reliure trop profonde, mouvements de l'ouvrage pendant la numérisation (figure 1.5b),. . . .
– La **mauvaise colorimétrie** d'un document provient d'un mauvais réglage du scanner (balance des blancs, mire couleur). Ces réglages doivent être souvent réajustés pour s'adapter aux changements extérieurs, par exemple le changement de lumière ambiante.
– Des **défauts chromatiques** se créent souvent autour des pixels d'encre et sont le résultat d'une mauvaise synchronisation des capteurs colorimétriques.
– Le **bruit** est un signal aléatoire causé par le capteur du scanner. Le bruit numérique peut avoir plusieurs origines. Parmi celles-ci on peut citer le bruit thermique qui augmente avec la température du capteur et le bruit *poivre et sel*, conséquence d'erreurs de transmission des données ou de la défaillance de certains composants des capteurs CCD.
– L'**orientation** du document peut ne pas être respectée. En effet, il peut arriver que l'opérateur fasse tourner le document.
– L'**illumination non uniforme** est une caractéristique obtenue lorsque la reliure du document est profonde ou que le document est gondolé. L'illumination n'est plus uniforme et des ombres se créent.
– **Informations manquantes**. Certains documents contiennent des informations sur plusieurs couches par exemple les pages à déplier ou contenants des "post-its". Il arrive que l'opérateur oublie de numériser ces informations.
– Les **défauts existants** peuvent aussi être amplifiés par certains réglages du scanner. Par exemple, la transparence du verso est bien plus visible si l'on place une feuille blanche entre la page à numériser et la page suivante comme le montre la figure 1.9b. À l'inverse, si on place une feuille sombre, la transparence est atténuée (figure 1.9a).



(a) (b)

(c) (d)

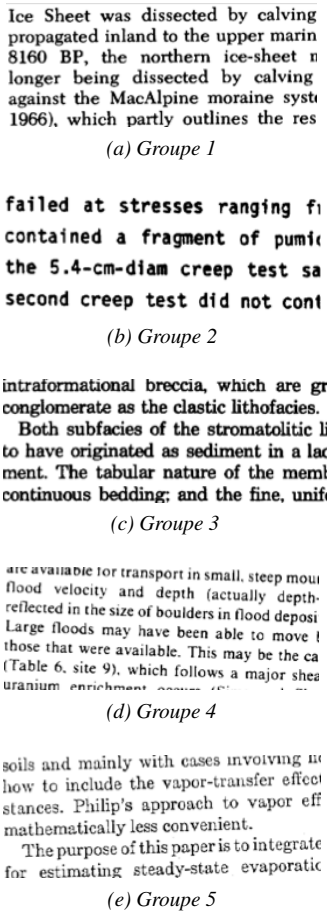
FIGURE 1.10 – *Exemples de défauts engendrés par la numérisation :* **??** *, l'opérateur a oublié d'enlever le marque-page, (b) montre une page qui aurait pu être dépliée par l'opérateur. (c) le document est mal orienté, (d) l'opérateur aurait dû décorner le document.*

1.3 Relation entre la qualité d'une image de document et les performances d'algorithmes

Si la structure, la complexité ou la variabilité des documents sont des facteurs majeurs de la baisse des performances des algorithmes [RKN93, Ric93, RJN96], nous montrerons dans cette section que les dégradations d'une page peuvent également être à l'origine d'un grand nombre d'erreurs. Nous commencerons par faire une étude de l'influence des dégradations sur la globalité de la chaîne, puis nous étudierons leurs influences sur chacun des maillons.

1.3.1 Études sur la chaîne globale de traitement

Certains systèmes doivent être considérés comme une seule et même entité. En effet, certains algorithmes sont des systèmes complexes qui incluent un ensemble d'algorithmes : prétraitements, binarisation, extraction de bordures, extraction de structure physique et logique, ainsi que des moteurs de reconnaissance de caractères. Étant donné que chaque étape est connue, ces systèmes peuvent se servir de différents résultats intermédiaires pour améliorer les résultats finaux. C'est par exemple le cas des OCRs tel que Abbyy [Jac] et OCROpus [Bre08] qui en combinant plusieurs méthodes de binarisation peuvent obtenir de meilleurs résultats. D'autres algorithms proposent des phases de prétraitements (filtrage des composantes connexes ayant des statistiques comme leurs tailles, aires ou compacités aberrantes au regard des autres) afin de limiter leurs propres erreurs.


Ice Sheet was dissected by calving
propagated inland to the upper marin
8160 BP, the northern ice-sheet n
longer being dissected by calving
against the MacAlpine moraine syst
1966), which partly outlines the res

(a) Groupe 1

failed at stresses ranging fr
contained a fragment of pumic
the 5.4-cm-diam creep test sa
second creep test did not cont

(b) Groupe 2

intraformational breccia, which are gr
conglomerate as the clastic lithofacies.
Both subfacies of the stromatolitic li
to have originated as sediment in a lac
ment. The tabular nature of the meml
continuous bedding; and the fine, unif

(c) Groupe 3

are available for transport in small, steep mou
flood velocity and depth (actually depth-
reflected in the size of boulders in flood deposi
Large floods may have been able to move
those that were available. This may be the ca
(Table 6, site 9), which follows a major shea
uranium enrichment

(d) Groupe 4

soils and mainly with cases involving n
how to include the vapor-transfer effect
stances. Philip's approach to vapor eff
mathematically less convenient.
The purpose of this paper is to integrate
for estimating steady-state evaporatic

(e) Groupe 5

FIGURE *1.11 – Groupes de qualité de l'étude [RKN93].*

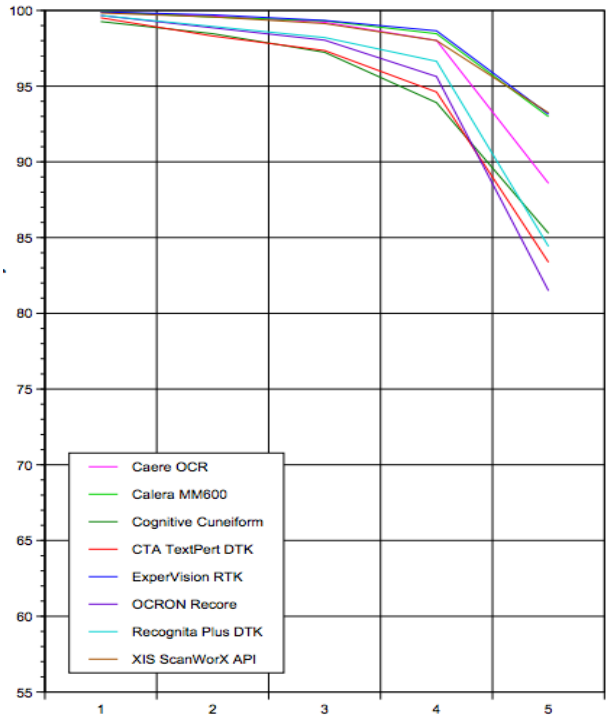


FIGURE 1.12 – *Étude sur la performance des OCRs en fonction de la qualité de la page [RKN93]. En ordonnée, la précision de l'OCR. En abscisse, le groupe de qualité de l'image (du meilleur groupe au plus mauvais - figure 1.11). On constate de façon globale que la précision de l'OCR baisse en fonction de la qualité de la page. Néanmoins, la classification sous forme de groupe ne nous permet pas de conclure sur la nature de leurs relations (linéaires ou polynomiales).*

Des études [RKN93, KVJ+12, Fra11] permettent d'avoir une idée du lien entre les dégradations d'une image de document et la performance d'un système de reconnaissance de caractères. En général ces dernières montrent que plus le document est dégradé plus les performances de l'OCR baissent.

La première [RKN93], date de 1993 et évalue les performances de différents OCRs. Elle analyse leurs précisions sur un corpus de documents contemporains. Ces documents sont divisés en 5 groupes en fonction de leurs qualités. Un exemple d'images de documents par groupe de qualité est présenté en figure 1.11. Les différents résultats montrent que la qualité des documents influe sur les résultats de l'OCR. En effet, on constate sur la figure 1.12 que la moyenne du taux de reconnaissance des différents OCRs chute à partir du groupe de qualité numéro 3.

La seconde étude, présentée dans [KVJ+12] montre la même relation entre la qualité des caractères et les performances de l'OCR *OCROpus* à la différence qu'elle est menée sur des documents anciens en niveaux de gris. Un modèle de dégradation de caractères est présenté afin de générer un corpus de documents semi-synthétiques avec vérité-terrain de taille importante. Ici, seules la quantité et l'importance des dégradations sur les caractères sont variabilisées. Les auteurs montrent que la qualité des caractères influe sur la performance d'*OCROpus*. Nous détaillons de façon plus précise les résultats obtenus par cette étude en sous-section 1.3.2.3.

Une autre étude [Fra11], réalisée pendant le projet IMPACT sur les performances de l'OCR en tant que système complet, présente deux expériences. La première partie de l'étude a permis de comparer les résultats d'Abbyy sur des images codées, en couleur, en niveaux de gris et en binaire. La seconde analyse la précision de l'OCR en fonction de la résolution de l'image (300dpi, 400dpi puis 500dpi). Ces expériences montrent que l'OCR à de meilleurs résultats sur les images en niveaux de gris en 400 dpi. Le passage du document à la couleur et dans une résolution supérieure à 400 dpi génère une légère baisse des résultats. Pourtant, les auteurs de l'étude font l'hypothèse que le moteur de l'OCR se base sur une image binarisée. Or, le fait que les résultats sont meilleurs en niveaux de gris montre que soit l'hypothèse est fausse, soit des traitements supplémentaires sont appliqués à l'image en niveaux de gris avant sa binarisation.

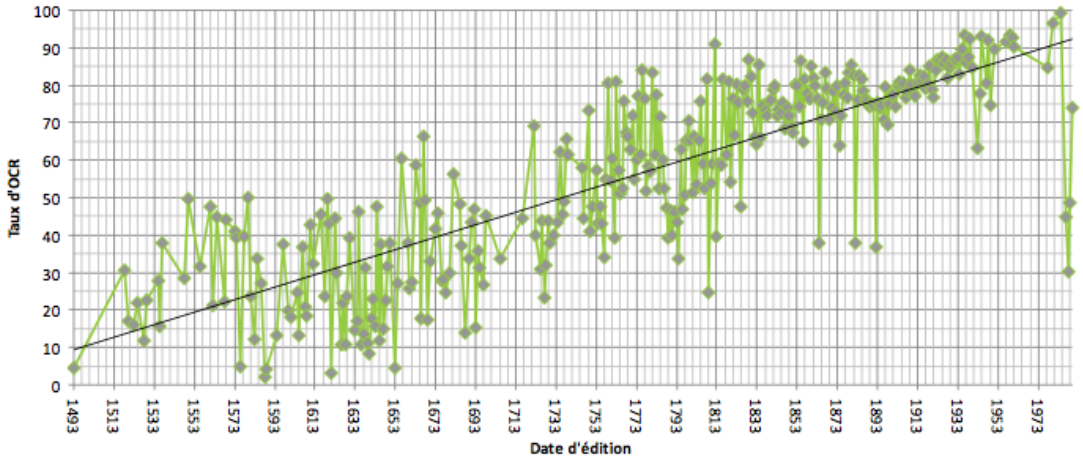
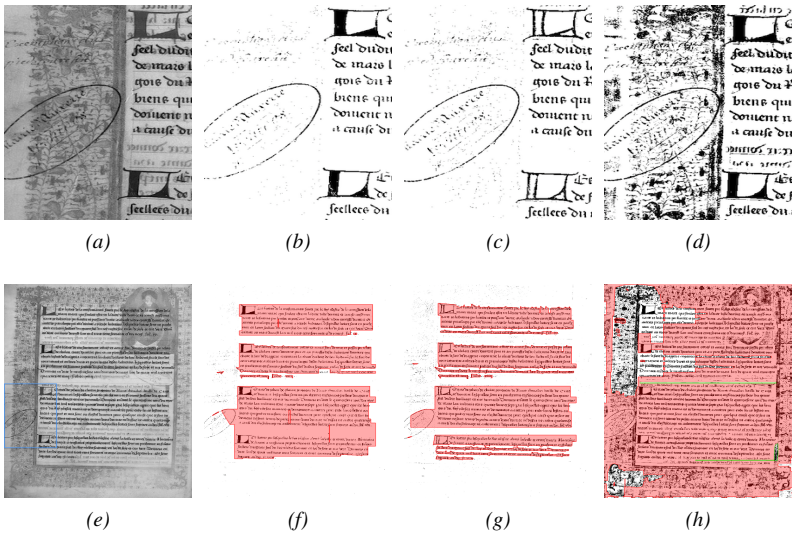


FIGURE 1.13 – *Taux de reconnaissance d'OCRs en fonction de la date d'édition (étude réalisée par la Bibliothèque Nationale de France)*

Dans une étude récente, la Bibliothèque Nationale de France montre la relation entre la date d'édition et le taux d'erreur d'un OCR. Les résultats présentés en figure 1.13 montrent bien que plus le document est ancien, plus le taux de reconnaissance est bas. La relation entre la date d'édition et la qualité d'un

document n'est pas établie dans cette étude et les baisses de performance de l'OCR peuvent provenir de la complexité (fonte, langue, structure physique) des documents anciens.

Si l'on considère la chaîne de traitements comme une série de N étapes, les résultats de l'algorithme à la position K dépendront des algorithmes précédents (de 0 à $K-1$ inclus). En effet, il est pratiquement impossible d'obtenir le meilleur résultat pour chaque algorithme. Les erreurs vont donc se cumuler rendant la tâche des algorithmes suivants de plus en plus complexe. Prenons l'exemple de la figure 1.14, sur laquelle un document ancien est binarisé. Selon l'algorithme de binarisation utilisé, certaines erreurs apparaissent. Ces erreurs ont un impact immédiat sur l'algorithme suivant (ici un algorithme d'analyse de structure physique [VD+12]). Ce même algorithme génère beaucoup plus d'erreurs avec l'image binarisée par la méthode de Bernsen [Ber86] que sur l'image binarisée par la méthode d'Otsu [Ots75].



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

(e) *(f)* *(g)* *(h)*

FIGURE *1.14 – Impact de la binarisation sur un algorithme d'extraction de structure physique (algorithme de R. Vieux [VD+12]) : a un zoom sur l'image originale (en e), b, binarisation Otsu, c, binarisation Sauvola, d, binarisation Bernsen, e, l'image originale, f, segmentation sur la binarisation d'Otsu, g, segmentation sur la binarisation de Sauvola, h, segmentation sur la binarisation de Bernsen.*

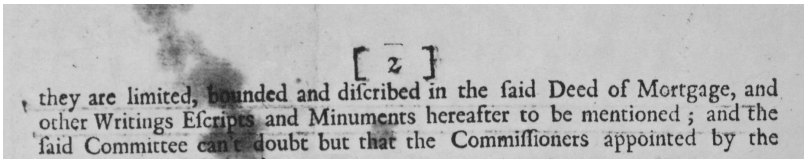
Si l'on étudie une chaîne classique de traitement, on peut se rendre compte que différentes étapes ou algorithmes sont critiques. Nous pensons que la binarisation et l'extraction de structure physique sont des étapes déterminantes dans la chaîne. La plupart des méthodes d'extraction de caractères reposent sur la binarisation. Elle doit être capable de reconnaitre les différentes dégradations de l'image afin de fournir une extraction en composantes connexes de l'encre. L'extraction de structure physique est une autre étape critique dans la chaîne de traitements. En effet, elle extrait les blocs de texte, les illustrations, les titres, les marges .... Si l'extraction de structure physique échoue, l'extraction de structure logique échouera avec une plus forte probabilité. L'OCR ne disposera pas des bons blocs de texte pour travailler ni des informations logiques liant ces blocs. De même, si les blocs sont mal découpés on peut retrouver dans un même bloc des changements de police et de fonte qui pénalisent les performances de l'OCR.

À notre connaissance, une seule étude récente [LLS11] propose d'analyser l'importance de chaque maillon dans une chaîne globale. Cette dernière étudie l'impact de différents algorithmes de binarisation, de segmentation et de reconnaissance de caractères sur la chaîne globale de traitements en testant toutes les combinaisons possibles. Les résultats montrent que l'une des méthodes de binarisation (NCI_BI) améliore de manière significative les résultats finaux de la chaîne. Néanmoins, ces résultats sont à nuancer en raison de la faible quantité d'algorithmes testés.

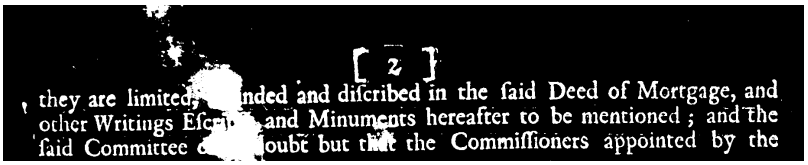
1.3.2 Impact des dégradations sur les maillons principaux d'une chaîne classique d'analyse

1.3.2.1 Les dégradations ayant une influence sur les résultats de la binarisation

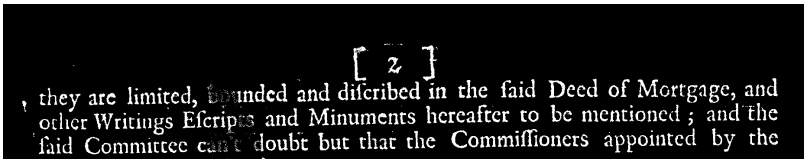
La binarisation d'un document a pour objectif une classification des pixels de l'image en deux classes : les pixels d'encre et ceux du fond. Les algorithmes de binarisation sont de manière générale sensibles aux dégradations qui perturbent la distribution des niveaux de gris. Nous appellerons ce type de dégradations les perturbations *fond-encre*. Les taches de petite ou moyenne taille, la transparence, les problèmes de luminosité, les ombres dues aux ondulations d'un document, l'usure du tampon d'impression, l'effacement de l'encre ou à l'inverse la diffusion trop importante de l'encre dans le papier sont considérés comme des perturbations *fond-encre*.



(a) Document original en niveaux de gris



(b) Document binarisé avec la méthode d'Otsu [Ots75]



(c) Document binarisé avec la méthode de Sauvola [SP00]

FIGURE 1.15 – *Exemples de résultats de binarisation sur la même image (image extraite du corpus DIBCO [GNP09])*

Nous pouvons diviser les algorithmes de binarisation en deux familles : les algorithmes qui travaillent sur l'image globale et ceux qui travaillent localement sur des parties de l'image. La binarisation est un sujet actif de la recherche en analyse de documents et une très grande quantité de méthodes ont été proposées comme le montre l'état de l'art de P. Stathis [SKP08a].

Les algorithmes globaux se basent sur différents descripteurs calculés sur la totalité de l'image. Les descripteurs et techniques sont quant à eux très variés. Certains algorithmes [RL83, Sez90, RYS95] se basent sur la forme de l'histogramme des niveaux de gris. Ces algorithmes recherchent deux pics (le pic d'encre et celui du fond) et un creux entre ces derniers. Ils utilisent des outils tels que les enveloppes convexes, l'approximation de l'histogramme à l'aide de fonctions par échelons ou encore du lissage autorégressif. D'autres algorithmes [Ots75, Llo85, KI86, KI85, SSW88] utilisent l'histogramme des niveaux de gris pour faire de la classification à deux classes. La plupart des méthodes modélisent l'histogramme comme le mélange de deux gaussiennes afin de calculer le seuil optimal qui sépare ces deux classes. Certaines méthodes [KSW85, LL93, LT98, Sha94, SWY97, Pal96] utilisent l'entropie de la distribution des niveaux de gris de l'image.

Ces familles d'algorithmes sont sensibles aux perturbations fond-encre modifiant la distribution des niveaux de gris de l'image, et ce de façon globale. Par exemple, si le fond n'est pas homogène (problème d'illumination dû à la reliure du document par exemple), certains pixels gris foncé de fond peuvent être classés comme pixels d'encre (figure 1.15.b). En ce qui concerne l'encre, les niveaux de gris peuvent aussi varier. Les parties claires d'un caractère peuvent par erreur être classées comme pixels de fond.

La seconde grande famille de méthode de binarisation est dite adaptative. Un seuil est calculé pour chaque pixel de l'image. Les algorithmes tels que [SP00, Ber86, WR83, Nib85] utilisent des indicateurs statistiques (moyenne [WR83, Nib85], écart-type [Nib85, SP00], ...). Ces indicateurs sont calculés sur une portion de l'image (fenêtre) centrée sur le pixel courant.
Ces algorithmes sont généralement plus robustes aux perturbations globales de la distribution des niveaux de gris. Par contre, ils sont moins robustes aux dégradations de tailles moyennes. En effet, pour trouver le seuil exact de séparation des deux classes, il faut que les deux types de pixels (encre et fond) soient présents dans la fenêtre. Il existe sur ce type d'algorithme beaucoup de configurations possibles menant soit à un échec soit à une bonne classification :

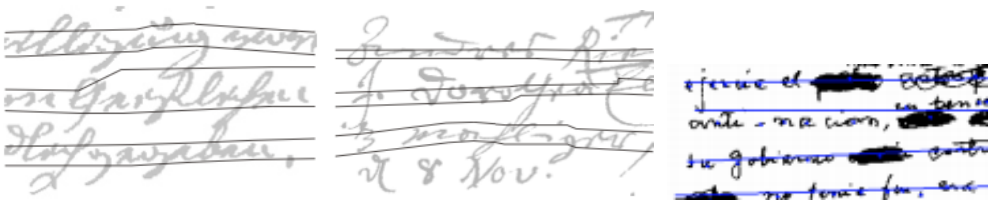
- la fenêtre peut contenir seulement de petites dégradations et des pixels d'encre. L'algorithme ne détectera pas de variations importantes entre la dégradation et l'encre et considèrera l'encre comme du fond (figure 1.15.c, grande tache).
- la fenêtre peut contenir des pixels de dégradation et de fond. La variation de niveaux de gris entre les deux types de pixels pouvant être importante, la dégradation peut être considérée comme de l'encre.
- la fenêtre peut aussi contenir une dégradation de forte intensité (claire), et des pixels d'encre. La variation d'intensité entre l'encre et la dégradation étant importante, l'algorithme peut alors correctement binariser cette partie. Par exemple, sur la figure 1.15.c, la méthode de Sauvola a correctement binarisé le mot *that* (3e ligne), la fenêtre devait en effet contenir assez d'informations pour que le mot *that* soit correctement binarisé.

Il est important de remarquer que la localisation de la dégradation et la taille de la fenêtre sont ici des caractéristiques déterminantes. La taille de la fenêtre doit être suffisamment grande pour les petites taches, mais pas trop petite pour ne pas considérer le bruit comme de l'encre. De façon générale, plus l'image de document contient du bruit (petites et moyennes composantes grises) éloigné du texte, plus les performances seront à la baisse. Ces petites dégradations sont la plupart du temps dues à la transparence du verso, aux taches d'usure, à de mauvaises conservations ou à la texture du support, etc.

1.3.2.2 Les dégradations ayant une influence sur les résultats des algorithmes d'analyse de structure du document

On distingue deux types de structures différentes [MRK03] : la structure physique et la structure logique. L'analyse de la structure physique a pour but de segmenter l'image de document en blocs de textes, illustrations, etc. Ces algorithmes reposent souvent sur des règles typographiques et typologiques [LSZT07] par exemple les alinéas, l'espace interligne, l'espace inter mots ou inter lettres. De manière générale, ce type d'algorithme est sensible à la variabilité des attributs. Ces outils sont bien adaptés pour des documents à forte structure formatée comme le sont les documents contemporains. C'est rarement le cas des documents anciens où les règles sont différentes d'un document à un autre. Par exemple, les valeurs utilisées en imprimerie sont souvent différentes de celles utilisées par les moines copistes. La figure 1.17 illustre les difficultés que rencontrent ces algorithmes. Ces algorithmes sont également sensibles à certaines dégradations, comme l'orientation du document, la courbure due à la reliure, les ondulations de la page, le chevauchement de lignes lorsque deux composantes se touchent. Ces dégradations influent directement sur les valeurs des attributs.

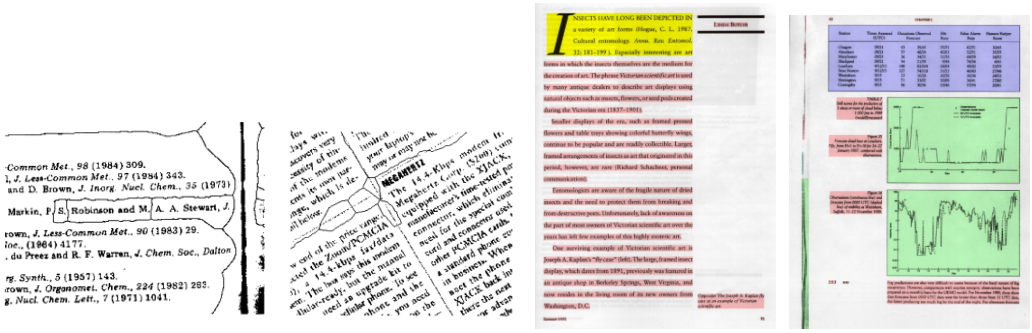
Certains algorithmes [MB01, MS99, SGS93, AK04, HD03] extraient les lignes d'un document en se basant sur les profils des projections verticales et horizontales. Ces profils sont calculés en projetant la somme des pixels le long d'un axe (x ou y) sur un histogramme. L'extraction de la ligne repose sur les caractéristiques de ces histogrammes. Une méthode similaire [LeB97] est adaptée aux images en



(a) (b) (c)

FIGURE 1.16 – *Exemples d'erreurs réalisés par différents algorithmes d'extraction de lignes : a. le centre de la seconde ligne est mal identifié dû au fait que les deux lignes se chevauchent partiellement [FT01], b. La quatrième ligne a été oubliée [FT01] c. Le centre de la seconde ligne est mal identifié dû aux différentes ratures et corrections du texte manuscrit [LSHF95] (changement de direction de la ligne).*

niveaux de gris en accumulant cette fois-ci les gradients de l'image toujours le long de l'axe horizontal. D'autres méthodes d'extraction de lignes se basent sur une analyse de la distribution des « run-length » [WCW82, SG04b]. Les pixels noirs dont la distance est inférieure à un seuil donné le long de l'axe horizontal sont agglomérés pour former les lignes de textes. La transformation de Hough [Hou62] est aussi très utilisée par certains algorithmes [LSHF95, PS99] d'extraction des lignes de textes.
Des dégradations comme la courbure (due à la reliure), l'ondulation ou l'orientation des pages diminuent les performances des algorithmes précédents. En effet, les lignes de texte ne sont plus droites et les profils ne peuvent plus être utilisés pour détecter les lignes. Dans le cas de la reliure et de l'ondulation, l'erreur est locale à la dégradation et l'algorithme peut fonctionner correctement sur le reste du document.



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

FIGURE 1.17 – *Erreurs d'algorithmes d'extraction de structures physiques : a. la segmentation est sensible à l'espace entre les mots qui varie dans le paragraphe[KSI98]. b. un trait séparant les blocs de textes mène à une sur segmentation[KSI98], c. le* I *majuscule engendre une erreur de classification, d.[JY98] la reliure a été classée comme illustration[JY98] .*

Certaines méthodes [LSF94] tentent de regrouper les éléments structurels d'un document avec une approche ascendante. Les approches ascendantes partent des pixels puis les regroupent en composantes connexes pour former les caractères. Ces caractères sont ensuite regroupés en mots puis en lignes pour finir par constituer des paragraphes. Parmi ces algorithmes, nous pouvons citer [O'G93] et [KSI98] qui se basent sur l'algorithme de Voronoi, [WWC82] qui utilise les *run-lenght* [JY98] et [FK88] qui est basé sur un algorithme de séparation de chaines de caractères. Dans [FT01] les éléments de base sont les composantes connexes, elles sont au fur et à mesure regroupées en suivant certaines directions (0°,

45°, 90°, 125°). Une approximation de la ligne avec deux courbes (baseline, x-height line) est réalisée dans le but d'identifier le centre des lignes.
Les approches descendantes, à l'inverse, partent du document dans son ensemble, le découpent de façon itérative en un ensemble d'éléments de plus petites tailles (pages, paragraphes, lignes, mots, lettres). La découpe s'arrête en fonction d'un critère permettant d'établir la segmentation finale. Parmi ces méthodes on peut citer [NSV92] basée sur un algorithme de type *X-Y-cut* et [BJF90] basée sur une couverture orientée forme (shape-directed-covers-based).
Ces méthodes sont en général plus robustes à la courbure des lignes ou aux changements de direction. Par contre, elles sont sensibles à des dégradations comme le chevauchement des lignes (figure 1.16.a)

On remarque ici aussi que certaines dégradations peuvent influencer les algorithmes tant globalement que localement. Par exemple, l'orientation du document est une dégradation globale tandis que le chevauchement de certaines lignes venant du simple fait que deux composantes se touchent est une dégradation locale qui n'influencera le résultat de l'analyse que sur une partie du document.

L'analyse de la structure logique d'un document consiste à labelliser les différentes zones obtenues lors de la segmentation physique (lettres, mots, paragraphes, illustrations, etc.). Afin de labelliser ces blocs, certains algorithmes se basent sur un ensemble de règles ([NS95],[TA90], [YATT91], [Fis91], [DP91]) ou des grammaires formelles ([KNSV93], [IA91], [TI94]). Cette catégorie d'algorithmes repose essentiellement sur les performances de l'algorithme d'extraction de la structure physique. Ainsi, les dégradations ayant une influence sur l'analyse de structure physique auront également une influence sur les algorithmes d'analyse de structure logique.

1.3.2.3 Problèmes pour la reconnaissance de caractères

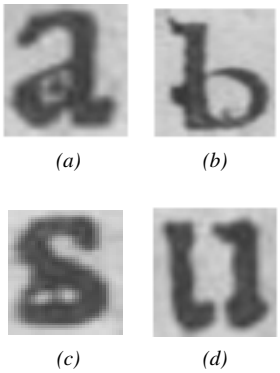
La reconnaissance de caractères se fait en plusieurs étapes : l'extraction de composantes connexes qui constituent les lettres, le calcul de différents descripteurs (compacité, moments de Zernike, ...) sur chacune des composantes, puis le calcul de similarité (basé sur les descripteurs) avec un ensemble de lettres de référence.
Ces systèmes sont sensibles à la complexité des lettres du document : ils doivent être entraînés sur un corpus d'apprentissage assez représentatif et utiliser des descripteurs robustes afin de ne pas confondre certains symboles de formes similaires (par exemple, dans la police fraktur le ſ (f) et le ſ (s) sont très proches, ou dans d'autres polices le O le 0 et le Q).

Les algorithmes de reconnaissance de caractères sont aussi sensibles aux dégradations que subissent les caractères : encre effacée ou qui à l'inverse bave, caractères cassés, caractères qui se touchent, ...
Une étude proposée dans [KVJ+12] propose d'analyser la distance entre la transcription d'une image propre non dégradée et la transcription d'une image dont les caractères sont de plus en plus dégradés. Le résultat (figures 1.19 et 1.18) montre que plus l'image de document possède des caractères dégradés, plus le taux d'erreur OCR augmente. On remarque que la nature de la courbe n'est pas linéaire : les taux restent relativement stables pendant un moment avant de connaitre une augmentation très forte. La nature de cette courbe montre la capacité de l'algorithme à s'autocorriger (via un dictionnaire par exemple).

1.4 Évaluer la qualité dans le but de prédire les performances d'algorithmes ou de sélectionner automatiquement le meilleur

La numérisation est perçue comme une manière de préserver notre patrimoine culturel. Il est crucial de pouvoir **garantir la qualité** des images numérisées en utilisant des algorithmes. Le projet de recherche DIGIDOC (Document Image diGitisation with Interactive DescriptiOn Capability) [13], étroitement lié à

13. http://digidoc.labri.fr/pmwiki.php ; https://bitbucket.org/digidoc



(a) *(b)*

(c) *(d)*

FIGURE *1.18 – Exemples de caractères dégradés :a. b. c. encre qui* bave*, d. caractère cassé.*

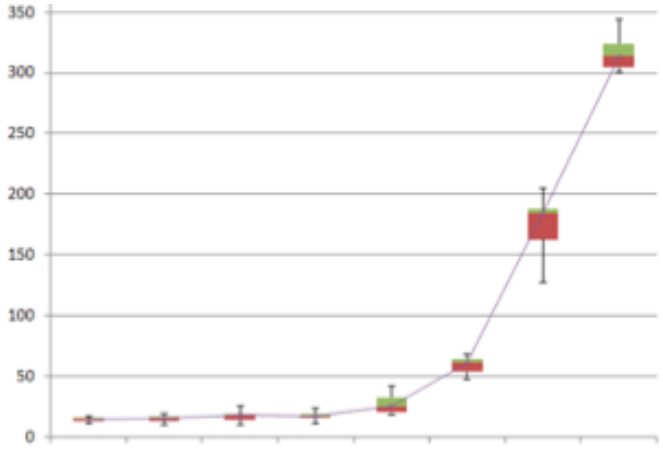


FIGURE 1.19 – *Étude réalisée dans [KVJ+12] montrant la relation entre la dégradation de caractères et la précision d'un système de reconnaissance de caractères.*

cette thèse, a en partie pour but de résoudre ce type de problèmes.

Comme nous l'avons montré précédemment, la complexité, et la qualité des documents anciens impactent sérieusement sur les performances de la chaîne de traitement que ce soit de manière globale sur l'intégralité de la chaîne ou de manière locale sur un des maillons de cette chaîne.
Par exemple, comme illustré sur la figure 1.13 un OCR peut avoir des taux de reconnaissance qui avoisinent seulement les 50%. Ces mauvais résultats ont des conséquences financières importantes, car il est nécessaire de corriger manuellement les erreurs produites par l'OCR. Dans cette configuration précise, il est plus rentable de retranscrire directement tout le document plutôt que de corriger les erreurs de l'OCR. Cet exemple montre clairement l'intérêt de pouvoir prédire globalement les résultats de l'OCR sur ce type de documents afin de choisir la stratégie la plus productive.

Différentes dégradations peuvent coexister au sein d'un même document, ce qui rend difficile la mise en œuvre des algorithmes de restauration. Par exemple, il est fréquent qu'un document requière un algorithme de correction de la transparence ou de la courbure, de réorientation de la page et de binarisation, etc. Cette combinaison de traitements est laissée à la responsabilité de l'expert qui se base sur son expérience. Même s'il a toutes les compétences requises, il lui est difficile d'appréhender les conséquences des restaurations successives sur la globalité des traitements. La plupart du temps, il se base sur la qualité visuelle qui correspond à un usage d'archivage, mais qui ne correspond pas toujours aux attentes de l'OCR. Une approche plus rationnelle serait de conditionner la chaîne de restauration en fonction de l'usage attendu, ce conditionnement consisterait à sélectionner les algorithmes conduisant à une optimisation des résultats pour l'usage retenu.

Cette thèse propose une approche différente permettant de résoudre en partie ces problèmes. La qualité de l'image n'est pas corrigée, mais évaluée dans le but de **prédire les performances des algorithmes**. Cette prédiction peut se placer à plusieurs niveaux dans la chaîne de traitements :
– au début, pour identifier le type de dégradations présentes sur le document,
– avant la chaîne de traitement pour prédire le résultat final d'un ensemble d'algorithmes,
– dans la chaîne de traitement et à la sortie d'un algorithme afin de prédire les performances de l'algorithme suivant. Il devient alors envisageable de créer des chaînes de traitement dynamiques optimisées pour chaque image.

L'évaluation de la qualité d'une image se fait à l'aide de **différents descripteurs**. Dans le chapitre suivant, nous présenterons des descripteurs qui se basent à la fois sur des **informations globales et**

locales. En effet, nous avons montré que certaines dégradations provoquent des erreurs très localisées (par exemple, sur un caractère ou une zone en particulier de l'image) tandis que d'autres entrainent des erreurs globales (par exemple l'extraction de lignes par une méthode utilisant les profils sur un document mal orienté). Dans le troisième chapitre, ces descripteurs seront entrainés sur des corpus de documents dans le but de créer des modèles de prédiction (chaque algorithme aura son propre modèle prédictif).

Chapitre 2

Évaluation de la qualité par des descripteurs

L'un des sujets de cette thèse porte sur la définition de descripteurs permettant de caractériser la qualité d'un document. Ces descripteurs peuvent ensuite être utilisés pour prédire les performances de différents algorithmes de traitement ou pour sélectionner l'algorithme donnant les meilleurs résultats pour une image donnée. Ce second chapitre est consacré à l'étude et à la définition de ces descripteurs.

La première section (2.1) de ce chapitre commence par un rapide état de l'art sur les descripteurs les plus usuels. Cet état de l'art a pour objectif d'identifier les descripteurs ou méthodes pouvant éventuellement être utilisés pour la création de descripteurs orientés qualité. Dans une première partie, nous aborderons les descripteurs utilisés dans le cadre des images naturelles (sous-section 2.1.2), puis ceux proposés pour la description d'images de documents binaires. Une deuxième partie de cette section sera consacrée aux méthodes de restauration ou de modélisation de dégradations (sous-section 2.1.3). Bien que les méthodes de restauration et de modélisation de dégradations n'utilisent pas toujours des descripteurs, elles proposent cependant une caractérisation de plus ou moins bas niveau de certaines dégradations (par exemple les profils horizontaux et verticaux permettant de redresser un document). Cette caractérisation peut servir à la création de descripteurs pour la qualité d'une image de document. Enfin, basées sur les analyses précédentes, nous proposerons une méthodologie permettant de créer des descripteurs qualité et reposant sur l'analyse des dégradations d'une image en fonction de leurs influences sur des algorithmes.

La seconde section (2.2) détaillera de nouveaux descripteurs dédiés à la caractérisation des perturbations fond-encre. Nous les utiliserons pour montrer leurs intérêts dans le cadre des algorithmes de binarisation qui sont fortement dépendants de la qualité de la séparation fond-encre. Ces descripteurs sont calculés à partir d'une séparation en couches d'informations de l'image de document.

La dernière section (2.3) propose d'illustrer l'intérêt de notre méthodologie au cas de la caractérisation de la transparence.

2.1 Vers des descripteurs de qualité d'une image de document

Dans cette section, nous présenterons rapidement les techniques, méthodes et descripteurs existants permettant soit de caractériser le contenu d'une image, soit d'évaluer directement la qualité d'une image. Ainsi, une première partie est consacrée aux mesures de la qualité d'images naturelles compressées. Une deuxième partie présente les descripteurs les plus usuels permettant de caractériser le contenu d'une image naturelle. Nous nous intéresserons ensuite aux images de documents en présentant tout d'abord un ensemble de descripteurs utilisés pour la caractérisation d'images de documents binaires, puis aux méthodes de modélisation et de restauration de dégradations.

2.1.1 Mesurer la qualité d'une image à l'aide d'une image de référence

Un grand nombre de descripteurs [SB09] ont été créés pour évaluer la qualité d'une image naturelle en reposant sur une comparaison de l'image à mesurer avec une image de référence. Les plus courants, tels que le SNR (signal to noise ratio), le PSNR (peak signal to noise ratio) [HTG08] ou SSIM (Structural SIMilarity index) [WBSS04] sont dans la plupart des cas utilisés dans l'objectif de mesurer la dégradation engendrée par l'application d'algorithmes de compression.

Ces descripteurs peuvent être utilisés pour mesurer la qualité d'images de documents compressées. Dans [SB09], SSIM est mis en corrélation avec le taux d'erreur OCR d'images de plus en plus compressées. Les résultats, présentés en figure 2.1 montrent une corrélation forte entre les deux mesures. Néanmoins, on constate que le taux de reconnaissance OCR chute sur des niveaux de compression JPEG "moyens" (facteur de compression 65) alors que le descripteur SSIM reste en progression. SSIM est bien plus corrélé à la compression JPEG qu'au taux de reconnaissance OCR. Cela peut s'expliquer par le fait que l'OCR est un système complexe qui peut réagir différemment selon les pixels ou les caractères affectés par la compression JPEG.

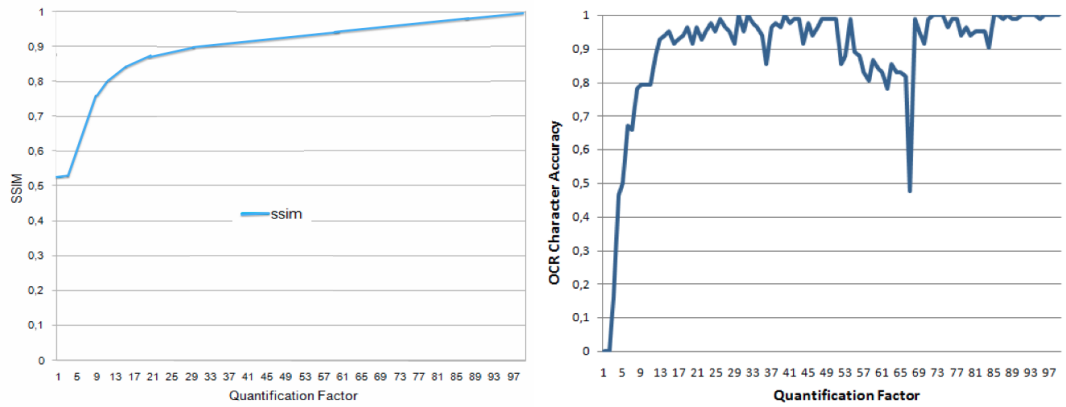


FIGURE 2.1 – *Étude proposée dans [SB09] de la corrélation entre SSIM [WBSS04], le ratio de compression JPEG (QuantificationFactor), et le taux de reconnaissance OCR. Même si les deux mesures sont corrélées, SSIM ne suffit pas à réellement expliquer les erreurs de l'OCR.*

Bien que ces descripteurs puissent être améliorés afin de mieux expliquer les résultats de l'OCR, nous nous concentrerons dans cette thèse sur l'évaluation de qualité d'images de documents sans images de référence. Certains descripteurs [TCC04] tels que le PSNR ont été adaptés pour s'affranchir de la contrainte à utiliser une image référence, mais ils restent dédiés à l'évaluation d'images compressées. Nous pensons qu'il est nécessaire de créer d'autres descripteurs dédiés aux documents pour proposer des modèles de prédiction précis.

2.1.2 Les descripteurs génériques

Depuis l'émergence du domaine de la vision par ordinateur, un grand nombre de descripteurs du contenu d'une image a été proposé [SWS+00]. Ces descripteurs sont très utilisés dans les applications du type extraction d'images par contenu (CBIR), de classification d'images, ou d'extraction d'objets dans une image. Dans cette section nous présenterons différents descripteurs classés en 4 catégories : couleur, texture, forme et points d'intérêts. La plupart des descripteurs présentés font partie de la norme MPEG7 [MKP02].

2.1.2.1 Descripteurs couleur

La couleur est une information très caractéristique d'une image ou d'un objet contenu dans une image. Un des descripteurs couleur très utilisés est celui relatif à la distribution globale des couleurs dans l'image. L'histogramme propose une caractérisation complète de cette distribution. Une autre représentation plus compacte des caractéristiques de la distribution des couleurs se base sur les moments statistiques. Cette représentation appelée moments colorimétriques décrit elle aussi les caractéristiques de la distribution. Le moment du premier ordre correspond à la moyenne, le second ordre à la variance et le troisième ordre à la *skewness* (mesure d'asymétrie de la distribution). Les moments colorimétriques sont souvent utilisés pour caractériser la couleur de façon compacte. En effet, seulement 9 valeurs sont utilisées (3 pour une image en niveaux de gris) ce qui permet d'avoir un codage compressé de la distribution des couleurs.

La norme MPEG7 propose l'utilisation, d'autres descripteurs : *Dominant Color Descriptor* [DMK+01], Scalable Color Descriptor [Cie01], Color Structure Descriptor [MBE01] et Color Layout Descriptor [KY01]. Ces descripteurs permettent de caractériser d'une certaine manière le contenu de l'image. Par exemple la couleur dominante *Dominant Color Descriptor* permet de distinguer une image en intérieur d'une image extérieur. Les différents descripteurs couleur sont associés à différents espaces colorimétriques (RGB, HSV, [SCB87] etc.). Dans le domaine du document, la plupart des traitements (segmentation, extraction de blocs, etc.) s'effectuent sur une image en niveau de gris ou binarisée. Seuls les usages liés à l'archivage considèrent la qualité colorimétrique des documents numérisés.
De fait, les descripteurs couleur MPEG7 ne correspondent pas à nos besoins. Bien évidemment, les descripteurs représentant la distribution des niveaux de gris tel que l'histogramme et les moments colorimétries seront considérés. Ces derniers sont, par exemple, très utilisés par les méthodes de binarisations. Par exemple, la méthode d'Otsu utilise l'histogramme de la distribution des niveaux de gris pour effectuer son seuillage.

2.1.2.2 Descripteurs texture

En analyse d'image, la texture se définit par la répétition d'un motif local dans une image. Les descripteurs de textures permettent de caractériser une texture en terme de rugosité, de répétitivité, direction et régularité. Les descripteurs textures proposés dans la norme MPEG7 sont les suivants [WRWC01] :
– Texture Browsing Descriptor [MOVY01], est un descripteur à cinq valeurs permettant de caractériser la régularité, la direction et la granularité d'une texture. Ce descripteur n'est pas invariant à la rotation ou au changement d'échelle.
– Homogeneous Texture Descriptor [RKK+01], fournit une représentation quantitative de la texture en se basant sur des analyses statistiques fréquentielles et locales de l'image. L'image est en effet partitionnée en suivant une découpe angulaire et radiale. HTD est souvent utilisé pour la mise en correspondance d'images en se basant sur leur similarité.
– Edge Histogram Descriptor [WPP02], encode la distribution des orientations des contours des régions d'une image. Ce descripteur est particulièrement utilisé pour faire correspondre des régions à texture non uniforme.

D'autres descripteurs non inclus dans MPEG7 sont aussi utilisés pour caractériser la texture d'une région. Un état de l'art plus complet est proposé par C. Chen dans [Che10]. Les algorithmes d'analyse et de traitement d'images de documents utilisent le plus souvent les suivants :
– Les *banques de filtres* [Sch01] sont utilisées pour caractériser la texture avec une approche statistique. L'image est convoluée avec plusieurs filtres à différentes échelles et orientations, la texture est caractérisée par la distribution des réponses du filtre.
– Local Binary Patterns [GRSR] sont des descripteurs multirésolution invariants en échelle et en rotation. Ils sont principalement utilisés pour la classification de textures.
– Les Filtres de Gabor [FS89] sont des techniques de filtrage permettant de décrire des textures localisées en fréquences et en orientation.

2.1.2.3 Descripteurs géométriques

Les descripteurs géométriques caractérisent la frontière ou la surface d'une région extraite de l'image. L'image doit donc être préalablement segmentée ou binarisée. La norme MPEG7 propose deux descripteurs géométriques [Bob01].

- Curvature Scale Space est un descripteur de forme basé sur les contours d'une région. Il est invariant en rotation et translation. L'idée de ce descripteur est qu'un contour peut être caractérisé par ses points d'inflexion. Le descripteur décrit l'évolution d'un contour lorsqu'il est de plus en plus convolué avec une gaussienne jusqu'à obtenir sa convexité.
- Angular Radial Transform est un descripteur de forme caractérisant le contenu de la région. Il se base sur une transformation angulaire radiale.

Ces deux derniers descripteurs sont rarement utilisés pour l'analyse d'images de documents. Classiquement les moments géométriques, eux aussi basés sur les moments statistiques, permettent d'obtenir certaines caractéristiques sur les formes (boites englobantes orientées, compacité, aire). D'autres caractéristiques comme le périmètre, la distribution des tangentes, l'enveloppe convexe, sont également fréquemment utilisées [CKLY93]. Ces descripteurs donnent plutôt une caractérisation globale de la forme. Dans le cadre de la reconnaissance des caractères, [BS07, TJT96] ou de symbole [HN04, VTRP08], ils sont souvent complétés par des descripteurs plus précis comme les descripteurs de Zernike[KH90] ou le résultat de transformations (le plus souvent Fourier et Fourier-Mellin). Un état de l'art plus complet est présent dans [LVSM02].

2.1.2.4 Descripteurs calculant des points d'intérêts

Ce type de descripteurs permet d'extraire un ensemble de points d'intérêts dans une image. Les points d'intérêts sont associés à une discontinuité locale importante. Les auteurs de [HS88], sont les premiers à utiliser se type de descripteurs pour caractériser les images naturelles en recherchant des points d'intérêts caractéristiques dans l'image (régions à textures ou caractéristiques isolées). Ces descripteurs sont ensuite utilisés dans [SM97] pour la reconnaissance d'image. Dans [Low04, Low99] , les descripteurs SIFT sont introduits et deviennent très utilisés. D'autres descripteurs s'en inspirent comme GLOH, PCA-SIFT, et SURF. Une évaluation de ces différents descripteurs est proposée dans [MS05].

Dans le cadre du document, largement inspiré par la reconnaissance d'objets dans les images ou vidéos, cette famille descripteurs est intensément utilisée pour la reconnaissance de symbole ou de logo [RL09, PAK10].

Dans le contexte de prédiction d'algorithmes, les auteurs de [KAM12] proposent d'utiliser les descripteurs SIFT et LBP (Local Binary Pattern) afin de prédire les performances d'OCRs. La capacité de ces descripteurs à pouvoir être utilisés sur des images en niveaux de gris permet aux auteurs de s'affranchir de l'étape de binarisation de l'image de document qui est souvent propre à chaque OCR.

Bien que cette approche semble intéressante, elle a besoin d'être affinée et complétée par d'autres descripteurs, car, à l'heure actuelle, les résultats prédits sont relativement imprécis.

2.1.3 Les descripteurs liés à la qualité d'une image de document

Dans cette sous-section, nous listerons d'abord un ensemble de descripteurs utilisés pour caractériser la qualité d'images binaires. Ces descripteurs sont utilisés pour sélectionner ou prédire des algorithmes de traitements d'images de documents (OCR, Restauration, etc.). Ensuite, nous essaierons d'identifier de nouveaux descripteurs ou d'utiliser certaines techniques afin de caractériser les différentes dégradations présentes sur un document. Cette partie se basera sur une analyse des méthodes existantes de restauration et de modélisation des dégradations.

2.1.3.1 Descripteurs pour la qualité d'images de documents binaires

Les premiers descripteurs de qualité d'images de documents sont introduits en 1995 [BKN95] et ont pour objectifs de prédire les résultats d'OCRs.
Dans cet article, les auteurs se concentrent sur des images binaires contenant du texte en analysant les composantes connexes noires et blanches de la page. Leur objectif est de classer un document en fonction d'une prédiction des résultats d'un OCR. Une première classe représente des documents pour lesquels les taux de reconnaissance seront bons. L'autre classe représentant le complémentaire. Cette classification est principalement basée sur le seuillage des descripteurs. Les descripteurs introduits sont le White Speckle Factor (WSF) et le Broken Character Factor (BCF).

(a) (b)

FIGURE 2.2 – *(a) l'encre trop diffusée dans le papier rempli les* trous *de la lettre* e. *(b) des caractères ayant subi une perte de connectivité.*

Le descripteur WSF se base sur l'hypothèse suivante : si l'encre du document est trop diffusée dans le papier, les caractères peuvent se toucher et les trous dans les lettres telles que *le a* et *le e* peuvent se combler comme le montre la figure 2.2a où le "trou" du dernier "e" est comblé. Le descripteur WSF mesure le nombre de composantes blanches dont l'aire est inférieure à un seuil epsilon (fixé à 9 pixels par les auteurs, mais qui doit être adapté à la résolution de l'image) :

$$WSF = \frac{\|\text{boites englobantes des composantes blanches dont l'aire est inférieure à epsilon pixels}\|}{\|\text{boites englobantes des composantes blanches}\|}$$

Le descripteur BCF quant à lui mesure la quantité de caractères cassés dans la page. Des exemples de caractères cassés sont présents en figure 2.2b. Pour cela, il mesure le nombre de composantes noires dont la hauteur et la largeur sont inférieures à 75% de la hauteur et de la largeur moyenne des composantes noires.

$$BCF = \frac{\|\text{composantes noires (taille < 0.75 (moyenne(tailles composantes noires)))}\|}{\|\text{composantes noires}\|}$$

En 1997 deux nouvelles mesures sont introduites [CHKW97] afin de mesurer les caractères qui se touchent (TCF) et les petites taches (small speckles) (SSF). La définition de ces descripteurs se base sur l'analyse de l'histogramme des aires des composantes connexes noires d'une image de document. Comme le montre la figure 2.3, cet histogramme contient plusieurs groupes déterminants : le premier regroupe toutes les petites taches (< 50 px sur la figure 2.3), le second les caractères non dégradés (> 50 et < 300 px sur la figure 2.3) et le troisième les caractères qui se touchent (> 300 px sur la figure 2.3). Les auteurs remarquent que plus les caractères du document sont dégradés, plus le nombre de composantes du premier (petites taches) et troisième (caractères qui se touchent) groupe augmente et par conséquent celui du second (caractères non dégradés) diminue. Il est bien évident que les seuils utilisés par les auteurs doivent être adaptés à la résolution des images actuelles.

Partant de ce constat, les auteurs définissent TCF comme le ratio entre le nombre de composantes qui sont basses et longues (caractères qui se touchent) et le nombre total de composantes. Selon les auteurs, pour qu'une composante connexe soit considérée comme basse et longue, il faut que son ratio hauteur-largeur soit inférieur à 0.75. Le calcul de SSF se base sur l'histogramme des aires des composantes connexes. SSF mesure le nombre de composantes connexes comprises dans le premier groupe. Les

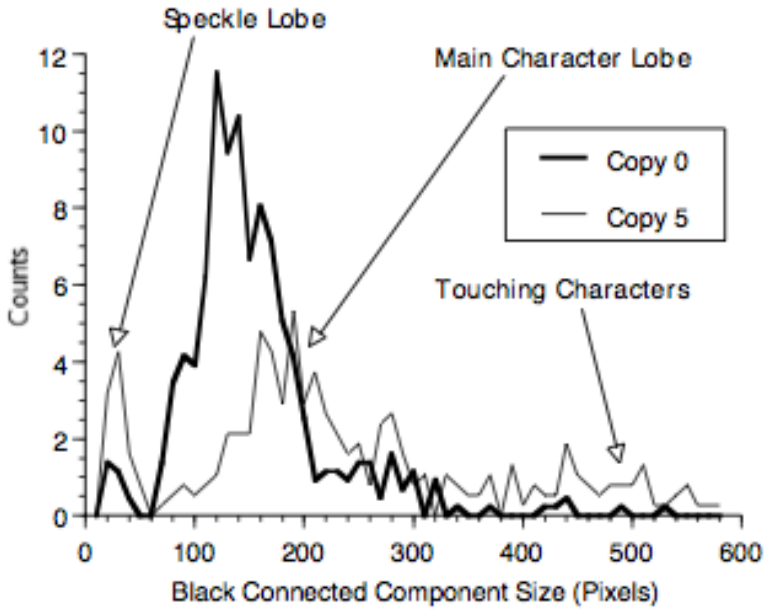


FIGURE 2.3 – *Histogrammes du nombre de composantes connexes en fonction de leurs tailles [CHKW97]. La courbe noire en gras correspond à un document non dégradé. La courbe grise correspond à un document dégradé. On remarque que, plus le document est dégradé, plus le nombre de composantes de faible taille augmente (caractères fragmentés), celui de taille moyenne (caractères normaux) diminue, et celui de grande taille (caractères fusionnés) augmente.*

auteurs norment SSF en divisant ce nombre par le carré du nombre de composantes comprises dans le groupe 2 et 3. En effet, ces derniers font l'hypothèse que le carré du nombre de composantes comprises dans le groupe 2 et 3 est toujours plus grand que le nombre de composantes du groupe 1. C'est-à-dire que le nombre de caractères sans dégradations ajouté au carré du nombre de caractères qui se touchent est toujours plus grand que le nombre de composantes de bruits. Cette hypothèse s'avère vraie sur leur corpus de document et permet aux auteurs de garantir que TCF soit compris entre 0 et 1.

En 1998 les auteurs de [GKN98] introduisent deux nouvelles mesures. La première Black Density Factor (BDF) a pour objectif de mesurer l'importance des caractères dont l'encre est trop diffusée dans le papier (figure 2.2b). Pour cela, les auteurs définissent la densité d'une composante connexe comme le ratio entre le nombre de pixels de la composante et l'aire de sa boite englobante. Puis, ils comptent le nombre de composantes connexes noires dont la densité est inférieure à 0.75 et norment la mesure en divisant ce nombre par le nombre total de composantes noires de l'image. La seconde Stroke Thickness Factor (STF) mesure l'épaisseur de l'encre des caractères. Pour mesurer l'épaisseur des caractères, les auteurs utilisent la médiane des *run-lenght* horizontaux de l'image. Dans l'article, un run-lenght correspond au nombre de pixels consécutifs le long de l'axe horizontal sur une composante connexe. De plus, comme l'épaisseur des caractères dépend de la fonte utilisée, la médiane des run-lenght est divisée par la hauteur moyenne d'un caractère. STF est alors formalisé par :

$$STF = \frac{\sum \text{des épaisseurs des caractères}}{\|\text{composantes connexes de l'image}\|}$$

Ce descripteur semble peu robuste aux changements de police au sein d'une même page.

En 1999 M.Cannon [CHK99] introduit une dernière mesure Font Size Factor (FSF) qui norme la taille de la fonte (afin d'être invariant en échelle). FSF est définie par :

$$FSF = \frac{\text{taille moyenne des caractères - taille d'un caractère minimum}}{\text{taille d'un caractère maximum - taille d'un caractère minimum}}$$

L'ensemble de ces mesures est utilisé pour prédire le résultat de différents OCRs [BKN95, CHKW97, GKN98, CHK99]. En 2003, ces mesures sont aussi utilisées pour créer un algorithme de sélection automatique de méthode de restauration [SCNS03]. Nous présenterons les méthodes utilisées pour faire cette prédiction dans le chapitre 3.

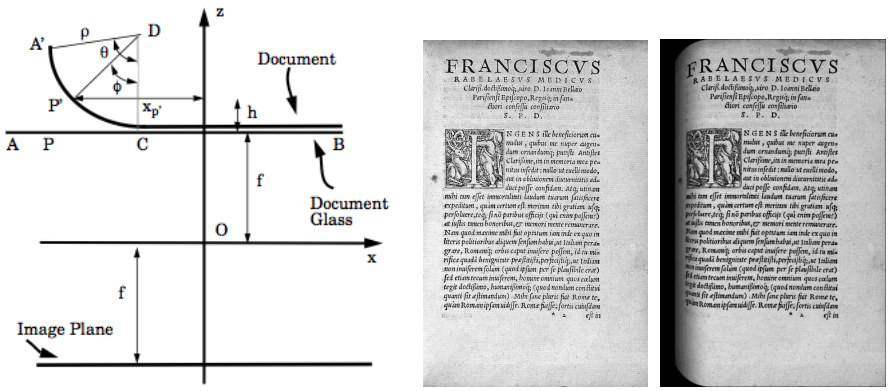
Les descripteurs présentés ont l'avantage de caractériser les principales dégradations que subissent les caractères. Malheureusement, elles ne peuvent s'utiliser que sur des images **binaires** dont le contenu est du **texte**. Comme ces mesures ne peuvent être utilisées qu'après binarisation, elles peuvent servir pour mesurer la qualité de la binarisation, pour prédire ou sélectionner un algorithme d'OCR. En aucun cas, elles ne peuvent servir à mesurer la qualité d'un document en niveau de gris. De plus ils ne prennent pas en considération des éléments influençant les performances de l'OCR par exemple la présence de texte en italique ou en gras, de titres ou de symboles similaires. Une perspective intéressante serait de compléter cette liste en tenant compte des caractéristiques précédentes.

2.1.3.2 Étude des méthodes de modélisation des dégradations

Les modèles de dégradations détaillés dans [Bai93, RL94], s'intéressent aux défauts engendrés par la phase de numérisation de documents par des scanners générant obligatoirement des images binaires. Ce modèle dégrade l'image en ajoutant des dégradations typiques des scanners et des photocopieurs des années 90. L'objectif de ces deux modèles est avant tout de créer des images numérisées synthétiques afin d'aider la phase d'apprentissage des OCRs. Un texte est alors généré puis transformé en image. Un ensemble de modifications qui ont pour objectif de simuler l'utilisation répétée d'un scanner est ensuite appliqué à cette image. Ces opérations ont plusieurs paramètres choisis aléatoirement. L'algorithme jouera par exemple sur la taille du texte, sa résolution, son orientation, la quantité de flou ou de bruit, et même sur la gigue qui est le délai de transmission entre l'émetteur et le récepteur.

Kanungo, dans [KHP93], propose deux modèles de dégradation. Le premier est global et modélise les dégradations liées à la profondeur de la reliure d'un document. Ces dégradations sont tout d'abord un problème d'illumination, puis un problème de focale qui a pour effet de rendre flous les caractères proches de la reliure. Ici, les auteurs se placent dans un environnement en 3 dimensions avec l'image et le capteur du scanner. L'image est fixe et le capteur se déplace le long d'un axe pour numériser l'image. Afin de donner un aspect de profondeur, et de courbure au document, les auteurs modélisent la reliure par un arc de cercle, comme le montre la figure 2.4a. Chaque point de l'image originale sera projeté sur cet arc. L'algorithme simule donc le passage d'une surface plane à une surface courbée. Avec cette méthode, les documents générés sont semi-synthétiques : c'est une image réelle précédemment numérisée et sans problème de reliure (figure 2.4b) qui est dégradée (figure 2.4c).
Le second modèle est lui local et a pour objectif cette fois-ci de modéliser la distorsion que subissent les caractères lors de la numérisation avec des scanners binaires. Cette distorsion provoque l'ajout de plusieurs types de dégradations (figure 2.5) : caractères fusionnés ou cassés, bordures dégradées, etc. Ce modèle est basé sur des opérateurs morphologiques qui en fonction d'une probabilité donnée en entrée de l'algorithme inversent (noir vers blanc ou blanc vers noir) les pixels des caractères. Ce modèle peut aussi être utilisé pour ajouter du bruit au fond du document. Dans [Zi05, ZD04] une approche similaire est utilisée dans l'objectif de créer de la vérité-terrain pour les algorithmes d'OCRs.

Les modèles présentés ci-dessus sont très largement utilisés par la communauté, mais s'intéressent uniquement aux dégradations binaires engendrées par la numérisation de documents contemporains en utilisant des scanners produisant des images binaires. Or, certaines dégradations présentes sur les documents anciens sont, contrairement aux documents modernes, résultantes d'un processus physique lié à



(a) (b) (c)

FIGURE 2.4 – *Modélisation de la courbure d'un document selon [KHP93] : a. modélisation de la reliure du document, b. une image originale sans défaut de reliure, c. sa version dégradée.*

(a) (b) (c)

FIGURE 2.5 – *Modéle de distorsion morphologique de [KHP93] : (a) caractère original, (b) et (c) deux exemples de distorsions différentes.*

l'âge et la mauvaise conservation du document. Ces dégradations sont omniprésentes sur les documents anciens et sur les documents de mauvaise qualité. De plus, les documents actuels sont numérisés en niveaux de gris. Ces modèles devraient être étendus pour permettre de générer des documents synthétiques représentant les caractéristiques des documents anciens.

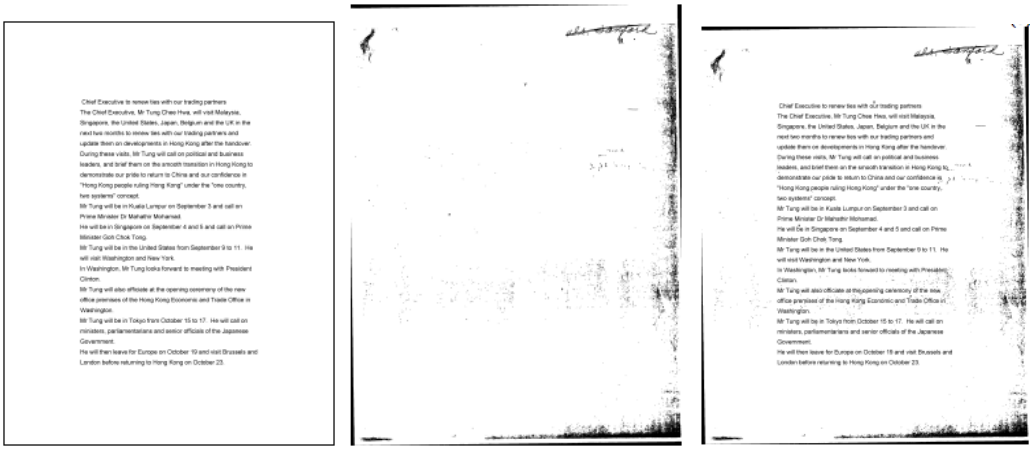
Les auteurs de [Zi05, ZD04] ne se contentent pas de modéliser des dégradations liées à la phase de numérisation. Une partie de leur système a pour objectif d'ajouter des dégradations telles que des taches ou de la transparence à des documents déjà numérisés. Les dégradations sont extraites à partir de document les contenant et sont ensuite ajoutées aux documents existants afin de les dégrader. Ce processus peut se modéliser de la façon suivante :

$$I_D = I + D$$

Avec I_D l'image semi-synthétique, I l'image originale et D les dégradations. L'opération $+$ est une opération de fusion qui est faite linéairement de façon additive en respectant la table de transformation 2.1.

Image originale	Modèle	Fusion
Blanc (255)	Noir (0)	Noir (0)
Blanc (255)	Blanc (255)	Blanc (255)
Noir (0)	Blanc (255)	Noir (0)
Noir (0)	Noir (0)	Noir (0)

TABLE 2.1 – *Table de transformation du modèle de dégradation présenté dans [Zi05, ZD04]*



(a) (b) (c)

FIGURE 2.6 – *Création de documents dégradés par utilisation de documents modèles [Zi05] : (a) le document original* propre*, (b) le modèle contenant des tâches, (c) la fusion par un processus linéaire additif respectant la table de transformation 2.1.*

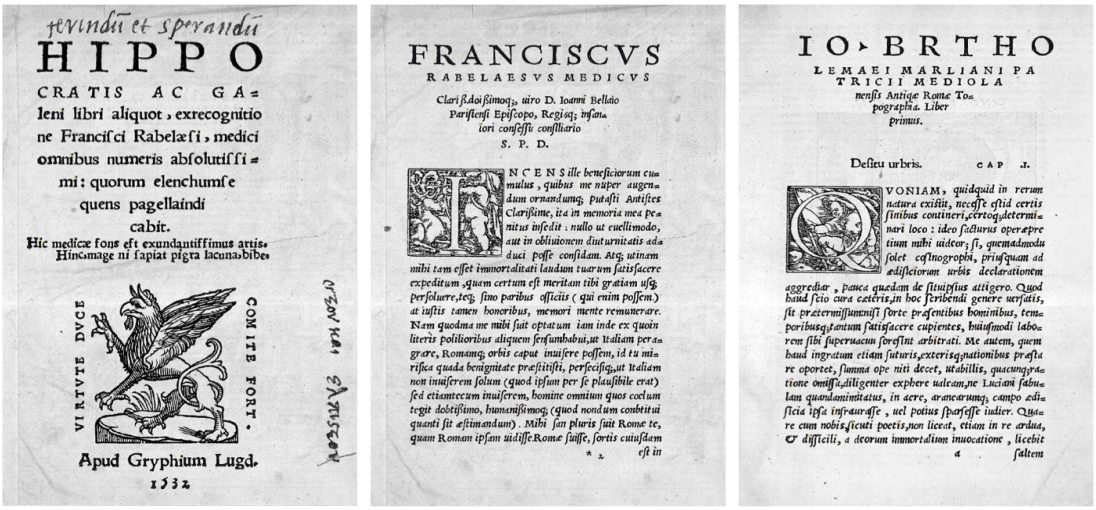
Cette méthode est pertinente pour créer de grandes quantités de documents synthétiques à condition de disposer de différents types de dégradations. Cela suppose d'avoir un nombre important de dégradations et de pouvoir extraire facilement les zones concernées par ces dernières. Là encore, ces techniques s'adressent seulement aux documents binaires et il serait intéressant de les étendre aux documents en niveaux de gris ou en couleurs.

Plus récemment dans [JVD+10] les auteurs proposent un nouvel environnement logiciel permettant de créer des documents anciens semi-synthétiques. La méthode reprend le principe proposé dans [Zi05, ZD04], mais en l'étendant. Au lieu de faire la combinaison de deux images (document propre et image modèle contenant les dégradations), l'article propose un éditeur permettant la création de documents présentant les caractéristiques de documents anciens : fontes anciennes, différents fonds, différentes illustrations, etc. Une vérité-terrain contenant le texte brut, la position des caractères, l'interligne, et d'autres paramètres peut alors être générée conjointement à l'image de document semi-synthétique. La méthode proposée peut ici être utilisée pour tout type de document (anciens, modernes, niveaux de gris, couleurs, etc.). Des exemples de documents générés sont présents en figure 2.7. Cet outil a tout d'abord été utilisé dans le domaine de la reconnaissance de fontes anciennes. Nous l'avons ensuite étendu en partie pour pouvoir appliquer des modèles de génération de transparence aux documents créés. Cette extension est détaillée en chapitre 4 traitant des problèmes liés à la création et au partage de vérités terrains.

D'autres initiatives de recherche visent à modéliser le défaut de transparence (apparence du verso à travers le recto). Cette dégradation est très souvent présente sur les documents anciens et les travaux la concernant sont nombreux. La section (2.3) lui est consacrée.

2.1.3.3 Identification des descripteurs présents dans les méthodes de restaurations

Les méthodes de restauration s'intéressent à la caractérisation et l'extraction des dégradations présentes sur des images de documents. Pour cela, certaines utilisent plusieurs descripteurs permettant soit d'extraire directement les pixels concernés par la dégradation, soit de caractériser la dégradation pour la traiter et ainsi restaurer l'image. Notre objectif est différent puisque nous désirons quantifier l'importance d'une ou plusieurs dégradations. Néanmoins, il est possible de s'inspirer de ces méthodes pour



(a) (b) (c)

FIGURE 2.7 – *Exemples de documents anciens semi-synthétiques créés par le logiciel présenté dans [JVD+10].*

créer de tels descripteurs.

En ce qui concerne le “dewarping” qui corrige les problèmes de déformation de la surface du document (ondulation, courbure de la reliure, pages gondolées, etc.). Deux approches sont proposées : la modélisation 3D et l’extraction des lignes en vue d’une approximation de ces dernières.
Les travaux sur la courbure de la reliure se sont, pour beaucoup, concentrés sur la reconstruction du modèle 3D (shape from shading) du document et en particulier de sa reliure. Les auteurs de [ZTF04] utilisent les informations liées à l’illumination non linéaire (niveaux de gris) de la reliure pour en calculer le modèle 3D. La méthode est évaluée sur 100 images numérisées par un scanner et présente de bons résultats. Malheureusement, les scanners de document anciens utilisent des sources de lumière plus homogènes [ULB05] et la méthode ne peut être appliquée. Pour contourner ce problème, d’autres méthodes utilisant des lasers [Pil01] ou plusieurs caméras [ULB04, YKKM04] ont été proposées toujours avec l’objectif de reconstruire un modèle 3D de la courbure du document.
Les méthodes précédentes [Pil01, ULB04, YKKM04], partagent le même inconvénient à savoir de nécessiter beaucoup de matériels souvent complexes et fastidieux à calibrer. Dans [ZT02, LMAB01, ZT01, ULB05] les auteurs procèdent tout d’abord à l’extraction des lignes de textes de l’image de document. Cette étape nécessite une extraction des caractères qui est obtenue par binarisation du document. Une fois cette opération faite les lignes sont approximées soit par des courbes de Béziers [LMAB01] soit par des régressions polynomiales [ZT02]. Dans le cas de [ZT01], seule la partie concernant la reliure est traitée. Les mots de la page sont classés (appartenant à la reliure, n’y appartenant pas) en fonction des caractéristiques de leurs boites englobantes. Les mots n’y appartenant pas ne seront pas modifiés ou traités. Les auteurs de [ULB05] utilisent aussi les caractéristiques des boites englobantes, mais sur les composantes connexes binarisées (les caractères).

Certains chercheurs se concentrent sur la correction de l’orientation du texte de l’image de document numérisé. La plupart des algorithmes d’estimation de l’orientation du texte présent dans la littérature peuvent être classés en trois catégories : les méthodes utilisant les profils de projection, les méthodes à base de regroupements de pixels ou de composantes connexes et les méthodes basées sur la transformation de Hough.
Les méthodes utilisant les profils de projections projettent le document en différents angles [Hou83]. Les pics correspondant aux lignes, et les creux correspondant aux interlignes peuvent ainsi être identifiés.

L'angle donnant la différence maximum entre le nombre de pics et de creux est considéré comme l'angle de redressement. Ce genre de méthode peut être assimilée à un algorithme de type "brut force" (test de toutes les combinaisons possibles) avec des temps de calcul particulièrement longs. Des approches statistiques (convergence rapide) [Bai95] et de partitionnement [AH90, PZ92] permettent de réduire ces temps de calcul.

Les méthodes [HYR86, JBWK99] et [CH94] regroupe les composantes connexes entre elles à l'aide de l'algorithme des plus proches voisins. L'histogramme des directions entre chaque pixel de deux composantes voisines est calculé et permet de définir l'orientation du document (pics dans l'histogramme). Cette méthode est généralisée dans [O'G93]. Dans [PC96] et [MY99], les auteurs détectent les interlignes en calculant la hauteur moyenne des caractères. D'autres comme [CHP95] utilisent des opérations morphologiques pour identifier les mots puis les lignes de textes, avant de calculer l'orientation du texte par un estimateur Bayesien.

Plusieurs méthodes [HFD90, LTW94, SG89] utilisent la transformée de Hough pour trouver l'orientation du document. Le problème de ces méthodes est leurs temps de calcul bien trop longs. Certaines méthodes [HFD90, LTW94] tentent de filtrer les pixels les plus importants afin d'obtenir des résultats plus précis et moins couteux en temps. Par exemple, [LTW94], utilise seulement les pixels au bas de chaque composante connexe.

Les méthodes de restauration des caractères sont peu nombreuses. Nous pouvons néanmoins citer [SKK95, HCW97] qui se base sur la correction apportée par l'OCR. En effet, les caractères dégradés sont souvent mal reconnus par les OCRs. Or, ces derniers utilisent des dictionnaires qui permettent via des calculs de probabilités de remplacer les caractères mal reconnus par les bons (par exemple le mot *dicument* peut être facilement corrigé par le mot *document*). Par contre, cette technique n'est pas très efficace sur les documents écrits dans des langues anciennes ou plusieurs langues à la fois, ainsi que sur les documents contenant beaucoup de noms propres, ou très fortement dégradés. Une approche proposée dans [Dro03] tente de résoudre ce problème en utilisant des connaissances individuelles sur chaque symbole (sans utiliser des aprioris sur le langage). Un mot devient un graphe (ou les noeuds représentent les lettres et, les arrêtes, représentent l'adjacence de ces lettres). Puis, pour chaque noeud l'algorithme liste l'ensemble des sous-graphes connectés possibles. Chacun de ces sous-graphes est alors comparé aux caractères connus dans une base de données. Le sous-graphe le plus similaire à un de ces éléments est alors choisi. On remplace ensuite l'ensemble des composantes du sous-graphe par le modèle de la base de données (caractères sans dégradations).

Ces différentes techniques ont l'inconvénient de seulement remplacer les caractères par des caractères modèles de référence sans dégradations. Une méthode présentée dans [ABE06] propose une réelle restauration des caractères cassés en utilisant des filtres de Gabor et les contours actifs.

Plusieurs méthodes s'intéressent à la restauration des perturbations fond-encre. Les auteurs de [SG04a] corrigent le fond du document en normalisant les intensités des pixels d'une image. R. Farrahi Moghaddam dans [MC09b] propose l'utilisation de différents descripteurs qui sont ensuite utilisés pour calculer les coefficients d'une méthode de restauration basée sur les équations aux dérivées partielles (du type Perona-Malik). Pour finir, [BSDLS11] se base sur la combinaison de deux filtres d'image au travers d'un masque. Le masque s'applique à une image dans laquelle le bruit présent sur le fond est amoindri par une méthode de débruitage appelée Total Variation Denoising. L'image résultante est alors filtrée par un algorithme de type "Non Local Means" pour réduire le bruit présent dans les zones de texte.

Ce état de l'art a permis de mettre en évidence que les algorithmes de restoration d'images de documents utilisent un très grand nombre de descripteurs. Il est impossible d'être exhaustif, mais retenons tout de même les descripteurs suivants : l'approximation les lignes (courbes de Béziers ou approximation polynomiale), boîtes englobantes, orientation du texte, profils verticaux et horizontaux, run-length, transformation de Hough, hauteur moyenne d'un caractère, filtre de Gabor, contours actifs, etc.

2.1.4 De l'analyse des besoins et de l'état de l'art à la création de descripteurs de qualité

Dans cette section, nous proposons une méthodologie permettant d'aboutir, pour une catégorie de défauts prédéfinis, à la création d'un ensemble de descripteurs qualité. En nous appuyant à la fois sur une analyse de l'état de l'art (descripteurs bas niveaux, algorithmes de restaurations, méthode de synthèse de défauts, etc.) et sur une synthèse des connaissances métiers relatives aux défauts que l'on souhaite caractériser, nous pensons qu'il est possible de définir des descripteurs pertinents et robustes. La figure 2.8 illustre les grandes étapes de notre méthodologie.

Collecte des connaissances métiers associées au défaut à caractériser

La partie "rose" de la figure 2.8 symbolise le travail relatif à l'analyse, à partir de cas réels, de l'influence que peuvent avoir des défauts spécifiques sur les performances d'un algorithme de traitement ou d'analyse d'images. Pour cela deux sous-étapes sont nécessaires.

La première est d'analyser, avec un expert du domaine, un ensemble de documents réels présentant le type de dégradation que l'on souhaite caractériser. Cette analyse doit permettre de qualifier les défauts. Ces critères qualitatifs peuvent être par exemple, l'intensité ou la quantité de pixels dégradés, la localisation en fonction d'autres éléments (textes, illustrations, marges, etc.), leur fréquence d'apparition, etc.

Dans un second temps il est nécessaire d'étudier l'influence des diverses formes prises par la dégradation que l'on souhaite caractériser sur les performances d'algorithmes de traitement ou d'analyse d'images de documents (binarisation, segmentation texte dessin et OCR dans le cadre de nos travaux). On peut par exemple chercher à identifier si les performances d'un algorithme sont corrélées à la quantité de la dégradation, à sa fréquence d'apparition, à sa localisation, etc.

La mise en commun des informations recueillies doit permettre d'aboutir à la définition d'une liste de qualificatifs décrivant les différentes formes du défaut à caractériser avec, pour chacun d'entre eux, son impact sur les performances d'algorithmes de traitement ou d'analyse d'images.

Extraction des pixels associés au défaut à caractériser

La seconde partie de notre méthodologie repose sur une étude de l'état de l'art permettant d'aboutir à l'extraction des pixels concernés par la dégradation. Pour cela, nous proposons l'étude systématique de méthodes décrivant la modélisation de défauts dans l'optique de les synthétiser ainsi que l'étude de méthodes de restauration. Ces méthodes décrivent rarement une technique permettant une extraction des pixels associés à un défaut, elles sont cependant une source d'information importante. Ainsi, nous montrerons dans la section 2.3.2 qu'il est par exemple possible, en s'inspirant de l'état de l'art, de proposer de nouvelles méthodes de segmentation adaptées à des défauts prédéfinis.

Définitions de nouveaux descripteurs qualité pour documents

À cette étape de notre méthodologie, nous avons donc mis en place une méthode d'extraction des pixels associés à un défaut et nous avons également une idée précise des différentes formes prises par ce défaut sur des documents réels. La dernière étape consiste en la définition de descripteurs qui seront calculés sur les pixels extraits.

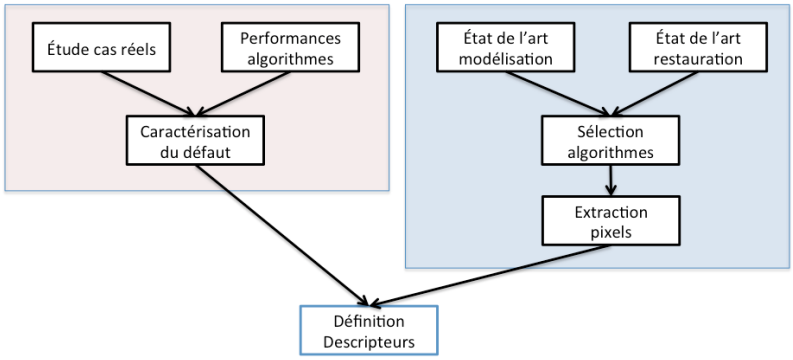


FIGURE 2.8 – *Méthodologie permettant d'aboutir à la définition des descripteurs de qualité d'images de documents*

L'objectif est que ces descripteurs permettent de retranscrire "numériquement" la majorité des observations faites sur les images réelles. Ils doivent également permettre de mettre en évidence une corrélation entre leurs valeurs prises par ces descripteurs et les performances d'algorithmes testés lors de l'étape initiale de notre méthodologie. Nous proposons au final de caractériser les pixels segmentés selon leurs couleurs, les formes ou les textures qu'ils créent et enfin leurs positions et leur quantité par rapport aux pixels qui ne sont pas associés au défaut. Lors de cette création de descripteurs, une attention particulière doit être portée à l'invariance de ces derniers à l'échelle de l'image et à son angle principal.

Dans la section suivante, nous illustrerons chaque point de notre méthodologie au travers de deux exemples de caractérisation de défauts typiques des documents anciens (la transparence et les perturbations fond-encre).

2.2 Proposition de descripteurs caractérisant les perturbations fond-encre

Cette section détaille la création de descripteurs caractérisant les perturbations fond-encre dans l'objectif de prédire les performances des algorithmes de binarisation.

2.2.1 Caractéristiques des perturbations fond-encre et étude des impacts sur les algorithmes de binarisation

Comme nous l'avons défini, les perturbations fond-encre modifient la distribution des niveaux de gris de l'image. En effet, les perturbations fond-encre correspondent à divers types de dégradations telles que les taches, la transparence, les problèmes d'illumination ou encore l'effacement de l'encre. On peut considérer que toutes les composantes connexes grises qui n'appartiennent pas à l'encre ni au fond sont des perturbations fond-encre.

Dans l'objectif de pouvoir définir un ensemble de descripteurs permettant d'expliquer les performances d'algorithmes de binarisation, nous avons exécuté un ensemble d'algorithmes sur un échantillon de documents anciens qui présentaient ce type de dégradations. Définir des descripteurs permettant d'évaluer la performance des algorithmes est une tâche complexe pour deux raisons :

1. Il est impossible de considérer une unique caractéristique pour évaluer globalement les performances de l'algorithme.
2. Les méthodes de binarisation étant basées sur des stratégies différentes, elles peuvent réagir de façon opposées à une même caractéristique. Par exemple, la figure 2.9 présente trois documents

(a) (b) (c)

(d) Binarisation Otsu [Ots75] (e) Binarisation Kittler [KI86] (f) Binarisation Li [LL93]

(g) Binarisation Sauvola [SP00] (h) Binarisation Otsu [Ots75] (i) Binarisation Shijian [SLT11]

FIGURE 2.9 – *Exemples d'erreurs provoquées par les perturbations fond-encre sur des méthodes de binarisation. La première ligne montre des extraits de documents originaux (extrait de la base DIBCO) la seconde ligne présente des erreurs de certaines méthodes de binarisation et la troisième ligne montre que certaines méthodes ne réagissent pas de la même façon aux dégradations.*

plus ou moins dégradés. Chacune de ces images de documents est binarisée avec 2 méthodes différentes. La première engendre un grand nombre d'erreurs alors que la seconde propose des résultats visuels plutôt bons.

Cependant, nos tests montrent que les méthodes de binarisation testées sont plus ou moins sensibles aux caractéristiques suivantes :

1. **La quantité** de composantes de dégradation par rapport au nombre de composantes d'encre. Certes, il semble évident que plus le document possède des perturbations fond-encre, plus la distribution des niveaux de gris de l'image est uniforme, plus le nombre d'erreurs des méthodes de binarisation augmente. Néanmoins, il semble que pour une même quantité de dégradation, les documents contenant un grand nombre de pixels d'encre sont moins sujets à erreur que ceux en contenant très peu.
2. **L'intensité** des dégradations par rapport au fond et à l'encre. La majorité des composantes de dégradation ont un niveau de gris supérieur à l'encre et inférieur au fond. Nous avons remarqué que certains algorithmes de binarisation génèrent plus d'erreurs dans deux cas différents : lorsque le niveau de gris moyen des dégradations est proche de l'encre et lorsque le niveau de gris des dégradations est distant de celui du fond. Il est important de noter que ces deux différents cas ne sont pas forcement liés, par exemple la figure 2.9b, présente une dégradation (petite tache d'humidité) dont l'intensité est loin de celle de l'encre, mais aussi loin de celle du fond (qui est très clair). La binarisation de Kittler [KI86] (figure 2.9e) génère une erreur sur cette dégradation.
3. **La taille** des composantes. La taille des composantes de dégradations varie fortement. Un problème d'illumination peut créer une composante grise dont l'aire est parfois supérieure à 20% de l'aire de la page. Les taches peuvent affecter plusieurs lignes de textes. Les composantes de

transparence ont une taille proche de celle des composantes de textes, mais sont souvent fragmentées. D'autres dégradations, comme le bruit, créent des composantes de très petites tailles (quelques pixels seulement). Nous avons aussi remarqué que les grandes composantes de dégradation sont dans la plupart des cas connectées à des composantes de texte. Tous ces différents éléments sous-tendent des distributions (de la taille des composantes) très différentes. Les algorithmes de binarisation sont plus ou moins sensibles aux différents types de distributions générées par ces perturbations. Par exemple, certains algorithmes semblent plus robustes aux perturbations fond-encre de grande taille (figures 2.9a et 2.9g).

4. **La position** des composantes des perturbations fond-encre semble aussi jouer sur les performances des algorithmes de binarisation. Certains algorithmes semblent être plus robustes lorsque la plupart des composantes de dégradation sont voisines de celles du texte comme le montrent les figures 2.9c et 2.9i. D'autres algorithmes semblent à l'inverse très sensibles à cette caractéristique ; c'est par exemple le cas des méthodes de binarisation adaptatives.

2.2.2 Identification des pixels de perturbation fond-encre

Dans cette thèse, nous définissons par *perturbations fond-encre* l'ensemble des dégradations perturbant les algorithmes de séparation fond-encre (binsarisation). Ces dégradations modifient la distribution des niveaux de gris de l'histogramme global et se caractérisent visuellement sur l'image de document par l'ajout de composantes grises plus ou moins sombres. Par exemple, les taches, la transparence, l'effacement de l'encre des caractères ou encore les problèmes d'illumination sont des perturbations fond-encre.

2.2.2.1 Sélection de méthodes de modélisation et de restauration des perturbations fond-encre

La plupart des modèles de dégradation proposent de produire un type de dégradation particulier, la transparence par exemple. Néanmoins, les perturbations fond-encre regroupent un grand nombre de types de dégradation. Nous nous sommes intéressés au modèle présenté par R.F. Moghaddam dans [MC09a], car la méthode permet de générer plusieurs types de dégradations présents dans les documents. Ces types de dégradations correspondent par exemple aux vieillissements du document, à la transparence recto verso, à l'ajout de taches, aux problèmes d'illuminations. Comme présentée sur le schéma 2.10, cette méthode de modélisation combine plusieurs couches associées à des types de dégradations vers une couche de destination en se basant sur une diffusion anisotropique. Ces couches peuvent être de différentes natures par exemple des couches sans altération (encre du recto, encre du verso) ou des couches altérées par des dégradations (fond, taches, etc.).

Nous définissons l'opérateur $DIFF(u, s, c)$ qui représente la diffusion d'une couche source s, vers une couche de destination u avec le coefficient c. La méthode se base sur l'équation de diffusion 2.1 :

$$\frac{\partial u}{\partial t} = \sum_{i \in sources} DIFF(u, s_i, c_i) \tag{2.1}$$

Avec :
– u, l'image de destination,
– $sources$, l'ensemble des images à diffuser (recto, verso, fond, taches, etc.),
– s_i, une image source,
– c_i, le coefficient de diffusion adapté à l'image s_i.

Le coefficient c_i associé à chacune des sources s_i permet de contrôler l'importance de la diffusion de cette couche sur le document final. Ainsi, en fonction de l'importance de la dégradation voulue on peut régler la diffusion de la source s_i vers sa destination u. On pourra contrôler l'importance de la diffusion de l'encre dans le papier, des perturbations environnementales affectant le document, de la transparence du verso, etc.

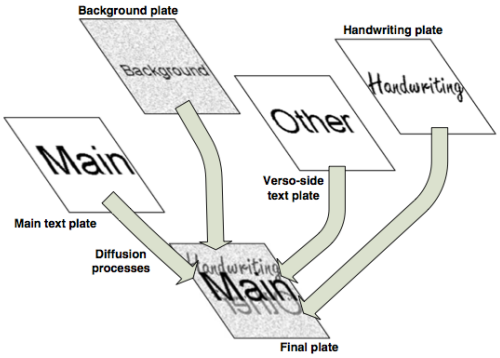
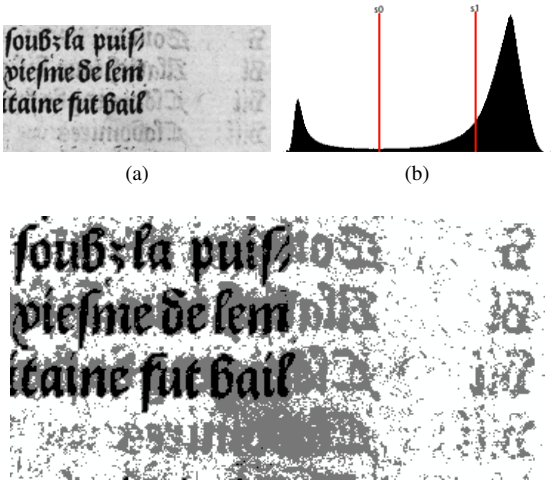


FIGURE 2.10 – *Schéma général de la méthode : un document peut être modélisé comme la diffusion de plusieurs couches d'information (recto, verso, dégradations) vers une image de destination (le document final). Image issue de [MC09a].*

2.2.2.2 Extraction des pixels par trinarisation

Sur les documents, la plupart des dégradations (transparence, taches, bruits, illumination non linéaire, etc.) apparaissent comme des composantes connexes avec des niveaux de gris différents de ceux du fond et de l'encre. Le document de la figure 2.11a présente plusieurs types de dégradations où le niveau de gris des pixels varie du sombre (taches d'encre) au clair (transparence faible).



(a) (b)

(c)

FIGURE 2.11 – *Exemple de résultats de la trinarisation : les pixels, dont l'intensité, est inférieure à s_0 correspondent aux pixels d'encre, ceux entre s_0 et s_1 aux dégradations, et ceux supérieurs à s_1 aux pixels de fond.*

En s'inspirant de la section précédente et de [MC09a], nous supposons qu'un document historique peut être modélisé comme la diffusion d'un ensemble de couches d'informations. Nous considérerons les dégradations de manière globale sans proposer une distinction précise de chaque catégorie de chaque dégradation. Au contraire, nous mesurons et caractérisons globalement les dégradations du document

en partitionnant en trois classes l'histogramme global de l'image de document. Les trois classes sont I l'ensemble des pixels d'encre, D l'ensemble des pixels de dégradation et B les pixels du fond. En notant $g(p)$ l'intensité d'un pixel p, ces partitions se définissent à partir de deux seuils s_0 et s_1 de la manière suivante :

1. $\mathcal{I} = \{p; g(p) \leq s_0\}$
2. $\mathcal{D} = \{p, s_0 < g(p) < s_1\}$
3. $\mathcal{B} = \{p, g(p) \geq s_1\}$

L'identification des deux seuils peut être réalisée par n'importe quel type d'algorithme de classification. Suite à nos expérimentations, l'algorithme de fragmentation *k-mean* avec le paramètre k fixé à trois classes s'avère être le plus pertinent. La figure 2.11 montre que la plupart des défauts d'un document ancien peuvent être extraits en utilisant ces deux seuils. Cependant, on peut remarquer que certains pixels gris (du fond, des bords des caractères, etc.) sont mal classés en tant que pixels de dégradation. Cette classification des pixels n'a pas pour objectif d'être très précise, l'importance est d'extraire une grande majorité des pixels de dégradations et des pixels gris pouvant perturber les algorithmes de binarisation. Ces pixels gris peuvent parfois appartenir aux bords des caractères ou au fond.

2.2.3 Proposition de nouveaux descripteurs pour la caractérisation de perturbations fond-encre

Cette classification des pixels nous permet de caractériser les défauts en fonctions de l'encre et du fond d'une image de document. Pour ce faire nous devons définir un ensemble de descripteurs. Cet ensemble peut être divisé en deux groupes distincts : les premiers sont extraits directement depuis l'histogramme global des niveaux de gris de l'image, les autres sont liés à la position des pixels de dégradation.

2.2.3.1 Descripteurs globaux

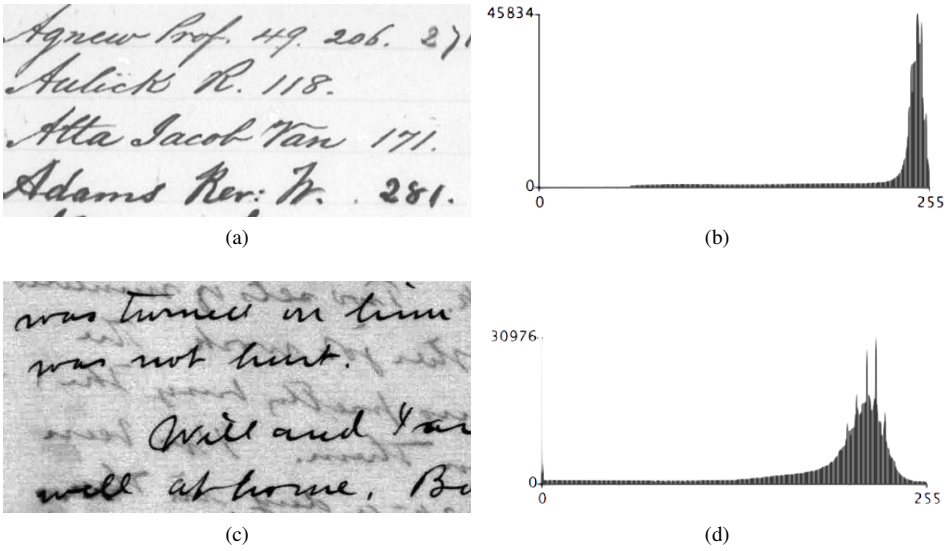
Moments colorimétriques

L'histogramme des niveaux de gris d'une image de document contient des informations qui caractérisent de façon globale la quantité et l'intensité (en niveaux de gris) des trois couches qui nous intéresse à savoir l'encre, les dégradations et le fond. La figure 2.12 et la table 2.4 illustre la différence entre l'histogramme d'une image "propre" et une image dégradée.

Nous proposons de calculer certains indicateurs statistiques globaux sur l'histogramme des niveaux de gris : la moyenne, la variance et la "skewness" . La skewness d'un histogramme est un coefficient associé à l'asymétrie d'une distribution. Dans la suite, nous dénoterons par μ la moyenne, v la variance, et s la skewness.
Les indicateurs statistiques sont aussi calculés sur les trois *sous-histogrammes* (définis par les seuils expliqués en section 2.2.2.2) afin de caractériser chaque distribution indépendamment (pour l'encre, les dégradations, et le fond).

Nous avons à ce stade douze mesures :
- μ, v, s
- $\mu_{\mathcal{I}}$, $v_{\mathcal{I}}$, $s_{\mathcal{I}}$
- $\mu_{\mathcal{D}}$, $v_{\mathcal{D}}$, $s_{\mathcal{D}}$
- $\mu_{\mathcal{B}}$, $v_{\mathcal{B}}$, $s_{\mathcal{B}}$

Mesures globales en relation avec l'encre et le fond



(a) (b)

(c) (d)

FIGURE 2.12 – *Exemples d'histogrammes des niveaux de gris sur des images de documents : a. un document ancien relativement propre, b. l'histogramme des niveaux de gris lui correspondant, c. un document ancien dégradé (transparence importante), d. l'histogramme des niveaux de gris lui correspondant. L'histogramme du document dégradé (d) est plus irrégulier, ses valeurs sont plus dispersées.*

Les indicateurs statistiques précédents caractérisent l'histogramme des niveaux de gris et permettent donc de mesurer les critères de quantité et d'intensité retenus en section 2.2.1 de façon générale. Cependant ils ne peuvent à eux seuls caractériser la relation entre les couches d'encre, de dégradations et du fond. C'est pour cela que nous devons introduire deux autres mesures, elles aussi extraites à partir de l'histogramme des niveaux de gris afin de caractériser les relations entre les couches.
La différence entre les niveaux de gris moyens des trois couches d'informations (encre, dégradations et fond) semble directement corrélée aux résultats de la binarisation. Par exemple, si le niveau de gris moyen de la couche de dégradation est proche de celui de l'encre, il est probable que certains des pixels de dégradations seront considérés comme des pixels d'encre lors de la binarisation.
En ce sens, nous définissons tout d'abord deux descripteurs $\mathcal{MI}_I$ et $\mathcal{MI}_B$: $\mathcal{MI}_I$ correspond à la distance entre le niveau de gris moyen de la couche d'encre et le niveau de gris moyen de la couche de dégradation ; $\mathcal{MI}_B$ quant à lui, correspond à la distance entre le niveau de gris moyen de la couche de dégradations et celui du fond.

$$\mathcal{MI}_I = \frac{\mu_{\mathcal{D}} - \mu_{\mathcal{I}}}{255} \quad \mathcal{MI}_B = \frac{\mu_{\mathcal{B}} - \mu_{\mathcal{D}}}{255}$$

Les niveaux de gris des différentes couches ne sont pas les seules caractéristiques qui peuvent avoir des impacts sur les résultats d'une méthode de binarisation (ou d'autres algorithmes). En effet, la quantité de dégradations présente sur une image de document est elle aussi directement liée aux performances de la binarisation. Nous proposons donc une dernière mesure globale $\mathcal{MQ}$ correspondant au ratio entre la quantité de pixels de dégradation et celle des pixels d'encre :

$$\mathcal{MQ} = \frac{\|\mathcal{D}\|}{\|\mathcal{I}\|}$$

Cette première famille de descripteurs caractérise la qualité d'un document dans sa globalité à l'aide de 15 descripteurs : 12 d'entre eux sont des descripteurs génériques sur des histogrammes et les 3 derniers sont dédiés au traitement et à l'analyse d'images de documents.

2.2.3.2 Descripteurs locaux

La localisation des pixels de dégradation est une caractéristique significative de la qualité d'une image de document, et cela tant en terme de perception qu'en terme de performances d'algorithmes. Nous ne considèrerons pas les pixels individuellement, mais les composantes connexes de l'image. Une composante connexe de dégradation peut interférer de différentes façons avec une composante connexe de texte (figure 2.13). Par exemple, certaines méthodes de binarisation peuvent éventuellement faire des erreurs si des pixels de dégradations de couleur sombre viennent chevaucher ceux d'encres. Autrement dit, une composante connexe d'encre peut être déformée si cette dernière est connectée à une composante de dégradation. Dans d'autres cas, certaines méthodes de binarisation sont plus sensibles aux composantes de dégradation qui ne sont pas connectées à une composante de texte.

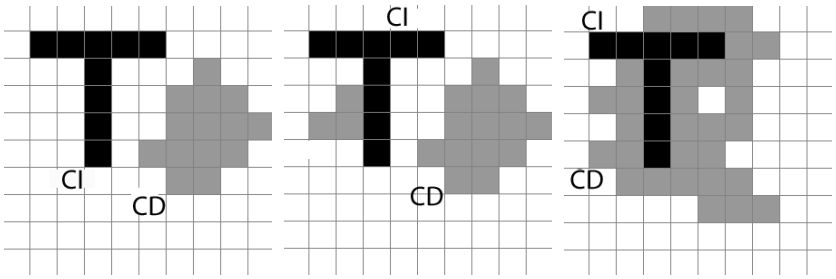


FIGURE 2.13 – *Les différentes localisations possibles d'une composante de dégradation ($c_{\mathcal{D}}$) en fonction d'une composante de texte ($c_{\mathcal{I}}$) : a. $c_{\mathcal{D}}$ n'est pas connecté à une composante de texte, b. une petite composante de dégradation vient se connecter à $c_{\mathcal{I}}$, c. une large composante de dégradation est connectée à $c_{\mathcal{I}}$.*

Les descripteurs locaux ont pour objectif de mesurer quantitativement l'impact de la localisation des composantes connexes grises (dégradations) vis-à-vis de celles du texte. Nous noterons, S l'ensemble des pixels d'une image de document et $CC(S)$ l'ensemble des composantes 4-connexes de S. Dans le reste de cette section nous utiliserons les notations suivantes : $\mathcal{C}_I = CC(\mathcal{I})$, $\mathcal{C}_D = CC(\mathcal{D})$ et $\mathcal{C}_B = CC(\mathcal{B})$.

Soit $c_{\mathcal{I}} \in \mathcal{C}_I$ une composante d'encre et $c_{\mathcal{D}} \in \mathcal{C}_D$ une composante de dégradation. Nous définissons SG comme le prédicat retournant "vrai" si $c_{\mathcal{I}}$ et $c_{\mathcal{D}}$ sont connectées :

$$SG(c_{\mathcal{I}}, c_{\mathcal{D}}) = \exists (p_{\mathcal{I}}, p_{\mathcal{D}}) \in c_{\mathcal{I}} \times c_{\mathcal{D}} \mid p_{\mathcal{I}} \text{ et } p_{\mathcal{D}} \text{ sont 4-connexes}$$

On peut alors distinguer trois cas différents où la localisation d'une composante de dégradation peut avoir une importance :

1. Si $c_{\mathcal{I}}$ et $c_{\mathcal{D}}$ ne sont pas connectées (2.13.a), le caractère original (représenté par la composante $c_{\mathcal{I}}$) ne sera pas affecté. La composante de dégradation $c_{\mathcal{D}}$ étant éloignée du texte, elle n'influencera pas la binarisation de ci. Par contre, cette composante étant isolée du texte, certains algorithmes de binarisation pourront la considérer comme étant du texte et donc commettre une erreur de binarisation. De façon à considérer ces configurations sur l'ensemble de l'image, nous définissons l'ensemble $C_{\mathcal{MA}}$ comme l'ensemble des composantes de dégradations n'étant pas connectées à une composante d'encre :

   $$C_{\mathcal{MA}} = \{c_{\mathcal{D}} \in \mathcal{C}_D \mid \forall c_{\mathcal{I}} \in \mathcal{C}_I, SG(c_{\mathcal{I}}, c_{\mathcal{D}}) = faux\}$$

   On mesurera globalement (sur toute l'image) le nombre de ces configurations en utilisant le descripteur $\mathcal{MA}$ traduisant la quantité relative de composantes d'encre et de composantes de dégradation non connectées :

$$\mathcal{MA} = \frac{\|C_{\mathcal{MA}}\|}{\|\mathcal{C}_I\|}$$

2. Si $c_{\mathcal{I}}$ et $c_{\mathcal{D}}$ sont au contraire connectées (2.13.b), le caractère original $c_{\mathcal{I}}$, pourra être mal binarisé. En effet, certains pixels de $c_{\mathcal{D}}$ seront considérés comme du texte et agglomérés à la composante $c_{\mathcal{I}}$. De façon à considérer ces configurations sur la totalité de l'image, nous définissons $C_{\mathcal{MS}}$ comme l'ensemble des composantes d'encre qui sont connectées à au moins une composante de dégradation :

$$C_{\mathcal{MS}} = \{c_{\mathcal{I}} \in \mathcal{C}_I \mid \exists c_{\mathcal{D}} \in \mathcal{C}_D, SG(c_{\mathcal{I}}, c_{\mathcal{D}})\}$$

Le descripteur associé à ce type de configuration est $\mathcal{MS}$ et se définit comme le ratio entre le nombre de composantes d'encre qui sont connectées à au moins une composante de dégradation ($C_{\mathcal{MS}}$) et le nombre total de composantes d'encre :

$$\mathcal{MS} = \frac{\|C_{\mathcal{MS}}\|}{\|\mathcal{C}_I\|}$$

3. Le descripteur $\mathcal{MS}$ mesure uniquement le nombre de composantes d'encre connectées à des composantes de dégradations. Il est aussi important de mesurer à quel point les composantes de texte peuvent être modifiées ou étendues. En ce sens, $\mathcal{MSG}$ mesure la déformation (en terme de nombre de pixels) subite par la composante d'encre. Le descripteur $\mathcal{MSG}$ est défini comme suit :

$$\mathcal{MSG} = \frac{Moy_{\{(c_{\mathcal{I}}, c_{\mathcal{D}}) | SG(c_{\mathcal{I}}, c_{\mathcal{D}})\}}(\|c_{\mathcal{I}}\| + \|c_{\mathcal{D}}\|)}{Moy_{c_{\mathcal{I}} \in \mathcal{C}_I}(\|c_{\mathcal{I}}\|)}$$

Avec,
– $\|c\|$ l'aire d'une composante connexe c.
– $Moy_{c \in C}(\|c\|)$ la moyenne des aires des composantes de l'ensemble C.
Plus la valeur de $\mathcal{MSG}$ est grande, plus le document est susceptible de contenir de grandes taches connectées aux composantes d'encre. Utilisé conjointement avec la mesure $\mathcal{MI}_I$, il permet de prédire si la tache sera ou non mal binarisée.

Le tableau 2.2, présente les valeurs des 3 mesures locales sur les schémas présents en figure 2.13.

	Figure 2.13.a	Figure 2.13.b	Figure 2.13.c
$\mathcal{MA}$	1	1	0
$\mathcal{MS}$	0	1	1
$\mathcal{MSG}$	0	1.3	4.3

TABLE 2.2 – *Descripteurs locaux calculés sur les exemples de la figure 2.13.* $\mathcal{MA}$ *est égal à* 1 *sur la figure 2.13.a et sur la figure 2.13.b étant donné que* 1 *composante de dégradation n'est pas connectée à aucune composante d'encre et qu'il existe* 1 *composante d'encre.* $\mathcal{MSG}$ *est égal à 0 sur la figure 2.13.a étant donné qu'aucune composante n'est connectée.* $\mathcal{MSG}$ *à une plus petite valeur sur la figure 2.13.b que sur la figure 2.13.c dû au fait que l'aire de l'union de la composante d'encre et celle de dégradation est plus petite.*

2.2.3.3 Conclusion sur les descripteurs de perturbation fond-encre

Précédemment nous avons listé les quatre caractéristiques des perturbations fond-encre qui influaient sur les performances d'algorithmes de binarisation à savoir : la quantité, l'intensité, la taille et la position des composantes de dégradation. Chaque descripteur est lié (directement ou indirectement) à une ou plusieurs de ces caractéristiques comme le résume le tableau 2.3.

Caractèristique	Mesures
Quantité	$\mathcal{MQ}$
Intensité	$\mathcal{MI}_I$, $\mathcal{MI}_B$, μ, v, s, $\mu_\mathcal{I}$, $v_\mathcal{I}$, $s_\mathcal{I}$, $\mu_\mathcal{D}$, $v_\mathcal{D}$, $s_\mathcal{D}$, $\mu_\mathcal{B}$, $v_\mathcal{B}$, $s_\mathcal{B}$
Taille	$\mathcal{MSG}$
Position	$\mathcal{MSG}$, $\mathcal{MS}$, $\mathcal{MA}$

TABLE 2.3 – *Les descripteurs en fonction des caractéristiques influençant les algorithmes de binarisation.*

On peut remarquer que le nombre de descripteurs (surtout concernant l'intensité) est assez important. Cela vient du fait que nous utilisons les moments statistiques sur trois différents histogrammes. Bien entendu, nous sélectionnerons, pour chaque modèle de prédiction, le sous-ensemble de descripteurs le plus significatif pour un algorithme donné (voir chapitre 3).

2.2.3.4 Présentations des mesures sur des exemples réels

Dans la section précédente, nous avons défini un vecteur de 18 descripteurs afin de caractériser la qualité globale d'une image de document pour les algorithmes de binarisation.

Afin de montrer l'intérêt de ce vecteur, nous analyserons tout d'abord les résultats de deux méthodes de binarisation sur 2 images de documents contenant des types de dégradations très différentes (tableau 2.4). Nous tenterons ensuite d'expliquer visuellement le résultat de ces deux méthodes de binarisation afin de montrer que les descripteurs peuvent permettre de déduire intuitivement la meilleure méthode de binarisation à utiliser.

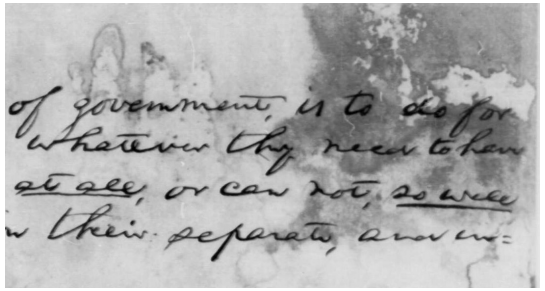
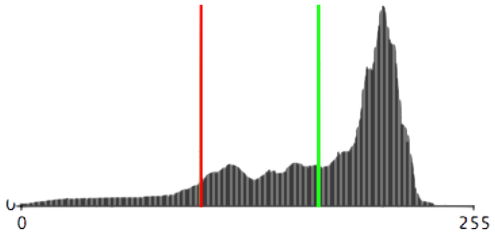
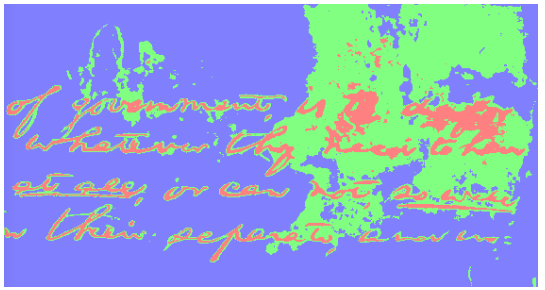
Description des images

Ces deux images sont issues du corpus DIBCO [GNP09] qui propose des images de documents fortement dégradées accompagnées d'une annotation contenant une binarisation manuelle (vérité-terrain). Chaque pixel d'encre a été labellisé par un expert. Nous comparons la binarisation produite par un algorithme à cette binarisation manuelle de référence en utilisant le f-score. Les méthodes de binarisation testées sont les plus utilisées par la communauté à savoir Otsu [Ots75] une méthode de seuillage globale et Sauvola [SP00] une méthode de binarisation locale (fenêtre glissante).

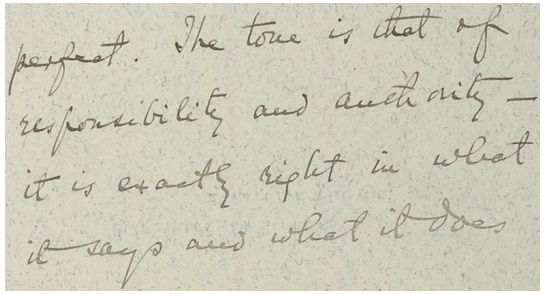
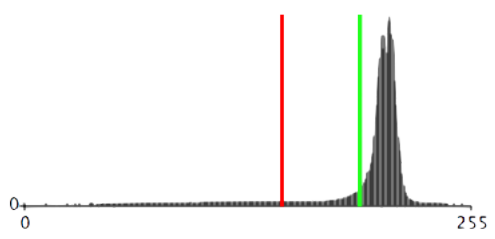
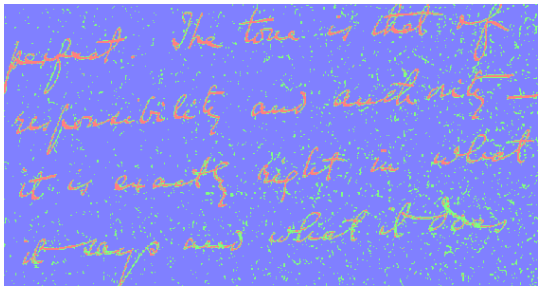
La première image (tableau 2.4 ligne 1) contient des taches très larges venant chevaucher partiellement des lignes de texte et les niveaux de gris de ces dernières sont proches de ceux du texte. Comme Otsu est une méthode de binarisation globale, il est probable que certains pixels de ces taches soient mal binarisés. Une méthode locale comme Sauvola a quant à elle plus de chance d'obtenir de bons résultats. Cela est confirmé par la valeur des f-score (0.4 pour Otsu et 0.7 pour Sauvola). La seconde image (tableau 2.4 ligne 2), présente un fond très bruité et la couleur de l'encre est très claire et proche de celle du fond. Sur cette image Sauvola n'est pas robuste aux bruits présents sur le fond et Otsu peut éventuellement avoir des pertes de précision dues aux faibles niveaux de gris de l'encre. Cela est confirmé par la valeur des f-scores (0.8 pour Otsu et 0.4 pour Sauvola).

Analyse des mesures

Même si les indicateurs statistiques génériques calculés sur l'histogramme décrivent une grande partie de la distribution des pixels des images, ils ne sont pas suffisants pour en déduire la meilleure méthode de binarisation. La moyenne des niveaux de gris d'encre (μ_I) de la première image est plus faible que pour la seconde image. Cela indique que la couche d'encre peut être plus facilement extraite avec une méthode globale (comme Otsu). Cependant, la valeur de s_I (skewness de la couche d'encre) est négative

Image	GrayScale Histogram	3-mean clusters
		

$\mathcal{MI}_I$	$\mathcal{MI}_B$	$\mathcal{MQ}$	$\mathcal{MA}$	$\mathcal{MS}$	$\mathcal{MSG}$	s_I	s_D	s_B	v_I	v_D	v_B	μ_I	μ_D	μ_B	s	v	μ
0.2	0.1	0.3	0.05	0.2	3,6	-0.4	-0.05	-0.5	741	392	161	66	135	199	-1.25	2065	171

Image	GrayScale Histogram	3-mean clusters
		

$\mathcal{MI}_I$	$\mathcal{MI}_B$	$\mathcal{MQ}$	$\mathcal{MA}$	$\mathcal{MS}$	$\mathcal{MSG}$	s_I	s_D	s_B	v_I	v_D	v_B	μ_I	μ_D	μ_B	s	v	μ
0.13	0.2	0.03	0.3	0.2	1.4	-0.6	-0.02	-0.5	257	206	30	98	146	189	-3	356	185

TABLE 2.4 – *Deux images de documents issues du corpus DIBCO et leur vecteur de 18 mesures. Les mesures proposées dans ce chapitre caractérisent différents types de dégradation (taches d'encre, encre effacée, bruits, . . .). En analysant manuellement ces mesures, il est possible de donner une indication quant au meilleur algorithme de binarisation possible.*

ce qui indique que la majorité des pixels d'encre sont répartis à droite de la distribution : il y a donc plus de pixels gris que de pixels très sombres. La *skewness* (s) du second histogramme est bien plus haute que celle de la première image. Cela indique que le fond de la seconde image peut être facilement extrait par un algorithme de seuillage global. Cela est aussi confirmé par la variance (v). Sans analyser les autres mesures, il semblerait, au premier abord, qu'une méthode de binarisation globale comme Otsu ait de meilleurs résultats sur la seconde image.

Sur la première image, les valeurs de $\mathcal{MI}_I$ et $\mathcal{MI}_B$ sont faibles. Cela indique qu'une méthode de binarisation globale a de fortes chances d'échouer. La valeur de $\mathcal{MSG}$ est aussi très élevée, ce qui indique que le document possède des taches très larges connectées au texte. Les méthodes de binarisation adaptative comme Sauvola ont généralement de meilleurs résultats sur ces types de dégradation.

Sur la seconde image, les valeurs de $\mathcal{MI}_I$ et $\mathcal{MI}_B$ sont encore plus faibles : Otsu risque de générer des erreurs. Mais d'autres mesures comme s ou la faible valeur de v indiquent que les erreurs d'Otsu seront assez faibles. De plus, la valeur de $\mathcal{MA}$ est très haute, ce qui indique que le document possède un grand nombre de composantes qui ne sont pas connectées au texte. Ce type de dégradation est connu pour entrainer des erreurs sur les méthodes de binarisation locales comme Sauvola.

L'analyse des mesures semble indiquer qu'il est préférable d'utiliser Sauvola pour la première image, et Otsu pour la seconde. Cela est confirmé par la valeur des f-score des deux méthodes. Le chapitre 3 détaillera comment utiliser nos descripteurs pour créer des modèles statistiques de prédiction du performances de plusieurs méthodes de binarisation.

2.3 Vers la création d'autres descripteurs de qualité : le cas de la transparence

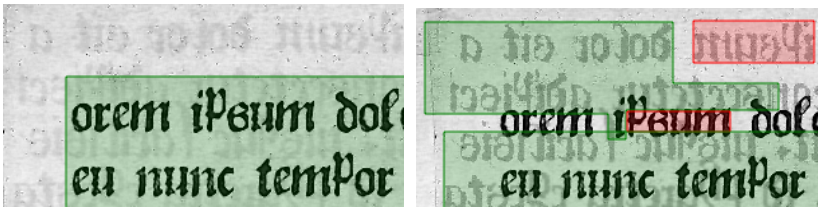
La transparence est une dégradation observée dans un grand nombre de documents anciens. L' analyse de résultats d'OCRs sur plusieurs images de documents contenant de la transparence, nous pousse à faire l'hypothèse que cette dégradation influe les performances de ce type de systèmes. Par exemple, la figure 2.14 montre le résultat d'Abbyy FineReader 9 [1] sur deux documents semi-synthétiques ou seuls, l'intensité des pixels de transparence varie. Les zones vertes sont des zones qu'Abbyy considère comme du texte, les zones rouges sont des zones pour lesquelles le moteur donne un faible taux de confiance. Plusieurs constatations peuvent être faites sur ces résultats :

– sur le document présenté en figure 2.14 où le niveau de transparence est faible, les résultats d'Abbyy sont de qualité.
– sur la figure 2.14b, on remarque que certaines composantes de transparence sont reconnues comme des composantes de texte et classées en zone verte.

Nous nous intéresserons, en titre de perspective, à l'étude de l'influence de la transparence sur les performances des systèmes de type OCR afin de prouver statistiquement en section 3.4 que la transparence diminue les performances des OCRs. Pour cela, il est nécessaire de disposer de descripteurs caractérisant la dégradation de type "transparence" afin de créer des modèles prédictifs du taux de reconnaissance OCR. L'étude de la précision des modèles créés permettra de conclure quant à l'influence de la transparence sur ces systèmes.

Comme la transparence est une dégradation fond-encre, les descripteurs proposés dans la section 2.2.1 peuvent en partie fournir une caractérisation de la transparence. Cependant, si on veut avoir une caractérisation fine de la transparence il est impératif de pouvoir distinguer les pixels de transparence des autres pixels de type dégradation fond-encre. Cela nécessite de spécialiser la brique d'extraction des pixels présentés dans le schéma 2.8. Si on dispose seulement des pixels de transparence, on remplacera dans nos mesures la classe “perturbation” par la classe “transparence”.

1. http ://www.abbyyeu.com/



(a) (b)

FIGURE 2.14 – *Analyse de l'influence sur l'OCR. À gauche un document semi-synthétique. À droite ce même document avec une transparence plus prononcée (modèle de dégradation présenté dans [MC09a]). L'analyse de structure physique d'Abbyy Fine Reader est exécutée sur les deux images. Les zones vertes correspondent aux zones de texte, les rouges aux zones sur lesquelles Abbyy hésite. Ces résultats montrent que certes, l'intensité de la transparence influe sur son résultat, mais aussi que les caractéristiques des composantes de transparence mettent en erreur la classification (zones de texte) faite par l'OCR.*

Dans cette section, nous nous concentrons sur une extraction fine de la transparence. Dans un premier temps, nous présentons les méthodes de modélisation et de restauration de la transparence qui utilisent à la fois le recto et le verso d'une même page. Ces méthodes ont la contrainte de devoir recaler ou aligner les pixels du recto et du verso. Les méthodes de recalage existantes sont peu performantes et longues en temps de calcul. C'est pour cela que nous proposerons, dans un second temps, une nouvelle méthode de recalage recto verso permettant d'identifier précisément les pixels de transparence.

2.3.1 Méthode de modélisation et de restauration de la transparence basée sur l'analyse du verso

La transparence de l'encre à travers un support papier est un phénomène complexe. G. Sharma [Sha01] modélise la transparence du verso en analysant le comportement de la lumière éclairant le document (figure 2.15). Il suppose qu'une partie de cette lumière traverse le papier, se réfléchit sur le dos du scanner avant de retraverser le papier. La lumière réfléchie arrive aux capteurs, mais les caractères du verso étant le plus souvent noir, ils provoquent une baisse de luminosité.

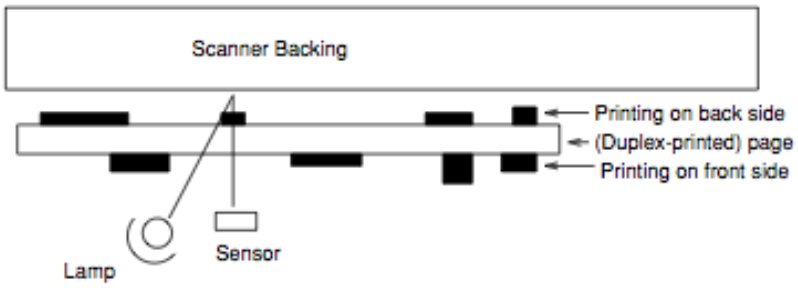


FIGURE 2.15 – *Modélisation de la transparence par la diffusion de la lumière dans le papier présentée dans [Sha01]*

Ce modèle est complexe à implémenter étant donné qu'un grand nombre de paramètres inconnus intervient par exemple le taux de reflechissement du dos du scanner.
Les auteurs de [Zi05] quant à eux se détachent du scanner pour modéliser la transparence. Pour générer un document transparent, ils combinent deux images. Une représente le recto et l'autre le verso. Ces deux images sont ensuite combinées. La combinaison des deux images se base sur l'équation suivante :

$$I_D = \theta\left(I_R, \mathcal{H} \otimes \mathcal{R}(I_V)\right) + N$$

Avec :
– I_D : l'image dégradée transparente.
– θ : fonction de transformation
– I_R : l'image recto.
– I_V : l'image verso.
– $\mathcal{H}$: une matrice de flou.
– $\mathcal{R}$: est une fonction permettant de retourner l'image horizontalement.
– N : paramètre permettant de moduler la quantité de bruit.
Comme le montre l'équation précédente, l'image du verso est tout d'abord retournée horizontalement puis un léger flou lui est appliqué. Ce flou permet de créer un effet de transparence plus réaliste. La transparence sera ajoutée par la fonction θ qui est définie en chaque pixel par :

$$\theta_{(i,j)}(I_R, B) = \begin{cases} I_{R(i,j)} - \alpha(I_{R(i,j)} - B(i,j) + n_{(i,j)}), & \text{si}(I_{R(i,j)} - I_B > \mathcal{T})) \\ I_R, & \text{sinon} \end{cases}$$

Avec :
– B : l'image ajoutée par transparence (dans notre cas $B = \mathcal{H} \otimes \mathcal{R}(I_V)$),
– α et $\mathcal{T}$ sont des paramètres permettant de contrôler le niveau de transparence.

FIGURE 2.16 – *Exemple de document transparent généré par la méthode présentée dans [Zi05]*

Cette méthode peut être facilement paramétrée afin de produire de larges collections d'images synthétiques. Dans la réalité, l'effet de transparence n'est pas toujours linéaire, car le support des documents anciens est souvent un papier très texturé. Sur ce type de support, l'encre du verso ne se diffusera pas de la même façon en fonction de la disposition et de l'épaisseur des fibres de papier. Un exemple de résultats de cette méthode est présenté en figure 2.16.
Les travaux de [MC09a] qui modélisent de manière générale les perturbations fond-encre peuvent s'adapter naturellement à la modélisation de la transparence. L'intérêt de ces travaux reposant sur la diffusion anisotropique est de permettre de modéliser de façon non linéaire la transparence.

Conformément à notre démarche, nous allons nous intéresser aux méthodes de restauration de la transparence.

Les méthodes de restauration peuvent être classées en deux catégories : celles n'utilisant pas le verso, et celles l'utilisant. Les premières telles que [MC09b, SG04a] ne se limitent pas à la transparence et corrigent l'ensemble des perturbations fond-encre. En effet, les niveaux de gris de la transparence sont très similaires à ceux des perturbations fond-encre. La transparence ne peut pas être extraite par une méthode de seuillage globale. La seconde famille de méthode de restauration [TSB07, NS02, DP01] propose une meilleure segmentation des pixels de transparence en utilisant le verso du document. Ces dernières supposent que les deux images sont recalées.

Nous allons donc maintenant nous intéresser aux méthodes de recalage afin d'obtenir une extraction fine des pixels de transparence.

2.3.2 Identification des pixels de transparence par recalage

Comme vue précédemment, la transparence résulte de la superposition de l'encre du verso sur recto. Une extraction précise de la transparence demande à identifier sur le recto les pixels du verso qui appartienne à l'encre. Comme les deux pages sont numérisées séparément, le verso peut être décalé ou tourné de quelques millimètres par rapport au recto. Comme les paramètres optiques et la configuration du scanner sont fixes, le verso ne subit généralement pas de transformation d'échelle. Le recalage du recto et du verso à pour but de trouver la transformation qui aligne au mieux les deux images.

Dans [Bro92] les auteurs présentent un certain nombre d'algorithmes ayant pour objectif de recaler un ensemble d'images dans le même système de coordonnées. La plupart de ces méthodes ne sont pas applicables au problème du recalage recto verso. En effet, le recto et le verso n'ont pas les mêmes niveaux de gris ni la même topologie : la seule information pertinente pour le recalage est la transparence qui est une version dégradée de l'encre du verso. Basées sur cette information, plusieurs méthodes de recalage recto verso ont été implémentées.
Dans [DP01, DD05] un algorithme d'optimisation est utilisé pour trouver la matrice de transformation minimisant la différence entre le recto et le verso retourné horizontalement. La différence entre les deux images est calculée par une fonction de coût basée sur la valeur de niveaux de gris de chaque pixel. Une seconde [MD09] méthode de recalage utilise la transformation de Fourrier-Mellin pour réduire les temps de calcul.
Ces deux méthodes sont comparées dans [TBS09] et les résultats montrent un léger avantage en précision à la première.
Il est à remarquer qu'aucune des deux méthodes ne proposent une mesure de confiance.

Le processus de recalage que nous présentons se base sur le fait que la transparence à la même structure physique que l'encre du verso (à un retournement horizontal près). Les deux structures ont les mêmes mots, lignes, paragraphes et figures.
Notre méthode se divise en plusieurs étapes (figure 2.17). Tout d'abord, nous utilisons la méthode de trinarisation précédente pour extraire les perturbations fond-encre sur le recto. La transparence étant présente dans les pixels extraits. La deuxième étape estime l'angle de rotation en appliquant un algorithme de redressement à la fois sur les pixels extraits lors de la première étape et le verso binarisé. La dernière étape permet d'extraire les profils horizontaux et verticaux des perturbations fond-encre sur le recto et de l'encre sur le verso. Ces quatre profils sont ensuite recalés deux à deux en utilisant un algorithme de Dynamic Time Warping (DTW) [Nie04].

L'objectif de notre méthode est de recaler le recto et le verso avec au moins une aussi bonne précision que les méthodes existantes, mais des temps de calcul bien inférieurs rendant leurs utilisations réalistes. De plus, nous proposons une mesure de confiance permettant de détecter les recalages dont la précision n'est pas suffisante. Ce coefficient de confiance peut aussi être utilisé comme descripteur pour mesurer la similarité entre la transparence du recto et l'encre du verso.

2.3.2.1 Identification des pixels de transparence et de bruits

Notre méthode est basée sur l'analyse de la structure physique des pixels de perturbation fond-encre du côté recto et de l'encre du côté verso (qui est retourné horizontalement). Les pixels d'encre du verso sont identifiés par une méthode de binarisation. Dans nos tests nous utilisons la méthode d'Otsu qui se montre suffisante pour nos besoins, mais d'autres méthodes peuvent être utilisées. Pour extraire la transparence, nous trinarisons le recto avec la méthode présentée en section 2.2.2.2. Cette méthode permet d'obtenir trois classes de pixels : les pixels d'encre, les pixels dégradés et les pixels du fond.

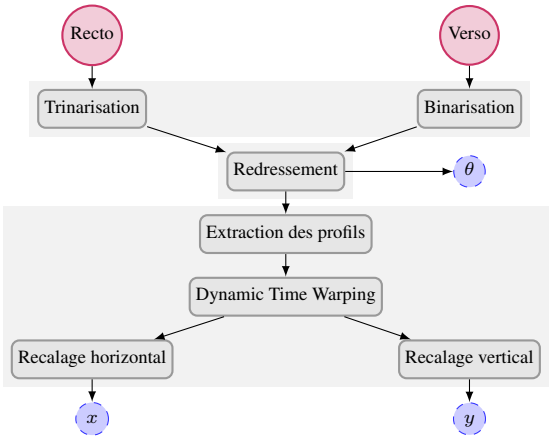
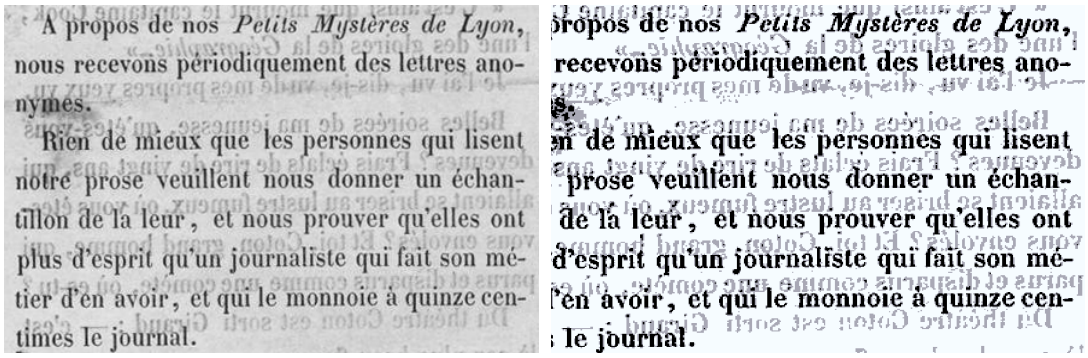


FIGURE 2.17 – *Schéma global de notre nouvelle méthode de recalage recto verso : les cercles en pointillés correspondent aux paramètres calculés de la transformation affine (θ le paramètre de rotation, x le décalage horizontal et y le décalage vertical)*

Un exemple de résultat de la trinarisation sur un document contenant de la transparence est présenté en figure 2.18.



(a) (b)

FIGURE 2.18 – *a. Extrait d'un document ancien contenant de la transparence, b. résultat de la méthode de trainarisation sur l'image de gauche : les pixels d'encre sont coloriés en noir, ceux de transparence en gris et ceux du fond en blanc.*

Comme on peut le voir sur la figure 2.18 la trinarisation du recto n'a pas besoin d'identifier parfaitement les pixels de transparence. Ce qui est important est juste d'identifier suffisamment de pixels pour ensuite extraire des informations sur les lignes et les colonnes de transparence.

2.3.2.2 Estimation de l'angle de rotation entre le recto et le verso

La seconde étape de notre méthode permet de recaler en rotation le recto et le verso. Plusieurs méthodes de redressement de document existent [Bre03, Pil01, BS01, MY99, JBWK99]. Ces méthodes estiment l'angle permettant de redresser les lignes de texte et travaillent en général sur une image binarisée. Par suite, elles peuvent être appliquées au verso binarisé afin de trouver son angle de redressement (θ_{verso}). Par contre, il n'est pas suffisant de les appliquer sur le recto binarisé pour trouver l'angle permettant un recalage entre les deux côtés de la page. En effet, les lignes d'encre et de transparence

peuvent avoir des orientations différentes sur les documents anciens comme le montre la figure 2.19. Pour contourner ce problème, nous appliquons une méthode de redressement non pas sur l'encre du recto, mais sur les pixels de perturbation fond-encre extraits par la trinarisation.

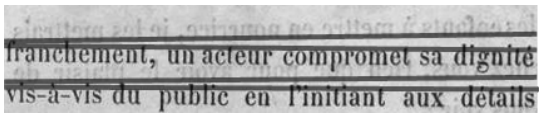


FIGURE 2.19 – *Les lignes de transparence et celles d'encre peuvent avoir des orientations différentes*

Étant donné que le recto et le verso sont redressés séparément, nous obtenons deux angles. Un pour le recto θ_{recto} et un pour le verso θ_{verso}. Il n'est pas nécessaire de recaler le verso, il suffit simplement de recaler le recto d'un angle θ égal à $\theta_{recto} - \theta_{verso}$. Dans nos tests, nous avons utilisé la méthode de redressement proposé par T. Breuel dans [Bre03] et implémentée dans la bibliothèque de fonction d'OCRopus. Un exemple du résultat de cette méthode sur le recto d'une image de documents anciens contenant des lignes, un titre et une figure sont présentés en figure 2.20.

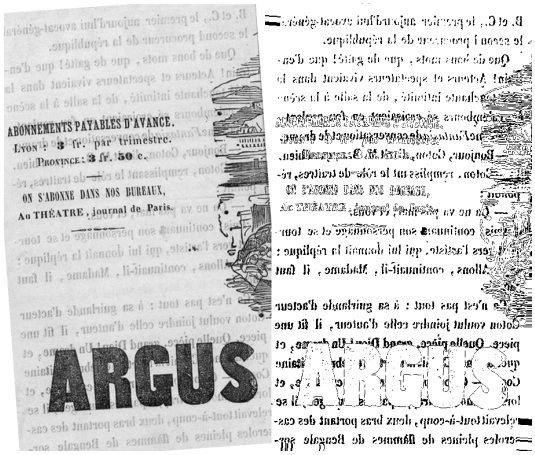
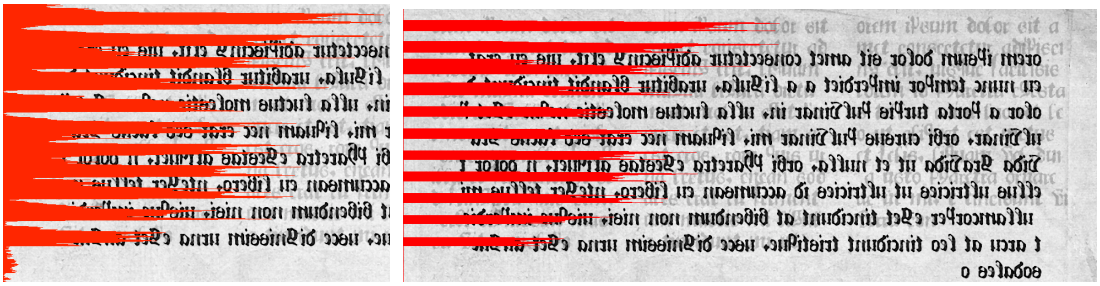


FIGURE 2.20 – *Exemple du redressement de la transparence d'un recto en utilisant la méthode proposée par T. Breuel [Bre03] sur les pixels de transparence extraits lors de l'étape de trinarisation du recto.*

2.3.2.3 Identification des lignes et des colonnes de texte pour l'alignement vertical et horizontal

Cette étape a pour objectif de calculer les décalages horizontaux et verticaux entre le recto et le verso d'une page. Premièrement, les lignes de transparence du recto et celles d'encre du verso sont extraites puis alignées. Puis, la même technique est appliquée aux colonnes de transparence du recto et celles d'encre du verso.

Dans [LSZT07] les auteurs proposent un inventaire de plusieurs algorithmes d'extraction de lignes dont l'algorithme défini, dans [MS99]. Cet algorithme se base sur les profils verticaux. Les profils verticaux sont obtenus en comptant les pixels (correspondant à de la transparence pour le recto et de l'encre pour le recto) le long de l'axe horizontal pour chaque ligne de l'image. Un exemple montrant les profils verticaux de transparence est présenté en figure 2.21a. Étant donné que le calcul de profils verticaux est robuste à la fragmentation des caractères il convient parfaitement à la transparence dont les composantes sont très fragmentées.



(a) (b)

FIGURE 2.21 – *Profils verticaux sur un document semi-synthétique. a. le profil brut. b. le profil filtré permettant de mettre en avant les lignes de transparence.*

Les profils verticaux de transparence (recto) et d'encre (verso) sont très similaires et le recalage vertical peut être réalisé sur ces deux histogrammes. Néanmoins, la précision de cette étape dépend de la quantité de pixels gris qui ne sont pas de la transparence (taches, bruits). Or, ces pixels sont comptabilisés lors du calcul du profil vertical de la transparence. Pour obtenir une meilleure précision, il est nécessaire de nettoyer les profils du recto. Pour cela, chaque pic du profil en dessous d'un seuil T_0 est remplacé par 0. Le seuil T_0 a, en effet, pour objectif de classer les pics du profil en deux ensembles : les pics résultants des lignes ou de grandes taches (profils[i] > T_0) et ceux résultants du bruit (profils[i] < T_0). Par suite, ce processus de seuillage permet d'enlever tous les pics correspondant aux bruits de l'image. L'espace interligne de transparence est alors plus marqué, ce qui améliore la précision du recalage.

Nos tests montrent que la meilleure valeur pour T_0 est la moyenne trimée. La moyenne trimée est calculée en enlevant 5% des plus petits et plus grands pics de l'histogramme lors du calcul de la moyenne. Autrement dit nous supposons que 5% des pics de l'histogramme peuvent être des valeurs aberrantes. Un exemple du résultat de ce filtrage est présent en figure 2.21b.

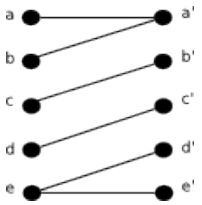


FIGURE 2.22 – *L'analyse de la matrice de correspondance obtenue par DTW montre des pics appariés* $\{(b, a'), (c, b'), (d, c'), (e, d')\}$*, insérés* $\{(a, a')\}$ *ou supprimés* $\{(e, e')\}$.

Maintenant que les deux profils verticaux (celui de transparence et celui de l'encre) ont été filtrés en supprimant les valeurs aberrantes, nous proposons de calculer l'alignement optimal minimisant la différence entre ces deux histogrammes. Cela peut être réalisé par l'algorithme Dynamic Time Warping [Nie04]. L'algorithme DTW mesure la similarité entre deux séquences qui peut varier en temps et en fréquence. Le résultat de cet algorithme peut être analysé pour trouver le meilleur alignement entre les deux séquences (ici les deux histogrammes). La fonction de distance donnée à l'algorithme DTW doit évaluer la similarité entre deux valeurs des deux séquences. Dans notre cas, la similarité entre deux pics est égale à la différence arithmétique :

$$d(i, j) = |R[i] - V[j]|$$

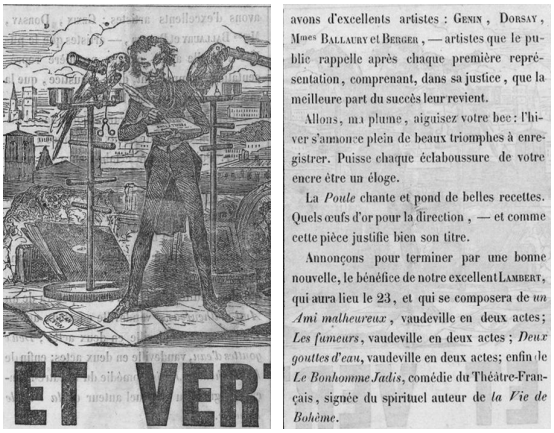
Avec R le profil de la transparence (recto), V le profil de l'encre (verso), $i, j \in \{0...\mathrm{H}\}$, H correspondant à la hauteur de l'image (taille du profil verticale).

Comme le montre la figure 2.22, l'algorithme de DTW considère que les pics peuvent être insérés, supprimés ou appariés. Le décalage entre les deux histogrammes n'est donc pas forcement constant sur tout l'histogramme. Pour répondre à ce problème, nous avons décidé de calculer le décalage moyen sur tous les pics appariés (soit les couples $\{(b, a'), (c, b'), (d, c'), (e, d')\}$ sur la figure 2.22). Ce décalage moyen correspond au paramètre y du recalage vertical.

Pour calculer le décalage horizontal, nous appliquons la même technique aux profils horizontaux qui correspondent aux colonnes de texte de l'image de document. Par contre, la précision de ce recalage dépend du nombre de colonnes du document. Plus le nombre de colonnes est important dans le document, plus le recalage est précis. Le cas le plus imprécis est lorsqu'il n'y a qu'une seule colonne.

2.3.2.4 Comparaisons avec l'état de l'art

Pour nous comparer avec l'état de l'art, nous avons réalisé deux types de tests. Tout d'abord, des tests visuels sur des documents anciens réels, puis pour obtenir des statistiques précises, des tests sur des documents semi-synthétiques.



(a) (b)

FIGURE 2.23 – *Comparaison de notre méthode avec celle présentée dans [DP01] sur un document ancien contenant de la transparence (à gauche le recto, à droite le verso). La registration manuelle donne la transformation affine suivante :* $x = 4$, $y = 1$, $\theta = 1$ *Notre méthode :* $x = 5$, $y = 2$, $\theta = 1,3$ *la méthode présentée dans [DP01] :* $x = -34$, $y = -40$, $\theta = 1$

À notre connaissance, il n'existe pas de corpus libres disposant d'une vérité-terrain sur le recalage. Par suite, afin de tester notre méthode, nous avons manuellement recalé des documents contenant de la transparence et issue de la collection BML [2]. Les deux méthodes présentent des résultats similaires. Mais dans certains cas, la méthode présentée dans [DP01] a généré d'importantes erreurs (jusqu'à environ 40 pixels, avec une hauteur de ligne de 13 pixels environ) tandis que les résultats de notre méthode sont plus proches de ceux obtenus par recalage manuel (environ une dizaine de pixels). Un exemple de ce type de document est présenté en figure 2.23. Les erreurs obtenues par [DP01] peuvent être expliquées par le fait que comme cette dernière se base sur l'intensité des pixels de transparence et d'encre, si l'intensité de la transparence (gris clair) est trop éloignée de ceux de l'encre (noire) la transparence est assimilée à du fond. Cela pénalise la mise en correspondance.

2. http ://collections.bm-lyon.fr

Afin de faire des tests à plus grande échelle, nous avons aussi utilisé des documents semi-synthétiques. Ceci nous permet de créer des documents, avec différents niveaux de transparence, différentes quantités de texte, différentes lignes et colonnes. Pour constituer la base, nous avons utilisé le logiciel présenté dans [JVD+10]. Quarante-huit rectos ont été ainsi générés. À chacun de ces rectos, nous appliquons un modèle de génération de transparence [MC09a] de façon à obtenir neuf versions du recto avec des niveaux de transparence différents (la moyenne des niveaux de gris de transparence varie entre 245 et 159). Cela fait un total de 432 documents. À chacune de ces images, nous appliquons une transformation affine dont les paramètres θ, x et y sont choisies aléatoirement dans un intervalle correspondant à des cas réels (pour x et y nous pouvons avoir un décalage maximum de 15% de la taille de l'image, pour θ un angle maximum de 20 degrés).

Méthode de recalage	Erreur en rotation			
	Max	Min	Moyenne	Écart-type
Notre Methode	0.25	-0.03	0.15	0.06
La méthode [DP01]	18	0	7.19	4.45
	Erreur horizontale			
	Max	Min	Moyenne	Écart-type
Notre Methode	11	0	1.17	2.10
La méthode [DP01]	39	0	2.04	6.77
	Erreur Verticale			
	Max	Min	Moyenne	Écart-type
Notre Methode	1	0	0.51	0.53
La méthode [DP01]	38	0	1.81	5.04
	Temps moyen de calcul			
Notre méthode	12s			
La méthode [DP01]	598s			

TABLE 2.5 – *Comparaison de la précision avec la méthode présentée dans [DP01]. Les deux méthodes sont implémentées en C++ et testées sur un ordinateur disposant de 8Go 1067 MHz DDR3 et un processeur Intel Core i7 @ 2.8 Ghz. La taille moyenne d'une image est* $2000 * 2811$ *pixels*

Les résultats présentés en table 2.5 confirment les résultats obtenus visuellement sur les documents réels. Les deux méthodes présentent une erreur moyenne assez similaire avec un léger avantage pour la nôtre. C'est dans les cas difficiles que notre méthode montre une meilleure estimation des paramètres. En effet, l'erreur horizontale maximale pour notre méthode est de 11 pixels et de 39 pixels pour la méthode de l'état de l'art. Nous sommes encore plus précis sur le recalage des lignes (recalage vertical) avec une erreur maximale de 1 pixel contre 38 pour [DP01].

De plus les temps de calcul sont améliorés par un facteur 50 (12 secondes de moyenne pour notre méthode contre 598 secondes pour [DP01]). Les deux méthodes ayant été implémentées sans optimisations particulières.

2.3.2.5 Le cas des transformations non linéaires

La méthode présentée montre de bons résultats par rapport à la méthode présentée dans [DP01]. Néanmoins un problème persiste : certains documents anciens sont courbés dû aux usages du temps (humidité, pliures, ...). Dans ces cas, la transformation nécessaire au recalage du recto et du verso ne peut être affine. Pour minimiser ce problème, il serait possible de recaler le document en le découpant en plusieurs zones et d'appliquer notre méthode de recalage sur ces zones. Certes, les transformations obtenues sur le document seraient toujours affines, mais le décalage restant serait minimisé. Cette technique a été testée et semble montrer de bons résultats, mais nous avons encore du mal à trouver un moyen automatique pour identifier les zones à recaler ainsi que pour faire l'interpolation des pixels soumis à plusieurs transformations.

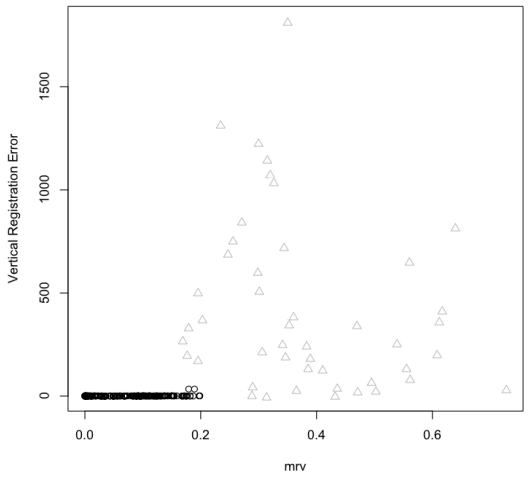


FIGURE 2.24 – *La mesure m_{rv} mise en relation avec l'erreur verticale du recalage. Les cercles correspondent à des documents contenant de la transparence, les triangles correspondent à des documents sans transparence. On remarque qu'au-dessus d'un certain seuil (ici 0.19), on ne peut plus garantir la précision du recalage. De plus les documents dont la mesure m_{rv} est supérieure à 0.19 ne sont pas transparents. Par conséquent, m_{rv} peut être utilisé comme un descripteur permettant de mesurer si le document contient de la transparence.*

2.3.3 Conclusion sur notre méthode de recalage recto verso

Notre méthode répond pour l'instant à nos besoins de précision et de temps de calcul, mais ne propose pas encore de système permettant de savoir si le recalage est pertinent. Le recalage ne sera pertinent que si la transparence est suffisamment présente sur le document. Comme notre algorithme se base sur la présence de la transparence, il est nécessaire de mesurer la pertinence des résultats du recalage en fonction de la présence ou non de la transparence.

Pour cela nous utilisons l'erreur résultante de la DTW afin d'obtenir une mesure de confiance que nous appelons m_{rv}. m_{rv} peut-être utilisé pour détecter les images ne contenant pas assez de transparence pour être recalées précisément. Nous avons étudié cette mesure sur l'ensemble de notre corpus de documents semi-synthétique. Pour mener cette étude, nous avons rajouté dans le corpus précédent 48 images ne contenant pas de transparence. La figure 2.24 montre la mesure m_{rv} en relation avec l'erreur du recalage vertical. Comme on peut le constater, la mesure peut être utilisée comme un seuil pour garantir la précision du recalage. Sur notre corpus de document, le recalage est presque parfait si la valeur de m_{rv} est inférieur à 0.19.

2.4 Conclusion du chapitre

Dans ce chapitre nous avons présenté notre méthodologie pour la définition de descripteurs permettant de caractériser certaines dégradations présentes sur les images de documents anciens. Cette méthodologie peut être divisée en trois grandes étapes : la sélection des critères influant sur les résultats d'un type d'algorithme, l'extraction des pixels concernés par la dégradation puis pour finir la définition de descripteurs correspondant aux critères précédemment retenus.

Cette méthodologie est appliquée aux dégradations de type perturbation fond-encre dans l'objectif prédire les erreurs des algorithmes de binarisation. Trois familles de descripteurs sont ainsi utilisées :

1. les descripteurs génériques globaux (moments de l'histogramme des niveaux de gris) applicables

en dehors du contexte d'images de documents,

2. trois descripteurs globaux dédiés au document et mesurant l'intensité de la dégradation en fonction de celle de l'encre et du fond ainsi que la quantité de pixels de dégradation en fonction de la quantité de pixels d'encre,
3. trois descripteurs locaux mesurant la taille et la position des composantes connexes des perturbations fond-encre en fonction de celles du texte.

En guise de perspectives, cette même méthodologie est appliquée au défaut de transparence. La transparence étant une dégradation incluse dans les perturbations fond-encre, les mêmes descripteurs précédemment définis sont utilisés pour la caractériser. Néanmoins, nous proposons une nouvelle méthode d'extraction des pixels de transparence qui s'appuie sur le recalage du recto et du verso d'une même page. Cette méthode d'extraction de pixels se montre en moyenne aussi précise, mais 50 fois plus rapide que la méthode de l'état de l'art. Les erreurs maximales (cas critiques) sont aussi inférieures.

Ces descripteurs sont définis avec l'objectif de prédire le résultat de certains algorithmes d'analyse et de traitement des images de documents. Nous expliquerons dans le chapitre suivant comment utiliser ces descripteurs dans le but de prédire le résultat d'algorithmes. Nous présenterons un exemple d'utilisation sur les algorithmes de binarisation et analyserons la faisabilité de cette même méthode sur deux algorithmes d'OCRs.

Chapitre 3

Prédiction de résultats d'algorithmes de traitement d'images de documents

Dans le chapitre précédent, nous avons présenté des descripteurs pouvant caractériser la qualité d'une image de document en niveau de gris. Dans ce chapitre, nous montrerons comment ces derniers peuvent être utilisés pour prédire le résultat d'algorithmes d'analyse et de traitement d'images de documents. Notre objectif est de pouvoir quantifier numériquement les performances d'un algorithme (que l'on nomera *al* par la suite) sur une nouvelle image. Ces travaux s'inscrivent dans le cadre de l'apprentissage supervisé. La démarche générale sera constituée de deux parties : la première consiste à créer un corpus d'apprentissage qui associe les performances d'un algorithme à un ensemble d'images de référence ; la deuxième étape consiste à fournir un modèle capable de prédire les performances d'un algorithme sur une nouvelle image. Notre démarche schématisée en figure 3.1 peut être résumée par les étapes suivantes :

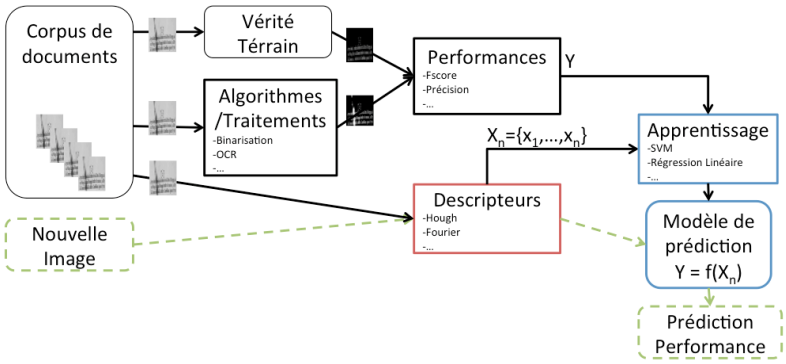


FIGURE 3.1 – Prédiction des performances d'algorithmes de traitements d'images de documents - *Création du modèle prédictif par apprentissage (en bleu) et prédiction des performances sur de nouvelles images (en vert) : l'ensemble des valeurs des descripteurs (en rouge) du chapitre précédent et des performances d'un algorithme calculées sur un corpus de documents disposant d'une vérité-terrain sont utilisées pour "apprendre" un modèle prédictif (en bleu). Le modèle prédictif peut ensuite être utilisé sur des nouvelles images de documents (en vert).*

1. **Création du corpus d'apprentissage dédié aux performances d'un algorithme.** Les données d'entrées de cette partie sont un ensemble d'images annotées représentant les résultats que devrait fournir l'algorithme *al* sur une image. Suivant l'algorithme, les annotations peuvent porter sur la globalité de l'image ou sur certaines parties de l'image (annotations locales). Ensuite nous appliquons l'algorithme *al* sur l'ensemble des images annotées. Nous pouvons ainsi construire un nouveau corpus d'apprentissage constitué de l'association d'une image, et d'un vecteur numérique (par exemple le f-score) représentant les performances de l'algorithme.
2. **Création du modèle de prédiction.** La deuxième étape, est une étape classique d'apprentissage

supervisé qui a pour objectif de fournir un modèle de prédiction basé sur un ensemble de descripteurs et dont la base d'apprentissage a été définie à la première étape.

La première section (3.1) de ce chapitre est consacrée à la présentation d'algorithmes permettant de créer des modèles prédictifs par apprentissage. Plus particulièrement dans le domaine de l'OCR, nous détaillerons les approches similaires à la nôtre permettant de prédire les performances de ce type d'algorithmes. Cela nous permet de proposer en sous-section 3.2 une méthodologie rigoureuse permettant la création de modèles prédictifs.

Cette méthodologie sera validée par la création de modèles de prédiction pour plusieurs algorithmes de binarisation en section 3.3, ainsi que pour prédire les résultats de deux OCRs en fonction du défaut de transparence (visibilité du verso à travers le recto) en section 3.4.

3.1 La création d'un modèle statistique de prédiction

Cette section présente différents articles proposant, soit de prédire les performances d'un algorithme, soit de classer les futurs résultats d'algorithmes d'analyse et de traitement d'images de documents. Ses différents articles reposent sur des algorithmes d'apprentissage supervisés et utilisent la même démarche que celle présentée en figure 3.1. Ainsi, avant de présenter les résultats obtenus sur la prédiction d'algorithmes, nous présenterons les algorithmes d'apprentissages les plus couramment utilisés.

3.1.1 Algorithmes d'apprentissage supervisés

Il existe un très grand nombre d'algorithmes d'apprentissage supervisé : SVM (Support Vector Machine), réseaux de neurones, arbre de décisions, k-means, modèles de markov, classification ascendante hiérarchique, etc. Ces algorithmes sont utilisés dans de nombreux domaines comme celui de la reconnaissance de formes, de textes ou encore de paroles. Des états de l'art sont présentés dans de nombreux ouvrages ou articles [CM11, Vap99, B$^+$06]. On peut classer ces algorithmes en fonction de la classe de problèmes auxquels ils s'appliquent : la tâche de régression et la tâche de classification. Ces deux tâches sont différentes par leurs sorties :

– Les régresseurs proposent des fonctions numériques qui sont en général continues. Par exemple, un régresseur est capable de fournir un résultat représentant le pourcentage d'erreurs d'un OCR (entre 0% et 100%).
– Les classifieurs proposent de quantifier l'appartenance à une classe de résultats. Les réponses peuvent être sous forme d'un indice représentant l'index de la classe soit sous forme de probabilité d'appartenance à une classe. Par exemple, dans le cadre d'un OCR, ils permettent de décider si le pourcentage d'erreur peut être : bon ($> 90\%$), moyen ($> 50\% \leq 90\%$) ou mauvais ($< 50\%$).

Certaines méthodes d'apprentissage supervisées peuvent être utilisées tant comme classifieurs que régresseurs. Le choix de l'une ou de l'autre catégorie peut se faire en fonction des besoins.

Nous nous concentrons dans cette sous-section sur les algorithmes d'apprentissage utilisés dans la littérature pour la création de modèles de prédiction d'algorithmes de traitements et d'analyse d'images de document :

SVM (Support Vector Machine) : Développés dans les années 90 [Vap99], les SVMs sont intéressants, car ils positionnent au mieux le plan de séparation de deux classes et offrent la possibilité de mieux séparer les classes en montant en dimension l'espace des paramètres. Les SVM peuvent être utilisés pour des tâches de classification ou des tâches de régression.

Réseaux de neurones : De manière générale, un réseau de neurones est un modèle de prédiction. Leur conception est inspirée de la biologie. En effet, ils modélisent mathématiquement les connexions

neuronales du cerveau qui permettent à l'homme de raisonner. Ces derniers peuvent être utilisés tant pour des tâches de régression que pour des tâches de classification.

Arbres de décisions : Les arbres de décisions sont issus du domaine de l'aide à la décision. Ils permettent de prédire les valeurs d'une variable de sortie (une classe) à l'aide de différents descripteurs. Ce type de modèle prédictif est couramment utilisé en raison de sa lisibilité et de sa capacité à sélectionner automatiquement les variables discriminantes. Chaque sommet de l'arbre de décisions décrit la distribution de la variable à prédire en fonction des observations données en entrée. Les arrêtes (connexions entre les sommets) représentent les conditions permettant de répartir les observations du sommet vers d'autres sommets.

Classifieurs Bayesien : Les classifieurs Bayesien permettent de réaliser une tâche de classification en se basant sur la loi de distribution des informations (espérance, variance) pour créer un modèle probabiliste.

KNN (k-nearest neighbor) : KNN est un algorithme d'apprentissage supervisé permettant de répondre aux problèmes de classification et de régression. Ce dernier se base sur la distance, dans l'espace des descripteurs, des observations d'apprentissage.

Modèles de Markov : Les modèles de Markov sont des automates probabilistes. Un sommet de l'automate représentant un état, chaque arrête représente un changement d'état avec une probabilité calculée. Ce formalisme permet de résoudre des tâches de classification comme de régression.

3.1.2 Les modèles de prédiction des performances d'algorithmes existants de traitement et d'analyse d'images de documents

Dans le domaine du traitement de l'image de document, plusieurs travaux de recherche [BKN95, CHKW97, GKN98] proposent une démarche similaire à la nôtre. Ces travaux portent exclusivement sur le domaine de l'OCR et ont pour objectif soit de prédire leurs performances, soit de sélectionner le meilleur OCR pour une image donnée, soit d'améliorer leurs résultats en sélectionnant les méthodes de restauration les plus adaptées aux images. La plupart de ces articles se basent sur des descripteurs extraits à partir d'images binaires. Les descripteurs utilisés ont déjà été présentés en section 2.1.3.1.

Nous divisons cet état de l'art en trois parties. Premièrement nous présentons les initiatives dont l'objectif est de prédire les performances des OCRs. Deuxièmement, les initiatives permettant de sélectionner automatiquement l'OCR le plus adapté à une image. Et pour finir, les initiatives, dont les objectifs, sont la sélection automatique de méthodes de restauration pour améliorer les résultats des OCRs.

3.1.2.1 Modèles de prédiction des performances d'OCRs

Pour prédire les résultats de l'OCR, les auteurs de [BKN95], se basent sur un ensemble de descripteurs binaires (section 2.1.3.1). Les auteurs utilisent deux corpus de documents : le premier contient 460 images binaires, le second contient 200 images en niveaux de gris. Les descripteurs sélectionnés ne s'utilisant que sur des images de documents binaires, le second corpus a été binarisé avec un seuil global (sur l'histogramme des niveaux de gris) fixé à 127. La classification des résultats de l'OCR se réalise en deux classes : les bons résultats ($\geq 90\%$) et les mauvais résultats ($< 90\%$). Pour cela deux méthodes de classification sont utilisées et comparées. La première méthode de classification utilise un arbre de décisions pour séparer les données (droites en pointillées sur la figure 3.2). La seconde se base sur un classifieur statistique pour obtenir des courbes non linéaires séparant les données (courbes pleines sur la figure 3.2).

La première méthode de classification utilise trois différentes règles de décisions (arbre de décisions) présentées sur l'arbre de décisions 1. Ces règles de décisions ont été réalisées par les auteurs en étudiant le comportement des descripteurs vis-à-vis du résultat de l'OCR. Cette méthode de classification présente une erreur de 15% (15 images de mauvaise qualité ont été classées comme bonnes et 53 bonnes ont été classées comme mauvaises) sur le premier corpus d'images de documents. Les erreurs de classifications de type mauvaise qualité vers bonne qualité ont été attentivement étudiées par les auteurs. Ces derniers concluent que ces images contenaient toutes des tableaux, formules mathématiques, figures. L'OCR montre de mauvais résultats dus à la complexité d'analyse de ces images et non dus à leurs qualités. Les résultats de cette méthode de prédiction sur le second corpus sont similaires (erreur de 13,5%).

Arbre de décisions 1 *[BKN95]*

quality=quality_is_good
si WSF (White Speckle Factor) ≥ 0.1 **alors**
  quality=quality_is_poor
fin si
si BCF (Brocken Character Factor) > 0.7 **alors**
  quality=quality_is_poor
fin si
si max(moyenne des largeurs des composantes blanches, moyenne des hauteurs des composantes blanches) ET $\frac{\|\text{composantes noires}\|}{\|\text{composantes blanches}\|} < 1.5$ **alors**
  quality=quality_is_poor
fin si

La seconde méthode de prédiction utilisée se base sur un classifieur statistique (nearest mean classifier) utilisant la distance de Mahalanobis. L'objectif de cette classification est de s'affranchir de la contrainte linéaire formulée par les règles 1 et 2 de la première méthode. La distance de Mahalanobis se différencie de la distance euclidienne par le fait qu'elle prend en compte la corrélation d'une série de données (ici BCF et WSF). La figure 3.2 montre les frontières de décision ainsi obtenues. Les résultats de cette méthode de prédiction sont similaires.

Deux conclusions sont formulées par les auteurs. Tout d'abord, ils estiment qu'il est nécessaire de trouver d'autres descripteurs de qualité afin de mieux décrire la qualité d'une image avant de pouvoir utiliser des méthodes de prédictions statistiques. Deuxièmement, la qualité de l'image n'est pas le seul facteur des baisses de performance de l'OCR, la complexité (en terme de contenu et de mise en page) est aussi à l'origine d'un grand nombre d'erreurs or, elle n'est pas évaluée par les descripteurs utilisés.

Dans [CHKW97] les descripteurs Speckle [1], TCF [2] et WSF [3] sont utilisés conjointement pour prédire les résultats d'un OCR à l'aide d'une régression linéaire. Afin de contrôler le taux de dégradation des documents, les auteurs utilisent des documents semi-synthétiques. Chaque document du corpus est imprimé, puis photocopié et re numériser dix fois. Ainsi, plus le document est imprimé et photocopié, plus il est susceptible de présenter des dégradations. Au total, le corpus contient 18 images de documents.

Les auteurs de cet article font l'hypothèse que ces descripteurs peuvent être combinés linéairement pour prédire le taux d'erreur de l'OCR de la façon suivante :

$$z = \alpha * Speckle + \beta * TCF + \delta * WSF + \lambda$$

1. nombre de composantes noires de petite taille
2. Touching Character Factor
3. White Speckle factor

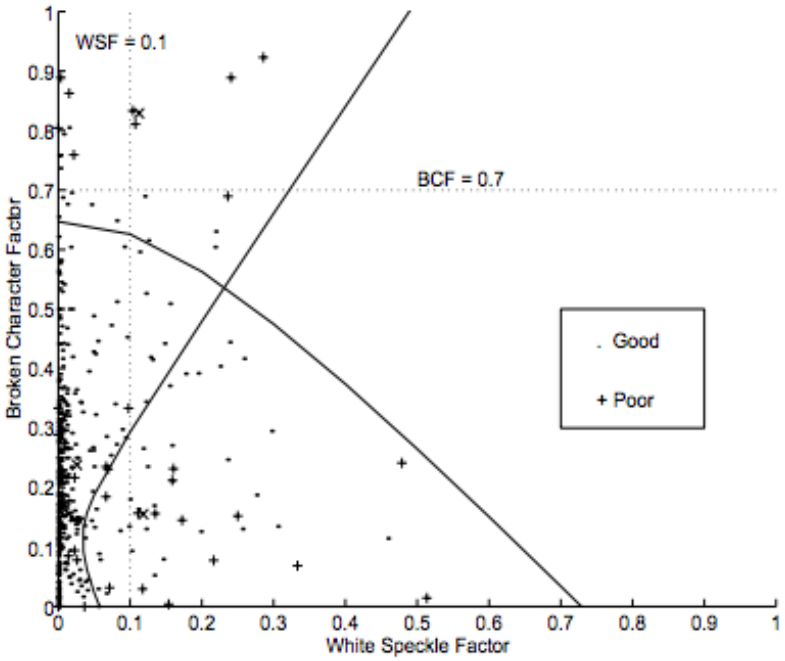


FIGURE 3.2 – *Droites et courbes représentant l'arbre de décisions construit dans [BKN95] pour prédire les performances d'OCRs.*

Avec, z le taux d'erreur de l'OCR, α, β , δ, λ sont calculés à partir d'un corpus d'entrainement en résolvant le système d'équations suivant :

$$\begin{aligned} z_0 &= \alpha * Speckle_0 + \beta * TCF_0 + \delta * WSF_0 + \lambda \\ \vdots &= \vdots \\ z_i &= \alpha * Speckle_i + \beta * TCF_i + \delta * WSF_i + \lambda \\ \vdots &= \vdots \\ z_9 &= \alpha * Speckle_9 + \beta * TCF_9 + \delta * WSF_9 + \lambda \end{aligned}$$

z_i représente la génération de photocopie d'un document ($i \in (0...9)$). Ce système est surdimensionné (6 équations, 3 inconnues) et en général n'admet pas de solution sauf cas particulier. Une manière de résoudre ce problème est de minimiser l'erreur à la moyenne au carré afin trouver α, β , δ, λ. La figure 3.3 compare le résultat de la prédiction (en pointillé) et le résultat attendu en fonction du nombre de photocopies appliquées au document.

Les résultats de cette méthode semblent "bons". Néanmoins plusieurs remarques sont à faire. Premièrement, les auteurs utilisent des documents semi-synthétiques ou seules des dégradations mesurées par Speckle, TCF et WSF interviennent. Deuxièmement, aucune étude n'est réalisée quant à la pertinence (sur le modèle final) des différents descripteurs. Certains peuvent n'apporter que très peu de précision sur le modèle. Troisièmement, le modèle n'est pas validé statistiquement sur un échantillon du corpus de document. En effet, le modèle semble être entrainé sur 100% du corpus et testé sur ces mêmes documents.

Les auteurs de [GKN98] utilisent la plupart des descripteurs présentés en section 2.1.3.1 toujours dans l'objectif de prédire l'OCR, mais cette fois-ci en utilisant un réseau de neurones (perceptron entrainé par rétro propagation). Le corpus d'images utilisé est ici constitué manuellement avec des zones de texte

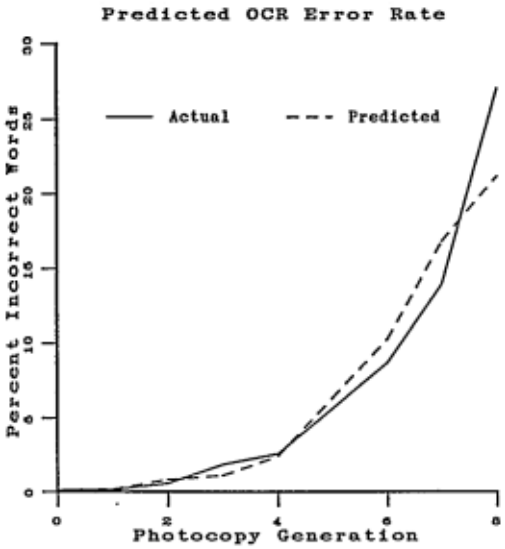


FIGURE 3.3 – *Prédiction de l'OCR proposée dans [CHKW97]. La courbe pleine représente le vrai taux d'erreur, la courbe en pointillés les taux d'erreur prédis.*

extraites de différentes images de documents. La sortie du réseau de neurones est un chiffre comprit entre 0 et 1 et correspondant au taux de reconnaissance OCR prédits pour une zone de texte donnée. Le résultat du réseau de neurones est ensuite classé en deux classes : mauvais (taux inférieur à 90%) et bon (taux supérieur à 90%).

Comme on peut le voir sur la matrice de confusion présentée en table 3.1, seulement 9 images de mauvaise qualité (résultats de l'OCR $< 90\%$) sur 23 ont été bien classées et deux images classées comme mauvaises étaient en faite de bonne qualité. Néanmoins, cette méthode est précise pour prédire les cas ou l'OCR réalise de bons résultats (99% de bonne prédiction). Ces résultats peuvent être expliqués par la faible quantité de documents dégradés (inférieur à 5%) présents dans le corpus.

	Prédiction	
	mauvais	bon
mauvais	9	14
bon	2	477

TABLE 3.1 – *Matrice présentant les résultats de la prédiction d'OCRs en deux classes (bon et mauvais) proposées dans [GKN98].*

3.1.2.2 Sélection automatique du meilleur OCR pour une image donnée

Un autre besoin est de pouvoir en fonction d'une image donnée de sélectionner l'OCR qui donnera les meilleurs résultats. Si l'on est capable de prédire les résultats des OCRs, on peut bien évidemment sélectionner l'OCR qui donne les meilleurs résultats. Toujours basée sur des descripteurs binaires une approche différente a été proposée par les auteurs de [APSS03]. Ce système se décompose en plusieurs étapes (figure 3.4) :

Tout d'abord, un classifieur est utilisé pour séparer les images de documents qui, globalement, contiennent des caractères cassés (Broken) ou pas. Pour les images qui n'appartiennent pas à la classe "Broken", un

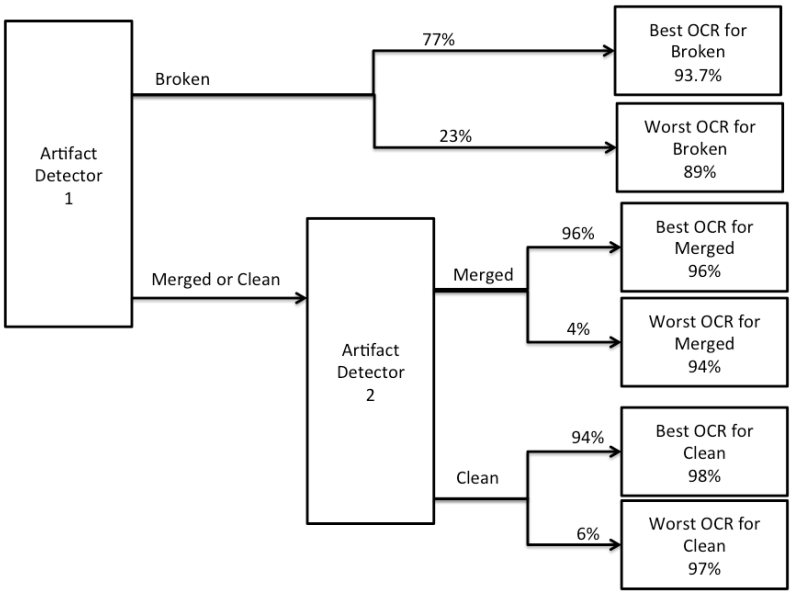


FIGURE 3.4 – *Système de sélection du meilleur OCR en fonction de la qualité des caractères d'une page présenté dans [APSS03].*

deuxième classifieur est utilisé pour distinguer les images contenant des caractères fusionnés des autres images. À la fin de cette étape, les documents sont donc classés en 3 classes : cassé, fusionné, sans dégradations. Chaque classifieurs utilisé dans cette étape se base sur les k plus proches voisins (KNN) avec k égale à 2. Les auteurs constituent leurs corpus de vérités-terrains à l'aide de deux Fax. En effet, les auteurs remarquent que selon le fax utilisé, les caractères du document sont plus ou moins cassés ou fusionnés. Au final, 64 images sont ainsi générées pour valider leur méthode.

Ensuite, pour chaque classe le système choisit l'OCR le plus performant parmi deux. Le choix de cet OCR repose sur deux analyses réalisées par les auteurs :

– La première analyse les résultats de l'OCR en fonction des différentes classes de documents. En effet, il est possible d'associer à chaque classe (cassé, fusionné ou sans dégradations) l'OCR qui fournit les meilleurs résultats.
– Deuxièmement, les auteurs remarquent que certains documents sont mal classés. En effet, la mise en classe de l'étape précédente réalise un certain pourcentage d'erreurs. En fonction du pourcentage d'erreur de la classification, le système va choisir (par pondération) un OCR ou l'autre. Par conséquent, même si certaines images sont mal classées, elles peuvent probablement être en entrée de l'OCR qui leur correspond le mieux.

Ce système est comparé à une méthode de sélection aléatoire d'OCRs. En moyenne, le système de sélection aléatoire réalise un score de 0.95. Le système présenté réalise un score égal à 0.96. On constate donc une amélioration des résultats. La méthode semble être pénalisée par les mauvais taux de classification en classes (cassé, fusionné, ou sans dégradations) des images (23% d'erreur). De plus, cette méthode n'est pas comparée à un système optimal, il est donc impossible de le comparer aux résultats optimums. De plus, nous ne savons pas combien de fois le système a été testé et, comme il est basé sur une part d'aléatoire, il est possible que les résultats obtenus ne reflètent pas la réalité (ils peuvent être en moyennes meilleures ou pires).

3.1.2.3 Sélection automatique de méthodes de restauration en se basant sur des modèles prédictifs

D'autres articles de recherche ont pour objectif de sélectionner, pour une image de document donnée, la méthode (ou la combinaison de méthodes) de restauration proposant les meilleurs résultats. La méthodologie proposée dans ces articles est très proche de celle utilisée pour prédire le résultat d'un OCR. Tout d'abord, un ensemble de descripteurs est calculé sur un corpus d'apprentissage d'images de documents, puis des modèles de prédiction (classification) basés sur ces descripteurs sont générés.

Dans [SCNS03], les auteurs ont pour objectif d'améliorer les performances de l'OCR en sélectionnant pour chaque image de document un algorithme de restauration optimal qui traitera l'image avant l'exécution de l'OCR. Leur méthode est schématisée en figure 3.5.

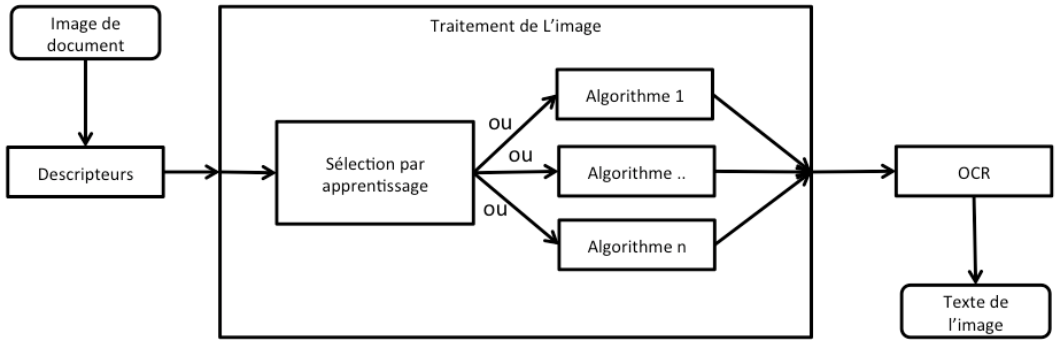


FIGURE 3.5 – *Système de sélection automatique de méthodes de restauration optimales proposées dans [SCNS03]*

Les descripteurs binaires présentés en section 2.1.3.1 sont utilisés pour la sélection automatique de l'algorithme optimal. Quatre méthodes de restauration sont proposées dans le système. Ces dernières sont toutes basées sur des opérateurs morphologiques d'ouverture ou de fermeture. Ces méthodes diffèrent de par la forme et la taille des éléments structurants. La sélection de l'algorithme est réalisée par un ensemble de règles de décisions définies par analyse de la distribution de chaque descripteur sur le corpus de document. L'article ne présente pas l'ensemble des règles choisies, mais seulement 22 exemples tels que :

Arbre de décisions 1 *Exemple de règle de décision utilisée dans l'article [SCNS03]*

si BCF (Broken Character Factor) > 0 **alors**
  Méthode par ouverture
fin si

Les auteurs évaluent leur méthode sur trois différents OCRs avec un corpus de document contenant 370 images. Ils comparent le taux de reconnaissance de l'OCR sans utilisation de méthode de restauration, puis avec une sélection manuelle et, pour finir, avec leur système de sélection automatique. Les résultats, présents en tableau 3.2, montrent une nette amélioration des performances de l'OCR, par rapport à la version sans restauration. Néanmoins l'écart constaté avec la sélection manuelle est important (environ 5%), d'autant plus que les auteurs ne comparent pas leur méthode avec la sélection optimale qui peut être obtenue à partir de la vérité-terrain de l'OCR en utilisant une méthode de type "brute-force" (test de toutes les méthodes possibles).

OCR	Tx de reco. sans filtre	Tx de reco. avec le meilleur filtre	Tx de reco. avec filtre automatiquement sélectionné
OCR 1	65.04	85.87	80.85
OCR 2	71.47	83.54	80.75
OCR 3	66.86	93.10	90.37

TABLE 3.2 – *Amélioration des résultats de trois OCRs constatée dans [SCNS03]*

Les auteurs de [CHK99] utilisent une approche similaire dans l'objectif de sélectionner une méthode de restauration optimale pour chaque image de document et par conséquent d'améliorer les performances de l'OCR. Les auteurs disposent d'un corpus (de 139 documents) et de leurs vérités-terrains OCR associées ainsi que de 14 méthodes de restauration. Pour assigner à une image donnée la meilleure méthode de restoration les auteurs ont appliqué tous les algorithmes de restauration sur cette image et ont sélectionné la méthode qui maximisait le taux de reconnaissance de l'OCR. Deux types de vérités-terrains sont constitués : la première est basée sur le taux de reconnaissance de caractères et la deuxième sur le taux de reconnaissance de mots. Le modèle de prédiction est obtenu en calculant les descripteurs présentés en section 2.1.3.1 à l'aide d'un classifieur linéaire (basé sur des réseaux de neurones). Une étape de cross-validation est utilisée pour juger les qualités prédictives du modèle : pour chacun des 139 documents, le modèle est entrainé sur les 138 documents restants et validé sur le document courant.

Sur les 139 documents, 66% ont été bien classifiés en se basant sur le taux d'erreur OCR niveau caractère. Sur la vérité-terrain basée mot, les résultats baissent (64% de bonne classification). Malgré ces erreurs de classification, l'utilisation de cette méthode améliore les résultats de l'OCR. Pour la classification basée "caractères", la méthode baisse le taux d'erreur de 20,27% à 12,60% ce qui est proche de la sélection optimale dont le taux de reconnaissance OCR égale à 11,34%. Pour la méthode de classification basée "mots", le taux d'erreur passe de 32,17% à 24,42% (la sélection optimale montre un taux d'erreur égale à 22,61%).

Les auteurs de [KAM12] proposent d'utiliser les descripteurs SIFT et LBP (Local Binary Pattern) afin de prédire les OCRs. Le modèle de prédiction se base sur un classifieur SVM. La capacité de ces descripteurs à pouvoir être utilisés sur des images en niveaux de gris permet aux auteurs de s'affranchir de l'étape de binarisation de l'image de document qui est souvent propre à chaque OCR. Bien que cette approche semble intéressante, elle a besoin d'être affinée et complétée par d'autres descripteurs, car, à l'heure actuelle, les résultats prédits sont relativement imprécis. Cependant, elle permet de distinguer efficacement deux classes de documents : ceux dont les performances OCR sont inférieures à 70% et ceux pour lesquels le résultat de l'OCR est supérieur à 90%. Ces résultats ont été obtenus sur une base de 2270 images sélectionnées par les auteurs et en utilisant deux OCRs différents : Abbyy FineReader et Omnipage.

3.1.3 Conclusion sur les modèles de prédiction existants

Il est impossible de comparer entre eux, les résultats des modèles de prédiction présentés. En effet, ces derniers ne se basent pas sur les mêmes corpus de documents ni sur les mêmes algorithmes (différents OCRs et différentes méthodes de restauration). Néanmoins, il est quand même possible de formuler plusieurs constats sur les différentes méthodes présentées :

– Comme nous l'avons dit, la plupart des méthodes présentées se basent sur des descripteurs qui s'appliquent uniquement à des images binaires. L'évolution des moyens techniques de numérisation nous amène à penser que l'utilisation de descripteurs utilisant les images en niveaux de gris voire même en couleurs sera plus pertinente.

– Les OCRs sont devenus des systèmes complexes composés de plusieurs maillons tels que la binarisation ou l'extraction de structures physiques. Comme nous l'avons mentionné dans la section 1.3 il est intéressant de prédire le comportement global de ces systèmes. Néanmoins, il nous semble pertinent de créer des modèles prédictifs pour chaque maillon d'une chaine d'analyse et de traitements d'images de documents.
– Pour les propositions de classification des résultats d'OCRs, nous aurions aimé voir apparaitre un plus grand nombre de classes. Le choix d'utiliser seulement deux classes, qui peut être à l'origine une contrainte économique, peut rapidement présenter des limites. Par exemple, la Bibliothèque Nationale de France utilise plusieurs classes permettant de distribuer les documents aux bonnes ressources. Par exemple lorsque l'OCR réalise plus de 99% de bons résultats, le document est mis en ligne ; entre 90% et 99% le résultat de l'OCR est corrigé ; en dessous de 60% le document est retranscrit entièrement.
– Aucune des initiatives présentées ne s'intéresse à la pertinence des descripteurs utilisés, sur le modèle final. Certains modèles de prédiction utilisent un grand nombre de descripteurs et peu de documents pour apprendre et valider leurs modèles. Dans le cas d'un modèle où l'espace de descripteurs est de grande dimension, il est judicieux soit de réduire en dimension, soit de supprimer les descripteurs redondants.
– Pour finir, certaines méthodes ne semblent pas faire de validations statistiques. Or, un modèle de prédiction peut être sur-paramètré ou présenter la caractéristique d'avoir été créé avec "sur apprentissage". Dans les deux cas, le modèle sera beaucoup moins précis.

Ces problèmes d'analyse statistique sont complexes. Dans la section suivante, nous essayons de répondre à ces problèmes par la définition d'un protocole rigoureux et basé sur des régressions linéaires pour prédire le résultat des algorithmes.

3.2 Création de modèles de prédiction par régression linéaire multivariée

Dans l'objectif de prédire l'algorithme optimal pour une image donnée, mais aussi de disposer d'un ordre de grandeur entre deux prédictions, nous avons défini un protocole expérimental basé sur un régresseur linéaire.

Une prédiction précise des résultats d'un algorithme n'aurait pu être réalisée en utilisant un classifieur. En effet, les classifieurs sont basés sur une décomposition ou une séparation de l'espace des paramètres en classes. Ce qui revient à faire une quantification de la fonction de prédiction. Par exemple la performance d'une binarisation, mesurée par le f-score ([0...1]) peut-être classée en *mauvais* ([0...0.3]), *moyen* [0.3...0.7] et *bon* [0.7...1]. Cette mise en classe fait perdre l'aspect continu des valeurs et il est impossible d'avoir une notion de distance fine entre deux classes : une image peut être prédite à 0.29 (*mauvais*) et une autre à 0.3 (*moyen*) alors que la différence numérique entre les deux prédictions n'est pas si importante.

Pour choisir un algorithme optimal, il est nécessaire d'avoir une fonction de prédiction qui permette d'obtenir un ordre total entre les algorithmes. Ce n'est pas le cas d'un classifieur, car si les résultats de prédiction des deux algorithmes appartiennent à la même classe il n'est pas possible de les comparer. Par exemple, supposons deux modèles de prédiction (tous aussi précis) pour deux algorithmes différents et une image. Si les deux modèles se basent sur un régresseur, le modèle 1 prédit un score de 0.92 tandis que le modèle 2 prédit 0.93. Il est alors très simple de choisir l'algorithme 2 pour cette image. Dans le cas contraire, s'ils se basent sur un classifieur, les modèles auraient tous les deux prédit la classe *bon*. Le choix de l'algorithme se serait alors fait au hasard alors que souvent une amélioration de 0.01 peut s'avérée importante pour les algorithmes qui suivront dans la chaine de traitements.

Notre protocole se divise en plusieurs étapes schématisées sur la figure 3.6. Tout d'abord, nous définissons un modèle de prédiction à partir de la totalité du corpus. Cette étape détaillée en section 3.2.1

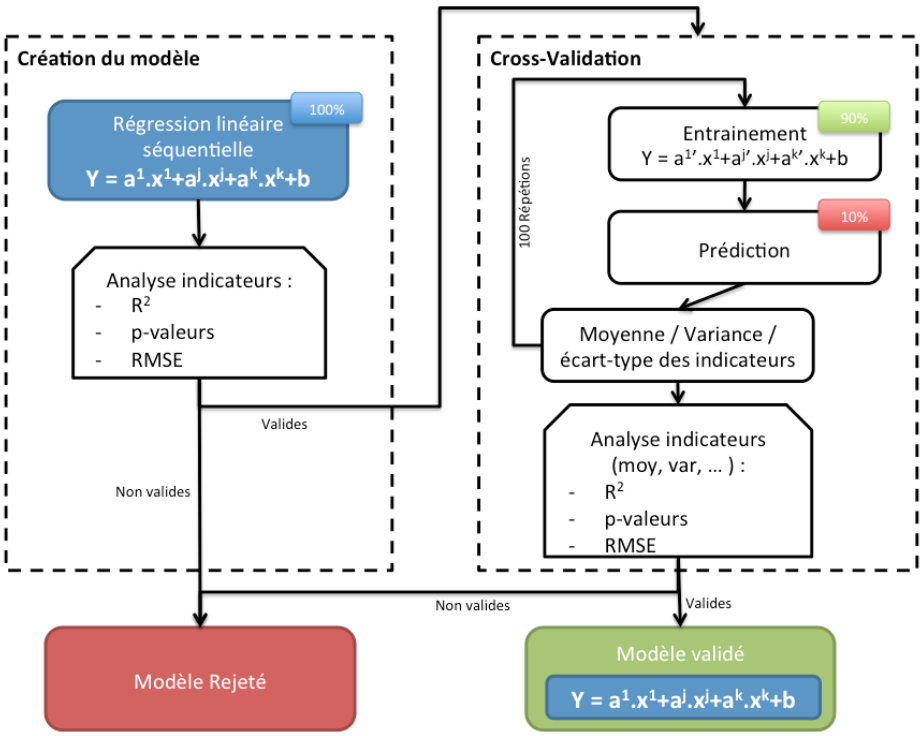


FIGURE 3.6 – *Protocole de création et de validation de modèles prédictifs basés sur des régresseurs statistiques. Le modèle, créé par une régression (linéaire) séquentielle en bleu sur le schéma, est statistiquement validé par une cross validation. Des indicateurs statistiques tels que le R^2 ou le RMSE sont utilisés pour juger la précision du modèle.*

permet de mesurer la pertinence du modèle, mais aussi de sélectionner les descripteurs les plus pertinents. Une phase de pondération (p-valeurs) permet de mesurer la pertinence d'un descripteur. Le modèle ainsi créé contient un ensemble de descripteurs et leurs coefficients. La deuxième étape est détaillée en section 3.2.2 et a pour objectif de valider statistiquement le modèle obtenu. Pour cela nous réalisons une cross-validation. Le principe de la cross-validation est de diviser aléatoirement et un certain nombre de fois, le corpus d'images de documents en deux ensembles : le corpus d'apprentissage (90% du corpus) et le corpus de validation (10% du corpus). Ce processus est itéré plusieurs fois. À chaque itération, nous entrainons le modèle sur le corpus d'entrainement (nous utilisons les descripteurs sélectionnés précédemment, mais leurs coefficients changent), puis prédisons sur le corpus de validation. Nous enregistrons les indicateurs statistiques afin de pouvoir les analyser une fois toutes les itérations terminées. Si les indicateurs calculés lors de la validation sont corrects, nous validons le modèle initial. Ce protocole est applicable à d'autres régresseurs statistiques tels que les SVMs.

3.2.1 Création d'un modèle de prédiction par régression linéaire multivariée stepwise

Les régressions linéaires multivariée

De manière formelle, une régression linéaire multivariée modélise la relation entre une variable y et un vecteur de variables x :

$$y = u + \beta_0 * x_0 + \beta_1 * x_1 + ... + \beta_n * x_n$$

Avec :

– y, la variable à expliquer.
– $(x_0, x_1, ..., x_n)$, les variables explicatives précédemment calculées ou renseignées.
– $(\beta_0, \beta_1, ..., \beta_n)$, les coefficients estimés par la régression pour chaque variable explicative.
– u, est la constante de régression, qui peut être considérée comme l'erreur résiduelle non modélisée par le modèle, on l'appel l'*intercepte*.

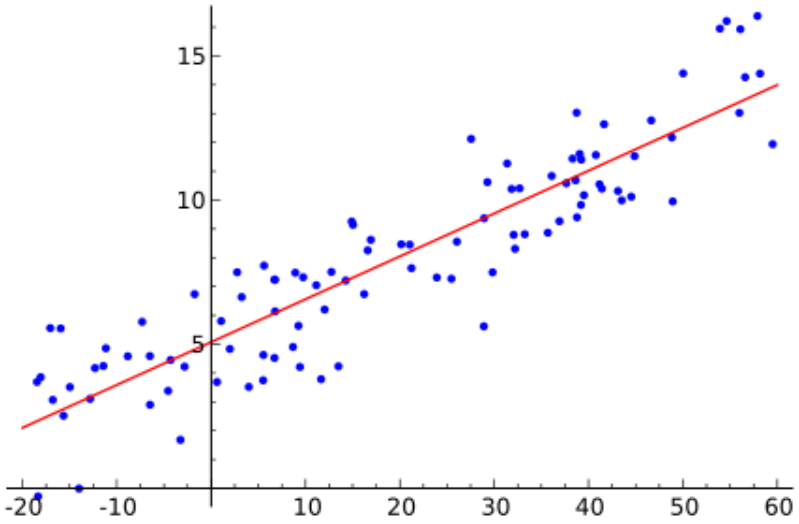


FIGURE 3.7 – *Régression linéaire simple (avec 2 dimensions) : en ordonnée, la variable à expliquer, en abscisse, la variable explicative. La fonction de régression est* $y = 0,14 * x + 5$.

La régression linéaire simple comme celle présentée en figure figure 3.7, est un cas particulier de la régression linéaire multivariée où la variable à expliquer doit être modélisée par une seule variable explicative ($y = u + \beta * x$). Sur la figure 3.7 nous avons en ordonnée la variable à expliquer (y) et en abscisse la variable explicative (x). Une régression linéaire simple nous conduit à la fonction suivante : $y = 0,14 * x + 5$ ($\beta = 0,14$ et $u = 5$).

Le modèle de régression linéaire est souvent estimé par la méthode des moindres carrés, mais il existe de nombreuses autres méthodes pour estimer ce modèle. On peut par exemple l'estimer par le maximum de la vraisemblance ou encore par inférence bayésienne.

Une fois le modèle créer, un ensemble d'indicateurs permet d'évaluer sa précision :
– (**Root**) **M**ean **S**quare **E**rror est une mesure quantifiant la différence entre les valeurs estimées et les vraies valeurs à estimer. Cet estimateur est calculé en fonction de la variance et du biais. La formule de cette mesure est :

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(x_{1,i} - x_{2,i})^2}{n}}$$

avec n le nombre d'observations, $x_{1,i}$ les observations et $x_{2,i}$ les valeurs prédites.
– R^2 [DSP66] est un coefficient de détermination, cette mesure est normée. Son interprétation est la suivante : plus la valeur du R^2 est proche de 1, plus le modèle est susceptible de bien prédire de futurs résultats. Une valeur égale à 1 signifie que le modèle prédit parfaitement les données. En général il est admis qu'une valeur $R^2 \geq 0.7$ est suffisante pour avoir une *bonne* précision.
– **p-valeur** [Moy06] : pour un descripteur donné, la valeur p est la probabilité de commettre une erreur si ce descripteur est présent dans le modèle de prédiction. De manière formelle, la p-valeur est la probabilité d'obtenir la même valeur (coefficient du descripteur) si l'hypothèse nulle (descripteur non

significatif) est vraie. Cette dernière doit être la plus petite possible. En général, on considère les seuils suivants :
  - < 0.01 : très bonne pertinence. Très faible probabilité d'obtenir le même coefficient si l'hypothèse nulle est vraie. Le descripteur est très significatif.
  - $0.01 - 0.05$: bonne pertinence.
  - $0.05 - 0.1$: faible pertinence.
  - $0.1 >$: pas de pertinence. Il faut rejeter le descripteur.
- **écart type** : l'écart type est la racine carrée de la variance. Il est donc similaire au RMSE, mais ne prend pas en compte le biais.

Sélection des descripteurs les plus significatifs

Notre objectif est d'estimer au mieux la relation entre le résultat d'un algorithme et la qualité d'un document. Le problème essentiel est de prendre en compte seulement les descripteurs les plus pertinents afin d'éviter tout phénomène de confusion. En effet, il est possible que la régression utilise tellement de descripteurs que le modèle se trouve sur paramétré, permettant alors de représenter parfaitement les données d'entrainement (over-fit). Lorsque le modèle est sur paramétré, les coefficients sont estimés avec une très mauvaise précision (grande variance) compte tenu du nombre élevé de variables. Autrement dit, les p-valeurs correspondant aux coefficients des descripteurs sont très élevés (> 0.1). Il est donc nécessaire d'adopter une stratégie de sélection de descripteurs en fonction de l'algorithme à prédire. Si l'on restreint le modèle aux descripteurs les plus pertinents, on peut espérer qu'il sera plus performant lors de la phase de prédiction (il sera moins sensible aux valeurs aberrantes). De plus, la précision du modèle sera plus facile à interpréter.

Pour restreindre le modèle aux descripteurs les plus pertinents, nous utilisons une régression séquentielle. Une régression séquentielle est une régression linéaire où le choix des descripteurs est réalisé de manière automatique en introduisant progressivement les descripteurs au modèle et en enlevant à chaque étape les descripteurs qui ne sont plus pertinents. Ce sont les descripteurs qui contribuent le plus à la précision du modèle qui seront sélectionnés. Le processus de sélection peut être résumé par les étapes suivantes :

1. ajouter un nouveau descripteur.
2. recalculer la régression avec le nouveau descripteur
3. si le R^2 est plus proche de 1 et que la p-valeur du descripteur est < 0.1 on le garde.
   (a) on enlève tous les descripteurs qui ne sont plus pertinents (p-valeur > 0.1)
4. on passe au descripteur suivant.

Le modèle de prédiction ainsi obtenu (en bleu sur la figure 3.6) doit ensuite être validé statistiquement par cross-validation.

3.2.2 Validation statistique de modèle par Cross-Validation

Comme nous venons de le voir, un ensemble d'indicateurs permet de mesurer la précision du modèle de prédiction. Cependant, cette précision est calculée sur les données d'apprentissage et ne permet pas de valider définitivement le modèle. Il est nécessaire de réaliser une cross-validation pour valider le modèle de prédiction en mesurant ses qualités prédictives réelles.

La cross-validation est une technique utilisée pour vérifier les capacités prédictives d'un modèle lorsque le corpus contient très peu de données. En effet, il est nécessaire, pour la création d'un modèle de prédiction, de disposer, certes d'un nombre assez important d'échantillons pour pouvoir entrainer le modèle, mais aussi, d'assez d'échantillons pour évaluer la qualité (ou précision) de la prédiction sur des données qui n'ont pas servies à l'apprentissage. Il se pose alors naturellement le problème de la

partition (même aléatoire) du corpus : il est possible que les deux partitions soient les seules (ou une des seules) qui permettent de valider le modèle. Le modèle de prédiction est alors dépendant des données d'apprentissage et ne sera pas assez précis sur d'autres données.

Le problème des corpus contenant peu d'échantillons est particulièrement présent en analyse et traitement d'images de documents où la vérité-terrain des corpus publics libres de droits est souvent complexe et longue à constituer. Par exemple, le corpus DIBCO [GNP09] servant aux compétitions de binarisation ne contient qu'une trentaine d'images.

La cross-validation propose de réaliser un grand nombre d'échantillonnages aléatoires du corpus. On divise alors, aléatoirement, k fois le corpus de document : 90% des échantillons sont utilisés pour l'entrainement et 10% sont utilisés pour la validation. Les indicateurs statistiques de précision du modèle (RMSE, R^2, etc.) sont moyennés sur l'ensemble des k réalisations. C'est à la fin des k réalisations que nous pourrons valider le modèle. De plus, nous analysons la variance des indicateurs et des coefficients des descripteurs afin de savoir si le modèle est robuste ou non aux données d'apprentissage.

3.3 Application à la binarization

Nous voulons dans cette section évaluer la pertinence de notre méthodologie à savoir l'utilisation de descripteurs pour prédire les performances d'algorithmes. Nous nous concentrons, dans cette section, sur la prédiction de méthodes de binarisation. Nous voulons être capables de créer un modèle prédictif précis pour chaque algorithme de binarisation, et ce indépendamment de leurs concepts théoriques.

3.3.1 Les méthodes à prédire

Étant donné qu'il existe un très grand nombre d'algorithmes de binarisation et qu'il est difficile, voir impossible ,de tester ces descripteurs sur la totalité des méthodes, nous avons donc identifié quatre familles de méthodes qui reposent sur des concepts scientifiques différents. Pour chacune de ces familles, nous avons sélectionné au moins un algorithme en favorisant ceux qui sont utilisés dans le cadre de l'analyse et du traitement d'images de documents.

1. les méthodes globales basées sur les valeurs de l'histogramme des niveaux de gris :
   - Kittler [KI85],
   - Otsu [Ots75].
2. les méthodes globales basées sur la forme de l'histogramme des niveaux de gris :
   - Ridler [C$^+$78],
3. les méthodes basées sur l'entropie :
   - Kapur [KSW85],
   - Li [LT98],
   - Sahoo [SWY97],
   - Shanbag [Sha94]
4. les méthodes de seuillage local :
   - Bernsen [Ber86],
   - White [WR83],
   - Sauvola [SP00]
5. et pour finir, les méthodes combinant plusieurs de ces techniques :
   - Shijian [SLT11] (gagnante du concours de binarisation d'ICDAR 2011).

Nous allons donc dans les sections suivantes créer un modèle de prédiction pour chacun des algorithmes cités. Certains de ces algorithmes doivent être paramétrés afin de proposer de bons résultats. La paramétrisation optimale d'un algorithme pour un corpus d'image donné est un problème complexe qui

Méthode	Paramètres	
Bernsen	Taille Fenêtre	31
White	Taille Fenêtre	15
	bias	2
Sauvola	Taille Fenêtre	15
	R	128
	K	0.5

TABLE 3.3 – *Paramètres choisis des méthodes de binarisation.*

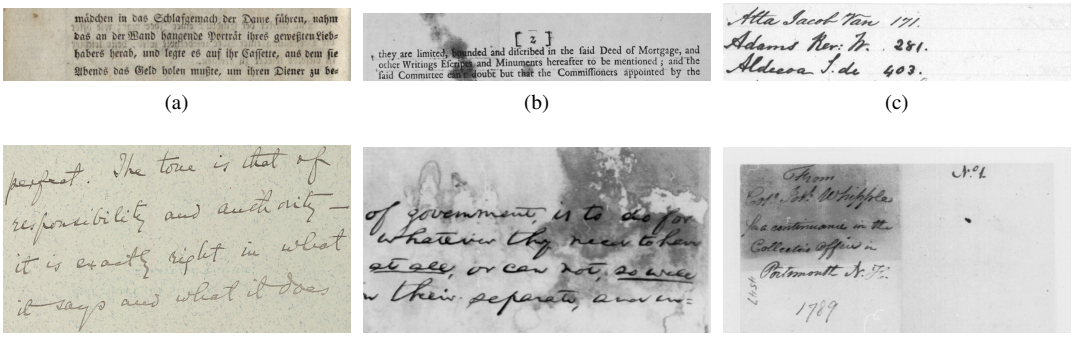
sort du contexte de cette thèse. Ici, nous considérons que pour un algorithme donné, il est nécessaire de créer autant de modèles prédictifs que de jeux de paramètres associés. Ainsi, chaque modèle de prédiction est associé à un ensemble composé de l'algorithme et de son jeu de paramètres. Ces paramètres sont présentés sur le tableau 3.3.

Afin de comparer le résultat d'une méthode de binarisation avec sa vérité-terrain, nous utilisons la mesure F-Score, qui peut être considérée comme la moyenne entre la précision et le rappel.

$$\text{F-Score} = 2 * \frac{precision * rappel}{precision + rappel}$$

3.3.2 Le corpus de document

Dans l'objectif de créer des modèles de prédiction, nous avons besoin d'un corpus représentatif des images de documents que nous serons amenés à traiter. De plus, un grand nombre de dégradations influençant directement la qualité des résultats des algorithmes de binarisation doit être présent sur les images de ce corpus.



(a) (b) (c)

(d) (e) (f)

FIGURE 3.8 – *Extraits du corpus de documents DIBCO et H-DIBCO [GNP09] utilisés pour les compétitions de binarisation.*

La création de vérités terrains pour les méthodes binarisation est une tâche complexe et fastidieuse. De ce fait, les bases existantes contiennent peu de documents. Il existe plusieurs corpus (DIBCO [4] et H-DIBCO [5] [GNP09]) utilisés pour les concours de méthodes de binarisations. Chaque année depuis 2009, la vérité-terrain d'une dizaine d'images est réalisée. Afin d'avoir le plus d'images possible, nous avons

4. http ://users.iit.demokritos.gr/ bgat/DIBCO2009/
5. http ://users.iit.demokritos.gr/ bgat/H-DIBCO2010/

fusionné l'ensemble des images proposées sur chaque année au sein d'un même corpus contenant au total 36 images de documents. Ce corpus semble pertinent, car l'analyse des performances des méthodes (tableau 3.4) montre que chaque algorithme est évalué sur des cas difficiles (f-score < 0.6) et faciles (f-score > 0.8). Cela correspond bien à notre besoin de pouvoir prédire le résultat d'une méthode tant dans les cas faciles que difficiles. Un extrait du corpus de documents utilisé est présent en figure 3.8.

F-Score	Mean	Std. Dev.	Min	Max
Sauvola	0.55	0.28	0.1	0.98
Otsu	0.81	0.14	0.28	0.96
Shijian	0.89	0.12	0.21	0.95
Bernsen	0.48	0.47	0.1	0.85
Kapur	0.84	0.07	0.63	0.94
Kittler	0.77	0.16	0.24	0.95
Li	0.69	0.2	0.1	0.96
Ridler	0.82	0.14	0.28	0.96
Sahoo	0.82	0.009	0.50	0.96
White	0.4	0.22	0	0.83
Shanbag	0.81	0.11	0.49	0.93

TABLE 3.4 – *Indicateurs statistiques de 11 méthodes de binarisation sur le corpus d'images de documents DIBCO [GNP09]. Selon les images, chaque méthode de binarisation peut aboutir à de bons et de mauvais résultats.*

3.3.3 Prédiction à l'aide d'une régression linaire multivariée

Pour l'ensemble des méthodes présentées en sous-section 3.3.1, nous avons créé un modèle de prédiction basé sur une régression linéaire multivariée en suivant la méthodologie proposée en section 3.2. Pour rappel voici les principales étapes :

1. Sélection des descripteurs les plus significatifs.
2. Entrainement d'un modèle sur 90% du corpus (images choisies aléatoirement).
3. Validation sur les 10% restants.
4. Cross-validation : partitionnement, apprentissage et validation répétés 100 fois.

Nous proposons de détailler précisément les résultats obtenus pour les méthodes de binarisation les plus couramment utilisées dans l'analyse et le traitement d'images de documents à savoir : Otsu, Sauvola et Shijian. Pour ces trois méthodes, nous présentons les coefficients associés aux descripteurs les plus significatifs, leurs p-valeurs, et l'intercepte de la fonction prédictive. Par suite, si un descripteur n'est pas mentionné c'est qu'il n'a pas été retenu dans le modèle final (descripteur pas assez significatif). Pour finir, nous analysons le R^2, l'erreur moyenne et le RMSE moyennés sur les 100 prédictions effectuées lors de la cross-validation.

Modèle de prédiction pour la méthode d'Otsu

Les descripteurs sélectionnés pour la méthode Otsu sont $\mathcal{MI}_I$, v_I , v_B, μ_B, μ and v (voir tableau 3.5 pour les valeurs des coefficients). La sélection de ces descripteurs pour cette méthode de binarisation peut être expliquée par le fait que Otsu est une méthode de seuillage globale. Il est donc normal de voir apparaitre les descripteurs tels que $\mathcal{MI}_I$, μ et v avec de si faible de p-valeurs (pour rappel, plus une p-valeurs est faible plus le descripteur associé est significatif).

	Feature coef.	p-value
Intercept	$1.187e+00$	< 0.0001
$\mathcal{MI}_I$	$1.244e+00$	< 0.0001
v_I	$2.422e-02$	< 0.1
v_B	$-4.336e-02$	< 0.01
μ_B	$-2.662e-02$	< 0.0001
μ	$2.445e-02$	< 0.0001
v	$3.262e-04$	< 0.0001
$R^2 = 0.93$		

TABLE 3.5 – *Modèle de prédiction proposé pour la méthode de binarisation d'Otsu.*

L'étape de cross-validation donne un R^2 moyen égal à 0.987 (le R^2 optimal est égal à 1) ce qui est considéré comme excellent. Nous avons aussi réalisé une régression linéaire (univariée cette fois-ci) entre les résultats prédits et la vérité-terrain. Le coefficient de cette régression est égal à 0.989, soit très proche de 1 (une prédiction parfaite aurait une droite de régression égale à "vérité = 1 * prédiction").

Modèle de prédiction pour la méthode de Sauvola

En ce qui concerne Sauvola, les descripteurs sélectionnés sont $\mathcal{MI}_B$, $\mathcal{MQ}$, $\mathcal{MA}$, μ, s, s_I, v_I. Comme la méthode de Sauvola est basée sur une approche locale et adaptative, les descripteurs locaux tels que $\mathcal{MA}$ sont sélectionnés. Cependant, les descripteurs locaux $\mathcal{MS}$ et $\mathcal{MSG}$ ne sont pas assez significatifs pour être sélectionnés. Cela peut venir soit des paramètres choisis pour cet algorithme soit par le simple fait que la méthode de Sauvola n'est pas assez perturbée par les dégradations grises connectées aux composantes d'encre, ou une combinaison de ces deux hypothèses. Une analyse visuelle des résultats proposés par Sauvola montre en tout cas que cet algorithme est très robuste à ce type de dégradation et très sensible aux dégradations mesurées par $\mathcal{MA}$ (composantes grises non connectées à l'encre).

	Features coef.	p-value
Intercept	1.61+00	< 0.0001
$\mathcal{MI}_B$	1.19	< 0.01
$\mathcal{MQ}$	-1.1	< 0.0005
$\mathcal{MA}$	2.3e-01	< 0.05
μ	-4.56e-03	< 0.0001
s	7.709e-02	< 0.0001
s_I	1.431e-01	< 0.0001
v_I	4.264e-04	< 0.0001
$R^2 = 0.9077$		

TABLE 3.6 – *Modèle de prédiction proposé pour la méthode de binarisation de Sauvola.*

La cross-validation montre là encore un R^2 moyen très bon (0.99). La régression linéaire entre les valeurs prédites et les vrais résultats montrent un coefficient égal à 1.0007. Ces résultats sont très encourageants quant à la problématique de l'utilisation de ce modèle dans un contexte industriel.

Modèle de prédiction pour la méthode de Shijian

La méthode de binarisation de Shijian est un peu particulière puisqu'elle repose sur la combinaison de plusieurs techniques pour proposer son résultat. Les descripteurs sélectionnés sont : $\mathcal{MI}_B$, $\mathcal{MA}$, $\mathcal{MSG}$, var, s_I, s_D, et μ_I. Les coefficients associés aux différents descripteurs sont présentés dans le tableau 3.7.

	Features coef.	p-value
Intercept	1.068e+00	< 0.0001
$\mathcal{MI}_B$	-7.971e-01	< 0.05
$\mathcal{MA}$	3.162e-02	< 0.0001
$\mathcal{MSG}$	-3.276e-02	< 0.0001
var	-1.389e-04	< 0.0001
s_I	3.882e-02	< 0.0001
s_D	1.328e-01	< 0.001
μ_I	-4.004e-04	< 0.5
$R^2 = 0.86$		

TABLE 3.7 – *Modèle de prédiction proposé pour la méthode de binarisation de Shijian.*

En ce qui concerne les résultats de la cross-validation, les résultats sont aussi très bons. Le R^2 moyen est égal à 0.99 et le coefficient moyen entre les valeurs prédites et les vraies valeurs est égal à 1.06.

Analyse des modèles de prédiction sur les autres méthodes de binarisation

De façon similaire, nous avons réalisé des modèles de prédiction pour l'ensemble des autres méthodes de binarisation sélectionnées (en section 3.3.1). Tous les modèles de prédiction obtenus montrent un R^2 supérieur à 0.7 (tableau 3.8) sauf celui pour Sahoo mais qui n'est pas non plus très éloigné (0.67). On remarque que le descripteur $\mathcal{MS}$ n'est jamais sélectionné lors de l'étape de régression séquentielle. Nous pensons que cela vient de la nature de $\mathcal{MS}$ qui se base sur une analyse des composantes connexes. En effet, la méthode d'évaluation des méthodes de binarisation est, quant à elle, basée pixel (f-score). Avec cette méthode d'évaluation, $\mathcal{MSG}$ devient plus significatif que $\mathcal{MS}$. Cela n'aurait peut-être pas été le cas avec une autre méthode d'évaluation de performance basée composantes.

Method	Selected Features	R^2	Erreur Moyenne	RMSE
Bernsen	$\mathcal{MI}_I; \mathcal{MA}; \mathcal{MSG}; v; v_D; v_I$	0.8281	6%	0.08
Kapur	$\mathcal{MI}_I; \mathcal{MA}; \mu; v; s_D; v_I; \mu_D; \mu_I$	0.782	2%	0.03
Kittler	$\mathcal{MI}_I; \mathcal{MQ}; s; v_I; \mu_B; v_B$	0.8388	5%	0.07
Li	$\mathcal{MI}_I; \mathcal{MA}; \mathcal{MSG}; \mu; v; v_I; \mu_D; \mu_I$	0.8095	11%	0.13
Riddler	$\mathcal{MI}_I; v; v_D; v_I$	0.7507	5%	0.08
Sahoo	$\mathcal{MI}_I; \mu; s_B; v_I; \mu_D; \mu_I$	0.6781	5%	0.06
Shanbag	$\mathcal{MI}_I; s; v; s_D; s_I; v_D; v_I$	0.7289	6%	0.07
White	$\mathcal{MI}_I; \mathcal{MSG}; s; v; \mu_D; \mu_I; v_D$	0.9244	7%	0.08

TABLE 3.8 – *Analyse de la précision des modèles des 8 autres méthodes de binarisation. Les descripteurs sélectionnés sont différents d'une méthode à l'autre. La précision et la robustesse des modèles de prédiction est bonne ($R^2 \geq 0.7$, $RMSE < 0.13$). Dans le pire des cas, l'erreur moyenne est de 11% soit plus ou moins 0.11 de f-score et nous constatons au meilleur des cas de seulement 2% d'erreur en moyenne.*

Utilisation d'autres regresseurs

Les modèles de prédictions créés sont précis et les résultats présentés sont encourageants. Nous pensons que ces derniers peuvent être utilisés en production avec une phase d'apprentissage réalisée sur un plus grand nombre d'images. Cependant, nous avons fait une hypothèse forte qui nécessite d'être vérifiée : nous supposons en effet que la relation entre le résultat des méthodes de binarisation et nos descripteurs est linéaire. Pour vérifier cette hypothèse, il est possible de créer des modèles prédictifs non linéaires et de comparer les qualités prédictives de ces derniers.

Nous utilisons ici les SVMs. En effet, nous avons vu que ces derniers peuvent être utilisés pour des tâches de classification, mais aussi pour des tâches de régression. Les SVMs permettent de choisir un noyau permettant de représenter au mieux la relation entre les variables prédictives et la variable à prédire. Les noyaux les plus usuels sont de forme linéaire, de forme polynomiale, ou de forme radiale. Le choix d'un noyau pour SVM n'est pas évident. Comme nous voulons voir la performance de ces derniers sur des noyaux non linéaire, il nous reste le choix entre les noyaux polynomial et radial. Nous avons finalement choisi un noyau radial de façon générale, car ce dernier permet d'approximer également des polynômes [Sch94]. Nous avons étudié la précision de ces modèles à l'aide de deux mesures : le RMSE et l'erreur moyenne. Ces résultats sont présentés sur le tableau 3.9.

Méthode de binarisation	Erreur moyenne	RMSE
Sauvola	7%	0.10
Otsu	6%	0.11
Shijian	3%	0.07
Bernsen	9% (+3)	0.12 (+0.04)
Kapur	3% (+1)	0.05 (+0.02)
Kittler	6% (-1)	0.09 (+0.02)
Li	12% (+1)	0.17 (+0.04)
Ridler	5% (=)	0.08 (=)
Sahoo	4% (-1)	0.07 (-0.01)
Shanbag	6% (=)	0.09 (+0.02)
White	10% (+3)	0.14 (+0.08)

TABLE 3.9 – *Modèles de prédiction réalisés avec des régresseurs SVM. La précision des modèles obtenus est très semblable à ceux obtenus avec des régresseurs linéaires (tableau 3.8).*

Les résultats sont très semblables à ceux obtenus avec une régression linéaire. On constate néanmoins une légère baisse globale sauf pour les méthodes de binarisation de Kapur et Sahoo. Il est à noter que les résultats présentés pour les régressions linéaires et les SVMs, sont obtenus à partir des mêmes ensembles de documents (100 partitions aléatoires).

3.3.4 Vers une méthode de binarisation optimale

Dans la section précédente, nous avons présenté des modèles de prédiction, qui à partir d'une image de document et de l'ensemble de ses descripteurs, sont capables d'estimer les performances de plusieurs méthodes de binarisation. Ainsi, pour toute image de document, il est possible de prédire les taux d'erreurs de plusieurs méthodes de binarisation.

Les applications sont nombreuses, mais nous nous concentrons dans cette section sur la sélection automatique de méthode de binarisation et ce au cas par cas, image par image. En effet, il n'existe pas, à l'heure actuelle, de méthodes de binarisation qui serait optimale pour un ensemble de documents. Ces méthodes ont leurs avantages, inconvénients et faiblesses. Les performances d'une méthode de binarisation dépendent réellement du type de bruit auquel elle est sensible. Par exemple, nous avons vu que Sauvola était moins sensible aux variations de lumière que Otsu, mais plus sensible aux petites taches sombres. Il arrive que des images soient mieux binarisées avec des méthodes classiques qu'avec des méthodes plus récentes comme celle proposée par Shijian.

Disposant des modèles de prédiction précédents, il est possible de créer un processus de binarisation qui utiliserait les différents modèles pour choisir la méthode présentant les meilleures performances sur une image donnée.

Le tableau 3.10 présente des statistiques de performance (f-score) sur les résultats obtenus en binarisant le corpus d'images de documents DIBCO [GNP09]. La première ligne correspond aux f-scores théoriques qui peuvent être obtenus en choisissant la meilleure méthode de binarisation pour chaque image (cela est uniquement possible en disposant de la vérité-terrain). La seconde ligne correspond aux résultats obtenus par la méthode de binarisation qui présente la meilleure moyenne sur notre corpus de document à savoir Shijian. Les troisième et quatrième lignes correspondent à la sélection automatique de méthodes de binarisation en se basant respectivement sur les modèles de prédiction basés sur des régresseurs linéaires et sur des régresseurs SVM. La dernière ligne quant à elle, prédit la méthode de binarisation optimale en utilisant un classifieur SVM. Les classes à prédire correspondent au nom de la méthode de binarisation.

F-Score	Mean	Std. Dev.	Min	Max
Sélection optimale	0.917	0.03	0.77	0.96
Shijian	0.891	0.06	0.21	0.96
Sélection automatique (regression-lineaire)	0.906	0.04	0.61	0.96
Sélection automatique (regression-SVM)	0.89	0.06	0.21	0.96
Sélection automatique (classifieur-SVM)	0.90	0.1	0.21	0.96

TABLE 3.10 – *Résultats des méthodes de sélection de méthodes de binarisation optimales sur le corpus de document DIBCO [GNP09].*

Les méthodes permettant de sélectionner la méthode de binarisation optimale présentent des résultats similaires. En effet, les méthodes se basant sur des SVM présentent un score légèrement inférieur à celle utilisant des régresseurs linéaires, mais cette faible différence peut s'expliquer par le fait qu'elles ne sont pas en mesure de sélectionner une meilleure méthode de binarisation pour l'image la plus difficile (0.21 de f-score). Par conséquent la moyenne globale des méthodes de sélection basées sur des SVMs baisse.

Nous pouvons analyser les résultats de la méthode basée sur des régresseurs linéaires (celle présentant les meilleurs résultats) de plusieurs façons :
– Comparée à la méthode utilisant toujours Shijian, notre méthode augmente les résultats de 1,5 % et nous ne sommes plus qu'à 1,1% de la sélection optimale.
– Si l'on isole les images pour lesquelles la méthode de binarisation sélectionnée est Shijian, nous constatons une amélioration moyenne de 5%. Cela concerne environ 22% des images du corpus.
– Si l'on regarde les résultats de plus près, notre méthode augmente les performances dans le cas critique (l'image la plus complexe à binariser) de 56%. En effet, le f-score minimal est égal à 0.77 (soit seulement 0.12 de moins que le f-score moyen de Shijian). De plus, nous améliorons le f-score maximal de Shijian de 1% passant donc à 0.96. Les résultats, dans le pire ou le meilleur des cas sont exactement les mêmes que pour la méthode utilisant la vérité-terrain.
– Pour finir, nous avons étudié plus finement la classification vis-à-vis de la vérité-terrain : 70% des images se sont vu attribuer la méthode de binarisation optimale.

L'utilisation de notre méthode de sélection automatique de méthode peut améliorer de façon significative la binarisation d'un ensemble de documents et, par suite, les performances d'algorithmes reposant sur une étape de binarisation.

3.4 Perspectives, vers la prédiction de l'OCR en fonction de la transparence

Dans cette section notre objectif est d'évaluer notre méthodologie sur des systèmes complexes composés de plusieurs maillons algorithmiques (comme le sont les OCRs) et aussi sur d'autres mesures

d'évaluation de performances (par exemple orientées composantes à la différence du f-score de la binarisation qui est orientée pixel). Pour cela, nous allons créer des modèles de prédictions des performances de deux OCRs (Abbyy FineReader 9 [6] et OCROpus 0.4 [7]). Pour mesurer les performances d'un OCR, nous utiliserons la distance de Levenstein [SM01] qui est orientée composantes (caractères). Malheureusement les vérités-terrains disponibles pour l'évaluation des performances d'OCRs sont essentiellement constituées de documents modernes (sans dégradations) avec des mises en pages complexes. Cela ne correspond pas à nos besoins étant donné que nous n'évaluons pas la complexité d'analyse des images de documents, mais seulement leurs qualités. Pour contourner ce problème, nous aurons recours à des documents semi-synthétiques. Ces documents présenteront différents niveaux de transparence (dégradation qui sera générée par le modèle présenté dans [MC09a]).

Notre méthodologie est similaire à celle utilisée pour créer un modèle de prédiction pour une méthode de binarisation. La différence porte sur la création du corpus de documents que nous avons nous-mêmes généré :

– premièrement, nous créons un corpus d'images de documents contenant de la transparence et leurs vérités-terrains (texte de l'OCR). Ce corpus est divisé aléatoirement en deux sous-ensembles : le corpus d'apprentissage du modèle de prédiction et le corpus de validation.
– deuxièmement, nous utilisons une régression linéaire multivariée et séquentielle pour sélectionner les descripteurs les plus pertinents et ainsi construire le modèle prédictif.
– dernièrement, le modèle est validé statistiquement par cross-validation.

3.4.1 Le corpus de document

Les corpus de documents que nous connaissons et contenant la vérité-terrain OCR ne correspondent pas à nos besoins pour plusieurs raisons. Premièrement, ils sont constitués de documents très peu dégradés (ces derniers sont souvent renumérisés ou restaurés). Deuxièmement, ils sont pour la plupart destinés aux compétitions d'OCRs et contiennent généralement des images complexes avec des tableaux, figures, différentes fontes, des structures physiques modernes (comme des magazines par exemple). Troisièmement, les documents utilisés pour créer les modèles de prédiction d'OCRs présentés en section 3.1.2 ne sont pas pertinents, car les images sont déjà binarisées et la transparence est un défaut caractérisé par des composantes en niveaux de gris.

Descripteurs	Moyenne	Écart-type	Minimum	Maximum
$\mathcal{MI}_I$	0.57	0.089392	0.373813	0.822115
$\mathcal{MI}_B$	0.22	0.08041874	0.0536898	0.377166
$\mathcal{MQ}$	16,47	71.97754	0.698371	442.018
$\mathcal{MA}$	0.49	0.1354490	0.290468	0.945854
$\mathcal{MS}$	0.37	0.1007081	0.0578035	0.587493
$\mathcal{MSG}$	2.23	0.5172265	0	2.60243
Taux d'erreur OCR [SM01]				
Abbyy Fine Reader	59,5	19,5	20,5	100
OCRopus	77,5	11.00037	48,5	100

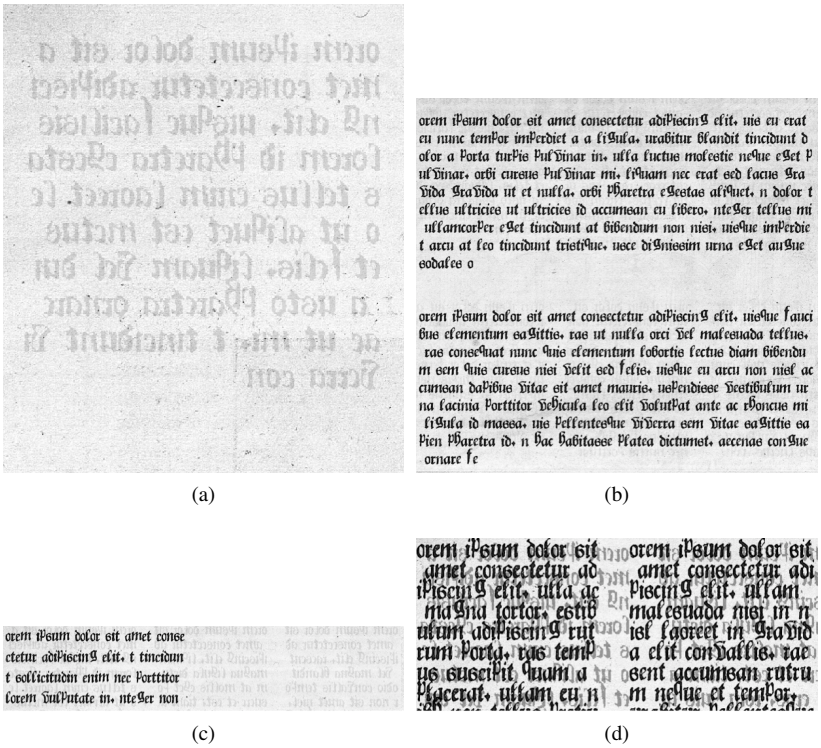
TABLE 3.11 – *Distribution des descripteurs sur le corpus de documents ainsi que leurs valeurs théoriques maximales et minimales. Au regard des descripteurs, le corpus est hétérogène. On remarque une nette différence entre les performances de Abbyy et OCRopus. Cette différence peut être expliquée par le fait que OCRopus n'a pas été entrainé lors de cette expérimentation.*

6. http ://france.abbyy.com/
7. http ://code.google.com/p/ocropus/

Pour ces raisons, nous avons décidé de construire nous même un corpus de documents contenant des images avec différents niveaux de transparence. Nous avons pour cela utilisé le logiciel présenté dans [JVD+10] et permettant de créer des documents anciens semi-synthétiques avec leurs vérités-terrains associées. Nous avons intégré au logiciel deux nouveaux modules : le premier pour générer du texte de type *Lorem Ipsum* [8] et le second pour ajouter de la transparence à une image de document via le modèle de dégradation présenté dans [MC09a]. Ce logiciel est détaillé en section 4.1.1.2.

Plusieurs paramètres et caractéristiques d'un document ont été variabilisés pour générer 190 images de documents. Ces paramètres, fixés aléatoirement sont :
- pour le texte nous générons du *Lorem Ipsum* [9],
- la taille des marges varie de 50 à 200 pixels,
- l'espace inter-ligne varie de 30 à 50 pixels,
- le nombre de colonnes et de blocs de lignes varie de 1 à 3, l'espace inter colonne et interligne varie aussi entre 30 et 100 pixels,
- le pourcentage de texte sur le document final varie de 0% (pour générer des pages blanches), à 100%,
- le fond est choisi aléatoirement parmi trois fonds extraits de documents anciens,
- une fonte (parmi trois) est aussi choisie aléatoirement, chacune des fontes est extraite manuellement depuis des documents anciens numérisés, un caractère est représenté par plusieurs images ce qui permet de varier les dégradations présentes sur un même caractère,
- à chaque image de document, nous ajoutons plus ou moins de transparence, certaines images ne contiennent pas de transparence.

L'ensemble de ces paramètres permet de générer un grand nombre d'images de documents présentant des mises en pages et des niveaux de transparence différents (pages blanches 3.9a, pages sans transparence 3.9, avec une colonne 3.9c ou deux colonnes 3.9d, etc.).



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

FIGURE 3.9 – *Extraits du corpus d'images de documents transparentes et générées avec le logiciel présenté dans [JVD+10].*

8. http ://www.lipsum.com/
9. www.loremipsum.fr

La distribution des descripteurs ainsi que les taux d'erreurs des deux OCRs sont présentés sur le tableau 3.11. Les valeurs observées montrent que le corpus de document est hétérogène si l'on considère les caractéristiques de la transparence. En effet, les valeurs des descripteurs sont bien distribuées autour de leur moyenne, les minimums et maximums sont aussi proches des cas théoriques.

3.4.2 Résultats

3.4.2.1 Apprentissage

Avant de créer un modèle de prédiction pour chacun des OCRs, nous avons réalisé une régression linéaire univariée entre chaque descripteur et le taux d'erreur de l'OCR. Ces régressions montrent à chaque fois une légère corrélation entre le descripteur et les performances de l'OCR mais ne sont pas assez précises pour prédire les résultats des OCRs individuellement. En suivant la méthodologie présentée en section 3.2.1, nous avons ensuite utilisé une régression linéaire multivariée et séquentielle avec l'ensemble de nos descripteurs afin d'expliquer précisément le résultat de l'OCR. Cette étape permet de sélectionner les descripteurs les plus significatifs pour chaque OCR et de construire les modèles de prédiction présentés en tableau 3.12.

	OCR Abbyy		OCRopus	
Descripteurs	coefficient	p-valeur	coefficient	p-valeur
$\mathcal{MI}_I$	97.11	<0.0001	74.12	<0.0001
$\mathcal{MI}_B$	195.25	<0.0001	147.09	<0.0001
$\mathcal{MQ}$	-	-	0.16	0.003
$\mathcal{MA}$	19.8	<0.0001	-16,99	0.02
$\mathcal{MS}$	-	-	-74,5	<0.0001
$\mathcal{MSG}$	-22.19	0.0008	16.08	<0.0001

TABLE 3.12 – *Modèles de prédiction des taux d'erreurs des deux OCRs (OCRopus et Abbyy Fine Reader). Il est à noter que contrairement à Abbyy FineReader, la totalité des descripteurs est sélectionnée pour expliquer le taux d'erreur d'OCRopus.*

Abbyy FineReader 9

Les descripteurs sélectionnés pour Abbyy sont $\mathcal{MI}_I$, $\mathcal{MA}$, $\mathcal{MSG}$ et $\mathcal{MI}_B$. Le fait que $\mathcal{MQ}$ et $\mathcal{MS}$ ne soient pas inclus dans le modèle de prédiction final semble montrer que $\mathcal{MA}$, $\mathcal{MSG}$ expliquent suffisamment les erreurs d'Abbyy en terme de quantité et de localisation des composantes de transparence. De plus, plusieurs indicateurs statistiques sur le modèle obtenu ont été calculés. Le R^2 qui, pour rappel, donne une indication sur la force du modèle à expliquer le taux de l'OCR, est égale à 0.96 (la valeur optimale théorique étant 1). Le $RMSE$ qui mesure la moyenne des erreurs au carré est égal à 12.77. Ces deux indicateurs statistiques montrent que le modèle explique précisément les données d'apprentissage.

OCRopus 0.4

Pour OCRopus, les descripteurs sélectionnés sont $\mathcal{MI}_I$, $\mathcal{MI}_B$, $\mathcal{MQ}$, $\mathcal{MA}$, $\mathcal{MS}$ et $\mathcal{MSG}$. Chacun des descripteurs est suffisamment significatif pour être inclus dans le modèle final. Le R^2 est égale à 0.99 et le $RMSE$ à 7.5. Ainsi, le modèle prédictif créé pour OCRopus semble légèrement mieux expliquer le résultat de l'OCR. Cela peut être expliqué par le fait que cet OCR est bien moins complexe qu'Abbyy. Par exemple, il semble, d'après l'étude des sources du projet, que OCRopus ne se base que sur une seule binarisation (en l'occurrence Otsu) alors que les méthodes de binarisations d'Abbyy sont plus complexes et propriétaires.

3.4.2.2 Validation

Afin de vérifier la précision des deux modèles, nous avons réalisé une cross-validation. À chaque itération, le corpus de document semi-synthétique est aléatoirement divisé en deux ensembles : 90 % constituent le corpus d'apprentissage, et 10% le corpus de validation. Puis, différents indicateurs sont calculés : le R^2 et le coefficient de la régression linéaire univariée entre la prédiction et le vrai taux d'erreurs (qui doivent être le plus proche de 1), ainsi que le $RMSE$ (qui doit être le plus petit possible). Ces derniers sont moyennés sur la totalité des itérations. L'analyse de ces indicateurs (en tableau 3.13) montre que les deux modèles sont précis.

	Slope Coefficient	R-square	RMSE
OCR Abbyy FineReader	1.006	0.97	11.03
OCROpus	0.99	0.99	7

TABLE 3.13 – *Indicateurs statistiques mesurant les qualités prédictives des deux modèles de prédiction.*

En effet, les R^2 sont très proches de 1 (0.97 pour Abbyy et 0.99 pour OCRopus). Les coefficients des régressions linéaires univariées entre la prédiction et le vrai taux d'erreurs sont aussi proches de 1 pour les deux modèles. Pour finir, le RMSE des deux modèles est faible si l'on considère les bornes minimales et maximales de la prédiction à savoir de 0 à 100.

3.4.3 Vers des modèles de prédictions d'OCRs

La précision des modèles de prédiction des deux OCRs, montre que nos six descripteurs permettent d'expliquer le comportement d'un OCR en fonction de la transparence et donc que la transparence est une dégradation ayant une forte influence sur les erreurs de l'OCR. Cela montre aussi que nos descripteurs peuvent être utilisés pour prédire un système (combinaisons de plusieurs algorithmes) dans son ensemble et pas seulement maillon par maillon comme nous l'avons fait pour les méthodes de binarisation.

Cependant il reste encore beaucoup de travail de recherche pour arriver à créer un modèle de prédiction OCR. En effet, ici seule la transparence est considérée, alors que les OCRs sont sensibles à beaucoup d'autres caractéristiques comme la complexité de la mise en page, la fonte, les figures, les tableaux, etc., d'un document. De plus, les descripteurs de qualité d'images binaires présentés en section 2.1.3.1 ne peuvent pas être considérés lors de la création du modèle prédictif. En effet, nous serions obligés de calculer ces descripteurs après la phase de binarisation alors que nos descripteurs se calculent sur une image en niveau de gris. Par conséquent, et si l'on veut prédire l'OCR, il est nécessaire de créer d'autres descripteurs caractérisant la complexité globale d'un document pour un OCR.

3.5 Conclusion du chapitre

Dans ce chapitre, nous avons présenté un protocole strict permettant la création de modèles prédictifs et se basant sur les descripteurs présentés en section 2.2.3. Cette méthodologie se résume en deux principales étapes :

1. Création du modèle prédictif en sélectionnant les descripteurs les plus significatifs.
2. Validation par cross-validation en effectuant plusieurs séries composées d'une phase d'entrainement du modèle sur 90% d'un corpus et d'une phase de prédiction sur les 10% restants.

De plus, une étude bibliographique des différents modèles de prédiction proposés, pour l'analyse et le traitement d'images de documents, montre qu'il n'existe pas encore d'algorithmes permettant de prédire

les performances de méthodes de binarisation. En effet, la plupart des modèles de prédiction existants se basent sur des descripteurs binaires calculés post-binarisation ou sur des images où les zones de textes sont déjà extraites et binarisées.

Pour résoudre ces problèmes, nous avons proposé des modèles de prédictions pour 11 méthodes de binarisation. Leurs qualités prédictives sont certes variables (11% d'erreur moyenne au maximum, et 2% au minimum), mais suffisantes pour créer un algorithme de sélection automatique de méthodes de binarisation qui soit optimal pour chaque image. Cette sélection automatique permet un gain de 56% sur l'image la plus complexe de notre corpus de tests et un gain moyen de 1.5% par rapport à la méthode de binarisation gagnante de la compétition d'ICDAR. Cette amélioration moyenne des performances s'explique, certes par la détection du pire des cas, mais aussi par la sélection des meilleures méthodes pour 22% des images du corpus. En isolant ces cas spécifiques, nous obtenons un gain en performance d'environ 5% par rapport à Shijian.

De plus, notre méthodologie est évaluée sur deux OCRs modernes à savoir Abbyy FineReader 9 et OCRopus 0.4. La précision des modèles ainsi créés montre deux choses : premièrement que la dégradation de type transparence à une forte influence sur les performances des OCRs et deuxièmement que nos descripteurs peuvent être utilisés pour prédire des systèmes logiciels complexes constitués de plusieurs algorithmes.

Les résultats obtenus par nos modèles de prédiction apportent de nombreuses perspectives de recherche. Premièrement on peut penser à la prédiction d'OCRs. Pour cela, nos descripteurs, caractérisant les dégradations fond-encre, doivent être associés à d'autres, caractérisant la complexité d'analyse d'un document, afin de prédire les performances d'algorithmes de plus haut niveau comme l'analyse de structure physique et logique. Ces descripteurs peuvent par exemple concerner la taille de la fonte ou les structures géométriques d'une page (tableaux, figures, etc.). Deuxièmement, il est surement possible, avec les descripteurs introduits, de donner une indication quant à la qualité subjective ressentie par le lecteur d'une image de document (qualité perceptuelle). Cela permettrait par exemple de rediriger le lecteur vers une autre version du document qui serait de meilleure qualité. La principale difficulté réside dans le choix ou la création de vérités terrains pouvant être utilisées pour réaliser ces perspectives. Or, la création, le partage et l'utilisation de vérités terrains est un problème très complexe qui n'est pas encore complètement résolu. Dans le chapitre suivant nous présentons nos travaux portant sur ce sujet, dans le but de simplifier et d'accélérer la création et la diffusion de vérités terrains.

Chapitre 4

Vérité-terrain pour images de documents anciens : création, utilisation, diffusion

Dans un souci d'évaluation des performances de nos algorithmes, nous nous sommes heurtés à la problématique de création de vérités terrains. En effet, à l'heure actuelle, seuls les domaines de recherches reconnus, comme l'extraction de structures physiques, la reconnaissance de symboles, ou encore la reconnaissance de caractères (OCR), possèdent un choix important de corpus d'images de documents avec leurs vérités-terrains associés. Cependant il n'existe peu voire aucun corpus de documents pour les axes de recherches comme l'évaluation de la qualité ou la restauration d'images de documents.

Nous évaluons l'ensemble de nos algorithmes et de nos approches, en mesurant la pertinence des modèles de prédiction d'algorithmes issus d'axes de recherches (binarisation ou OCR). Les corpus de documents utilisés ont été créés en adéquation avec les types d'algorithmes à évaluer, mais ne permettent pas une évaluation locale des maillons de notre chaîne : précision de l'extraction de pixels de dégradation, robustesse des descripteurs, etc. L'évaluation de ces maillons nécessite la création de vérité-terrains dédiée.

Lors de la création d'un corpus d'images de documents, deux étapes doivent être considérées : la création de la vérité-terrain et sa diffusion (utilisation et partage).

La création de la vérité-terrain est une étape complexe. Le but étant de pouvoir évaluer le plus objectivement possible un algorithme, il est nécessaire de sélectionner les images les plus adéquates, de disposer d'un échantillon suffisant et représentatif des images que l'algorithme sera susceptible de traiter, de valider les informations de vérités-terrains, etc. Ces contraintes sont souvent difficiles à réaliser : faible quantité d'images libres de droits, temps passé à la création, inadéquation des images avec l'algorithme à tester, etc. Nous verrons dans la première section de ce chapitre comment contourner ces contraintes en considérant trois types différents de vérités-terrains : les documents synthétiques ou semi-synthétiques, les annotations réalisées par des experts du domaine et les informations perceptuelles. Ces vérités-terrains peuvent être utilisées conjointement pour évaluer des algorithmes.

La diffusion, l'utilisation et le partage d'un corpus sont aussi des étapes importantes qui ne doivent pas être négligées. En effet, la production scientifique requiert de plus en plus souvent l'évaluation d'un algorithme sur un corpus de documents reconnu. Dans le cas où il n'existerait pas de tels corpus, de fournir publiquement la base de test. De plus, ces étapes permettent de disposer d'une base de référence pour comparer et évaluer de nouvelles approches et de garantir une pérennité du corpus : correction et enrichissement par la communauté de la recherche sur les images de documents. Là aussi de nombreuses difficultés sont à considérer : définition des structures (polygones, boites englobantes, etc.), définition des règles de gestions, interopérabilité, format de fichiers, etc. Il est aussi nécessaire de prendre en compte les différents moyens permettant de valoriser le corpus ainsi que les moyens de l'héberger de façon pérenne. La seconde section de ce chapitre répond en partie à ces problèmes en

proposant l'utilisation d'une plateforme logicielle distribuée permettant de combiner des logiciels de création, d'utilisation, de diffusion et de valorisation de vérités-terrains.

4.1 Les différents modes d'acquisition d'une vérité-terrain pour les images de document

D'après Baird [Bai07], la qualité d'un système de reconnaissance (de caractères, d'écriture manuscrite, de chèques) dépend des caractéristiques et des outils de classification choisis, mais aussi de la qualité de la base d'apprentissage. Cette idée s'applique également au problème de l'évaluation de performances des algorithmes d'analyse d'images.

Nous avons identifié trois façons de créer une vérité-terrain pour l'évaluation d'algorithmes d'analyse d'images.

Une première méthode consiste à créer des documents synthétiques [KH99, KK02]. Ce type d'approche consiste à générer synthétiquement des images de documents. Cette approche peut être utilisée pour générer rapidement de grande quantité de documents afin soit d'enrichir des bases d'apprentissage, soit d'évaluer la robustesse d'algorithmes à certaines caractéristiques pour lesquelles on a défini un modèle. Les informations associées à la vérité-terrain sont en général très précises (si on maitrise la génération du document, on maitrise aussi la génération de la vérité-terrain). Pour cette approche le réalisme des documents peut être critiquable, mais l'annotation se trouve être extrêmement précise. Afin de se rapprocher davantage de documents réels, il est possible de générer des documents semi-synthétiques. Dans ce cas, les images de documents sont constituées de différents éléments (caractères, figures, etc.) extraits de documents réels. Pour cette approche la précision de l'annotation dépend partiellement de la précision des éléments extraits constituant la vérité-terrain. Par exemple, si l'on considère l'extraction de caractères présents dans un document réel, la vérité-terrain des dégradations présentes ou même le détourage de ces éléments n'est pas disponible.

La seconde méthode, la plus répandue, consiste à annoter manuellement une base d'images réelles en utilisant une interface utilisateur [CPA11, HLK03]. Cette méthode à l'avantage de proposer des images réelles. Néanmoins, ce type de vérité-terrain possède deux inconvénients :

1. Elle est complexe à constituer : images libres de droits difficiles à trouver, temps passé à l'annotation de l'image, etc.
2. Les informations annotées peuvent souffrir d'un manque de précision. En effet, comme ces dernières sont créées par des utilisateurs, une part de subjectivité est introduite dans la vérité-terrain.

La troisième [MB11] et dernière catégorie de méthodes correspond à la création d'informations perceptuelles et subjectives souvent liées à la qualité d'une image : qualification de la dégradation, de la lisibilité, restauration d'images de documents, etc.

Nous sommes convaincus que ces trois types de vérités-terrains sont complémentaires et peuvent être utilisées à différents moments du cycle de vie d'un algorithme. Par exemple, dans le cas d'un algorithme de restauration de la transparence. Les documents synthétiques et semi-synthétiques permettent de tester rapidement l'algorithme sur des cas simples. Avec ce type de documents, il est possible de répondre à la question suivante : jusqu'à quel niveau de transparence, l'algorithme est capable de restaurer cette dégradation ? Les documents réels sont ensuite un bon moyen de valider l'algorithme ou d'évaluer ses performances vis-à-vis de l'état de l'art. Par exemple, on peut évaluer l'amélioration produite sur un OCR par l'algorithme de restauration. L'utilisation d'informations perceptuelles peut être un moyen de savoir si l'algorithme fournit, dans la majorité des cas, des résultats similaires à ceux attendus par des utilisateurs finaux. En effet, l'objectif de cet algorithme n'est pas nécessairement d'améliorer les résultats de l'OCR, mais de corriger la transparence afin d'améliorer la qualité perceptuelle de l'image.

Cette section s'intéresse aux différentes façons de constituer une vérité-terrain pour l'analyse d'images de documents. Nous y présentons une à une les trois modalités de création de vérité-terrain. Pour chacune de ces modalités, nous réalisons un état de l'art des méthodes et outils existants, afin de comprendre comment ces derniers répondent aux problèmes liés à la création de vérités-terrains : les contraintes de temps et d'objectivité. Nous proposerons ensuite des solutions aux problèmes restants identifiés.

4.1.1 Génération de documents synthétiques ou semi-synthétiques

Un document synthétique est une image de document qui est créée de toutes pièces par un utilisateur ou un programme. On parle de documents semi-synthétiques lorsqu'une partie des éléments constituant l'image de document créée provient de documents réels. La génération automatique d'images est un domaine de recherche largement étudié comme nous le verrons dans l'état de l'art. Une partie des outils existants reposent sur une modélisation d'un type de dégradation dans l'objectif de pouvoir trouver une transformation inverse permettant la restauration. Une autre partie a pour objectif d'enrichir les bases d'apprentissage d'algorithmes pour en améliorer les performances. Enfin, certains outils ont pour seul objectif d'évaluer la robustesse d'algorithmes en proposant des cas critiques (exagérant certaines dégradations) rares. Dans tous les cas, cette approche permet d'obtenir rapidement une grande variété d'images en jouant sur différents paramètres comme le type de fonte, les défauts présents ou la mise en page. La génération automatique de documents est une première solution paliant aux contraintes de temps induites par la création de vérités-terrains.

4.1.1.1 État de l'art

Les travaux sur la génération d'images de documents ont d'abord porté sur la création de bases de données destinées à la validation et à l'apprentissage de systèmes de reconnaissance de caractères. Les auteurs de [KH99, KK02] se sont intéressés à la génération d'images de texte dégradées. Étant donné un document idéal produit en utilisant LaTeX [1], la méthode proposée consiste à l'imprimer puis à le scanner pour en obtenir une version dégradée. Sur cet exemple, la vérité-terrain consiste à définir le label et la boîte englobante de chaque caractère, elle est obtenue en mettant en correspondance l'image idéale et l'image scannée. Les mêmes auteurs [KHB+00] ont ensuite proposé un modèle de dégradation permettant d'obtenir une image de texte dégradée sans recourir à un processus physique. Ce modèle permet de contrôler l'inversion de pixels (texte-fond) et le niveau de flou.

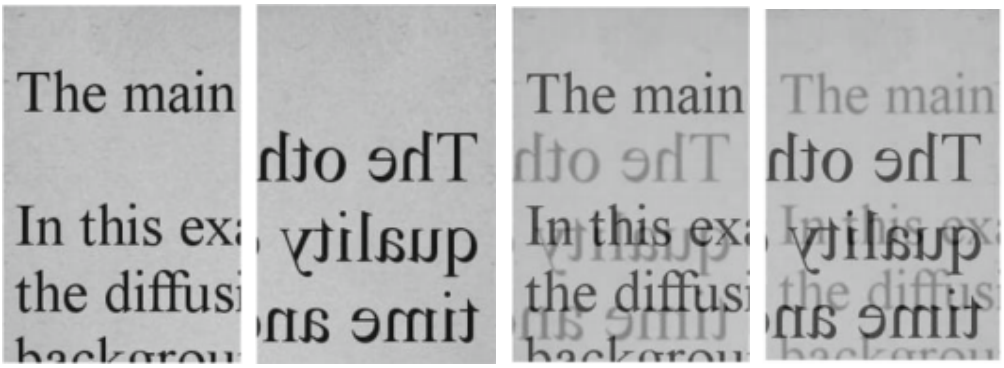
Plus récemment, [HBAT07] ont proposé un logiciel permettant d'intégrer des spécifications lors de l'étape de génération de la vérité-terrain d'images de documents contemporains. Il est ainsi possible de définir une structure logique à respecter (une DTD au format XML), des règles de formatage à appliquer à cette structure logique (feuille de style), la disposition des différents blocs proposés par la DTD (zone de texte, illustrations ...) et enfin un ordre de lecture des différents éléments de contenu.

Dans le domaine des documents techniques, les images synthétiques sont utilisées pour évaluer la performance d'algorithmes de reconnaissance de symboles. Dans [AYS+00], des symboles tirés et retaillés aléatoirement sont positionnés dans l'image de façon à éviter les recouvrements. L'image obtenue est ensuite bruitée. Une approche similaire consistant à ajouter différents types de bruit à une image de dessin technique est proposée dans [ZWDL03]. Plus récemment, [DVPK10] ont proposé une méthode de génération permettant d'obtenir un résultat plus réaliste en ajoutant des connaissances haut-niveau : les symboles sont placés sur un fond prédéfini suivant un ensemble de contraintes de position spécifiques à un domaine particulier comme l'architecture ou l'électronique.

Les travaux concernant la génération d'images ayant les caractéristiques de documents anciens sont peu nombreux. Dans [MC09a], on trouve un modèle de dégradation permettant de simuler la transparence (diffusion de l'encre du verso sur le recto). Ce modèle est ensuite utilisé dans un processus de

1. http ://www.latex-project.org/

restauration d'image. Par ailleurs, les auteurs de [SKP08b] proposent de générer des documents dégradés pour évaluer des algorithmes de binarisation. Les images dégradées sont obtenues en composant une image sans défaut (la vérité-terrain) et une image de fond obtenue en scannant des pages blanches de documents du XVIII e siècle. Les fonds comportent tous les défauts spécifiques aux documents anciens : variations d'intensité, taches, transparence du verso.



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

FIGURE 4.1 – *Exemples d'images de documents semi-synthétiques générées avec le modèle de dégradation d'ajout de transparence proposé par [MC09a]. (a), (b) recto et verso. (c), (d) les images transparentes.*

Comme on peut le voir, un grand nombre d'outils ont été proposés pour générer automatiquement des images de documents synthétiques ou semi-synthétiques. Cependant il n'existe pas de proposition permettant de créer des images de documents anciens semi-synthétiques. C'est à dire utilisant des fonds, des fontes et des mises en pages typiques de documents anciens. La génération de documents synthétiques anciens est nécessaire pour l'évaluation des performances d'algorithmes tant les documents anciens présentent de nombreuses caractéristiques spécifiques (dégradations, fontes et mises en pages, etc.). Pour cette raison, la section suivante est consacrée à la création d'un logiciel permettant de générer automatiquement ce type de documents.

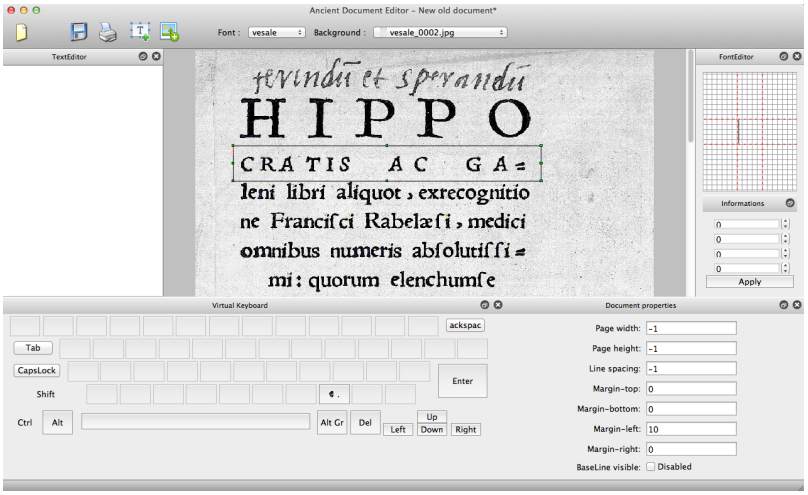
4.1.1.2 Proposition d'un logiciel pour la génération d'images de documents anciens semi-synthétiques

Dans [JVD+10] nous avons décrit une technique de génération d'images de documents semi-synthétiques. Ces images sont générées en utilisant des éléments (caractères, figures, etc.) extraits de documents réels. Comme chaque élément inséré dans le document semi-synthétique possède sa propre vérité-terrain (par exemple, chaque imagette d'un caractère est associée à une lettre), le document résultant possède lui aussi une vérité-terrain contenant les informations sur les éléments qui le composent.

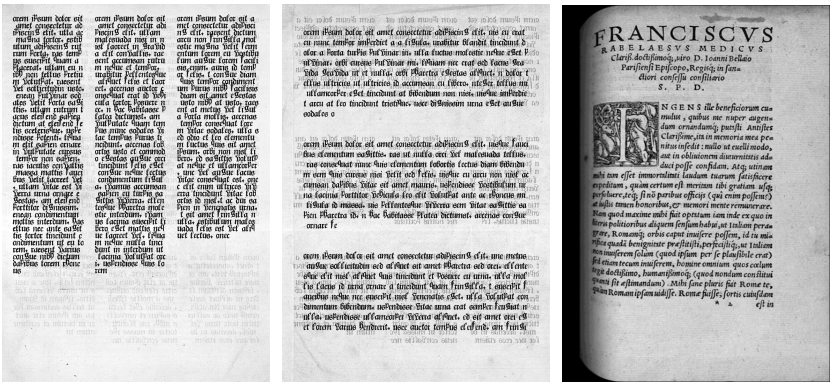
Nous présentons ici plusieurs évolutions de nos précédents travaux permettant d'accélérer la création de vérités-terrains : la création de documents de façon collaborative, la génération automatique de documents semi-synthétiques, et l'ajout de nouveaux modèles de dégradations.

Afin de disposer de documents semi-synthétiques réalistes, il est possible à l'aide du logiciel de constituer des documents qui s'obtiennent par recomposition d'éléments de documents anciens comme présentés en figure 4.2a. Cependant, cette tâche reste une étape longue et fastidieuse. Pour accélérer cette étape, nous avons rendu le logiciel collaboratif : plusieurs chercheurs (ou créateur de vérité-terrain) peuvent maintenant travailler ensemble et en temps réel sur le même document.

Il a été montré [Bai07] que les documents semi-synthétiques mêmes peu réalistes sont utiles à plusieurs niveaux : tests de robustesse, apprentissage, expérimentations. Par conséquent, la génération d'un document synthétique réaliste n'est pas toujours le principal problème auquel le logiciel doit répondre. Pour les problématiques de tests de robustesse ou l'amélioration de bases d'apprentissages, nous avons rendu possible la génération de documents semi-synthétiques de façon aléatoire. Cela permet de générer un très grand nombre de documents rapidement. Les paramètres choisis aléatoirement sont par exemple le nombre de colonne ou de lignes, l'espace inter-ligne ou inter mot, etc. De plus, nous utilisons du texte autogénéré de type Lorem Lipsum [2]. Les figures 4.2b et 4.2c montrent deux documents exemples que le logiciel est capable de générer.



(a)



(b) (c) (d)

FIGURE 4.2 – *Logiciel pour la génération d'images de documents anciens semi-synthétiques (a) et des exemples de documents générés (b,c,d).*

Pour finir, nous avons ajouté plusieurs modèles de dégradation. Il est maintenant possible de générer des documents pour lesquels l'utilisateur a spécifié les défauts présents et leurs caractéristiques.

– **Défaut de transparence :** le premier modèle de dégradation intégré est celui de l'apparition de l'encre du verso sur le recto (effet de transparence). Nous nous sommes basés sur le modèle présenté par les auteurs de [MC09a]. Ce modèle simule la diffusion de l'encre d'une image source (verso) vers une image cible (recto) avec un coefficient de diffusion qui permet de régler la quantité de transparence présente sur le recto. Ce modèle de dégradation est détaillé en section 2.2.2.1.

2. http://fr.lipsum.com/

- **Défaut d'illumination aux abords de la reliure :** le second modèle que nous pouvons simuler correspond à une déformation de la page sur la partie proche de la reliure : la page semble "courbée" sur un de ses bords. Ce défaut apparaît dès lors que l'ouvrage est volumineux et qu'il est numérisé sur un scanner plat. Nous avons utilisé le modèle géométrique proposé par les auteurs de [KHP94]. Ce modèle permet de spécifier un ensemble de paramètres parmi lesquels on trouve l'épaisseur de la reliure, la longueur de la page, la position et les paramètres du bloc optique. Une partie du modèle permet également de simuler le défaut d'illumination induit par cette déformation de la page. En effet, la surface à numériser n'étant pas plane, la lumière ne se réfléchit plus uniformément. L'hypothèse sur laquelle s'appuient les auteurs de ce modèle est que l'illumination en un point de l'image est inversement proportionnelle à la distance qui le sépare de la source lumineuse. De ce fait plus une page se courbe, plus l'extrémité de cette dernière s'éloigne de la source lumineuse et moins la partie courbée est éclairée. Les auteurs de l'article utilisent ce modèle pour corriger le défaut de courbure et l'illumination qui en découle. Nous l'avons adapté pour, au contraire, générer artificiellement ce défaut.
- **Dégradation des caractères :** pour finir, nous pouvons générer des dégradations de l'encre des caractères. En effet, il arrive qu'au fil des siècles, l'encre dans le papier finisse par disparaitre à certains endroits. Sur des lettres, ce défaut donne une impression visuelle de caractères dégradés sur lesquels il est difficile d'observer des contours nets et épais. Comme pour le défaut de courbure, nous avons adapté un modèle qui, à l'origine, vise à corriger ce problème sur des documents réels. Le modèle détaillé dans [KH99] se base sur l'hypothèse que, dans le temps, le niveau de gris d'un pixel d'encre change d'intensité en fonction de sa distance à la frontière de la forme à laquelle il appartient, ce qui permet de définir la probabilité pour un pixel noir de devenir blanc. Il est à noter que ce modèle de dégradation à été étendu aux images en niveaux de gris [KVJ+12].

Nous avons mené deux expérimentations montrant l'utilité de tester des algorithmes d'analyse d'images de documents sur des données dont l'utilisateur maitrise le contenu.

Prédiction des performances d'OCRs

Dans la section 3.4, nous avons utilisé le générateur de documents semi-synthétiques dans le cadre d'un test visant à créer un modèle de prédiction pour deux OCRs différents : Abbyy FineReader et OCRopus. Un total de 190 images de documents ont été générées automatiquement et en jouant sur plusieurs paramètres fixés aléatoirement :

- pour le texte, nous générons des paragraphes de type *Lorem Ipsum* [3],
- la taille des marges varie de 50 à 200 pixels,
- l'espace inter-ligne varie de 30 à 50 pixels,
- le nombre de colonnes et de blocs de lignes varie de 1 à 3, l'espace inter colonne et interligne varie aussi entre 30 et 100 pixels,
- le pourcentage de texte sur le document final varie de 0% (pour générer des pages blanches), à 100%,
- le fond est choisi aléatoirement parmi trois fonds extraits de documents anciens,
- une fonte est aussi choisie aléatoirement ; chacune des fontes est extraite manuellement depuis des documents anciens numérisés ; un caractère est représenté par plusieurs images ce qui permet de varier les dégradations présentes sur un même caractère,
- à chaque image de document, nous ajoutons plus ou moins de transparence, certaines images ne contiennent pas de transparence.
- la taille des images est $2824 * 4268$.

Mis à part le modèle de prédiction obtenu, il a été possible, grâce à ce logiciel, de déterminer la robustesse des deux OCRs vis-à-vis du défaut de transparence. Les tests ont permis de faire ressortir les niveaux de transparence pour lesquels le taux de reconnaissance des OCRs chute significativement. Les résultats obtenus montrent que même si Abbyy propose, en moyenne, de bien meilleurs résultats

3. www.loremipsum.fr

qu'OCRopus, et ce sans utiliser de dictionnaires, le taux d'erreurs moyen sur des documents anciens transparents peut avoisiner les 60%.

Classification de fontes dans les documents anciens

Dans [JVD+10] nous avons effectué un ensemble de tests visant à déterminer si les méthodes de classification de fontes contemporaines pouvaient être utilisées pour la classification de fontes anciennes. Le générateur de document a été ainsi utilisé pour générer 7820 zones de textes à l'aide de 12 fontes différentes, le nombre de lignes par zones pouvant varier de 1 à 15. Les résultats montrent qu'une simple transposition des algorithmes d'analyse de fontes contemporaines ne permet pas de différencier correctement les 7820 images réparties sur 12 fontes différentes. En effet, dans le meilleur des cas (90% de la base en apprentissage et 10% pour la reconnaissance) le taux de reconnaissance ne dépasse que de très peu les 60%.

4.1.1.3 Perspectives

Ces travaux ouvrent de nombreuses perspectives de recherche. Tout d'abord certains modèles de dégradation comme celui utilisé pour dégrader les caractères ne s'utilisent que sur des versions binarisées d'images de documents anciens. Il est par conséquent nécessaire de trouver un moyen de les adapter aux images en niveaux de gris. De plus, le modèle de dégradation générant une distorsion de l'image due à la profondeur de la reliure et intégré au logiciel ne permet pas de modéliser des ondulations (humidité, usures, etc.) réalistes et propres aux documents anciens. Des travaux proposés dans le cadre du projet ANR DIGIDOC [4] se concentrent sur l'adaptation de ces modèles de dégradation au cas des documents anciens (niveaux de gris et couleurs) [KVJ+12]. Il serait aussi intéressant de mesurer le réalisme des images autogénérées. Ces mesures peuvent être réalisées à plusieurs niveaux :

– pour prouver statistiquement un modèle de dégradation, il est possible d'analyser la similarité des résultats d'évaluateurs de performances à la fois sur un corpus synthétique et sur un corpus réel (documents modèles) ;
– pour mesurer le réalisme d'une image synthétique, des questionnaires utilisateurs peuvent être réalisés (à l'aveugle) sur des images réelles et synthétiques. Par exemple, il est possible de demander à un utilisateur, si l'image qu'il visionne est selon lui, synthétique ou réelle. Il est alors possible de mesurer le nombre de mauvaises réponses pour les images synthétiques et celui de bonne réponse pour les images réelles afin de déterminer si les images générées sont réalistes.

Dans les deux cas, il est nécessaire de lier les vérités terrains de documents réels et celles de documents synthétiques afin de les utiliser conjointement. En section 4.2 nous présentons une plateforme logicielle qui peut être utilisée pour ce genre de tâches.

4.1.2 Annotation d'un document réel par un expert

Si posséder une large base d'images de documents synthétiques ou semi-synthétiques permet de tester rapidement et de manière précise une chaine de traitements, cela n'exclut en aucun cas l'étape d'évaluation de performances sur des documents réels. Ce problème a été abordé à plusieurs reprises par la communauté scientifique *via* la conception de logiciels capables d'annoter les éléments d'une image de documents. L'annotation des éléments d'une image de document est réalisée manuellement par un utilisateur qui doit détourer (par des rectangles ou des polygones) ces derniers et leur affecter un label. La zone détourée et le label constituent l'annotation de vérité-terrain.

4.1.2.1 État de l'art

Les logiciels existants

4. Voir : http://digidoc.labri.fr/ et https://sites.google.com/site/kieuvancuong/.

La majorité des auteurs proposant une méthode de segmentation ou de classification d'images de documents font référence à une base annotée manuellement soit pour l'évaluation de performances soit pour constituer des bases d'apprentissage (détourage de zones contenant du texte, des illustrations...). La vérité-terrain est ensuite comparée aux résultats obtenus. Malgré la simplicité des besoins et concepts à implémenter (détourage de zones, labellisation des mêmes zones) un très grand nombre de systèmes logiciels ont été conçus par la communauté de la recherche en analyse et traitement d'images de documents (spécificités résumées tableau 4.1) :

- Pink Panther (1998) [YV98] implémentait déjà un très grand nombre de concepts : détourage de zones sous forme de polygones, labellisation des zones, ordre de lecture, relations entre zones ... Développé en C, ce logiciel est multiplateforme. Le format de sortie de la vérité-terrain est en texte ACSII. Le développement du logiciel semble avoir été abandonné, ce dernier n'étant plus disponible au téléchargement.
- TrueViz (2003) [HLK03] est un logiciel possédant des fonctionnalités semblables à Pink Panther. Il a été créé en 2003. Ce logiciel multiplateforme (développé en Java et C) se concentre sur la création de vérités-terrains pour l'OCR ainsi que la visualisation, la correction et l'évaluation des résultats d'OCRs. Le format de représentation de la vérité-terrain est un fichier XML, mais plusieurs importeurs/exportateur vers des formats ACSII existent. Ce logiciel semble toujours maintenu.
- PerfectDoc (2005) [YSS05] est un logiciel qui ajoute aux fonctionnalités de TrueViz une visualisation multi pages d'un document ainsi que la possibilité de préciser l'ordre de lecture "multi pages". Ce logiciel multiplateforme est lui aussi écrit en Java. Les auteurs portent une attention particulière à son ergonomie. Cependant la création de zones d'informations sous forme de polygones n'est pas possible.
- PixLabeler (2009) [SLS09] est un logiciel propriétaire, multiplateforme et écrit en Java se concentrant sur la création de vérités-terrains au niveau pixels. Plusieurs outils ergonomiques sont proposés pour accélérer le travail de l'expert. Cependant l'annotation d'un pixel se réalise à partir d'une image binarisée et il ne semble pas possible de pouvoir corriger cette binarisation dans le cas de pixels manquants.
- GEDI Ground truth editor and document interface (2010) [DZL10] est un logiciel Open source, multiplateforme et écrit en Java. Ce logiciel riche en fonctionnalités se distingue de ses "concurrents" en proposant que l'annotation du document soit réalisée en suivant un "workflow" (par exemple, extraction des lignes, transcription, extraction des mots ...). Le logiciel propose aussi un éditeur de scripts pour automatiser certaines tâches répétitives sur plusieurs pages similaires, ainsi que la possibilité de contrôler certains éléments de visualisation depuis d'autres programmes externes.
- DAE Document Analyse and Exploitation (2011) [LL11a] est une application web écrite en JavaScript permettant d'annoter (sous forme de rectangle) les différents éléments d'une page. Cette application se distingue des autres par son approche collaborative. En effet, l'application étant hébergée sur un site web, l'ensemble des informations constituant la vérité-terrain sont enregistrées sur une base de données qui peut être modifiée par plusieurs chercheurs ou experts.
- Aletheia (2011) [CPA11] est aussi un logiciel d'annotation de documents. Les auteurs de ce dernier remarquent que les applications existantes comme GEDI ou TrueViz manquent d'ergonomie et d'interactivité permettant d'augmenter la productivité des chercheurs ou experts, et qu'ils présentent des problèmes de performances (dûes au langage Java selon les auteurs) sur des images de grande taille ($> 30Mo$). Pour contourner ces problèmes, les auteurs décident donc de créer une application C++ se concentrant sur l'ergonomie et la correction des résultats d'algorithmes (extraction de structures physiques et logiques, OCR, ...) afin de construire la vérité-terrain de façon semi-automatique.

Nous remarquons une très grande diversité de logiciels permettant la création de vérités-terrains pour les algorithmes d'extraction de structures physiques et logiques ainsi que pour les OCRs. Cette diversité permet à chacun de sélectionner l'outil qui conviendra le mieux à ses besoins et à ses goûts. Par exemple, TrueViz semble mieux adapté à l'annotation manuelle de "petits" corpus alors que l'automatisation proposée par GEDI ou Aletheia permet de travailler sur de plus grands corpus.

Il est à noter que beaucoup de logiciels proposent des approches innovantes permettant de répondre en partie aux contraintes de temps passé à la création de vérités-terrains. DAE met en avant une approche

Application	Format Export	Langage	OpenSource	multi plateforme	semi-automatique
Pink Panther	ACII	C	Inconnu	Oui	Non
TrueViz	XML	Java	Oui	Oui	Non
PerfectDoc	XML	Java	Oui	Oui	Inconnu
PixLabeler	XML	Java	Non	Oui	Non
GEDI	XML	Java	Oui	Oui	Oui (Externe)
DAE	Pas d'export	JavScript	Oui	Oui	Non
Aletheia	XML	C++	Non	Non	Oui

TABLE 4.1 – *Tableau résumant les spécificités techniques et fonctionnelles des logiciels d'annotation existants.*

collaborative pour que plusieurs chercheurs puissent contribuer ensemble à la création d'un même corpus de documents. C'est aussi le cas du système EPEIRES qui permet à un ensemble d'utilisateurs et d'experts de construire ensemble une vérité-terrain pour les d'images de documents graphiques. Les images sont centralisées sur un serveur et il est possible pour les experts de corriger les annotations. Ce système n'a pas donné lieu à une publication académique, mais il est décrit dans [DVPK10]. GEDI propose quant à lui plusieurs modules permettant d'automatiser une partie des étapes liées à la création de vérités terrains. Pour finir, nous remarquons l'effort réalisé par les développeurs de Aletheia pour proposer une application ergonomique et semi-automatique.

Multiplication des formats de fichiers

L'utilisation de fichiers XML pour représenter les vérités terrain permet de garantir une certaine interopérabilité entre différentes applications (évaluateur de performances, apprentissage, etc.). Cependant, nous remarquons que chaque application propose sa propre formalisation XML et ce, même si la plupart des informations sont semblables d'un logiciel à un autre. En effet, la plupart de ces formats représentent des éléments communs : zones (polygones ou rectangle) typées (Ligne, Figure, Mot, Tableau, Paragraphe) et possédant des valeurs (textes, images, binarisation ...), informations sur le document (date d'édition, de création, identifiant), et des références vers d'autres fichiers pour permettre l'extensibilité. Ainsi, un logiciel d'évaluation de performances doit être capable de lire et comprendre plusieurs formalisations XML

Conclusion sur l'état de l'art

Cet état de l'art des différents logiciels d'annotation de documents par des experts permet de dresser une liste des fonctionnalités qui doivent être implémentées sur ce type de logiciels :

– Interactions riches avec l'utilisateur (proposées par GEDI et Aletheia) : corrections d'algorithmes, outils ergonomiques dédiés à la création d'une annotation spécifique .... Cet aspect permet d'accélérer les temps de création d'une base.
– Évolutive (à l'image de GEDI) : l'ajout de composants logiciels (interacteurs, algorithmes, etc.) doit se faire simplement afin que chacun puisse adapter le logiciel à des besoins spécifiques.
– Collaboration (proposée par DAE) : la vérité-terrain créée doit être facilement enrichie (création, modification, correction) par plusieurs personnes (experts, modérateurs, chercheurs) en même temps. Cet aspect permet de garantir l'objectivité scientifique de la base et d'accélérer les temps de création d'une base.
– S'abstraire d'un format de fichier spécifique (à l'image de TrueViz) : comme nous avons pu le voir, il n'existe pas encore de format de fichiers standard pour représenter une vérité-terrain. Pour cela, il est important de ne pas avoir de dépendances vers un format plutôt qu'un autre et d'être compatible avec le plus de formats possible.

– Multiplateforme : afin encore une fois d'accélérer les temps consacrés à la création et à l'utilisation d'une base (intégration de différentes applications), il est nécessaire de proposer un logiciel utilisable sur la plupart des plateformes actuelles.

L'ensemble des logiciels présentés se concentre sur la création de vérités-terrains pour l'analyse de structures physiques ou logiques, ainsi que l'OCR. Par conséquent, ils ne permettent pas de répondre à nos besoins : créé une vérité-terrain des "dégradations" d'images de documents. Dans la sous-section suivante, nous présenterons notre proposition de logiciel répondant aux exigences citées précédemment et permettant la construction d'une vérité-terrain de la qualité d'une image de document.

4.1.2.2 Proposition d'un logiciel collaboratif dédié à l'annotation de la qualité des documents anciens

Il n'existe pas à l'heure actuelle de logiciel dédié à la création de vérités terrains pour la qualité des images de documents. Ce type de vérités-terrains peut pourtant être utile aux algorithmes de restauration comme [SG04a, MC09b, BSDLS11] qui se concentrent sur la correction des documents anciens soufrant de perturbations fond-encre. Dans ces articles, les algorithmes sont souvent évalués sur des vérités terrains non partagées soit de façon visuelle (subjectif) soit en analysant les améliorations produites sur des algorithmes de plus haut niveau comme les OCRs. L'utilisation d'un logiciel simplifiant la création de ce type de vérités terrains permettrait de se répartir les tâches d'annotation d'images de documents et de disposer d'une base d'images libres de droits dont l'objectivité scientifique serait établie par nos pairs.

Dans cette section, nous présentons un logiciel multiplateforme et écrit en C++, permettant à un ensemble d'experts ou de chercheurs d'annoter une image de document. Nous nous concentrons sur la création de vérités terrains de plusieurs types d'informations :
– Annotation des dégradations (taches, transparence, etc.).
– Sélection du verso correspondant à une image (recto).
– Recalage de la paire recto verso.
Ce logiciel doit répondre aux contraintes exposées dans la section précédente afin de garantir l'objectivité scientifique des annotations et d'accélérer le temps passé à la création de cette vérité-terrain.

Cette application fait partie intégrante d'une plateforme logicielle présentée plus tard en section 4.2. Cette plateforme permet au logiciel d'être collaboratif. En effet, la plateforme reposant sur une architecture client-serveur, chaque modification appliquée à une image est synchronisée de façon transparente sur l'ensemble des postes clients. Ces mécanismes sont un peu plus détaillés en section 4.2.

Pour garantir l'évolutivité du logiciel, nous l'avons basée sur un moteur de "plug-ins". Ainsi, le logiciel dans sa version basique une interface vide sur laquelle chaque chercheur peut intégrer ses propres composants. Les "plug-ins" peuvent être de plusieurs types :
– Vues : les vues sont des composants logiciels présentant à l'utilisateur les données constituant la vérité-terrain. Cela permet aux chercheurs de modifier la façon dont certains éléments sont présentés à l'utilisateur et par conséquent de concevoir les vues les plus adaptées aux types de vérités-terrains à afficher.
– Algorithmes : les algorithmes sont des composants qui ont pour objectif de simplifier le travail de l'utilisateur. Par exemple, si l'on prend la création de vérités-terrains pour la binarisation, un algorithme possible peut être la méthode de binarisation de Sauvola. L'utilisateur pourra alors exécuter cet algorithme pour ensuite corriger les erreurs réalisées par ce dernier. L'utilisation de "plug-ins" à ce niveau permet de proposer à l'utilisateur des algorithmes récents de l'état de l'art.
– Intéracteur : les interacteurs sont des composants qui permettent de contrôler les vues (déplacements, rotation, ajout, correction des résultats proposés par un algorithme, etc.). La création de "plug-ins" d'interaction permet de proposer à l'utilisateur une ergonomie évolutive et pouvant s'adapter à des besoins spécifiques.

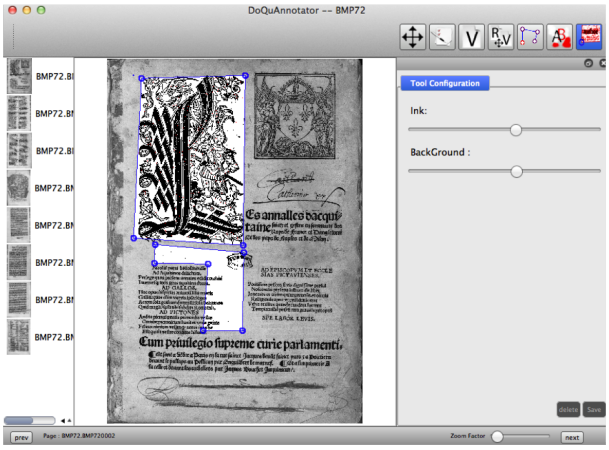


FIGURE 4.3 – *Exemple d'une option du logiciel d'annotation. Via cette interface, il est possible de délimiter une zone sous la forme d'un polygone et d'y associer des attributs (en exemple, sur la figure, une trinarisation).*

– Importateur et exportateur : ce dernier type de composants permet à chaque chercheur de proposer une conversion des informations (constituants la vérité terrain) vers d'autres types de format de fichier (PAGE, GEDI, etc.), ou alors d'importer des informations depuis un format de fichier spécifique. Seules les informations pouvant être représentées d'un côté ou de l'autre de la conversion seront considérées. Cette fonctionnalité se base sur la notion de dépôt qui sera détaillé en section 4.2.2.1.

Un certain nombre de plug-ins "prototypes" ont été réalisés et apportent les fonctionnalités suivantes :
– visualiser le rendu d'une binarisation ou d'une trinarisation en fonction des paramètres que l'utilisateur a précisés. Les résultats de ces algorithmes peuvent être corrigés avec un outil pour changer un pixel de classe. Ces algorithmes peuvent être appliqués sur des zones spécifiées par l'utilisateur comme le montre la figure 4.3.
– détourer, sous forme de polygones, différentes parties d'une image et y associer des attributs prédéfinis (*tache*, *transparence*, *trous*, *caractères dégradés*, etc.). Pour chacun de ces attributs, il est possible de définir une valeur (plusieurs types sont supportés *booléen*, *nombre*, *chaine de caractères*, *date*, etc.).
– associer à une image son verso.
– trouver la transformation affine permettant de recaler le recto et son verso.

4.1.3 Acquisition de vérités-terrains de type perceptuel

Nous nous intéressons ici aux algorithmes qui mesurent la "qualité" d'une image. Par qualité, on entend un ensemble d'informations subjectives sur l'image comme la présence plus ou moins gênante de défauts visibles. Ce type d'information peut être nécessaire à plusieurs types d'algorithmes par exemple :
– Les d'algorithmes de compression d'images de documents afin d'optimiser la place occupée par l'image sans perdre en lisibilité ;
– Le contrôle qualité automatique qui est lié à un jugement subjectif réalisé par un expert ayant pour objectif de déterminer si la qualité de l'image numérique correspond au cahier des charges.
– Ou encore, la preuve statistique de modèles de dégradations (création de documents synthétiques ou semi-synthétiques).

Dans cette section nous présenterons les différentes techniques utilisées dans l'état de l'art pour créer des vérités terrains perceptuelles, puis, nous proposerons notre propre méthode et la validerons statistiquement.

4.1.3.1 État de l'art sur Acquisition d'informations perceptuelles

Pour évaluer la qualité perceptuelle d'une image, les algorithmes proposés dans la littérature s'appuient sur des critères objectifs et quantifiables [MB11] : écart relatif à une image de référence (structural similarity index, visual information fidelity), analyse des contours pour mesurer l'importance du flou ou les artéfacts liés à la compression. Morthy et *al* s'intéressent à l'évaluation de ces algorithmes qui s'appliquent à une grande variété d'images (images 2D, images animées, images 3D). En général, la qualité visuelle perçue est mesurée en montrant une série d'images à un ensemble d'observateurs. Chacun d'eux note les images sur une échelle de qualité prédéfinie. La qualité d'une image est finalement mesurée par sa note moyenne (MOS : mean opinion score). On peut alors évaluer un algorithme qui produit une mesure dite objective de qualité visuelle en corrélant ses résultats avec ceux de l'étude perceptuelle. L'article cite des exemples de bases de données utilisées pour ce type d'étude et en particulier celle décrite dans [PLZ+09]. Cette base comporte 1700 images construites en appliquant des distorsions variées à un ensemble d'images réelles, dites de référence. Attribuer un score de qualité à chaque image a mobilisé 800 juges ce qui met en évidence la principale difficulté de la création d'une vérité-terrain de type perceptuel : les moyens humains nécessaires peuvent être considérables. Le travail effectué par chaque juge consiste à évaluer la qualité de toutes les versions distordues d'une même image de référence (68 images). L'originalité de l'approche choisie pour attribuer un score à chaque image réside dans le fait que l'évaluation n'est pas absolue, mais relative. En effet une opération élémentaire consiste à comparer deux images distordues relativement à l'image de référence : le testeur choisit l'image qui lui parait la plus proche de l'image de référence. En utilisant le *swiss competition principle* pour limiter le nombre de paires d'images à comparer, chaque image se voit attribuer une note entre 0 et 9.

Cependant, on ne trouve dans la littérature aucune proposition spécifique aux images de documents. Dans ce domaine, les informations perceptuelles qu'on cherche à acquérir sont liées à l'impact des défauts typiques des images de documents sur la lisibilité : transparence, présence de taches, surcompression de l'image... De plus, nous ne disposons pas d'images de références. Dans la section sous-section suivante, nous présentons un logiciel utilisant une tablette tactile et permettant de trier une base d'images selon un critère de perception visuel.

4.1.3.2 Création de vérités-terrains qualitative par classement relatif d'images

Dans cette sous-section, nous proposons une méthodologie permettant d'associer un jugement qualitatif à un ensemble d'images. C'est ce que nous appelons une vérité-terrain de type perceptuelle.

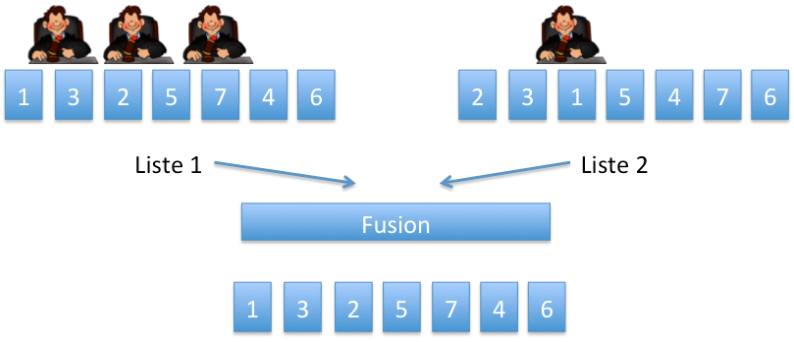


FIGURE 4.4 – *Aspect collaboratif du logiciel : plusieurs classements sont réalisés par plusieurs juges. Le classement final est obtenu par fusion (moyenne des rangs) de l'ensemble des classements.*

L'approche classique consistant, pour un évaluateur, à noter la qualité d'une image selon une échelle absolue est la méthode la plus utilisée (*cf.* section 4.1.3.1). Cependant cette méthode apporte certains

biais influant sur l'objectivité scientifique de la vérité-terrain créée. Tout d'abord, le juge à besoin de se représenter l'échelle de notation afin d'associer un ordre de grandeur à l'importance des dégradations. Pour cela il faut avoir préalablement visionné l'ensemble des images du corpus pour déterminer les images critiques (maximums et minimums). De ce fait, le corpus d'images n'est plus maintenable : l'ajout d'une image qui ne serait pas comprise dans l'échelle de notations du juge obligerait à renoter l'ensemble des images du corpus. D'autres biais techniques, comme les différences de calibration et de résolution entre les écrans utilisés pour les tests, perturbent les résultats.

FIGURE 4.6 – *Interface de sélection de l'image répondant le plus au critère subjectif.*

Afin de garantir l'objectivité scientifique de la vérité-terrain, nous avons donc choisi de ne pas utiliser un système de notation absolue, mais de construire notre vérité-terrain perceptuelle à partir d'une succession de comparaisons d'images répondant à une question ciblée sur un critère de qualité particulier. L'ensemble de ces comparaisons constitue une liste d'images triées selon la réponse des différents juges. L'application étant collaborative, plusieurs juges peuvent contribuer ensemble, à la création de la vérité-terrain. Cela permet tout d'abord d'accélérer les temps de création de la base, mais aussi de minimiser les "erreurs subjectives" sur la liste. De plus, afin de minimiser les "erreurs subjectives" commises sur une image, nous utilisons plusieurs listes qui seront fusionnées pour constituer la liste finale. Cette méthode est schématisée en figure 4.4. Par ailleurs, nous proposons d'utiliser un environnement technique unique pour les tests : un iPad. Cet environnement nous permet de disposer d'un même écran pour chaque juge, de contrôler la luminosité de ce dernier.

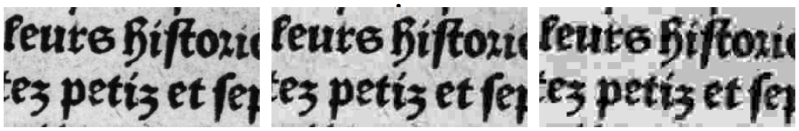
Pour construire une liste d'images triées selon un critère perceptuel, nous procédons par insertions successives. Pour déterminer l'emplacement d'une nouvelle image dans une liste triée, l'image est comparée successivement à un sous-ensemble des éléments de la base. La comparaison consiste à sélectionner, selon le critère de tri, la meilleure de deux images proposées (voir figure 4.6). En fonction de la réponse, l'image à classer est comparée à un autre élément de la base. Cette action est répétée tant que le rang de l'image à classer n'a pas été déterminé. L'insertion est une insertion par dichotomie. Une image est ainsi classée en $\log_2(N)$ comparaisons maximum. Afin d'avoir un classement pertinent, l'utilisateur n'est pas autorisé à juger les deux images proposées comme étant équivalentes.

Expérimentation et validation de la méthode

Pour évaluer cette méthode de création de vérités-terrains, nous avons utilisé deux bases d'images synthétiques. Les images synthétiques présentes dans ces deux bases sont de plus en plus dégradées en fonction d'un critère. Il est donc possible de trier l'ensemble des images en fonction de ce même critère.

Le test statistique a pour objectif de prouver que notre méthode permet de réaliser un tri similaire à celui des bases d'images synthétiques. La première base (figure 4.7a) contient 100 images avec 4 différents niveaux de transparence. La seconde base (figure 4.7b) contient 24 images compressées, avec l'algorithme JPEG 2000, sur 8 niveaux de qualité (entre 0 et 100). Sur chacune de ces bases d'images, nous avons demandé à plusieurs utilisateurs de juger leur qualité en fonction d'un critère perceptuel. Pour la première base, nous avons posé la question : "quelle image possède le plus de transparence" ? Et, pour la seconde base la question : "quelle image est la plus compressée" ? Dans ce cas simple la qualité perçue d'une image est directement liée à son taux de compression ou à son taux de transparence, ce qui permet de valider la vérité-terrain acquise.

(a)



(b)

FIGURE 4.7 – *Les bases d'images synthétiques utilisées pour valider notre méthode. Sur la première ligne, les images transparentes, sur la seconde les images compressées.*

Pour trier la base d'images de manière perceptuelle, nous avons choisit 4 utilisateurs n'étant pas experts dans l'analyse et le traitement d'images. Chaque utilisateur a contribué à une seule et même liste. L'ordre des images présentes dans les 4 listes construites a été ensuite comparé avec l'ordre de la vérité-terrain grâce au test statistique de Kappa. Le test de Kappa a pour but de mesurer l'accord observé entre deux jugements qualitatifs :

$$K = \frac{P_o - P_a}{1 - P_a}$$

Avec, P_o la proportion d'accord observée et P_a la proportion d'accord aléatoire [Gre90].

Le coefficient de Kappa K est compris entre -1 et 1. L'accord est d'autant plus élevé que la valeur de Kappa est proche de 1 et l'accord maximal est atteint ($K = 1$) lorsque $P_o = 1$ et $P_a = 0.5$. On estime qu'un accord est bon quand le coefficient de Kappa est supérieur à 0.60 [C$^+$60]. Le Kappa utilisé est pondéré afin d'accorder moins d'importance aux petits écarts de rang entre deux jugements [Coh68].

	Liste 1	Liste 2	Liste 3	Liste 4	Liste fusionnées
Kappa (compression jpeg)	0.78	0.82	0.86	0.68	0.88
Kappa (transparence synthétique)	0.88	0.93	0.85	0.75	0.93

TABLE 4.2 – *Coefficient de Kappa entre chaque liste (une liste par utilisateur et la liste fusionnée) et le classement idéal. Chaque utilisateur a fait un bon score (> 0.60), mais la liste fusionnée est encore meilleure puisqu'elle minimise les erreurs faites par les utilisateurs. Les coefficients de Kappa supérieurs à 0.80 peuvent être considérés comme excellents [C$^+$60].*

Le tableau 4.2 synthétise les résultats obtenus lors de nos tests. Les utilisateurs ont tous réalisé un bon score indépendamment des uns des autres, ce qui confirme que le protocole proposé est pertinent. La liste fusionnée possède un score encore meilleur dans la mesure où elle lisse les écarts entre les rangs fixés par les différents utilisateurs. Par conséquent, la méthode proposée permet de construire une liste d'images triées en fonction d'un critère de qualité, et ce de façon objective sans nécessiter d'images de références. La liste d'images ainsi créée peut être utilisée comme vérité-terrain.

Nous avons pour perspective d'accélérer encore le temps de création de vérités-terrains par l'utilisation d'algorithmes qui, basés sur des descripteurs, seraient capables de limiter le nombre de comparaisons réalisées par le juge.

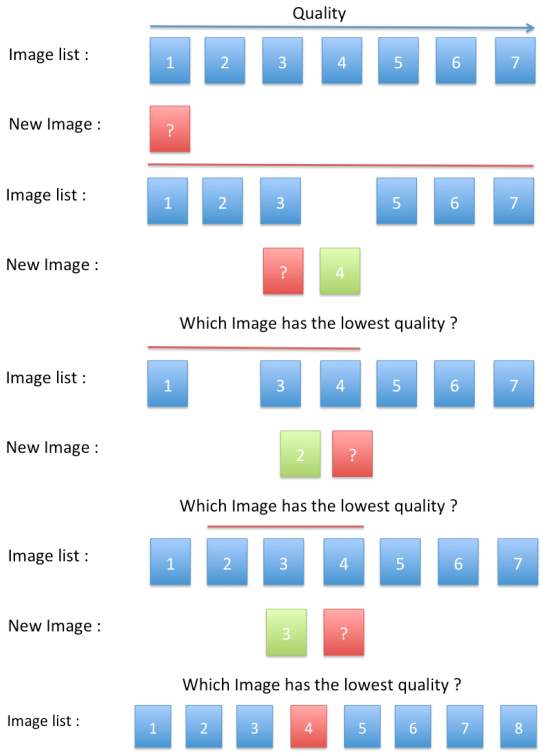


FIGURE 4.5 – *Processus de création d'une liste par dichotomie*

4.2 Plateforme collaborative de création et partage de vérités-terrains

Dans cette section nous proposons une solution logiciel permettant de simplifier la création, la diffusion et l'utilisation de vérités terrains. Pour répondre à des préoccupations collaboratives ou de distribution de l'information, les techniques liées à internet se sont imposées. À l'heure actuelle, les logiciels sont pensés avec des architectures distribuées se basant sur des services web. On parle d'architectures SOA (Architecture Orientée Service) et EDA (Architecture Dirigée par Événements) [Bie09]. Les problèmes de diffusion de vérités terrains sont similaires à ceux rencontrés dans d'autres domaines (synchronisation des données, droits de diffusion, coûts, problèmes techniques). Il existe plusieurs sites web répondant à ces problèmes et proposant d'héberger un ensemble, de données variées et de services (accès, traitements, etc.) pour manipuler ces dernières. De façon générale, on peut citer Dropbox [5], Amazon S3 [6], GoogleDrive [7], et d'autres dédiés à l'image comme PicasaWeb [8] et FilckR [9]. En recherche, il existe un très grand nombre de projets utilisant ce type d'architectures parmi lesquels on peut citer par exemple DGEMap [BVHBA09] sur la recherche sur le génome humain, le projet CICC (Chemical Informatics and Cyberinfrastructure Collaboratory) [DGG+07] de recherche en chimie utilise aussi une architecture distribuée essentiellement basée sur des services web, en astronomie le projet européen HELIO [BCA+11] (Heliophysics Integrated Observatory) utilise aussi une plateforme web. L'ensemble de ces sites web propose différentes façons d'accéder à l'ensemble des services proposés. Les bibliothèques numériques ne font pas exceptions. On peut citer par exemple Gallica [10] ou OpenLibrary [11] qui proposent toutes deux

5. https://www.dropbox.com/
6. http://aws.amazon.com/fr/s3/
7. https://drive.google.com/
8. picasaweb.google.com/
9. www.flickr.com/
10. gallica.bnf.fr/
11. http://openlibrary.org/

des services web permettant de réaliser des recherches sur ces mêmes sites web. Ces API permettent de créer des logiciels capables d'abstraire certains détails techniques. On parle d'"application connectée". Dans la suite de cette section, nous utilisons le terme plateforme logiciel comme l'ensemble constitué d'"application connectée" (clients) et de différentes applications web (serveurs).

La création d'une plateforme logicielle dédiée à la recherche sur les documents anciens est une tâche complexe. Tout d'abord, se pose des problèmes techniques liés à l'utilisation de grandes quantités d'informations (images et annotations) sur des architectures distribuées par exemple, la haute disponibilité des informations, le coup des transferts réseau, etc. À ces problèmes techniques, se rajoute de vrais problèmes scientifiques issus de notre domaine métier : structuration de la vérité-terrain (boites englobantes, polygones, scan-lines), type de services proposés (recherches, algorithmes), etc. Comme le soulignent de nombreuses publications (dont les principales sont [SRG03, Nag10, LL11b, LL12]), ces problèmes scientifiques font l'objet de réflexions importantes au sein de la communauté de la recherche en analyse et traitement d'images de documents. Ces réflexions ont mené à la réalisation de deux plateformes logicielles : DAE [LL12] (Document Analysis and Exploitation) et celle utilisée par les membres du projet IMPACT [12] (IMProving ACcess to Text). La plateforme DAE est aussi une source d'inspiration pour des projets de recherches en dehors du domaine des images de documents comme dans [DWH12] présentant de façon conceptuelle une plateforme dédiée à l'accès d'images médicales. De manière générale, ces plateformes s'intéressent aux problèmes suivants :
– la pérennisation ainsi que la diffusion et l'utilisation des corpus d'images,
– la pérennisation, la diffusion et l'utilisation des algorithmes de recherches applicables aux images,
– garantir la reproductibilité des expérimentations,

Dans cette section, nous commencerons par présenter les plateformes web existantes et dédiées à la recherche sur l'analyse et le traitement d'images de documents. Cet état de l'art nous permettra de voir, de façon plus détaillée, à quels problèmes ces dernières répondent ainsi que les avantages à utiliser ce type d'architecture logicielle. Nous montrerons par la même occasion que ces plateformes ne répondent pas entièrement à nos besoins de création, diffusion, partage et utilisation de vérité-terrains. C'est pour cette raison que nous proposons une plateforme web dont l'objectif principal est de résoudre en partie ces besoins. La dernière partie de cette section est consacrée à la description des concepts implémentés sur cette plateforme.

4.2.1 Les plateformes logicielles dédiées à la recherche sur l'analyse et le traitement d'images de documents.

La communauté de recherche pour l'analyse et le traitement d'images de documents propose, à notre connaissance deux principales plateformes logicielles distribuées et dédiées à l'exploitation académique d'images de documents. Nous allons voir que ces deux plateformes proposent de réelles innovations tant techniques que scientifiques et qui les rendent intéressantes pour l'exploitation structurée d'images de documents (centralisation de corpus, partage d'algorithmes). Il est important de signaler dès à présent que nous n'avons pas pour objectif de concurrencer ou de remplacer ces deux plateformes, mais de proposer une solution indépendante, compatibles avec les solutions existantes dont l'objectif principal est de disposer d'un ensemble de fonctionnalités dédiées à la création et au partage de vérités-terrains.

4.2.1.1 Présentation des plateformes logicielles distribuées dédiées à la recherche sur les images de documents les plus avancées.

La première a été utilisée dans le cadre du projet européen IMPACT [13] et se base sur les outils proposés par le projet *myGrid* [14]. La seconde, DAE [15] (Document Analysis and Exploitation) utilise aussi en partie

12. http ://www.digitisation.eu/
13. http://www.impact-project.eu/
14. http:/www.mygrid.org.uk/
15. http://dae.cse.lehigh.edu

les outils proposés par *myGrid*, mais propose sa propre solution quant au stockage et à la centralisation de vérités terrains. Pour finir, nous parlerons du projet DARE [LL11b] qui est à l'initiative des auteurs de DAE. Ce projet essaye d'imaginer les usages et fonctionnalités nécessaires d'une telle plateforme dans un futur proche.

Le projet IMPACT se focalise sur la mise à disposition d'implémentations d'algorithmes. L'idée du projet IMPACT est d'encapsuler un algorithme au sein d'un service web. Les services web permettent de simplifier l'utilisation de l'algorithme en enlevant certaines contraintes techniques. Chaque auteur d'algorithmes doit maintenir ses propres algorithmes et son propre serveur. Ici, tout algorithme de traitement est délocalisé et distribué. De plus, le projet IMPACT ne propose pas d'hébergement mutualisé d'algorithmes. Ainsi, chaque créateur d'algorithmes est responsable de la gestion du ou des serveurs permettant l'exécution d'algorithmes.

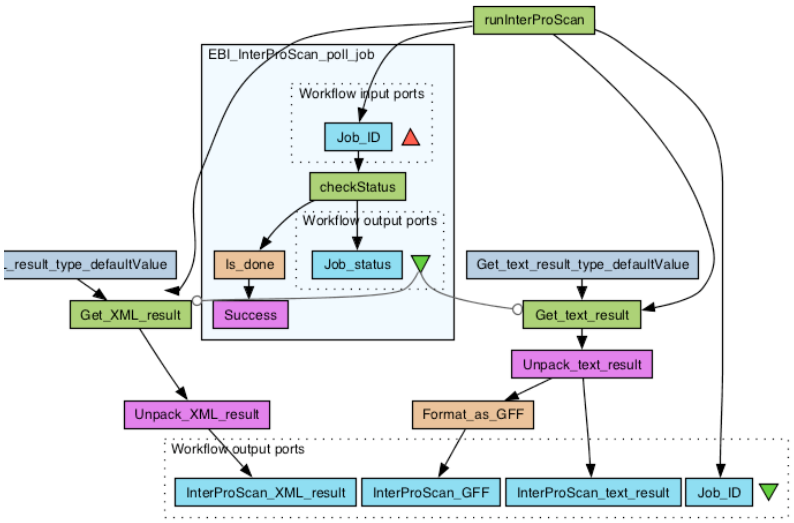


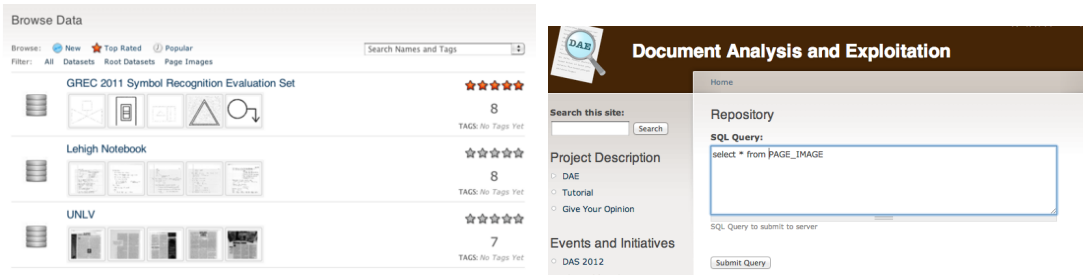
FIGURE 4.8 – *Une chaine de traitement dans le logiciel Taverna. Cet outil permet d'ordonnancer de façon simple, un ensemble de services webs pour construire des chaînes de traitements complexes.*

Pour cela les membres du projet IMPACT utilisent les outils proposés par *myGrid* [SRG03] qui est un projet de recherche européen dont l'objectif est d'utiliser les technologies issues du "clood computing" dans différents domaines de recherche scientifique. Ainsi, les algorithmes proposés dans le cadre d' IMPACT sont utilisés par des outils d'ordonnancement de services web comme Taverna (figure 4.8). Il devient alors possible et rapide de créer une chaine d'algorithmes répondant à une tâche précise ou même de comparer plusieurs algorithmes sur une liste d'images. Les membres du projet de recherche SCAPE [16] (dédié à la préservation du patrimoine culturel européen) ont déclaré [17] utiliser cet outil pour ordonnancer les tâches de préservation des bibliothèques numériques européennes.

Le projet DAE (Document Analysis and Exploitation) [LL10, KSH] est une plateforme web avec de nombreux objectifs. En 2008 [KSH], les auteurs décrivent les fonctionnalités dont devrait disposer la plateforme DAE, à savoir : être capable d'héberger des images de documents organisées par corpus ainsi que la vérité-terrain associée (structure physique et OCR, etc.) et la mesure des performances des algorithmes sur ces mêmes corpus. L'accès à ces informations se ferait en exploitant une base de données (à l'aide du langage SQL). En 2010 [LL10], une première version de la plateforme est rendue publique et accessible. Le projet semble alors se focaliser sur la création et la correction collaborative et semi-automatique de vérités terrain. DAE propose ainsi une interface web permettant d'annoter la structure

16. http ://www.scape-project.eu/
17. http ://www.taverna.org.uk/introduction/related-projects/scape/

physique d'un document. Il est aussi possible de rechercher des éléments particuliers en écrivant des requêtes SQL (voir figure 4.9). En 2011, le développement de la plateforme s'oriente vers la mise à disposition d'algorithmes sous forme de services web [LLS11, LL11a] à l'image du projet IMPACT. Ce choix de direction semble être confirmé dans l'article [LL12] en 2012.



(a) Exploration des différents corpus à l'aide d'un site web (b) Recherche par requêtes SQL

FIGURE 4.9 – *La plateforme DAE (Document Analysis and Exploitation)*

Conscient des limitations de la plateforme DAE du fait d'une mauvaise anticipation des possibilités technologiques, les auteurs se proposent d'imaginer les fonctionnalités proposées par une telle plateforme en 2021 [LL11b]. Pour cela, ils se basent sur l'histoire de Jane une jeune chercheuse en image de documents. Dans cette fiction, la notion de vérités terrains est remplacée par la notion d'interprétation. Une interprétation est une annotation concernant un élément ou plusieurs éléments d'une page et renseignée par un chercheur via une interface web. Toutes les interprétations sont indexées et versionnées. Chaque chercheur peut contribuer à l'ajout ou la modification d'une interprétation. De plus, l'évaluation de performances d'algorithmes est certifiée par un serveur qui s'occupe de l'expérimentation (choix du corpus et des paramètres). Le certificat numérique obtenu peut être alors utilisé lors de la publication de l'algorithme testé : ce dernier contient une URL (lien web) vers l'ensemble des images testées ainsi que leurs interprétations. Cela permet d'évaluer un autre algorithme avec le même protocole expérimental. Dans cet article, l'ensemble des idées présentées s'articule autour d'une plateforme centrale (appeler DARE pour Document Analysis Research Engine) [18] ainsi que la notion forte de "social computing".

En conclusion, on remarque que, contrairement à d'autres domaines de recherche où l'utilisation de plateformes logicielles distribuées est courante, les plateformes proposées dans le domaine de recherche sur les images de documents sont des projets très récents. L'utilisation de ce type d'outils dans notre communauté est pour l'instant faible, mais le nombre de publications scientifiques ou de tutoriaux de conférence portant sur ces plateformes témoigne de l'intérêt scientifique grandissant pour les fonctionnalités proposées par ces dernières.

4.2.1.2 Innovations à apporter à ces plateformes

Nous nous inscrivons dans cette tendance et sommes convaincus que l'utilisation de logiciels distribués peut aider à la réalisation de travaux de recherche sur les images de documents anciens. Nous avons identifié un ensemble de problèmes, qui ne sont pas clairement résolus par les plateformes existantes et qui sont, à notre avis, d'une grande importance pour notre activité.

Un accès décentralisé et contrôlé aux données

La centralisation sur un même serveur des corpus d'images de documents est un atout, mais présente aussi un frein pour les producteurs de corpus désireux de garder le contrôle sur la diffusion de leurs

18. http://dae.cse.lehigh.edu/WIKI/index.php/Dare_Paradigm

images (authentification, cryptage, statistiques sur les accès à la base). Cette centralisation implique aussi une dépendance forte vers une plateforme spécifique : en cas d'indisponibilité du site, l'ensemble des données (images et vérités-terrains) n'est plus disponible aux applications clientes. Afin que chacun puisse contrôler les données qu'il expose au reste de la communauté scientifique, et d'éviter les dépendances fortes vers des plateformes spécifiques, il est nécessaire d'utiliser plusieurs serveurs et d'abstraire la provenance des informations ainsi que le protocole de communication entre les clients et les serveurs.

Associer une interface de programmation applicative (API) à la plateforme.

Nous sommes convaincus que l'ensemble des informations hébergées par les différentes plateformes doit être accessible et modifiable de façon simple et riche. Les plateformes web doivent permettre aux chercheurs de pouvoir enrichir, par la création de logiciels clients, les fonctionnalités des plateformes. Par exemple, la plupart des logiciels de création de vérités terrains présentées dans la section précédente sont collaboratifs. Cet aspect collaboratif exige que ces applications puissent utiliser un ensemble de services (proposés par les plateformes distantes) permettant d'accéder et de manipuler les informations présentées aux utilisateurs. La manipulation et l'accès aux informations sont même, dans ce cas, réalisés en temps réel. L'utilisation de services web exposant les fonctionnalités d'algorithmes de recherches est un bon début, mais se doit d'être étendue à l'accès, la manipulation et la recherche avancée d'informations sur différents corpus d'images de documents.

Structuration de la vérité-terrain

Il existe un grand nombre de formats de fichiers utilisés pour représenter, soit le résultat de l'exécution d'un algorithme, soit d'une vérité-terrain (PAGE, GEDI, TruViz). Les plateformes web actuelles sont fortement liées à certains formats. Par conséquent, il n'est pas possible d'associer un évaluateur de performances, et des algorithmes à évaluer, si ces derniers ne s'accordent pas sur la mise en place d'un format de fichier spécifique (ou alors d'utiliser des services de conversions). Bien entendu, la seule solution pérenne est la standardisation de la vérité-terrain au sein d'un même format de fichier. Malheureusement, il n'est pas possible, pour l'instant, de standardiser la représentation de ces informations tant la recherche sur l'évaluation de performance est actives (le standard serait trop souvent soumis à des modifications). Les plateformes web peuvent proposer une solution intermédiaire, ou les informations seraient présentées aux clients dans le format désiré. Ce type de concept est utilisé depuis longtemps pour l'internationalisation de site web : l'information est la même, mais la langue change selon la localisation du client.

Couplage lâche des différentes applications et plateformes, transmission d'événements

Comme nous l'avons dit une plateforme logicielle distribuée est un ensemble complexe de logiciels communicants dédiés à un domaine métier particulier. Ces logiciels sont parfois interdépendants. Par exemple : un logiciel d'évaluation de performances à besoin d'un ensemble d'images annotées et donc d'un logiciel permettant de créer de la vérité-terrain ; un logiciel d'extraction de structure physique peut avoir besoin d'une version binarisée d'une image et donc d'un algorithme de binarisation. Pour l'instant ces dépendances ne peuvent être réalisées que par des liens forts entre les logiciels. PLus précisément, chaque logiciel est directement dépendant de la disponibilité d'un service donné à une adresse donnée. Ce type de couplage sur des architectures distribuées complexes est en constante évolution..

Reprenons l'exemple du logiciel d'évaluation de performances. Dans le modèle actuel, l'évaluateur doit être chainé après un logiciel de création de vérités-terrains. Or, la personne créant la vérité-terrain et celle exécutant l'évaluation de performances sont deux chercheurs différents. Ainsi, c'est la personne disposant du logiciel de création de vérité-terrain qui doit "connecter" son propre logiciel à un ou plusieurs évaluateurs de performances. Cette façon de faire est viable sur des cas simples et isolés, mais

devient très difficile sur des cas plus complexes où le nombre d'algorithmes et d'évaluateurs de performances évolue constamment.

Nous sommes convaincus qu'un couplage lâche entre les logiciels doit être favorisé. Ce couplage lâche peut être réalisé par la transmission d'événements. Dans cette architecture, les logiciels clients s'abonnent de façon transparente à des événements que les plateformes se chargent de transmettre. Ainsi, si l'on reprend l'exemple de l'évaluateur de performances, ce dernier peut s'abonner à un événement de type "nouvelle vérité-terrain créée". Le logiciel de création de vérité-terrain quant à lui peut publier la vérité-terrain créée sur la plateforme. Lors de cette publication, la plateforme va émettre l'événement "nouvelle vérité-terrain créée", à sa réception, l'évaluateur pourra commencer un nouveau cycle d'évaluation.

Garantir l'objectivité scientifique des résultats d'expérimentations.

Les plateformes actuelles proposent pour l'instant de versionner les corpus (images et vérité-terrain), afin de garantir la reproductibilité des expérimentations. Les problèmes liés aux modifications apportées aux corpus de documents sont donc bien pris en compte. Cependant, il est possible que des erreurs aient été réalisées lors de la création de la vérité-terrain. Les résultats d'expérimentations qui sont liées à une version du corpus ne sont donc plus objectifs et la correction de la vérité-terrain ne permet pas de corriger le problème. Nous pensons qu'il est préférable de procéder à des mises à jour automatiques des résultats d'expérimentations à chaque modification du corpus. Les corrections sont signalées automatiquement afin de reproduire les tests de performance le plus rapidement possible. Cette approche garantit, dans une certaine mesure, l'objectivité scientifique des résultats d'expérimentations d'un ensemble d'algorithmes, et ce malgré les modifications apportées aux corpus.

4.2.2 Proposition d'une plateforme logicielle distribuée complémentaire

L'ensemble des logiciels, de création de vérités-terrains, présentés en section 4.1 repose sur une plateforme logicielle proposant les fonctionnalités listées précédemment :

1. Un accès décentralisé et contrôlé aux données : les logiciels produisent la vérité-terrain d'images pouvant provenir de différents fournisseurs.
2. Associer une interface de programmation applicative (API) à la plateforme : les logiciels sont collaboratifs et doivent donc pouvoir accéder et modifier simplement des informations hébergées sur une plateforme.
3. Structuration de la vérité-terrain : chaque logiciel ne gère qu'un seul et unique format de fichier, mais reste néanmoins compatible avec d'autres via l'utilisation de la plateforme.
4. Couplage lâche des différentes applications et plateformes, transmission d'événements : ce couplage lâche nous permet de synchroniser nos applications afin que chaque utilisateur puisse visualiser les modifications réalisées par les autres en temps réels.

Bien entendu, les fonctionnalités présentées peuvent être conçues de manières différentes. Dans cette sous-section, nous présentons les concepts principaux de cette plateforme logicielle tout en apportant les fonctionnalités attendues. Nous ne traiterons pas ici des détails techniques d'implémentation. Deux modules logiciels transverses principaux ont été développés pour répondre à ces besoins :
– dépôts de données,
– notifications événementielles.
Ces concepts peuvent être utilisés pour la création d'un pont logiciel liant étroitement les données (par exemple provenant de la plateforme DAE), la vérité-terrain et les algorithmes (par exemple des web services IMPACT ou autre type de logiciels).

4.2.2.1 Les dépôts de données

Le concept de dépôt de données est un des axes principaux de la plateforme proposant entre autres les fonctionnalités suivantes :

1. Un accès décentralisé et contrôlé aux données
2. Structuration de la vérité-terrain

Un dépôt est capable d'héberger un ensemble d'informations liées à notre domaine métier (pages, ouvrages, paragraphes, etc.), mais aussi de proposer un ensemble de services effectuant diverses opérations sur les données hébergées (ajout, modification, suppression, recherche par critères, conversion de formats, etc.). En ce sens, un dépôt est une application basée sur une architecture orientée services (SOA) [FM07] où chaque service est un composant métier autosuffisant et respectant un contrat. Ce type d'architecture logiciel propose un couplage externe lâche avec les applications clientes (on utilise le plus souvent des services web). Ainsi, chaque application cliente intégrée à la plateforme peut interagir avec différents dépôts de façon transparente, pour peu que les services exposés respectent le même contrat. On pourra alors traiter des informations provenant de plusieurs sources sans se préoccuper de leurs provenances. Cela permet aussi de répartir différentes informations sur plusieurs dépôts. Par exemple, sur la figure 4.10 la vérité-terrain des images du dépôt 1 peut être hébergée sur le dépôt 3. Le dépôt 1 proposant un ou plusieurs services pour manipuler des images et le dépôt 3 proposant un ou plusieurs services pour manipuler de la vérité-terrain.

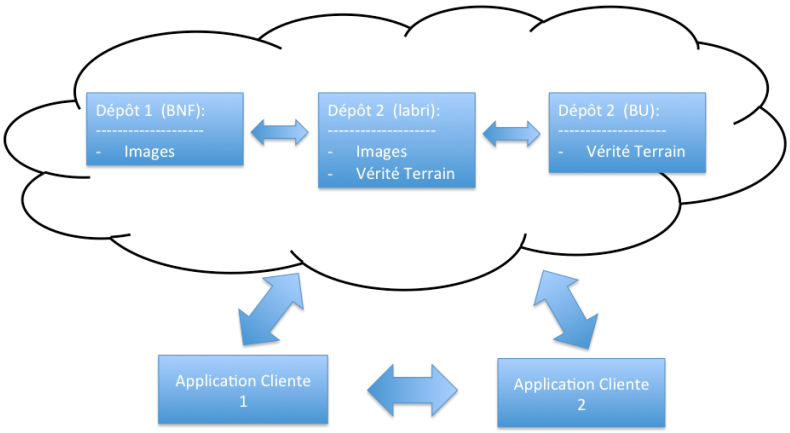


FIGURE 4.10 – *Le concept de dépôt permettant d'abstraire le format et la localisation de corpus d'images de documents.*

Le concept de dépôt est donc une solution technique à un grand nombre de problèmes scientifiques dont les principaux sont :

– **Partager des corpus de document de grande taille :** financièrement, le stockage d'images de documents est très couteux. Répartir ainsi les images et données associées sur différents serveurs et dans plusieurs laboratoires permet de répartir aussi les coûts liés à ce stockage.
– **Contrôler la diffusion des images :** certaines organisations ont besoin de contrôler la diffusion de leurs images. Par exemple, certaines images de certains organismes ne peuvent être accédées que par un certain nombre d'utilisateurs authentifiés. D'autres organismes peuvent avoir besoin de crypter les données envoyées sur le réseau. Là aussi la notion de dépôt permet aux applications clientes de pouvoir consommer ces données en ajoutant une couche d'abstraction sur ces problèmes d'authentification, de cryptage et décryptage des flux réseau. Les dépôts font aussi abstraction du protocole de communication. Cela permet par exemple de créer des dépôts qui ne seraient que de simples serveurs FTP ou SSH et où toutes données seraient enregistrées dans un fichier plat.

– **Gestion de plusieurs formats de vérité-terrain :** Chaque dépôt fournissant des services manipulant des vérités-terrains utilise une (ou plusieurs) représentation de ses informations par exemple, du XML, du JSON, du CSV (comma-separated values), un fichier PAGE ou encore une base de données. Cependant les services, exposent aux applications clientes, un ensemble de structures typées définies par contrats (contrats compatibles avec les informations du dépôt). Chaque type de service est capable de comprendre ou de générer un format de fichier spécifique, mais les informations (structures de données) présentées au consommateur du service sont toujours les mêmes (page, paragraphes, polygones, etc.). Le format réel de la vérité-terrain est donc abstrait et les applications clientes des dépôts ne manipulent pas des formats spécifiques.

4.2.2.2 Notifications événementielles : plateforme collaborative

Une notification événementielle est un message adressé, lors d'un événement, à un ensemble d'applications sur une architecture distribuée. Cette notion est construite sur la base d'une architecture orientée événements (EDA - Event Driven Architecture). Combiné avec la notion de dépôts, chaque service émet un événement s'il a réalisé une opération donnée. Ces opérations peuvent être de différentes natures, par exemple, l'ajout, la suppression, ou la modification, et concerner un type d'information particulier (ouvrage, page, paragraphe, vérité-terrain, etc.). Les notifications événementielles nous permettent deux choses : une synchronisation transparente des applications de création de vérités-terrains et de contrôler des applications dans un milieu hétérogène distribué.

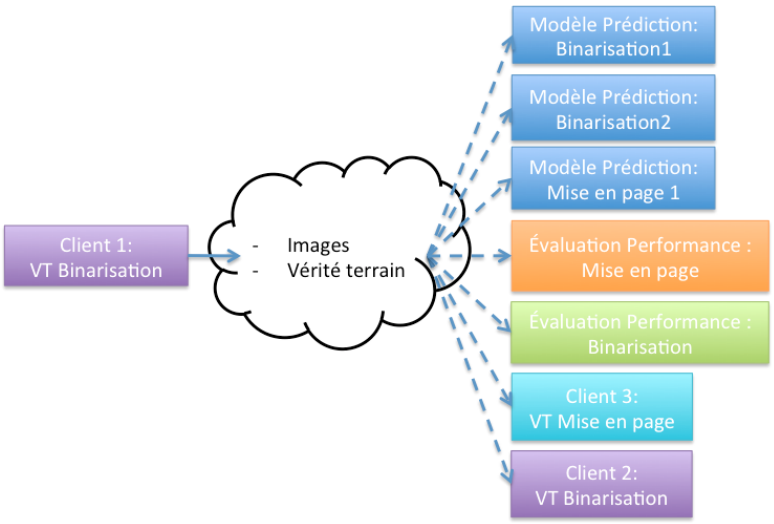


FIGURE 4.11 – *Exemples de notifications événementielles : les informations visualisées par certaines applications (Client 1 et 2) peuvent être synchronisées, d'autres types d'applications (modèles de prédiction, évaluateur de performance) peuvent être contrôlés par des événements.*

Synchronisation des données

L'automatisation de la synchronisation de données sur un ensemble d'applications clientes ne peut être réalisée en utilisant seulement la notion de dépôts (ces derniers permettent seulement de partager un ensemble de données entre les applications clientes). Cette synchronisation est pourtant essentielle pour les applications collaboratives. En effet, sans synchronisation, les informations créées ou modifiées sont détachées du serveur et sont enregistrées seulement lorsque le travail sur ces dernières est terminé. Plus ces informations sont détachées longtemps, plus il est susceptible qu'elles soient modifiées de façon concurrente, générant alors des erreurs de conflits. Comme la création d'une vérité-terrain

peut demander plusieurs dizaines de minutes, ou même plusieurs heures, les informations constituant la vérité-terrain sont détachées pendant un long moment pendant lequel les autres utilisateurs ne sont alors pas tenus informés des modifications réalisées. Par conséquent, il est possible qu'un autre utilisateur génère des conflits en modifiant parallèlement des informations détachées du serveur. Par exemple voici une série d'événements pouvant conduire à un conflit. Nous supposons deux utilisateurs u_1 et u_2, une annotation de vérité-terrain i et notons i_k les versions modifiées de i :

1. u_1 modifie i donnant i_1 ;
2. en parallèle, u_2 réalise une autre modification transformant i en i_2 ;
3. l'utilisateur u_1 décide alors de sauvegarder i_1 sur le serveur et ne rencontre pas de problème ;
4. c'est au moment ou u_2 sauvegarde i_2 que soit les informations de i_1 vont être écrasées soit une erreur de conflit sera générée.

La synchronisation des informations permet de limiter ces cas de figure étant donné que chaque modification va être mise à jour sur le serveur puis sur les clients pratiquement (modulo des contraintes réseau) en temps réel. Dans notre exemple précédent, l'information i est mise à jour au moment même ou u_1 la modifie, ainsi cette modification est transmise à u_2 qui modifie à son tour i_1 et l'enregistre. Pour se retrouver en conflit, il faudrait que deux utilisateurs modifient la même information exactement au même moment. La synchronisation automatique des données permet donc de disposer d'applications réellement collaboratives dans lesquelles les utilisateurs peuvent visualiser et modifier les mêmes informations en même temps.

Contrôler des applications distantes

La synchronisation d'applications clientes n'est pas le seul apport des notifications événementielles. En effet, ces dernières permettent aussi de contrôler des applications, et ce avec un couplage très faible. Chaque chercheur peut alors créer une application répondant aux notifications envoyées par un ou plusieurs dépôts. En fonction de la notification, l'application peut traiter l'information concernée de différentes façons. Prenons un exemple. Soit une application capable de binariser une image. Cette dernière peut s'abonner à la notification "nouvelles images à binariser". Ainsi, à chaque ajout d'une image et du déclenchement de l'événement "nouvelles images à binariser" l'application sera automatiquement informée et binarisera alors l'image. Le résultat de la binarisation peut être envoyé sur un dépôt. D'autres exemples et cas d'utilisation sont présentés dans la section suivante.

4.2.2.3 Perspective d'utilisations de la plateforme

L'utilisation de dépôts et de notifications permet comme nous l'avons présenté de réaliser des applications collaboratives permettant de créer de la vérité-terrain. Comme le montre la figure 4.11 il est aussi possible de répondre à certains besoins à l'aide des concepts introduits.

Utilisation de la plateforme pour l'évaluation de performance automatique

Il est possible de créer des programmes dédiés à l'évaluation de performances d'algorithmes, dont les résultats, seraient automatiquement mis à jour à chaque ajout de vérités-terrains correspondant aux algorithmes à évaluer. La communauté scientifique pourra alors disposer des performances moyennes d'algorithmes au fur et à mesure que les vérités terrains sont constituées.

Par exemple, supposons un logiciel de création de vérités-terrains pour les algorithmes de binarisation. Les images du corpus peuvent être hébergées sur le site web DAE. Les images correspondant à la vérité-terrain (image binarisée de référence) peuvent quant à elles être hébergées sur notre plateforme web. À chaque ajout d'une image binarisée au corpus, une notification sera envoyée aux logiciels d'évaluation de performance. Ces derniers peuvent alors récupérer l'ensemble des résultats calculés sur les images précédentes et y ajouter le score d'un ou plusieurs algorithmes de binarisation sur cette nouvelle image. Les résultats pourront ensuite être mis à jour sur la plateforme web.

Bien entendu, certains algorithmes peuvent être dédiés à un certain type d'images (documents modernes ou anciens, images binaires en niveaux de gris ou en couleurs, avec figures ou sans figures, etc.). Par conséquent, l'évaluation d'un algorithme sur des images qui ne lui correspondent pas n'a pas de sens. Pour éviter ce type de problèmes, il est bien entendu possible de spécifier le type d'images auquel les notifications doivent correspondre. Cela est possible par plusieurs moyens (nom du corpus, ouvrages, annotations dynamiques liées à la page, etc.).

Utilisation de la plateforme pour maintenir à jour des modèles de prédictions

La création d'un modèle de prédictions pour un algorithme donné nécessite un corpus d'images de documents associées à leurs vérités terrain. Les corpus actuels étant figés dans le temps, les modèles de prédiction liés à ces corpus sont pour l'instant statiques. En effet, le modèle de prédiction doit être mis à jour (réapprentissage), si de nouvelles images sont disponibles. Il est donc nécessaire de versionner ces modèles de prédictions afin que l'utilisateur puisse se mettre à jour.

Le système de notification événementielle permet de mettre à jour automatiquement ces modèles de prédictions à chaque ajout (ou modification) d'une image (ou d'informations) constituant une vérité-terrain exploitable par le modèle de prédiction. L'utilisateur du modèle dispose alors d'un modèle de prédiction dynamique qui reste en apprentissage permanent sans que l'utilisateur ait besoin d'intervenir.

Sélection automatique d'algorithmes en utilisant les fonctionnalités présentées

À l'heure actuelle, les dépendances algorithmiques d'un logiciel, processus ou algorithme de plus haut niveau, sont exprimées statiquement dans le code. Par exemple, OCROpus utilise deux méthodes de binarisation (Sauvola et Otsu) qui sont alternativement choisies selon les besoins. Ainsi, l'utilisateur ne peut pas dynamiquement choisir une méthode de binarisation plutôt qu'une autre. Le concept d'injection de dépendances est un mécanisme permettant de créer dynamiquement les dépendances entre les différents modules d'un même logiciel. Ce procédé basé généralement sur un fichier de configuration permet de déterminer dynamiquement les algorithmes à utiliser lors de l'exécution du logiciel.

Ce concept est particulièrement utile lorsqu'un logiciel repose sur un type d'algorithme pouvant être implémenté par un très grand nombre de méthodes. C'est le cas de la binarisation avec des méthodes telles que Sauvola, Otsu, Shijian, etc., et de façon générale avec la plupart des algorithmes de traitement d'images de documents (OCR, extraction de structure physique ou logique, etc.). De plus, avec les initiatives du projet européen IMPACT, un grand nombre d'algorithmes a été implémenté sous forme de services web. Il devient donc possible d'utiliser une méthode sans être limité par des contraintes techniques.

Il se pose alors le problème du choix de la bonne méthode, celle la plus adaptée à l'image courante. Dans cette thèse nous avons vu une stratégie permettant de répondre à ce problème : la prédiction de la méthode optimale pour une image donnée. L'utilisation de cette stratégie permet de spécifier à l'exécution du programme la méthode la plus adaptée. Comme nous l'avons présenté, il est possible *via* les concepts introduits, de disposer d'évaluateurs de performances et de modèles de prédictions qui seraient dynamiques. La stratégie d'injection de méthodes pouvant reposer sur les résultats (mis à jour automatiquement) de ces derniers, la sélection de méthodes devient réellement dynamique au fur et à mesure que la communauté scientifique propose, soit de nouvelles méthodes, soit de nouveaux corpus. Le logiciel ainsi construit n'est plus dépendant de méthodes spécifiques et est capable de se baser sur les performances proposées par l'état de l'art.

4.3 Conclusion

Dans ce chapitre nous nous sommes intéressés à la création, au partage et à l'utilisation de vérités terrains. Concernant la création, nous avons identifié trois types de vérités-terrains : les documents semi-synthétiques, l'annotation d'informations par un expert et la vérité-terrain perceptuelle. Pour chacun de ces niveaux, nous avons proposé des logiciels liés à notre besoin de création et partage de vérité-terrain relative à la qualité des images de documents :

– Création de documents semi-synthétiques à l'aide de fontes et de fonds extraits de documents réels et génération de transparence à l'aide d'une image "verso".
– Annotation de la qualité d'un document par un expert et permettant par exemple de recaler une paire d'images recto verso.
– Un logiciel permettant de "noter" la qualité perceptuelle de façon objective et rapide.

Une attention particulière a été portée sur les problèmes d'objectivité scientifique, de temps passé à la construction d'une vérité-terrain, de partage, et d'utilisation des vérités terrains. Ainsi, ces logiciels sont tous construits sur une plateforme logicielle qui les rend collaboratifs et évolutifs. Cette plateforme s'inscrit dans la continuité des efforts déjà proposés par les projets tels que IMPACT ou DAE, mais en proposant, en plus, un pont entre les vérités terrains (telles que celles proposées par DAE) et les algorithmes les utilisant (tels que ceux proposés dans le projet IMPACT). Ce pont logiciel permet entre autres de simplifier le partage et l'utilisation des vérités terrains en ajoutant une couche d'abstraction permettant de se détacher des formats spécifiques tels que PAGE ou GEDI.

L'utilisation, au sein d'une même plateforme, des trois différents niveaux de vérités terrain apporte un grand nombre de perspectives. En effet, il devient possible d'utiliser ces trois niveaux conjointement afin de, par exemple, prouver statistiquement le réalisme d'images synthétique, d'annoter de façon précise les dégradations générées par un modèle de dégradation, ou encore d'expliquer le comportement d'un lecteur et sa perception de la qualité en fonction des annotations apportées aux documents.

La valorisation et l'adoption d'une technologie par la communauté scientifique ou partie de cette communauté sortent du cadre de cette thèse. Nous nous sommes donc concentrés durant ces trois années à la réalisation de prototypes plus qu'à la réalisation de logiciels directement exploitables en production. Cependant, la plateforme construite est utilisée dans le cadre du projet ANR DIGIDOC où un ingénieur est responsable de la mise en production et de la maintenance de ces modules. Pour l'instant, le projet utilise un dépôt contentant 1653 pages (documents réels et semi-synthétiques) ainsi que 31 algorithmes utilisables en web services.

Conclusion

Ce manuscrit de thèse a permis tout d'abord de mettre en avant la difficulté des algorithmes de recherche à traiter des images de documents dégradés. Mis à part ces difficultés, nous avons montré que chaque algorithme est plus ou moins sensible à certains types de dégradation. Par exemple, certaines méthodes de binarisation sont plus sensibles aux bruits globaux et d'autres aux bruits locaux. Ainsi, la sélection de l'algorithme le plus adaptée à l'image et donc le plus robuste aux types de dégradation qui la compose est une étape indispensable qui a pour l'instant été rarement envisagée dans les chaînes de traitements. L'approche proposée dans cette thèse consiste à utiliser des descripteurs permettant de caractériser les dégradations d'une image de document et de les utiliser dans des modèles de prédiction des performances d'algorithmes de même type. Une fois les performances de chaque algorithme disponible prédites, il est alors possible de sélectionner celui qui proposera les meilleurs résultats pour une image donnée.

Afin de construire des descripteurs caractérisant les dégradations d'une image de document, nous avons proposé une méthodologie qui se divise en plusieurs étapes :
– la caractérisation de la dégradation en analysant des cas réels et l'influence de cette dégradation sur les performances d'un type d'algorithme,
– l'extraction des pixels de la dégradation,
– et pour finir, la définition de descripteurs qui se calculent sur les pixels extraits à l'étape précédente.
Cette méthodologie a été utilisée pour deux types de dégradation : les dégradations fond-encre et la transparence.

Nous avons proposé 18 descripteurs pour les perturbations fond-encre. Les descripteurs se composent de descripteurs colorimétries classiques (moyenne, variance, etc. sur l'histogramme des niveaux de gris), et des nouveaux descripteurs se basent leurs fondements sur les caractéristiques et les influences de ce type de dégradations en mesurant par exemple la localisation des dégradations vis-à-vis de l'encre.

Cette méthodologie a également été testée dans le cadre de la caractérisation de la transparence et de son influence sur les performances de l'OCR. Cet objectif a donné lien à la proposition d'une nouvelle méthode de recalage recto verso qui permet, par projection des pixels d'encre au verso sur le recto d'obtenir de façon précise les pixels de transparence. Cette méthode propose une précision similaire à la méthode couramment utilisée dans l'état de l'art, mais une complexité en temps réduite.

Afin de prédire les performances d'un algorithme en nous appuyant sur les descripteurs définis, nous avons utilisé un protocole strict basé sur l'apprentissage d'un régresseur permettant d'obtenir un modèle de prédiction. Ce protocole est ensuite utilisé pour construire des modèles de prédiction des performances des méthodes de binarisation les plus couramment utilisées en analyse et traitement d'images de documents. Chacun des modèles créés montre de très bonnes capacités prédictives sur notre corpus d'expérimentation. Afin de valider notre approche, nous avons utilisé conjointement l'ensemble des modèles prédictifs dans un processus de sélection automatique de la méthode de binarisation proposant les meilleurs résultats pour chaque nouvelle image. Ce système permet une augmentation significative des performances sur la binarisation. Nous avons aussi validé notre approche sur des systèmes complexes tels que les OCRs. Pour cela deux modèles de prédiction basés sur la mesure du défaut de transparence ont été créés et présentent tous deux de très bonnes capacités prédictives.

La très faible quantité de corpus d'images de documents anciens fortement dégradés auxquels est associée une vérité-terrain nous a permis de soulever tout un ensemble de problèmes liés à l'évaluation des

performances d'algorithmes. Nous avons proposé de créer une suite logicielle permettant de simplifier la création, la diffusion et l'utilisation de vérités-terrains. Cette suite logicielle se base sur une architecture distribuée permettant l'utilisation conjointe de plusieurs méthodes de création de vérité-terrain. En plus de la méthode classique où la vérité-terrain est créée par un expert sur des images réelles, l'originalité de nos travaux est de proposer des outils permettant de générer des documents synthétiques ou semi-synthétiques et d'acquérir des informations perceptuelles. Dans cette thèse, ces logiciels ont été utilisés pour créer la base de documents semi-synthétiques utilisée pour la création de modèles de prédiction des performances d'OCRs en fonction du défaut de transparence. Ces derniers sont aussi utilisés dans le cadre du projet DIGIDOC où plusieurs bases ont été créées (bruit local de type poivre et sel, déformation due à l'humidité ou à la profondeur de la reliure, etc.).

Cette thèse propose de prédire les performances d'algorithmes par une analyse de la qualité des images de documents. Cependant, la qualité du document n'est pas la seule caractéristique influant sur les résultats des algorithmes. En effet, la complexité d'analyse, par exemple la présence de figures, de tableaux, de formules mathématiques, ou de mise en page complexe à l'image des magazines modernes, est aussi une caractéristique ayant une forte influence sur les performances générales des algorithmes de traitements et d'analyse d'images de documents. Cette caractéristique doit être considérée lors de la création de modèles de prédictions d'algorithmes hauts niveaux, tels que les OCRs, en créant des descripteurs dédiés. La création de tels descripteurs et leurs utilisations dans notre méthodologie sont de réelles perspectives à court terme. Nous pensons que l'étude de la complexité d'analyse d'une image de document doit se faire par l'utilisation de descripteurs à plusieurs niveaux, chaque niveau se base sur les descripteurs de plus bas niveau pour ajouter des informations de plus en plus lié au document :

- Les descripteurs bas niveaux ne nécessitent pas de connaissance sur le document. C'est par exemple le cas des descripteurs géométriques tels que les moments de Zernike [KH90] les transformations de Fourrier-Mellin ou de façon plus générale la plupart des descripteurs utilisés en reconnaissance de symbole [LVSM02]. Ces descripteurs peuvent être utilisés pour caractériser les composantes connexes d'une page. Il peut aussi être intéressant d'étudier les méthodes de partitionnement utilisées pour la reconnaissance de caractère et d'utiliser des descripteurs caractérisant la pertinence des partitions comme [DB79, Dun73, Ran71, Jac01, FM83]. D'autres descripteurs comme les moments statistiques calculés sur les profils verticaux et horizontaux peuvent caractériser d'une certaine façon la mise en page d'un document (nombre de ligne ou de colonnes).
- Les descripteurs moyens-niveaux utilisent des connaissances sur la composition physique du document par exemple, la taille moyenne d'un caractère, l'intensité moyenne (local ou global) de l'encre, etc. Plusieurs descripteurs de ce type ont déjà été utilisés pour caractériser des images de documents binaires [BKN95, CHK99, CHKW97]. Ces descripteurs doivent être adapté aux images en niveaux de gris, et peuvent être complétés par des descripteurs résultants d'analyse plus complexes comme l'étude de la similarité structurelle de deux documents. Il est ici possible d'utiliser la théorie des graphes pour créer un tel descripteur [CDD07].
- Pour finir, les descripteurs hauts-niveaux pourraient être des résultats d'analyse de la structure logique d'un document. Ces descripteurs sont nécessaires pour caractériser les images de documents susceptibles de contenir des éléments caractéristiques d'une mise en page complexe comme des formules mathématiques, des figures, etc. La thèse de L. Robadey, [Rob01] présente de tels descripteurs hauts-niveaux dans l'objectif d'extraire la structure logique d'un document.

Une perspective à plus long terme est l'automatisation du contrôle qualité de la numérisation d'un ouvrage. La méthodologie présentée dans cette thèse pour la création de modèles de prédiction des performances d'algorithmes de traitements peut être un début d'approche pour réaliser cette tâche. Cependant de nombreux verrous scientifiques sont à considérer. En effet, la validation de la qualité d'une image numérisée fait appel à des notions perceptuelles : une image correctement numérisée peut en effet présenter de nombreuses dégradations provenant du document original qui ne sont pas considérées lors de ce contrôle, mais pourtant caractériser par des descripteurs. Cependant, ces notions perceptuelles peuvent être renseignées par l'opérateur lors d'une étape d'apprentissage. En effet, l'automatisation du contrôle qualité peut être réalisée à deux niveaux : pendant la numérisation ou après la phase de numérisation. Ces deux approches reposent sur des outils scientifiques communs comme l'apprentissage

supervisé, semi-supervisé ou non supervisé [Mit97], incrémental ou statique [BBL09]. Les descripteurs peuvent aussi être similaires comme l'histogramme des niveaux de gris, la densité des pixels noirs et blancs, le gradient, le contraste ou encore l'entropie. Cependant ces approches induisent des avantages et inconvénients différents qui sont à considérer dans la conception de tels systèmes :

– Automatiser le contrôle qualité pendant la phase de numérisation permet de détecter les images mal numérisées au plus vite, mais aussi d'utiliser les connaissances métier de l'opérateur pour guider l'algorithme d'évaluation de la qualité. Ainsi, les choix réalisés par l'opérateur peuvent être appris par l'algorithme. Ce type d'approche est choisi par le projet DIGIDOC dont l'un des objectifs est la réalisation d'un scanner cognitif capable d'adapter ses réglages en fonction du document à numériser et de la qualité de l'image numérisée attendue par l'opérateur. Par conséquent, l'approche choisie par ce projet aborde les problématiques d'apprentissage incrémental et semi-supervisé. Les descripteurs considérés pour le moment sont de bas-niveaux.
– Automatiser le contrôle qualité après la phase de numérisation permet d'utiliser des connaissances (descripteurs) calculées sur la totalité de l'ouvrage. En effet, comme la totalité des images est disponible, le contrôle qualité réalisé par un utilisateur peut être accéléré en proposant toutes les images de l'ouvrage similaires (en terme de qualité) à une image qui n'aurait pas été validée par l'utilisateur. Cette méthode se rapproche des techniques de CBIR (Content Based Image Retreival) [DJLW08] mais en utilisant des descripteurs caractérisant la qualité d'une image.

Les travaux de recherche présentés dans ce manuscrit soulèvent donc deux perceptives de recherche à plus ou moins long terme. La première est la caractérisation de la complexité d'analyse, pour un algorithme, d'une image de document. La deuxième est l'automatisation du contrôle qualité, soit pendant la phase de numérisation, soit après cette dernière.

Bibliographie

[ABE06] B. Allier, N. Bali, and H. Emptoz. Automatic accurate broken character restoration for patrimonial documents. *International Journal on Document Analysis and Recognition*, 8(4) :246–261, 2006.

[AH90] T. Akiyama and N. Hagita. Automated entry system for printed documents. *Pattern recognition*, 23(11) :1141–1154, 1990.

[AK04] A. Antonacopoulos and D. Karatzas. Document image analysis for world war ii personal records. In *International Workshop on Document Image Analysis for Libraries*, pages 336–341. IEEE, 2004.

[APSS03] V. Ablavsky, J. Pollak, M. Snorrason, and M.R. Stevens. Ocr accuracy prediction as a script identification problem. In *Proceedings of the 2003 Symposium on Document Image Understanding Technology*, 2003.

[AYS+00] S. Aksoy, M. Ye, M.L. Schauf, M. Song, Y. Wang, Robert M. Haralick, J.R. Parker, J. Pivovarov, D. Royko, C. Sun, and G. Farnebäck. Algorithm performance contest. *Pattern Recognition*, pages 48–70, 2000.

[B+06] C.M. Bishop et al. *Pattern recognition and machine learning*. Springer, 2006.

[Bai93] H.S. Baird. Calibration of document image defect models. In *Proc. of Second Annual Symposium on Document Analysis and Information Retrieval*, pages 1–16, 1993.

[Bai95] H.S. Baird. The skew angle of printed documents. In *Document image analysis*, pages 204–208. IEEE Computer Society Press, 1995.

[Bai07] H.S Baird. The state of the art of document image degradation modelling. *Digital Document Processing*, 2007.

[BBL09] K. Boukharouba, L. Bako, and S. Lecoeuche. Incremental and decremental multi-category classification by support vector machines. In *International Conference on Machine Learning and Applications*, pages 294–300. IEEE, 2009.

[BCA+11] R.D Bentley, A. Csillaghy, J. Aboudarham, C. Jacquey, MA Hapgood, K. Bocchialini, M. Messerotti, J. Brooke, P. Gallagher, P. Fox, et al. Helio : The heliophysics integrated observatory. *Advances in Space Research*, 47(12) :2235–2239, 2011.

[Ber86] J. Bernsen. Dynamic thresholding of gray level images. *ICPR : Proc. Intl. Conf. Patt. Recog*, pages 1251–1255, 1986.

[Bie09] A. Bien. *Real World Java EE Patterns Rethinking Best Practices*. 2009.

[BJF90] H.S. Baird, S.E. Jones, and S.J. Fortune. Image segmentation by shape-directed covers. In *International Conference on Pattern Recognition*, volume 1, pages 820–825. IEEE, 1990.

[BKN95] L.R. Blando, J. Kanai, and T.A. Nartker. Prediction of ocr accuracy using simple image features. In *International Conference on Document Analysis and Recognition*, volume 1, pages 319–322. IEEE, 1995.

[Bob01] M. Bober. Mpeg-7 visual shape descriptors. *Circuits and Systems for Video Technology, IEEE Transactions on*, 11(6) :716–719, 2001.

[Bre03] T.M. Breuel. High performance document layout analysis. In *Proceedings of the Symposium on Document Image Understanding Technology*, pages 209–218, 2003.

[Bre08] T.M. Breuel. The ocropus open source ocr system. In *Proceedings IS&T/SPIE 20th Annual Symposium*, volume 2008, 2008.

[Bro92] L.G. Brown. A survey of image registration techniques. *ACM computing surveys (CSUR)*, 24(4) :325–376, 1992.

[BS01] M.S. Brown and W.B. Seales. Document restoration using 3d shape : a general deskewing algorithm for arbitrarily warped documents. In *International Conference on Computer Vision*, volume 2, pages 367–374. IEEE, 2001.

[BS07] A. Broumandnia and J. Shanbehzadeh. Fast zernike wavelet moments for farsi character recognition. *Image and Vision Computing*, 25(5) :717–726, 2007.

[BSDLS11] E.H. Barney Smith, J. Darbon, and L. Likforman-Sulem. A mask-based enhancement method for historical documents. *Document Recognition and Retrieval XVIII,*, 2011.

[BVHBA09] A. Barker, J.I. Van Hemert, R.A. Baldock, and M.P. Atkinson. An e-infrastructure for collaborative research in human embryo development. *International Symposium on Cluster Computing and the Grid*, 2009.

[C+60] J. Cohen et al. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1) :37–46, 1960.

[C+78] R.T.W. Calvard et al. Picture thresholding using an iterative slection method. *IEEE Transactions on Systems Man and Cybernetics*, 8(Aug) :630–632, 1978.

[CDD07] Fanny Chevalier, Maylis Delest, and Jean-Philippe Domenger. A heuristic for the retrieval of objects in low resolution video. In *CBMI*, pages 144–151, 2007.

[CH94] S. Chen and R.M. Haralick. An automatic algorithm for text skew estimation in document images using recursive morphological transforms. In *International Conference on Image Processing*, volume 1, pages 139–143. IEEE, 1994.

[Che10] C. Chen. *Handbook of pattern recognition and computer vision*. World Scientific, 2010.

[CHK99] M. Cannon, J. Hochberg, and P. Kelly. Quality assessment and restoration of typewritten document images. *International Journal on Document Analysis and Recognition*, 2(2) :80–89, 1999.

[CHKW97] M. Cannon, J. Hochberg, P. Kelly, and J. White. An automated system for numerically rating document image quality. In *Proceedings of SDIUT*, pages 162–167, 1997.

[CHP95] S. Chen, R.M. Haralick, and I.T Phillips. Automatic text skew estimation in document images. In *International Conference on Document Analysis and Recognition*, volume 2, pages 1153–1156. IEEE, 1995.

[Cie01] L. Cieplinski. Mpeg-7 color descriptors and their applications. In *Computer Analysis of Images and Patterns*, pages 11–20. Springer, 2001.

[CKLY93] T. Cheng, J. Khan, H. Liu, and D.Y.Y Yun. A symbol recognition system. In *International Conference on Document Analysis and Recognition*, pages 918–921. IEEE, 1993.

[CM11] A. Cornuéjols and L. Miclet. *Apprentissage artificiel*. Eyrolles, 2011.

[Coh68] J. Cohen. Weighted kappa : Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological bulletin*, 70(4) :213, 1968.

[CPA11] C. Clausner, S. Pletschacher, and A. Antonacopoulos. Aletheia - an advanced document layout and text ground-truthing system for production environments. In *International Conference on document Analysis and Recognition*, pages 48–52, 2011.

[DB79] D.L. Davies and D.W. Bouldin. A cluster separation measure. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (2) :224–227, 1979.

[DD05] E. Dubois and P. Dano. Joint compression and restoration of documents with bleed-through. *IS&T Archiving*, pages 170–174, 2005.

[DGG+07] X. Dong, K.E. Gilbert, R. Guha, R. Heiland, J. Kim, M.E. Pierce, G.C. Fox, and D.J. Wild. Web service infrastructure for chemoinformatics. *Journal of chemical information and modeling*, 47(4) :1303–1307, 2007.

[DJLW08] R. Datta, D. Joshi, J. Li, and J.Z. Wang. Image retrieval : Ideas, influences, and trends of the new age. *ACM Computing Surveys (CSUR)*, 40(2) :5, 2008.

[DMK+01] Y. Deng, BS Manjunath, C. Kenney, M.S. Moore, and H. Shin. An efficient color representation for image retrieval. *Transactions on Image Processing*, 10(1) :140–147, 2001.

[DP91] D. Derrien-Peden. Frame-based system for macro-typographical structure analysis in scientific papers. In *Pro ceedings of International Conference on Document Analysis and Recognition*, pages 311–319, 1991.

[DP01] E. Dubois and A. Pathak. Reduction of bleed-through in scanned manuscript documents. In *Society for Imaging science and technology*, 2001.

[Dro03] M. Droettboom. Correcting broken characters in the recognition of historical printed documents. In *Proceedings of the 3rd ACM/IEEE-CS joint conference on Digital libraries*, pages 364–366. IEEE Computer Society, 2003.

[DSP66] N.R. Draper, H. Smith, and E. Pownell. *Applied regression analysis*, volume 3. Wiley New York, 1966.

[Dun73] J.C. Dunn. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. 1973.

[DVPK10] M Delalandre, E. Valveny, T Pridmore, and D. Karatzas. Generation of synthetic documents for performance evaluation of symbol recognition & spotting systems. *International Journal on Document Analysis and Recognition*, 13 :187–207, 2010.

[DWH12] T.M. Deserno, P. Welter, and A. Horsch. Towards a repository for standardized medical image and signal case data annotated with ground truth. *Journal of digital imaging*, pages 1–14, 2012.

[DZL10] D. Doermann, E. Zotkina, and H. Li. Gedi-a groundtruthing environment for document images. In *International Workshop on Document Analysis Systems (DAS 2010)*, 2010.

[Fis91] J.L. Fisher. Logical structure descriptions of segmented document images. *Proceedings of International Con ference on Document Analysis and Recognition*, pages 302–310, 1991.

[FK88] L.A. Fletcher and R. Kasturi. A robust algorithm for text string separation from mixed text/graphics images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 10(6) :910–918, 1988.

[FM83] E.B. Fowlkes and C.L. Mallows. A method for comparing two hierarchical clusterings. *Journal of the American statistical association*, 78(383) :553–569, 1983.

[FM07] X. Fournier-Morelf. *SOA, Le guide de l'architecte*. Dunod, 2007.

[Fra11] G. Franzini. Impact final conference – case study : Scanning parameters. http://impactocr.wordpress.com/2011/10/24/case-study-scanning-parameters/, October 2011.

[FS89] I. Fogel and D. Sagi. Gabor filters as texture discriminator. *Biological cybernetics*, 61(2) :103–113, 1989.

[FT01] M. Feldbach and K.D. Tonnies. Line detection and segmentation in historical church registers. In *International Conference on Document Analysis and Recognition*, pages 743–747. IEEE, 2001.

[GKN98] J. Gonzalez, J. Kanai, and T.A. Nartker. Prediction of ocr accuracy using a neural network. *Series in machine perception and artificial intelligence*, 29 :356–370, 1998.

[GNP09] B. Gatos, K. Ntirogiannis, and I. Pratikakis. Icdar 2009 document image binarization contest (dibco 2009). In *International Conference on Document Analysis and Recognition*, pages 1375–1382. IEEE, 2009.

[Gre90] B. Grenier. *Décision médicale : analyse et stratégie de la décision dans la practique médicale*. Masson, 1990.

[GRSR] A.R. Garcia, B. Rurangwa, N. Strokina, and P. Rybalov. Local binary patterns.

[HBAT07] P. Heroux, E. Barbu, S. Adam, and E. Trupin. Automatic ground-truth generation for document image analysis and understanding. In *International Conference on Document Analysis and Recognition*, pages 476 –480, 2007.

[HCW97] S. Harding, W. Croft, and C. Weir. Probabilistic retrieval of ocr degraded text using n-grams. *Research and advanced technology for digital libraries*, pages 345–359, 1997.

[HD03] J. He and A.C. Downton. User-assisted archive document image analysis for digital library construction. In *Seventh International Conference on Document Analysis and Recognition*, pages 498–502. IEEE, 2003.

[HFD90] S.C. Hinds, J.L. Fisher, and D.P. D'Amato. A document skew detection method using run-length encoding and the hough transform. In *International Conference on Pattern Recognition*, volume 1, pages 464–468. IEEE, 1990.

[HLK03] C. Ha Lee and T. Kanungo. The architecture of trueviz : A groundtruth/metadata editing and visualizing toolkit. *Pattern recognition*, 36(3) :811–825, 2003.

[HN04] H. Hse and A.R. Newton. Sketched symbol recognition using zernike moments. In *International Conference on Pattern Recognition*, volume 1, pages 367–370. IEEE, 2004.

[Hou62] P.V.C. Hough. Method and means for recognizing complex patterns, December 18 1962. US Patent 3,069,654.

[Hou83] H.S. Hou. *Digital document processing*. John Wiley & Sons, Inc., 1983.

[HS88] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, page 50. Manchester, UK, 1988.

[HTG08] Q. Huynh-Thu and M. Ghanbari. Scope of validity of psnr in image/video quality assessment. *Electronics letters*, 44(13) :800–801, 2008.

[HYR86] A. Hashizume, P.S. Yeh, and A. Rosenfeld. A method of detecting the orientation of aligned components. *Pattern Recognition Letters*, 4(2) :125–132, 1986.

[IA91] R. Ingold and D. Armangil. A top-down document analysis method for logical structure recognition. In *International Conference on Document Analysis and Recognition*, pages 41–49, 1991.

[Jac] JL Jacobi. Abbyy finereader 10 professional edition review (august 3, 2011).

[Jac01] P. Jaccard. *Etude comparative de la distribution florale dans une portion des Alpes et du Jura*. Impr. Corbaz, 1901.

[JBWK99] X. Jiang, H. Bunke, and D. Widmer-Kljajo. Skew detection of document images by focused nearest-neighbor clustering. In *International Conference on Document Analysis and Recognition*, pages 629–632. IEEE, 1999.

[JVD+10] N. Journet, A. Vialard, J.P. Domenger, et al. Analyse de fontes anciennes : de la génération de données synthétiques à la reconnaissance. *CIFED10*, 2010.

[JY98] A.K. Jain and B. Yu. Document representation and its application to page decomposition. *Transactions on Pattern Analysis and Machine Intelligence*, 20(3) :294–308, 1998.

[KAM12] N. Ragot K. Ait-Mohand, T. Paquet. Prédiction de la performance des systèmes d'ocr. *CIFED12*, 2012.

[KH90] A. Khotanzad and Y.H. Hong. Invariant image recognition by zernike moments. *Transactions on Pattern Analysis and Machine Intelligence*, 12(5) :489–497, 1990.

[KH99] T. Kanungo and R.M. Haralick. An automatic closed-loop methodology for generating character groundtruth for scanned documents. *Transactions on Pattern Analysis and Machine Intelligence*, 21(2) :179–183, 1999.

[KHB+00] T. Kanungo, R.M. Haralick, H.S. Baird, W. Stuezle, and D. Madigan. A statistical, nonparametric methodology for document degradation model validation. *Transactions on Pattern Analysis and Machine Intelligence*, 22(11) :1209–1223, 2000.

[KHP93] T. Kanungo, R.M. Haralick, and I. Phillips. Global and local document degradation models. In *International Conference on Image Processing*, pages 730–734. IEEE, 1993.

[KHP94] T. Kanungo, R.M. Haralick, and I. Phillips. Non-linear local and global document degradation models. *Journal of Imaging Systems and Technology*, 5(4) :220–30, 1994.

[KI85] J. Kittler and J. Illingworth. On threshold selection using clustering criteria. *Systems, Man and Cybernetics*, 15(5) :652–654, 1985.

[KI86] J. Kittler and J. Illingworth. Minimum error thresholding. *Pattern recognition*, 19(1) :41–47, 1986.

[KK02] D.W. Kim and T. Kanungo. Attributed point matching for automatic groundtruth generation. *International Journal on Document Analysis and Recognition*, 5 :47–66, 2002.

[KNSV93] M. Krishnamoorthy, G. Nagy, S. Seth, and M. Viswanathan. Syntactic segmentation and labeling of digitized pages from technical journals. *Transactions on Pattern Analysis and Machine Intelligence*, 15(7) :737–747, 1993.

[KSH] H.F. Korth, D. Song, and J. Heflin. Metadata for structured document datasets. In *Proceedings of the 8th IAPR International Workshop on Document Analysis Systems*, pages 547–550.

[KSI98] K. Kise, A. Sato, and M. Iwata. Segmentation of page images using the area voronoi diagram. *Computer Vision and Image Understanding*, 70(3) :370–382, 1998.

[KSW85] J.N. Kapur, P.K. Sahoo, and A.K.C. Wong. A new method for gray-level picture thresholding using the entropy of the histogram. *Computer vision, graphics, and image processing*, 29(3) :273–285, 1985.

[KVJ+12] V.C. Kieu, M. Visani, N. Journet, J.P. Domenger, and R. Mullot. A character degradation model for grayscale ancient document images. *ICPR : Proc. Intl. Conf. Patt. Recog*, 2012.

[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*, volume 1, pages 674–677. IEEE, 2001.

[LeB97] F. LeBourgeois. Robust multifont ocr system from gray level images. In *International Conference on Document Analysis and Recognition*, volume 1, pages 1–5. IEEE, 1997.

[LL93] C.H. Li and CK Lee. Minimum cross entropy thresholding. *Pattern Recognition*, 26(4) :617–625, 1993.

[LL10] B. Lamiroy and D. Lopresti. A platform for storing, visualizing, and interpreting collections of noisy documents. In *Proceedings of the fourth workshop on Analytics for noisy unstructured text data*, pages 11–18. ACM, 2010.

[LL11a] B. Lamiroy and D. Lopresti. An open architecture for end-to-end document analysis benchmarking. In *International Conference on Document Analysis and Recognition*, pages 42–47. IEEE, 2011.

[LL11b] D. Lopresti and B. Lamiroy. Document analysis research in the year 2021. *Modern Approaches in Applied Intelligence*, pages 264–274, 2011.

[LL12] B. Lamiroy and D. Lopresti. The non-geek's guide to the dae platform. In *International Workshop on Document Analysis Systems*, pages 27–32. IEEE, 2012.

[Llo85] DE Lloyd. Automatic target classification using moment invariant of image shapes. *IDN AW126, RAE, Farnborough, Reino Unido*, 1985.

[LLS11] B. Lamiroy, D. Lopresti, and T. Sun. Document analysis algorithm contributions in end-to-end applications : Report on the icdar 2011 contest. In *International Conference on Document Analysis and Recognition*, pages 1521–1525. IEEE, 2011.

[LMAB01] O. Lavialle, X. Molines, F. Angella, and P. Baylou. Active contours network to straighten distorted text lines. In *International Conference on Image Processing*, volume 3, pages 748–751. IEEE, 2001.

[Low99] D.G. Lowe. Object recognition from local scale-invariant features. In *International Conference on Computer Vision*, volume 2, pages 1150–1157. Ieee, 1999.

[Low04] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2) :91–110, 2004.

[LSF94] L. Likforman-Sulem and C. Faure. Extracting text lines in handwritten documents by perceptual grouping. *Advances in handwriting and drawing : a multidisciplinary approach*, pages 117–135, 1994.

[LSHF95] L. Likforman-Sulem, A. Hanimyan, and C. Faure. A hough based algorithm for extracting text lines in handwritten documents. In *International Conference on Document Analysis and Recognition*, volume 2, pages 774–777. IEEE, 1995.

[LSZT07] L. Likforman-Sulem, A. Zahour, and B. Taconet. Text line segmentation of historical documents : a survey. *International Journal on Document Analysis and Recognition*, 9(2) :123–138, 2007.

[LT98] CH Li and PKS Tam. An iterative algorithm for minimum cross entropy thresholding. *Pattern Recognition Letters*, 19(8) :771–776, 1998.

[LTW94] D.S. Le, G.R. Thoma, and H. Wechsler. Automated page orientation and skew angle detection for binary document images. *Pattern Recognition*, 27(10) :1325–1344, 1994.

[LVSM02] J. Lladós, E. Valveny, G. Sánchez, and E. Martí. Symbol recognition : Current advances and perspectives. *Graphics Recognition Algorithms and Applications*, pages 104–128, 2002.

[MB01] U.V. Marti and H. Bunke. On the influence of vocabulary size and language models in unconstrained handwritten text recognition. In *International Conference on Document Analysis and Recognition*, pages 260–265. IEEE, 2001.

[MB11] A. Moorthy and A. Bovik. Visual quality assessment algorithms : what does the future hold ? *Multimedia Tools and Applications*, 51 :675–696, 2011.

[MBE01] D.S. Messing, P. Beek, and J.H. Errico. The mpeg-7 colour structure descriptor : Image description using colour and local spatial information. In *International Conference on Image Processing*, volume 1, pages 670–673. IEEE, 2001.

[MC09a] R.F. Moghaddam and M. Cheriet. Low quality document image modeling and enhancement. *International journal on document analysis and recognition*, 11(4) :183–201, 2009.

[MC09b] R.F. Moghaddam and M. Cheriet. Rsldi : Restoration of single-sided low-quality document images. *Pattern Recognition*, 42(12) :3355–3364, 2009.

[MD09] S. Marchand and P. Desbarats. IBISA : Image-Based Identification / Search for Archaeology. volume VAST-STAR, Short and Project Proceedings, pages 57–60, 2009.

[Mit97] T.M. Mitchell. Machine learning. 1997. *Burr Ridge, IL : McGraw Hill*, 1997.

[MKP02] J.M. Martínez, R. Koenen, and F. Pereira. Mpeg-7 : the generic multimedia content description standard, part 1. *Multimedia, IEEE*, 9(2) :78–87, 2002.

[MMS10] S. Marinai, E. Marino, and G. Soda. Table of contents recognition for converting pdf documents in e-book formats. In *Proceedings of the 10th ACM symposium on Document engineering*, pages 73–76. ACM, 2010.

[MMS11] S. Marinai, E. Marino, and G. Soda. Conversion of pdf books in epub format. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pages 478–482. IEEE, 2011.

[MOVY01] B.S. Manjunath, J.R. Ohm, V.V. Vasudevan, and A. Yamada. Color and texture descriptors. *Circuits and Systems for Video Technology, IEEE Transactions on*, 11(6) :703–715, 2001.

[Moy06] L.A. Moyé. *Statistical reasoning in medicine : the intuitive P-value primer*. Springer, 2006.

[MRK03] S. Mao, A. Rosenfeld, and T. Kanungo. Document structure analysis algorithms : a literature survey. In *Proc. SPIE Electronic Imaging*, volume 5010, pages 197–207, 2003.

[MS99] R. Manmatha and N. Srimal. Scale space technique for word segmentation in handwritten documents. *Scale-Space Theories in Computer Vision*, pages 22–33, 1999.

[MS05] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(10) :1615–1630, 2005.

[MY99] H. Ma and Z. Yu. An enhanced skew angle estimation technique for binary document images. In *International Conference on Document Analysis and Recognition*, pages 165–168. IEEE, 1999.

[Nag10] G. Nagy. Document systems analysis : Testing, testing, testing. In *Proceedings of the Ninth IAPR International Workshop on Document Analysis Systems*, page 1, 2010.

[Nib85] W. Niblack. *An introduction to digital image processing*. Strandberg Publishing Company, 1985.

[Nie04] R. Niels. Dynamic time warping. *Artificial Intelligence*, 2004.

[NS95] D. Niyogi and S.N. Srihari. Knowledge-based derivation of document logical structure. In *International Conference on Document Analysis and Recognition*, volume 1, pages 472–475. IEEE, 1995.

[NS02] H. Nishida and T. Suzuki. Correcting show-through effects on document images by multiscale analysis. In *16th International Conference on Pattern Recognition*, volume 3, pages 65–68. IEEE, 2002.

[NSV92] G. Nagy, S. Seth, and M. Viswanathan. A prototype document image analysis system for technical journals. *Computer*, 25(7) :10–22, 1992.

[O'G93] L. O'Gorman. The document spectrum for page layout analysis. *Transactions on Pattern Analysis and Machine Intelligence*, 15(11) :1162–1173, 1993.

[Ots75] N. Otsu. A threshold selection method from gray-level histograms. *Automatica*, 11 :285–296, 1975.

[PAK10] A.P. Psyllos, C.N.E. Anagnostopoulos, and E. Kayafas. Vehicle logo recognition using a sift-based enhanced matching scheme. *Intelligent Transportation Systems, IEEE Transactions on*, 11(2) :322–328, 2010.

[Pal96] N.R. Pal. On minimum cross-entropy thresholding. *Pattern Recognition*, 29(4) :575–580, 1996.

[PC96] U. Pal and BB Chaudhuri. An improved document skew angle estimation technique. *Pattern Recognition Letters*, 17(8) :899–904, 1996.

[Pil01] M. Pilu. Deskewing perspectively distorted documents : An approach based on perceptual organization. page 100, 2001.

[PLZ+09] N. Ponomarenko, V. Lukin, A. Zelensky, K. Egiazarian, M. Carli, and F. Battisti. Tid2008 - a database for evaluation of full-reference visual quality assessment metrics. *Advances of Modern Radioelectronics*, 10 :30–45, 2009.

[PS99] Y. Pu and Z. Shi. A natural learning algorithm based on hough transform for text lines extraction in handwritten documents. *Advances in Handwriting Recognition by Seong-Whan Lee, World Scientific Publishing*, pages 141–152, 1999.

[PZ92] T. Pavlidis and J. Zhou. Page segmentation and classification. *CVGIP : Graphical models and image processing*, 54(6) :484–496, 1992.

[Ran71] W.M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336) :846–850, 1971.

[Ric93] S.V. Rice. Annual research report. Technical report, Information Science Research Institute, 1993.

[RJN96] S.V. Rice, F.R. Jenkins, and T.A. Nartker. *The fifth annual test of OCR accuracy*. Information Science Research Institute, 1996.

[RKK+01] Y.M. Ro, M. Kim, H.K. Kang, BS Manjunath, and J. Kim. Mpeg-7 homogeneous texture descriptor. *ETRI journal*, 23(2) :41–51, 2001.

[RKN93] S.V. Rice, J. Kanai, and T.A. Nartker. An evaluation of ocr accuracy. *Information Science Research Institute, 1993 Annual Research Report*, pages 9–20, 1993.

[RL83] A. Rosenfeld and D.E. LaTorre. Histogram concavity analysis as an aid in threshold selection(in image processing). *IEEE Transactions on Systems, Man, and Cybernetics*, 13 :231–235, 1983.

[RL94] P.G. Rootling and R.P. Loce. Digital halftoning. *Digital image processing methods*, 42 :363, 1994.

[RL09] M. Rusinol and J. Llados. Logo spotting by a bag-of-words approach for document categorization. In *International Conference on Document Analysis and Recognition*, pages 111–115. IEEE, 2009.

[Rob01] L. Robadey. *2 (CREM) : Une méthode de reconnaissance structurelle de documents complexes basée sur des patterns bidimensionnels*. PhD thesis, PhD thesis, Departement d'informatique de l'université de Fribourg, 2001.

[RYS95] N. Ramesh, J.H. Yoo, and IK Sethi. Thresholding based on histogram approximation. In *Vision, Image and Signal Processing, IEE Proceedings-*, volume 142, pages 271–279. IET, 1995.

[SB09] J. Silvestre-Blanes. Compress-image quality measures in image-processing applications. In *Emerging Technologies & Factory Automation, 2009. ETFA 2009. IEEE Conference on*, pages 1–4. IEEE, 2009.

[SCB87] M.W. Schwarz, W.B. Cowan, and J.C. Beatty. An experimental comparison of rgb, yiq, lab, hsv, and opponent color models. *ACM Transactions on Graphics (TOG)*, 6(2) :123–158, 1987.

[Sch94] R. Schaback. Reproduction of polynomials by radial basis functions. *Wavelets, Images, and Surface Fitting, P.J. Laurent, A. Le M ehaut e and LL Schumaker, AKPeters, Boston, 1994, 459*, 1994.

[Sch01] C. Schmid. Constructing models for content-based image retrieval. In *Conference on Computer Vision and Pattern Recognition*, volume 2, pages II–39. IEEE, 2001.

[SCNS03] A. Souza, M. Cheriet, S. Naoi, and C.Y. Suen. Automatic filter selection using image quality assessment. In *Seventh International Conference on Document Analysis and Recognition,*, pages 508–512. IEEE, 2003.

[Sez90] M.I. Sezan. A peak detection algorithm and its application to histogram-based image data reduction. *Computer vision, graphics, and image processing*, 49(1) :36–51, 1990.

[SG89] S.N. Srihari and V. Govindaraju. Analysis of textual images using the hough transform. *Machine Vision and Applications*, 2(3) :141–153, 1989.

[SG04a] Z. Shi and V. Govindaraju. Historical document image enhancement using background light intensity normalization. In *Pattern Recognition*, volume 1, pages 473–476. IEEE, 2004.

[SG04b] Z. Shi and V. Govindaraju. Line separation for complex document images using fuzzy runlength. In *International Workshop on Document Image Analysis for Libraries*, pages 306–312. IEEE, 2004.

[SGS93] V. Shapiro, G. Gluhchev, and V. Sgurev. Handwritten document image segmentation and analysis. *Pattern Recognition Letters*, 14(1) :71–78, 1993.

[Sha94] A.G. Shanbhag. Utilization of information measure as a means of image thresholding. *CVGIP : Graphical Models and Image Processing*, 56(5) :414–419, 1994.

[Sha01] G. Sharma. Show-through cancellation in scans of duplex printed documents. *Transactions on Image Processing*, 10(5) :736–754, 2001.

[SKK95] P. Stubberud, J. Kanai, and V. Kalluri. Adaptive image restoration of text images that contain touching or broken characters. In *International Conference on Document Analysis and Recognition*, volume 2, pages 778–781. IEEE, 1995.

[SKP08a] P. Stathis, E. Kavallieratou, and N. Papamarkos. An evaluation survey of binarization algorithms on historical documents. In *International Conference on Pattern Recognition*, pages 1–4. IEEE, 2008.

[SKP08b] P. Stathis, E. Kavallieratou, and N. Papamarkos. An evaluation technique for binarization algorithms. *Journal of Universal Computer Science*, 14(8) :3011–3030, 2008.

[SLS09] E. Saund, J. Lin, and P. Sarkar. Pixlabeler : User interface for pixel-level labeling of elements in document images. In *International Conference on Document Analysis and Recognition*, pages 646–650. IEEE, 2009.

[SLT11] B. Su, S. Lu, and C.L. Tan. Combination of document image binarization techniques. In *International Conference on Document Analysis and Recognition*, pages 22–26. IEEE, 2011.

[SM97] C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *Transactions on Pattern Analysis and Machine Intelligence*, 19(5) :530–535, 1997.

[SM01] R.W. Soukoreff and I.S. MacKenzie. Measuring errors in text entry tasks : An application of the levenshtein string distance statistic. In *CHI'01 extended abstracts on Human factors in computing systems*, pages 319–320. ACM, 2001.

[SP00] J. Sauvola and M. Pietikäinen. Adaptive document image binarization. *Pattern Recognition*, 33(2) :225–236, 2000.

[SRG03] R.D. Stevens, A.J. Robinson, and C.A. Goble. mygrid : personalised bioinformatics on the information grid. *Bioinformatics*, 19(suppl 1) :i302–i304, 2003.

[SSW88] P.K. Sahoo, S. Soltani, and AKC Wong. A survey of thresholding techniques. *Computer vision, graphics, and image processing*, 41(2) :233–260, 1988.

[SWS+00] A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *Transactions on Pattern Analysis and Machine Intelligence*, 22(12) :1349–1380, 2000.

[SWY97] P. Sahoo, C. Wilkins, and J. Yeager. Threshold selection using renyi's entropy. *Pattern recognition*, 30(1) :71–84, 1997.

[TA90] S. Tsujimoto and H. Asada. Understanding multi-articled documents. In *International Conference on Pattern Recognition*, volume 1, pages 551–556. IEEE, 1990.

[TBS09] A. Tonazzini, G. Bianco, and E. Salerno. Registration and enhancement of double-sided degraded manuscripts acquired in multispectral modality. In *International Conference on Document Analysis and Recognition*, pages 546–550. IEEE, 2009.

[TCC04] D.S. Turaga, Y. Chen, and J. Caviedes. No reference psnr estimation for compressed pictures. *Signal Processing : Image Communication*, 19(2) :173–184, 2004.

[TI94] Y. Tateisi and N. Itoh. Using stochastic syntactic analysis for extracting a logical structure from a document image. In *International Conference on Pattern Recognition*, volume 2, pages 391–394. IEEE, 1994.

[TJT96] O.D. Trier, A.K. Jain, and T. Taxt. Feature extraction methods for character recognition - a survey. *Pattern recognition*, 29(4) :641–662, 1996.

[TSB07] A. Tonazzini, E. Salerno, and L. Bedini. Fast correction of bleed-through distortion in grayscale documents by a blind source separation technique. *International Journal on Document Analysis and Recognition*, 10(1) :17–25, 2007.

[ULB04] A. Ulges, C.H. Lampert, and T. Breuel. Document capture using stereo vision. In *Proceedings of the 2004 ACM symposium on Document engineering*, pages 198–200. ACM, 2004.

[ULB05] A. Ulges, C.H. Lampert, and T.M. Breuel. Document image dewarping using robust estimation of curled text lines. In *Eighth International Conference on Document Analysis and Recognition*, pages 1001–1005. IEEE, 2005.

[Vap99] V. Vapnik. *The nature of statistical learning theory*. springer, 1999.

[VD+12] R. Vieux, J.P. Domenger, et al. Segmentation non supervisée d'images de document en paragraphes. In *Actes du Douzième Colloque International Francophone sur l'Écrit et le Document*, pages 415–430, 2012.

[VTRP08] E. Valveny, S. Tabbone, O. Ramos, and E. Philippot. Performance characterization of shape descriptors for symbol representation. *Graphics Recognition. Recent Advances and New Opportunities*, pages 278–287, 2008.

[WBSS04] Z. Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment : From error visibility to structural similarity. *Transactions on Image Processing*, 13(4) :600–612, 2004.

[WCW82] K.Y. Wong, R.G. Casey, and F.M. Wahl. Document analysis system. *IBM journal of research and development*, 26(6) :647–656, 1982.

[WPP02] C.S. Won, D.K. Park, and S.J. Park. Efficient use of mpeg-7 edge histogram descriptor. *Etri Journal*, 24(1) :23–30, 2002.

[WR83] J.M. White and G.D. Rohrer. Image thresholding for optical character recognition and other applications requiring character image extraction. *IBM Journal of Research and Development*, 27(4) :400–411, 1983.

[WRWC01] P. Wu, Y. Ro, C. Won, and Y. Choi. Texture descriptors in mpeg-7. In *Computer Analysis of Images and Patterns*, pages 21–28. Springer, 2001.

[WWC82] F.M. Wahl, K.Y. Wong, and R.G. Casey. Block segmentation and text extraction in mixed text/image documents. *Computer Graphics and Image Processing*, 20(4) :375–390, 1982.

[YATT91] A. Yamashita, T. Amano, I. Takahashi, and K. Toyokawa. A model based layout understanding method for the document recognition system. In *International Conference on Image Processing*, pages 130–138, 1991.

[YKKM04] A. Yamashita, A. Kawarago, T. Kaneko, and K.T. Miura. Shape reconstruction and image restoration for non-flat surfaces of documents with a stereo vision system. In *17th International Conference on Document Analysis and Recognition*, volume 1, pages 482–485. IEEE, 2004.

[YSS05] S. Yacoub, V. Saxena, and S.N. Sami. Perfectdoc : A ground truthing environment for complex documents. In *Eighth International Conference on Document Analysis and Recognition*, pages 452–456. IEEE, 2005.

[YV98] B.A. Yanikoglu and L. Vincent. Pink panther : a complete environment for ground-truthing and benchmarking document page segmentation. *Pattern Recognition*, 31(9) :1191–1204, 1998.

[ZD04] G. Zi and D. Doermann. Document image ground truth generation from electronic text. In *17th International Conference on Document Analysis and Recognition*, volume 2, pages 663–666. IEEE, 2004.

[Zi05] G. Zi. Groundtruth generation and document image degradation. Technical report, DTIC Document, 2005.

[ZT01] Z. Zhang and C.L. Tan. Restoration of images scanned from thick bound documents. In *International Conference on Image Processing*, volume 1, pages 1074–1077. IEEE, 2001.

[ZT02] Z. Zheng and C. L. Tan. Straightening warped text lines using polynomial regression. In *International Conference on Image Processing.*, volume 3, pages 977 – 980 vol.3, june 2002.

[ZTF04] Z. Zhang, C.L. Tan, and L. Fan. Restoration of curved document images through 3d shape modeling. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–10. IEEE, 2004.

[ZWDL03] Jian Zhai, Liu Wenyin, Dov Dori, and Qing Li. A line drawings degradation model for performance characterization. *International Conference on Document Analysis and Recognition*, 2 :1020–1024, 2003.