Contributions à la localisation adaptative de zones informatives sur des images de documents

Maroua Hammami

THÈSE

Pour obtenir le diplôme de doctorat

Spécialité Informatique

Préparée au sein de « l'université de Rouen Normandie »

Contributions à la localisation adaptative de zones informatives sur des images de documents

Présentée et soutenue par
Maroua HAMMAMI

Thèse soutenue publiquement le (25/11/2016) devant le jury composé de		
M. Josep LLADOS	Professeur, Université de Barcelone	Rapporteur
M. Jean-Yves RAMEL	Professeur, Université de Tours	Rapporteur
Mme. Caroline PETITJEAN	Maître de conférences, Université de Rouen Normandie	Examinatrice
M. Jean Christophe BURIE	Professeur, Université de La Rochelle	Examinateur
M. Vincent POULAIN D'ANDECY	Manager de Recherche, Itesoft YOOZ	Examinateur
M. Sébastien ADAM	Professeur, Université de Rouen Normandie	Directeur de thèse
M. Pierre HEROUX	Maître de conférences, Université de Rouen Normandie	Encadrant de thèse

Thèse dirigée par M. Sébastien ADAM, laboratoire LITIS

À la mémoire de
ma grand-mère Salha & ma tante Souad

"Live your life each day as you would climb a mountain.
An occasional glance toward the summit keeps the goal in mind,
but many beautiful scenes are to be observed from each new vantage point."

Remerciements

Un travail réussi n'est que le fruit de l'implication de toute une équipe. Chacun contribue à sa manière et à sa façon. Remercier ces personnes est la moindre chose que je peux leur offrir.

Tout d'abord, j'adresse mes remerciements à Monsieur Thierry PAQUET de m'avoir autorisée à effectuer ma thèse au sein du laboratoire d'Informatique, du Traitement de l'Information et des Systèmes. Je remercie Messieurs Josep LLADOS et Jean Yves RAMEL pour avoir accepté d'évaluer le manuscrit de ma thèse et je remercie les différents membres de jury, Madame Caroline PETITJEAN et Monsieur Jean-Christophe BURIE pour l'intérêt porté à mes travaux.

Je tiens à exprimer ma gratitude à mon directeur de thèse Sébastien ADAM et mon encadrant Pierre HÉROUX pour le temps qu'ils m'ont consacré, les conseils et le suivi attentif et continu de ma thèse. Je les remercie également pour leur patience et leurs encouragements durant ces années de préparation de ma thèse. Mes remerciements vont aussi à Messieurs Vincent Poulain D'ANDECY et Sadok KEBAIER pour leur collaboration et le suivi de ma thèse. Je voudrais aussi exprimer toute ma gratitude à Julien LEROUGE pour sa collaboration au développement de la solution et les suggestions et conseils durant ma thèse. J'espère qu'on aura l'occasion de travailler dans d'autres projets ensemble.

Je voudrais également remercier tout le personnel du laboratoire, et les membres de l'équipe DocApp "Document et Apprentissage" pour leur coopération et l'ambiance au sein du laboratoire. J'adresse des remerciements particuliers à Madame Fabienne BOQUET, Messieurs Fabrice HERTEL et Arnaud CITERIN qui étaient présents et rapides à répondre à tous mes besoins administratifs et techniques. Je remercie aussi mes collègues du LITIS en particulier Sovann, Wassim, Selma, Maroua et Gautier pour la bonne ambiance et les nombreux bons moments passés ensemble.

Enfin, j'aimerais remercier ma famille, mes amis et toutes les personnes qui m'ont aidée ou encouragée pendant ces années de thèse. Je remercie tout d'abord mes parents qui n'ont jamais cessé de m'encourager à poursuivre mes rêves et pour leurs aides morales et financières. Je remercie aussi mes frères Amine et Marouen pour leurs conseils et supports. Je remercie aussi Monsieur Jean et Madame Nicole de m'avoir accueillie au sein de leur famille. Je remercie aussi mes tantes Habiba et Faouzia ainsi que mes cousins et cousines Nada, Mariem, Rima et Raef pour les superbes moments passés ensemble et leurs encouragements. Je remercie mes amis pour leur patience et leur encouragement, Kenza, les amis du club alpin de Rouen en particulier l'équipe de choc Samuel et Frédo et les amis que j'ai rencontrés dans les sessions de course à pied en particulier Stéphanie et sa petite famille, Delphine et Pierre.

Résumé

Nous présentons dans cette thèse nos contributions pour la localisation automatique d'informations dans des images de documents en couleur. Notre objectif est de proposer une solution qui permet de localiser automatiquement une information, préalablement définie par un utilisateur sur un document modèle, dans un flux entrant de documents de la même catégorie. Les documents étant susceptibles de provenir de différentes sources de numérisation dont nous ignorons les caractéristiques, nous proposons de caractériser l'information cible par un positionnement relatif, invariant aux translations et aux changements d'échelle. Le modèle proposé dans ce cadre repose sur un graphe d'adjacence de régions qui décrit l'agencement d'ancres de repérage extraites du document. Le graphe obtenu permet à la fois de décrire l'information recherchée mais aussi la structure du document cible, indépendamment des coordonnées géométriques. Notre seconde contribution est une formulation linéaire en nombres entiers du problème de recherche d'isomorphisme de sous-graphes tolérant aux erreurs. La résolution de cette formulation permet d'extraire le sous-graphe correspondant à l'information recherchée dans le document cible. Enfin, notre troisième contribution concerne l'optimisation des paramètres d'une chaîne de traitement pour améliorer la performance de la localisation de l'information. L'approche proposée, qui repose sur un algorithme évolutionnaire, permet au système de s'adapter automatiquement à la classe de document traitée.

Mots clés : analyse de documents, segmentation d'images couleur, graphes, isomorphisme de sous-graphes, optimisation, algorithmes génétiques.

Abstract

Our contributions in this thesis are dealing with field spotting in colored document images. Our goal is to end up with a solution that automatically returns the right position of a region of interest (ROI), defined by a user on a reference document, into a flow of documents from the same class. As documents are provided from different and unknown sources, the position of the required information seems to be variable from an instance to another. Hence, absolute positions are considered as weak features to fill this kind of tasks. In this thesis, we propose a system that automatically localizes the information based on a relative position built with an adjacency graph.

Our solution is divided into 3 modules : the first one is dedicated to turning the image into an adjacency graph built with immutable informative zones. This structure, independent of coordinates, is used to describe the position of the ROI as well as the structure of the target document. Our second contribution is related to the subgraph isomorphism tolerant to topology distortions. Our goal is to operationnalize the structure representation proposed above in order to get the best matching between the graph describing the ROI and a subgraph from the target graph describing the document layout. In the last module, we focus on an optimization problem and we propose a solution leading to evolve performances of our system by optimizing the parameters of the process line. Our technique is based on an evolutionary system and is able to be automatically adapted to the processed document class.

Keywords : document Analysis, color image segmentation, graphs, subgraph isomorphism, optimization, genetic algorithms

Table des matières

Chapitre 1

Introduction générale

1.1 Lecture automatique de documents

Depuis une cinquantaine d'années, les évolutions scientifiques et technologiques ont permis le développement de nouveaux champs disciplinaires dont les applications n'ont cessé de se multiplier et de se démocratiser avec la révolution numérique. C'est notamment le cas dans les domaines scientifiques du traitement d'images numériques, de la reconnaissance de formes et de l'intelligence artificielle qui se sont nourris des évolutions technologiques en termes de dispositifs d'acquisition d'images (scanners, matrices CCD...), de stockage (mémoire), de transport (réseau), de traitement (ordinateurs, processeurs spécialisés, GPU...).

Parmi les domaines applicatifs fortement impactés par ces évolutions, celui de la dématérialisation de document a historiquement été l'un précurseur, avec les système de lecture d'adresses ou de chèques, par exemple. Le terme générique de dématérialisation recouvre plusieurs applications différentes. Chacune d'elles tente d'apporter une solution "numérique" au défi posé par le volume des informations disponibles au format papier. La réponse générique qu'apportent les applications repose dans un premier temps sur l'acquisition numérique de l'image (ou des images) de documents. Ensuite, les traitements qui sont appliqués aux documents numérisés diffèrent selon l'usage qui sera fait de ces documents, mais le but est toujours d'offrir à l'utilisateur un degré d'interaction avec le document au moins équivalent à celui qu'il aurait eu avec le papier, tout en relevant le défi de la masse des données. Il peut par exemple s'agir d'une "simple" numérisation visant la préservation et la diffusion de *fac-similé* numériques. Mais au delà, l'outil numérique permet la mise en œuvre de techniques d'indexation et de recherche dans les fonds documentaires, là où la navigation dans un fonds de documents physiques serait fastidieuse voire impossible. Ces techniques d'indexation par le contenu ont en particulier été rendues possibles par le développement des logiciels de reconnaissance optique de caractères.

Parmi l'ensemble des applications relatives à la dématérialisation des docu-

ments, dans le cadre de cette thèse, nous nous intéressons plus particulièrement à la lecture automatique de documents qui permet aux organisations d'alimenter leur système d'information avec des informations extraites automatiquement d'un flux de documents.

On peut définir la lecture automatique de document (LAD) comme le processus qui lit automatiquement sur des images de documents des informations spécifiées par un modèle de lecture. La notion de lecture intègre des opérations de localisation, d'extraction et de reconnaissance de l'information à lire. Le modèle de lecture recouvre un certain nombre de méta-données concernant les informations à lire.

Les méta-données du modèle de lecture peuvent être de différentes natures. Elles peuvent par exemple spécifier de manière absolue ou relative la position, la dimension de l'information à lire. Elles peuvent également indiquer la nature de cette information (textuelle, numérique, imprimée, manuscrite en capitale d'imprimerie ou cursive), le ou les formats sous lesquels l'information est susceptible d'apparaître. Il peut s'agir d'expressions régulières permettant de spécifier par exemple des dates, des numéros identifiants devant respecter une syntaxe particulière. Enfin, au plus haut niveau, les méta-données peuvent définir des règles de cohérence permettant de valider l'information lue. Il peut s'agir par exemple d'une règle qui indique qu'un champ décrivant un montant correspond bien au total de plusieurs autres champs, ou que le champ indiquant un patronyme figure bien dans une base de données.

1.2 Problématiques

La lecture automatique de documents est un processus qui s'applique le plus souvent sur des documents de type formulaire tels que ceux présentés sur la figure 1.1. Un formulaire est un document émis par une organisation pour recueillir des informations auprès de différents utilisateurs. Les emplacements sur lesquels il est attendu que les utilisateurs saisissent les informations sont spécifiés sur le fond imprimé du formulaire. Il peut s'agir de mots d'ap-

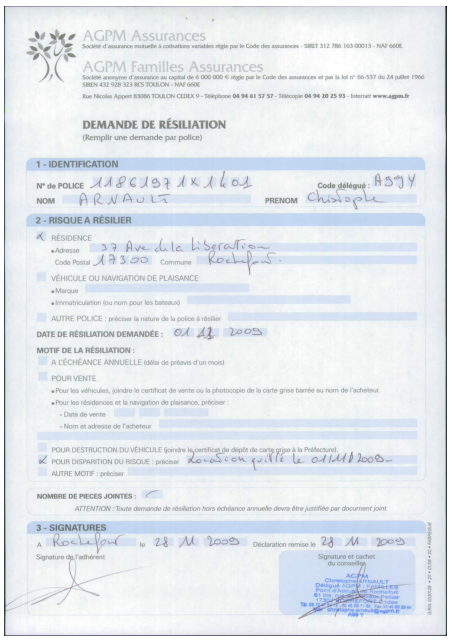
pel (≪ Nom ≫ , ≪ rue ≫, ≪ code postal ≫, ≪ numéro d'adhérent ≫, ≪ date ≫, ≪ montant ≫. . .) suivis de zones de saisie matérialisées le plus souvent par une ligne, des pointillés, des cases à remplir ou des peignes. Chaque utilisateur destinataire d'un formulaire le renseigne en inscrivant en surimpression du fond, et aux endroits spécifiés, les informations demandées. Après retour du formulaire auprès de l'organisation émettrice, la lecture automatique de document consiste alors à localiser, extraire et reconnaître automatiquement ces informations inscrites en surimpression sur l'image numérisée du formulaire rempli pour alimenter le système d'information.

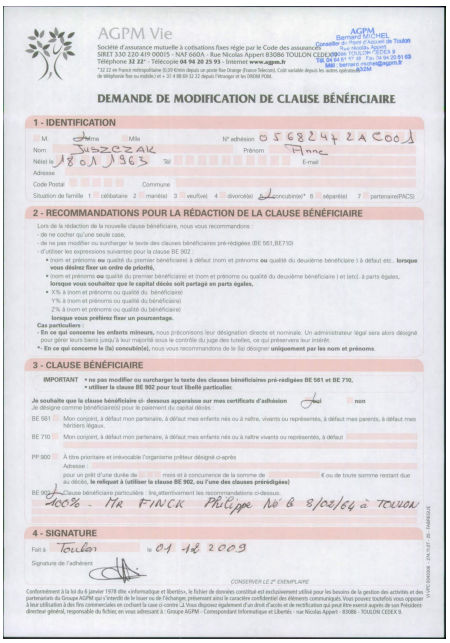
Les concepteurs de solutions de LAD sont confrontés à plusieurs défis. Les premières solutions de LAD ont été développées par le passé pour certains types de formulaires (par exemple la feuille de soin de la sécurité sociale) en raison du volume important de données à traiter régulièrement. Pour des raisons évidentes de rentabilité, les éditeurs de solutions de LAD cherchent dorénavant à proposer des systèmes génériques, capables de s'adapter à tout type de formulaires, déployables directement sur site et ne nécessitant pas de paramétrage impliquant des connaissances en traitement d'images. Dans le même ordre d'idées, la spécification du modèle de lecture associé à un type de formulaires doit être intuitive et ergonomique. De ce point de vue, les solutions de LAD déployables sur site doivent pouvoir offrir la possibilité de spécifier le modèle de lecture par l'exemple en indiquant, via une interface graphique, la localisation de l'information à lire sur une unique instance de formulaire de la classe.

Comme évoqué plus haut, les systèmes de LAD doivent être en mesure de traiter différents types de formulaires. Or, ces dernières années, les formulaires édités en couleur ont été de plus en plus nombreux. Les systèmes existants n'ont que très rarement exploité cette information colorimétrique, préférant ne traiter que la couche luminance (niveau de gris) pour simplifier et uniformiser la chaîne de traitement quel que soit le type de formulaire. Or, cette projection de l'espace colorimétrique dans le seul plan de luminance engendre une perte

(a) Exemple demande de résiliation

(b) Exemple demande la modification de la clause bénéficiaire





(c) Exemple de demande d'adhésion

FIGURE 1.1 – Exemples de documents administratifs et commerciaux en couleur

d'information qui peut être préjudiciable et rendre difficile la segmentation de deux objets qui étaient aisément discernables dans l'espace colorimétrique

initial.

Enfin, un des derniers défis technologiques auquel doivent faire face les concepteurs de solutions de LAD est celui de la variabilité des informations à lire. Cette variabilité qui peut intervenir à différents niveaux rend très difficile la définition d'un modèle de lecture applicable à l'ensemble des instances d'une même classe de formulaires. La variabilité peut tout d'abord être inhérente à l'information à saisir. On peut par exemple évoquer le cas où l'information à saisir est un montant comme ce pourrait être le cas sur une déclaration de revenus. En fonction de sa valeur, le nombre de chiffres nécessaires à l'expression de ce montant peut être très variable d'une instance à l'autre au sein de la même classe de document. Une autre source de variabilité qui peut être rencontrée tient dans le format utilisé. Le meilleur exemple est peut-être celui d'une date qui peut être exprimée selon le format DD/MM/YY, mais l'année peut également être saisie via quatre chiffres, et le mois être exprimé en toutes lettres... Une autre source de variabilité est celle inhérente à l'information manuscrite. Selon les scripteurs, la même information prendra des formes différentes et occupera plus ou moins d'espace. Par ailleurs, si le fond du formulaire propose un guide pour localiser ou circonscrire l'information ajoutée, l'expérience montre que ces guides ne sont pas scrupuleusement respectés et que le système de LAD doit pouvoir tolérer des écarts aux bornes fixées par le formulaire. Enfin, une autre source de variabilité tient aux conditions de numérisation. En effet, la numérisation de grands volumes de document qui s'opère à une cadence élevée peut engendrer des variations notamment en translation d'une instance à l'autre. L'ensemble de ces sources de variabilité illustre les limites d'un modèle de lecture au sein duquel les méta-données relatives au positionnement de l'information à lire seraient définies de façon absolue.

Finalement, on peut résumer les défis à relever de la façon suivante. Il s'agit de proposer une approche permettant de localiser une information à lire sur différentes instances d'une classe de formulaires. Cette approche doit être

robuste à différentes sources de variabilité et exploiter la représentation colorimétrique de l'image numérisée. Cette approche doit également être efficace quand bien même la désignation de l'information à extraire n'aura été faite que sur un exemple unique de la classe. L'approche doit pouvoir s'appliquer à différentes classes de formulaires sans requérir une expertise en traitement d'images, c'est-à-dire qu'il est nécessaire que son paramétrage soit automatisé.

L'ensemble des défis technologiques évoqués plus haut peuvent être exprimés en termes de problématiques scientifiques auxquelles nous proposons d'apporter des éléments de réponse dans le cadre de cette thèse. La première problématique concerne la localisation d'information sur une image de document robuste à la variabilité à partir d'une unique instance exemple en exploitant les données colorimétriques.

La seconde problématique concerne l'adaptation en ligne à une classe de formulaires du paramétrage d'une chaîne de traitement d'images au fil des instances rencontrées.

1.3 Proposition

L'approche que nous proposons pour adresser la première des deux problématiques repose sur les éléments suivants. Considérant les sources de variabilité rencontrées, les modèles de lecture basés sur un positionnement absolu de l'information à extraire montrent leurs limites. Il paraît alors plus opportun de spécifier les méta-données ayant trait au positionnement de façon relative. La question se pose alors du repère à utiliser pour définir ce positionnement relatif. Les documents de type formulaire tels que ceux présentés sur la figure 1.1 présentent deux couches d'information : celle correspondant aux informations saisies, dont certaines doivent être récupérées par le système de LAD, et la couche pré-imprimée correspondant au fond du formulaire. Cette dernière offre un repère naturel à partir duquel il est possible de situer les informations à extraire. En effet, si la numérisation du document engendre des transformations telles que la translation, la rotation ou un changement d'échelle (par

exemple dû à un changement de résolution dans l'acquisition), cette transformation linéaire est appliquée de façon similaire aux deux couches d'information. Ainsi, le positionnement de l'information à extraire reste inchangé dans le repère lié au fond du document. Pour pallier cette difficulté, nous proposons de ne pas situer l'information à extraire uniquement dans un système de coordonnées à deux dimensions, mais de multiplier les éléments de référence du positionnement. Ainsi, nous proposons de définir la position d'une information à extraire relativement à la structure physique du fond du formulaire. Cette structure physique est modélisée par une représentation structurelle invariante aux transformations susceptibles d'intervenir lors de la numérisation. Les formulaires couleurs édités de nos jours ont la caractéristique de présenter des rectangles ou des cadres qui modélisent des en-têtes ou des zones de saisie d'information. Ainsi, nous proposons une modélisation de la structure physique du fond du formulaire par le biais d'une représentation structurelle basée sur la localisation des rectangles de couleur homogène : un graphe d'adjacence de régions.

Sur cette base, il est possible de modéliser l'environnement d'une zone d'information relativement à la représentation structurelle du formulaire. Nous appelons cette modélisation de l'environnement, contexte structurel. Ainsi, le contexte structurel d'une zone d'information est le sous-graphe, restreint à un voisinage, de la représentation structurelle globale du document. La localisation d'une zone d'information similaire au sein d'une autre occurrence de formulaire peut alors se ramener à rechercher un contexte structurel identique au sein de la représentation structurelle de cette nouvelle instance, soit une recherche d'isomorphisme de sous-graphe. Il s'agit de retrouver, au sein de la représentation structurelle du document cible, une occurrence du contexte structurel de la zone d'information recherchée. Les chapitres 2 et 3 de ce manuscrit décrivent respectivement le processus d'extraction de la représentation structurelle modélisant la structure physique du fond du formulaire et une procédure originale de recherche d'isomorphisme robuste aux bruits inhérents

aux méthodes de traitement d'images et de reconnaissance de formes mises en œuvre.

Nous présentons dans un premier temps la chaîne générique de traitements fournissant une représentation structurelle. Différents paramétrages de cette chaîne générique de traitement peuvent être adoptés. Chacun de ces paramétrages, pour un même document, fournit une représentation structurelle différente. Nous montrons que différents paramétrages peuvent ainsi conduire à des performances différentes du point de vue de la localisation des zones d'information. L'objectif est alors d'identifier les paramétrages conduisant aux meilleures performances tout en remarquant que les paramétrages optimaux sont différents d'une classe de formulaires à l'autre.

Nous proposons alors une méthode d'optimisation qui vise à déterminer, pour une classe donnée, le paramétrage qui conduira aux meilleures performances du point de vue de la localisation des zones. Cette méthode d'optimisation tente de répondre à deux difficultés. D'une part, nous tentons d'optimiser la performance en localisation alors même qu'elle ne peut être mesurée en l'absence de vérité terrain. D'autre part, nous cherchons à optimiser la performance sur la globalité d'une classe alors que nous n'en disposons que d'un échantillon très restreint. L'approche que nous proposons repose sur un algorithme génétique. Les individus, solutions potentielles, encodent chancun un paramétrage différent de la chaîne de traitements. La fonction qui mesure l'adaptation d'un individu doit être liée à la performance en localisation du paramétrage correspondant tout en sachant que cette dernière ne peut être mesurée. Dans notre proposition, la fonction d'évaluation d'un individu, et donc du paramétrage correspondant de la chaîne de traitements, est une mesure de la stabilité des représentations structurelles issues de ce paramétrage pour la classe de document considérée. Cette proposition repose sur l'hypothèse que la recherche d'un contexte structurel de la zone recherchée au sein d'une autre occurrence de représentation structurelle présente davantage de chance d'aboutir si les représentations structurelles sont elles-mêmes identiques entre elles. La

mesure de la stabilité calculée sur un échantillon de représentations structurelles est elle-même un estimateur de la mesure de stabilité sur la globalité de la classe. La qualité de cet estimateur s'améliore avec la taille de l'échantillon, c'est-à-dire en prenant en considération les documents rencontrés au fil de la vie du système.

1.4 Plan de lecture

La suite de ce manuscrit est structurée en trois chapitres qui, chacun, détaillent la position du problème abordé, dressent un état de l'art relatif à ces problématiques, présentent nos contributions en la matière et décrivent leur validation expérimentale.

Le chapitre 2 est consacré à la problématique de représentation structurelle des images de formulaires en couleur. Après une revue de l'état de l'art, il détaille notre proposition qui permet de modéliser la tâche de construction de la représentation structurelle d'un formulaire basée sur les rectangles de couleur homogène comme une chaîne générique de traitements. L'instanciation de cette chaîne générique de traitement correspond à un paramétrage particulier. Ce paramétrage définit par exemple si un traitement de la chaîne est déclenché ou non, le choix des espaces colorimétriques au sein desquels s'opèrent les traitements, ou encore des valeurs de paramètres pour les traitements qui composent cette chaîne.

Le chapitre 3 présente la méthode mise en œuvre pour localiser, au sein de la représentation structurelle du document cible, le sous-graphe modélisant le contexte structurel de la zone d'information recherchée. La méthode proposée se veut robuste aux différences pouvant exister entre le sous-graphe recherché et son occurrence au sein de la représentation cible. Elle se base sur une recherche de sous-graphe à coût d'édition minimal modélisée comme une programme linéaire en nombres binaires.

Le chapitre 4 adresse le problème de l'adaptation du paramétrage de la chaîne de traitements à chacune des classes de formulaires tout en ne disposant

que de très peu d'exemples pour chacune d'elles. Nous tentons d'apporter une première réponse à cette problématique en proposant une approche basée sur un algorithme génétique qui vise à optimiser la performance du système de localisation des zones d'information.

Enfin, le chapitre 5 établit une conclusion générale de nos travaux et propose des perspectives d'amélioration.

Chapitre 2

De l'image vers une représentation structurelle

Sommaire

2.1 Introduction

Dans l'introduction générale de ce mémoire, nous avons défini les problématiques scientifiques abordées dans la thèse et nous avons donné un aperçu global du système de lecture automatique de documents qui a été développé pour répondre à ces problématiques. Nous avons en particulier mis l'accent sur les contraintes particulières auxquelles est soumis le système, dont la nécessité de pouvoir extraire des informations cibles dans des images de document obtenues des conditions d'acquisition variables.

Pour pallier cette contrainte, nous proposons dans cette thèse que l'information recherchée soit positionnée de façon relative par rapport à des éléments immuables du document. La description de l'agencement de ces éléments immuables constitue alors un modèle de la structure physique du document. Au vu des contraintes énoncées précédemment, ce modèle doit être invariant aux translations, aux changements d'échelle et à des variations colorimétriques.

Dans ce chapitre, nous présentons le modèle de structure physique utilisé dans le cadre de nos travaux. Il repose sur une représentation sous forme de graphe d'adjacence de régions, qui contient des informations invariables et pertinentes positionnées de façon relative les unes par rapport aux autres. Ainsi, nous garantissons que nous pouvons décrire (en phase d'apprentissage) puis localiser (en phase de production) une information requête, même si ses coordonnées dans l'image numérisée sont variables.

Pour construire ce modèle, la transformation de l'image de document en un graphe consiste d'abord à extraire des régions particulières du document qui vont correspondre aux nœuds du graphe. Puis, ces nœuds sont reliés par des arcs exprimant une relation de visibilité afin de décrire leur agencement dans le document, sans utiliser leurs positions absolues. Les attributs affectés aux nœuds et aux arcs sont normalisés, ce qui permet de décrire la structure du document sous forme de graphe attribué d'adjacence de régions, de façon invariante aux translations et aux changements d'échelle.

Dans le cadre de cette thèse, les images que nous traitons représentent des

documents à caractère administratif ou commercial qui ont généralement une structure particulière permettant de faciliter leur lecture. Dans ce genre de formulaires, deux types d'information importantes sont présents pour guider le lecteur : une partie textuelle, composée de texte imprimé et une partie graphique, généralement composée de zones de couleur homogène et de forme rectangulaire (voir figure 1.1).

Dans la phase d'analyse d'une image de document, les informations textuelles doivent être obtenues par un processus de reconnaissance, généralement coûteux et susceptible de produire des erreurs, surtout lorsque le lexique n'est pas maîtrisé comme c'est le cas lorsque des noms propres apparaissent sur les documents. C'est pourquoi nous avons plutôt choisi dans cette thèse de baser notre modèle sur les régions rectangulaires de couleur homogène.

Notre modèle de structure physique du document utilise donc les rectangles de couleur homogène comme ancre pour positionner les informations et ces ancres sont les nœuds de notre graphe. Pour construire les arcs du graphe représentant les documents, nous avons opté pour la relation de visibilité qui est rarement utilisée dans la littérature.

La suite de ce chapitre est structurée de la façon suivante : tout d'abord, nous présentons un état de l'art concernant les représentations structurelles de document. Ensuite, nous détaillons notre approche d'extraction des régions rectangulaires de couleur homogène et présentons notre proposition pour la construction de la connexion entre ces régions. Puis, nous présentons de premières expérimentations et de premiers résultats relatifs à l'extraction des régions rectangulaires. La représentation complète sous forme de graphe sera quant à elle évaluée dans le chapitre suivant, après la présentation de l'approche utilisée pour apparier les graphes.

2.2 État de l'art

Dans cette section, nous présentons quelques approches de la littérature qui abordent une problématique similaire à celle traitée dans ce chapitre et qui

vise donc l'extraction d'une structure de type graphe pour décrire le contenu d'une image de document.

Les graphes ont été très souvent utilisés dans la littérature pour modéliser le contenu de documents et pour manipuler leurs éléments. C'est particulièrement le cas pour les documents contenant des parties graphiques, dans lesquels l'agencement des éléments est propice à une description sous forme de graphe. Sans viser l'exhaustivité, on pourra citer des travaux sur l'analyse de plans architecturaux (Lladós et al., 2001, Locteau et al., 2007, Barbu et al., 2006, Dutta et al., 2013a), sur l'analyse de schémas électriques (Qureshi et al., 2007), ou l'analyse de bandes dessinées (Le et al., 2015, Ho et al., 2013). On relève également des travaux exploitant des représentations à base de graphes pour d'autres catégories de documents comme les documents patrimoniaux (Jouili et al., 2010, Mehri et al., 2015, Garz et al., 2016), les documents manuscrits (Fischer et al., 2013) ou encore les documents contenant des formules mathématiques (Lemaitre et al., 2005).

Devant cette importante quantité de travaux existants, nous avons choisi de nous focaliser dans cette section sur les approches qui traitent des documents dont la structure est proche de celle des documents que nous traitons dans cette thèse. Pour cette revue de l'existant, nous distinguons deux familles de méthodes de construction de la structure physique d'un document de type formulaire : celles qui n'utilisent pas de reconnaissance du texte et celles qui s'appuient sur l'utilisation d'un système d'OCR, dans le but de chercher des mots-clés.

2.2.1 Méthodes pré-OCR

Dans (Peanho et al., 2012), les auteurs présentent une approche d'extraction d'information depuis des images de facture dans le but de reconstruire le contenu sémantique du document. La méthode utilisée dans ces travaux repose sur la transformation de chaque document d'une même classe en une représentation structurelle de type graphe relationnel attribué. Pour plus de

détails sur les graphes relationnels attribué, nous référons le lecteur à (Cesarini et al., 1998). Partant de l'hypothèse que les documents à traiter proviennent de la même classe, un document est sélectionné pour représenter le modèle de cette dernière et définir manuellement une interprétation des différents éléments constituant le graphe qui le représente. Les informations des autres instances de documents peuvent ainsi être interprétées à l'aide d'un appariement entre la structure modèle et celles qui représentent les documents cibles.

L'approche proposée repose sur la construction de graphes dans lesquels les nœuds correspondent à des régions de texte et les arcs représentent les liens entre les différents attributs.

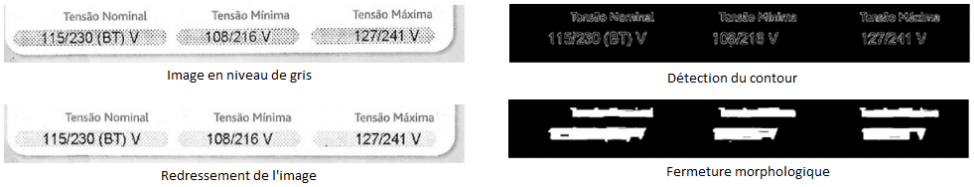


FIGURE 2.1 – Segmentation de régions de texte (Peanho et al., 2012)

L'extraction des régions textuelles se base sur la détection des contours. Comme l'illustre la figure 2.1, le traitement de la facture convertie en niveaux de gris commence par un redressement de l'image s'appuyant sur la transformée de Hough. Puis, les contours sont extraits, le bruit est supprimé et une fermeture morphologique est appliquée pour détecter les caractères appartenant au même mot. Cette méthode permet de trouver les informations textuelles dans l'image de document.

La deuxième étape de l'approche consiste en la construction du graphe relationnel attribué. L'objectif est d'obtenir une représentation qui relie les champs à leurs valeurs ou un titre aux attributs qui lui sont liés. L'approche proposée permet l'intervention de l'utilisateur pour corriger des erreurs de détection ou de construction de graphe. La dernière étape de l'approche effectue l'appariement entre le graphe modèle, dont l'interprétation sémantique

est définie manuellement, et le graphe cible.

L'utilisation de cette méthode nécessite une connaissance *a priori* de la classe du document et une intervention de l'utilisateur pour définir les différents champs. De plus, la conversion de l'image couleur en niveaux de gris peut engendrer une perte d'information textuelle importante, en particulier pour les images à basse résolution et présentant de fortes distorsions dues à leur impression et leur numérisation.

Dans l'article (Carton et al., 2015), les auteurs présentent une méthode d'extraction de connaissances et de génération de règles d'inférence sur des images de documents, en particulier des registres de mariage anciens.

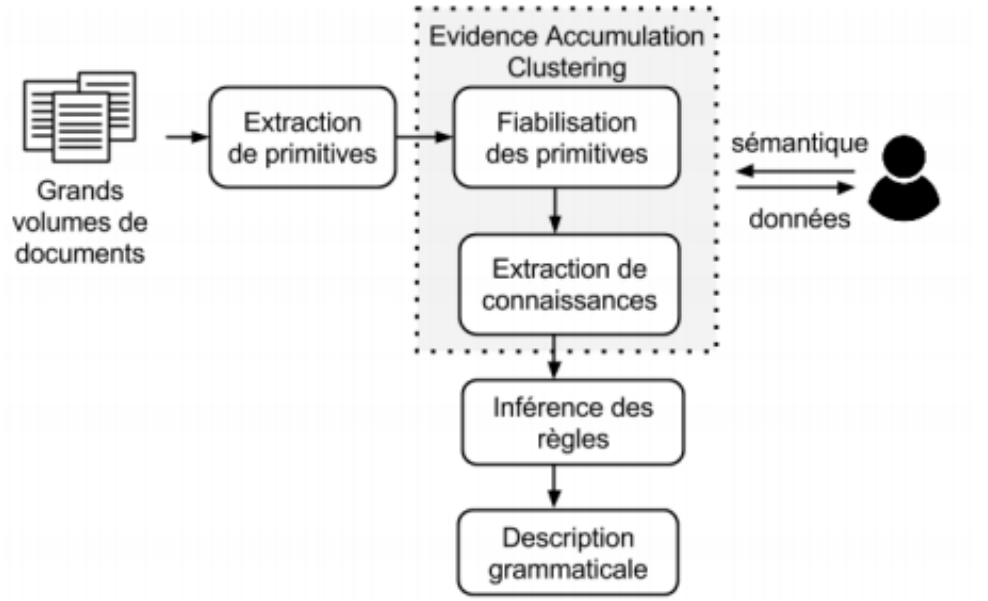


FIGURE 2.2 – Schéma général de la méthode d'inférence de règles sans vérité terrain (Carton et al., 2015)

Comme l'illustre la figure 2.2, le traitement commence par une extraction de primitives qui représentent des mots clés du document tels que les mois ou les années. Ces éléments sont détectés par une méthode de localisation d'objets à l'aide des petites zones de l'image nommées points d'intérêt. Ces derniers représentent des variations locales de luminosité et ont été utilisés pour la recherche de mots imprimés ou manuscrits dans les travaux présentés

dans (Camillerapp, 2012). De la même façon, Carton et al. utilisent cette méthode pour la localisation de mots clés qui sont, par la suite, classifiés et filtrés de manière à avoir des éléments pertinents pour la génération de règles d'inférence. Nous ne détaillerons pas ces étapes mais nous signalons que dans cette approche, le système applique l'OCR pour la fiabilisation des primitives et que l'interaction entre l'utilisateur et le système est importante pour garantir la bonne localisation de l'information.

2.2.2 Méthodes post-OCR

Dans (Rusiñol et al., 2013), les auteurs présentent une méthode d'extraction d'information dans des images de documents administratifs. Comme dans notre cas, un utilisateur sélectionne les champs qu'il désire extraire, et l'objectif est de pouvoir identifier ces mêmes informations dans d'autres instances de documents. Comme le montre la figure 2.3, les auteurs proposent de représenter le document par un graphe étoile dont le "noyau" représente l'élément cible à rechercher et le reste des nœuds correspondent à des champs spécifiques, comme un numéro de téléphone ou une adresse, qui ont été identifiés à partir de l'application de l'OCR. La connexion entre le champ cible et le mot est caractérisée par une distance polaire, comme le montre la figure 2.3.

L'inconvénient de cette méthode est qu'il n'y a pas de distinction entre le texte imprimé sur le fond du document et le texte qui a pu être ajouté et qui induit de la variabilité dans la structure du graphe. De plus, la performance de l'OCR dépend de la qualité de la numérisation du document et quand il s'agit de texte difficile à lire, ceci a aussi un impact sur la représentation structurelle de la région cible.

Dans (Hamza et al., 2008), les auteurs présentent une approche pour construire la représentation d'une facture sous forme de graphe dans le but d'une analyse de document s'appuyant sur le raisonnement à partir de cas (RAPC). Il s'agit d'une stratégie de résolution de nouveaux problèmes à partir de précédentes expériences (Hamza et al., 2007).

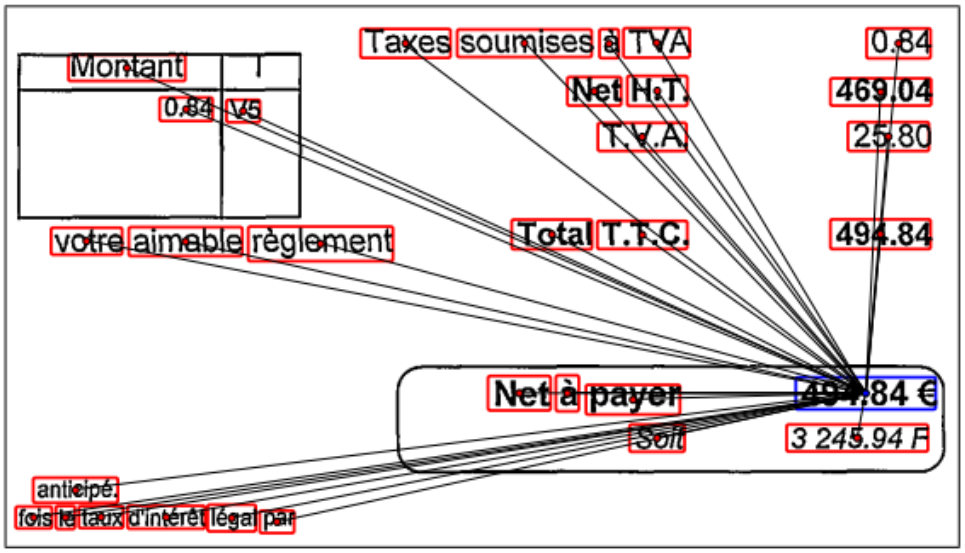


FIGURE 2.3 – La représentation structurelle du document : les zones en rouge représentent les mots identifiés par l'OCR et la zone bleu correspond au champ cible à localiser (Rusiñol et al., 2013)

La méthode s'appuie sur une liste de mots contenus dans un document sur lequel l'OCR a été appliqué. Le graphe construit décrit l'agencement de ces mots de manière à obtenir 4 types de nœuds :

- **mot** : c'est le résultat de l'OCR et il est décrit par des attributs position et une étiquette pour définir sa nature.
- **champs** : C'est un ensemble de *mots* voisins horizontalement alignés.
- **ligne horizontale** : c'est un ensemble de champs voisins dans le document.
- **bloc vertical** est l'ensemble de champs verticalement alignés.

Les auteurs identifient 2 types de structure pouvant se trouver dans l'image d'une facture : KWS, qui s'appuie sur des mots clés (Total, montant,..) et PS qui se base sur les tableaux. La figure 2.4 illustre un exemple de facture où 3 structures de type KWS et une de type PS ont été identifiées. La structure obtenue est considérée comme le modèle de la classe à laquelle le document appartient. Ainsi, chaque nouvelle instance de cette classe sera analysée et interprétée à l'aide du modèle obtenu en utilisant un calcul de distance d'ap-

pariement entre graphes.

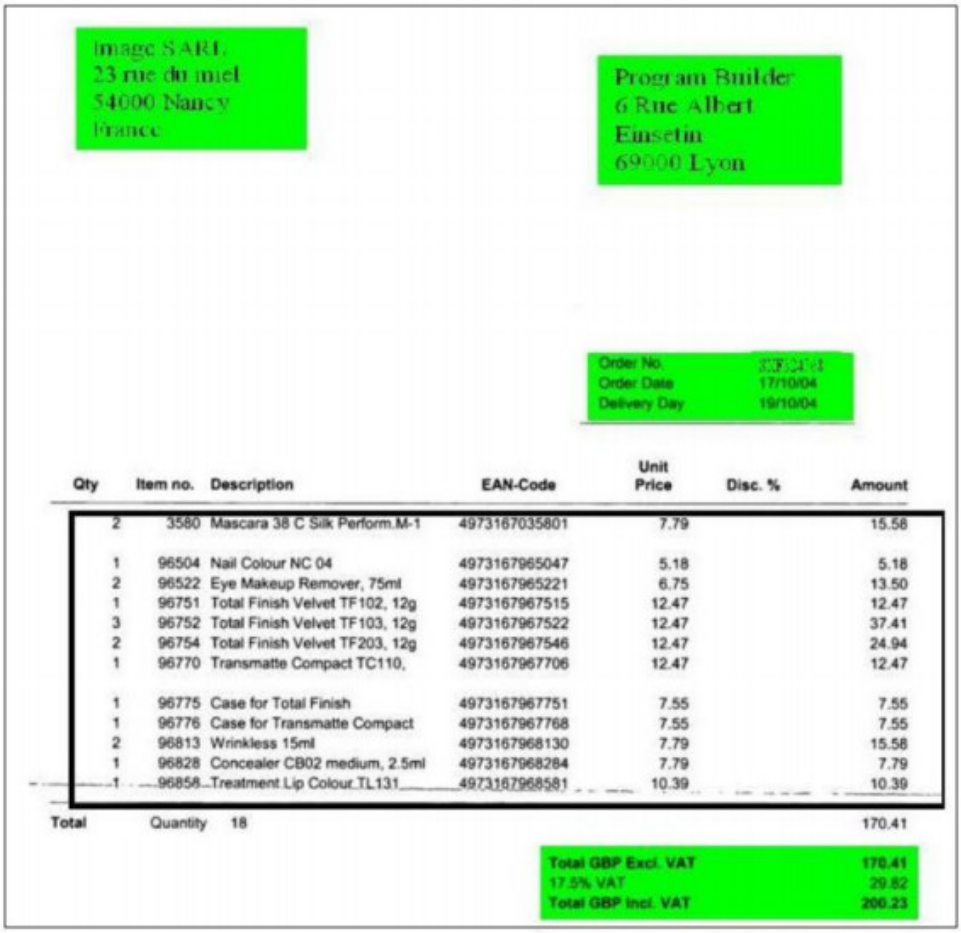


FIGURE 2.4 – Exemple de facture : les zones vertes représentent les structures de type KWS et les zones avec un contour foncé représentent les structures de type PS. (Hamza et al., 2007)

Cette méthode est efficace lorsque la classe à laquelle appartient le document est connue et elle s'adapte au type particulier de formulaires que sont les factures.

Dans (Schulz et al., 2009), les auteurs présentent un système d'extraction d'information textuelle, appelé $smartFIX$, à partir de factures provenant de différentes sources (numérisées, par Fax, par mail, etc..). L'approche décrite consiste à représenter le document sous forme d'un graphe et appliquer une méthode d'appariement entre ce dernier et un graphe enregistré dans la base

d'apprentissage. La première étape du processus consiste à appliquer des traitements d'images tels que la rotation, le redressement et la binarisation afin de préparer l'image pour l'étape de reconnaissance d'écriture en utilisant un *OCR*. Les informations textuelles sont reconnues et répertoriées sous 2 types de classes à savoir les mots clés et les valeurs. L'étape suivante consiste à construire un graphe d'adjacence entre les nœuds attribués par des étiquettes : *clé* s'il s'agit d'un noeuds représentant un mot clé et *valeur* s'il s'agit d'une valeur (date, nombre ...). Les arcs représentent la relation entre le mot clé et la valeur et sont étiquetés par rapport à leur position (bas, haut, gauche ou droite). L'extraction de connaissances est assurée par l'appariement du graphe obtenu avec le graphe stocké dans la base d'apprentissage. Pour ce faire, les auteurs utilisent la distance d'édition pour calculer la similarité entre 2 graphes. Afin de combler les erreurs de reconnaissance d'écriture et de classification, des corrections manuelles sont possibles dès lors que le résultat rendu par le système est considéré comme incertain.

2.2.3 Conclusion

À travers cette analyse de la littérature, nous retenons quatre éléments importants pour extraire la structure à partir de l'image de document. Premièrement, la construction de la structure du document doit s'appuyer sur une extraction de régions immuables, suivie de la création d'arcs entres ces différents éléments et la région recherchée. Deuxièmement, il est important d'identifier des éléments qui sont le plus stables d'une instance à une autre car ils sont la base d'un bon repérage de l'information. Troisièmement, l'information colorimétrique est rarement exploitée dans l'analyse des documents alors qu'une telle information semble être utile pour la bonne localisation de l'information. Quatrièmement, on s'intéresse à la relation entre champs et attributs mais on néglige souvent la disposition de ces différents champs les uns par rapport aux autres.

Dans la section suivante, nous présentons notre proposition pour extraire la

structure de documents sous forme de graphe. Notre approche est composée de 2 grandes parties. La première consiste en l'extraction des zones informatives tandis que la deuxième s'intéresse à établir des relations entre ces différentes régions.

2.3 Extraction des zones informatives

Dans cette section, nous présentons l'approche que nous avons proposée pour l'extraction des zones informatives qui vont devenir les nœuds du graphe représentant le document. Dans le cadre de notre travail, ces régions doivent être des éléments immuables d'une instance de document à une autre dans une catégorie donnée.

Les documents administratifs et commerciaux traités dans cette thèse sont majoritairement des formulaires complétés par un usager, comme le montre la figure 1.1. De tels documents sont caractérisés par une structure particulière qui permet de guider l'usager pour renseigner les zones à compléter. Cette structure est composée de texte mais aussi de zones graphiques, souvent de forme rectangulaire et de couleur homogène.

Concernant les informations textuelles, on en distingue deux types dans les documents : (i) le texte imprimé, qui est initialement présent sur le fond du document et (ii) le texte imprimé ou manuscrit qui est ajouté au fil du processus administratif ou commercial. Le premier est immuable et peut donc être utilisé pour positionner une information cible, alors que le second est variable d'une instance à une autre. Toutefois, la variabilité du deuxième type de texte rend complexe la tâche de séparation des deux types d'information textuelle, qui permettrait pourtant de tirer parti du premier type en l'utilisant comme élément immuable dans la représentation structurelle du document.

Contrairement au texte, les rectangles de couleur homogène sont toujours présents dans n'importe quelle instance d'une classe de document et, hormis d'éventuels surlignages, de telles zones ne sont pas ajoutées lors du cycle de vie du document.

C'est pourquoi nous nous intéressons dans cette section à la localisation de telles régions rectangulaires de couleur homogène. Tout d'abord, nous présentons les travaux portant sur cette même problématique. Ensuite, nous détaillons notre approche. Les expériences menées pour évaluer cette approche seront présentés dans la section 2.5.

2.3.1 État de l'art

L'analyse des images numériques en couleur est un domaine plus complexe que celui dédié à l'analyse des images en niveaux de gris. La représentation plus riche, sur plusieurs plans, rend en effet la tâche de segmentation de l'image plus difficile. Il semble toutefois judicieux d'exploiter une telle représentation qui appporte une information complémentaire, au lieu de la transformer dès le début du processus d'analyse de l'image.

Pour extraire les régions rectangulaires de couleur homogène, il est naturellement nécessaire de séparer l'image en plusieurs couches couleur. Leur nombre dépend du nombre de couleurs présentes dans l'image et donc de la classe de document à traiter. Les images que nous traitons sont à basse résolution et contiennent une quantité importante de distorsions. Il faut alors identifier les caractéristiques colorimétriques qui permettent au mieux d'effectuer cette séparation. En effet, les informations couleur peuvent être représentées de différentes façons, notamment selon différents espaces couleurs, dont certains sont mieux adaptés. Le bon choix de représentation garantit une meilleure segmentation de l'image, sans pour autant engendrer des déformations du contenu des documents et en particulier des régions rectangulaires.

Dans cette partie, nous présentons les différents espaces couleur identifiés dans la littérature pour décrire l'information colorimétrique. Ensuite, nous présenterons quelques travaux qui portent sur la segmentation des images de documents couleur.

L'information colorimétrique

Comme l'illustre la figure 2.5 extraite de (Vandenbroucke, 2000), différents espaces sont proposés dans la littérature pour représenter la couleur dans des images numériques. Dans cette étude, les différents espaces sont regroupés dans 4 familles différentes :

- **Les systèmes de primaires :** dans cette famille, nous trouvons l'espace RGB qui est généralement utilisé comme un point de départ pour l'acquisition de l'image et l'espace XYZ qui introduit l'axe de luminance dans la représentation de la couleur.
- **Les systèmes luminance-chrominance :** les espaces couleur de ce groupe se caractérisent par le fait d'avoir un axe de luminance dans la représentation de la couleur. Nous pouvons retrouver ces différents espaces couleur par une transformation d'un des espaces du premier groupe.
- **Les systèmes perceptuels :** ils représentent la couleur en utilisant la luminance, la teinte et la saturation.
- **Les systèmes d'axes indépendants :** ils permettent de représenter la couleur avec des composantes portant des informations non redondantes.

Ces différents systèmes de représentation sont intéressants à étudier pour comprendre l'acquisition de l'information colorimétrique. Cependant, ces différentes représentations sont sensibles à la qualité d'acquisition des images qui dépend de la résolution et de la qualité de la lumière réfléchie lors de la numérisation mais aussi de la qualité du document d'origine qui peut présenter des déformations et distorsions (qualité d'impression, pliage...).

Sur la figure 2.6, nous présentons un document qui a été numérisé avec différentes résolutions et sa représentation dans l'espace couleur RGB. À l'œil nu, on peut distinguer 3 couleurs différentes dans l'image :

- le blanc qui est la couleur du fond
- le rose qui est la couleur des champs de saisie
- le noir qui est la couleur du texte

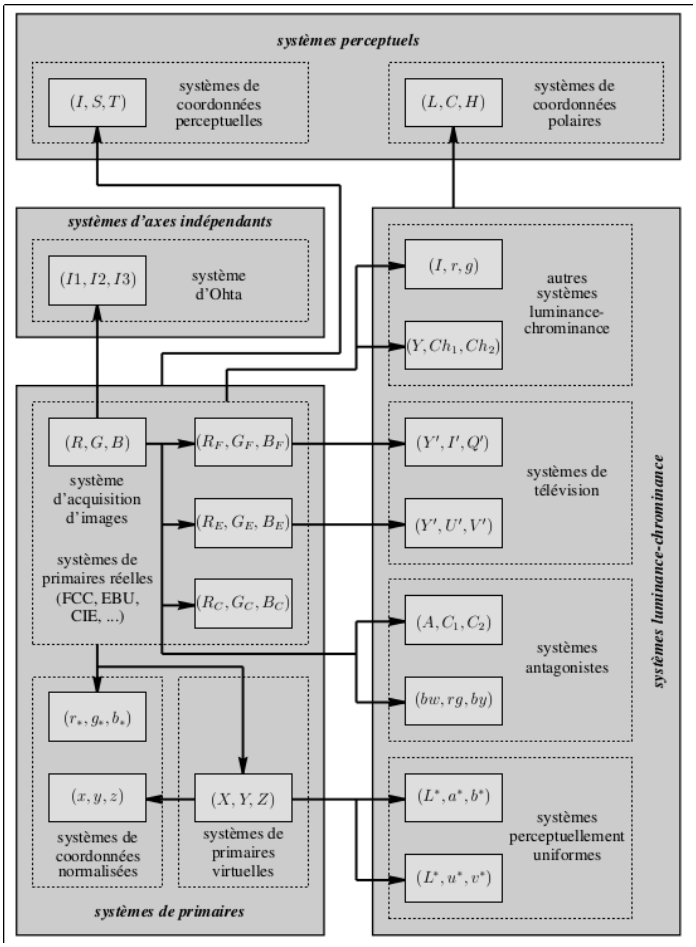
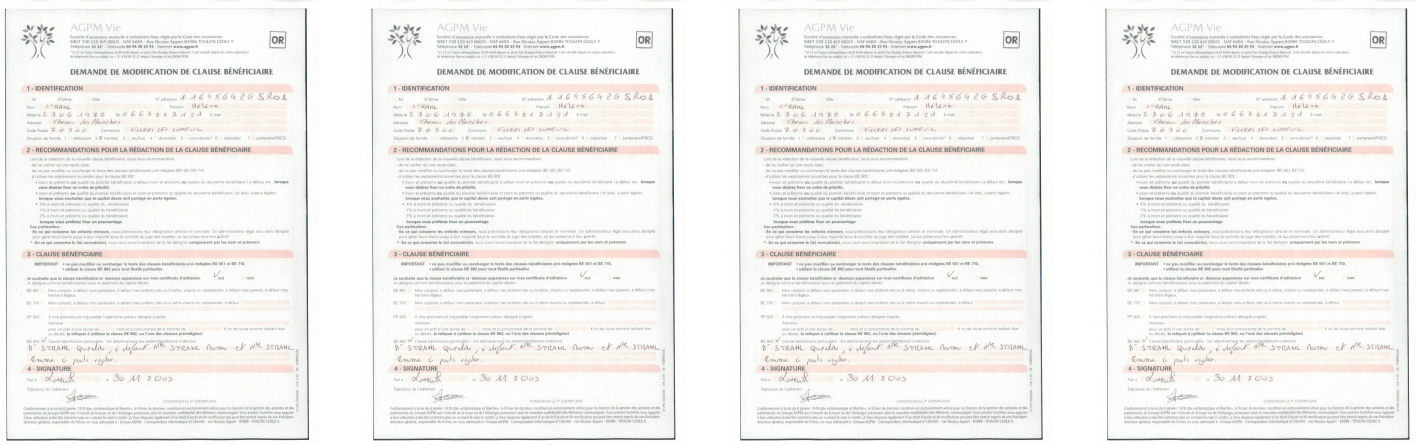


FIGURE 2.5 – Les familles de systèmes de représentation de la couleur (Vandenbroucke, 2000)

Cependant, la représentation de l'image dans l'espace couleur RGB montre que cette séparation en trois couleurs n'est pas évidente, et confirme que ce que voit l'œil humain n'est qu'une perception colorimétrique. Par conséquent, la distinction des différents objets dans l'image devient plus complexe et une segmentation de l'image est indispensable dans le processus de reconnaissance des formes.

(a) 200 pp (b) 300 pp (c) 400 pp (d) 600 pp



(e) 200 pp (f) 300 pp (g) 400 pp (h) 600 pp

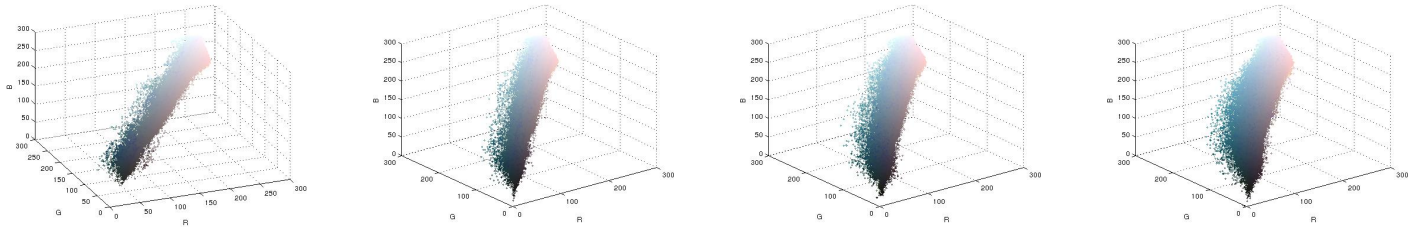


FIGURE 2.6 – Nuages pixels d'une image numérisée sous différentes résolutions

La représentation d'une image couleur dans l'espace approprié est une question de recherche abordée dans de nombreux travaux de la littérature. Dans (Loo and Tan, 2004) ou (Vandenbroucke et al., 2003), les auteurs présentent des études sur les différents espaces couleur et leur efficacité dans la segmentation des images. Ces travaux montrent qu'il n'y a pas d'espace couleur générique que l'on peut utiliser pour obtenir une bonne segmentation pour tout type d'image. En effet, l'espace couleur à utiliser dépend de l'objectif de la segmentation. Jusqu'à ce jour, dans les travaux proposés, on ne trouve aucune méthode permettant de segmenter l'image indépendamment du domaine d'application ou du type de l'image. Dans l'article (Lee et al., 1994), les auteurs présentent une étude sur les différentes transformations d'espace couleur pour avoir une bonne segmentation de l'image et affirment que le choix de l'espace couleur est également dépendant de l'objet à segmenter. La section suivante est consacrée à l'étude de quelques propositions de la littérature pour la segmentation des images de documents en couleur.

La segmentation des images de document

Afin de pouvoir reconnaître les objets dans une image couleur de document, une bonne segmentation est nécessaire. On trouve dans la littérature plusieurs travaux qui se sont intéressés à la segmentation du document sans passer par une image en niveaux de gris. Les travaux que nous avons étudiés s'appliquent sur des images de documents couleur, et proposent des méthodes de segmentation qui permettent de récupérer essentiellement des objets textuels.

Dans (Tsai and Lee, 2002), les auteurs proposent une technique de binarisation des images en couleur utilisant la luminance et la saturation comme caractéristiques colorimétriques. Les auteurs s'appuient sur ces caractéristiques pour construire un arbre de décision permettant de retrouver le meilleur seuil pour la segmentation.

Dans Leydier et al. (2004), une segmentation adaptative est proposée en

utilisant le *Kmeans* adaptable pour les images de documents manuscrits. Cette technique est utilisée pour restaurer l'information textuelle dans des images de documents historiques.

Selon les auteurs de l'article (Luccheseyz and Mitray, 2001), la segmentation peut être définie de deux manières différentes ; on peut la voir comme étant le processus permettant d'extraire des régions de couleur homogène et disjointes comme on peut la définir par l'ensemble de techniques qui cherchent à déterminer le contour limitant les différentes régions. Dès lors que l'on traite des images à basse résolution, cette étape de bas niveau est primordiale dans le processus de l'analyse du contenu de l'image.

Récemment, la litérature s'intéresse de plus en plus au traitement d'images en couleur, en essayant de proposer de nouvelles techniques permettant une segmentation efficace en tenant compte de l'information colorimétrique présente dans l'image.

Dans la section précédente, nous avons détaillé les espaces de représentation de la couleur dans les images numérisées. Nous avons montré que la perception de la couleur des objets à l'oeil nu est différente de celle "perçue" par les espaces couleur. Pour cette raison, il est important d'appliquer une réduction de couleurs pour segmenter les images. Dans la littérature, plusieurs propositions ont été introduites pour classifier les différentes nuances de couleur. On présente dans la suite les travaux les plus importants qui sont liés à la segmentation des images de document en couleur.

Nous les avons catégorisés en 2 grandes familles ; la première est liée aux caractéristiques sémantiques de la couleur et la deuxième est liée aux caractéristiques statistiques de la couleur.

- Segmentation par apprentissage du nom de la couleur

Dans la littérature, des travaux ont été proposés pour la segmentation de l'image en s'appuyant sur une sémantique ou des études psychologiques sur le perception humaine de la couleur.

Dans l'article (Van De Weijer et al., 2009), les auteurs s'intéressent à la

sémantique de la couleur présente dans l'image. En utilisant des méthodes d'analyse du contenu de documents tels que le LDA (Latent Dirichlet Allocation) (Barnard et al., 2003) et le PLSA (Probabilistic Latent Semantic Analysis) (Hofmann, 1999), leur proposition d'algorithme vise à apprendre le nom des différentes couleurs présentes dans l'image. Les auteurs confirment aussi que le fait d'utiliser l'espace couleur RGB ne permet pas de distinguer les différentes couleurs, à cause de la dépendance entre les différents plans. Ils proposent alors d'utiliser l'espace couleur $CIELAB$ qui est le plus proche de la perception de l'œil humain.

Dans l'article (Párraga et al., 2009), les auteurs présentent une classification de la couleur en se basant sur des études anthropologique et linguistique. En effet, il a été montré que l'homme identifie 11 couleurs principales et que les autres ne sont que des nuances. En utilisant l'espace couleur $CIELAB$ dont la présentation de l'information colorimétrique est proche de la perception de l'œil humain, ils proposent une approche permettant de décomposer l'espace en 11 régions différentes modulées par une fonction floue pour la rendre adaptable aux différentes contraintes de l'acquisition de l'image.

Dans les articles (Khan et al., 2012a), (Khan et al., 2012b) et (Benavente et al., 2008), les auteurs utilisent les noms attribués aux 11 différentes couleurs comme étant une caractéristique des objets identifiés. Pour l'identification des 11 couleurs, nous renvoyons le lecteur à (Berlin and Kay, 1991) où les auteurs présentent leur étude pour l'identification des 11 couleurs basiques.

Il est montré dans ces travaux que l'identification des couleurs basiques permet de manipuler plus facilement les images couleur car elle permet de réduire leur nombre et d'obtenir une image réduite à 11 couleur. Cependant, une telle approximation entraîne la suppression de la saturation de la couleur. Tout comme la teinte, la saturation est importante pour différencier les différents objets dans l'image. Dans la littérature, on trouve des travaux qui se sont intéressés à la valeur des pixels indépendamment de leur couleur basique. À partir de ces caractéristiques numériques, des méthodes de segmentation à

base d'approche de classification sont proposées.

- Fast Integral Meanshift

Dans l'article (Lebourgeois et al., 2013), les auteurs proposent une méthode de segmentation des images de document dans le but de garantir une bonne extraction de l'information textuelle. Ils utilisent le Meanshift dans une version accélérée grâce à l'utilisation des images intégrales (Viola and Jones, 2004). Le Meanshift, qui est présenté dans (Fukunaga and Hostetler, 1975, Cheng, 1995, Comaniciu and Meer, 2002), est un algorithme de classification non supervisée basé sur une fonction de densité maximale. D'une manière itérative, une fenêtre de Parzen est décalée sur un point stationnaire selon la fonction estimée de la densité du gradient dans l'espace couleur.

L'avantage de l'utilisation du meanshift est qu'aucune définition des clusters n'est imposée ; seule l'estimation de la fonction de densité est utilisée. Cependant, l'inconvénient majeur de cette approche est qu'elle est gourmande en temps de calcul. Dans (Lebourgeois et al., 2013), les auteurs proposent une amélioration à base d'images intégrales.

Sur la figure 2.7, il est clair à l'œil nu qu'il y a une meilleure qualité de l'image. Nous avons testé cette approche sur nos images et nous avons constaté deux points importants. Premièrement, le temps de calcul de la segmentation dépend de la taille de l'image. Considérant que nos documents sont de grande taille, nous avons observé un temps de traitement important. Deuxièmement, le résultat obtenu ne correspond pas à ce que nous désirons. En effet, il est bien clair que l'information textuelle est mieux représentée. Cependant, les autres informations présentent toujours des distorsions notamment les régions rectangulaires de couleur homogène.

- Une classification hiérarchique des couleurs dominantes

Dans l'article (Carel et al., 2013), les auteurs présentent une méthode de détection des couleurs dominantes dans les documents administratifs. Cette étape est nécessaire dans le processus de traitement des images de documents administratifs pour pouvoir pallier les problèmes liés à l'impression et à la

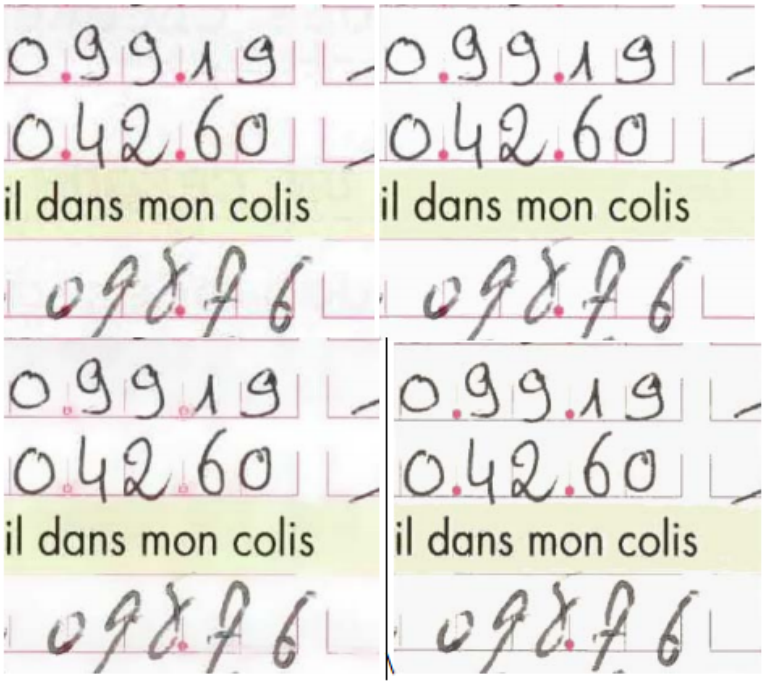


FIGURE 2.7 – De gauche à droite de haut en bas : l'image originale, Meanshift global, Meanshift spatial et meanshift integral (Lebourgeois et al., 2013).

numérisation de ce type de document. Dans l'approche proposée, les couleurs sont déterminées sans aucune connaissance a priori du nombre de couleurs présentes dans le document et visibles à l'œil nu. De plus, étant donné le cadre industriel de cette approche, aucune intervention humaine n'est autorisée dans la définition du nombre de couches, ce qui constitue une limite pour utiliser les approches "flat clustering". Les auteurs proposent une approche hybride qui consiste à appliquer un clustering hiérarchique basé sur un arbre de décision. Pour définir les différentes feuilles, ils utilisent l'algorithme *kmeans* avec $K = 2$. Partant d'un cluster contenant des pixels de l'image, une étape de division est appliquée en utilisant *kmeans* pour partager les pixels en deux clusters suivant la ressemblance colorimétrique. Ce processus est itéré jusqu'à ce que tous les clusters enfants ne contiennent que des pixels de couleur homogène. Une illustration de l'algorithme est donnée sur la figure 2.8.

Dans l'article (Carel et al., 2015), les auteurs présentent une nouvelle approche de segmentation de documents administratifs se basant sur la méthode

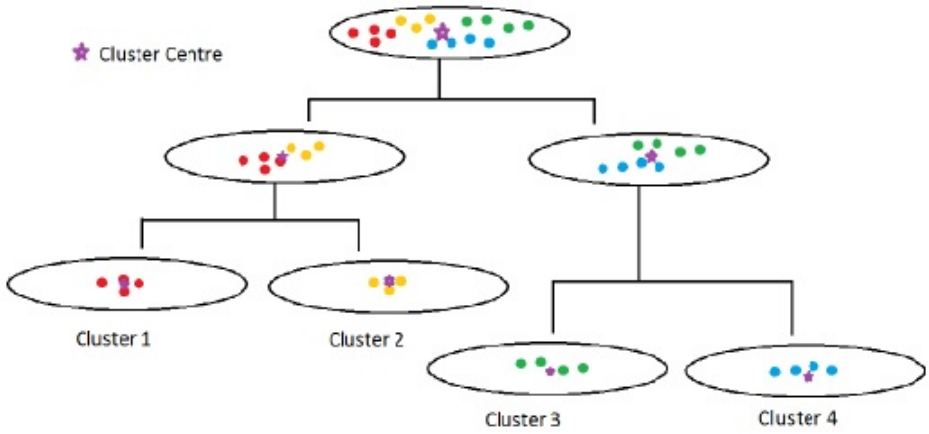


FIGURE 2.8 – Approche couleur dominante

SLIC (Simple Linear Iterative Clustering) proposée dans (Achanta et al., 2012), avec une approche multirésolution. La méthode SLIC se base sur l'algorithme superpixel. Plusieurs algorithmes ont été proposés dans la littérature. L'article (Achanta et al., 2012) les catégorise en deux principales familles. La première se base sur une représentation sous forme de graphes tels que chaque pixel de l'image est un noeud et un arc représente le degré de similarité de couleur entre deux pixels. Comme dans les articles (Shi and Malik, 2000, Felzenszwalb and Huttenlocher, 2004, Moore et al., 2008, Veksler et al., 2010), l'objectif consiste à déterminer les superpixels qui minimisent la fonction de coût. La deuxième approche consiste à utiliser le gradient ascendant. En partant d'un seul cluster dont le centre est initialisé aléatoirement, l'idée est d'itérer la division des pixels dans des clusters jusqu'à ce que le critère d'arrêt soit atteint. Dans la littérature, plusieurs travaux ont été proposés dont les articles suivants : (Comaniciu and Meer, 2002, Vedaldi and Soatto, 2008, Vincent and Soille, 1991, Levinshtein et al., 2009)

Dans toutes les approches citées et traitant les images de documents en couleur, l'information la plus importante à segmenter est le texte, ce qui, dans l'esprit de la conception de notre approche, n'est pas aussi important

que les autres éléments contenus dans l'image. Dans la partie suivante, nous présentons notre approche de segmentation de l'image et d'extraction des régions rectangulaires.

2.3.2 Notre approche

Dans cette sous-section, nous présentons notre méthode d'extraction des régions rectangulaires de couleur homogène. Nous nous intéressons à la détection de ces zones car elles permettent de définir la structure du document en représentant des éléments immuables dans toutes les instances provenant de la même catégorie de document. Nous nous sommes inspirés de la littérature pour définir notre modèle générique. Comme dans (Nikolaou and Papamarkos, 2009a) et (Loo and Tan, 2004), le traitement des images de document est composé de trois différentes parties. Tout d'abord une phase de pré-traitement est appliquée afin d'éliminer le bruit dans l'image. Puis, une quantification couleur permet d'extraire les régions de couleur homogène. Enfin, un filtre de forme est appliqué pour extraire des formes particulières. Dans notre cas, il s'agit des formes rectangulaires.

La figure 2.9 présente le modèle de traitement que nous détaillons par la suite.

Prétraitement

L'objectif de la phase de pré-traitement de l'image est de préparer celle-ci pour les traitements ultérieurs de segmentation couleur. Il s'agit d'une part d'éliminer les informations inutiles et d'autre part, de représenter les pixels avec les caractéristiques colorimétriques les plus adéquates.

- **Élimination des éléments inutiles**

Dans la recherche des régions rectangulaires de couleur homogènes, nous considérons qu'il y a deux grandes informations potentiellement non nécessaires et qu'il est intéressant d'éliminer au début du processus. Il s'agit des pixels

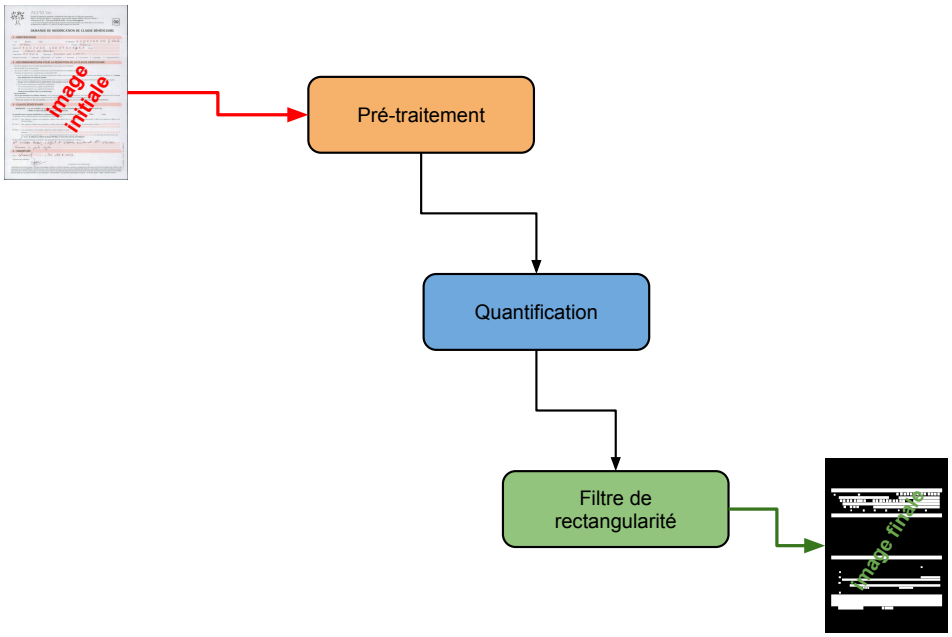


FIGURE 2.9 – Modèle générique de l'extraction des zones informatives

correspondant au bruit et à l'information textuelle. Pour cette raison, nous présentons les deux filtres que nous utilisons.

– **Filtre de bruit**

La numérisation des documents peut ajouter du bruit à l'image, ce qui crée un grand nombre de couleurs dans l'image. Afin de remédier à ce problème de numérisation, nous proposons d'utiliser un filtre de bruit qui permet de nettoyer l'image sans engendrer de distorsions ni créer de nouvelles valeurs. Dans la littérature, on trouve deux grandes familles de filtres que l'on peut utiliser dans les images en couleur (Aptoula and Lefevre, 2007, Chanussot, 1998) et qui sont illustrés sur la figure 2.10 :

– Les filtres marginaux : il s'agit de filtres appliqués sur chaque plan de couleur séparément. Dans ce cas, la corrélation entre les différents plans de couleur est complètement ignorée. Par conséquent, toute l'information susceptible d'être importante pour améliorer la qualité de l'image est éliminée par le processus du filtrage.

– Les filtres vectoriels : contrairement aux filtres marginaux, les filtres vec-

toriels traitent les plans de couleur constituant l'image simultanément, ce qui permet de tenir compte de la corrélation entre les différents plans. Cependant, ce type de filtre engendre une déformation des contours.

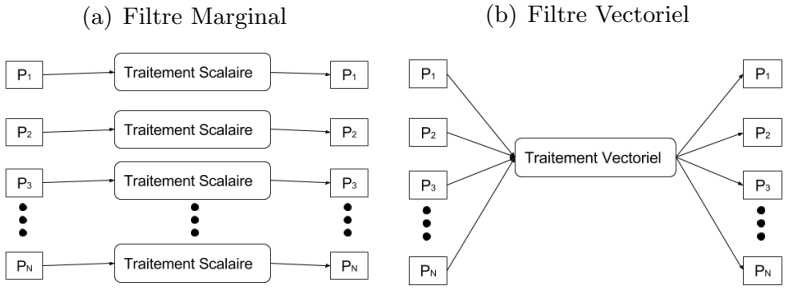


FIGURE 2.10 – Les types de traitement du bruit dans les images couleur

Afin de combler les déficits des différents types de filtres cités précédemment, nous avons utilisé l'approche $EPSF$ proposée dans l'article (Nikolaou and Papamarkos, 2009b), qui consiste en un filtre de bruit préservant les contours. Ce filtre adaptatif permet de recalculer la valeur de chaque pixel par rapport aux voisins qui sont collectés en appliquant une fenêtre de dimension $n \times n$ (Harwood et al., 1987), (Perona and Malik, 1990). Pour des questions de gain de temps, nous utilisons un masque de convolution de dimension 3×3. Puis, une distance de Manhattan est calculée et normalisée comme suit :

$$d_i = \frac{|R_{a_c} - R_{a_i}| + |G_{a_c} - G_{a_i}| + |B_{a_c} - B_{a_i}|}{3 * 255}, 0 < d_i < 1, i = 1, ..., 8 \quad (2.1)$$

où R_{a_c}, G_{a_c} et B_{a_c} sont les valeurs RGB du pixel au centre du masque. La distance résultante est utilisée pour calculer les coefficients en utilisant la formule suivante :

$$c_i = (1 - d_i)^p, p >= 1 \quad (2.2)$$

Ceci se traduit par le fait que plus les pixels sont proches en termes de valeur, plus le coefficient est important. Ainsi, nous obtenons le masque de convolution

suivant :

$$\frac{1}{\sum_{i=1}^{8} c_i} \begin{bmatrix} c_1 & c_2 & c_3 \\ c_4 & 0 & c_5 \\ c_6 & c_7 & c_8 \end{bmatrix} \tag{2.3}$$

Deux paramètres sont à fixer pour appliquer le filtre $EPSF$: l'exposant p de l'équation 2.2, qui contrôle la quantité de lissage du bruit et le nombre d'itérations *iter* pour l'application du filtre.

– **Filtre du texte**

Dans le processus d'identification des régions rectangulaires de couleur homogène, l'information textuelle n'est pas importante et dans certaines situations, il est plus judicieux de l'éliminer pour ne se focaliser que sur le reste du document.

Dans la littérature, les méthodes d'Inpainting permettent d'affecter aux les pixels de texte une valeur calculée à partir des pixels les plus proches. Plusieurs versions d'algorithme ont été proposées et nous faisons référence à l'article (Janarthanan and Jananii, 2012) qui présente une étude sur les différentes approches. La figure 2.11 illustre un exemple de résultat d'inpainting sur un document de notre corpus.

L'utilisation d'une méthode d'inpainting nécessite au préalable l'identification d'un masque qui contient les éléments à supprimer de l'image initiale. Comme nous nous intéressons à éliminer le texte, il est important de choisir une méthode de binarisation adaptative qui permet de récupérer les pixels correspondant au texte.

Il existe de nombreuses méthodes de binarisation permettant d'extraire le texte dans la littérature. Nous avons choisi celle présentée dans l'article (Gaceb et al., 2013) au regard des performances obtenues par cette méthode sur les documents de notre base d'évaluation. Sur la figure 2.12, nous présentons une comparaison de résultat entre la binarisation citée dans (Gaceb et al., 2013) et une autre en utilisant Otsu. Nous arrivons à obtenir les pixels qui représentent uniquement le texte alors qu'avec la méthode d'Otsu, plusieurs informations

(a) Image initiale (b) L'inpainting

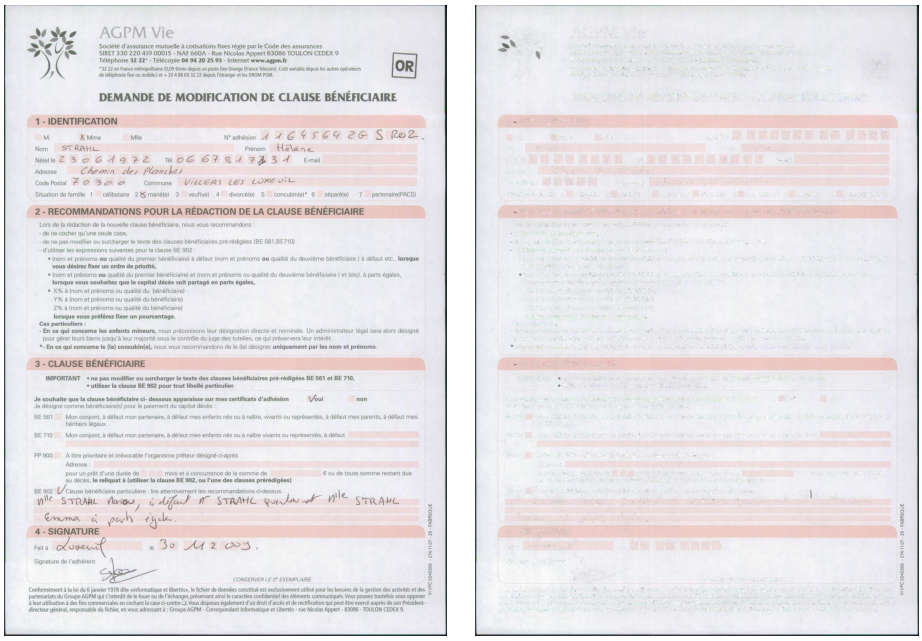


FIGURE 2.11 – Résultat de la suppression des éléments textuels (a) l'image initiale, (b) Résultat de l'*inpainting*

de bruit ont été détectées comme information textuelle.

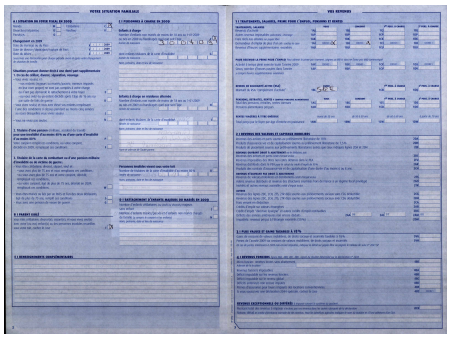
- Représentation de l'information colorimétrique

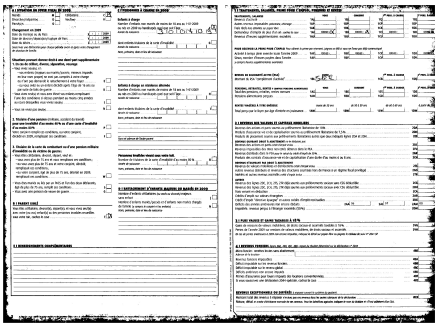
Dans la littérature, plusieurs espaces couleur ont été utilisés pour traiter des images couleur. Deux différents types d'espace couleur sont présents ; ceux dont les plans sont corrélés et ceux dont les plans ne le sont pas. Les études menées ont montré que la représentation de l'information colorimétrique dépend de la résolution de l'image mais aussi des objets à segmenter. Parfois il est intéressant d'utiliser un espace couleur donné alors que pour d'autres problèmes, il est plus judicieux d'utiliser des plans particuliers ou une combinaison de plans de différents espaces couleur.

La quantification colorimétrique

Cette étape est importante dans l'analyse du contenu de l'image et l'extraction des informations nécessaires pour les traitements suivants. Nous avons eu recours à l'application de la quantification de couleur afin de pouvoir seg-

(a) Image initiale (b) Binarisation globale





(c) Binarisation intelligence (Gaceb et al., 2013)

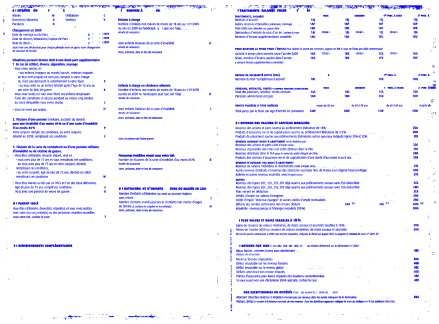


FIGURE 2.12 – Une comparaison entre des résultats de binarisation (a) l'image initiale, (b) une binarisation globale en utilisant Otsu and (c) une binarisation intelligente en utilisant l'approche de l'article (Gaceb et al., 2013)

menter l'image en un nombre fini de couleur et homogénéiser les zones dont nous avons besoin.

La quantification a pour but de regrouper les pixels selon une similarité particulière. Dans notre cas, nous cherchons à regrouper les pixels par rapport à la similarité colorimétrique. C'est pour cette raison que le choix des plans de couleur dans l'étape du prétraitement de l'image est important pour une réduction efficace du nombre de couleurs dans l'image.

Plusieurs approches peuvent être utilisées pour la réduction de la couleur dans les images de document. Nous retrouvons le *Meanshit* qui a été appliqué dans (Lebourgeois et al., 2013) et (Nikolaou and Papamarkos, 2009b) ou le SLIC qui est utilisé dans (Carel et al., 2015).

Dans le système décrit dans cette thèse, nous proposons d'utiliser une

approche simple qui ne dépend pas de plusieurs paramètres et dont le temps de calcul est réduit. Notre choix s'est porté sur l'algorithme $Kmeans$, pour lequel seul le nombre de couches doit être spécifié. L'influence de ces paramètres sera étudiée dans les expérimentations.

Filtrage de formes

Suite à la quantification de couleur, un nombre défini de K couches est obtenu, il est donc possible de séparer l'image quantifiée en K images binaires correspondant à chacune des couches. Par conséquent, nous pouvons identifier des composantes connexes et les filtrer afin de ne garder que les éléments dont la forme répond à certains critères.

Dans le cadre de l'extraction des régions rectangulaires, nous calculons un taux de rectangularité proposé dans (Rosin, 2003), qui est le rapport entre les aires des surfaces de la composante connexe CC et de la boîte englobante BE. Cette mesure est en effet la plus efficace dans le cadre de leurs expérimentations. Une région est considérée rectangulaire si le taux dépasse un seuil de rectangularité θ. Dans nos expériences, nous avons fixé ce seuil à 70%.

$$\frac{S_{CC}}{S_{BB}} \geq \theta \tag{2.4}$$

2.3.3 Conclusion

Dans cette partie, nous avons présenté notre approche d'extraction des régions immuables. Cette approche sera évaluée dans la section 2.5. Nous montrerons alors que le paramétrage des traitements est important pour garantir une stabilité du système à repérer les mêmes ancres dans les différentes instances de documents provenant de la même catégorie. Dans le dernier chapitre de la thèse, nous expliquerons comment définir automatiquement les bons traitements et le meilleur paramétrage à appliquer sur les images d'une même catégorie.

À partir de cette étape, nous pouvons construire la représentation structurelle qui décrit la disposition des éléments dans le document. La structure est composée des nœuds et d'un ensemble de connexions entre ces derniers. Si on considère que les nœuds représentent les zones informatives dont nous venons de détailler l'extraction, il nous reste à définir les relations qu'elles entretiennent. Dans la section suivante, nous nous intéressons à la construction de ces connexions.

2.4 Construction de la structure

Dans cette section, nous décrivons notre proposition de graphes d'adjacence de régions. Ces graphes sont construits à partir des régions extraites par l'approche détaillée dans la section précédente. Notre objectif est de retrouver une représentation structurelle du document permettant de fournir une description suffisante du document sans pour autant produire une construction trop complexe.

Nous choisissons de construire les arcs entre nœuds selon le principe de visibilité. Cette notion a été introduite dans (Lozano-Pérez and Wesley, 1979) pour construire le graphe permettant de connecter 2 points du chemin en évitant tout obstacle. Comme défini dans (Wismath, 1992), la visibilité consiste à lier 2 objets tel qu'aucun autre objet n'intersecte ce lien. La méthode de création de graphe de visibilité a été utilisée dans plusieurs domaines de l'informatique comme la représentation des circuits intégrés (Locteau et al., 2007) ou les plans de mouvement (Wismath, 1992).

Dans nos travaux, nous considérons que deux zones sont visibles si et seulement s'il y a une certaine visibilité horizontale ou verticale entre les deux. Autrement dit, aucune autre zone ne peut chevaucher la marge visible. Sur la figure 2.13, nous présentons un exemple de construction de graphe en nous basant sur le principe de visibilité. À gauche, nous présentons un exemple où il y a une visibilité verticale entre les nœuds. Les zones Z_1 et Z_2 sont interconnectées comme les zones Z_1 et Z_3. En revanche, il n'y a pas de connexion

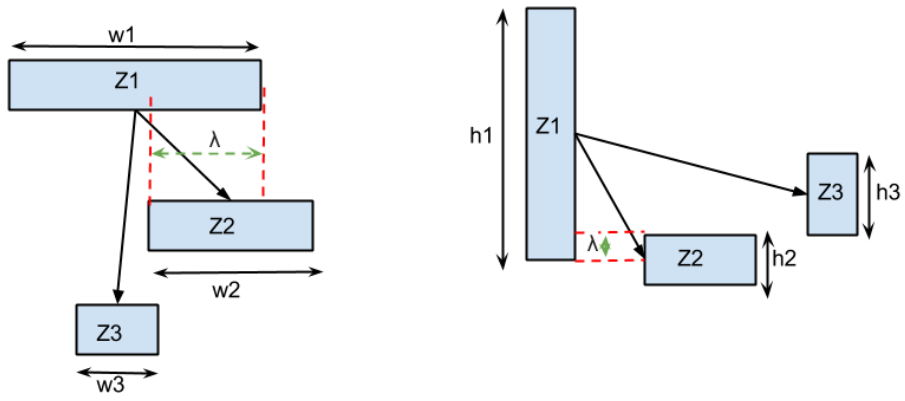


FIGURE 2.13 – Explication de la visibilité sur un exemple de graphe à gauche visibilité verticale, à droite visibilité horizontale

entre les zones Z_2 et Z_3. De la même manière, nous exposons un exemple de connexion entre les zones selon la visibilité horizontale dans la figure 2.13 à droite.

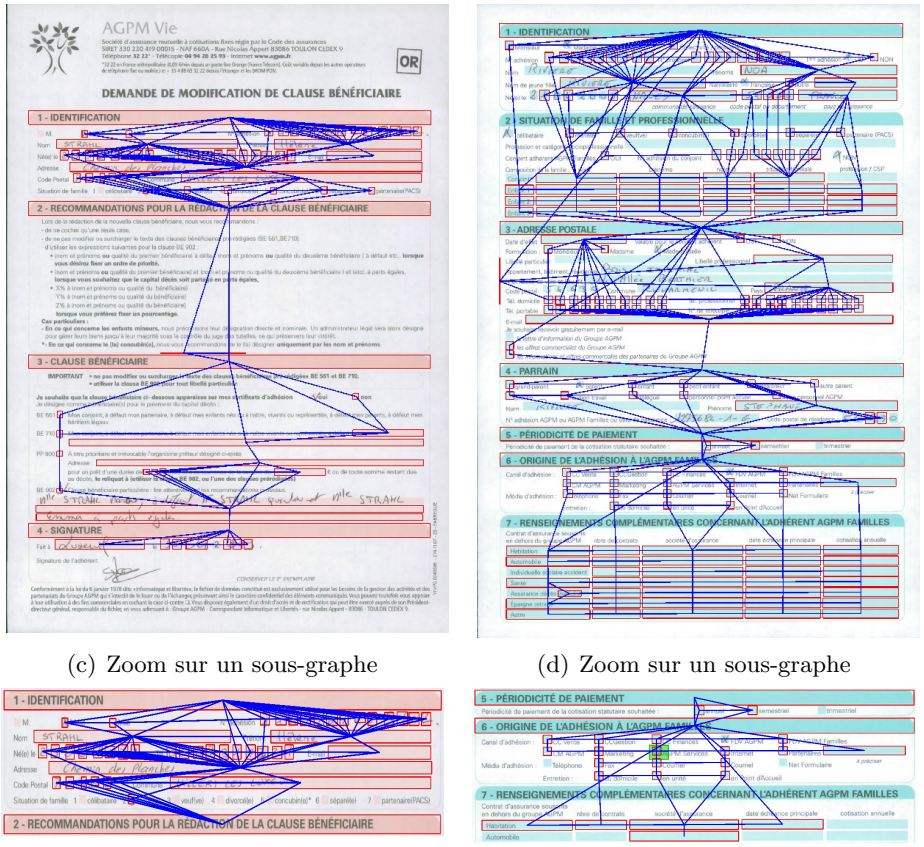
Contrairement aux autres approches, cette représentation de graphes nous garantit qu'une zone peut être décrite non seulement par un contexte local, autrement dit les voisins, mais aussi par une description globale par rapport à tout le contenu du document. Dans la figure 2.14, nous présentons des exemples de graphes obtenus par notre approche.

Afin de limiter la complexité du graphe que nous obtenons, nous rajoutons un seuil de visibilité λ permettant de vérifier si deux nœuds sont suffisamment visibles l'un à l'autre pour créer un arc.

Outre sa structure, la qualité de la représentation sous forme de graphe dépend également du choix des caractéristiques portées par les nœuds et les arcs. Les graphes que nous proposons de construire décrivent l'agencement des zones informatives que nous extrayons à l'aide de notre chaîne de traitement présentée précédemment. Ces zones sont de forme rectangulaire et de couleur homogène. Pour cette raison, nous proposons d'attribuer aux nœuds des informations colorimétriques et géométriques qui décrivent la couleur et les

dimensions de la région. Les informations géométriques sont normalisées par rapport à la taille du document pour garantir l'invariance aux changements d'échelle. Quant aux arcs, nous proposons de les décrire par des distances horizontale et verticale, elles aussi normalisées. Nous considérons que ces informations sont suffisantes pour la vérification de la catégorie du document et l'obtention d'une représentation indépendante des coordonnées géométriques.

(a) Graphe d'un document complet (b) Graphe d'un document complet



(c) Zoom sur un sous-graphe (d) Zoom sur un sous-graphe

FIGURE 2.14 – Exemples de graphes extraits par l'approche proposée

2.5 Expériences et résultats

Dans cette partie, nous présentons les expériences réalisées pour évaluer l'approche d'extraction des zones informatives décrite dans la section 2.3. Les expériences relatives aux graphes définis dans la section 2.4 seront présentées

dans le chapitre suivant, après la présentation de notre algorithme de recherche de sous-graphes. L'objectif des expérimentations décrites ici est d'évaluer la qualité des zones extraites avec notre approche par rapport à une vérité terrain définie manuellement. Dans ce cadre, nous avons utilisé la métrique *Zonemap* proposée dans (Galibert et al., 2014), qui permet une analyse fine des performances obtenues.

2.5.1 Base de données et protocole d'évaluation

Base de document

La base de documents utilisée pour les tests contient 130 documents administratifs et commerciaux en couleurs répartis en 8 classes différentes en fonction du modèle de formulaire. La figure 2.15 donne un exemple de document pour chacune de ces classes.

FIGURE 2.15 – Exemples de documents pour chacune de 8 classes

En utilisant l'outil GEDI (Groundtruthing Environment for Document Images), nous avons généré manuellement la vérité terrain contenant les régions homogènes. Certaines zones sont remplies alors que d'autres ne le sont pas. La figure 2.16 propose quelques exemples de zones extraites. Le tableau 2.1 décrit le nombre d'images ainsi que le nombre de zones identifiées dans la vérité terrain pour chaque classe. En moyenne, 73 zones sont identifiées par document.

(a) Commune

(b) cases à cocher

(c) Date de naissance

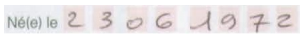


FIGURE 2.16 – Exemple de zones informatives

Classes	Nbr images	Nbr zones
1	11	80
2	31	59
3	29	231
4	14	21
5	12	14
6	11	166
7	9	74
8	13	80
Total	130	moy : 73

TABLE 2.1 – Nombre de zones homogènes dans chaque classe

Présentation de la métrique

Pour évaluer la qualité des résultats fournis par l'approche proposée dans ce chapitre, nous avons utilisé la métrique zonemap proposée lors du projet MAURDOR et définie dans l'article Galibert et al. (2014). Cette métrique, complexe à mettre en œuvre et consommatrice de ressources, permet en effet une analyse fine de la performance de l'extracteur de zones en quantifiant les erreurs selon différentes configurations. Ces configurations sont présentées ci-dessous et illustrées dans la figure 2.17.

- Match : comme indiqué dans la figure 2.17 (a), cette configuration permet de calculer l'erreur liée au chevauchement entre une zone hypothèse et une zone de référence.
- Merge : il s'agit de calculer l'erreur quand plusieurs zones de référence sont chevauchées par une zone d'hypothèse. Ce cas de figure correspond à une fusion de zones (figure 2.17 (b))

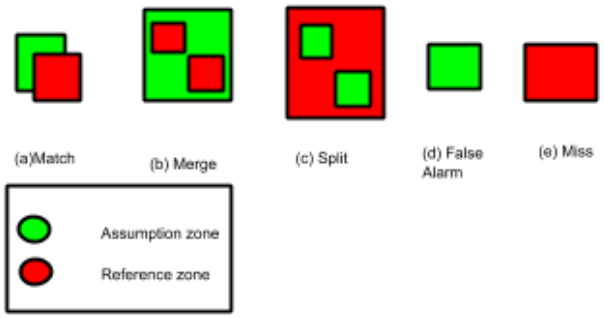


FIGURE 2.17 – Les différentes configuration de la métrique ZoneMap

- Split : inversement au cas précédent, nous calculons ici l'erreur quand plusieurs zones d'hypothèses se chevauchent avec une même zone de référence. Ce cas correspond à la figure 2.17 (c).
- FalseAlarm : ce sont les zones détéctées mais qui n'ont pas de correspondant dans la vérité terrain. Ce cas correspond à la figure 2.17 (d).
- Miss : ce sont les zones qui sont référées dans la vérité terrain mais qui n'ont pas pu être détectées avec l'extracteur. Ce cas correspond à la figure 2.17 (e).

Il est possible d'attribuer des poids différents à chacun des types d'erreur. Dans nos expériences, nous avons mis des poids uniformes.

Configuration du système

Pour ces expériences, nous avons instancié le modèle générique de chaîne de traitement proposé sur la figure 2.9. Par ailleurs, nous avons cherché à étudier l'impact de chaque traitement sur la performance de notre extracteur. Les paramètres étudiés sont :

- l'impact du prétraitement par filtrage $EPSF$;
- l'impact du prétraitement par *inpainting* ;
- le paramètre k de l'algorithme des k-moyennes tel que $k \in [2, 4]$; Ce choix d'intervalle vient du fait que les documents de notre corpus contiennent généralement 2 couleurs différentes : celle des zones que nous voulons ex-

traire et celle de la couche d'arrière plan qui est généralement de couleur unie. Nous allons jusqu'à la valeur 4 car les documents contiennent aussi généralement des éléments textuels et du bruit de couleurs différentes.

– l'impact de l'espace colorimétrique de travail : RGB, $YCbCr$, $CIELab$ ou le niveau de gris $Gris$.

Avec ces différentes configurations, nous testons donc, pour chaque document, 48 chaînes différentes de traitement d'extraction des régions homogènes. Les paramètres du filtre EPSF ainsi que le seuil de rectangularité ont été fixés de façon empirique ($p = 10$, $iter = 5$, $\theta = 0.7$). Notons que, dans le chapitre 4, nous avons étendu l'étude des espaces couleurs à des combinaisons de plans provenant de différents espaces.

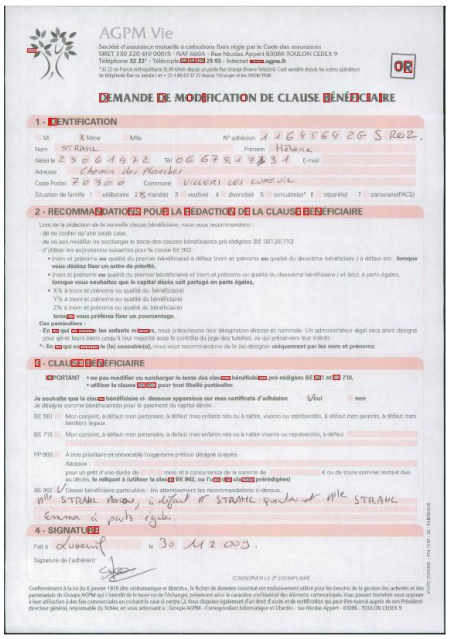
Résultats

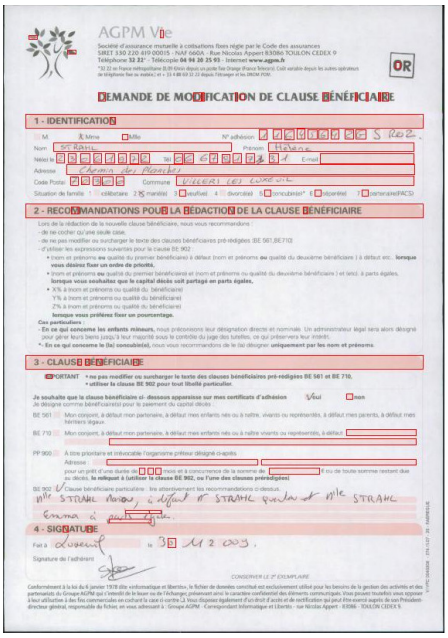
Sur la figure 2.18, nous illustrons quelques exemples de résultats obtenus avec différentes configurations de paramètres. On y constate l'impact du choix de configuration, qui fait grandement varier la qualité des résultats.

Dans le tableau 2.2, nous présentons les résultats des erreurs par type de configuration de la métrique Zonemap appliquées sur toute la base de documents $Itesoft$. Les différents résultats prouvent l'utilité de l'utilisation de l'information colorimétrique pour la détection des régions de couleur homogène car les meilleures performances correspondent à des configurations de notre chaîne de traitement où on utilise un espace couleur RGB, $YCbCr$ ou $CIELAB$. Dans les 2 dernières colonnes du tableau, nous constatons que nous avons moins d'erreurs de détection des bonne zones lorsqu'on utilise un espace couleur autre que le RGB. Par conséquent, ceci explique pourquoi nous avons des valeurs minimales au niveau des erreurs de $Match$, $Split$ et $Merge$ quand on utilise l'espace RGB.

Dans le tableau 2.3, nous présentons l'erreur totale selon la métrique $ZoneMap$ par classe et par configuration. Nous remarquons que la meilleure configuration n'est pas la même pour toutes les classes. L'application du filtre

(a) EPSF=1, In=0, $K = 2$, $\{Lab\}$ (b) EPSF=1, In=1, $K = 3$, $\{Lab\}$





(c) EPSF=1, In=0, $K = 2$, $\{Lab\}$ (d) EPSF=1, In=1, $K = 3$, $\{Lab\}$

FIGURE 2.18 – Exemples de zones extraites

EPSF et/ou de l'*inpainting* sont utiles pour certaines classes telle que la classe 7 tandis que pour d'autres, ce n'est pas le cas (pour les classes 1 et 5 par exemple). Nous observons également que l'utilisation de l'information colorimétrique est importante dans l'extraction des zones pour certaines classes alors que pour d'autres, ce n'est pas le cas.

TABLE 2.2 – Résultats expérimentaux

Prétraitement		**Quantification**		**Configuration Zonemap**				
EPSF	**Inpaint**	**K**	**Espace couleur**	**Match**	**Merge**	**Split**	**FalseAlarm**	Miss
Oui	Oui	2	RGB	0.1	0.02	0.17	0.02	0.47
			YCbCr	0.08	0.02	0.2	0.02	0.45
			Lab	0.09	0.03	0.14	0.02	0.38
			Gris	0.11	0.04	0.2	0.01	0.51
		3	RGB	0.11	0.03	0.13	0.02	0.3
			YCbCr	0.11	0.04	0.13	0.01	0.28
			Lab	0.13	0.05	0.13	0.01	0.2
			Gris	0.12	0.03	0.14	0.02	0.29
		4	RGB	0.11	0.05	0.09	0.03	0.25
			YCbCr	0.12	0.03	0.11	0.03	0.24
			Lab	0.13	0.04	0.1	0.04	0.18
			Gris	0.11	0.04	0.13	0.02	0.23
	Non	2	RGB	0.1	**0.01**	0.17	0.02	0.47
			YCbCr	0.08	0.27	0.2	0.02	0.44
			Lab	0.09	0.3	0.15	0.02	0.37
			Gris	0.11	0.28	0.2	0.01	0.51
		3	RGB	0.11	0.08	0.13	0.02	0.29
			YCbCr	0.11	0.04	0.13	0.01	0.28
			Lab	0.13	0.05	0.13	0.01	0.19
			Gris	0.12	0.1	0.15	0.02	0.28
		4	RGB	0.11	0.05	0.09	0.03	0.25
			YCbCr	0.12	0.03	0.11	0.03	0.24
			Lab	0.13	0.04	0.11	0.04	**0.18**
			Gris	0.11	0.04	0.12	0.02	0.23
Non	Oui	2	RGB	0.06	0.3	0.21	0.01	0.49
			YCbCr	0.06	0.04	0.21	0.02	0.48
			Lab	0.07	0.03	0.18	0.02	0.42
			Gris	0.06	0.04	0.21	0.01	0.59
		3	RGB	0.11	0.04	0.13	0.01	0.32
			YCbCr	0.12	0.04	0.12	0.01	0.32
			Lab	0.14	0.06	0.14	0.01	0.2
			Gris	0.12	0.04	0.14	0.01	0.32
		4	RGB	0.11	0.05	0.08	0.02	0.26
			YCbCr	0.13	0.04	0.1	0.02	0.24
			Lab	0.13	0.06	0.12	0.04	0.18
			Gris	0.11	0.06	0.11	0.01	0.28
	Non	2	RGB	**0.05**	0.33	0.24	0.01	0.48
			YCbCr	0.06	0.03	0.24	0.01	0.47
			Lab	0.07	0.03	0.18	0.02	0.41
			Gris	0.06	0.04	0.23	0.01	0.58
		3	RGB	0.11	0.06	0.13	0.01	0.32
			YCbCr	0.12	0.09	0.13	**0**	0.31
			Lab	0.14	0.06	0.14	0.01	0.2
			Gris	0.12	0.11	0.14	0.01	0.31
		4	RGB	0.11	0.94	**0.08**	0.02	0.27
			YCbCr	0.14	0.43	0.09	0.02	0.23
			Lab	0.3	0.06	0.12	0.04	0.18
			Gris	0.11	0.05	0.1	0.01	0.27

En utilisant cette métrique, nous avons eu un premier avis sur l'efficacité de notre extracteur générique. Nous avons eu recours à l'utilisation de la vérité terrain sur notre base de documents. Cependant, dans le cadre de l'application d'itesoft, nous ne pouvons pas nous appuyer sur cette métrique car nous ne possédons pas de vérité terrain. Dans le chapitre 4, nous nous intéressons à la recherche de la meilleure configuration pour chaque classe de documents en exploitant une évaluation du système sans vérité terrain.

TABLE 2.3 – Résultats expérimentaux

Prétraitement		Quantification		Classes							Base
EPSF	Inpaint	K	Espace couleur	1(11)	2(31)	3(29)	4(14)	5(12)	6(11)	7(9)	8(13)	
Oui	Oui	2	RGB	1.07	1.01	0.68	0.21	0.11	0.99	1.02	1.05	0.76
			YCbCr	1.09	1.04	0.68	0.17	0.1	0.99	1	1.07	0.77
			Lab	1.13	0.9	0.39	0.11	0.03	0.99	0.63	1.03	0.65
			Gris	1.09	1.02	0.79	0.69	0.28	1	1	1.06	0.87
		3	RGB	0.83	0.78	0.58	0.09	0.075	0.85	0.63	0.88	0.59
			YCbCr	0.83	0.76	0.56	0.14	0.07	0.82	0.57	0.88	0.58
			Lab	0.73	0.67	0.45	0.17	0.07	0.81	**0.51**	0.81	0.53
			Gris	0.87	0.81	0.65	0.14	0.04	0.85	0.53	0.86	0.59
		4	RGB	0.79	0.61	0.41	0.07	0.05	0.8	0.64	0.89	0.53
			YCbCr	0.67	0.76	0.41	0.07	0.04	0.72	0.68	0.83	0.52
			Lab	0.69	0.7	0.38	0.13	0.02	0.66	0.51	0.83	**0.49**
			Gris	0.87	**0.55**	0.49	0.06	0.04	0.77	0.55	**0.77**	0.51
	Non	2	RGB	1.07	1.01	0.71	0.21	0.11	0.95	1.02	1.05	0.76
			YCbCr	1.09	1.04	0.69	0.17	0.1	2.88	1	1.07	1
			Lab	1.13	0.92	0.39	0.11	0.03	3.12	0.63	1.03	0.92
			Gris	1.09	1.02	0.8	0.69	0.28	2.88	1	1.06	1.1
		3	RGB	0.83	0.78	0.59	0.1	0.07	1.18	0.63	0.88	0.63
			YCbCr	0.83	0.76	0.57	0.14	0.07	0.77	0.57	0.88	0.57
			Lab	0.73	0.67	0.45	0.17	0.07	0.79	**0.51**	0.81	0.53
			Gris	0.87	0.81	0.66	0.14	0.04	1.33	0.53	0.86	0.65
		4	RGB	0.79	0.61	0.52	0.07	0.05	0.79	0.64	0.89	0.53
			YCbCr	0.68	0.76	0.41	0.07	0.04	0.7	0.68	0.83	0.52
			Lab	0.69	0.7	**0.37**	0.13	0.02	0.66	0.51	0.83	**0.49**
			Gris	0.87	0.55	0.49	0.06	0.04	0.77	0.55	0.77	0.51
Non	Oui	2	RGB	1.07	1.03	2.92	0.31	0.05	0.98	1.03	1.21	1.08
			YCbCr	1.09	1.03	0.71	0.35	0.05	0.98	0.99	1.21	0.8
			Lab	1.23	0.93	0.48	0.16	0.03	0.99	0.68	1.19	0.71
			Gris	1.09	1.04	0.81	0.79	0.39	0.98	1.03	1.21	0.92
		3	RGB	0.83	0.79	0.7	0.17	0.07	0.8	0.66	0.86	0.61
			YCbCr	0.83	0.77	0.69	0.17	0.07	0.8	0.64	0.88	0.61
			Lab	0.81	0.7	0.51	0.18	0.07	0.76	0.56	0.82	0.55
			Gris	0.86	0.81	0.74	0.18	0.07	0.79	0.63	0.86	0.62
		4	RGB	0.78	0.64	0.4	0.1	0.03	0.82	0.58	0.84	0.53
			YCbCr	0.67	0.8	0.4	**0.06**	**0.02**	0.76	0.69	0.86	0.53
			Lab	0.61	0.71	0.38	0.25	0.07	**0.64**	0.55	1.03	0.53
			Gris	0.87	0.64	0.45	0.13	0.08	0.76	0.75	0.89	0.57
	Non	2	RGB	1.21	1.03	3.18	0.31	0.05	0.93	1.03	1.21	1.12
			YCbCr	1.22	1.03	0.7	0.35	0.05	0.93	0.99	1.21	0.81
			Lab	1.22	0.93	0.48	0.16	0.02	0.94	0.68	1.18	0.7
			Gris	1.23	1.04	0.81	0.78	0.39	0.93	1.03	1.21	0.93
		3	RGB	0.85	0.79	0.72	0.17	0.07	0.93	0.65	0.86	0.63
			YCbCr	0.85	0.77	0.71	0.17	0.07	1.18	0.64	0.88	0.66
			Lab	0.81	0.7	0.52	0.18	0.07	0.75	0.57	0.81	0.55
			Gris	0.87	0.81	0.76	0.18	0.07	1.29	0.63	0.86	0.68
		4	RGB	7.85	0.64	0.4	0.1	0.03	0.8	0.58	0.94	1.42
			YCbCr	3.7	0.8	0.4	0.06	**0.02**	0.74	0.69	0.86	0.91
			Lab	**0.61**	0.71	0.38	0.25	0.07	0.63	0.55	1.03	0.53
			Gris	0.72	0.64	0.45	0.13	0.07	0.76	0.75	0.89	0.55

2.6 Conclusion

Dans ce chapitre, nous avons présenté un modele de lecture de documents de type formulaires. Son objectif est de caractériser les parties immuables des documents d'une même classe, pour les utiliser comme ancres de repérage d'une information à localiser. La solution proposée est composée de deux grandes étapes. La première étape consiste en l'extraction de régions immuables. Afin d'obtenir ces zones, l'image, qui peut présenter des distorsions engendrées par l'impression et la numérisation, nécessite parfois des prétraitements pour corriger ces anomalies et obtenir une bonne segmentation. Nous avons proposé un modèle générique qui a besoin d'être configuré. Nos expériences montrent que le comportement du système diffère d'une configuration à une autre et qu'il est nécessaire d'arriver à déterminer la bonne configuration pour chaque catégorie de document. Ce point sera traité dans la suite du manuscrit.

La deuxième étape consiste en la construction d'un graphe avec des arcs liant les différentes régions. En effet, les nœuds seuls, même accompagnés de leurs attributs, sont insuffisants pour décrire la structure du document. Une modélisation des agencements relatifs des rectangles les uns par rapport aux autres est nécessaire pour décrire plus précisément la structure topologique du document. Dans ce chapitre, nous proposons de modéliser la notion de voisinage par le biais d'une relation de visibilité entre les rectangles. Ainsi, deux nœuds sont liés par un arc si les rectangles correspondant sont considérés visibles l'un de l'autre.

Dans la partie expérimentation, nous avons montré que l'approche permettait, si elle bien configurée pour une classe donnée, d'extraire la plupart des rectangles de couleur homogène. Dans le chapitre suivant, nous présentons comment nous exploitons cette représentation pour localiser les régions d'intérêt.

Chapitre 3

Recherche d'isomorphisme de sous-graphes pour la localisation d'information

Sommaire

3.1 Introduction

Dans le chapitre précédent, nous avons présenté une approche permettant de transformer une image de document en une représentation sous forme de graphe. La représentation obtenue correspond à un modèle physique du document qui décrit l'agencement d'éléments constitutifs (des régions rectangulaires de couleur homogène) en fonction d'une relation originale de visibilité. La structuration proposée, y compris au niveau des attributs portés par les nœuds et les arcs du graphe, a pour caractéristique principale d'être invariante à la translation, au changement d'échelle et à des modifications de couleurs. L'objectif sous-jacent est que des représentations similaires soient obtenues pour des documents d'une même classe mais numérisés dans des conditions d'acquisition variables (position du document sur le scanner, résolution de la numérisation, rendus variables des couleurs).

Ce chapitre est dédié à l'opérationnalisation de la représentation proposée et constitue une seconde contribution importante de la thèse. Nous y décrivons comment, à partir de la définition par un utilisateur d'une zone d'intérêt (la requête) sur l'image d'un document modèle, le système développé permet d'identifier la zone correspondante dans d'autres occurrences de la même classe de document, en maintenant les bonnes propriétés d'invariance mentionnées dans le paragraphe précédent. Pour ce faire, le système s'appuie naturellement sur des graphes pour représenter structurellement la requête correspondant à la zone d'intérêt. La zone "image" recherchée est ainsi décrite par un graphe de "petite" taille qui positionne cette zone (elle aussi rectangulaire) par rapport aux éléments constitutifs du document modèle, toujours avec la relation de visibilité.

Dans ce contexte, le problème de l'identification de la zone d'intérêt dans un document inconnu consiste alors à rechercher une occurrence du graphe modèle, correspondant à la zone d'intérêt, dans un graphe cible modélisant le document à traiter. Dans la communauté scientifique de la reconnaissance de formes à base de graphes, ce problème est connu comme étant celui de

la recherche d'isomorphisme de sous-graphes. Toutefois, lorsque les graphes sont construits à partir de chaînes de traitements d'images appliquées à des données qui peuvent être bruitées, les algorithmes classiques de la littérature, qui recherchent des occurrences exactes du sous-graphe dans le graphe cible, deviennent inopérants.

Dans ce chapitre, nous décrivons l'approche originale développée dans le cadre de la thèse et publiée dans (Lerouge et al., 2016) pour résoudre ce problème particulier. L'approche s'affranchit d'éventuelles modifications des attributs et de la topologie du graphe cible, pour tenir compte de la variabilité des résultats de traitements antérieurs. Le système repose sur une adaptation de la distance d'édition entre graphes, que nous formulons sous la forme d'un programme linéaire en nombres binaires, résolu par un solveur mathématique proposé dans la communauté de la recherche opérationnelle. Le système est évalué sur un problème synthétique et deux problèmes applicatifs, dont celui de la thèse.

Le chapitre est structuré de la façon suivante. Dans une première section, nous définissons formellement le problème traité et positionnons ce problème par rapport aux travaux de la littérature. Puis, la section suivante décrit la principale formulation proposée pour modéliser ce problème sous forme de programme linéaire en nombres binaires, ainsi que quelques variantes. La section suivante est dédiée aux expérimentations et aux résultats obtenus qui sont présentés et discutés.

3.2 Définition et positionnement du problème

Dans cette section, nous définissons formellement le problème de la recherche d'occurrence d'un sous-graphe modèle au sein d'un graphe cible, lorsque des modifications de la topologie des graphes et de leur étiquetage doivent être tolérées. Nous considérons ici des graphes simples et attribués sur les nœuds et sur les arcs :

Définition 1. Un graphe simple attribué est un 4-tuple $G = (\mathcal{V}, \mathcal{E}, \mu, v)$ tel que :

- $\mathcal{V}$ est un ensemble fini de nœuds dans G,
- $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ est un ensemble fini d'arcs dans G,
- $\mu : \mathcal{V} \longrightarrow \mathcal{L}_{\mathcal{V}}$ est la fonction d'étiquetage des nœuds avec $\mathcal{L}_{\mathcal{V}}$ l'ensemble des labels des nœuds,
- $v : \mathcal{E} \longrightarrow \mathcal{L}_{\mathcal{E}}$ est la fonction d'étiquetage des arcs avec $\mathcal{E}$ l'ensemble des labels des arcs. Si $\mathcal{E}$ est défini comme étant une relation symétrique, $(u, v) \in \mathcal{E} \Longleftrightarrow (v, u) \in \mathcal{E}, \forall u, v \in \mathcal{V} \times \mathcal{V}$ alors G est considéré comme un graphe non dirigé. Inversement, il sera considéré comme étant dirigé.

Avec une telle définition, un arc $e = (u, v) \in \mathcal{E}$ est identifié par la paire constituée d'un nœud de départ u et d'un nœud d'arrivée v. Avec ces notations, il est possible d'introduire le concept de sous-graphe :

Définition 2. Soient $G_1 = (\mathcal{V}_1, \mathcal{E}_1, \mu_1, v_1)$ et $G_2 = (\mathcal{V}_2, \mathcal{E}_2, \mu_2, v_2)$ deux graphes simples attribués. G_1 est un sous-graphe de G_2 ($G_1 \subseteq G_2$) si et seulement si

- $\mathcal{V}_1 \subseteq \mathcal{V}_2$
- $\mathcal{E}_1 \subseteq \mathcal{E}_2 \cap \mathcal{V}_1 \times \mathcal{V}_2$
- $\mu_1(v) = \mu_2(v), \forall v \in \mathcal{V}_1$
- $v_2(e) = v_2(e), \forall e \in \mathcal{E}_1$

Dans la littérature de l'analyse de graphes, la notion d'isomorphisme est fondamentale pour comparer des graphes :

Définition 3. L'isomorphisme entre 2 graphes $G_1 = (\mathcal{V}_1, \mathcal{E}_1, \mu_1, v_1)$ et $G_2 = (\mathcal{V}_2, \mathcal{E}_2, \mu_2, v_2)$ est une fonction bijective $f : \mathcal{V}_1 \longrightarrow \mathcal{V}_2$ telle que $\forall u, v \in \mathcal{V}_1$: $(u, v) \in \mathcal{E}_1 \Longleftrightarrow (f(u), f(v)) \in \mathcal{E}_2$.

À partir des deux définitions précédentes, il est possible d'introduire la notion d'isomorphisme de sous-graphe :

Définition 4. Soient $G_1 = (\mathcal{V}_1, \mathcal{E}_1, \mu_1, v_1)$ et $G_2 = (\mathcal{V}_2, \mathcal{E}_2, \mu_2, v_2)$ deux graphes simples attribués. Une fonction injective $f : \mathcal{V}_1 \longrightarrow \mathcal{V}_2$ est un isomorphisme

de sous-graphe de G_1 à G_2 s'il existe un sous-graphe $G \subseteq G_2$ tel que $f(.)$ est un isomorphisme de graphes entre G_1 et G.

La recherche d'isomorphismes de sous-graphes a donné lieu à une littérature abondante (Cordella et al., 1999, 2004, Ghahraman et al., 1980, Solnon, 2010, Ullmann, 1976, Wong et al., 1990) dans différents domaines d'application dont la biologie, la chimie, les réseaux sociaux ou l'analyse d'images de scène. Les algorithmes décrits dans ces contributions sont théoriquement très intéressants pour des applications de reconnaissance de formes car ils permettent de résoudre le paradigme bien connu de segmentation/reconnaissance. Toutefois, d'un point de vue pratique, les algorithmes existants présentent deux inconvénients majeurs. Le premier est leur complexité algorithmique. En effet, la recherche d'isomorphisme de sous-graphes appartient aux problèmes NP-complets (Garey and Johnson, 1990), ce qui ne permet pas de traiter des graphes de grande taille. Le second inconvénient est l'exigence d'un appariement exact, qui n'est pas toujours présent dans des applications où les graphes peuvent présenter des distorsions engendrées par une procédure de numérisation ou par un processus de construction des graphes (squelettisation, segmentation de régions, etc). L'appariement doit alors s'adapter à cette différence par le biais de la relaxation de certaines contraintes et accepter des substitutions des attributs des nœuds et des arcs mais aussi des différences de structure.

Dans ce contexte, une première étape a été franchie au sein de l'équipe "Document et Apprentissage" du LITIS, avec les travaux présentés dans (Le Bodic et al., 2012). Dans cette contribution, la recherche d'isomorphisme de sous-graphe est formulée comme un problème d'optimisation, dont le but est de trouver le sous-graphe du graphe cible, isomorphe au graphe modèle, qui minimise une fonction de coût d'appariement avec celui-ci.

Ainsi, à partir d'une distance $d : \mathcal{G} \times \mathcal{G} \rightarrow \mathbb{R}^+$ exprimant la dissimilarité entre deux graphes, l'objectif est de trouver le sous-graphe G du graphe cible

G_2 isomorphe à G_1 et dont la distance avec le graphe requête G_1 est minimale.

$$G = \underset{G_i \subseteq G_2}{\operatorname{argmin}}\, d(G_1, G_i) \text{ sous la contrainte } G_i \text{ isomorphe à } G_2 \tag{3.1}$$

Dans (Le Bodic et al., 2012), la distance d est la somme des coûts d'édition des arcs et des nœuds. Ces coûts sont définis en fonction des attributs. Le problème d'optimisation est formulé comme un programme linéaire dans lequel l'objectif est de minimiser d. Des contraintes linéaires sont définies pour imposer l'isomorphisme. L'approche proposée permet ainsi de tolérer des modifications des attributs des nœuds et des arcs. Toutefois, les contraintes du problème d'optimisation imposent la présence d'un isomorphisme de structure entre le graphe modèle et le sous-graphe du graphe cible. De ce fait, l'approche proposée ne tolére pas l'absence d'un nœud ou d'un arc du graphe modèle dans le graphe cible.

Dans cette thèse, nous avons étendu le travail décrit dans (Le Bodic et al., 2012) en intégrant la possibilité que des nœuds ou des arcs du sous-graphe modèle ne soient pas présents dans le sous-graphe du graphe cible. Pour ce faire, il est nécessaire d'une part de modifier les contraintes du problème d'optimisation, et d'autre part, d'intégrer dans le calcul de la distance d des coûts correspondant aux modifications de la topologie.

Dans la littérature, cette notion d'édition de la structure des graphes est fréquemment rencontrée dans les travaux concernant la distance d'édition entre graphe. La distance d'édition entre graphes d_{GED} a été le sujet de très nombreuses contributions au cours de la dernière décennie. Elle permet d'évaluer la dissimilarité entre deux graphes, en tolérant des modifications de topologie. De nombreuses approches ont été proposées pour calculer ou approximer la valeur de d_{GED} (Fischer et al., 2015, Riesen and Bunke, 2015, Almohamad and Duffuaa, 1993, Justice and Hero, 2006). Le lecteur intéressé trouvera de très bons états de l'art dans (Gao et al., 2010, Riesen, 2015).

Définition 5. Soient G_1 et G_2 deux graphes, la distance d'édition de graphes

entre G_1 et G_2 est définie par :

$$d_{GED}(G_1, G_2) = \min_{o=(o_1,\ldots,o_k)\in\mathcal{O}} \sum_i c(o_i) \tag{3.2}$$

Dans cette équation, $\mathcal{O}$ désigne l'ensemble de tous les chemins d'édition $o = (o_1, \ldots, o_k)$ permettant de transformer G_1 en G_2. Une opération élémentaire d'édition o_i peut correspondre aux opération suivantes :

- une substitution de nœuds $(v_1 \to v_2)$
- une substitution d'arcs $(e_1 \to e_2)$
- une suppression de nœud $(v_1 \to \epsilon)$
- une suppression d'arcs $(e_1 \to \epsilon)$
- une insertion de nœud $(\epsilon \to v_2)$
- une insertion d'arcs $(\epsilon \to e_2)$

où $v_1 \in \mathcal{V}_1$, $v_2 \in \mathcal{V}_2$, $e_1 \in \mathcal{E}_1$, $e_2 \in \mathcal{E}_2$ et ϵ un élément inexistant permettant de modéliser les opérations d'insertion et de suppression.

Dans l'équation 3.2, $c(.)$ est une fonction donnant le coût des opérations élémentaires d'édition o_i listées ci-dessus. Cette fonction doit satisfaire les contraintes suivantes :

- $c(v_1 \to v_2) \leq c(v_1 \to v) + c(v \to v_2)\forall$
- $c(e_1 \to e_2) \leq c(e_1 \to e) + c(e \to e_2)$
- $c(v_1 \to \epsilon) \leq c(v_1 \to v) + c(v \to \epsilon)$
- $c(e_1 \to \epsilon) \leq c(e_1 \to e) + c(e \to \epsilon)$
- $c(\epsilon \to v_2) \leq c(\epsilon \to v) + c(v \to v_2)$
- $c(\epsilon \to e_2) \leq c(\epsilon \to e) + c(e \to e_2)$

Si la fonction de coût satisfait aussi les conditions d'identité positive et de symétrie ainsi que l'inégalité triangulaire des opérations élémentaires d'édition o_i, la distance d'édition des graphes est alors formellement une métrique.

Dans le cadre de notre travail, notre objectif est de trouver le sous-graphe G de G_2 qui minimise la distance à G_1. On se trouve donc confronté à un cadre restreint de la distance d'édition puisque la distance n'a pas à prendre

en considération de coûts d'insertion de nœud ou d'arc dans G_1. En effet, si un chemin d'édition contenant des insertions transforme le graphe G_1 en un graphe G_i, il existera un sous-graphe de G_i qui résultera du même chemin d'édition sans insertion dans G_1. Le coût de ce chemin sera inférieur, et donc le sous-graphe sera une meilleure solution.

Dans la section suivante, nous montrons que ce problème peut être formulé comme un programme linéaire en nombres binaires.

3.3 Formulation linéaire en nombres binaires pour la recherche de sous-graphes

Afin de résoudre le problème défini dans la section précédente, nous proposons dans cette thèse de le formuler comme un Programme Linéaire en Nombres Binaires (PLNB). Un PLNB est une restriction d'un programme linéaire en nombres entiers dans lequel les variables utilisées sont binaires. Ces techniques relèvent du domaine plus général de la programmation mathématique.

3.3.1 La programmation linéaire en nombres binaires

La forme générale d'un programme linéaire en nombres binaires est la suivante :

$$\min_x \ c^T x \tag{3.3}$$

$$\text{s.t. } Ax \leq b \tag{3.4}$$

$$x \in \{0,1\}^n \tag{3.5}$$

où $c \in \mathbb{R}^n$, $A \in \mathcal{M}_{m,n}(\mathbb{R})$ et $b \in \mathbb{R}^m$ sont les données du problème. La solution est représentée sous forme d'un vecteur x de n variables binaires. A et b sont utilisés pour définir des contraintes linéaires d'inégalité. Une solution possible du problème est un vecteur $x \in \{0,1\}^n$ tel que les contraintes 3.4 sont respectées. La fonction objectif $c^T x$ est une combinaison linéaire des variables binaires x. La solution optimale est celle qui minimise la fonction objectif 3.3

sur l'ensemble des solutions possibles.

3.3.2 Formulation linéaire du problème MCSM

Afin de formuler le problème de recherche de sous-graphe à coût minimum sous forme de programme linéaire en nombres binaires, nous définissons 4 ensembles de variables binaires :

- $\forall i \in \mathcal{V}_1, \forall k \in \mathcal{V}_2, x_{i,k} \in \{0,1\}$ permet de représenter la substitution des attributs des nœuds $(i \to k)$. $x_{i,k}$ vaut 1 si i est substitué par k, 0 sinon.
- $\forall i \in \mathcal{V}_1, \alpha_i \in \{0,1\}$ permet de représenter la suppression des nœuds $(i \to \epsilon)$. α_i vaut 1 si i est supprimé de G_1 et 0 sinon.
- $\forall ij \in \mathcal{E}_1, \forall kl \in \mathcal{E}_2, y_{ij,kl} \in \{0,1\}$ permet de représenter la substitution des arcs $(ij \to kl)$. $y_{ij,kl}$ vaut 1 si ij est substitué par kl, 0 sinon.
- $\forall ij \in \mathcal{E}_1, \beta_{ij} \in \{0,1\}$ permet de représenter la suppression des arcs $(ij \to \epsilon)$. β_{ij} vaut 1 si ij est supprimé de G_1 et 0 sinon.

On note $\mathbf{x} = (x_{i,k})_{i \in \mathcal{V}_1, k \in \mathcal{V}_2}$, $\boldsymbol{\alpha} = (\alpha_i)_{i \in \mathcal{V}_1}$, $\mathbf{y} = (y_{ij,kl})_{ij \in \mathcal{E}_1, kl \in \mathcal{E}_2}$ et $\boldsymbol{\beta} = (\beta_{ij})_{ij \in \mathcal{E}_1}$. Si on dispose d'une fonction coût $c(.)$ telle que définie dans la section 3.2, la fonction objectif du programme linéaire en nombres binaires est alors la somme des coûts des opérations d'édition élémentaires o_i nécessaires pour apparier G_1 au sous-graphe $G \subseteq G_2$:

$$\min_{\mathbf{x},\mathbf{y},\boldsymbol{\alpha},\boldsymbol{\beta}} \left(\sum_{i \in \mathcal{V}_1} \sum_{k \in \mathcal{V}_2} x_{i,k} \cdot c(i \to k) + \sum_{i \in \mathcal{V}_1} \alpha_i \cdot c(i \to \epsilon) \right. \\ \left. + \sum_{ij \in \mathcal{E}_1} \sum_{kl \in \mathcal{E}_2} y_{ij,kl} \cdot c(ij \to kl) + \sum_{ij \in \mathcal{E}_1} \beta_{ij} \cdot c(ij \to \epsilon) \right) \quad (3.6)$$

Afin d'obliger le 4-tuple $(\mathbf{x}, \mathbf{y}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ à décrire un chemin d'édition valide $o \in \mathcal{O}$ transformant G_1 en $G \subseteq G_2$, les contraintes suivantes sont nécessaires :

- Un nœud de G_1 peut être apparié avec au plus un nœud de G_2 :

$$\sum_{k \in \mathcal{V}_2} x_{i,k} \leq 1 \quad \forall i \in \mathcal{V}_1 \quad (3.7)$$

Si un nœud de G_1 n'est apparié avec aucun nœud de G_2, il est supprimé :

$$\alpha_i = 1 - \sum_{k \in \mathcal{V}_2} x_{i,k} \quad \forall i \in \mathcal{V}_1 \tag{3.8}$$

- Un nœud de G_2 peut être apparié avec au plus un nœud de G_1 :

$$\sum_{i \in \mathcal{V}_1} x_{i,k} \leq 1 \quad \forall k \in \mathcal{V}_2 \tag{3.9}$$

- un arc de G_1 peut être apparié avec au plus un arc de G_2, tant que les 2 extrémités ont été correctement appariées. Cette contrainte quadratique peut être transformée en deux contraintes linéaires :

$$\sum_{l \in \mathcal{V}_2, kl \in \mathcal{E}_2} y_{ij,kl} \leq x_{i,k} \quad \forall ij \in \mathcal{E}_1, \forall k \in \mathcal{V}_2 \tag{3.10}$$

$$\sum_{k \in \mathcal{V}_2, kl \in \mathcal{E}_2} y_{ij,kl} \leq x_{j,l} \quad \forall ij \in \mathcal{E}_1, \forall l \in \mathcal{V}_2 \tag{3.11}$$

Si un arc de G_1 n'est apparié avec aucun arc de G_2, il est supprimé :

$$\beta_{ij} = 1 - \sum_{kl \in \mathcal{E}_2} y_{ij,kl} \quad \forall ij \in \mathcal{E}_1 \tag{3.12}$$

Les équations 3.8 et 3.12 ne sont pas nécessaires dans les contraintes du programme PLNB car elles sont respectées implicitement quand les contraintes 3.7, 3.10 et 3.11 sont satisfaites. Afin de réduire la taille de l'espace de recherche (autrement dit, le nombre de variables), nous remplaçons les variables de suppression dans la fonction objectif par leurs expressions issues de 3.8 et 3.12. Ainsi, nous obtenons la formulation linéaire en nombres binaires suivante :

$$\begin{aligned} \min_{\mathbf{x},\mathbf{y}} \Bigg(& \sum_{i \in \mathcal{V}_1} \sum_{k \in \mathcal{V}_2} x_{i,k} \cdot (c(i \to k) - c(i \to \epsilon)) + \sum_{i \in \mathcal{V}_1} c(i \to \epsilon) \\ & + \sum_{ij \in \mathcal{E}_1} \sum_{kl \in \mathcal{E}_2} y_{ij,kl} \cdot (c(ij \to kl) - c(ij \to \epsilon)) + \sum_{ij \in \mathcal{E}_1} c(ij \to \epsilon) \Bigg) \end{aligned} \tag{3.13a}$$

sous les contraintes :

$$\sum_{k \in \mathcal{V}_2} x_{i,k} \leq 1 \qquad \forall i \in \mathcal{V}_1 \tag{3.13b}$$

$$\sum_{i \in \mathcal{V}_1} x_{i,k} \leq 1 \qquad \forall k \in \mathcal{V}_2 \tag{3.13c}$$

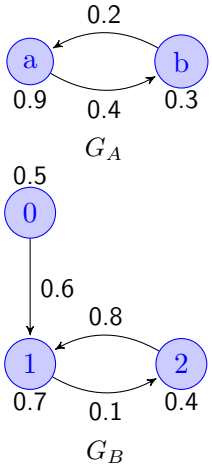
$$\sum_{l \in \mathcal{V}_2, kl \in \mathcal{E}_2} y_{ij,kl} \leq x_{i,k} \quad \forall ij \in \mathcal{E}_1, \forall k \in \mathcal{V}_2 \tag{3.13d}$$

$$\sum_{k \in \mathcal{V}_2, kl \in \mathcal{E}_2} y_{ij,kl} \leq x_{j,l} \quad \forall ij \in \mathcal{E}_1, \forall l \in \mathcal{V}_2 \tag{3.13e}$$

$$x_{i,k} \in \{0,1\} \quad \forall i \in \mathcal{V}_1, \forall k \in \mathcal{V}_2 \tag{3.13f}$$

$$y_{ij,kl} \in \{0,1\} \quad \forall ij \in \mathcal{E}_1, \forall kl \in \mathcal{E}_2 \tag{3.13g}$$

La figure 3.1 illustre une telle modélisation du problème sur des graphes synthétiques, en donnant les valeurs des variables et de la fonction objectif. Dans ce problème les coûts de suppression de nœuds et arcs sont fixés à 1. Dans le premier cas (colonne de gauche), le graphe G_A est recherché dans le graphe G_B. Dans le second cas (colonne de droite), le graphe G_B est recherché dans le graphe G_A. Ce second cas illustre la suppression d'un nœud et d'un arc, générant un "surcoût" de 2.



modèle : G_A cible : G_B	modèle : G_B cible : G_A
$x_{a,0} = 0$	$x_{0,a} = 0$
$x_{a,1} = 1$	$x_{0,b} = 0$
$x_{a,2} = 0$	$x_{1,a} = 1$
$x_{b,0} = 0$	$x_{1,b} = 0$
$x_{b,1} = 0$	$x_{2,a} = 0$
$x_{b,2} = 1$	$x_{2,b} = 1$
$y_{ab,01} = 0$	$y_{01,ab} = 0$
$y_{ab,12} = 1$	$y_{01,ba} = 0$
$y_{ab,21} = 0$	$y_{12,ab} = 1$
$y_{ba,01} = 0$	$y_{12,ba} = 0$
$y_{ba,12} = 0$	$y_{21,ab} = 0$
$y_{ba,21} = 1$	$y_{21,ba} = 1$
$d = 1.2$	$d = 3.2$

FIGURE 3.1 – Exemple de valeurs de variable et la fonction objectif obtenu par MCSM sur jeu de test

La formulation proposée est dans la forme décrite par les équations 3.13a à 3.13g est applicable aux graphes simples dirigés. Toutefois, par une simple modification des notations, elle est aussi théoriquement valable pour les multi-graphes. Par ailleurs, même si les variantes concernées ne sont pas évaluées dans cette thèse, nous montrons ci-après que la formulation peut être modifiée pour traîter le problème de recherche de sous-graphe induit, et pour traîter les graphes non dirigés.

3.3.3 Une extension pour les sous-graphes induits

La recherche de sous-graphe induit est un problème plus contraint que le précédent, défini par :

Définition 6. Soient $G_1 = (\mathcal{V}_1, \mathcal{E}_1, \mu_1, \nu_1)$ et $G_2 = (\mathcal{V}_2, \mathcal{E}_2, \mu_2, \nu_2)$ deux graphes. Une fonction injective $f : \mathcal{V}_1 \rightarrow \mathcal{V}_2$ est un isomorphisme des sous-graphes induits de G_1 à G_2 si et seulement si $\forall (u, v) \in \mathcal{V}_1 \times \mathcal{V}_1, (u, v) \in \mathcal{E}_1 \Leftrightarrow (f(u), f(v)) \in \mathcal{E}_2$.

Par conséquent, suivant la définition 6, de nouvelles contraintes doivent être ajoutées dans le programme linéaire en nombres binaires :

$$\sum_{i \in \mathcal{V}_1} x_{i,k} + \sum_{j \in \mathcal{V}_1} x_{j,l} - \sum_{ij \in \mathcal{E}_1} y_{ij,kl} \leq 1 \quad \forall kl \in \mathcal{E}_2 \tag{3.14}$$

Ces contraintes permettent d'assurer que tous les arcs du sous-graphe associé dans G_2 sont appariés avec des arcs du graphe G_1.

3.3.4 Une extension pour les graphes non-dirigés

La formulation donnée par les équations 3.13c à 3.13g est dédiée aux graphes dirigés. Nous proposons une extension pour les graphes non dirigés $G = (\mathcal{V}, \mathcal{E}, \mu, \nu)$. Dans un graphe non dirigé, nous avons :

$$ij \in \mathcal{E} \Leftrightarrow ji \in \mathcal{E}, \forall i, j \in \mathcal{V} \times \mathcal{V}$$

Ainsi, ayant deux arcs non dirigés, il existe deux façons de les apparier. Ceci nous amène à remplacer les équations 3.13d et 3.13e par l'équation suivante dans le programme linéaire en nombres entiers :

$$\sum_{l \in \mathcal{V}_2 : kl \in \mathcal{E}_2} y_{ij,kl} \leq x_{i,k} + x_{j,k} \quad \forall ij \in \mathcal{E}_1, \forall k \in \mathcal{V}_2 \tag{3.15}$$

Il est à noter aussi que les variables $x_{i,k}$ et $x_{j,k}$ sont mutuellement exclusives (autrement dit, il est impossible d'affecter la valeur 1 au deux simultanément) à cause de la contrainte 3.9. Ainsi, la contrainte 3.15 garantit toujours qu'un arc de G_1 est apparié à au plus un arc de G_2.

3.3.5 Implémentation de la formulation : gestion d'instances multiples

Une fois la formulation du MCSM implémentée dans un solveur de programme linéaire en nombres entiers, l'appariement des deux graphes (modèle et cible) en utilisant les coûts d'édition fixés a priori produit le meilleur résultat de correspondance un à un des nœuds et des arcs, avec la possibilité de supprimer des nœuds et des arcs du graphe modèle. Telle que définie dans les équations 3.13c à 3.13g, la formulation PLNB est capable de retourner la solution optimale. Selon le contexte de l'application, il est possible que le graphe cible possède plusieurs instances du graphe modèle. Il existe plusieurs alternatives pour gérer ce cas de solutions multiples (Danna et al., 2007). Dans le cadre de nos travaux, nous avons choisi d'appeler de manière itérative le programme et de rejeter les solutions optimales obtenues après chaque appel, pour ne pas les retrouver à nouveau. Une telle solution est linéaire au regard du nombre d'instances.

Il existe plusieurs façons pour rejeter les solutions optimales $(\bar{x}\ \bar{y})^T$. L'idée générale est de rajouter au modèle une nouvelle contrainte qui coupe la solution courante. Ainsi, la solution optimale en cours devient impossible dans la prochaine exécution. Dans le cadre de nos travaux, nous avons rajouté la

contrainte suivante :

$$\sum_{i \in \mathcal{V}_1, k \in \mathcal{V}_2} \left(\sum_{j \in \mathcal{V}_1} \bar{x}_{j,k} \right) * x_{i,k} = 0$$

Ceci permet de rejeter tout nœud de $\mathcal{V}_2$ ayant été utilisé dans la solution optimale en cours $(\bar{x}\ \bar{y})^T$. Autrement dit, pour chaque nœud k de l'ensemble $\mathcal{V}_2$, s'il existe un nœud j de l'ensemble $\mathcal{V}_1$ apparié à k, alors $x_{i,k}$ vaut 0 pour chaque nœud i de l'ensemble $\mathcal{V}_1$.

3.4 Expérimentations et résultats

Cette section a pour objectif d'évaluer expérimentalement l'approche proposée dans ce chapitre pour la recherche de sous-graphe tolérante aux modifications de topologie et d'étiquetage. De telles expérimentations sont rendues difficiles par l'absence de bases de données de référence dédiées à ce type de tâche. En effet, s'il existe des bases de graphes dédiées à la recherche d'isomorphisme de sous-graphes (Cordella et al., 2004), elles entrent toutes dans un cadre dans lequel il existe un isomorphisme de sous-graphe entre le graphe modèle et le graphe cible.

Dans ce contexte, nos expérimentations ont été scindées en trois parties :

- des expérimentations sur une problématique de localisation de symboles dans des documents graphiques. Ces expérimentations avaient pour objectif de valider expérimentalement la formulation proposée et de montrer l'apport de la tolérance aux modifications de topologie sur un problème historiquement important au sein de l'équipe "document et apprentissage" du LITIS ;
- des expérimentations sur des graphes synthétiques, dont le but était essentiellement de comparer les temps de calcul obtenus par la nouvelle formulation par rapport à celle proposée dans (Le Bodic et al., 2012) ;
- des expérimentations sur les graphes décrits dans le chapitre précédent, pour l'application de localisation de zones informatives dans des images

de formulaires fournies par la société Itesoft.

Les sous-sections suivantes décrivent ces 3 expérimentations.

3.4.1 Localisation de symboles

La localisation de symboles est un problème d'analyse d'images de documents dont l'objectif est de détecter les occurrences de symboles modèles dans des documents cibles. Les méthodes structurelles sont souvent adoptées pour résoudre ce problème car (i) les symboles peuvent généralement être définis comme une composition de partitions et (ii) les symboles sont fréquemment connectés à d'autres éléments dans l'image, ce qui rend difficile une segmentation explicite de ces objets qui permettrait l'utilisation d'approches statistiques. Dans ce contexte, les outils de reconnaissance de forme à base de graphes sont donc particulièrement appropriés.

Représentation structurelle

Dans nos expériences, les images de symboles et de documents sont représentées en utilisant des graphes d'adjacence de régions $G = (\mathcal{V}, \mathcal{E}, \mu, \nu)$ tel que $\mathcal{V}$ désigne l'ensemble des régions de l'image et $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ représente l'ensemble des relations d'adjacence entre les régions. Dans nos travaux, la fonction $\mu : \mathcal{V} \rightarrow \mathbb{R}^{26}$ décrit la morphologie de la région avec son aire et 25 moments de Zernike calculés en utilisant l'approche détaillée dans (Teague, 1980). la fonction $\nu : \mathcal{E} \rightarrow \mathbb{R}^2$ représente quant à elle deux propriétés différentes de la relation d'adjacence :

– l'échelle relative entre les deux régions :

$$\min(A(i), A(j)) / \max(A(i), A(j))$$

où $A(i)$ désigne l'aire de la région i,

– la distance euclidienne entre les centres de gravité des deux régions nor-

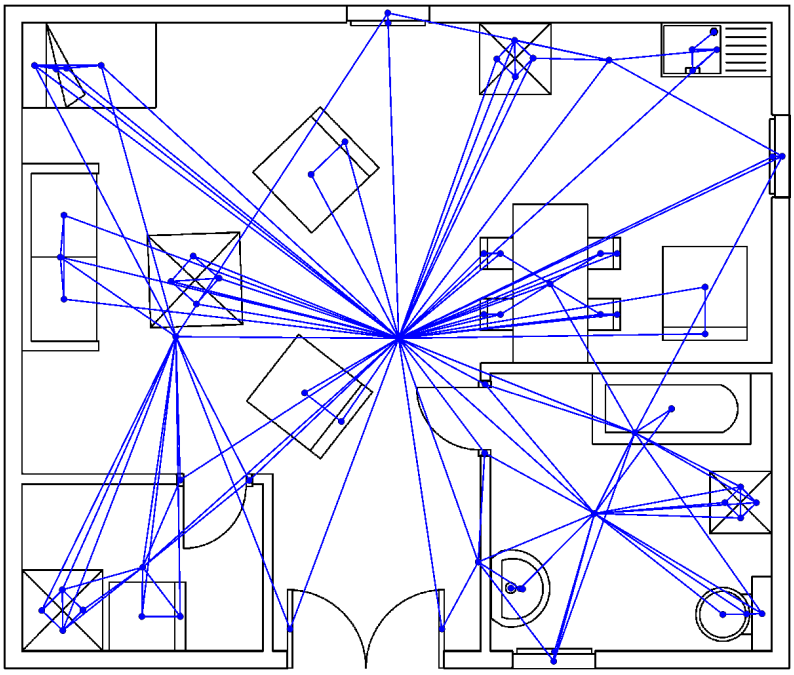


FIGURE 3.2 – un exemple de graphe d'adjacence des régions d'une image de la base `floorplan-05`

malisée par la racine de leur aire totale :

$$d_e(g_i, g_j)/\sqrt{A(i) + A(j)}$$

où $d_e(g_i, g_j)$ est la distance euclidienne entre les centres de gravité des régions i et j.

Bases de données

Les images de documents que nous avons utilisées pour construire notre base de graphes proviennent du 5^{eme} plan de la base `floorplan` décrite dans (Delalandre et al., 2010). Il s'agit d'images représentant plusieurs dispositions de différents symboles sur des modèles de plans architecturaux. Dans la figure 3.2, nous présentons un exemple de plan dont nous avons extrait le graphe d'adjacence des régions (pour des questions de lisibilité de la figure, les attributs du graphe n'apparaissent pas). Les graphes représentant les documents

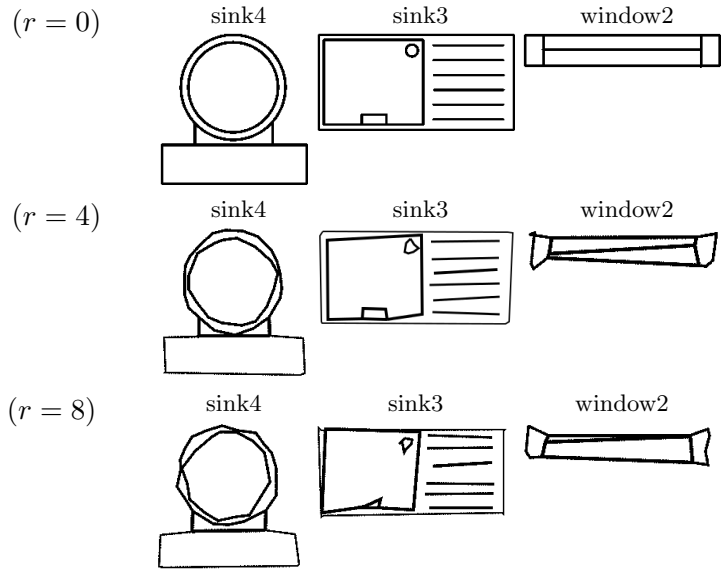


FIGURE 3.3 – Des exemple de symboles pour $r = 0$ (sans dégradation) $r = 4$ et $r = 8$

contiennent en moyenne 121 nœuds et 525 arcs. L'objectif associé à cette base correspond à la localisation des instances de 11 modèles de symboles sur lesquels nous avons appliqué quelques distorsions. Sur la figure 3.3, nous illustrons des exemples de symboles modèles. Ces derniers contiennent en moyenne 4 nœuds et 7 arcs.

Dans la base originale, les images de documents ainsi que les occurrences de symboles modèles ne diffèrent que par la taille et l'orientation. De telles modifications impactent essentiellement les attributs des nœuds et des arcs dans le graphe correspondant, même si la squeletisation produit parfois quelques artefacts. Afin de mieux élaluer l'approche MCSM que nous proposons, nous avons déformé de façon synthétique les modèles de symboles au niveau de l'image de façon à ce que ces modifications dans l'image aient un impact sur la topologie du graphe. Ce choix de distordre les symboles modèles et non les plans nous permet de garantir que la vérité terrain fournie par M. Delalandre reste valable. Pour générer ces distorsions, nous commençons par vectoriser l'image, en utilisant les algorithmes décrits dans (Di Baja and Thiel, 1996) et (Wall and Danielsson, 1984). Puis, nous appliquons un bruit vectoriel en

utilisant l'approche proposée dans (Dutta et al., 2013b) avec un paramètre r contrôlant la déformation. Enfin, nous regénérons l'image sous forme de bitmap. Quelques exemples de déformation en faisant varier la valeur du paramètre r sont illustrés sur la figure 3.3.

Protocole expérimental

La base de documents décrite ci-dessus, initialement composée de 100 documents, a été scindée en deux parties de même taille. Une moitié est utilisée pour paramétrer le système et la seconde moitié pour l'évaluer. Ainsi, pour l'évaluation, nous obtenons pour un degré de bruit donné un ensemble composé de $11 \times 50 = 550$ requêtes, où une requête correspond à une paire (graphe modèle, graphe cible). Afin de garantir la pertinence des résultats retournés, le processus de génération de bruit a été répliqué 10 fois pour obtenir $10 \times 550 = 5500$ requêtes.

Pour résoudre le problème MSCM avec l'approche décrite dans ce chapitre, il est nécessaire de définir les coûts d'édition. Pour le coût de substitution $c(i \to k)$ et $c(ij \to kl)$, nous utilisons une distance euclidienne pondérée sur les attributs des nœuds et arcs comme suit :

- $c(i \to k) = \sqrt{\sum_{n=1}^{26} {w_n}^2 (\mu(i)_n - \mu(j)_n)^2}$
- $c(ij \to kl) = \sqrt{\sum_{n=1}^{2} {\alpha_n}^2 (\nu(ij)_n - \nu(kl)_n)^2}$

Pour les expériences décrites dans cette section, nous avons défini les valeurs des w_n et α selon les distributions de la différence absolue des valeurs des attributs entre les requêtes et les cibles, dans la base d'apprentissage de 50 documents. Une fois ces valeurs définies, nous testons différentes valeurs de coût de suppression : $C = 5, 10, 20, 40, 80$.

Étant donné qu'il peut y avoir plusieurs instances d'un même symbole dans un document donné, la stratégie présentée en 3.3.4 est utilisée pour trouver ces instances multiples. Cependant, cette stratégie nécessite de définir un critère d'arrêt pour éviter au maximum les fausses détections. Dans nos expériences, nous avons appris des seuils th_i (où i représente la classe du symbole) appliqués

sur la fonction objectif de notre programme linéaire en nombres binaires. Ces seuils sont naturellement appris sur la base d'apprentissage, pour chaque classe de symbole. En utilisant ces valeurs, la recherche des sous-graphes est interrompue dès que la valeur de la fonction objectif dépasse le valeur du seuil.

Dans notre évaluation, nous comparons les résultats obtenus par notre méthode avec les informations de la vérité terrain fournies au niveau des images de plans dans la base. La comparaison consiste à vérifier si un nœud apparié dans le graphe cible appartient bien à une instance du symbole modèle. Dans la base, les symboles sont délimités par des rectangles. Dans nos expériences, nous considérons qu'un symbole est retrouvé si au moins la moitié de ses nœuds ont été appariés avec des régions de symbole. En utilisant ce critère, nous pouvons calculer les métriques classiques de recherche d'information (précision, rappel, F1-score) afin de pouvoir caractériser la performance du système de détection.

Dans toutes les expériences, nous confrontons les résultats obtenus avec notre système MCSM avec ceux de la technique proposée dans (Le Bodic et al., 2012) qui n'est tolérante qu'à la substitution dans l'isomorphisme des sous-graphes (Substitution Tolerant Only Subgraph Matching - STOSM). Notons que dans (Le Bodic et al., 2012), les résultats sont comparés avec une autre approche de la littérature.

Dans un souci de reproductibilité des résultat, les deux approches sont intégrées dans l'outil GEM++ qui est disponible à l'URL [1] et détaillé dans (Hammami et al., 2015) et (Lerouge et al., 2015). La même configuration a été utilisée pour l'évaluation des 2 approches. Les graphes utilisés sont également disponibles à cette même URL.

Résultats obtenus

Dans le tableau 3.1, nous présentons les résultats obtenus en utilisant les 2 approches MCSM et STOSM pour $r = 0, 4, 8$. Dans le cas du MCSM, nous comparons aussi les résultats obtenus avec 5 valeurs différentes de C.

1. `http://litis-ilpiso.univ-rouen.fr/ILPIso/`

TABLE 3.1 – Valeur moyenne du score F1 sur le taux d'appariement des 5500 requêtes de la base de test

Method	MCSM					STOSM
r \ C	5	10	20	40	80	-
0	0.95	0.99	1.00	1.00	1.00	0.99
4	0.75	0.92	0.94	0.94	0.94	0.93
8	0.58	0.80	0.84	0.85	0.84	-

Nous remarquons que la valeur du score F1 diminue quand la valeur du paramètre r augmente. Cependant, en comparant les 2 approches, bien que nous obtenions les mêmes performances de détection quand $r = 0$ et $r = 4$, l'approche STOSM atteint sa limite quand $r = 8$. En effet, nous n'arrivons pas à obtenir des résultats sur toutes les requêtes (valeur manquante dans le tableau 3.1). Pour une analyse plus fine, nous détaillons les résultats de précision et de rappel par classe quand $C = 40$ et $r = 8$ dans le tableau 3.2. Les résultats obtenus sur la classe *sink*3 illustrent le fait que notre approche MCSM est plus performante que la méthode STOSM car des distorsions des nœuds et arcs apparaissent souvent sur ce type de symbole. Ce phénomène est illustré dans l'exemple de la figure 3.4 pour un cas de *sink*3. En bruitant l'image, nous produisons des faux nœuds et arcs dans le graphe modèle. Par conséquent, nous obtenons de faux appariements lorsqu'on utilise l'approche STOSM, alors que lorsqu'on autorise des opérations de suppression, nous parvenons avec notre approche MCSM à retrouver l'objet correct. Dans la figure 3.5, nous présentons un autre exemple de recherche de symbole de type *window*2. Dans ce cas nous obtenons un mauvais appariement dû à un coût très bas de suppression.

Dans le tableau 3.3, nous comparons les temps de calcul des deux approches sur la base de test. Les résultats montrent qu'en utilisant l'approche MCSM avec un coût de suppression $C = 40$, la recherche est 10 fois plus rapide que l'approche STOSM, même en cas d'absence de bruit ($r = 0$). Ceci s'explique par l'algorithme utilisé par les solveurs du programme linéaire en

TABLE 3.2 – Précision et rappel de la base de test détaillée par classe , pour $r = 8$ et $C = 40$

	MCSM ($C = 40$)			STOSM		
symbol	rec.	prec.	F1	rec.	prec.	F1
bed	0.90	1	0.95	0.81	1	0.89
sink1	0.90	1	0.95	0.90	1	0.95
sink3	0.74	1	0.85	0.10	1	0.18
sink4	0.10	0.02	0.03	0	—	—
sofa1	0.98	0.99	0.98	0.98	0.99	0.98
sofa2	0.99	1	0.99	1	1	1
table1	0.97	1	0.98	0.79	1	0.88
table2	1	1	1	1	1	1
tub	1	1	1	1	1	1
window1	0.50	1	0.67	0.50	1	0.67
window2	0.90	1	0.95	0.90	1	0.95
overall	0.82	0.91	0.85	0.72	—	—

nombres binaires (branch-and-cut), et en particulier la borne inférieure qu'ils utilisent (programme linéaire avec relaxation continue) pour l'initialisation de la recherche arborescente.

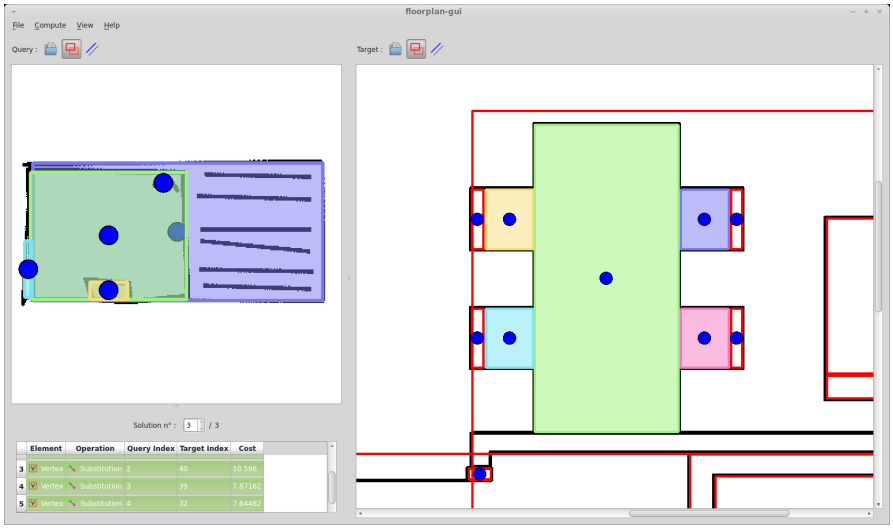
TABLE 3.3 – Temps de calcul moyen dans le cas où une instance est correctement détectée, en secondes

Method	MCSM					STOSM
r \ C	5	10	20	40	80	-
0	0.11	0.13	0.17	0.41	3.21	3.09
4	0.16	0.13	0.19	0.52	4.72	4.92
8	0.22	0.18	0.27	1.14	10.05	9.26

3.4.2 Graphes synthétiques

Afin d'évaluer les capacités de passage à l'échelle de la nouvelle formulation, nous avons mené des expériences sur la base de données synthétiques utilisée dans Le Bodic et al. (2012). Dans cette base, le nombre de nœuds dans le graphe modèle ($n_{\mathcal{S}}$) et dans les graphes cibles ($n_{\mathcal{G}}$) varient ainsi que la densité d'arcs, contrôlée par le paramètre p. Pour chaque configuration de $\{n_{\mathcal{S}}, n_{\mathcal{G}}, p\}$, 5 graphes modèles sont recherchés dans 5 graphes cibles, ce qui correspond

STOSM



MCSM

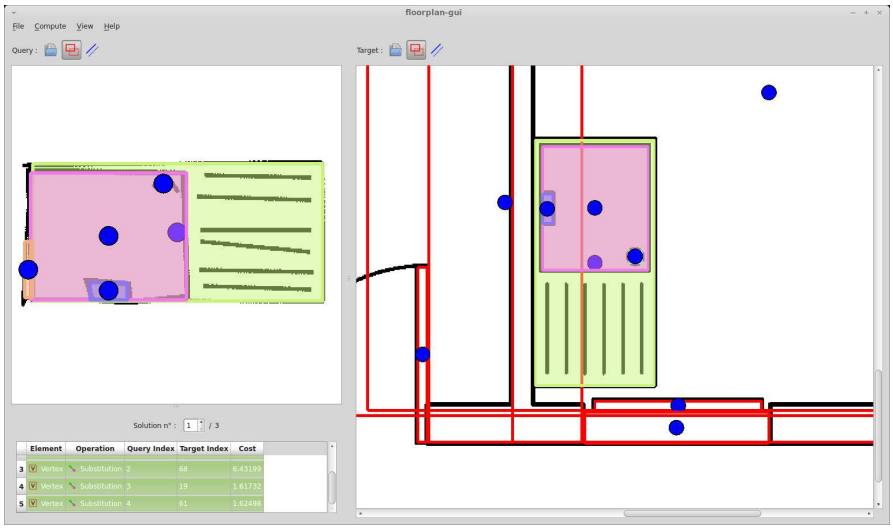


FIGURE 3.4 – Comparaison des résultats d'une requête de `sink3` sur un plan de sol donné en utilisant MCSM et STOSM. la requête se trouve à gauche de la cible est à droite. Dans la requête le nœud se trouvant en bas gauche représente un bruit. L'appariement de chaque paire de nœud est représenté par une couleur différente.

à 25 requêtes. Selon le protocole de génération des données (voir Le Bodic et al. (2012) pour plus de détails), le graphe cible contient le graphe modèle seulement pour 5 requêtes parmi 25. Les temps obtenus sont décrits dans le tableau 3.4. Dans ces expériences, la recherche est stoppée quand le temps de traitement atteint 300 secondes.

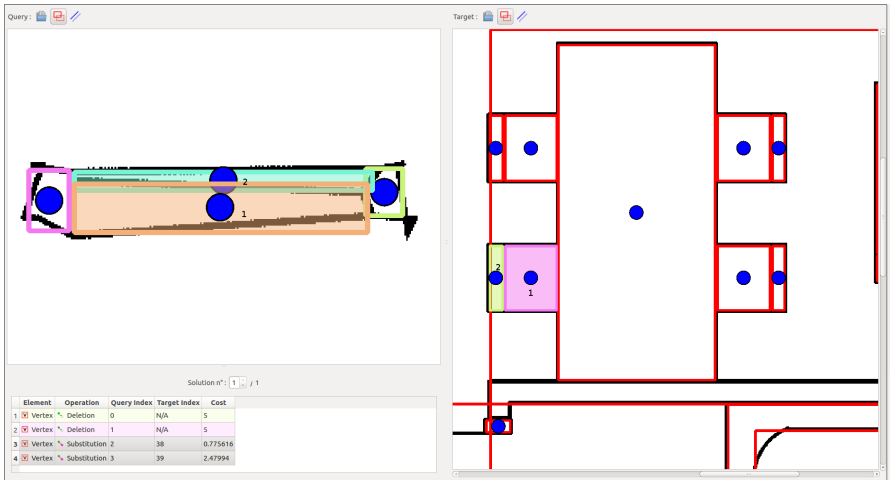


FIGURE 3.5 – Exemple de résultat de la recherche d'un symbole de type `window2` quand on autorise la suppression d'un mauvais appariemment.

Ces résultats montrent que les requêtes sur de grands graphes peuvent aussi être résolues en utilisant notre nouvelle formulation mais montrent également ses limites en particulier dans les graphes denses.

TABLE 3.4 – Temps de calcul médian pour les différentes configurations dans la base synthétique, en secondes

p	$n_{\mathcal{G}} = \lvert V_{\mathcal{G}} \rvert$	$n_{\mathcal{S}} = \lvert V_{\mathcal{S}} \rvert$		
		10	25	50
0.01	50	0.02	0.04	0.21
	100	0.03	0.15	0.27
	250	0.13	0.31	1.23
	500	0.24	0.76	5.70
0.05	50	0.04	0.95	1.24
	100	0.16	6.25	-
	250	0.33	-	-
	500	3.10	-	-
0.1	50	0.36	-	-
	100	0.39	-	-
	250	10.82	-	-
	500	17.09	-	--

Afin de comparer le temps de calcul des MCSM et STOSM, nous avons également utilisé un sous-ensemble de la base décrite précédemment, en ne gardant que les 5 requêtes parmi les 25 requêtes pour lesquelles un isomor-

phisme de sous-graphe existe. Nous calculons le temps median pris par les 2 méthodes pour trouver la solution unique. Les résultats obtenus sont présentés dans le tableau 3.5. Ces résultats confirment ceux obtenus dans l'application de la détection de symboles. Dans la dernière cellule du tableau, nous obtenons un échec de mémoire dans les 2 cas.

TABLE 3.5 – Temps de calcul médian des MCSM (valeur à gauche) et STOSM (valeur à droite) dans les sous-ensembles de la base synthétique, en secondes

p	$n_{\mathcal{G}}$	$n_{\mathcal{S}}$		
		10	25	50
0.01	50	0.02 / 0.03	0.04 / 0.07	0.17 / 0.19
	100	0.03 / 0.05	0.15 / 0.15	0.25 / 0.39
	250	0.12 / 0.16	0.29 / 0.29	0.93 / 1.03
	500	0.23 / 0.26	0.76 / 0.75	3.61 / 3.08
0.05	50	0.04 / 0.05	0.21 / 0.24	0.84 / 0.99
	100	0.16 / 0.17	0.50 / 0.56	2.88 / 3.37
	250	0.31 / 0.38	3.88 / 5.21	16.69 / 21.54
	500	2.93 / 3.01	24.34 / 46.12	138.1 / 276.4
0.1	50	0.17 / 0.19	0.51 / 0.65	3.43 / 5.11
	100	0.25 / 0.28	1.87 / 2.42	8.96 / 13.10
	250	2.74 / 3.28	15.58 / 25.56	106.8 / 157.6
	500	11.89 / 14.27	171.2 / 320.1	- / -

3.4.3 Base *Itesoft*

Dans cette sous-section, nous appliquons l'approche de recherche de sous-graphe proposée dans ce chapitre sur les graphes issus d'images de formulaires décrits dans le chapitre précédent de la thèse. Les expériences visent ainsi à évaluer une tâche de localisation d'information dans des images de formulaires. Le protocole expérimental proposé simule ce contexte applicatif. Ce protocole est détaillé dans le paragraphe qui suit, avant de présenter et commenter les résultats.

Données et protocole expérimental

La base sur laquelle nous opérons notre évaluation est composée de 130 images de formulaires commerciaux ou administratifs répartis en 8 classes

illustrées sur la figure 2.15. L'effectif de chaque classe est donné entre parenthèse en en-tête du tableau 3.6. Comme indiqué précédemment, la classe de chaque document est supposée connue au moment de la recherche puisqu'elle a été déterminée par un classifieur. De ce fait, les expériences sont menées indépendamment dans chaque classe.

Pour chaque formulaire de chaque classe, 36 graphes de visibilité différents sont extraits. Ces 36 graphes correspondent à autant de paramétrages différents afin de déterminer l'impact de chacun des paramètres sur les performances de localisation d'information. Les paramètres étudiés sont :

- l'impact du prétraitement par filtrage $EPSF$;
- l'impact du prétraitement par inpainting ;
- le paramètre k de l'algorithme des k-moyennes ;
- l'impact de l'espace colorimétrique de travail : RVB, $YCbCr$ ou $CIELab$.

Pour simuler la définition des régions d'intérêt par l'utilisateur, des graphes requêtes sont générés aléatoirement de la façon suivante. Dans un premier temps, le document d'apprentissage est sélectionné aléatoirement parmi les documents de la classe. Un nœud germe représentant la région d'intérêt [2] est alors sélectionné également aléatoirement au sein de la représentation structurelle du document d'apprentisage. L'ensemble des nœuds est alors complété avec les nœuds directement connectés au nœud germe. Enfin, le graphe requête est construit comme étant le sous-graphe de la représentation structurelle du document d'apprentissage induit par l'ensemble des nœuds sélectionnés. Les instances du graphe requête sont ensuite recherchées dans les représentations structurelles des documents autres que celui sélectionné pour l'apprentissage. Afin d'évaluer les capacités en généralisation de l'approche proposée, ce processus est reproduit 50 fois au sein de la classe de sorte que 50 graphes requêtes soient générés. Au total, cela nous a conduit à opérer 6100 recherches d'isomorphisme de sous-graphe pour chacune des 36 configurations d'extraction.

2. Pour des question liées à l'évaluation, la région d'intérêt est nécessairement un nœud du graphe correspondant à un rectangle coloré. Cependant, l'approche peut être généralisée à n'importe quelle zone de l'image

Pour chaque recherche, l'algorithme de recherche d'isomorphisme décrit dans ce chapitre fournit l'ensemble des mises en correspondances entre les nœuds et entre les arcs et le coût associé à cette mise en correspondance.

L'évaluation de notre approche requiert de vérifier que les nœuds ont été correctement associés lors de la recherche d'isomorphisme de sous-graphe. Nous examinons donc la sortie proposée par l'algorithme d'isomorphisme et notamment la liste des nœuds effectivement associés S_{ij} et celle des nœuds créés C_i. Nous calculons alors la distance de Jaccard $Jacc(Z_i, Z_j)$ entre les paires de zones Z_i et Z_j des associations S_{ij}. Nous considérons qu'une association est correcte si la distance $Jacc(Z_i, Z_j)$ est supérieure à un paramètre α que nous avons fixé à la valeur 0,1. Les associations considérées correctes sont notées M_{ij}. Nous calculons ensuite le score de mise en correspondance de la requête visant à rechercher le sous-graphe q_i et le graphe t_j comme étant le taux de mises en correspondance correctes selon la formule donnée par l'équation (3.16).

$$match(q_i, t_i) = \frac{|M_{ij}|}{|S_{ij}| + |C_i|} \tag{3.16}$$

Enfin, pour chaque classe, nous calculons la performance de la recherche de zone selon l'équation (3.17). Dans cette formule, n correspond au nombre de documents dans la classe. Nous considérons que le sous-graphe a été correctement retrouvé si $match$ est supérieur à un paramètre β dont nous avons fixé la valeur à 0,1 dans nos expériences.

$$Perf = \frac{\sum_{i=1}^{50} \sum_{j=1}^{n-1} \mathbf{1}_{match(q_i,t_j)>\beta}}{50 * (n-1)} \tag{3.17}$$

Résultats

Les résultats obtenus sont donnés dans le tableau 3.6. Ce tableau présente les performances obtenues pour chacune des classes et pour chaque configuration de traitement, une configuration de traitement étant définie par le fait que le filtre EPSF ait été appliqué ou non, que l'inpainting ait été appliqué

ou non, le choix de l'espace couleur (RVB, $YCbCr$ ou $CIELab$) et la valeur de k utilisée pour l'algorithme des k-moyennes ($k \in \{2, 3, 4\}$). Ce paramétrage définit 36 configurations différentes. La dernière colonne du tableau 3.6 indique la valeur de $Perf$ sur la globalité de la base.

Pour aider à la lecture de ce tableau, la meilleure performance sur l'ensemble des configurations est indiquée en gras. On peut observer que la meilleure performance est toujours obtenue avec une configuration qui inclut le traitement d'inpainting. Nous remarquons également que la quantification en deux couches couleurs ($k = 2$) conduit à la meilleure performance dans la plupart des cas. Etant données ces observations, nous émettons l'hypothèse que, dès lors que la couche texte a été éliminée, la séparation en deux couches couleurs conduit aux meilleures performances. Il est en revanche difficile de tirer des conclusions quant à la nécessité d'appliquer ou non le filtrage EPSF ou sur l'espace colorimétrique le plus adapté. En effet, les meilleures performances sont obtenues avec des configurations différentes de ces deux derniers paramètres selon les classes. Nous emettons l'hypothèse que les configurations conduisant aux meilleurs performances diffèrent selon les classes car les classes diffèrent entre elles notamment du point de vue du nombre et de la nature des couleurs qui les composent et sur la densité du texte. Ce phénomène illustre le besoin de pouvoir déterminer automatiquement la configuration la plus adaptée pour chaque classe de documents.

Au global, la performance la plus adaptée sur la globalité de la base permet de retrouver la zone recherchée dans 86,9% des cas. Cependant, il existe de grandes disparités selon les classes. En effet, pour certaines classes la zone est retrouvée dans 100% des cas alors qu'elle n'est retrouvée que dans 72% des cas pour les classes les plus difficiles.

TABLE 3.6 – Résultats expérimentaux

Prétraitement		Quantification		Classes						Base
EPSF	Inpaint	K	Espace couleur	1(11)	2(31)	3(29)	4(14)	5(12)	6(11)	7(9)	8(13)	
Oui	Oui	2	RGB	58,85%	59,57%	75,44%	98,00%	99,33%	60,90%	69,33%	50,62%	69,67%
			YCbCr	**83,45%**	89,35%	92,82%	**99,28%**	98,16%	44,72%	**92,66%**	58,29%	84,84%
			Lab	76,00%	82,00%	**93,44%**	98,57%	**100,00%**	45,27%	90,00%	45,98%	81,25%
		3	RGB	61,45%	65,54%	59,65%	90,29%	**100,00%**	78,54%	88,44%	42,92%	70,09%
			YCbCr	79,81%	47,74%	72,13%	90,28%	**100,00%**	44,18%	43,77%	58,39%	65,77%
			Lab	68,54%	66,83%	84,96%	91,71%	97,16%	38,00%	56,22%	31,29%	69,70%
		4	RGB	70,90%	58,70%	66,62%	90,42%	**100,00%**	39,63%	61,11%	61,07%	67,46%
			YCbCr	52,72%	55,87%	60,82%	84,85%	98,16%	37,63%	47,33%	49,53%	60,86%
			Lab	49,63%	32,38%	69,51%	88,71%	98,16%	45,81%	48,44%	38%	57,07%
	Non	2	RGB	37,09%	25,03%	53,86%	62,64%	70,16%	21,09%	34,88%	30,15%	41,34%
			YCbCr	65,45%	59,87%	82,41%	92,00%	92,66%	49,09%	83,55%	37,07%	70,21%
			Lab	76,00%	47,41%	78,27%	86,28%	93,00%	27,09%	60,22%	36,30%	63,16%
		3	RGB	74,00%	51,54%	43,93%	90,14%	74,67%	45,09%	65,11%	50,30%	58,13%
			YCbCr	70,72%	62,96%	73,79%	87,71%	82,66%	40,36%	77,55%	51,07%	68,38%
			Lab	62,36%	62,32%	72,48%	87,14%	85,00%	34,90%	59,77%	41,07%	64,64%
		4	RGB	56,00%	61,93%	64,55%	87,85%	72,33%	38,72%	69,11%	24,92%	60,53%
			YCbCr	43,63%	52,00%	58,75%	92,14%	77,50%	36,90%	60,66%	36,46%	57,24%
			Lab	56,36%	59,80%	64,55%	85,85%	75,16%	24,18%	65,33%	36,92%	59,84%
Non	Oui	2	RGB	59,09%	48,95%	59,37%	87,71%	99,16%	61,09%	62,88%	39,28%	61,94%
			YCbCr	80,90%	**90,38%**	89,58%	99%	**100,00%**	**83,81%**	88,66%	55,21%	**86,90%**
			Lab	77,63%	74,96%	83,37%	90,57%	**100,00%**	52,36%	85,77%	47,82%	77,08%
		3	RGB	55,45%	65,03%	59,65%	87,85%	98,33%	83,63%	86,44%	29,09%	68,00%
			YCbCr	74,70%	51,48%	85,37%	92,42%	**100,00%**	38,90%	36,44%	51,53%	67,78%
			Lab	50,00%	55,03%	79,93%	89,14%	95,50%	40,36%	50,88%	32,92%	63,83%
		4	RGB	79,45%	59,41%	70,96%	86,42%	95,16%	41,81%	48,44%	**72,00%**	68,93%
			YCbCr	51,27%	41,35%	67,31%	93,42%	99,66%	42,36%	38,66%	43,84%	59,10%
			Lab	56,18%	34,77%	65,44%	90,28%	96,50%	34,90%	44,88%	31,07%	55,43%
	Non	2	RGB	43,09%	22,83%	49,87%	55,31%	69,33%	31,81%	47,33%	23,84%	40,89%
			YCbCr	63,63%	60,83%	78,48%	82,42%	89,83%	35,09%	69,55%	30,46%	65,35%
			Lab	58,00%	52,83%	64,82%	84,57%	84,66%	18,90%	64,66%	40,15%	58,95%
		3	RGB	63,63%	48,12%	59,51%	90,14%	78,16%	70,18%	65,55%	41,36%	61,53%
			YCbCr	67,81%	72,45%	76,48%	85,85%	82,33%	40,18%	66,88%	46,46%	69,58%
			Lab	53,09%	56,83%	67,79%	85,00%	80,50%	46,36%	42,66%	46,00%	61,16%
		4	RGB	49,27%	59,74%	66,82%	85,00%	67,66%	52,54%	53,77%	24,30%	59,20%
			YCbCr	56,00%	53,74%	62,27%	83,28%	79,83%	36,90%	59,11%	46,32%	59,49%
			Lab	52,90%	57,87%	50,55%	81,28%	74,83%	40,90%	59,33%	40,00%	56,66%

Les figures 3.4.3, 3.4.3 et 3.4.3 donnent quelques exemples de localisation de zones. Ces figures présentent l'interface utilisateur sur laquelle le document de gauche correspond au document requête. Sur ce document requête, le cadre vert indique la zone recherchée, les zones bleues sont celles appartenant au contexte de voisinage. Le document de droite est le document cible. Sur ce document, les cadres rouges indiquent les zones de couleur homogènes qui ont été détectées. Les zones bleues correspondent à celles qui ont été mises en correspondance par l'algorithme de recherche d'isomorphisme de sous-graphe avec celle modélisant le contexte de voisinage de la zone recherchée. Enfin, la zone verte est la zone correspondant à la zone recherchée.

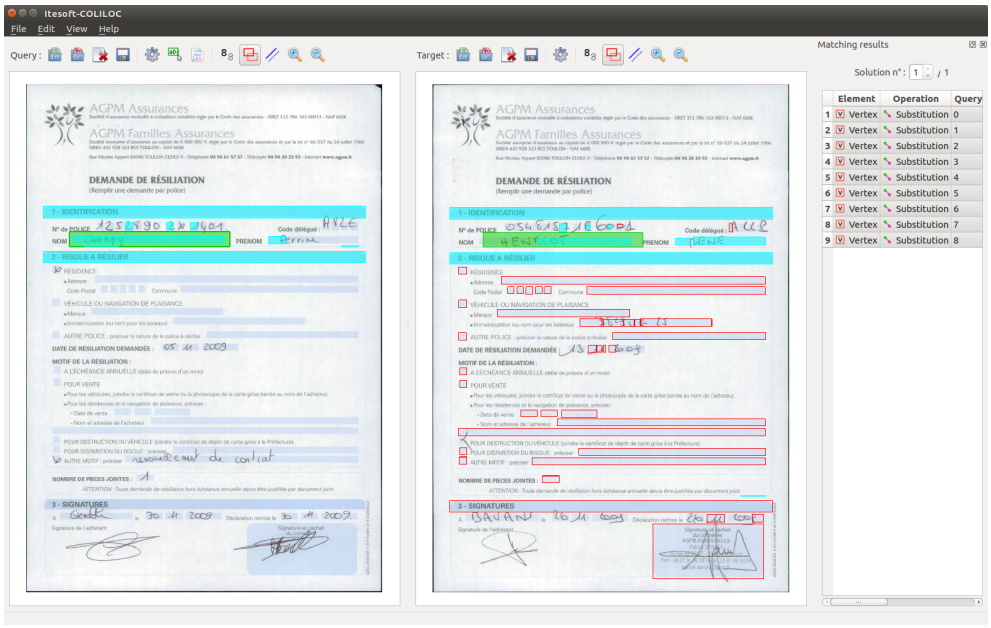


FIGURE 3.6 – Exemple de localisation

3.5 Conclusion

Dans ce chapitre, nous avons abordé la problématique de la localisation d'information dans un document. En nous appuyant sur une représentation structurelle du document cible et d'un graphe décrivant la région d'intérêt,

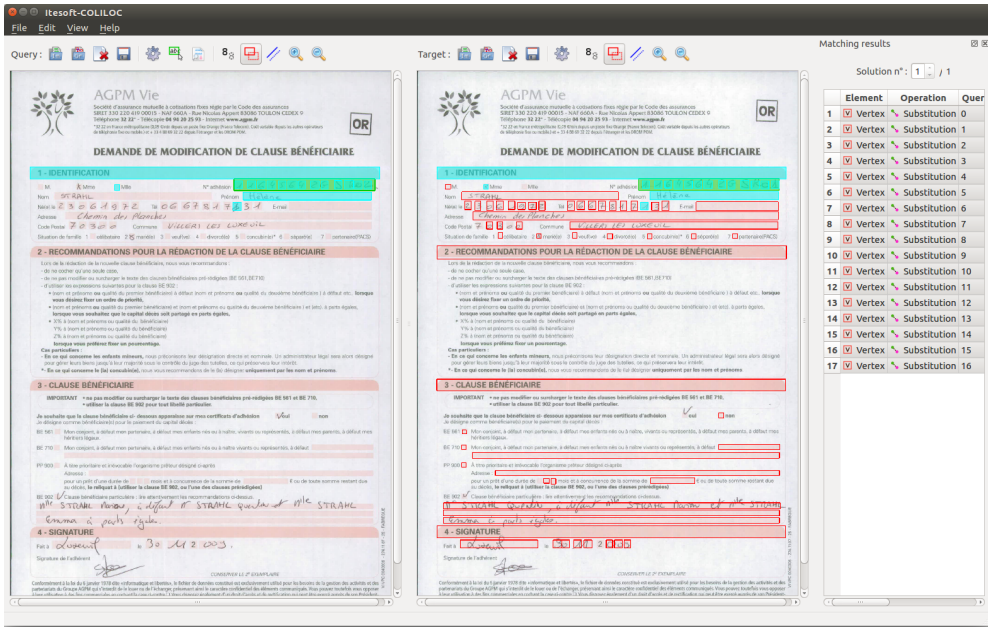


FIGURE 3.7 – Exemple de localisation

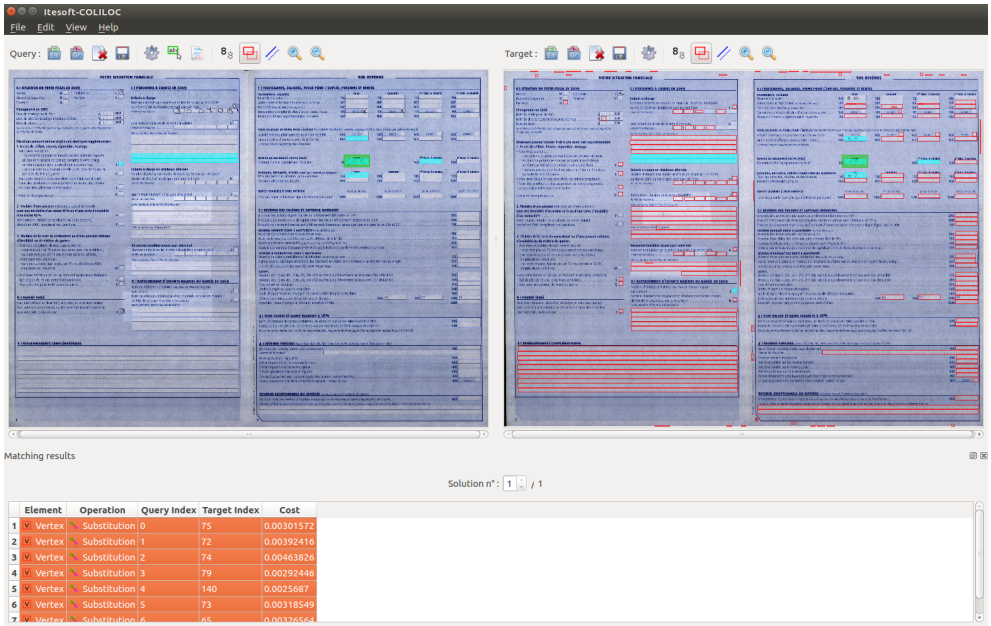


FIGURE 3.8 – Exemple de localisation

nous avons proposé une approche originale de recherche des occurrences d'un graphe modèle dans un graphe cible. L'approche repose sur une formulation

linéaire en nombres binaires, dont l'objectif est de minimiser un coût d'appariement, tout en tolérant la disparition de nœuds et d'arcs dans le graphe cible.

L'approche proposée a été évaluée sur 3 bases de graphes. La première évaluation, sur une problématique de recherche de symbole dans documents graphiques, a permis de montrer l'apport des opérations de suppression. L'approche permet en effet de retrouver des occurrences de symboles distordus que les approches de la littérature ne permettent pas de trouver. La seconde évaluation a montré que cette prise en compte de modification de la topologie n'entraîne pas de surcoût sur les temps de calcul, par rapport à une approche qui ne tolère que des modification d'attributs. Enfin, la dernière évaluation, réalisée sur les documents de l'application ITESOFT, a confirmé les bonnes propriétés de l'approche proposée. Les résultats que nous avons obtenus montrent en effet que notre système de localisation est efficace pour retrouver des zones informatives. De plus, nous avons montré qu'il est possible d'optimiser le système si nous pouvons attribuer à chaque catégorie de document un meilleur scénario d'extraction des éléments immuables. Dans le chapitre suivant, nous nous intéressons à cette problématique afin de pouvoir améliorer la performance de la localisation de l'information.

Chapitre 4

Vers un système adaptatif

Sommaire

4.1 Introduction

Dans le second chapitre de ce mémoire, nous nous sommes intéressés à la représentation du document sous forme de graphe composé d'éléments immuables. Notre approche d'extraction de ces éléments immuables est composée de trois grandes étapes : un pré-traitement de l'image afin d'éliminer le bruit, une quantification de couleur pour retrouver les régions de couleur homogène et une étape de filtrage des régions afin de ne garder que les éléments pertinents. Nous avons montré qu'à chacune de ces étapes, il était nécessaire de définir la valeur de plusieurs paramètres pour configurer le système.

Dans le troisième chapitre, dédié à l'application de l'isomorphisme de sous-graphes pour la localisation de l'information, les expérimentations ont montré que la meilleure performance en localisation était obtenue, pour chaque classe, avec un paramétrage différent. La détermination d'un paramétrage optimal de la chaîne d'extraction de graphe est donc une problématique importante du point de vue des performances en localisation au sein de chaque classe. Ce contexte n'est bien évidemment pas propre à notre application et se retrouve dans la plupart des problèmes d'analyse d'images de documents.

Dans ce chapitre, nous présentons une contribution qui offre un premier élément de réponse à cette problématique de la détermination adaptative du paramétrage optimal d'un système en présence d'un espace de paramètres potentiellement vaste. L'objectif est de déterminer, pour chacune des classes, le paramétrage qui conduira aux meilleures performances en localisation. Dans ce cadre, nous sommes confrontés à deux difficultés. D'une part, les informations de vérité terrain relative aux documents traités ne sont pas accessibles dans les conditions d'utilisation réelles (en production). La seule information de vérité terrain connue est la position de la zone informative sur le document d'apprentissage. D'autre part, lors du cycle de vie du système, ce dernier n'a accès qu'à une connaissance très partielle de la classe de documents, à travers très peu d'exemples. Même si le nombre de documents de la classe vus par le système va croître, les premiers documents doivent également être traités avec

un paramétrage aussi bon que possible, étant données les connaissances accessibles à ce moment-là. Le processus d'optimisation doit donc opérer dans un contexte évolutif dans lequel la performance en localisation d'un paramétrage n'est pas calculable. La difficulté est accentuée par la dimension de l'espace des paramètres qui rend son exploration exhaustive impossible.

Pour contourner ces difficultés, nous proposons dans cette thèse d'optimiser le système par un algorithme génétique reposant sur un critère différent de l'objectif initial de l'application, pour lequel nous avons validé la corrélation avec l'objectif initial. Ce nouveau critère est lui même affiné au fur et à mesure de la vie du système puisque mesuré sur un nombre croissant de documents.

Le chapitre est organisé de la manière suivante. Dans un premier temps, après avoir détaillé le cycle de vie du système de localisation des zones d'intérêt en soulignant les difficultés auxquelles nous sommes confrontés, nous formalisons de façon générale le problème en insistant sur les verrous scientifiques. Dès lors, nous présentons la solution que nous proposons, basée sur l'utilisation d'algorithmes génétiques. Les choix relatifs à la configuration de l'algorithme génétique adopté sont justifiés au regard d'un état de l'art. Nous présentons ensuite les expériences réalisées dans le cadre de ce travail exploratoire et les résultats observés qui nous ont permis de valider cette première approche. Après une analyse de ces premiers résultats, nous suggérons quelques pistes d'amélioration.

4.2 Position du problème

4.2.1 Description du cycle de vie du système de localisation des zones d'intérêt

Le système de localisation des zones d'intérêt est composé principalement de deux processus : un processus en-ligne et un processus hors ligne tel que représenté sur la figure 4.1.

Etant donné un paramétrage du système d'extraction de la représentation

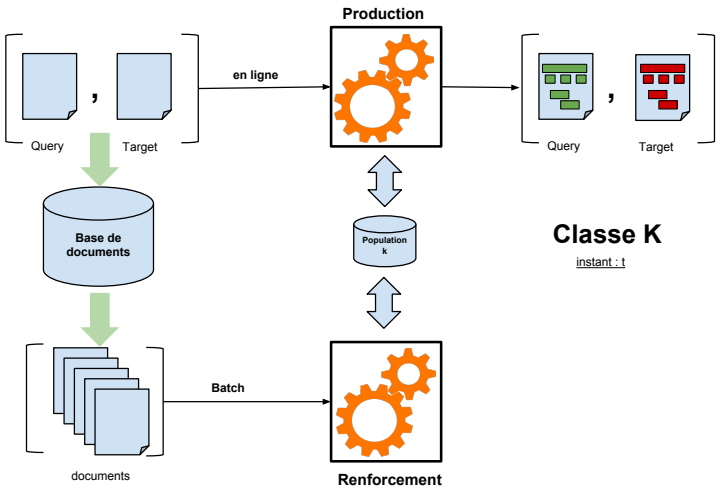


FIGURE 4.1 – Illustration des deux processus en-ligne et hors-ligne

structurelle, un document d'apprentissage sur lequel l'utilisateur a spécifié la zone à localiser, et un document à traiter, le processus en ligne va (i) extraire la representation structurelle pour chacun des deux documents selon le paramétrage, (ii) déterminer le sous-graphe exprimant le voisinage contextuel de la zone à extraire, et (iii) trouver le sous-graphe le plus proche au sein de la représentation structurelle du document à traiter par isomorphisme de sous-graphe. Le processus en ligne fait l'objet des chapitres 2 et 3 de ce manuscrit.

Le processus hors-ligne a pour objectif de déterminer, pour la classe de documents considérée, le paramétrage qui offrira les performances optimales du point de vue de la localisation des zones. Ceci doit s'opérer sans avoir connaissance des informations de vérité terrain qui permettraient de mesure la performance et en n'ayant qu'une connaissance partielle de la classe. En effet, seuls sont connus les documents de la classe ayant été traités par le système à ce moment-là. Le système n'a bien évidemment pas connaissance des documents de la classe qu'il aura à traiter dans le futur.

Le cycle de vie du système commencera avec une vue très restreinte de la classe, à savoir le seul document d'apprentissage utilisé par l'utilisateur pour indiquer la zone à rechercher et le premier document à traiter. Dès lors, une alternance des processus en-ligne et hors ligne, va, d'une part, traiter

un nombre croissant de documents dans la classe, et d'autre part, adapter le paramétrage en exploitant la connaissance de plus en plus complète de la classe, à travers le nombre croissant d'instances vues.

4.2.2 Formalisation du problème

Le problème que nous cherchons à résoudre est un problème d'optimisation donné par l'équation 4.1. À un instant t_i donné, correspondant au lancement du processus hors ligne, la valeur à optimiser est le taux de bonne localisation $T_{P,t>t_i}$ pour les documents de la classe que le système aura à traiter postérieurement à t_i, sur l'ensemble des paramètrages P. L'espace de paramètres dans lequel se situe P est multi-dimensionnel et potentiellement vaste. Cet espace peut être constitué de paramètres de différentes natures, à savoir binaire, discrète, entière et/ou réelle.

$$P_{opt,t>t_i} = \underset{P}{\operatorname{argmax}}\, T_{P,t>t_i} \tag{4.1}$$

Puisqu'à l'instant t_i, les documents qui se présenteront au système postérieurement à t_i sont inconnus, la résolution du problème précédent est impossible. En nous basant sur l'hypothèse que les documents sont extraits d'une même distribution, qu'ils soient vus avant ou après t_i, nous allons estimer $T_{P,t>t_i}$ par $T_{P,t\leq t_i}$, le taux de bonne localisation sur les documents vus jusqu'à l'instant t_i.

Par ailleurs, on peut supposer que $T_{P,t>t_i}$ est indépendant de t_i et peut être confondu avec la mesure $T_{P,c}$, taux de bonne localisation sur la globalité de la classe c.

$$\widehat{P_{opt,t>t_i}} = \underset{P}{\operatorname{argmax}}\, \widehat{T_{P,c}} = \underset{P}{\operatorname{argmax}}\, T_{P,t\leq t_i} \tag{4.2}$$

$T_{P,t\leq t_i}$ est un estimateur de $T_{P,c} \approx T_{P,t>t_i}$. Cet estimateur s'améliore du point de vue statistique avec le nombre de documents connus du système. Ainsi $T_{P,t\leq t_j}$ sera un meilleur estimateur que $T_{P,t\leq t_i}$ si $t_j > t_i$.

L'absence d'information de vérité terrain rend impossible la mesure de $T_{P,t\leq t_i}$. Pour contourner cette difficulté, nous allons résoudre le problème en optimisant $F_{P,t\leq t_i}$ une fonction mesurable, idéalement croissante en fonction $T_{P,t\leq t_i}$, et au moins corrélée positivement.

$$\widehat{P_{opt,t>t_i}} = \underset{P}{\operatorname{argmax}}\, F_{P,t\leq t_i} \tag{4.3}$$

Une des particularités notoires de notre problème d'optimisation provient du fait que l'objectif que nous cherchons à atteindre n'est qu'estimé. Ainsi, au sein de notre problème d'optimisation, la mesure de performance accessible pour un paramétrage P donné dépend de l'ensemble de documents sur lequel elle est estimée. Autrement dit, la qualité estimée d'un paramétrage P du système peut être amenée à évoluer au cours de son cycle de vie.

4.2.3 Présentation de l'approche proposée

Afin de résoudre le problème d'optimisation avec estimation évolutive de la fonction objectif, nous proposons une approche basée sur les algorithmes génétiques. Ce choix est en particulier motivé par le fait que les algorithmes génétiques permettent de gérer une population d'individus représentant chacun une solution potentielle. La gestion d'une population de solutions potentielles autorise la couverture simultanée de différentes régions de l'espace des solutions, permettant ainsi d'atténuer l'inconvénient lié à la découverte d'optimum locaux.

Un autre argument fort, plaidant pour l'utilisation d'une méthode d'optimisation exploitant une population de solutions potentielles vient du fait que, comme évoqué plus haut, la qualité d'une solution peut évoluer au fil de la vie du système. En gérant une population dont l'évolution globale tend vers l'optimum, tout en conservant une diversité, les remises en cause de la qualité des solutions vont se traduire par une modification de l'ordre des différentes solutions. La solution optimale à l'instant t_i ne le sera plus obligatoirement à l'instant $t_j > t_i$. Dans le même temps, le réordonnancement des solutions de

la population au regard de la nouvelle estimation de la qualité des solutions permettra d'identifier la meilleure solution à l'instant t_j dans la population.

Dans les sections qui suivent, nous rappelons d'abord le principe de fonctionnement des algorithmes génétiques et justifions les choix que nous avons opérés pour notre cas d'usage. Puis, nous détaillons le système proposé avant de décrire les expérimentations.

4.3 Les algorithmes génétiques

Les algorithmes génétiques appartiennent à la famille des algorithmes évolutionnaires dont l'objectif est de résoudre des problèmes d'optimisation. Leur mécanisme est inspiré de la théorie de la sélection naturelle et de la génétique. Ils reposent sur le principe de l'évolution d'une population d'individus dans laquelle ceux qui sont les plus adaptés à leur environnement ont davantage de chance de se reproduire et les individus issus de la reproduction reçoivent un patrimoine génétique en provenance de leurs parents.

Les algorithmes génétiques ont été introduits pour la première fois par Holland dans (Holland, 1975) où les auteurs ont démontré la robustesse de ces systèmes dans l'exploration d'espaces complexes. Reconnus pour leurs performances, les algorithmes génétiques ont alors été appliqués à de très nombreux domaines dont l'intelligence économique (Chung et al., 2003), la finance (Shin and Lee, 2002), le médical (Pareek and Patidar, 2016), l'énergie (Blaifi et al., 2016) ou l'ingénierie (Demirtas, 2011), (Kaya and Nalbantoğlu, 2016).

Dans des contextes plus proches du nôtre, les algorithmes génétiques ont été largement utilisés pour optimiser des machines d'apprentissage. On peut citer à titre d'exemple les travaux décrits dans (Lorena and De Carvalho, 2008), (Friedrichs and Igel, 2005), (Miller et al., 2003), (Chatelain et al., 2010), (Bernard et al., 2016) (Santos et al., 2015), (Frohlich et al., 2003) ou (Raveaux et al., 2011). Ils ont aussi été utilisés pour améliorer la qualité des images par optimisation des paramètres liés à des filtres déjà existants (Lee et al., 2005, Munteanu and Rosa, 2000, 2001).

4.3.1 Structure générale d'un algorithme génétique

Un algorithme génétique est un processus itératif qui gère une population d'individus (cf. Figure 4.2). Un individu encode une solution potentielle du problème d'optimisation que l'algorithme cherche à résoudre. Chaque itération renouvelle la population. La population de chaque itération de l'algorithme est appelée une génération. Chaque génération est obtenue par application des opérateurs génétiques que sont la sélection, le croisement et la mutation après que les individus de la population courante ont été évalués.

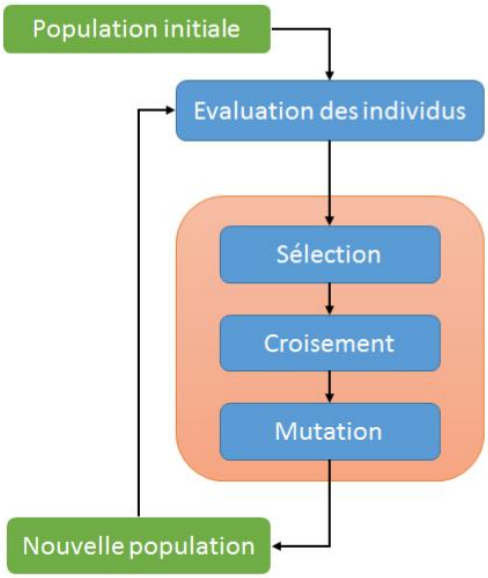


FIGURE 4.2 – Structure d'un algorithme génétique

L'évaluation d'un individu consiste à lui associer une mesure de son adaptation, c'est-à-dire de la qualité de la solution qu'il encode vis à vis du problème à résoudre.

L'opérateur de sélection vise à sélectionner les individus participant à la création de le génération suivante par reproduction. Cette sélection repose sur une probabilité d'être sélectionné, qui elle même est issue de l'évaluation des individus. L'idée sous-jacente consiste à donner aux individus les mieux adaptés de meilleures chances de participer au processus de reproduction, et donc de transmettre le code génétique des meilleurs individus au fil des générations.

La reproduction consiste ensuite à appliquer l'opérateur de croisement puis l'opérateur de mutation. L'opérateur de croisement produit un individu dont le patrimoine génétique est constitué de parties de patrimoine génétique de chacun des individus parents. L'opérateur de mutation quant à lui introduit des modifications au code génétique de l'individu issu du croisement. Il va, avec une probabilité relativement faible, modifier le code génétique des individus créés, permettant d'atteindre de nouvelles régions de l'espace de recherche. Cela vise également à maintenir une diversité dans la population et à éviter une convergence de l'ensemble de la population vers une unique région de l'espace de recherche qui conduirait à la découverte d'un optimum qui ne serait que local.

A l'issue de la reproduction, différentes politiques peuvent être adoptées pour la création de la nouvelle population. Si les premières versions d'algorithmes génétiques reposaient sur un renouvellement complet à chaque itération, il est maintenant courant de mettre en œuvre une politique élitiste qui garantit que les meilleurs individus de la génération suivante auront une mesure d'adaptation au moins aussi bonne que ceux de la génération précédente. Plusieurs implémentations sont envisageables, la plus stricte consiste à créer une population temporaire composée de l'ensemble des individus parents et des individus enfants, puis à ne conserver dans la génération suivante que les meilleurs. Cette solution est de nature à produire une convergence rapide de la population en limitant l'exploration de l'espace de recherche. Une solution intermédiaire consiste à placer dans la génération suivante uniquement les meilleurs individus de la génération courante, le reste de la génération suivante étant constitué d'individus produits par reproduction. Ainsi, nous avons la garantie que les meilleurs individus de la génération suivante auront des performances au moins aussi bonnes que ceux de la génération courante.

Finalement, il est nécessaire de définir un critère de fin du processus itératif. Les critères de fin les plus couramment utilisés sont ceux qui imposent un nombre fixé de générations ou qui examine la convergence. Dans ce dernier

cas, l'algorithme se termine lorsque la population ou une mesure de sa qualité globale n'évolue plus pendant un certains nombre d'itérations.

Dans la sous-section suivante, nous listons les choix les plus fréquents dans la littérature concernant les opérateurs mis en œuvre dans les algorithmes génétiques.

4.3.2 Opérateurs génétiques

Nous présentons dans les paragraphes qui suivent différentes options relatives à la mise en œuvre des opérateurs génétiques.

La sélection

La sélection est l'opérateur utilisé pour déterminer les individus qui participent à la génération d'une nouvelle population (Blickle and Thiele, 1995). Il s'agit d'un processus de tirage aléatoire dans la population courante qui aura tendance à privilégier les individus codant les meilleures solutions. Les solutions les moins bonnes auront ainsi tendance à être abandonnées. Le tirage aléatoire est donc affecté par la valeur de la fonction d'évaluation.

Différentes méthodes de sélection ont été proposées dans la littérature. Elles sont comparées dans (Shukla et al., 2015). Nous présumons que l'évaluation des individus se fait par la fonction $fitness$.

- **La sélection par roulette :**

  Avec cette stratégie, la probabilité de sélection d'un individu à chaque tirage se fait proportionnellement à la valeur de $fitness$. Un tel opérateur est représenté par une roulette où les différents secteurs sont associés à chaque individu. La surface du secteur est proportionnelle à la valeur de $fitness$ de l'individu associé (*cf.* figure 4.3).

  La méthode de la roulette favorise la sélection des individus ayant des valeurs de $fitness$ les plus grandes. La sélection est biaisée et se concentre sur les individus dont la valeur d'évaluation est importante. Cette méthode a parfois tendance à ne sélectionner que les meilleurs individus

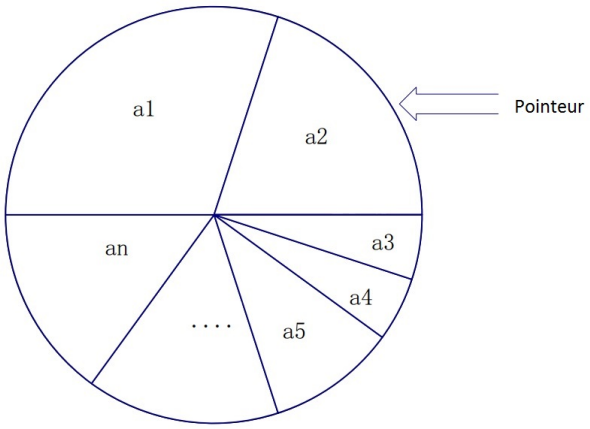


FIGURE 4.3 – Exemple montrant une flèche qui pointe sur le secteur a_2 correspondant au 2^{eme} individu. Le calcul des proportions est expliqué dans (Zhong et al., 2005)

et provoque des convergences prématurées vers des optimums locaux (Kumar, 2012).

- **La sélection par tournoi :**

  Plutôt que de se baser sur la valeur de la fonction d'évaluation, dont la mise au point est parfois difficile, la méthode de sélection par tournoi opère une sélection basée sur l'ordonnancement des individus au regard de cette fonction d'évaluation. Un tournoi procède dans un premier temps à un tirage aléatoire équiprobable de n individus dans la population. La fonction d'évaluation n'entre en considération que dans un second temps. Les individus sélectionnés seront les m meilleurs individus parmi les n au regard de la valeur de la fonction d'évaluation. La figure 4.4 présente un exemple de sélection par tournoi où $n = 3$ et $m = 1$.

  Dans (Goldberg and Deb, 1991), les auteurs ont expliqué que le choix du nombre n d'individus sélectionnés par tournoi n'a aucun impact sur la performance de l'algorithme génétique. Choisir 2 ou plusieurs individus au moment du tirage a le même effet. Pour cette raison, plusieurs travaux se sont contentés d'une configuration pour laquelle $n = 2$ et $m = 1$ à chaque tournoi (Ratnam et al., 2015, Coello and Montes, 2002, Lorena

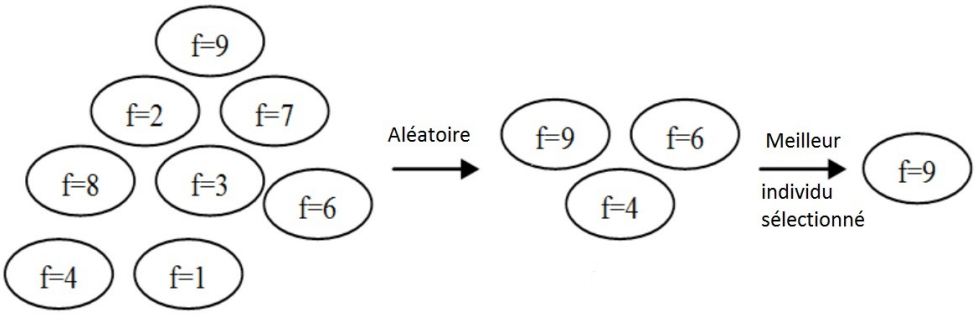


FIGURE 4.4 – Exemple issu de (Noraini and Geraghty, 2011) qui explique le principe de la sélection par tournoi. La population contient 8 individus dont 3 ont été tirés aléatoirement. Parmi ces derniers, l'individu dont la $fitness = 9$ est le meilleur, d'où sa sélection

and De Carvalho, 2008).

L'avantage de la méthode de sélection par tournoi réside dans son efficacité en temps de calcul pour comparer les différents individus choisis aléatoirement. Seul l'ordre au regard de la fonction d'évalution importe au moment de l'exécution du tournoi. De plus, la tâche de sélection par tournoi peut facilement être parallélisable sachant que la sélection d'un individu ne dépend pas de ses congénères contrairement aux autres méthodes de sélection (Shukla et al., 2015).

- **La sélection par classement :**

  Cette méthode, introduite par Baker, utilise une fonction linéaire pour attribuer à chaque individu une probabilité de tirage (Baker, 1985, 1987, Grefenstette and Baker, 1989). Comme les autres méthodes, les individus sont triés en se basant sur la valeur de $fitness$. Ensuite une fonction de probabilité est calculée pour chacun selon son ordre. La sélection des individus se base sur cette probabilité. Dans la littérature, plusieurs calculs de probabilité sont proposés. Dans (Shukla et al., 2015, Back, 1994), les auteurs présentent une fonction linéaire et une autre exponentielle.

D'autres méthodes de sélection sont présentées dans la littérature. Par exemple, dans (Yang et al., 2015) la sélection est réalisée par la méthode de Boltzmann et de l'état stable. Cependant, les travaux les plus récents utilisent

préférentiellement les méthodes par tournoi et par roulette.

Dans (Zhong et al., 2005), les auteurs présentent une étude comparative entre la sélection par tournoi et la sélection par roulette dans un algorithme génétique simple. Les résultats montrent que la sélection par tournoi est plus performante que celle opérée par roulette.

Dans (Julstrom, 1999), les auteurs s'intéressent plutôt au temps de calcul pris pour trouver l'optimal global en utilisant un algorithme génétique. Ils comparent différentes méthodes de sélection des individus. Les résultats de leurs expériences montrent que la sélection par tournoi reste la meilleure option en termes de performance et de temps de calcul.

Le croisement

Une fois l'opération de sélection appliquée, nous disposons des individus choisis pour participer au processus de reproduction permettant de générer de nouveaux individus. L'opérateur qui s'applique ensuite est l'opérateur de croisement. Cet opérateur applique le principe de la reproduction des chromosomes dans l'évolution naturelle. Le croisement vise à générer un individus dont le code génétique est constitué de recopies de certaines parties du code génétique de ses parents (Santos et al., 2015).

L'implémentation d'un tel opérateur dépend bien entendu de la façon dont est représenté le code génétique des individus. Le plus souvent, le gène est une séquence de taille fixe.

Dans la littérature, même si rien ne l'impose, les approches les plus fréquemment utilisées opèrent avec deux individus parents. Les méthodes de croisement agissant sur un code génétique présenté sous forme de séquence sont réparties en trois catégories différentes (Hwang and He, 2006, Kaya, 2011) :

- **Croisement point à point :** Il s'agit de sélectionner aléatoirement une position p entre 1 et la taille de l'individu. Le croisement s'effectue à la position p comme illustré dans la figure 4.5.
- **Croisement multi-points :** Il utilise le même principe que la catégorie

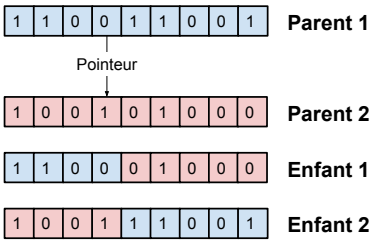


FIGURE 4.5 – Exemple de croisement point à point

précédente sauf que plusieurs positions sont choisies aléatoirement. Dans la figure 4.6, nous présentons un exemple où 2 positions sont choisies.

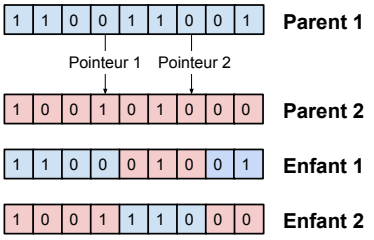


FIGURE 4.6 – Exemple de croisement multi-points et génération de 2 nouveaux individus

- **Croisement uniforme :** Cette méthode consiste à effectuer un échange entre les parents position par position. La permutation des éléments est conditionnée par un gêne temporaire *gene_temp* de valeurs binaires tirées selon la loi uniforme. À une position p si *gene_temp* est à 1 alors l'enfant 1 prend la valeur de parent 1 et l'enfant 2 prend la valeur du parent 2 et inversement dans le cas contraire comme illustré dans la figure 4.7.

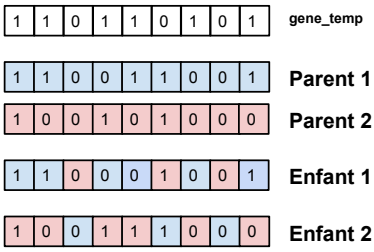


FIGURE 4.7 – Exemple de croisement à point et génération de 2 nouveaux individus

D'autres méthodes sont proposées mais elles sont adaptées à un domaine d'application précis tel que le croisement composé dans (Kaya, 2011).

Dans (Wu and Chow, 1995), les auteurs montrent que la méthode multi-point est plus performante que celle à une position. Dans (Jenkins, 1997), les auteurs rajoutent que le fait d'utiliser le croisement à un seul point ralentit la recherche de la solution optimale.

La mutation

La mutation est un opérateur qui permet d'effectuer une petite modification dans la représentation d'individu de façon à le doter de caractéristiques différentes de ses parents. Cet opérateur vise à explorer de nouvelles régions de l'espace de recherche. Si cette nouvelle caractéristique améliore la qualité de l'individu, elle aura tendance à perdurer au fil des générations du fait des opérations de sélection et croisement, alors qu'elle disparaîtra si la qualité de l'individu est détériorée (Santos et al., 2015), (Zhong et al., 2005).

L'opération de mutation va réintroduire de la diversité dans la population lorsque celle-ci converge et permet, ce faisant, l'exploration de nouvelles régions de l'espace de recherche. En effet, il a été démontré qu'un manque de diversité ralentit prématurément, voire stoppe l'évolution (Ratnam et al., 2015). En revanche, si la probabilité de mutation est grande, la recherche de la solution optimale revient alors à une recherche aléatoire primitive (Amer et al., 2011). Dans (Kaya, 2011, Soremekun et al., 2001, Ratnam et al., 2015), les auteurs ont cherché à identifier la valeur optimale de la probabilité de mutation. Leurs études révèlent que le système converge mieux lorsque cette probabilité est de l'ordre de 1%.

L'évaluation des individus

Nous avons évoqué précédemment le fait que l'opérateur de sélection exploite une fonction d'évaluation appelée $fitness$. Cette fonction est un élément primordial de l'algorithme génétique, car cette fonction mesure l'adaptation

d'un individu, c'est-à-dire la qualité de la solution encodée par l'individu pour le problème à résoudre. Cette fonction est à définir différemment pour chaque problème. Elle dépend bien évidemment de l'individu lui-même, mais bien souvent également d'une base sur laquelle est opérée l'évaluation de la solution encodée par l'individu (Miller et al., 2003, Fröhlich et al., 2002). Généralement, cette base d'évaluation est utilisée constamment et pour tous les individus. Nous n'avons pas connaissance de travaux dans lesquels, la base sur laquelle est évaluée la solution encodée par l'individu est amenée à évoluer elle-même, comme c'est le cas dans notre contexte.

Dans (Paulinas and Ušinskas, 2015), les auteurs présentent des exemples de fonction d'évaluation pour des problèmes de recherche de paramétrage optimal de méthodes de filtrage et de segmentation d'images.

L'élitisme

L'élitisme n'est pas un opérateur génétique à proprement parler dans le sens où il n'opère pas sur un individu. Il s'agit plutôt d'une politique de gestion de la population visant à garantir que les meilleurs individus d'une génération auront une valeur de fonction d'adaptation au moins aussi bonne que ceux de la génération précédente.

Contrairement aux autres opérateurs dont l'objectif est d'explorer de nouvelles solutions de l'espace de recherche, l'élitisme va faire perdurer les meilleurs individus d'une génération sur la suivante. En effet, l'exploration de nouvelles solutions au problème par le seul biais des opérateurs génétiques utilisés pour la reproduction (sélection, croisement, mutation) ne permet pas de garantir la convergence par améliorations successives. Il est même possible que le meilleur individu d'une génération ne soit plus représenté dans la génération suivante et la probabilité qu'un individu de même code génétique réintègre la population lors de génération suivante est très faible (Fogel, 1994).

De Jong fut le premier à introduire l'élitisme dans (DeJong, 1975). L'idée était de remplacer l'enfant le moins performant par le meilleur parent de la

génération précédente. Sa méthode reste à caractère exploratoire étant donné le nombre important de nouveaux individus par rapport aux anciens et ne garantit pas la conservation de la solution optimale dans la population.

Dans (Soremekun et al., 2001), le problème du nombre d'individus élite à conserver est soulevé. Les auteurs signalent qu'élire un seul individu ne permet pas d'obtenir une population contenant les solutions optimales quand il s'agit d'un problème à multiples solutions. Il existe alors d'autres méthodes permettant de conserver plusieurs individus. Nous parlons alors de l'élitisme multiple qui a pour objectif de sélectionner N meilleures solutions de la population courante et les copier dans la nouvelle génération. Du point de vue du temps de calcul, ce procédé accélère l'étape d'exploration des nouveaux individus puisqu'on réduit le nombre d'enfants à générer pour la nouvelle génération. Néanmoins, la valeur de N doit être définie judicieusement de manière à éviter de perdre les solutions optimales et de ralentir la convergence du système. Dans (Soremekun, 1997), l'auteur étudie la valeur de N et présente un élitisme variable où le nombre des meilleurs parents sélectionnés à chaque génération est variable. L'objectif est de privilégier l'exploration dans les premières itérations de l'algorithme et l'exploitation plus tard en augmentant la valeur de N. Cependant, cette méthode n'apporte pas de solution généralisable car son efficacité dépend généralement du problème à étudier.

4.4 Proposition

Dans cette thèse, nous nous proposons d'utiliser un algorithme génétique pour trouver la configuration du système d'extraction de la représentation structurelle des documents qui conduira aux meilleures performances en termes de localisation des zones recherchées, pour une classe donnée. Au fil de la vie du système, nous disposons d'un nombre initialement restreint mais croissant d'images représentant la classe.

La figure 4.8 rappelle le processus générique du système d'extraction des zones informatives. Il est composé de trois modules principaux. Les traite-

ments préliminaires visent à éliminer les informations, telles le bruit et ou les informations textuelles, qui pourraient perturber les traitements ultérieurs. Nous devons ensuite choisir l'espace colorimétrique dans lequel va être réalisée la quantification, qui va permettre de définir des régions de couleur homogène. Enfin, un dernier module va effectuer un filtrage des composantes connexes de façon à ne garder que les éléments avec des formes spécifiques, des rectangles dans notre cas de figure.

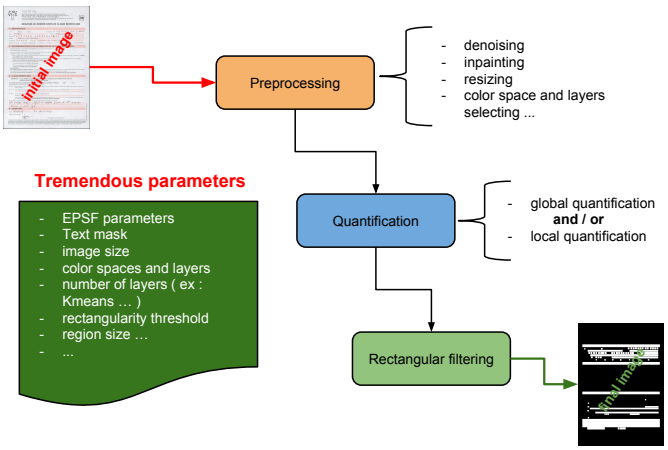


FIGURE 4.8 – Processus générique de l'extraction des zones informatives

4.4.1 Configuration de l'algorithme génétique

Ce processus générique doit être instancié en fixant un certain nombre de paramètres. L'algorithme génétique proposé va chercher à identifier à plusieurs instants de la vie du système, le paramétrage optimum au regard des informations disponibles. Le code génétique des individus manipulés par l'algorithme génétique encodera le paramétrage du système d'extraction de la représentation structurelle, que nous appellerons également "scénario d'extraction".

Les choix opérés pour l'implantation de l'algorithme génétique sont les suivants :

– Dans un premier temps, nous proposons de représenter le code génétique

des individus manipulés par l'algorithme génétique et représentant les solutions potentielles du problème d'optimisation, via une séquence de valeurs binaires.

Au sein de cette séquence binaire, certaines positions seront associées à des choix entre deux alternatives. Ils indiquent par exemple si un traitement doit être déclenché ou non dans le scénario d'extraction. Ils peuvent également indiquer si une caractéristique est intégrée ou non dans le traitement.

Par ailleurs, les valeurs numériques de paramètres peuvent être représentées par leur représentation binaire, qui constituera alors une sous-partie du code génétique binaire.

- Considérant l'analyse de l'état de l'art, nous avons opté pour un opérateur de sélection par tournoi. Les individus participant au processus de reproduction seront ceux dont la valeur d'adaptation sera la meilleure parmi deux individus tirés aléatoirement et de façon uniforme au sein de la population.
- L'analyse de l'état de l'art nous a également conduit à choisir pour opérateur de croisement, celui opérant avec un croisement multi-point entre deux individus parents. Une fois les deux parents sélectionnés, la reproduction génère deux individus par croisement de leur code génétique autour de deux positions choisies aléatoirement.
- Étant donné le choix opéré pour le codage des individus, l'opérateur de mutation va intervertir une des valeurs binaires stockée dans le code génétique. La position de cette valeur est elle-même choisie aléatoirement avec une probabilité uniforme.
- Nous opérons avec une population de taille fixe N avec une politique d'élitisme. À chaque génération, les n meilleurs individus de la génération précédente sont intégrés dans la nouvelle génération. Les $N-n$ individus restant sont générés par le processus de reproduction. Ce processus de reproduction consiste à sélectionner les individus parents, à générer de

nouveaux individus obtenus par croisement des parents, ces nouveaux individus peuvent être affectés de mutation avant d'intégrer la nouvelle génération. Ce processus est illustré sur la figure 4.9.

– Comme évoqué précédemment, le cycle de vie du système ne permet pas d'évaluer les individus au regard de l'objectif final, à savoir les meilleures performances en localisation, sur l'ensemble des documents que le système aura à traiter. De ce fait, la performance de chaque individu ne peut-être qu'estimée à un instant donné à partir uniquement des documents que le système aura rencontrés jusqu'à ce moment. Les fonctions utilisées pour qualifier la performance estimée d'un individu font l'objet de la section 4.4.2 où elles seront présentées en détail.
– Enfin, chaque déclenchement du processus hors-ligne d'optimisation du paramétrage de la chaîne d'extraction de la représentation structurelle correspond à une exécution de l'algorithme génétique. Nous utilisons un nombre fixé de générations comme critère d'arrêt de l'algorithme. Les motivations pour ce choix sont les suivantes. D'une part, choisir un nombre de générations fixé permet de garantir la terminaison de l'algorithme. Le choix du nombre de générations peut être établi au regard des contraintes relatives au temps d'exécution. D'autre part, si la fonction d'adaptation des individus doit guider les évolutions des générations successives, il est inutile d'en trouver la valeur optimale, cette dernière n'étant qu'une estimation de l'objectif réel. En effet, l'estimation de la qualité d'un individu est établie sur la base des documents vus par le système. Lors d'un nouveau déclenchement du processus hors-ligne d'optimisation, l'estimation de la qualité sera faite sur une base d'évaluation différente. Ce qui serait la valeur optimale de l'estimateur sur une base donnée, ne le serait plus nécessairement sur une autre base.

Le processus détaillé précédemment est guidé par plusieurs paramètres :

- Le nombre de générations. C'est le critère d'arrêt de l'algorithme ;
- La taille N de la population ;

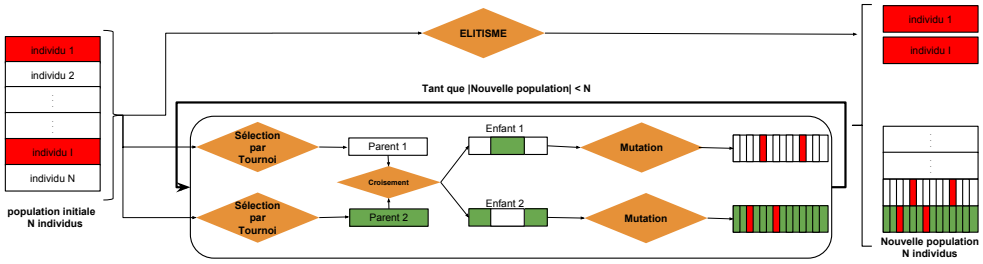


FIGURE 4.9 – Génération d'une nouvelle population

- Le nombre n de meilleurs individus à répliquer directement dans la génération suivante ;
- Le taux de mutation : probabilité qu'un individu généré voit modifiée une des valeurs binaires constituant son code génétique ;
- La fonction qui permet d'évaluer la performance estimée des individus.

4.4.2 Fonction d'évaluation des individus

Nous présentons dans cette section la fonction d'évaluation des individus. Il s'agit de la fonction qui permet de mesurer la performance de la solution encodée par un individu manipulé par l'algorithme génétique. Rappelons que dans notre cas, une telle mesure n'est pas disponible. En effet, lors du cycle de vie du système, les informations de vérité terrain qui permettraient d'évaluer la performance en localisation de chacun des paramétrages ne sont pas disponibles. Par ailleurs, à un instant donné le système n'a connaissance que des instances de document déjà traitées. La mesure de performance estimée ne peut donc se faire que sur cette base. Cependant, le nombre d'instances traitées par le système va augmenter et il sera possible d'intégrer de nouvelles instances dans le calcul de l'estimation, ce qui affinera l'estimation.

La fonction d'évaluation des individus que nous proposons utilise une mesure de la stabilité des représentations structurelles. Cette mesure vise à évaluer la similarité entre les représentations structurelles associées à un scénario d'extraction au sein d'une même classe de documents. Nous émettons l'hypothèse que cette mesure est corrélée avec les performances en localisation

opérée par recherche d'isomorphisme de sous-graphe. L'idée sous-jacente est que la recherche du sous-graphe de la représentation structurelle du document d'apprentissage au sein de la représentation structurelle d'un document cible a davantage de chance d'être efficace si les representations structurelles sont elles-mêmes similaires. Ainsi, un scénario d'extraction, représenté par un individu de l'algorithme génétique, sera d'autant meilleur du point de vue de l'objectif final que les représentations structurelles issues de ce scénario d'extraction pour les documents de la classe seront similaires entre-elles. À l'inverse, un scénario qui proposerait des représentations structurelles de faible similarité sera faiblement évalué car la recherche d'une occurrence d'un sous-graphe de la représentation structurelle du document d'apprentissage au sein de celle du document cible aura très peu de chance d'aboutir si les représentations structurelles sont dissemblables. Cette hypothèse sera vérifiée dans la partie expérimentale.

Considérant d_1 et d_2 deux documents de la même classe, et sc un scénario d'extraction de la représentation structurelle, l'équation 4.4 définit la mesure de similarité entre les deux représentations structurelles proposées par sc respectivement extraite de d_1 et d_2.

$$stab(d_1, d_2, sc) = \frac{\sum_{i=1}^{|zon(d_1,sc)|} \sum_{j=1}^{|zon(d_2,sc)|} \delta(Jac(zon(d_1, sc)[i], zon(d_2, sc)[j]) > \theta)}{min(|zon(d_1, sc)|, |zon(d_2, sc)|)} \tag{4.4}$$

Dans cette formule, $zon(d, sc)$ représente l'ensemble des zones extraites par sc sur le document d, $zon(d, sc)[i]$ représente la i^{e} zone extraite par sc sur le document d, $Jac(z_1, z_2)$ représente l'indice de Jaccard entre les zones z_1 et z_2. Cet indice représente le rapport entre la surface commune aux zones z_1 et z_2 et la surface de l'union de ces mêmes zones. Enfin, la fonction $\delta(x > \theta)$ est une fonction indicatrice valant 1 si x est supérieur à θ et 0 sinon. Ainsi, la fonction $stab$ établit la proportion de zones d'un document dont le chevauchement mesuré par $Jaccard$ est supérieur à θ, à ceci près qu'une zone de d_1 peut chevaucher plus d'une zone de d_2 et réciproquement.

Étant donné D un ensemble de documents de la même classe, nous définissons la mesure $stab(sc, D)$ qui moyenne la valeur de $stab(d_1, d_2, sc)$ pour tous les couples $(d_1, d_2) \in D^2, d_1 \neq d_2$.

$$stab(sc, D) = \frac{2.\sum_{i=1}^{|D|}\sum_{j=i+1}^{|D|} stab(D[i], D[j], sc)}{|D|.(|D|-1)} \tag{4.5}$$

Dans cette formule, $D[i]$ représente le i^{e} document de la classe D.

Durant la vie du système, les évaluations des différents individus doivent être opérées avec un ensemble D dont la taille sera croissante. Or, nous remarquons que le calcul de cette mesure est quadratique du point de vue de l'effectif de l'ensemble D. Par ailleurs, ce procédé nécessite la mémorisation de l'ensemble des documents vus depuis le début du cycle de vie, ce qui peut être coûteux du point de vue de l'occupation mémoire. Enfin, l'évaluation d'un individu apparaissant pour la première fois nécessite que l'extraction de la représentation structurelle selon les paramètres définis par l'individu soit effectuée sur l'ensemble des documents vus. Pour remédier à ces inconvénients, nous proposons de ne pas calculer la mesure de stabilité sur l'ensemble des documents vus à chaque exécution du processus hors-ligne. À chaque déclenchement du processus hors-ligne, nous proposons de ne calculer la mesure de stabilité que sur l'ensemble des documents vus depuis l'exécution précédente.

Nous désignons par L_t l'ensemble des documents traités par le processus hors-ligne entre sa $(t-1)^{e}$ et sa t^{e}. Dès lors, au t^{e} déclenchement du processus hors-ligne, $stab(sc, L_t)$ représente la mesure de stabilité correspondant au scénario d'extraction sc évaluée sur les documents rencontrés depuis la dernière exécution.

À partir de ces définitions, nous proposons trois mesures pour évaluer la fonction d'adaptation des individus dans l'algorithme génétique.

estimation sans mémoire : Lors de chaque évaluation appelée par l'algorithme génétique se déroulant durant le t^{e} déclenchement du processus hors-ligne, l'évaluation des individus n'est opérée que sur la base des

documents vus depuis l'exécution précédente.

$$fit_{no\_mem}(sc, t) = stab(sc, L_t) \tag{4.6}$$

estimation avec mémoire : Lors de chaque évaluation appelée par l'algorithme génétique se déroulant durant le t^{e} déclenchement du processus hors-ligne, le calcul de la mesure de stabilité $stab(sc, L_t)$ est opérée sur la base des documents vus depuis le dernier déclenchement. En revanche, l'évaluation de l'individu correspondant à sc ne repose pas uniquement sur $stab(sc, L_t)$, mais elle tient compte également des évaluations qui ont pu être effectuées lors des déclenchements précédents.

$$fit_{mem}(sc, t) = \frac{n(sc, L_{t-1}).fit_{mem}(sc, L_{t-1} + \frac{|L_t|.(|L_t|-1)}{2}.stab(sc, L_t)}{n(sc, L_{t-1}) + \frac{|L_t|.(|L_t|-1)}{2}} \tag{4.7}$$

Dans cette expression, $n(sc, L_{t-1})$ représente le nombre de paires de documents a partir desquelles $fit_{mem}(sc, L_{t-1})$ a été calculé.

estimation primée : Nous proposons également une troisième fonction pour l'évaluation des individus correspondant aux scénarios d'extraction. L'idée de cette troisième fonction est de promouvoir des individus apparus durant les dernières générations. Pour peu que leur performance estimée soit supérieure à celle de la moyenne des scénarios, leur estimation est bonifiée de telle sorte qu'ils puissent perdurer sur quelques itérations, leur donnant ainsi l'opportunité de confirmer ou d'infirmer leur caractère prometteur. Nous définissons alors la mesure $fit_{prime}(sc, t)$.

$$fit_{prime}(sc, t-1) = \frac{fit_{mem}(sc, t-1).\sum_{i=0}^{t-1}|L_i| + \overline{fit}.(|B_{t-1}| - \sum_{i=0}^{t-1}|L_i|)}{|B_{t-1}|} \tag{4.8}$$

Dans l'équation 4.8, les ensembles L_i représentent les ensembles de documents ayant fait l'objet d'une évaluation de stabilité pour le scénario sc, et l'ensemble B_{t-1} représente l'ensemble des documents vus par le

système jusqu'à l'instant $t-1$, que ceux-ci aient été traités ou non par le scénario sc. La fonction d'évaluation primée à la génération $t-1$ pour l'individu désigné pas sc est calculée comme une moyenne pondérée entre, d'une part, la fonction d'adaptation propre au scénario considéré, et d'autre part $\overline{fit}$, la performance moyenne de l'ensemble des scénarios. La pondération associée à chaque composante est $\frac{\sum_{i=0}^{t-1}|L_i|}{|B_{t-1}|}$, la proportion des documents sur lesquels la performance en stabilité a été mesurée pour le scénario sc, et $\frac{(|B_{t-1}|-\sum_{i=0}^{t-1}|L_i|)}{|B_{t-1}|}$ la proportion de documents sur lesquels cette proportion n'a pas été calculée.

Le critère de sélection est reformulé par :

$$fit_{prime}(sc,t) = \frac{|B_{t-1}|.fit_{prime}(sc,t-1) - stab(sc,L_t)}{|B_t|} \tag{4.9}$$

En remplaçant, $fit_{prime}(sc,t-1)$ par son expression donnée par l'équation 4.8, nous trouvons :

$$fit_{prime}(sc,t) = \overline{fit} + \frac{(fit_{mem}(sc,t) - \overline{fit}).\sum_{i=0}^{t}|L_t|}{|B_t|} \tag{4.10}$$

Considérant que, dans la formule 4.10, $|B_t|$ et $\overline{fit}$ sont constantes vis à vis de sc, le critère de sélection peut se baser uniquement par rapport à $(fit_{mem}(sc,t) - \overline{fit}).\sum_{i=0}^{t}|L_t|$, c'est-à-dire l'écart avec la moyenne des fonction d'évaluation, pondéré par le nombre de documents traités.

4.5 Évaluation expérimentale

Dans cette section, nous présentons les expériences que nous avons menées pour l'évaluation de nos propositions. Après avoir défini la structure des individus manipulés par notre algorithme génétique, nous validons d'abord expérimentalement l'hypothèse que nous avons formulée concernant la corrélation entre la mesure de stabilité proposée et les performances du système du point de vue de son objectif de localisation des zones informatives. Puis, nous pré-

sentons et discutons les résultats obtenus par notre algorithme.

,

4.5.1 Définition de la structure de l'individu

Nous avons choisi d'évaluer le système d'extraction des zones informatives utilisé dans les expériences détaillées dans (Hammami et al., 2015). Quatre éléments composent l'espace de paramètres qui doit être exploré, à savoir :

- L'utilisation du filtre de lissage (Nikolaou and Papamarkos, 2009a) : les expériences ont montré qu'$EPSF$ n'est pas obligatoire dans le traitement des images ; ceci dépend de la qualité de l'image.
- L'utilisation de l'*inpaint* : parfois l'application de ce traitement engendre des déformations dans l'image et cause la perte des éléments importants comme le contour des objets.
- Le choix des plans de couleur : nous avons choisi d'étudier les plans des espaces couleurs suivants RGB , Lab , $YCbCr$. Ce choix est justifié par les expériences préliminaires que nous avons effectuées et qui ont révélé les meilleures performances.
- Le choix de la valeur de K pour la quantification de couleur. Dans (Hammami et al., 2014), nous avons montré que le choix de la valeur de K est important pour avoir une bonne segmentation de l'image. Selon les précédentes expériences, nous choisissons d'étudier cette valeur dans l'intervalle $[2, 5]$

Comme nous avons choisi de représenter les individus sous forme d'une chaîne binaire, nous avons alors besoin de 13 bits pour représenter les différents paramètres à étudier. Le 1^{er} et le 2^{eme} bit définissent s'il y a utilisation du filtre $EPSF$ et de l'*inpaint* ou non. Si le bit correspondant à $EPSF$ est à 1 alors nous appliquons sur l'image le filtre $EPSF$ sinon l'image ne sera pas traitée. Il en est de même pour le traitement d'*inpainting*.

Ensuite, nous utilisons 2 bits pour représenter la valeur de K qui varie entre 2 et 5. Nous avons besoin de 2 bits car K peut prendre une valeur parmi

les 4 valeurs entières de l'intervalle $[2, 5]$. Le tableau 4.1 illustre la combinaison des 2 bits pour représenter les différentes valeurs :

Valeurs de K	**Combinaisons**
2	00
3	01
4	10
5	11

TABLE 4.1 – La représentation des différentes valeurs de K dans la structure de l'individu

Enfin, les 9 derniers bits représentent les différents plans des espaces couleur RGB, Lab et $YCbCr$. Là encore, un bit à 1 signifie que le plan couleur correspondant est utilisé. La figure 4.10 résume la structure de l'individu que nous utilisons dans notre algorithme génétique. Cette représentation nous permet d'avoir un espace de recherche comportant 8192 configurations différentes.

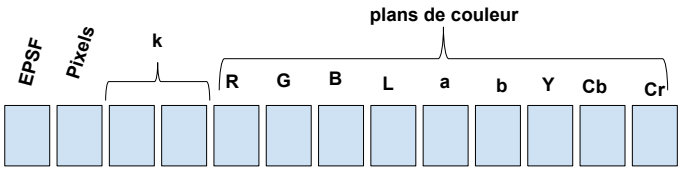


FIGURE 4.10 – Code génétique de 13 bits correspondant à 8192 scénarios

4.5.2 Examen de la corrélation entre la mesure de stabilité et la performance de la recherche de zones

Dans la section 4.4.2, nous avons émis l'hypothèse que la mesure de stabilité présentée précédemment qui évalue la similitude entre les representations structurelles fournies par un paramétrage donné était corrélée avec la fonction objectif non accessible. Cette hypothèse se base sur le raisonnement suivant : la probabilité que la recherche d'une occurrence d'un sous-graphe de G_1 au sein d'un graphe G_2 aboutisse, augmente avec la similarité entre G_1 et G_2.

Les expérimentations présentées ici visent à valider cette hypothèse. Dans ce cadre, pour chacun des 8192 (2^{13}) paramétrages possibles, nous avons

extrait les représentations structurelles pour chaque document de la base d'évaluation. Il a alors été possible de mesurer dans un premier temps la mesure de stabilité pour chacune des classes de cette base. La mesure de stabilité associée au paramétrage est alors calculée comme la valeur moyenne des mesure de stabilité.

Par ailleurs, toujours pour chaque paramétrage possible, et pour chaque classe de document, nous avons extrait aléatoirement 50 sous-graphes à partir du graphe décrivant le document. Ces sous-graphes sont des sous-graphes de la représentation structurelle induit par l'ensemble des nœuds contitué d'un nœud choisi aléatoirement et de l'ensemble de ces voisins immédiats. Pour chaque sous-graphe requête construit de cette façon, nous effectuons une recherche d'isomorphisme de sous-graphe au sein des représentations structurelles de la même classe. Nous considérons qu'un sous-graphe est retrouvé si la moitié au moins des appariements de nœuds sont corrects. Un appariement est considéré correct s'il y a un chevauchement mesuré par l'indice de Jaccard d'au moins 10% entre les zones correspondantes sur l'image. Dès lors, la mesure de performance de la recherche d'isomorphisme est la proportion de recherches d'isomorphisme ayant abouti.

La figure 4.11 représente la mesure de stabilité en fonction de la la mesure de performance en recherche de sous-graphe pour un ensemble de configurations tirées aléatoirement sur la classe 2 de notre base. La figure 4.12 représente graphiquement ces mêmes mesures pour l'ensemble des 8192 configurations pour pour chacune des classes de notre base d'évalutation.

Même s'il n'y a pas de relation liant de façon déterministe la performance de la recherche d'isomorphisme avec la mesure de stabilité, nous remarquons tout de même une corrélation certaine sur l'ensemble des classes qui vient valider l'hypothèse sur laquelle repose notre approche. Ainsi, sur l'ensemble des paramétrages, la probabilité d'avoir une recherche d'isomorphisme performante progressera avec la valeur de la mesure de stabilité.

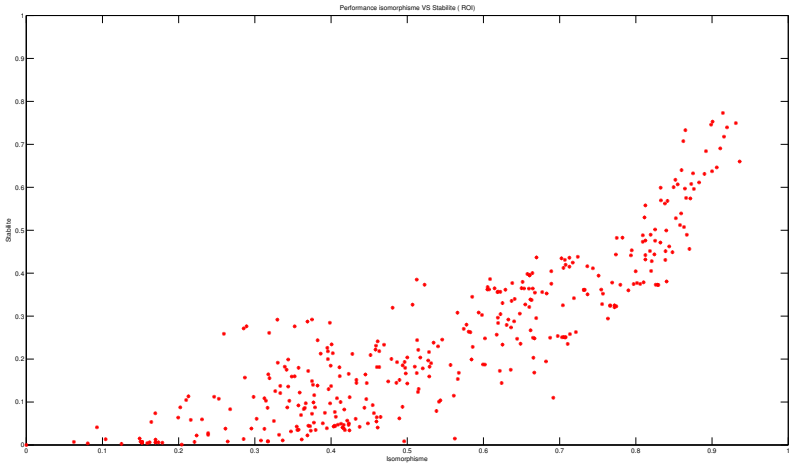


FIGURE 4.11 – Examen de la corrélation entre mesure de stabilité et performance de la recherche d'isomorphisme concernant la classe 2 sur un tirage aléatoire des configurations

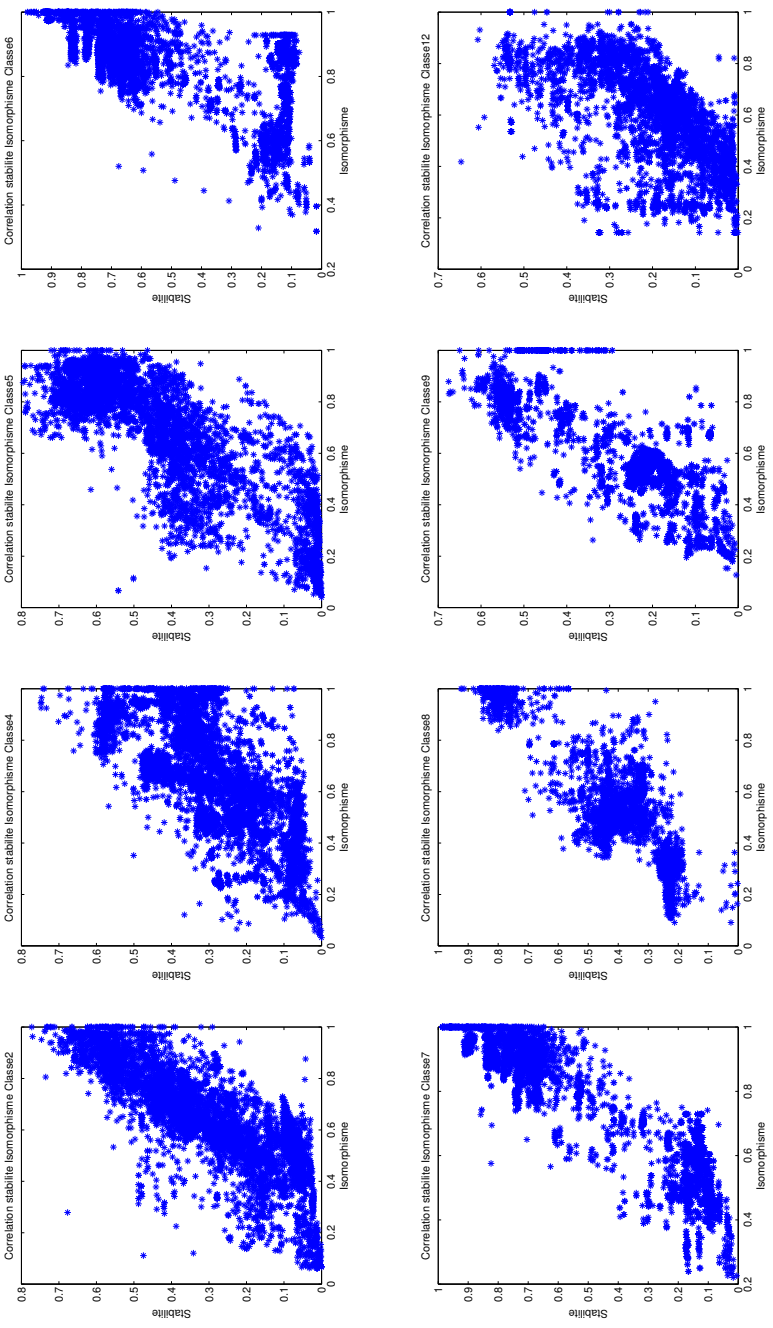


FIGURE 4.12 – Corrélation entre stabilité et performance d'isomorphisme des sous-graphes pour toutes les classes de la base *Iteso f t*

4.5.3 Évaluation expérimentale de l'approche proposée

Nous présentons dans les paragraphes qui suivent les résultats obtenus lors de l'évaluation de l'approche proposée. Le protocole expérimental est donné par l'algorithme 1. Pour chaque classe de documents, nous disposons d'un ensemble de documents. Cet ensemble est subdivisé en lots qui seront présentés à l'algorithme successivement de façon à simuler l'apport de nouveaux documents de la classe au fil de la vie du système.

Algorithme 1 : Protocole expérimental

input : $\{L_l\},\ l \in [1, nb\_lots]$
input : $nbIt$: nombre d'itération de l'algorithme génétique par lot de documents
input : $taillePop$: nombre d'individus dans la population
input : $nbElite$: taille de l'élite
output : $scenario$: Meilleur individu dans la population

```
1  InitPop (population_0,taillePop);
2  pour l ← 1 à nb_lots faire
3      pour i ← 1 à nbIt faire
4          MAJPerformance(population_0, L_l);
5          population_1 ← Elite(population_0,nbElite);
6          tant que taille(population_1) < taillePop faire
7              parents ← SelectionParTournoi(population_0);
8              enfants ← Croisement(parents);
9              enfants ← Mutation(enfants);
10             population_1 ← population_1 ∪ enfants;
11         fin
12         population_0 ← population_1;
13     fin
14 fin
15 scenario ← Elite(population_0, 1)
```

Les évaluations présentées ont été obtenues avec le paramétrage suivant :

- $taillePop$: la taille de la population a été fixée à 8 scénarios ;
- $nbElite$: le nombre de meilleurs individus recopiés sur la génération suivante a été fixé à 2 ;
- $nbIt$: le nombre d'itérations de l'algorithme génétique pour chaque lot de documents a été fixé à 10 ;

L'algorithme génétique a été paramétré de sorte que la probabilité qu'un bit du code génétique soit modifié par l'opérateur de croisement soit de 1/13. Ainsi, puisque les individus ont un code génétique de longueur 13, ils auront en moyenne 1 bit changé par l'opérateur de mutation.

La base de documents utilisée dans ces évaluations est identique à celle utilisée dans les évaluations du chapitre précédent. Cependant, considérant le faible effectif des classes, cette base a été artificiellement augmentée selon le processus décrit par la figure 4.13. Pour chaque document, nous extrayons le masque correspondant au texte à l'aide de la méthode de « Smart Binarization » (Gaceb et al., 2013). Nous isolons ainsi sur deux images distinctes les pixels correspondant au texte de ceux correspondant au fond du document. A partir de cette dernière image, nous reconstituons une image du document dans laquelle l'ensemble du texte a été supprimé par application du traitement d'*inpainting*. Il nous est alors possible d'insérer dans ce document la partie textuelle provenant d'un autre document de la même classe. Pour traiter les problèmes liés à des dimensions différentes, cette insertion s'opère après un traitement d'interpolation. De même, une translation est appliquée pour faire face au problème de recalage. De cette façon, partant d'une classe de documents dont l'effectif initial serait de n documents, nous portons son effectif à n^2.

À partir de la base de documents ainsi obtenue, nous générons un ensemble de lots $\{L_t\}$ pour chaque classe. Les lots sont constitués de 5 documents de la même classe tirés aléatoirement de la base augmentée.

Les figures 4.14, 4.15 et 4.16 présentent des exemples de résultat obtenus par application de l'algorithme 1 sur les différentes classes de notre base d'évaluation. Ces trois figures représentent trois configurations différentes du système du point de vue de la fonction utilisée pour évaluer les individus. La figure 4.14 montre les résultats obtenus dans le cas où la fonction d'évaluation est fit_{no_mem}. La figure 4.15 correspond à la configuration où la fonction d'évaluation est fit_{mem}. Enfin, la fonction fit_{prime} est utilisée comme

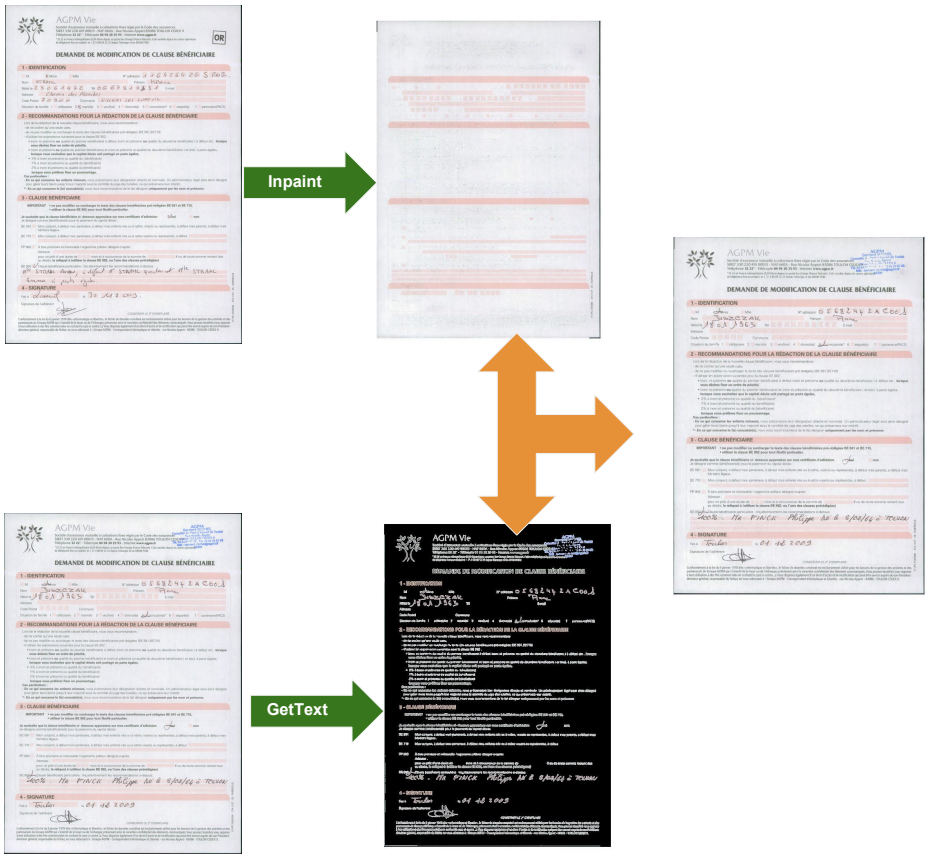


FIGURE 4.13 – Exemple de génération d'un nouveau document

fonction d'évaluation des individus sur la configuration à l'origine de la figure 4.16.

Sur ces figures, le tracé horizontal rouge indique l'oracle, c'est-à-dire la valeur maximale de la mesure de stabilité mesurée sur l'ensemble des documents de la classe après une exploration exhaustive de l'ensemble des 8192 configurations. L'exploration exhaustive ayant conduit à la détermination de cette valeur ne serait pas possible en condition réelle d'utilisation, même pour un jeu restreint de documents. En effet, l'exploration complète de l'espace de recherche est rendue rédhibitoire par le temps de traitement (application de la chaîne de traitement pour tous les documents et pour toutes les configurations). La courbe noire représente la valeur de la stabilité pour le meilleur individu à une itération donnée, mais cette mesure est calculée sur l'ensemble

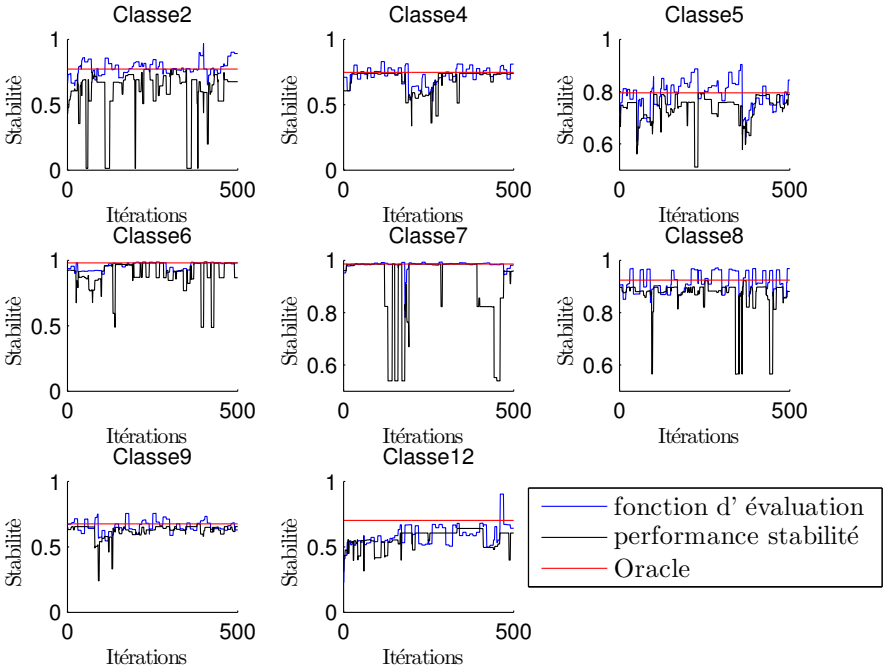


FIGURE 4.14 – Evolution sans mémoire

des documents de la classe (incluant ceux présents dans les lots ultérieurs). C'est cette mesure évaluée sur l'ensemble des documents de la classes y compris ceux encore inconnus du système qu'on souhaite en réalité optimiser dans un premier temps. Enfin, la courbe bleue représente la valeur de la fonction d'évaluation du meilleur individu à l'itération considérée. Cette fonction pilote l'algorithme génétique au regard des informations disponibles, c'est-à-dire exploitant uniquement le lot de documents courant et éventuellement les lots précédents.

Il est à noter que, considérant que les algorithmes génétiques utilisent des processus aléatoires, les résultats ne sont pas reproductibles. Il est donc difficile de tirer des conclusions de ces seules figures. Cependant, elles permettent d'illustrer certaines observations que nous avons pu faire sur des tests de plus grande envergure qu'il est difficile de synthétiser ici.

En premier lieu, nous remarquons que la courbe noire représentant les

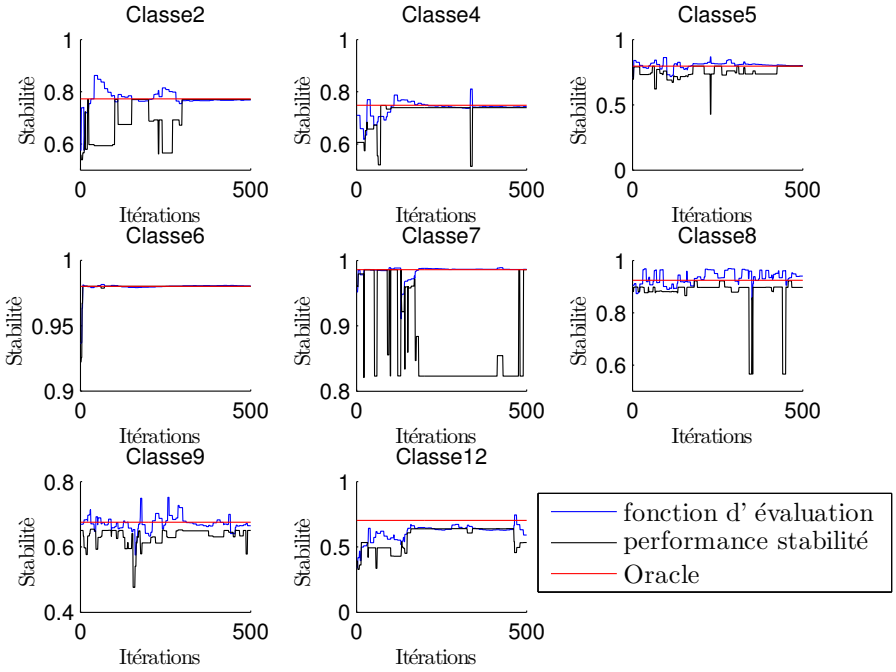


FIGURE 4.15 – Evolution avec mémoire

performances en stabilité sur la globalité de la classe du meilleur individu subit le moins d'irrégularités dans le cas où la fonction d'évaluation avec mémoire primée est utilisée. La valeur atteinte est certes un peu plus éloignée de la valeur optimale que pour les meilleures valeurs obtenues avec la fonction d'évaluation sans mémoire et celle avec mémoire simple, mais elle est à tendance plutôt croissante et plutôt stable à un niveau acceptable. Les deux autres fonctions d'évaluation (sans mémoire et avec mémoire simple) conduisent à des phénomènes de sur-apprentissage et de spécialisation aux lots.

En effet, on observe que la fonction d'évaluation sans mémoire est soumise à de fortes irrégularités qui traduisent une dépendance au lot d'images sur lequel la fonction est évaluée. Ainsi, en optimisant le paramétrage sur un lot d'images, les performances globales peuvent être atténuées.

Si la fonction d'évaluation avec mémoire atténue le phénomène, elle ne l'élimine pas. Ceci s'explique par le fait que la fonction d'évaluation avec

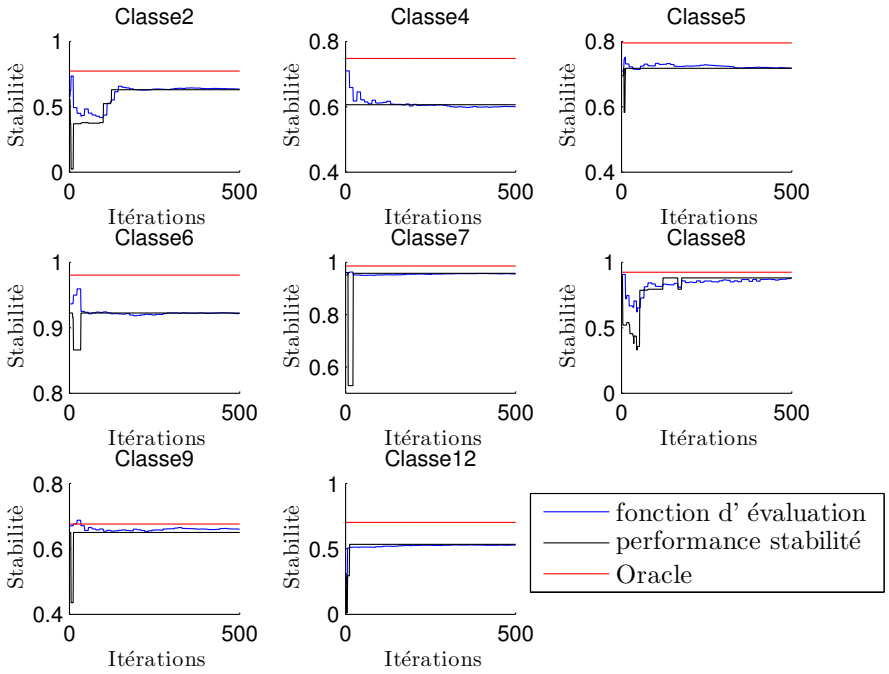


FIGURE 4.16 – Evolution avec mémoire primée

mémoire est une moyenne calculée sur la base des seuls lots où elle a été évaluée. Or, un scénario peut très bien se comporter sur quelques lots de la base, ceux sur lesquels sera calculée la moyenne, et avoir un comportement médiocre sur la globalité de la base. C'est l'hypothèse que nous formulons en observant que la chute de performance en stabilité et souvent corrélée à une augmentation de la fonction d'évaluation.

C'est cette observation qui nous a conduit à introduire la fonction d'évaluation avec mémoire primée qui fait intervenir une moyenne pondérée de l'évaluation propre au scénario et la moyenne des évaluations de l'ensemble des scénarios. La pondération est établie sur le nombre de lots de document.

Ces observations sont corroborées par la figure 4.17 qui montre que l'évaluation du meilleur individu reste plus stable dans le cas de la mémoire primée qui intègre le nombre de lots traités dans le calcul de la moyenne.

Nous présentons sur la figure 4.18 le détail de l'évolution des performances

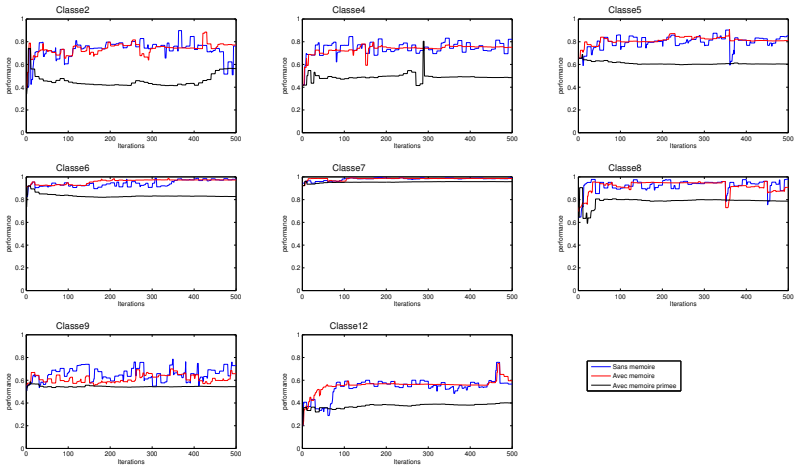


FIGURE 4.17 – Comparaison de la valeur de la fonction d'évaluation du meilleur individu

sur les premières itérations. On y remarque notamment que la performance du système est plus faible au début du processus, là où la population est initialisée de façon aléatoire. Puis les performances s'améliorent en faisant évoluer la population sur la base de la fonction d'évaluation des individus calculée sur le même lot de documents. La prise en compte d'un nouveau lot de documents s'accompagne d'une chute de performance alors que la valeur de la fonction d'évaluation du meilleur individu change brusquement, à la hausse ou à la baisse. Malgré cette chute, la mesure de performance reste supérieure à la valeur initiale. Par ailleurs, la mesure de performance se rétablit très rapidement au niveau de ses valeurs maximales.

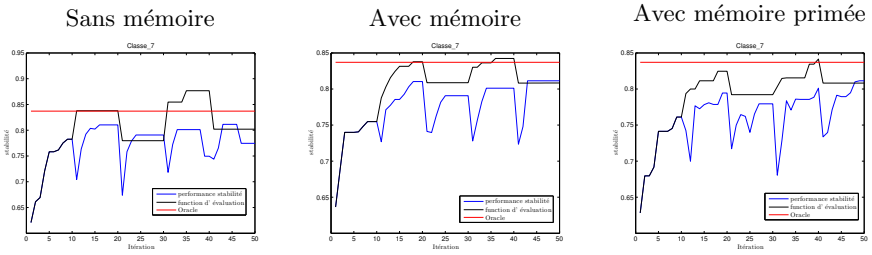


FIGURE 4.18 – Exemple de la classe 7

Enfin, nous présentons sur les figures 4.19, 4.20 et 4.21 l'évolution de la

performance de la recherche d'isomorphisme du meilleur individu sur une évolution de 500 itérations durant laquelle un nouveau lot de 5 documents est apporté toutes les 10 itérations. Globalement, nous remarquons sur ces figures, comme déjà évoqué précédemment, la corrélation qui existe entre la valeur de la fonction d'évaluation des individus et la performance de la recherche d'isomorphisme. En outre, en accord avec les observations que nous avions pu faire concernant l'évolution de la fonction d'évaluation, nous remarquons une meilleure stabilité globale de la fonction d'évaluation, et par voie de conséquence de la performance de la recherche d'isomorphisme dans le cas de l'usage de la fonction avec mémoire primée. Dans ce dernier cas, après quelques itérations la performance reste à un niveau stable et relativement élevé. Pour certaines classes, 100% des graphes requêtes sont retrouvés (la moitié des nœuds correctement appariés). Dans le cas des classes les plus difficiles, ce taux se situe au alentours de 80%. Pour les systèmes fonctionnant avec une fonction d'évaluation sans mémoire ou avec mémoire simple, la mesure de performance est affectée par les variations de la fonction d'évaluation et quitte fréquemment sa valeur optimale.

4.6 Conclusion

Dans ce chapitre, nous avons abordé la problématique de la recherche d'un paramétrage optimal d'un système pour lequel les informations permettant l'évaluation ne sont d'une part qu'uniquement estimées, et d'autre part, accessibles que progressivement dans le temps. Nous avons proposé une première réponse à cette problématique par le biais d'un système évolutionnaire construit sur un algorithme génétique dont la fonction objectif est corrélée avec l'objectif réel, la maximisation de la performance de la recherche des isomorphismes de sous-graphe dans notre cas d'application. Le choix de l'approche par algorithme génétique a été motivé non seulement par les bonnes performances que permettent d'atteindre ces algorithmes, mais également par le fait que leur fonctionnement repose sur la gestion d'une population d'individus

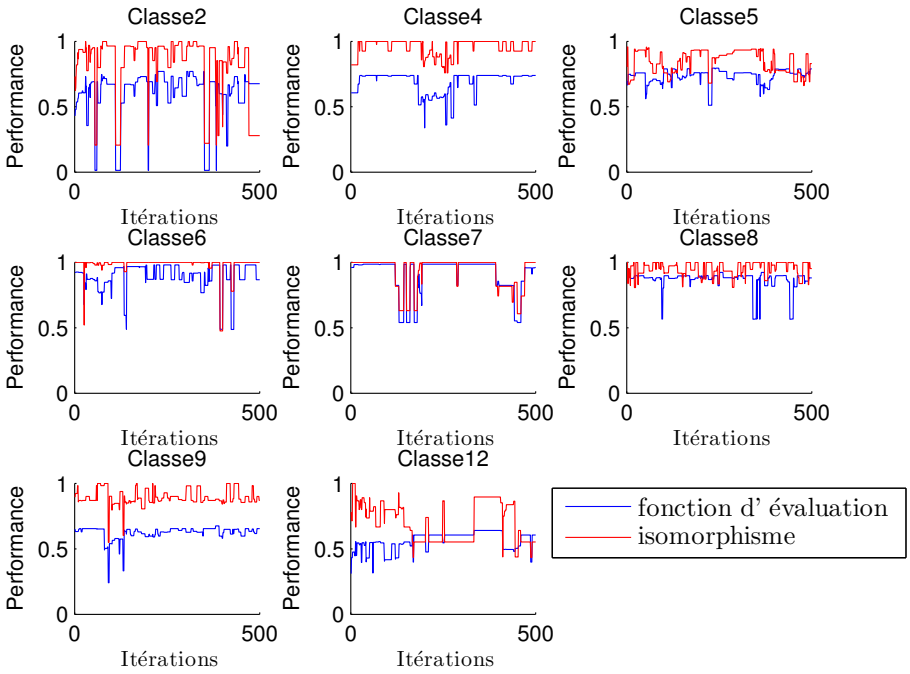


FIGURE 4.19 – Evolution de la performance en isomorphisme - Sans mémoire

encodant chacun une solution potentielle. La gestion d'une population de solutions permet de faire face à la difficulté posée par la remise en cause de la qualité des solutions qui intervient lorsque de nouvelles bases d'évalution doivent être considérées.

Dans notre cadre applicatif, nous avons proposé de baser l'évaluation de la qualité des solutions sur un critère mesurant la similitude des représentations structurelles produites en émettant l'hypothèse que la performance de l'isomorphisme est corrélée positivement avec cette mesure. Cette hypothèse a été validée expérimentalement. Nous avons également proposé trois fonctions d'évaluation différentes des individus intégrant cette mesure de stabilité. Ces trois fonctions diffèrent selon qu'elle prennent ou non en considération les valeurs précédentes, mais également l'effectif des documents sur lequel ont été faite les évaluations.

Les expérimentations menées ont montré une meilleure robustesse de la

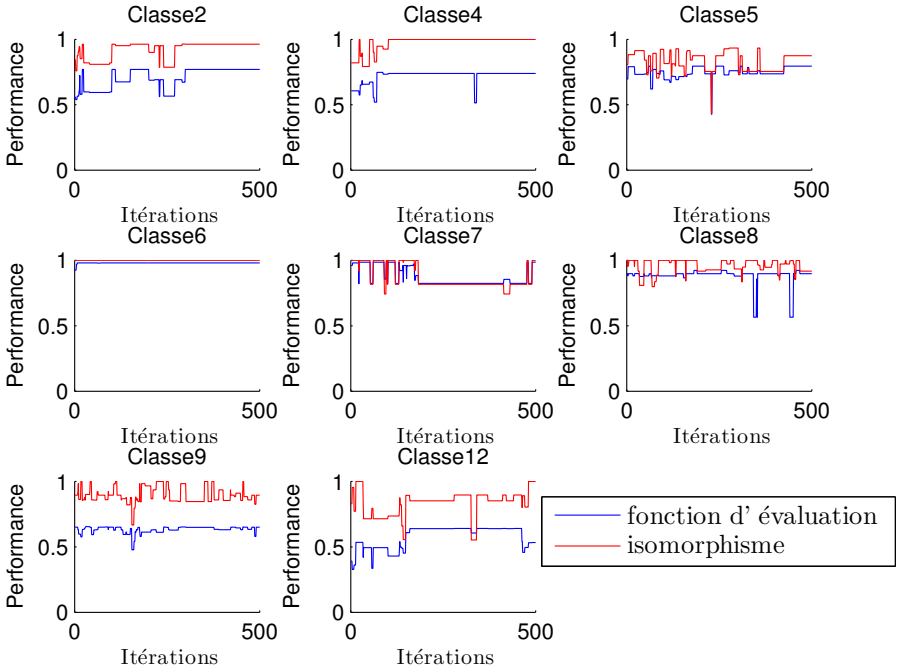


FIGURE 4.20 – Evolution de la performance en isomorphisme - Avec mémoire

fonction d'évaluation aux évolutions de la base lorsque qu'elle intégrait une mémorisation et la prise en compte du volume de la base d'évaluation. Finalement, le système utilisant cette configuration a montré des performances acceptables et stables au regard de l'objectif final de recherche d'isomorphisme.

L'approche proposée est une première réponse à la problématique de l'optimisation dans un cadre évolutif avec fonction objectif non accessible. Les limites de cette approche permettent d'ouvrir des perspectives de poursuite de ces travaux. En premier lieu, il convient de rappeler que les algorithmes génétiques sont des processus aléatoires et de ce fait ne sont pas reproductibles. Il n'est donc pas possible de garantir la qualité de la solution proposée au regard de l'objectif réel. Même une estimation nécessiterait de disposer d'une base avec vérité terrain qui n'est pas disponible dans les conditions réelles. Par ailleurs, il conviendrait de réaliser une étude complète de l'influence des différents paramètres tels que la taille de la population, le taille de l'élite,

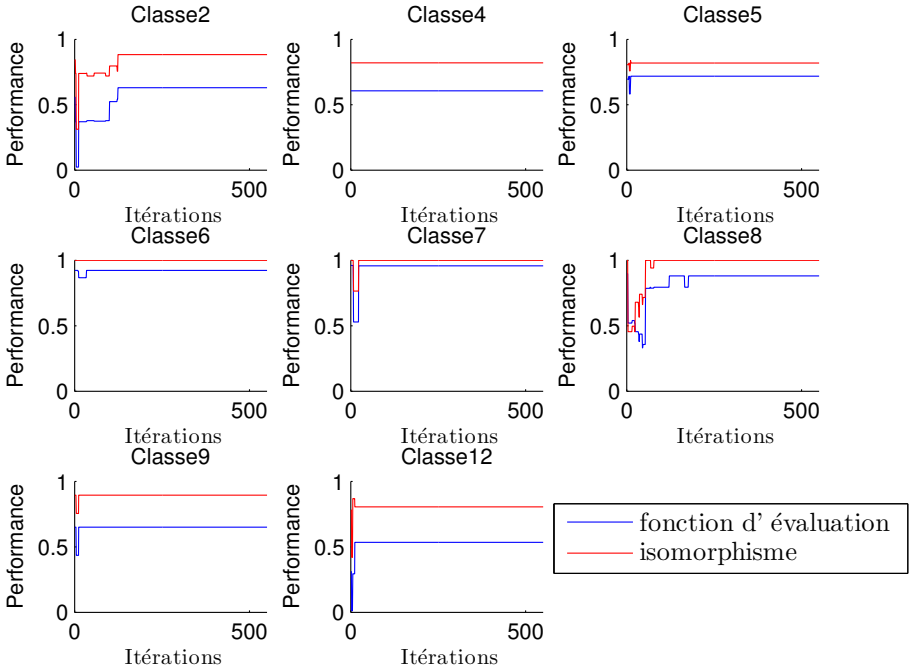


FIGURE 4.21 – Evolution de la performance en isomorphisme - Avec mémoire primée

le nombre d'itérations et les différents paramètres des opérateurs génétiques, en particulier le taux de mutation, notamment au regard de la dimension de l'espace de recherche et de la corrélation entre la fonction d'évaluation des individus et l'objectif réel.

Dans notre cadre applicatif, outre l'optimisation des paramètres de l'algorithme génétique, la piste à privilégier pour améliorer les performances nous semble être de trouver une alternative à la fonction d'évaluation que nous utilisons. La mesure de la stabilité pourrait par exemple être calculée directement à partir de la distance entre les représentations structurelles et non seulement sur le chevauchement des zones. La distance d'édition pourrait s'avérer un choix judicieux dans le sens où elle utilise le même formalisme que celui utilisé pour la recherche d'isomorphisme de sous-graphe. Le temps de calcul nécessaire au calcul de la distance d'édition peut cependant être rédhibitoire. Un compromis est donc à trouver entre temps de calcul et alignement de la

fonction d'évaluation sur l'objectif.

Enfin l'étude de l'influence des différents paramètres pourrait permettre d'envisager l'intégration de nouveaux traitements dans la chaîne, ce qui entraînerait une extension de l'espace de recherche.

Finalement, l'approche proposée permet de paramétrer de façon presque automatique la chaîne de traitement. L'analyse à postériori des différents paramétrages retenus au fil de l'évolution du système pourrait être expliquée par une expertise, voire contribuer à une extraction automatique de connaissances. De cette façon, la phase d'initialisation du processus au moment d'aborder une nouvelle classe pourrait se faire en appliquant les paramétrages les plus performants sur des classes de mêmes caractéristiques, plutôt que d'être faite de façon totalement aléatoire. Cela permettrait d'envisager une phase de convergence plus rapide pour cette nouvelle classe et des performances acceptables dès les premiers documents rencontrés.

Sous réserve de l'étude des différents paramètres, nous pourrions également envisager l'application de l'approche proposée à d'autres cadres applicatifs devant faire face à des problématiques similaires.

Chapitre 5

Conclusion Générale

Dans cette thèse, nous nous sommes intéressés à la conception d'une solution de lecture automatique de documents administratifs et commerciaux. Notre objectif applicatif était de proposer un système permettant de localiser une information à lire sur différentes instances d'une classe de formulaires. Les contraintes liées à l'application étaient les suivantes. L'approche devait être robuste à différentes sources de variabilité de l'image numérisée. L'approche devait être efficace même si la désignation de l'information à extraire n'était faite que sur un unique exemple de la classe. Enfin, l'approche devait pouvoir s'appliquer à différentes classes de formulaires sans requérir une expertise en traitement d'images, c'est-à-dire que son paramétrage devait être automatisé.

Pour relever ce défi, nous avons dans un premier temps étudié les travaux existants dans le domaine de la LAD, ce qui nous a permis d'identifier deux problématiques scientifiques à résoudre pour atteindre les objectifs applicatifs visés. La première problématique concerne la localisation robuste d'information sur une image de document à partir d'une unique instance d'apprentissage. La seconde problématique concerne l'automatisation du paramétrage d'une chaîne de traitement d'images au fil des instances rencontrées.

Pour adresser la première des deux problématiques, nous avons proposé de définir la position de l'information à extraire relativement à la structure physique du fond du formulaire. L'objectif était de s'affranchir de l'utilisation d'une étape de reconnaissance de texte pour cette tâche, qui peut s'avérer coûteuse et source d'erreur. Notre choix pour représenter cette structure physique s'est porté sur une représentation sous forme de graphes dans lequel les noeuds représentent des rectangles de couleur homogène et les arcs matérialisent des relations de visibilité, formant ainsi un graphe d'adjacence de régions. Pour construire ce graphe, qui constitue la première contribution de cette thèse, deux étapes de traitement ont été décrites dans le second chapitre du mémoire.

La première étape consiste en l'extraction de régions immuables caractérisées par une forme rectangulaire et une couleur homogène. Afin d'obtenir ces zones,

l'image, qui peut présenter des distorsions engendrées par l'impression et la numérisation, nécessite parfois des pré-traitements pour corriger ces anomalies et obtenir une bonne segmentation. Nous avons proposé un scenario générique de segmentation, qui a besoin d'être paramétré. Nos expériences menées sur une base de documents réels ont montré que l'approche permettait, si elle bien configurée pour une classe donnée, d'extraire la plupart des rectangles de couleur homogène. Elles ont aussi souligné que le comportement du système diffère d'une configuration à une autre et qu'il est nécessaire d'arriver à déterminer la bonne configuration pour chaque catégorie de document.

La deuxième étape consiste quant à elle en la construction d'un graphe avec des arcs liant les différentes régions. En effet, les nœuds seuls, même accompagnés de leurs attributs, sont insuffisants pour décrire la structure du document. Une modélisation des agencements relatifs des rectangles les uns par rapport aux autres est nécessaire pour décrire plus précisément la structure topologique du document. Dans ce mémoire, nous avons proposé de modéliser la notion de voisinage par le biais d'une relation de visibilité entre les rectangles. Ainsi, deux nœuds sont liés par un arc si les rectangles correspondant sont considérés visibles l'un de l'autre.

La seconde contribution importante de la thèse, décrite dans le chapitre 3, visait l'opérationnalisation de la représentation sous forme de graphe. À partir de la définition par un utilisateur d'une zone d'intérêt (la requête) sur l'image d'un document modèle, l'objectif était que le système puisse identifier la zone correspondante dans d'autres occurrences de la même classe de document, en maintenant les bonnes propriétés d'invariance. Pour ce faire, le système s'appuie sur des graphes pour représenter structurellement la requête correspondant à la zone d'intérêt. La zone "image" recherchée est ainsi décrite par un graphe de "petite" taille qui positionne cette zone (elle aussi rectangulaire) par rapport aux éléments constitutifs du document modèle, toujours avec la relation de visibilité. Dans ce contexte, le problème de l'identification de la zone d'intérêt dans un document inconnu consiste alors à rechercher

une occurence du graphe modèle, correspondant à la zone d'intérêt, dans un graphe cible modélisant le document à traîter. Dans ce cadre, nous avons proposé une approche reposant sur une formulation linéaire en nombres binaires, dont l'objectif est de minimiser un coût d'appariement, tout en tolérant la disparition de nœuds et d'arcs dans le graphe cible. L'approche proposée a été évaluée sur 3 bases de graphes. La première évaluation, sur une problématique de recherche de symboles dans documents graphiques, a permis de montrer l'apport des opérations de suppression. L'approche permet en effet de retrouver des occurrences de symboles distordus que les approches de la littérature ne permettent pas de trouver. La seconde évaluation a montré que cette prise en compte de modification de la topologie n'entraîne pas de surcoût sur les temps de calcul, par rapport à une approche qui ne tolère que des modification d'attributs. Enfin, la dernière évaluation, réalisée sur les documents de l'application ITESOFT, a confirmé les bonnes propriétés de l'approche proposée. Les résultats que nous avons obtenus montrent que notre système de localisation est efficace pour retrouver des zones informatives. De plus, nous avons montré qu'il est possible d'optimiser le système si nous pouvons attribuer à chaque catégorie de document un meilleur scénario d'extraction des éléments immuables.

Notre troisième contribution, plus exploratoire, concerne l'optimisation de la chaîne de traitement en vue d'une adaptation aux contraintes de chaque catégorie de documents. Nous avons proposé une approche basée sur un algorithme génétique dont l'objectif est d'optimiser les performances de localisation de zones d'information au fur et à mesure de l'apparition de nouveaux document. Le critère à optimiser n'étant pas calculable, l'algorithme génétique est guidé par une autre métrique dont nous avons montré qu'elle était corrélée. Par ailleurs, une stratégie permettant de prendre en considération un flux continu de nouveaux documents a été proposée. Les expériences montrent que le système arrive à converger vers une solution optimale et que les résultats sont corrélés avec les performances de localisation de l'information.

Même si des éléments de réponse aux différentes problématiques ont été apportées dans le cadre de cette thèse, bien des travaux restent ouverts à la suite de ce travail. Nous mentionnons dans ce qui suit les perspectives qui nous paraissent les plus importantes.

Sur notre première contribution, nous pensons que si l'approche structurelle proposée permet de lever les difficultés liées à un positionnement absolu des informations à extraire, il est important d'enrichir les graphes par d'autres éléments immuables. En effet, si l'utilisation de la couleur, et en particulier des rectangles de couleur homogène, a montré son intérêt dans nos expérimentation, elle ne permet évidemment pas de traiter tous les types de documents. Cette représentation pourrait être enrichie par la détection et la représentation des zones de texte imprimé, sans reconnaissance. Naturellement, cela nécessite de développer un système qui permet de distinguer les informations textuelles immuables des autres.

Sur notre seconde contribution, les résultats obtenus ont montré l'intérêt de la formulation en nombres binaires proposée, en particulier la tolérance aux modifications des graphes. Toutefois, les temps de traitement pour la recherche des sous-graphes sont importants, et deviennent même rédhibitoires lorsque les graphes à traiter atteignent des tailles significatives. Dans la littérature récente, de très nombreux travaux se sont intéressés à des approximations de la distance d'édition entre graphes pour réduire la complexité des approches exactes. L'une de nos pistes de travail actuelle concerne l'adaptation de notre formulation pour en réduire les temps de traitement.

Enfin, dans la troisième contribution, nous nous sommes intéressés à établir un algorithme génétique pour optimiser les paramètres de notre chaîne de traitement. Nous nous sommes dans ce cadre concentrés sur le système d'extraction des régions immuables. Naturellement, les poids utilisés dans le calcul de la distance d'édition sont également très important dans l'optimisation du systèmes et pourraient être intégrés dans l'algorithme évolutionnaire. A l'intersection des deux précédentes perspectives, remplacer le calcul de stabilité

dans l'évaluation des individus de l'algorithme génétique par un calcul de distance d'édition entre graphe pourrait fournir un indicateur plus performant pour évaluer la qualité des chaînes de traitement.

Publications de l'Auteur

Publications de l'Auteur

1. Hammami, Maroua, Pierre Héroux, and Sebastien Adam. "Extraction de zones informatives dans des images de formulaire en couleur." Colloque International Francophone sur l'Ecrit et le Document. 2014.

2. Hammami, Maroua, et al. "One-shot field spotting on colored forms using subgraph isomorphism." Document Analysis and Recognition (ICDAR), 2015 13th International Conference on. IEEE, 2015.

3. Lerouge, Julien, et al. "Minimum cost subgraph matching using a binary linear program." Pattern Recognition Letters 71 (2016) : 45-51.

4. Hammami, Maroua, et al. "Localisation automatique de champs de saisie sur des images de formulaires couleur par isomorphisme de sous-graphe." Colloque International Francophone sur l'Ecrit et le Document. 2016

Bibliographie

Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., and Susstrunk, S. (2012). Slic superpixels compared to state-of-the-art superpixel methods. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(11) :2274–2282.

Almohamad, H. and Duffuaa, S. O. (1993). A linear programming approach for the weighted graph matching problem. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 15(5) :522–525.

Amer, F. Z., El-Garhy, A. M., Awadalla, M. H., Rashad, S. M., and Abdien, A. K. (2011). A real-valued genetic algorithm to optimize the parameters of support vector machine for classification of multiple faults in npp. *Nukleonika*, 56 :323–332.

Aptoula, E. and Lefevre, S. (2007). A comparative study on multivariate mathematical morphology. *Pattern Recognition*, 40(11) :2914–2929.

Back, T. (1994). Selective pressure in evolutionary algorithms : A characterization of selection mechanisms. In *Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on*, pages 57–62. IEEE.

Baker, J. E. (1985). Adaptive selection methods for genetic algorithms. In *Proceedings of an International Conference on Genetic Algorithms and their applications*, pages 101–111. Hillsdale, New Jersey.

Baker, J. E. (1987). Reducing bias and inefficiency in the selection algorithm. In *Proceedings of the second international conference on genetic algorithms*, pages 14–21.

Barbu, E., Raveaux, R., Locteau, H., Adam, S., Héroux, P., and Trupin, E. (2006). Graph classification using genetic algorithm and graph probing application to symbol recognition. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 3, pages 296–299. IEEE.

Barnard, K., Duygulu, P., Forsyth, D., Freitas, N. d., Blei, D. M., and Jordan, M. I. (2003). Matching words and pictures. *Journal of machine learning research*, 3(Feb) :1107–1135.

Benavente, R., Vanrell, M., and Baldrich, R. (2008). Parametric fuzzy sets for automatic color naming. *JOSA A*, 25(10) :2582–2593.

Berlin, B. and Kay, P. (1991). *Basic color terms : Their universality and evolution*. Univ of California Press.

Bernard, S., Chatelain, C., Adam, S., and Sabourin, R. (2016). The multiclass ROC front method for cost-sensitive classification. *Pattern Recognition*, 52 :46 – 60.

Blaifi, S., Moulahoum, S., Colak, I., and Merrouche, W. (2016). An enhanced dynamic model of battery using genetic algorithm suitable for photovoltaic applications. *Applied Energy*, 169 :888–898.

Blickle, T. and Thiele, L. (1995). A comparison of selection schemes used in genetic algorithms. In *Gloriastrasse 35, CH-8092 Zurich : Swiss Federal Institute of Technology (ETH) Zurich, Computer Engineering and Communications Networks Lab (TIK)*.

Camillerapp, J. (2012). Utilisation des points d'intérêt pour rechercher des mots imprimés ou manuscrits dans des documents anciens. In *CIFED 2012-colloque international sur l'écrit et le document*.

Carel, E., Burie, J.-C., Courboulay, V., Ogier, J.-M., and d'Andecy, V. P. (2015). Multiresolution approach based on adaptive superpixels for administrative documents segmentation into color layers. In *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*, pages 566–570. IEEE.

Carel, E., Courboulay, V., Burie, J.-C., and Ogier, J.-M. (2013). Dominant color segmentation of administrative document images by hierarchical clustering. In *Proceedings of the 2013 ACM symposium on Document engineering*, pages 115–118. ACM.

Carton, C., Lemaitre, A., and Couasnon, B. (2015). Automatic and interactive rule inference without ground truth. In *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*, pages 696–700. IEEE.

Cesarini, F., Gori, M., Marinai, S., and Soda, G. (1998). Informys : A flexible invoice-like form-reader system. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(7) :730–745.

Chanussot, J. (1998). *Approches vectorielles ou marginales pour le traitement d'images multi-composantes.* PhD thesis.

Chatelain, C., Adam, S., Lecourtier, Y., Heutte, L., and Paquet, T. (2010). A multi-model selection framework for unknown and/or evolutive misclassification cost problems. *Pattern Recognition*, 43(3) :815–823.

Cheng, Y. (1995). Mean shift, mode seeking, and clustering. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 17(8) :790–799.

Chung, W., Chen, H., and Nunamaker, J. F. (2003). Business intelligence explorer : a knowledge map framework for discovering business intelligence on the web. In *System Sciences, 2003. Proceedings of the 36th Annual Hawaii International Conference on*, pages 10–pp. IEEE.

Coello, C. A. C. and Montes, E. M. (2002). Constraint-handling in genetic algorithms through the use of dominance-based tournament selection. *Advanced Engineering Informatics*, 16(3) :193–203.

Comaniciu, D. and Meer, P. (2002). Mean shift : A robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(5) :603–619.

Cordella, L. P., Foggia, P., Sansone, C., and Vento, M. (1999). Performance evaluation of the vf graph matching algorithm. In *Image Analysis and Processing, 1999. Proceedings. International Conference on*, pages 1172–1177. IEEE.

Cordella, L. P., Foggia, P., Sansone, C., and Vento, M. (2004). A (sub) graph isomorphism algorithm for matching large graphs. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(10) :1367–1372.

Danna, E., Fenelon, M., Gu, Z., and Wunderling, R. (2007). Generating multiple solutions for mixed integer programming problems. In *Integer Programming and Combinatorial Optimization*, pages 280–294. Springer.

DeJong, K. (1975). An analysis of the behavior of a class of genetic adaptive systems. *Ph. D. Thesis, University of Michigan*.

Delalandre, M., Valveny, E., Pridmore, T., and Karatzas, D. (2010). Generation of synthetic documents for performance evaluation of symbol recognition & spotting systems. *International Journal on Document Analysis and Recognition (IJDAR)*, 13(3) :187–207.

Demirtas, M. (2011). Off-line tuning of a pi speed controller for a permanent magnet brushless dc motor using dsp. *Energy conversion and management*, 52(1) :264–273.

Di Baja, G. S. and Thiel, E. (1996). Skeletonization algorithm running on path-based distance maps. *Image and vision Computing*, 14(1) :47–57.

Dutta, A., Lladós, J., Bunke, H., and Pal, U. (2013a). Near convex region adjacency graph and approximate neighborhood string matching for symbol spotting in graphical documents. In *2013 12th International Conference on Document Analysis and Recognition*, pages 1078–1082. IEEE.

Dutta, A., Lladós, J., and Pal, U. (2013b). A symbol spotting approach in graphical documents by hashing serialized graphs. *Pattern Recognition*, 46(3) :752–768.

Felzenszwalb, P. F. and Huttenlocher, D. P. (2004). Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2) :167–181.

Fischer, A., Suen, C. Y., Frinken, V., Riesen, K., and Bunke, H. (2013). A fast matching algorithm for graph-based handwriting recognition. In *International Workshop on Graph-Based Representations in Pattern Recognition*, pages 194–203. Springer.

Fischer, A., Suen, C. Y., Frinken, V., Riesen, K., and Bunke, H. (2015). Approximation of graph edit distance based on hausdorff matching. *Pattern Recognition*, 48(2) :331–343.

Fogel, D. B. (1994). An introduction to simulated evolutionary optimization. *IEEE transactions on neural networks*, 5(1) :3–14.

Friedrichs, F. and Igel, C. (2005). Evolutionary tuning of multiple svm parameters. *Neurocomputing*, 64 :107–117.

Frohlich, H., Chapelle, O., and Scholkopf, B. (2003). Feature selection for support vector machines by means of genetic algorithm. In *Tools with Artificial Intelligence, 2003. Proceedings. 15th IEEE International Conference on*, pages 142–148. IEEE.

Fröhlich, H., Ultsch, A., and Schölkopf, B. (2002). *Feature Selection for Support Vector Machines by Means of Genetic Algorithms*. PhD thesis, Master's

thesis, University of Marburg, 2002. http ://www-ra/informatik. unituebingen. de/mitarb/froehlich.

Fukunaga, K. and Hostetler, L. D. (1975). The estimation of the gradient of a density function, with applications in pattern recognition. *Information Theory, IEEE Transactions on*, 21(1) :32–40.

Gaceb, D., Lebourgeois, F., and Duong, J. (2013). Adaptative Smart-Binarization Method : For Images of Business Documents. *2013 12th International Conference on Document Analysis and Recognition*, pages 118–122.

Galibert, O., Kahn, J., and Oparin, I. (2014). The zonemap metric for page segmentation and area classification in scanned documents. In *Image Processing (ICIP), 2014 IEEE International Conference on*, pages 2594–2598. IEEE.

Gao, X., Xiao, B., Tao, D., and Li, X. (2010). A survey of graph edit distance. *Pattern Anal. Appl.*, 13(1) :113–129.

Garey, M. R. and Johnson, D. S. (1990). *Computers and Intractability ; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA.

Garz, A., Seuret, M., Simistira, F., Fischer, A., and Ingold, R. (2016). Creating ground truth for historical manuscripts with document graphs and scribbling interaction. In *2016 12th IAPR Workshop on Document Analysis Systems (DAS)*, pages 126–131.

Ghahraman, D. E., Wong, A. K., and Au, T. (1980). Graph optimal monomorphism algorithms. *Systems, Man and Cybernetics, IEEE Transactions on*, 10(4) :181–188.

Goldberg, D. E. and Deb, K. (1991). A comparative analysis of selection schemes used in genetic algorithms. *Foundations of genetic algorithms*, 1 :69–93.

Grefenstette, J. J. and Baker, J. E. (1989). How genetic algorithms work : A critical look at implicit parallelism. In *Proceedings of the third international conference on Genetic algorithms*, pages 20–27. Morgan Kaufmann Publishers Inc.

Hammami, M., Héroux, P., and Adam, S. (2014). Extraction de zones informatives dans des images de formulaire en couleur. In *CORIA 2014 - Conférence en Recherche d'Infomations et Applications- 11th French Information Retrieval Conference. CIFED 2014 Colloque International Francophone sur l'Ecrit et le Document, Nancy, France, March 19-23, 2014.*, pages 171–184.

Hammami, M., Heroux, P., Adam, S., and d'Andecy, V. P. (2015). One-shot field spotting on colored forms using subgraph isomorphism. In *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*, pages 586–590. IEEE.

Hamza, H., Belaid, Y., and Belaïd, A. (2007). A case-based reasoning approach for invoice structure extraction. In *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, volume 1, pages 327–331. IEEE.

Hamza, H., Belaïd, Y., Belaïd, A., and Chaudhuri, B. B. (2008). An end-to-end administrative document analysis system. In *Document Analysis Systems, 2008. DAS'08. The Eighth IAPR International Workshop on*, pages 175–182. IEEE.

Harwood, D., Subbarao, M., Hakalahti, H., and Davis, L. S. (1987). A new class of edge-preserving smoothing filters. *Pattern Recognition Letters*, 6(3) :155–162.

Ho, H. N., Rigaud, C., Burie, J.-C., and Ogier, J.-M. (2013). Redundant structure detection in attributed adjacency graphs for character detection in comics books. In *10th IAPR International Workshop on Graphics Recognition*.

Hofmann, T. (1999). Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57. ACM.

Holland, J. H. (1975). *Adaptation in natural and artificial systems : an introductory analysis with applications to biology, control, and artificial intelligence.* U Michigan Press.

Hwang, S.-F. and He, R.-S. (2006). A hybrid real-parameter genetic algorithm for function optimization. *Advanced Engineering Informatics*, 20(1) :7–21.

Janarthanan, V. and Jananii, G. (2012). A detailed survey on various image inpainting techniques. *Bonfring International Journal of Advances in Image Processing*, 2(2) :1.

Jenkins, W. (1997). On the application of natural algorithms to structural design optimization. *Engineering structures*, 19(4) :302–308.

Jouili, S., Coustaty, M., Tabbone, S., and Ogier, J.-M. (2010). Navidomass : Structural-based approaches towards handling historical documents. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 946–949. IEEE.

Julstrom, B. A. (1999). It's all the same to me : Revisiting rank-based probabilities and tournaments. In *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, volume 2. IEEE.

Justice, D. and Hero, A. (2006). A binary linear programming formulation of the graph edit distance. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(8) :1200–1214.

Kaya, İ. and Nalbantoğlu, M. (2016). Simultaneous tuning of cascaded controller design using genetic algorithm. *Electrical Engineering*, pages 1–7.

Kaya, M. (2011). The effects of two new crossover operators on genetic algorithm performance. *Applied Soft Computing*, 11(1) :881–890.

Khan, F. S., Anwer, R. M., van de Weijer, J., Bagdanov, A. D., Vanrell, M., and Lopez, A. M. (2012a). Color attributes for object detection. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3306–3313. IEEE.

Khan, F. S., Van de Weijer, J., and Vanrell, M. (2012b). Modulating shape features by color attention for object recognition. *International Journal of Computer Vision*, 98(1) :49–64.

Kumar, R. (2012). Blending roulette wheel selection & rank selection in genetic algorithms. *International Journal of Machine Learning and Computing*, 2(4) :365.

Le, T.-N., Luqman, M. M., Burie, J.-C., and Ogier, J.-M. (2015). A comic retrieval system based on multilayer graph representation and graph mining. In *International Workshop on Graph-Based Representations in Pattern Recognition*, pages 355–364. Springer.

Le Bodic, P., Héroux, P., Adam, S., and Lecourtier, Y. (2012). An integer linear program for substitution-tolerant subgraph isomorphism and its use for symbol spotting in technical drawings. *Pattern Recognition*, 45(12) :4214–4224.

Lebourgeois, F., Drira, F., Gaceb, D., and Duong, J. (2013). Fast Integral MeanShift : Application to Color Segmentation of Document Images. pages 52–56.

Lee, C.-S., Guo, S.-M., and Hsu, C.-Y. (2005). Genetic-based fuzzy image filter and its application to image processing. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 35(4) :694–711.

Lee, J.-H., Chang, B.-H., and Kim, S.-D. (1994). Comparison of colour transformations for image segmentation. *Electronics Letters*, 30(20) :1660–1661.

Lemaitre, A., Coüasnon, B., and Leplumey, I. (2005). Using a neighbourhood graph based on voronoï tessellation with dmos, a generic method for structured document recognition. In *International Workshop on Graphics Recognition*, pages 267–278. Springer.

Lerouge, J., Hammami, M., Héroux, P., and Adam, S. (2016). Minimum cost subgraph matching using a binary linear program. *Pattern Recognition Letters*, 71 :45–51.

Lerouge, J., Le Bodic, P., Héroux, P., and Adam, S. (2015). Gem++ : A tool for solving substitution-tolerant subgraph isomorphism. In *Graph-Based Representations in Pattern Recognition*, pages 128–137. Springer.

Levinshtein, A., Stere, A., Kutulakos, K. N., Fleet, D. J., Dickinson, S. J., and Siddiqi, K. (2009). Turbopixels : Fast superpixels using geometric flows. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(12) :2290–2297.

Leydier, Y., Le Bourgeois, F., and Emptoz, H. (2004). Serialized unsupervised classifier for adaptative color image segmentation : Application to digitized ancient manuscripts. *Proceedings - International Conference on Pattern Recognition*, 1 :494–497.

Lladós, J., Martí, E., and Villanueva, J. J. (2001). Symbol recognition by error-tolerant subgraph matching between region adjacency graphs. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(10) :1137–1143.

Locteau, H., Adam, S., Trupin, E., Labiche, J., and Héroux, P. (2007). Symbol spotting using full visibility graph representation. In *Workshop on Graphics Recognition*, pages 49–50.

Loo, P. K. and Tan, C. L. (2004). Adaptive region growing color segmentation for text using irregular pyramid. In *Document Analysis Systems VI*, pages 264–275. Springer.

Lorena, A. C. and De Carvalho, A. C. (2008). Evolutionary tuning of svm parameter values in multiclass problems. *Neurocomputing*, 71(16) :3326–3334.

Lozano-Pérez, T. and Wesley, M. A. (1979). An algorithm for planning collision-free paths among polyhedral obstacles. *Communications of the ACM*, 22(10) :560–570.

Luccheseyz, L. and Mitray, S. (2001). Color image segmentation : A state-of-the-art survey. *Proceedings of the Indian National Science Academy (INSA-A)*, 67(2) :207–221.

Mehri, M., Héroux, P., Lerouge, J., Gomez-Krämer, P., and Mullot, R. (2015). A structural signature based on texture for digitized historical book page categorization. In *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*, pages 116–120. IEEE.

Miller, M. T., Jerebko, A. K., Malley, J. D., and Summers, R. M. (2003). Feature selection for computer-aided polyp detection using genetic algorithms. In *Medical Imaging 2003*, pages 102–110. International Society for Optics and Photonics.

Moore, A. P., Prince, J., Warrell, J., Mohammed, U., and Jones, G. (2008). Superpixel lattices. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE.

Munteanu, C. and Rosa, A. (2000). Towards automatic image enhancement using genetic algorithms. In *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*, volume 2, pages 1535–1542. IEEE.

Munteanu, C. and Rosa, A. (2001). Color image enhancement using evolutionary principles and the retinex theory of color constancy. In *Neural Networks for Signal Processing XI, 2001. Proceedings of the 2001 IEEE Signal Processing Society Workshop*, pages 393–402. IEEE.

Nikolaou, N. and Papamarkos, N. (2009a). Color reduction for complex document images. *International Journal of Imaging Systems and Technology*, 19(1) :14–26.

Nikolaou, N. and Papamarkos, N. (2009b). Color reduction for complex document images. *International Journal of Imaging Systems and Technology*, 19(1) :14–26.

Noraini, M. R. and Geraghty, J. (2011). Genetic algorithm performance with different selection strategies in solving tsp.

Pareek, N. K. and Patidar, V. (2016). Medical image protection using genetic algorithm operations. *Soft Computing*, 20(2) :763–772.

Párraga, C., Benavente, R., Baldrich, R., and Vanrell, M. (2009). Psychophysical measurements to model intercolor regions of color-naming space. *Journal of Imaging Science and Technology*, 53(3) :31106–1.

Paulinas, M. and Ušinskas, A. (2015). A survey of genetic algorithms applications for image enhancement and segmentation. *Information Technology and control*, 36(3).

Peanho, C. A., Stagni, H., and da Silva, F. S. C. (2012). Semantic information extraction from images of complex documents. *Applied Intelligence*, 37(4) :543–557.

Perona, P. and Malik, J. (1990). Scale-space and edge detection using anisotropic diffusion. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 12(7) :629–639.

Qureshi, R., Ramel, J.-Y., Barret, D., and Cardot, H. (2007). Symbol spotting in graphical documents using graph representations. In *Workshop on Graphics Recognition (GREC)*, volume 5046, pages 91–103.

Ratnam, Y. M., Krishna, K. M., and Giribabu, P. (2015). Optimization procedure by using genetic algorithm.

Raveaux, R., Adam, S., Héroux, P., and Trupin, E. (2011). Learning graph prototypes for shape recognition. *Computer Vision and Image Understanding (CVIU)*, 115(7) :905 – 918.

Riesen, K. (2015). *Structural Pattern Recognition with Graph Edit Distance - Approximation Algorithms and Applications*. Advances in Comp. Vis. and Pattern Recogn. Springer.

Riesen, K. and Bunke, H. (2015). Improving bipartite graph edit distance approximation using various search strategies. *Pattern Recognition*, 48(4) :1349–1363.

Rosin, P. L. (2003). Measuring shape : ellipticity, rectangularity, and triangularity. *Machine Vision and Applications*, 14(3) :172–184.

Rusiñol, M., Benkhelfallah, T., and d'Andecy, V. P. (2013). Field extraction from administrative documents by incremental structural templates. In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, pages 1100–1104. IEEE.

Santos, S. G. T. C., Barros, R. S. M., and Júnior, P. M. G. (2015). Optimizing the parameters of drift detection methods using a genetic algorithm. In *Tools with Artificial Intelligence (ICTAI), 2015 IEEE 27th International Conference on*, pages 1077–1084. IEEE.

Schulz, F., Ebbecke, M., Gillmann, M., Adrian, B., Agne, S., and Dengel, A. (2009). Seizing the treasure : Transferring knowledge in invoice analysis. In *2009 10th International Conference on Document Analysis and Recognition*, pages 848–852. IEEE.

Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8) :888–905.

Shin, K.-S. and Lee, Y.-J. (2002). A genetic algorithm application in ban-

kruptcy prediction modeling. *Expert Systems with Applications*, 23(3) :321–328.

Shukla, A., Pandey, H. M., and Mehrotra, D. (2015). Comparative review of selection techniques in genetic algorithm. In *Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE), 2015 International Conference on*, pages 515–519. IEEE.

Solnon, C. (2010). Alldifferent-based filtering for subgraph isomorphism. *Artificial Intelligence*, 174(12) :850–864.

Soremekun, G., Gürdal, Z., Haftka, R., and Watson, L. (2001). Composite laminate design optimization by genetic algorithm with generalized elitist selection. *Computers & structures*, 79(2) :131–143.

Soremekun, G. A. (1997). Genetic algorithms for composite laminate design and optimization.

Teague, M. R. (1980). Image analysis via the general theory of moments*. *JOSA*, 70(8) :920–930.

Tsai, C.-M. and Lee, H.-J. (2002). Binarization of color document images via luminance and saturation color features. *Image Processing, IEEE Transactions on*, 11(4) :434–451.

Ullmann, J. R. (1976). An algorithm for subgraph isomorphism. *Journal of the ACM (JACM)*, 23(1) :31–42.

Van De Weijer, J., Schmid, C., Verbeek, J., and Larlus, D. (2009). Learning color names for real-world applications. *Image Processing, IEEE Transactions on*, 18(7) :1512–1523.

Vandenbroucke, N. (2000). *Segmentation d'images couleur par classification de pixels dans des espaces d'attributs colorimétriques adaptés. Application à l'analyse d'images de football.* PhD thesis, université des Sciences et Technologies de Lille.

Vandenbroucke, N., Macaire, L., and Postaire, J.-G. (2003). Color image segmentation by pixel classification in an adapted hybrid color space. application to soccer image analysis. *Computer Vision and Image Understanding*, 90(2) :190–216.

Vedaldi, A. and Soatto, S. (2008). Quick shift and kernel methods for mode seeking. In *Computer vision–ECCV 2008*, pages 705–718. Springer.

Veksler, O., Boykov, Y., and Mehrani, P. (2010). Superpixels and supervoxels in an energy optimization framework. In *Computer Vision–ECCV 2010*, pages 211–224. Springer.

Vincent, L. and Soille, P. (1991). Watersheds in digital spaces : an efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (6) :583–598.

Viola, P. and Jones, M. J. (2004). Robust real-time face detection. *International journal of computer vision*, 57(2) :137–154.

Wall, K. and Danielsson, P.-E. (1984). A fast sequential method for polygonal approximation of digitized curves. *Computer Vision, Graphics, and Image Processing*, 28(2) :220–227.

Wismath, S. K. (1992). Computing the full visibility graph of a set of line segments** this research was supported by nserc grant og-pin 007. *Information processing letters*, 42(5) :257–261.

Wong, A. K., You, M., and Chan, S. (1990). An algorithm for graph optimal monomorphism. *Systems, Man and Cybernetics, IEEE Transactions on*, 20(3) :628–638.

Wu, S.-J. and Chow, P.-T. (1995). Steady-state genetic algorithms for discrete optimization of trusses. *Computers & Structures*, 56(6) :979–991.

Yang, C.-L., Kuo, R., Chien, C.-H., and Quyen, N. T. P. (2015). Non-dominated sorting genetic algorithm using fuzzy membership chromosome for categorical data clustering. *Applied Soft Computing*, 30 :113–122.

Zhong, J., Hu, X., Zhang, J., and Gu, M. (2005). Comparison of performance between different selection strategies on simple genetic algorithms. In *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)*, volume 2, pages 1115–1121. IEEE.

Table des figures

Liste des tableaux