Détection de textes enfouis dans des bases d'images généralistes : un descripteur sémantique pour l'indexation

Thomas Retornaz

Collège doctoral
ED nº 431 : Information, communication, modélisation et simulation

Nº attribué par la bibliothèque

THÈSE

pour obtenir le grade de
Docteur de l'École des Mines de Paris
Spécialité « Morphologie Mathématique »

Présentée et soutenue publiquement par

Thomas RETORNAZ

le 23 Octobre 2007

Détection de textes enfouis dans des bases d'images généralistes. Un descripteur sémantique pour l'indexation.

Directeur de thèse : Beatriz MARCOTEGUI ITURMENDI

Jury :

Dominique Jeulin	*Président*
Jean-Marie Becker	*Rapporteur*
Jean-Michel Jolion	*Rapporteur*
Matthieu Cord	*Examinateur*
Jorg Kruger	*Examinateur*
Livier Reithler	*Examinateur*
Beatriz Marcotegui Iturmendi	*Examinateur*

La science ne sert guère qu'à nous donner une idée de l'étendue de notre ignorance.

Extrait de A la découverte de la science
FÉLICITÉ DE LAMENNAIS

Douter de tout ou tout croire sont deux solutions également commodes, qui l'une et l'autre nous dispensent de réfléchir.

Extrait de La science et l'hypothèse
HENRI POINCARÉ

Limites de notre ouïe - On n'entend que les questions auxquelles on est en mesure de trouver une réponse.

FRIEDRICH WILHELM NIETZSCHE

L'homme de science le sait bien, lui, que seule la science, a pu, au fil des siècles, lui apporter l'horloge pointeuse et le parcmètre automatique sans lesquels il n'est pas de bonheur terrestre possible.

Extrait de Vivons heureux en attendant la mort
PIERRE DESPROGES

Remerciements

Je remercie Dominique Jeulin qui m'a fait l'honneur de présider ce jury.

Je remercie Jean-Marie Becker et Jean-Michel Jolion qui ont bien voulu accepter la charge de rapporteur. Ainsi que mes trois examinateurs Matthieu Cord, Jorg Kruger et Livier Reithler, pour la qualité et la pertinence de leurs remarques sur ce manuscrit et au cours de la soutenance.

Je remercie Fernand Meyer de m'avoir accueilli au sein du laboratoire CMM. Je remercie également Beatriz Marcotegui Iturmendi d'avoir accepté de m'encadrer et d'avoir réussi à tenir bon face à mon pessimisme notoire.

Un grand merci également à toute l'équipe du laboratoire CMM. L'énumération des personnes que j'aimerais remercier pour leur support technique, leur gentillesse, leur amitié serait fastidieuse et j'aurais peur d'oublier par mégarde quelqu'un. Aussi je me contenterais de dire que tous les soutiens reçus m'ont été d'une aide irremplaçable. Ce travail n'aurait pas pu aboutir sans eux.

Enfin un très grand merci à ma petite Sun, pour ses encouragements et son soutien pendant cette thèse.

Résumé

Titre :Détection de textes enfouis dans des bases d'images généralistes. Un descripteur sémantique pour l'indexation.

Les bases de données multimédia, aussi bien personnelles que professionnelles, se développent considérablement et les outils automatiques pour leur gestion efficace deviennent indispensables. L'effort des chercheurs pour développer des outils d'indexation basés sur le contenu sont très importants, mais le fossé sémantique est difficile à franchir : les descripteurs de bas niveau généralement utilisés montrent leurs limites dans des cadres applicatifs de plus en plus ouverts. Le texte présent dans les images est souvent relié au contexte sémantique et constitue un descripteur pertinent.

Dans cette thèse nous proposons un système de localisation de texte dans des bases d'images génériques, qui tend à être robuste au changement d'échelle et aux déformations usuelles du texte enfoui. Notre système est basé sur un opérateur résiduel numérique, l'ouvert ultime. Dans une première partie nous étudions le comportement de cet opérateur sur des images réelles, et proposons des solutions pour pallier certaines limitations. Dans une seconde partie l'opérateur est inclus dans une chaîne de traitement et complété par différents outils de caractérisation du texte.

Les performances de notre approche sont évaluées sur deux bases d'images. Premièrement, nous avons pris part à la campagne d'évaluation ImagEval, remportant la première place dans la catégorie "localisation de texte". Deuxièmement pour situer notre approche par rapport à l'état de l'art, nous avons effectué des tests avec la base d'évaluation I.C.D.A.R. Enfin, un démonstrateur a été réalisé pour EADS. Pour des raisons de confidentialité, ces travaux n'ont pas pu être intégrés à ce manuscrit.

Mots clés : TEXTE ENFOUI, OPERATIONS RESIDUELLES NUMERIQUES, LOCALISATION, INDEXATION, DESCRIPTEUR DE HAUT NIVEAU, CAMPAGNE D EVALUATION, MORPHOLOGIE MATHEMATIQUE

Abstract

Title: Automatic detection of text from natural scenes.A semantic descriptor for content based image retrieval

Multimedia data bases, both personal and professional, are continuously growing and the need for automatic solutions becomes mandatory. Effort devoted by the research community to content-based image indexing is also growing, but the semantic gap is difficult to cross: the low level descriptors used for indexing are not efficient enough for an ergonomic manipulation of big and generic image data bases. The text present in a scene is usually linked to image semantic context and constitutes a relevant descriptor for content-based image indexing.

In this thesis we present an approach to automatic detection of text from natural scenes, which tends to handle the text in different sizes, orientations, and backgrounds. The system uses a non linear scale space based on the ultimate opening operator (a morphological numerical residue). In a first step, we study the action of this operator on real images, and propose solutions to overcome these intrinsic limitations. In a second step, the operator is used in a text detection framework which contains additionally various tools of text categorisation.

The robustness of our approach is proven on two different dataset. First we took part to ImagEval evaluation campaign and our approach was ranked first in the text localisation contest. Second, we produced result (using the same framework) on the free ICDAR dataset, the results obtained are comparable with those of the state of the art. Lastly, a demonstrator was carried out for EADS. Because of confidentiality, this work could not be integrated into this manuscript.

Keywords: SCENE TEXT, MORPHOLOGICAL NUMERICAL RESIDUE, LOCALIZATION, INDEXATION, HIGH LEVEL DESCRIPTOR, EVALUATION CAMPAIGN, MATHEMATICAL MORPHOLOGY

Table des matières

Remerciements **i**

Résumé **iii**

Abstract **v**

Table des matières **vii**

Liste des tableaux **x**

Table des figures **xi**

1 Introduction **1**
1.1 Motivation : fournir un descripteur de haut niveau pour la recherche d'image par le contenu . . . 1
1.2 Restriction de la problématique initiale et contexte de l'étude . . . 2
1.3 Plan de ce manuscrit . . . 2

I Problématique générale **5**

2 Problématique de l'indexation et apport de l'information textuelle : **9**
2.1 Définitions préliminaires . . . 9
2.2 Généralités sur l'indexation automatique d'images . . . 10
2.3 Apport de l'information textuelle : . . . 14
2.4 Qu'attend-on d'un système d'extraction de texte (*SET*) . . . 15

3 Comment appréhender le texte dans les images et les vidéos **17**
3.1 Définitions . . . 17
3.2 Propriétés du texte dans les images et les vidéos . . . 18
3.3 Conclusion . . . 23

4 Extraction de l'information textuelle : Revue des méthodes **25**
4.1 Détection des zones potentielles de texte . . . 25
4.2 Localisation des zones de texte . . . 26
4.3 Restauration . . . 34

4.4 Extraction 36
4.5 Critère d'évaluation 36
4.6 Conclusion 40

5 Premiers pas
Réflexion sur un existant **41**
5.1 Introduction de la problématique 41
5.2 Présentation de l'algorithme 41
5.3 Discussion sur la généralisation de l'algorithme 46
5.4 Conclusion 47

II Opérateurs Résiduels Numériques **51**

6 Présentation, définition et illustration des opérateurs résiduels numériques **55**
6.1 Transformations Résiduelles Numériques 55
6.2 Un cas particulier, l'ouverture ultime : 57
6.3 Extension de l'ouverture ultime à d'autres familles de critères : 59

7 Illustration de l'action de l'opérateur sur des images réelles **63**
7.1 Utilisation de l'ouverture/fermeture ultime : approche directe 63
7.2 Myopie de l'opérateur pour des structures imbriquées 68
7.3 Myopie de l'opérateur pour des zones dites de «transitions graduelles» 73
7.4 Quelques résultats satisfaisants de l'approche directe 79
7.5 Conclusion de l'approche directe 79
7.6 Utilisation de l'opérateur sur des images «réelles» : approche gradient 79
7.7 Quelques résultats satisfaisants de l'approche gradient 83
7.8 Conclusion de l'approche gradient 83
7.9 Conclusion 85

8 Implantation efficace de l'opérateur d'Ouverture Ultime **89**
8.1 Approche basique : 89
8.2 Réflexion pour la conception d'un algorithme efficace pour l' ouverture/fermeture ultime par critère : 90
8.3 Détails de l'implémentation proposée : 99
8.4 Modifications des algorithmes précédents pour la prise en compte des accumulations : 110
8.5 Détails de l'implémentation proposée : 113
8.6 Tests de performance et complexité 118
8.7 Conclusion 124
8.8 Perspectives 124

III Chaînes de traitements et résultats **127**

9 Chaînes de traitement **131**
9.1 Hypothèses 131
9.2 Discussion sur les pré-filtrages possibles 132
9.3 Modules de l'Approche Transformée 134

9.4 Modules de l'Approche Indicatrice 145
9.5 Méthode de filtrage par apprentissage 148
9.6 Regroupement itératif des composantes 176
9.7 Intégration d'un OCR ? 182
9.8 Conclusion 183

10 Présentation des bases traitées et résultats **185**
10.1 Rappels sur les méthodes d'évaluation 185
10.2 Campagne d'évaluation ImagEval 191
10.3 Position de nos travaux, Traitement de la base ICDAR 206
10.4 Discussion Générale 211
10.5 Quelques résultats préliminaires de reconnaissance 212

11 Conclusion Générale **215**
11.1 Apport de cette Thèse 215
11.2 Perspectives 216

A Glossaire **219**

Références **223**
. 223

Liste des tableaux

2.1 Comparaison des domaines *étendu* et *restreint* dans le cadre de la recherche d'images . . 13

7.1 Récapitulatif :forces et faiblesses des approches directe et gradient 85

9.1 Approche Transformée. Quantification des modules de filtrage *grossier*. 144
9.2 Comparaison des estimateurs de l'épaisseur sur des lettres *complexes* de la base annotée. 153
9.3 Comparaison des estimateurs de l'épaisseur sur des lettres *de bonne qualité* de la base annotée 154
9.4 Description du jeu de paramètres I. 161
9.5 Description du jeu de paramètres II. 162
9.6 Description du jeu de paramètres III. 162
9.7 Table de confusion du système d'apprentissage : jeu de paramètres I. 162
9.8 Table de confusion du système d'apprentissage : jeu de paramètres II. 162
9.9 Table de confusion du système d'apprentissage : jeu de paramètres III. 163

10.1 Paramètres utilisés pour la chaîne transformée. Campagne de test officiel d'ImagEval. . . 194
10.2 Paramètres utilisés pour la chaîne indicatrice. Campagne de test officiel d'ImagEval . . . 195
10.3 Performances globale de la chaîne transformée ($tr = 0.6\,;\, tp = 0.4$) 195
10.4 Performances globale de la chaîne indicatrice ($tr = 0.6\,;\, tp = 0.4$) 195
10.5 Distribution du nombre d'images localisées avec $WolfPonctual_{Hmean} \geq 0.8$ 196
10.6 Chaîne Transformée 201
10.7 Chaîne Indicatrice 201
10.8 Temps de calcul de nos chaînes de traitement : les résultats sont moyennés sur une vingtaine d'images (taille moyenne 1024*655) de la base officielle $imagEval$. Le test a été réalisé sur un Pentium©Core Duo cadencé à 2.4 Ghz et disposant de 2 Go de RAM. . . 201
10.9 Performances globales de la chaîne transformée sur la base ICDAR ($tr = 0.8\,;\, tp = 0.4$) 208
10.10 Performances globales de la chaîne indicatrice sur la base ICDAR ($tr = 0.8\,;\, tp = 0.4$) . 208

Table des figures

2.1 Représentation schématique d'un système d'indexation. 11
2.2 Exemple de CAPTCHA utilisant des images : «Trouver l'élément commun de ces images». Les caractéristiques primaires extraites des images ne peuvent seules coder le contenu sémantique des images. (Credit : `http://www.captcha.net/captchas/pix/`) . 13
2.3 Indexation basée sur des mots clés extraits de légendes ou de texte enfoui. 15
2.4 Descripteur associé à un texte en vue de son indexation [23]. 15
2.5 Architecture d'un système d'extraction de texte. 16

3.1 Définition d'un texte plongé dans un environnement tri-dimensionnel 18
3.2 Texte en sur-impression : Images issues des bases de tests de Demarty [22] et Wolf [110]. On notera la faible définition, la variabilité des tailles des fontes, ... 19
3.3 Texte Enfoui : Images issues de la compétition de localisation de texte ICDAR [53]. On notera la présence de fontes complexes, de problèmes d'illumination, de perspective, ... 19
3.4 Variabilité de la hauteur des fontes dans une image publicitaire (Base [110]) 20
3.5 Effet de l'anti-aliasing : le texte est perceptuellement agréable bien que fortement dégradé.(Base [110]) 21
3.6 La présence de fond complexe rend caduques les approches simples de binarisation. (Base [110]) 21
3.7 Différentes écritures : Latine, Arabe et idéophonographique. 21

4.1 Exemple didactique. Schéma algorithmique type d'un système de localisation de texte utilisant l'apprentissage 31
4.2 En ligne continue la boîte de la vérité terrain, en pointillé la boîte détectée. Pour les deux cas (a) et (b) les mesures de PRÉCISION et RAPPEL sont équivalentes 39
4.3 En ligne continue la/les boîte(s) de la vérité terrain, en pointillé la/les boîte(s) détectée(s). Bien que dans les deux cas (a) et (b) le résultat de la détection soit perceptuellement correct, on dépend de la granularité de la vérité terrain. Dans les deux cas on obtient un mauvais score de localisation. 39

5.1 Petit rappel de typographie : importance de la ligne de base dans les fontes romanes . . . 42
5.2 Étapes de l'algorithme de détection de texte dans les images clés de Demarty [22]. . . . 44
5.3 Exemples de détections de texte pour différentes images issues de la base de Demarty [22]. De gauche à droite : image initiale, résultat de l'algorithme, seuillage *entropique*(cf Pun [69]) de ce résultat. 45

5.4 On ne peut pas se passer d'une approche multi-résolution pour prendre en compte la variabilité des textes dans les images. Pour les textes en surimpression au sein d'une même image en haut comme pour les textes enfouis issus d'une même base en bas. . . . 47

6.1 Ouverture Ultime par des segments sur un profil de ligne 58
6.2 Exemples de caractéristiques d'un ensemble connexe permettant la définition de critères croissants et planaires. 61

7.1 Illustration de l'action de l'ouvert ultime, sur une image grossièrement biphasée. La famille de transformations utilisée est composée d'ouvertures morphologiques par des boules (i.e. hexagone) de tailles croissantes. 64
7.2 L'utilisation d'ouvertures ultimes basées sur des familles d'ouvertures morphologiques (ici utilisation d'ouvertures par des boules de tailles croissantes) est mal adaptée à la caractérisation des lettres. 65
Ouverture ultime avec critère surfacique. 66
Ouverture ultime avec critère d'élongation. 66
Ouverture ultime avec critère de hauteur. 66
Ouverture ultime avec critère de largeur. 66
7.3 Présentation sur une image réelle de l'utilisation de l'ouverture ultime pour les critères surfacique, d'élongation, de hauteur et de largeur. 66
7.4 L'utilisation du critère surfacique est à proscrire. Dans cet exemple R_θ et q_θ sont constantes (voire texte). Base d'image [53]. 68
7.5 Myopie de l'ouvert ultime pour des structures imbriquées 69
Fermeture ultime par hauteur, sans arrêt de l'algorithme. 71
Arrêt de l'algorithme avec λ égale à la moitié de la hauteur l'image. 71
Arrêt de l'algorithme avec λ égale au quart de la hauteur de l'image. 71
7.6 Présentation sur une image réelle des conséquences de la myopie de l'opérateur pour les structures imbriquées 71
Premier niveau de hiérarchie 72
Second niveau de hiérarchie 72
7.7 Illustration des deux premiers niveaux d'une hiérarchie de résidus utilisant des ouvertures ultimes surfaciques. 72
7.8 Myopie de l'ouvert ultime pour des transitions graduelles 73
7.9 Correction partielle de la myopie de l'ouvert ultime pour des transitions graduelles . . . 74
Fermeture ultime avec critère d'élongation sans arrêt de l'algorithme. 76
Fermeture ultime avec critère d'élongation sans arrêt de l'algorithme avec accumulation des résidus pour des transitions de tailles unitaires. 76
7.10 Comparaison pour une image réelle de l'utilisation de la Fermeture Ultime pour le critère d'élongation avec et sans utilisation de l'accumulation. Les transitions prises en compte sont de tailles unitaires. 76
7.11 Suivi des résidus dans les lettres "P" et "R" 77
7.12 Seconde stratégie de correction de la myopie de l'ouvert ultime pour des transitions graduelles : on propage l'accumulation à l'ensemble de la zone de transition 78
7.13 L'accumulation ne résout pas pour autant les problèmes de structures imbriquées 78

7.14 Approche Directe :Utilisation de l'opérateur de fermeture ultime par hauteur. Présentation de segmentation *satisfaisante* en l'absence de tout autre traitement. De gauche a droite : image originale, canal de luminance pour une polarité du texte, R_θ (Correction $\gamma = 3$), q_θ labellisée. Base ICDAR [53] 80
Fermeture ultime avec critère de d'élongation. 81
Fermeture ultime avec critère de Hauteur. 81
Fermeture ultime avec critère de Largeur. 81
7.15 Opérateur de fermeture ultime pour différents critères. Application sur le gradient de luminance 81
7.16 Le problème des structures imbriquées est bien sûr toujours présent pour l'approche gradient. (critère utilisé : hauteur) 82
7.17 En présence transition(i.e. frontière floue), l'application d'un gradient morphologique de taille unitaire suivi de l'utilisation de l'opérateur de fermeture ultime (ici utilisant un critère de hauteur) n'est pas appropriée. 83
Approche directe sans accumulation (Critère Hauteur). 84
Approche directe avec accumulation (Critère Hauteur). 84
7.18 Dans le cas de transition graduelle. L'approche directe, utilisant la stratégie d'accumulation, est plus exploitable. 84
7.19 L'approche *Gradient* appliquée brutalement n'est pas adaptée aux *petites lettres* (Critère utilisé :Hauteur). 85
7.20 Approche Gradient :utilisation de l'opérateur de fermeture ultime par hauteur. Présentation de segmentation *satisfaisante* en l'absence de tout autre traitement . De gauche à droite : image originale, gradient morphologique sur le canal luminance (Correction $\gamma = 3$), R_θ (Correction $\gamma = 3$), q_θ labellisée .Base ICDAR [53] 86

8.1 Détection d'un résidu engendré par des fermetures surfaciques de tailles croissantes . . 91
8.2 Étapes de l'implémentation par inondation de la fermeture ultime par critère surfacique . 93
8.3 Fermeture ultime utilisant le critère de largeur 93
8.4 Fusion des lacs et fermeture ultime utilisant le critère de surface 94
8.5 Premier cas de figure : résultat erroné 95
8.6 Second cas de figure : Résultat erroné 96
8.7 Résultat correct 97
8.8 Exemple de problèmes lors de la fusion de lacs pour le critère de hauteur 97
8.9 Exemple d'arrêt de l'opérateur de fermeture ultime utilisant le critère de hauteur 99
8.10 Fermeture ultime surfacique (avec un pas d'ouverture unitaire) par inondation prenant en compte l'accumulation. De gauche à droite l'image inondée, R_θ, q_θ et les deux images auxiliaires permettant la prise en compte de l'accumulation I_Σ, I_{indice}. De haut en bas les différentes étapes de l'inondation (avec sur la première colonne : en trait gras le niveau d'inondation, et en grisé les structures de l'image pouvant participer à l'accumulation). . 111
Temps de calcul : image 512x512, 2 minima, 457 niveaux de gris 120
Temps de calcul : image 512x512, 2000 minima, 22 niveaux de gris 120
8.11 Temps de calcul (images 512x512) des algorithmes en fonction du seuil d'arrêt sur les images synthétiques. Les temps des algorithmes avec et sans prise en compte de l'accumulation sont respectivement en pointillé et trait plein. Les figures a) et b) montrent le temps de calcul pour les critères de surface et de hauteur pour une image comprenant 2 minima et 457 teintes de gris. Les figures c) et d) pour une image comprenant 2000 minima et 22 niveaux de gris. 120

Temps de calcul : image 1600x1200, Critère Surfacique 121
Temps de calcul : image 1600x1200, Critère de Hauteur 121
8.12 Comparaison des temps de calcul des algorithmes fermetures ultimes et de fermetures par critère en fonction du seuil d'arrêt sur des images réelles. Le temps est moyenné sur 38 images (1600x1200) de la Base ICDAR et le calcul est effectué sur le gradient de luminance. 121
8.13 Temps de calcul des algorithmes en fonction de la taille des images pour une même densité de minima. Le seuil d'arrêt de l'algorithme est toujours fixé à sa valeur maximale. . . . 122
8.14 Configuration pour laquelle les algorithmes 8.3.2 et 8.5.1 sont quadratiques 123

9.1 Exemple type de texte non prise en compte par nos approches. Base $imagEval$ 133
9.2 Présentation des chaînes de traitement Transformée et Indicatrice 135
9.3 Module de seuillage de l'image transformée R_θ 136
9.4 Illustration du seuillage de la transformée.Ouverture ultime par hauteur avec arret au tiers de l'image. Cas 1 : l'indicatrice n'est pas utilisable en l'état. Le seuillage «bas» la transformée nous permet de récupérer des composantes connexes exploitables.Base ImagEval 137
9.5 Illustration du seuillage de la transformée.Ouverture ultime par hauteur avec arrêt au tiers de l'image. Cas 2 : l'indicatrice n'est pas utilisable en l'état. Le seuillage «bas» connecte les lettres avec le fond. L'utilisation du seuillage adaptatif permet de récupérer des composantes connexes exploitables. Base ImagEval 138
9.6 Architecture du module de filtrage grossier. 140
Perte d'information pour des zones de texte proches de zone texturés de l'image. 142
Perte d'information pour des zones de texte présentes sur un fond texturé. 142
9.7 Exemple de cas pathologiques où le filtrage grossier basé sur la cohérence spatiale supprime des zones de texte. Base $imagEval$ 142
9.8 Exemple de l'apport du filtrage grossier sur une image réelle. Base $imagEval$. 143
9.9 Métrique pour le filtrage grossier : En noir les composantes lettres qui ne sont pas détruites. En bleu les cas correctement traités : la lettre supprimée est comptée comme un faux positif, la composante non lettre et non supprimée est comptée comme un faux négatif. En rouge les biais de la métrique : la composante non lettre est comptée comme un faux positif, la lettre agrégée avec une autre composante n'est pas comptabilisée. 143
9.10 Différents types de valeurs perchées de l'image q_θ. 145
Ouverture Ultime par hauteur avec arrêt de l'opérateur au tiers de l'image. 147
Illustration de la suppression des valeurs perchées 147
Illustration de la suppression des petites composantes assimilées à du bruit 147
Illustration de la suppression de composantes de trop faible contraste 147
9.11 Illustrations des modules de filtrage de l'indicatrice. Module de suppression des valeurs perchées, filtrage des petites composantes parasites et filtrage sur le contraste. 147
9.12 Architecture du module de sélection des composantes par apprentissage. 148
9.13 Instabilité du maximum de la fonction distance comme estimation de l'épaisseur sur quelques lettres issues de l'annotation de la base ImagEval. 150
9.14 Exemple de lettre pour lesquelles les estimations de l'épaisseur et de la cohérence de celle-ci sont problématiques (Voire Tableau 9.2). 151
9.15 Utilisation du squelette par points d'ancrage (4-connexe) pour la détermination de l'épaisseur et de la consistance de l'épaisseur d'un ensemble. 152
9.16 Procédure d'apprentissage et validation. 160
9.17 Matrice de corrélation pour le jeu de descripteur 1 164

9.18 Projection des données «lettre», «non lettre» par couple de descripteurs (1/4 : première figure sur quatre). 166
9.19 Projection des données «lettre», «non lettres» par couple de descripteurs (2/4). 167
9.20 Projection des données «lettre», «non lettre» par couple de descripteurs (3/4). 168
9.21 Projection des données «lettre», «non lettre» par couple de descripteurs (4/4). 169
9.22 Contribution de chaque axe de l'A.C.P à la variance totale du jeu de données 170
9.23 Contribution des paramètres à la variance contenue dans les 15 premiers axes de l'A.C.P. 171
9.24 Contribution des descripteurs dans la fonction de prédiction du classifieur, pour le jeu de paramètres I, en utilisant l'expansion quadratique. 172
9.25 Projection des données «lettre», «non lettres» par couples pour les cinq descripteurs contribuant le plus à la fonction de discrimination. 173
9.26 Score du meilleur classifieur. Jeux de paramètres I avec expansion quadratique 174
9.27 Présentation de l'apport du filtrage par apprentissage (Premier jeu de descripteurs). Présentation de différents cas de figure : en haut filtrage satisfaisant, au milieu des faux positifs persistants, en bas un faux négatif. (Images issues de la campagne *Test à blanc* $imagEval$). 175
9.28 Schéma récapitulatif du module itératif de regroupement des composantes connexes. . . 177
9.29 Caractéristiques géométriques utilisées pour la fusion de boîtes 177
9.30 Justification de l'étape d'agrégation de nouvelles composantes à partir des premiers germes agrégés (voir texte) : le carré bleu représente le germe, les boîtes englobantes en trait pointillé bleu représentent des boîtes englobantes de lettres que l'on souhaite agréger, les boîtes englobantes en trait pointillé rouge représentent des boîtes non lettres qui ne doivent pas être agrégées. 180
9.31 Illustration de différentes étapes du module de regroupement itératif des composantes. . 181
9.32 Non robustesse d'un O.C.R donné sur une image de référence. Texte blanc sur fond noir pour trois fontes et deux tailles. O.C.R utilisé *Tesseract*© 183

10.1 Les différents types d'appariement entre boîtes issues de la vérité terrain (ligne pointillée) et boîtes localisées (ligne pleine). Illustration provenant de [111], utilisant un panagramme bien connu. 188
10.2 Présentation de la variabilité des types de textes rencontrés au sein de la campagne d'évalution ImagEval 192
10.3 Présentation de la granularité de l'annotation fournie par le comité d'$imagEval$. 193
10.4 Critère de fusion de boîtes défini par le comité d'$imagEval$, pour la création de la vérité terrain. 194
Chaîne transformée 197
Chaîne Indicatrice 197
10.5 Diagramme de performance sur la base officielle d'ImagEval. Colonne de gauche : variation de la contrainte de rappel tr (avec tp constant et égal à 0.4) ; Colonne de droite : variation de la contrainte de précision tp (avec tr constant et égal à 0.6). 197
10.6 Influence du paramètre S, Seuil bas de la transformée. Pour la chaîne "Transformée" et la chaîne "Indicatrice". Métrique utilisé $WolfOverall\ tr = 0.6\ tp = 0.4$ (cf. 190). 198
10.7 Pour S fixé à 9, influence du paramètre $Stop$, la valeur d'arrêt de l'ouvert ultime exprimée en pourcentage de la taille de l'image. Pour la chaîne "Transformée" et la chaîne "Indicatrice". Métrique utilisée $WolfOverall\ tr = 0.6\ tp = 0.4$ (cf. 190). 199

10.8 Pour S et $Stop$ fixés respectivement à 9 et 30% ; influence du paramètre $Size_{Expand}$, extension des germes en pixels. Pour la chaîne Transformée et la chaîne Indicatrice. Métrique utilisée $WolfOverall\ tr = 0.6\ tp = 0.4$ (cf. 190). 199
Exemples de résultats de localisation utilisant l'approche Indicatrice (Première partie). De gauche à droite : vérité terrain, nos résultats de localisation, évaluation des performances (Métrique ICDAR (cf. p. 186), $WolfPonctual$ (cf. p. 187)) 201
Exemples de résultats de localisation utilisant l'approche Indicatrice (Seconde partie). De gauche à droite : vérité terrain, nos résultats de localisation, évaluation des performances (Métrique ICDAR (cf. p.186), $WolfPonctual$ (cf. p. 187)) 202
10.9 Exemples de résultats de localisation utilisant l'approche Indicatrice (Troisième partie). De gauche à droite : vérité terrain, nos résultats de localisation, évaluation des performances (Métrique ICDAR (cf. p.186), $WolfPonctual$ (cf. p. 187)) 203
10.10Résultats complémentaires de localisation. Images issues de la base de test à blanc ImagEval. En bleu nos résultats de localisation, en rouge mise en lumière des faux positifs et faux négatifs 204
10.11Présentation de la base ICDAR et de la finesse de son annotation. 207
10.12Base ICDAR : Influence du paramètre S, Seuil bas de la transformée. Pour la chaîne Transformée et la chaîne Indicatrice. Métrique utilisée $WolfOverall\ tr = 0.8\ tp = 0.4$ (cf Section 10.1.2.4). 209
10.13Résultats qualitatifs de localisation sur la base ICDAR 210
10.14Quelques résultats de *Tesseract*©sur la base *ImagEval* ; Gauche :Image Originale ; Milieu : ensemble des composantes connexes annotées texte (Pour les deux polarités du texte) par une chaîne de traitement, Droite : Résultat de l'O.C.R sous forme de chaînes de caractères. 214

Liste des Algorithmes

5.1 Algorithme d'extraction de texte proposé par Demarty [22]. 43
8.1 Implantation générique de l'ouverture ultime 90
8.2 Structure SupportCritère 100
8.3 Algorithme de fermeture ultime par critère :Initialisation 101
8.4 Algorithme de fermeture ultime par critère : Propagation 102
8.5 CroissanceLac(prio-ws, Label, pixel) 103
8.6 FusionDeLacs(prio-ws,Label1,Label2) 104
8.7 MiseAJourResidus(prio-ws,Label,ValeurDuCritere) 104
8.8 TrouverLacReprésentant(Label) 105
8.9 Post-Traitement() 105
8.10 Implantation générique de l'Ouverture Ultime Avec Accumulation 112
8.11 Algorithme de fermeture ultime par critère avec accumulation 115
8.12 MiseAJourResidus(prio-ws,Label,ValeurDuCritere) 116
8.13 Post-Traitement() 117

1 Introduction

> Le système général des sciences et des arts est une espèce de labyrinthe, de chemin tortueux, où l'esprit s'engage sans trop connaître la route qu'il doit tenir.
>
> JEAN LE ROND D' ALEMBERT

Dans ce chapitre nous présenterons le cadre général de notre travail. Nous introduirons tout d'abord son intérêt pour le domaine de la recherche d'image par le contenu. Puis nous exposerons plus précisément dans quels cadres scientifiques et industriels nous avons fait évoluer nos travaux. Enfin nous décrirons la structure de ce manuscrit

1.1 Motivation : fournir un descripteur de haut niveau pour la recherche d'image par le contenu

Les bases de données multimédia, aussi bien personnelles que professionnelles, se développent considérablement et les outils automatiques pour leur gestion efficace deviennent indispensables. Cette remarque est certes un lieu commun, mais le problème n'a jamais été aussi criant qu'aujourd'hui ; de plus il ne pourra que croître dans le futur.

L'effort des chercheurs pour développer des outils d'indexation basés sur le contenu sont très importants, mais le fossé sémantique est difficile à franchir : les descripteurs de bas niveau généralement utilisés (eg : points d'intérêt, descripteurs de texture) montrent leurs limites dans des cadres applicatifs de plus en plus ouverts.

Une des pistes actuelles est de se tourner vers des descripteurs dits de «haut niveau», qui véhiculent (au moins partiellement) une véritable information sémantique sur le contenu de l'image.

Le texte présent dans une image qu'il soit enfoui ou en sur-impression est un exemple type de descripteur de haut niveau. En effet, il est généralement relié directement au contenu de la scène : il peut décrire celle-ci dans le cas du texte en sur-impression, ou apporter des informations clés dans le cas du texte enfoui (eg : nom de rues, enseignes). Notons que dans ce dernier cas nous ne pourrons pas prédire si ce texte aura un sens du point de vue de tout utilisateur.

1.2 Restriction de la problématique initiale et contexte de l'étude

Les constatations précédentes ont fait naître un intérêt croissant des chercheurs pour la localisation et la reconnaissance de texte dans les images et les vidéos. De nombreuses avancées ont été réalisées, tout particulièrement dans le domaine de la recherche de texte en sur-impression avec l'apparition de premiers systèmes complets (i.e. permettant d'extraire directement des mots clés à partir de séquences d'images), voir [110, 11, 32].

Cependant la recherche de texte enfoui et/ou de texte en sur-impression *complexe* reste quant à elle trop souvent cloisonnée à des bases de petites tailles au contenu maîtrisé. Or il existe actuellement une forte demande des détenteurs de grands fonds photographiques, pour des systèmes capables de traiter des bases hétérogènes et de grands volumes. Aussi on voit fleurir aujourd'hui des campagnes d'évaluation comme celles organisées au sein du réseau d'excellence Muscle. Le but de celles-ci est la mise en compétition de technologies issues du monde académique sur des bases communes et représentatives des attentes industrielles. Nous participons à la campagne ImagEval (projet "Techno-Vision" du ministère délégué à la recherche mise en place par le CEA-LIST). Cette campagne était divisée en diverses tâches, dont une concernait l'extraction de texte. Cette dernière, comportait au départ du projet, une phase d'extraction et de reconnaissance de texte (en sus de la phase de localisation). Celle-ci ayant été ensuite abandonnée faute de participants. Les bases d'images fournies proviennent de détenteurs de grands fonds photographiques comme Hachette, le Musée NIEPCE ou encore la Réunion des Musées Nationaux.

C'est dans ce contexte restreint de localisation de texte dans des bases d'images hétérogènes que se situeront nos travaux.

1.3 Plan de ce manuscrit

Notre manuscrit comportera trois grandes parties.

1.3.1 Première Partie : Problématique Générale

Dans cette partie introductive, nous expliciterons et justifierons l'intérêt de l'extraction de texte pour la reconnaissance d'image par le contenu. Ceci nous entraînera naturellement à la description de la taxonomie de l'objet texte et à la revue des méthodes existantes pour chacune des sous-familles identifiées (majoritairement les approches *textures* et par *composantes connexes*). Cette première partie se terminera par l'analyse d'un algorithme pré-existant au sein de notre laboratoire (cf. Demarty [22]) développé à l'origine pour l'extraction de textes dans les vidéos. Et nous nous intéressons à son intégration possible dans notre contexte d'étude. Il repose sur une utilisation d'outils de morphologie mathématique à niveau de gris, notamment l'utilisation d'un outil résiduel numérique, le chapeau haut de forme. Il s'agit d'un outil puissant mais qui impose une connaissance a priori sur la taille des textes que l'on souhaite extraire.

1.3.2 Seconde Partie : Opérateurs résiduels numériques

Pour pallier le problème de paramétrage inhérent à l'algorithme présenté en fin de notre partie introductive, nous nous sommes intéressés à l'étude de nouveaux opérateurs résiduels introduits récemment dans la littérature. Plus particulièrement, cette seconde partie est dédiée à une analyse approfondie de l'opérateur d'ouverture ultime, étendue au cadre numérique par Beucher [8]. En effet il s'agit d'un opérateur non paramétrique permettant de mettre en lumière les structures dominantes en

terme de contraste de l'image : plus précisément il permet de connaître en chaque point de l'image la taille de la structure dominante (i.e. la taille d'ouverture pour laquelle elle disparaît) ainsi que son contraste. Grâce à cet outil, nous devrions pouvoir nous passer de tout a priori sur la taille du texte à extraire.

Dans cette partie, nous introduirons tout d'abord son formalisme, puis nous illustrerons son comportement sur des images réelles. Ceci nous permettra de mettre en avant ses forces et faiblesses. Nous mettrons ainsi en évidence certaines de ses myopies et proposerons des pistes pour y pallier. Nous montrerons également que l'utilisation d'autres familles d'ouvertures (que celles utilisées jusqu'à présent) semble plus indiquée pour notre problématique. Enfin, la volonté de passer d'un outil d'étude à une «brique algorithmique» nous conduira à un chapitre dédié à son implémentation.

Les connaissances acquises sur cet opérateur nous permettront de développer nos chaînes de localisation de textes. La description de celles-ci ainsi que nos résultats feront l'objet de notre troisième partie.

1.3.3 Troisième Partie : Chaînes de traitement et résultats

Dans la partie précédente, l'étude de l'opérateur nous a permis de montrer sa capacité à fournir des segmentations satisfaisantes de nos images. Nous nous placerons ainsi au sein de la famille des approches par *composantes connexes*. Partant d'une segmentation de référence, nous devons dissocier les composantes «lettres» des autres composantes puis les regrouper pour former une liste de boîtes englobantes, cette dernière sera la représentation utilisée pour l'évaluation de nos travaux. Nous exposerons dans cette partie l'ensemble des outils nécessaires à ce cheminement. Nombre d'entre eux seront paramétriques et nous veillerons à exposer leur forces et faiblesses.

Les chaînes décrites seront ensuite évaluées sur trois projets. Le premier correspond au coeur de notre problématique initiale, notre participation à la campagne $imagEval$. Les deux autres permettront de valider la capacité de généralisation de nos approches. Il s'agira du développement d'un démonstrateur pour EADS sur le classement automatique d'un fond photographique et du traitement de la base *internationale* d'ICDAR qui permettra de situer nos approches par rapport à l'état de l'art. Nous terminerons cette partie par une discussion critique de nos approches et une introduction sur l'étape manquante de nos systèmes : la reconnaissance effective du texte.

Problématique générale

Cette première partie introductive est composée de différents volets. Le premier concerne la motivation première de nos travaux : l'intérêt de l'extraction de texte dans les images pour la recherche d'image par le contenu. Il est présenté au sein du Chapitre 2. Le second, présenté dans le Chapitre 3 décrira la taxonomie des types de textes à extraire, et montrera que l'objet «texte» est un concept certes trivial pour un humain, mais difficile à modéliser en termes d'analyse d'image. Nous passerons ensuite à la revue des méthodes actuelles d'extraction de texte dans les images et les vidéos dans le Chapitre 4. Enfin, le dernier volet, présenté dans le Chapitre 5, constituera le point d'entrée de nos travaux. Il s'agit de l'étude d'un algorithme développé au sein de notre laboratoire. La mise en lumière de ses limitations et la volonté de le rendre *plus générique* guideront nos travaux dans les parties suivantes.

2 Problématique de l'indexation et apport de l'information textuelle :

Pour créer un marché il faut inventer un problème, puis trouver sa solution.

Extrait de "Le principe de Dilbert"
SCOTT ADAMS

Cette brève introduction de la problématique de l'indexation d'images s'inspire des travaux de compilation suivants [87, 77, 3]. Elle a pour but de montrer l'intérêt de cette démarche ainsi que les problèmes incontournables qu'elle soulève. L'apport du texte comme descripteur d'un système d'indexation sera abordé en fin de chapitre.

2.1 Définitions préliminaires

Pour éviter dès maintenant des incompréhensions dues à des abus de langage on rappellera ici quelques distinctions de vocabulaire relatif à l'indexation d'images.

On nommera :

Définition 1 ***Cataloguage :*** *l'identification d'une image (ou d'une autre donnée) généralement par un numéro n'ayant pas de rapport avec son contenu. Ceci implique que le seul moyen de remonter à l'objet est de connaître à l'avance son identifiant.*

Définition 2 ***Indexation :*** *C'est au sens littéral l'action d'indexer. Le terme d'indexation est la représentation d'un concept ou d'un sujet pour l'inclusion dans un index et pour la recherche d'information. Le terme d'indexation peut être un terme du langage naturel ou un symbole (un indice de classification, par exemple). A contrario du "Cataloguage", l'indexation d'un objet implique la possibilité de sa recherche autrement que par la simple connaissance de son identifiant. Par abus de langage l'indexation sera vue comme l'action automatique, semi-automatique, manuelle de mise en correspondance d'un objet avec un thésaurus donné (voir ci-dessous).*

Définition 3 ***Thésaurus***[a] *: Un thésaurus est un ensemble structuré de termes choisis pour leur capacité à faciliter la description d'un domaine et à harmoniser la communication et le traitement de l'information à son sujet. Chaque terme appelé descripteur est aussi peu ambigu que possible et est préféré à des termes voisins (quasi-synonymie) ou synonymes, les non-descripteurs, pour tous les échanges significatifs.*

En pratique, le thésaurus est un outil documentaire d'indexation. Guidé par un thésaurus pertinent, il est possible de représenter tout document par une sélection rigoureuse de mots précis, appelés mots-clés. Il sera ensuite aisé d'en assurer une forme quelconque de gestion documentaire.

En mode consultation et exploitation des données, le thésaurus devient un instrument de recherche : disposant des vocabulaires et règles de l'indexation, l'utilisateur peut optimiser ses requêtes.

On doit retenir une chose, ***il n'y a pas de thésaurus générique****, pour une même base d'images (ou de textes) des acteurs différents proposeront des thésaurus différents adaptés à leur attentes propres.*

[a] Source WikiPédia

Ces remarques de vocabulaire effectuées, on présentera la problématique de l'indexation d'images et les problèmes importants à soulever avant de se lancer dans la construction d'un système d'indexation. On insistera dans la deuxième partie de ce chapitre sur l'apport de l'information textuelle comme descripteur d'un système d'indexation.

2.2 Généralités sur l'indexation automatique d'images

2.2.1 Nécessité de l'indexation

Le nombre de documents multimédia est en constante augmentation. Cette croissance quasi-exponentielle a mis en évidence la nécessité de développer des outils de stockage et de recherche adaptés à la sauvegarde et la réutilisation de cette manne d'information.

Les techniques usuelles basées sur l'annotation manuelle des images se heurtent en pratique à deux difficultés majeures :

1. Le travail manuel nécessaire pour l'annotation des images ou des vidéos est trop important[1] .
2. L'annotation effectuée est fortement dépendante de la personne. (i.e. en dehors d'un modus opérandi strict chaque individu possède sa propre perception d'un contenu de scène.)

Ces constatations sont évidentes dans le cadre de l'indexation d'images et de vidéos où les définitions d'un *modus operandi* et d'un thésaurus *générique* sont informulables. C'est pourquoi le concept d'indexation automatique d'image a vu le jour, c'est à dire utiliser les méthodes de traitement et d'analyse d'images pour extraire des descripteurs qui permettront la mise en oeuvre d'algorithmes de recherche : par mots clés, par images requêtes ....[2]

De façon générale un système d'indexation se décompose selon Aigrain et al. [2] en trois étapes :

1. Extraction d'attributs significatifs sur une base d'apprentissage et mise en forme sous une représentation la plus compacte et efficace possible.

[1] Notons que des outils d'annotation manuelle présentés sous forme de jeux pourraient résoudre une partit de ce problème ; comme *Google Image Labeler* disponible à l'adresse suivante `http://images.google.com/imagelabeler/`

[2] Il est intéressant de noter le parallèle avec les méthodes d'indexation syntaxique.

2. Définition d'une ou plusieurs mesures de similarité permettant d'ordonner la population d'apprentissage selon ses caractéristiques dominantes.
3. Une interface utilisateur permettant de sélectionner des critères de similarité pour :
   - Retrouver une image de la base à partir de ces caractéristiques.
   - Ordonner les images les plus similaires à une image requête.
   - Permettre l'interaction de l'utilisateur sur la pertinence des résultats.

On trouvera sur la Figure 2.1 l'architecture générale d'un système d'indexation (Rui et al. [77]), résumant les remarques précédentes. On retrouve deux phases principales : l'une *hors-ligne* pour le pré-traitement de la base de données et l'autre *en-ligne* pour une recherche plus ou moins interactive.

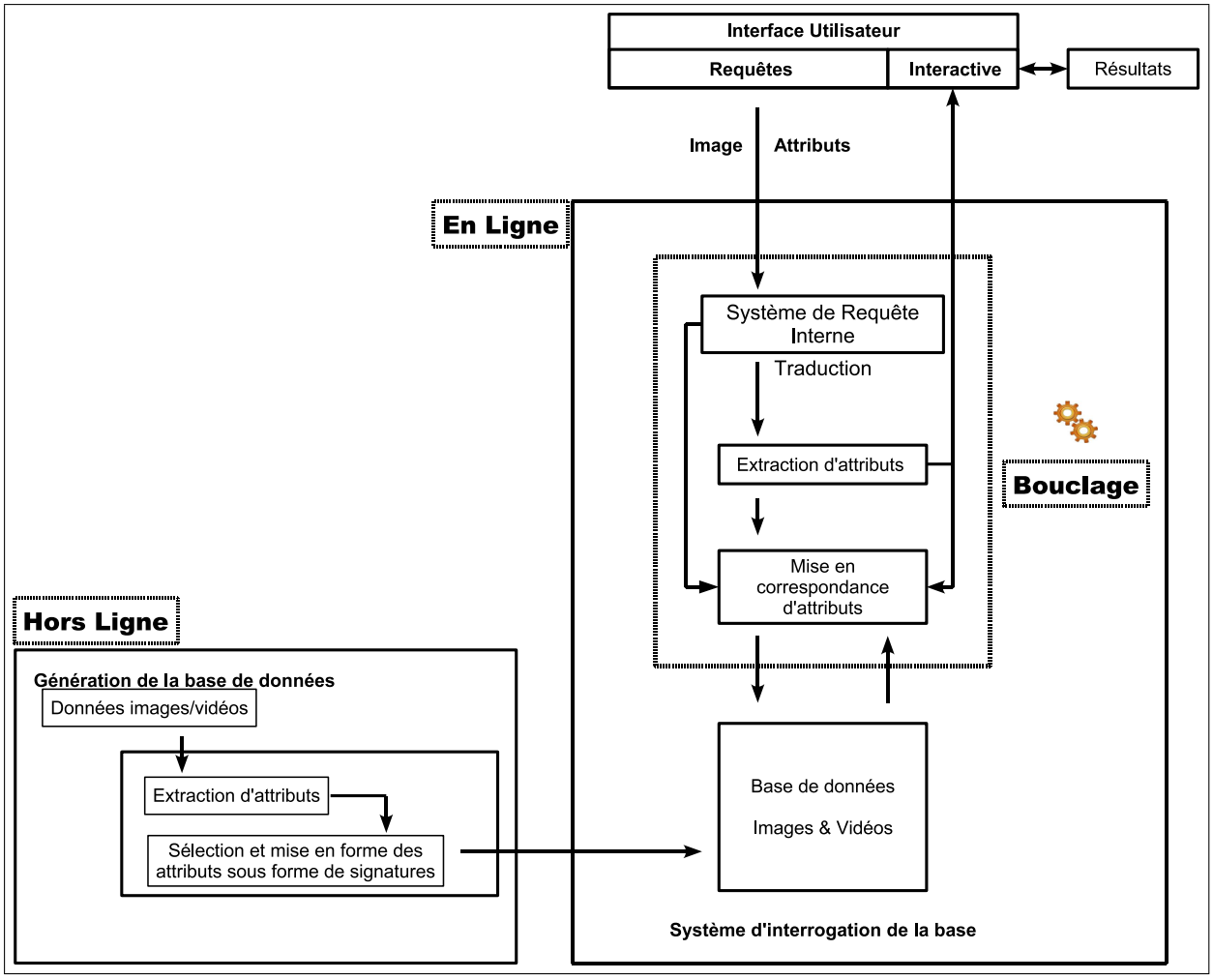


FIG. 2.1: Représentation schématique d'un système d'indexation.

2.2.2 Les problèmes latents

Du fait des variations d'illumination, des ombres, des occlusions, du facteur d'échelle, des déformations géométriques, de la méconnaissance de la prise de vue ..., on peut énoncer la remarque suivante :

Remarque 1 *Problème de la définition d'un objet :*
il existe une "infinité" d'images différentes décrivant/comprenant un même objet.

Une autre remarque suit aussitôt la précédente :

Remarque 2 *Interprétation d'une image :*
l'interprétation d'une image est fortement dépendante du contexte et du domaine dans lesquels elle se situe ainsi que de la personne qui l'interprète. Il n'y a pas d'unicité de sens.

Ces deux remarques permettent d'introduire le problème majeur posé par l'indexation d'image, *le fossé sémantique*.

2.2.3 Le fossé sémantique :

C'est l'un des problèmes majeurs en recherche d'image par le contenu (i.e. C.B.I.R *C*ontent *B*ased *I*mage *R*etrieval). Il est relié aux remarques 1 et 2.

2.2.3.1 Définition :

Une définition a été proposée par Smeulders et al. [87] :

Définition 4 ***The semantic gap :*** *"The semantic gap is the lack of coincidence between the information that one can extract from the visual data and the interpretation that the same data have for a user in a given situation."*

Cette formule exprime la difficulté de relier des caractéristiques primaires de l'image (texture, couleur, ...) avec la notion *sémantique* véhiculée par l'image pour n'importe quel utilisateur humain. On peut définir différents niveaux sémantiques de l'image comme ceux proposés par Wang et al. [107].

On pourra les ordonner de la manière suivante :

1. La catégorisation de scènes (nommée également reconnaissance d'attributs) : photo ou dessin, scène d'intérieur ou d'extérieur,...
2. La composition d'objets : un vélo et une voiture garée à côté d'une plage.
3. Les types sémantiques abstraits : personne heureuse, une bataille
4. La description sémantique *idéale* de la scène (Idéal qui dépendra néanmoins d'une application donnée).

On trouve sur la Figure 2.2 un test de type CAPTCHA [1] (Completely Automated Public Turing Test To Tell Computers And Humans Apart[3]) utilisant l'information sémantique contenue dans une série d'images pour différencier un utilisateur humain d'un automate.

Cet exemple nous permet une nouvelle fois de souligner le fait que les caractéristiques primaires ne peuvent traduire la notion sémantique véhiculée par une image dans un cadre purement généraliste.

[3] `www.captcha.net`

(a) Images d'éléphant (b) Images d'éléphant avec des distorsions

FIG. 2.2: Exemple de CAPTCHA utilisant des images : «Trouver l'élément commun de ces images». Les caractéristiques primaires extraites des images ne peuvent seules coder le contenu sémantique des images. (Credit : `http://www.captcha.net/captchas/pix/`)

2.2.3.2 Connaissance du domaine et fossé sémantique :

Il est bien évident que la connaissance a priori de tout ou partie du domaine de recherche aide l'indexation. Nous pouvons effectuer des distinctions graduelles entre un domaine d'application restreint et très large comme l'ont proposé Smeulders et al. [86] :

Définition 5 *Domaine restreint : un domaine restreint possède une variabilité limitée et prévisible dans les attributs qui décrivent son aspect.*

Définition 6 *Domaine étendu : un domaine étendu possède une variabilité illimitée et non prévisible dans les attributs qui décrivent son aspect, même si sa signification sémantique est constante.*

Ces définitions bien qu'artificielles, permettent de proposer une comparaison entre ces deux domaines résumée dans le Tableau 2.1.

	Restreint	**Étendu**
Variation de contenu	Faible	Important
Connaissance a priori	Spécifique	Générique
Sémantique	Homogène	Hétérogène
Connaissance de la prise de vue	Parfois disponible	Inconnue
Description du contenu	Objectif	Subjectif
Application recherchée	Spécifique	Générique
Outils	Modélisation, reconnaissance d'objets, ...	...
Invariants	Spécifiques	Génériques
Évaluation des outils	Quantitatif	Qualitatif

TAB. 2.1: Comparaison des domaines *étendu* et *restreint* dans le cadre de la recherche d'images

En conclusion préliminaire, on peut stipuler d'une part que l'infinie variabilité des scènes rendra périlleuse l'utilisation de modèles dans le cas du *Domaine étendu* et que d'autre part une grande attention devra être portée aux qualités d'invariance des descripteurs et méthodes utilisés.

2.2.4 Comment s'affranchir du fossé sémantique ?

Procédé par *Relevance FeedBack* (ou bouclage/contrôle de pertinence) : Pour réduire le fossé sémantique, une démarche classique consiste à interroger l'utilisateur sur la pertinence des résultats proposés par le système. Le processus d'indexation sera affiné progressivement en injectant l'information fournie par l'utilisateur. Cette technique avait été précédemment formulée pour l'indexation textuelle. On pourra se référer à l'ouvrage de référence de Salton [79] pour plus de détails.

Définition 7 *Le bouclage/contrôle de pertinence : ce processus peut être vu comme une déformation progressive du voisinage de recherche autour de la requête de l'utilisateur en fonction de ses interactions avec le système.*

En exploitant l'interaction homme-machine, on tente de répondre aux requêtes de haut niveau des utilisateurs alors que l'information est représentée par des attributs de bas niveau dans la base d'indexation.

Utiliser des descripteurs de *haut niveau* : Mélanger des modèles simples pour en faire émerger des concepts plus complexes, comme par exemple extraire les régions d'une image et procéder à la classification de chacune d'elles ce qui permettra de remonter au sens de la scène. On pourra ranger dans ce type d'étude les approches dites de *Catégorisation de scènes*. Ces approches consistent à résoudre des problèmes de *faible niveau sémantique* mais dont la composition permettra de remonter au sens de la scène. On pourra citer différents travaux ayant pour but : la catégorisation de scènes d'intérieur ou d'extérieur [74, 92], de déterminer si l'on a affaire à un dessin ou une photographie [46], de différencier les paysages urbains des paysages naturels [99], de détecter la présence de texture dans une scène [37], enfin de détecter des présences humaines avec des applications de surveillance du contenu pornographique des images et des vidéos [25, 106] et à la détection de visages [9].

2.3 Apport de l'information textuelle :

Le texte présent dans une image ou une vidéo est typiquement un descripteur de haut niveau qui véhicule la plupart du temps des informations concises et pertinentes sur le contenu d'une scène. C'est évident pour le texte surimposé (titre d'un reportage, date, nom des personnes présentes) mais le texte enfoui peut également apporter des informations intéressantes (noms des rues, titres d'articles de journaux,...).

L'extraction de mots clés est immédiate comme le présente la Figure 2.3 et un module d'extraction automatique de texte pourra venir compléter un système d'indexation plus complet couplant descripteurs de haut et bas niveaux dans le cadre de :

- L'annotation automatique de journaux télévisés
- L'édition et le catalogage de vidéos
- L'aide à l'indexation de fonds digitalisés
- ...

Typiquement, pour rendre conforme l'annotation d'un fond digitalisé à la norme MPEG7, on aimerait obtenir le type de descripteur présenté sur la Figure 2.4.

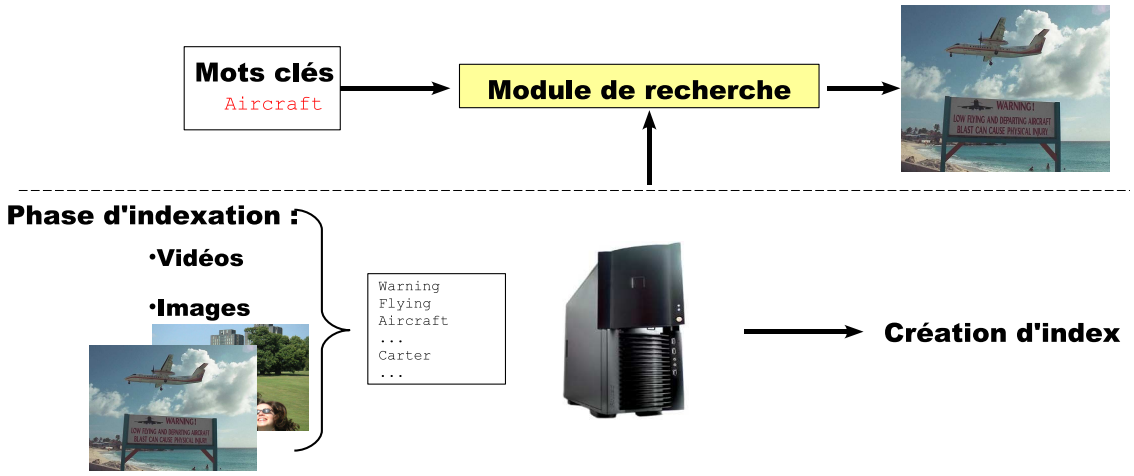


FIG. 2.3: Indexation basée sur des mots clés extraits de légendes ou de texte enfoui.

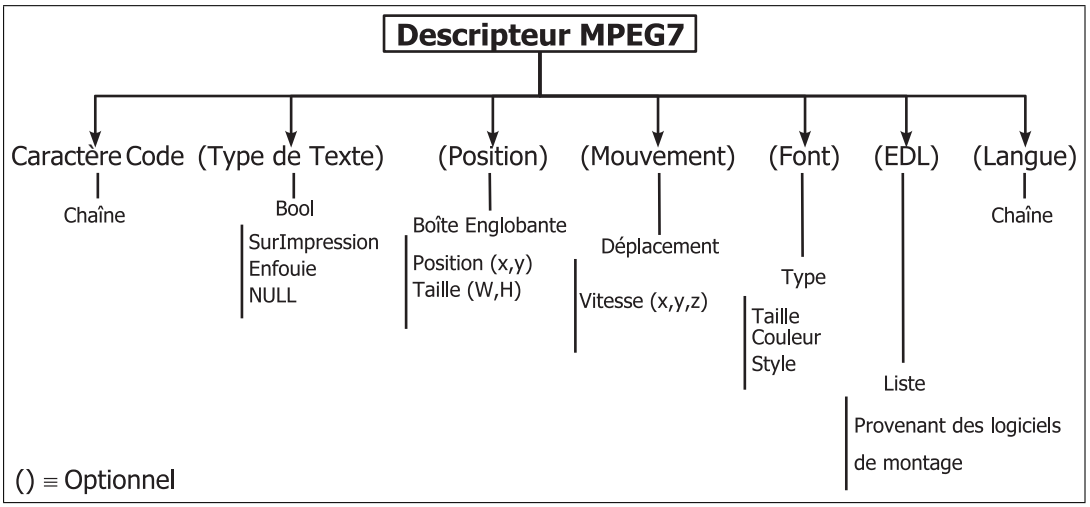


FIG. 2.4: Descripteur associé à un texte en vue de son indexation [23].

2.4 Qu'attend-on d'un système d'extraction de texte (*SET*)

Un tel système reçoit soit une image, soit un groupe d'images, celles-ci peuvent être en niveaux de gris, couleur, compressées ou non et les zones de texte à l'intérieur d'une séquence peuvent être statiques ou posséder un mouvement propre.

Un *SET* est généralement décomposé en cinq étapes [4] (cf Figure 2.5) :

DÉTECTION DE TEXTE : Détermine la *présence potentielle* de texte dans une image. Elle ne s'applique d'habitude qu'à des données vidéos. En effet, il serait trop coûteux de lancer pour l'ensemble des trames d'une vidéo, un système de localisation *précis*. Ces systèmes ne sont généralement pas mis en place pour des bases de données d'images.

[4] Ces étapes seront reprises en détail dans la Section 4 p.25

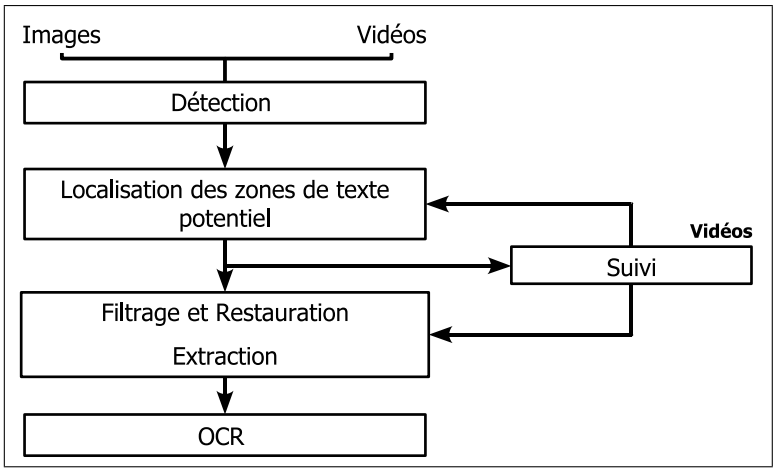


FIG. 2.5: Architecture d'un système d'extraction de texte.

LOCALISATION DES ZONES DE TEXTE : Localisation des zones de texte dans l'image et création de boîtes englobantes.

SUIVI : Des caractères identiques apparaissant dans plusieurs images consécutives à l'intérieur d'une vidéo, on utilise le suivi des zones de texte pour diminuer conjointement le temps de localisation dans les séquences et le nombre de fausses détections. Cette étape peut également être utilisée dans le module suivant.

RESTAURATION ET EXTRACTION : Étape où l'on sépare le texte du fond. Si le texte est de faible résolution et/ou s'il présente des déformations (échelle, perspective,...) une phase de restauration est entreprise : augmentation du contraste des lettres vis à vis du fond, correction des distorsions liées à la perspective, approche type "super-résolution" pour les séquences ... Puis on passe à l'extraction proprement dite le plus souvent réalisée par seuillage.

RECONNAISSANCE OPTIQUE DES CARACTÈRES (OPTICAL CHARACTER RECOGNITION) : Cette étape est généralement sous-traitée à un système annexe auquel on soumet les zones de texte restaurées et binarisées.

Dans la section suivante nous introduirons de manière plus formelle les différents types de textes que l'on rencontre dans les images et les vidéos. Une discussion sera entamée sur leurs propriétés et par extension sur les hypothèses les plus usitées.

3 Comment appréhender le texte dans les images et les vidéos

Ne me dites pas que ce problème est difficile. S'il n'était pas difficile, ce ne serait pas un problème.

FERDINAND FOCH

Dans ce chapitre nous introduirons les différents types de textes rencontrés dans les images et les vidéos.

3.1 Définitions

Selon Lienhart [48], on peut classer les différents types de textes apparaissant dans les images ou vidéos comme suit :

Définition 8 ***Texte en sur-impression (Overlay Text, voir Figure 3.2) :*** *Texte rajouté a posteriori en surimpression sur une scène. On pourra citer :*
- *les inscriptions/bandeaux des journaux télévisés*
- *les génériques de films*
- *le logo des chaînes et des sponsors*
- ...

Les textes publicitaires télévisuels sont aussi souvent classés dans cette catégorie.

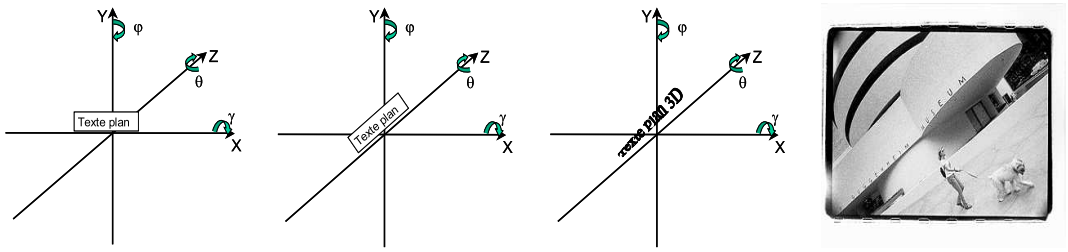
Définition 9 ***Texte Enfoui (Scene Text, voir Figure 3.3) :*** *Texte enregistré comme faisant partie intégrante d'une scène. On pourra citer :*
- *le nom des rues, des boutiques*
- *les inscriptions sur les murs ou les vêtements des passants*
- ...

On notera également que les documents couleur complexes (couvertures de livres, pochettes de CD) se retrouvent le plus souvent classés dans cette catégorie.

Une autre classification, proposée par Meyers et al. [61] s'intéresse au nombre de degrés de liberté d'une zone de texte plongée dans un environnement tri-dimensionnel :

Définition 10 ***Texte plongé dans une scène 3D :***
Les familles se regrouperont selon le nombre de degrés de liberté :

1. *Texte horizontal ajouté en surimpression parallèlement au plan de la prise de vue (θ =0,ϕ =0,γ =0) (cf Figure 3.1(a))*
2. *Texte ajouté en surimpression parallèlement au plan de la prise de vue (θ =Libre,ϕ =0,γ =0) (cf Figure 3.1(b))*
3. *Texte ajouté en surimpression sans contrainte de placement mais sur une surface plane (θ =Libre,ϕ =Libre,γ =Libre) (cf Figure 3.1(c))*
4. *Texte libre , sur surface libre également (θ =Libre,ϕ =Libre,γ =Libre) (cf Figure 3.1(d))*



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

FIG. 3.1: Définition d'un texte plongé dans un environnement tri-dimensionnel

3.2 Propriétés du texte dans les images et les vidéos

La difficulté des problèmes de reconnaissance de texte dans les images dépendra fortement du type **Texte Enfoui**, **Texte en sur-impression** (cf Définition 9 et 8) que l'on veut extraire mais également du vecteur de diffusion : vidéo compressée ou non , documents couleur ou noir blanc, type d'alphabet ...

On se référera aux exemples présents sur les Figures 3.2 et 3.3 pour appréhender la grande variabilité des textes que l'on peut rencontrer.

Il est important de noter dès maintenant le fossé entre les documents dits multimédia (textes issus de photographies, de vidéos, de couvertures de journaux complexes,...) et les documents usuels (livres, articles scannés). Nous reprendrons les remarques exposées par Wolf [110] et nous les compléterons au besoin :

Résolution : Les systèmes d'O.C.R se disent aujourd'hui multi-fonte et capables de taux de reconnaissance allant de 95 à 100% dans leur utilisation courante, c'est à dire la reconnaissance de **caractères typographiés noirs sur fond blanc** issus de documents scannés. Leur performance chute abruptement en présence de bruit ou pour des documents de résolution inférieure à 200 ou 300 dpi (on consultera pour exemple les résultats de la compétition Rice et al. [73]). Ces résolutions correspondent à une hauteur moyenne de plus de 40 pixels par lettre. On notera également que des tests réguliers sont réalisés sur la performance des O.C.R et ceux-ci obtiennent de manière constante des résultats inférieurs à un panel d'enfants du primaire [62].

FIG. 3.2: Texte en sur-impression : Images issues des bases de tests de Demarty [22] et Wolf [110]. On notera la faible définition, la variabilité des tailles des fontes, . . .

FIG. 3.3: Texte Enfoui : Images issues de la compétition de localisation de texte ICDAR [53]. On notera la présence de fontes complexes, de problèmes d'illumination, de perspective, . . .

Les résolutions actuelles des vidéos varient elles entre 160*100 pixels pour les vidéos en "streaming" sur internet et 720*480 pour des vidéos encodées en MPEG2. Ainsi la hauteur caractéristique d'une lettre de texte en surimpression au format CIF (384*288 pixels) avoisine 10 pixels.

De plus à l'inverse des documents scannés, le texte présent dans les documents multimédia peut avoir des tailles de fonte variées allant de 8 à 80 pixels de haut pour une image au format "cif", voir Figure 3.4.

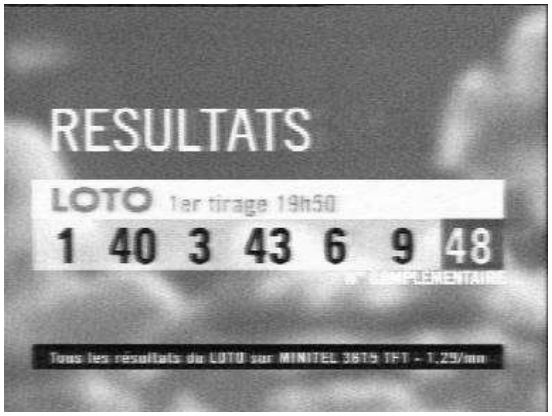


FIG. 3.4: Variabilité de la hauteur des fontes dans une image publicitaire (Base [110])

Ces problèmes de résolution indiquent qu'un système d'extraction de texte devra obligatoirement disposer d'une approche multi-résolution pour la détection/localisation des zones de texte et qu'une fois localisées celles-ci devront subir une augmentation artificielle de leur résolution (avec minimisation des distorsions, ...) avant d'être soumises à un O.C.R commercial.

Anti-aliasing et Compression : Pour des raisons de coût de stockage et de sauvegarde de bande passante, la plupart des documents multimédia sont souvent sous-échantillonnés et compressés. L'étape de sous-échantillonnage faisant apparaître des problèmes d'aliasing (marches d'escalier), on applique préalablement un filtre passe bas. Si cette étape d'anti-aliasing rend le document perceptuellement plus agréable, il va fortement dégrader les informations sur les contours des objets. A titre d'exemple on regardera la Figure 3.5, qui montre à quel point une information parfaitement lisible peut posséder une définition pauvre.

Fond : La plupart des documents scannés sont composés de deux couleurs uniformes représentant le fond et la forme, ce qui n'est absolument pas le cas des zones de texte présentes dans les images et les vidéos qui peuvent être inscrites sur un fond arbitrairement complexe. Ce fond empêche dès lors toute approche simple de binarisation par un seuillage global de la zone. On ne peut pas par exemple fixer un seuil optimal pour la zone de texte présente sur la Figure 3.6 :

Rehaussement artificiel du contraste : Du fait des problèmes de fond complexe préalablement décrits, les "designers" ajoutent souvent des ombres portées pour renforcer artificiellement le contraste des lettres. Les lettres sont psycho-visuellement plus lisibles mais la tâche de binarisation s'en trouve d'autant compliquée (Voir l'exemple Figure 3.6).

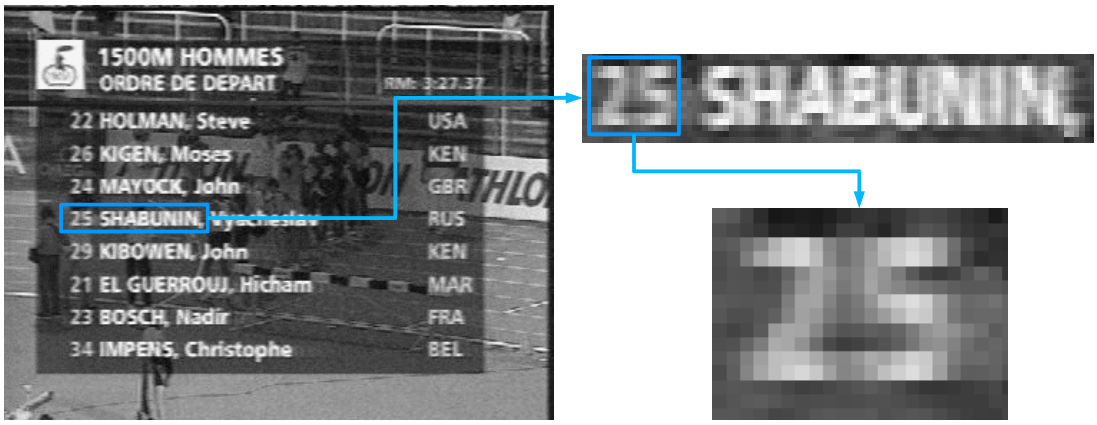


FIG. 3.5: Effet de l'anti-aliasing : le texte est perceptuellement agréable bien que fortement dégradé.(Base [110])

FIG. 3.6: La présence de fond complexe rend caduques les approches simples de binarisation. (Base [110])

Artwork, déformation, illumination : Il suffit de regarder les images présentes sur la Figure 3.3 pour prendre conscience de l'importante variété des textes plongés dans les scènes.

Dépendance de la langue : La plupart des travaux d'extraction de texte s'appuie sur la connaissance du type d'alphabet recherché ; aussi les travaux avec des écritures "Latines", "Arabes", ou "Idéophonographiques"(Chinoise, Coréenne, Japonaise) font le plus souvent l'objet d'études distinctes. On retrouvera sur la Figure 3.7 quelques exemples d'alphabets[1].

FIG. 3.7: Différentes écritures : Latine, Arabe et idéophonographique.

[1]On pourra ainsi regarder les travaux de Spitz [91] qui proposent des descripteurs permettant de différencier les écritures Latines des écritures idéophonographiques.

3.2.1 Quelles sont les hypothèses les plus souvent retenues :

Nous avons dans la section précédente énuméré un certain nombre de problèmes liés à la recherche de texte dans les images et les vidéos.

Malgré toutes les objections formulées précédemment, certaines hypothèses de travail sont souvent admises [48, 36].

CONTRASTE :La plupart des textes en sur-impression ou enfouis ont été définis pour être facilement lisibles pour un observateur humain. On pourrait donc espérer un fort contraste du texte vis à vis du fond. Ceci ne peut être garanti, pour le texte dans les vidéos que l'on retrouve souvent sur des fonds complexes et pour le texte dans les scènes comprenant de fortes variations d'illumination.

CONTRAINTES GÉOMÉTRIQUES :

- Taille : Bien que la taille des caractères puisse être extrêmement variable, certaines hypothèses peuvent être formulées en fonction de l'application.
- Alignement : Les caractères en surimpression apparaissent en grappe, typiquement on considère qu'il faut un minimum de trois caractères pour définir une zone potentielle de texte. Ils sont le plus souvent alignés horizontalement.

  Ceci ne s'applique pas au texte enfoui : il peut être aligné dans n'importe quelle direction ou pire encore comporter des effets typographiques de type "WordArt". Des déformations liées à la perspective peuvent également apparaître pour des textes non plans (cf.Figure 3.1).
- Distance inter-caractère : On considérera généralement que la distance inter-caractère pour un même mot ou une ligne de texte est proche d'une constante. Attention cette mesure correspond à la distance entre les boîtes englobantes et non à la distance "inter-trait". Cette dernière est pseudo-régulière : il suffit de comparer les chaînes **"CLJ"** et **"MWM"** pour le constater. Cette pseudo-régularité pourra encore se dégrader lors de l'utilisation de différents styles, fontes ou pour des textes non plans.

COULEUR : Les caractères tendent à avoir une certaine unicité de teinte, aussi beaucoup de recherches sont axées sur la caractérisation de lettres monochromes et contrastées. Cependant dans le cas de la vidéo, si la résolution est faible les effets de l'anti-aliasing rendent caduque cette hypothèse et en ce qui concerne les documents couleurs complexes ou le texte enfoui, les lignes de texte peuvent être polychromes, posséder des dégradés et pire encore des couleurs différentes (ou une texture propre) pour un même caractère.

MOUVEMENT : Dans une vidéo, les caractères sont le plus souvent présents sur plusieurs images consécutives de manière statique ou en possédant un mouvement propre. Cette propriété est utile pour le suivi d'une zone de texte le long d'une séquence et surtout pour les étapes de validation des zones de texte détectées et le rehaussement des caractères avant extraction. Le texte en surimpression a généralement un déplacement uniforme (le plus souvent borné aux déplacements horizontaux et verticaux). Le texte enfoui lui possède des mouvements imprévisibles dûs au déplacement des objets et du système d'acquisition.

COMPRESSION : Les images et les vidéos sont enregistrées sous forme compressée. On pourra tirer parti de cette mise en forme des données pour accélérer certaines étapes de détection et/ou de localisation.

3.3 Conclusion

Nous avons présenté un grand nombre des difficultés à résoudre pour la réalisation d'un système d'extraction de texte. Le point clé à retenir est que les choix algorithmiques lors de la construction d'un S.E.T sont étroitement liés au média de diffusion et l'application recherchée. Dans le chapitre suivant nous présenterons l'état des travaux réalisés pour chaque étape d'un S.E.T. Notons qu'il s'agira d'un découpage un peu artificiel tant les modules sont interdépendants pour une application donnée.

4 Extraction de l'information textuelle : Revue des méthodes

On résout les problèmes qu'on se pose et non les problèmes qui se posent.

HENRI POINCARÉ

Dans ce chapitre nous passerons en revue les méthodes et techniques nécessaires à un système d'extraction de texte complet. S'agissant d'un domaine en pleine maturation les documents traitant de tous les aspects d'un tel système sont rares. Toutefois quelques documents récents pourront venir compléter la lecture de ce chapitre. Le lecteur pourra consulter les travaux de Chen and Luettin [12] et Lienhart [48] pour la tâche spécifique de l'extraction de texte dans les vidéos et ceux de Jung et al. [36] et Liang et al. [47] pour un panorama applicatif plus général. Enfin notons qu'il serait discutable d'effectuer des comparaisons sur les performances algorithmiques des diverses approches, du fait de la méconnaissance partielle ou totale des jeux de données utilisées. Aussi les résultats *proposés par la suite, le seront à titre indicatif.*

4.1 Détection des zones potentielles de texte

4.1.1 Objectifs :

Lors de l'analyse d'un média on se doit logiquement de ne pas avoir de connaissance a priori. En effet la variabilité des sources que l'on doit pouvoir traiter est importante (eg : images à niveaux de gris ou en couleur provenant : de couvertures de magazines, photos, vidéos). Le rôle de l'étape de *détection* est justement de déterminer la présence potentielle de texte au sein d'un média sans pour autant le localiser.

Remarque 3 *Cependant beaucoup d'auteurs partent du principe que les images qu'ils ont à traiter contiennent du texte et donc cette phase de détection est souvent absente. Si cette hypothèse est valide pour des applications particulières comme l'analyse de documents elle l'est beaucoup moins en ce qui concerne l'indexation d'images en général et l'extraction de zones textuelles dans les vidéos.*

4.1.2 Principes et méthodes existantes

Les principales méthodes existantes ont été proposées pour la détection de texte en surimpression dans les vidéos. Elles reposent toutes sur le même concept : l'apparition et la disparition de zones de texte correspondent à un *changement de scène* particulier.

Kim [38] et Smith and Kanade [88] proposent d'utiliser des algorithmes existants de détection de changement de scène. Cependant on devra être attentif au seuil de détection utilisé pour ne pas être dépendant des changements de scène classique (eg : fondu, coupure). On pourrait, par exemple, réaliser cette détection au sein d'une vidéo préalablement décomposée grossièrement. Pour rendre plus robuste cette approche, Tang et al. [93] suggèrent d'*apprendre* à reconnaître spécifiquement ces transitions.

Une autre approche consiste à travailler directement sur les propriétés des algorithmes de compression et de transmission des vidéos. Aussi Crandall et al. [21] proposent un algorithme de détection tirant parti de la compression MPEG. L'hypothèse est que le nombre de blocs I (Intracoded) augmente lors de l'apparition de zones de texte dans les images B (Bidirectionnelles) et P (Prédites). Comme la méthode précédente elle est sensible aux changements abrupts de scènes et aux mouvements rapides à l'intérieur de la vidéo.

Toujours grâce aux propriétés des algorithmes de compression, on peut réaliser des algorithmes de localisation très rapides mais peu précis des zones potentielles de texte comme le proposent Zhong et al. [123]. Ceci permet de réduire le coût d'une localisation fine pour chaque image, une description de ces travaux est réalisée en Section 4.2.6 (p. 33).

4.2 Localisation des zones de texte

Il est difficile de donner une taxonomie précise des méthodes de localisation, aussi nous avons pris le parti de les classer en fonction de la caractéristique prépondérante des algorithmes utilisés.

Remarque 4 *La plupart des algorithmes ne traitant que des instances de textes horizontaux, on considérera pour la suite (si aucune mention contraire n'est présente) que cette hypothèse est vérifiée.*

4.2.1 Les approches par segmentation et regroupement spatial

Ce sont les premières méthodes qui ont été proposées pour la localisation de texte. Elles découlent en droite ligne des méthodes d'analyse de documents. Elles utilisent l'information colorimétrique ou de niveaux de gris des zones de texte et leur contraste présumé avec le fond. Elles travaillent "du bas vers le haut" (i.e. bottom-up) : identification de petites structures et regroupement de celles-ci pour former des régions. Chaque région est ensuite encapsulée par une boîte englobante, identifiant la zone de texte.

On peut les schématiser en quatre grandes étapes :

1. Quantification/Réduction des espaces couleur en un certain nombre de classes prédominantes.
2. Génération des composantes : par seuillage ou décomposition de l'espace couleur en plans chromatiques.
3. Filtrage et élimination (le plus souvent de manière heuristique) des composantes non texte.
4. Groupement des composantes pour reformer les caractères parfois morcelés puis les chaînes de caractères proprement dites.

Elles peuvent sembler intéressantes du fait de leur simplicité apparente, mais les étapes $1)$ et $2)$ peuvent scinder les caractères en de multiples fragments dans le cas de caractères polychromes ou de données bruitées et basse résolution. De plus les étapes $3)$ et $4)$ sont souvent critiques, dépendantes de nombreuses heuristiques très contraintes par l'application.

Ohya et al. [65] proposent une méthode basée sur un seuillage adaptatif de type variance inter-classe. Chaque composante connexe obtenue est encapsulée dans une boîte englobante et un test de contraste est réalisé entre le niveau de gris moyen de la composante et celui du fond délimité par la boîte. Si ce contraste est trop faible, la composante est éliminée. Les lettres pouvant être morcelées par l'étape de seuillage, on regroupe les composantes sur des critères de proximité et d'homogénéité des niveaux de gris. Les composantes restantes sont considérées comme des lettres candidates et sont comparées à un dictionnaire de référence défini par les auteurs. Cet algorithme est appliqué sur une centaine d'images de type *scene text* (eg : Panneau de signalisation, plaque minéralogique, enseigne de magasin) comportant des variations d'illumination, de taille et de type de caractères.

Zhong et al. [122] proposent un algorithme basé sur l'hypothèse d'unicité colorimétrique des lettres. L'histogramme RGB est d'abord quantifié en un nombre restreint de classes puis les pics dominants de celui-ci sont extraits récursivement. A chaque pic correspondra une image binaire qui sera labellisée en composantes connexes. Toutes les composantes extraites sont ensuite classées en texte/non texte par une batterie d'heuristiques (eg : aire minimale/maximale, élongation, alignement) Les quelques images de test sont composées de couvertures de livres et de pochettes de CD comportant différents types d'écritures, toutes couvrant une partie relativement importante de l'image. C'est un algorithme rapide mais trop spécialisé pour un type d'application.

Shim et al. [84] utilisent la propriété d'homogénéité en luminance des lettres et le contraste fort de celles-ci avec le fond. Ils labellisent l'image en λ zones plates ($\lambda = 10$) et calculent pour chaque composante connexe un grand nombre de descripteurs (la boîte englobante, l'aire, la moyenne, la variance, le barycentre, le nombre de trous, ...). Toutes les régions ayant une aire/hauteur/largeur supérieures à un certain seuil sont éliminées. L'image est ensuite labellisée à l'aide d'un critère purement connectif pour fusionner les caractères morcelés ou fusionnés avec le fond. Les descripteurs sont recalculés sur ces nouvelles composantes. Ensuite un processus de rehaussement de frontière itératif est appliqué entre les composantes connexes et le fond délimité par les boîtes englobantes. Les descripteurs sont réactualisés et les régions ne respectant par certains prédicats géométriques et de contraste sont éliminées. Les régions restantes sont ensuite regroupées sur des critères d'alignement spatial pour former les zones de texte candidates. Comme l'indiquent les auteurs, cette méthode doit être réadaptée en fonction de l'application : en effet, d'une part les prédicats géométriques utilisés sont très contraints par le format d'image CIF et aucune approche multi-échelle n'est proposée, et d'autre part les critères de contraste ne permettent pas la gestion de fond complexe. Cet algorithme a été réutilisé par Dimitrova et al. [23].

Lienhart [49], Lienhart and Effelsberg [50] considèrent les zones de texte comme des composantes connexes homogènes en taille et en couleur. Un algorithme de type *Division-Fusion* est appliqué pour obtenir une première segmentation de l'image. La partie division est réalisée sur des critères colorimétriques et la partie fusion tend à la fois à uniformiser les régions fusionnées et à corriger les erreurs de division dûes au bruit. Les régions résultantes sont ensuite triées par un ensemble d'heuristiques (hauteur, largeur, ratio et compacité). Une dilatation connecte les composantes adjacentes en grappes, et on associe à chaque grappe une boîte englobante. Pour chaque boîte, on étudie les fluctuations du gradient le long de son axe principal. Si celles-ci ne respectent pas certaines règles de périodicité et d'espacement, la boîte est invalidée. Enfin un algorithme de "Block Matching" (voir 4.2.7 p.34) au niveau **caractère** teste la consistance de la localisation sur plusieurs trames. Cet algorithme obtient

d'excellents résultats sur des génériques de films ou de séries possédant des fontes larges et espacées mais ses performances chutent pour des images de publicité.

Zhou *et al.* [125, 124] proposent un algorithme de clustering hiérarchique combinant distance couleur RGB et proximité spatiale des couleurs dans l'espace image. Comme l'indiquent les auteurs l'ajout de la pondération spatiale augmente grandement la qualité de l'algorithme et il donne des résultats relativement intéressants sur une base d'images provenant de bannières internet plus ou moins complexes.

Jain and Yu [33] reprennent le schéma proposé par Zhong et al. [122] et proposent la mise en place de stratégies séparées pour la prise en compte des caractères connectés et déconnectés. Cet algorithme est testé sur un jeu d'images variées comportant des images binaires, des bannières de site internet, des images couleurs et des vidéos, il ne traite que les cas des textes horizontaux et bien évidemment ses performances chutent considérablement (comme pour Zhong et al. [122]) si l'histogramme couleur est clairsemé.

Suivant le même principe de *clustering* couleur, suivi d'une analyse en composantes connexes permettant le regroupement de régions "similaires", Hase et al. [28] proposent une extraction des chaînes de caractères potentiels basée sur un algorithme de relaxation stochastique. Cet algorithme est testé sur des couvertures de journaux relativement complexes comportant des chaînes courbées et obtient des résultats de localisation relativement satisfaisants. Cependant la prise en compte de cette complexité ce traduit par des non détections et de nombreux faux positifs. Des améliorations à ce type d'algorithme ont été proposées par Wang and Kangas [108], d'une part l'utilisation d'un préfiltrage par diffusion anisotrope pour rendre plus robuste l'étape de clustering et d'autre part la mise en place d'un processus d'agrégation adapté pour les écritures "Latines" et "Idéophonographiques".

Enfin toujours sur la même approche, le système "Ashida" classé premier lors de la compétition ICDAR 2003[53] propose de : quantifier l'espace Luv et de procéder à la sélection des plans chromatiques à l'aide d'un algorithme du type "fuzzy-cmeans". Pour chaque plan chromatique les composantes connexes sont extraites et groupées sur des critères d'alignement (les composantes isolées sont supprimées). Les composantes restantes sont ensuite validées à l'aide de descripteurs géométriques (eg : variance des longueurs de plages dans différentes directions, nombre de composantes entrantes et sortantes, nombre de trous, ratio hauteur/largeur) soumis à une Machine à Vecteurs de Support. Ce système a été classé premier (sur 4, avec une performance globale [1] de 0.5 cf.[53]) au sein de la compétition.

4.2.2 Analyse ligne à ligne

Ces méthodes traitent séparément chaque ligne de l'image pour les classifier en texte/non texte potentiel. Les lignes sont ensuite regroupées sur des heuristiques pour former les zones de texte candidates.

Kim [38] utilise les mêmes processus de quantification et sélection des couleurs dominantes de l'image que Zhong et al. [122]. Pour chaque image binaire (associée à un pic de l'histogramme) il supprime les composantes connexes touchant le bord de l'image et plus longues horizontalement qu'un certain seuil. Puis sur l'hypothèse d'horizontalité des zones de texte, il calcule pour chaque ligne des paramètres statistiques (distribution des segments en nombre, en taille, nombre minimal de segments,...) permettant de la valider ou non comme ligne candidate. Les lignes retenues et adjacentes verticalement sont regroupées, les blocs ainsi formés sont soumis à un critère de hauteur puis

[1] i.e. moyenne harmonique entre précision et rappel

étudiés séparément. Pour chaque bloc un processus de découpage itératif par projection horizontale et verticale est réalisé, les sous-blocs formés sont validés au fur et à mesure sur différents critères (eg : longueur, hauteur, aire, caractéristiques de la signature de la projection horizontale). Enfin les sous-blocs restants sont regroupés sur des critères de proximité pour former les zones de texte. La base de test est composée de 50 images issues de vidéos diverses comportant des variations de taille et de police tout en conservant une bonne définition. Du fait du nombre très important de seuils à fixer expérimentalement, c'est une approche difficilement généralisable.

Mariano *et al.* [56, 55] reprennent ce principe d'analyse ligne à ligne mais en regroupant les segments sur des critères colorimétriques.Dans [55] les auteurs l'appliquent sur des textes en surimpression et pour la détection de texte sur des objets en mouvement (eg :voitures, cars).

Wong and Chen [113, 114] s'intéressent eux au gradient de l'image de luminance. Sur l'hypothèse que les zones de texte posséderont de fortes variations locales du gradient (nombreux passages entre fond et lettres sur une zone de texte), les segments potentiels pour chaque ligne sont sélectionnés. Les segments sont ensuite regroupés d'une ligne à l'autre sur des critères statistiques (moyenne, variance) calculés sur les niveaux de gris de l'image. Les blocs extraits sont filtrés sur des critères géométriques (aire minimale/maximale, hauteur, largeur, ratio, ...). Les zones de texte candidates extraites passent ensuite par un module de segmentation et restauration pour extraire les caractères binarisés.

Enfin Sin et al. [85] proposent une approche progressive : ils détectent les forts gradients horizontaux (par seuillage du gradient de Roberts) et conservent pour chaque ligne tous les segments comportant un nombre suffisant de ces contours. Si le nombre de ces segments (pour une ligne) dépasse un certain seuil, la ligne est validée. Les lignes adjacentes sont ensuite regroupées en bloc et seuls les blocs d'une hauteur suffisante sont conservés. Pour chaque bloc, on calcule de manière similaire le nombre de contours verticaux et horizontaux dans des fenêtres glissantes, toutes les régions comportant un nombre suffisant de contours significatifs sont conservées. Ensuite chaque région est transformée en une séquence unidimensionnelle, et la région est validée comme zone de texte si la fonction d'autocorrélation de son spectre de Fourier contient un nombre significatif de pics. Cet algorithme est utilisé dans le cas de la détection de *longs* textes sur des panneaux routiers. Cet algorithme comporte un grand nombre d'heuristiques non explicitées et ne peut être utilisé pour la détection de texte court.

4.2.3 Approches dérivatives

Pour ces approches, on ne regarde plus une zone de texte comme un assemblage de lettres mais plutôt comme une zone dense de traits, i.e. on va assimiler la zone de texte à une texture comportant des motifs plus ou moins réguliers.

Remarque 5 *En fonction de la taxonomie suivie par les auteurs, ces approches se retrouvent souvent dans une catégorie "approche texture".*

Un des problèmes majeurs est que l'on fait une supposition forte concernant la régularité de distribution des traits et de l'espacement entre ceux-ci pour une zone de texte. Si ce type de supposition est vérifiée *en moyenne* pour des chaînes relativement longues, elle n'est plus valide dès que l'on s'intéresse à des chaînes courtes et isolées (voir Section 3.2.1 "Distance inter-caractère" p.22). A cela s'ajoutent les problèmes de la variabilité des fontes et styles employés ainsi que l'inclinaison possible des chaînes.

Aussi si maints travaux de recherche de texte dans les images et les vidéos utilisent des descripteurs texturaux de "haut niveau" (eg : filtre de Gabor, transformée de Fourier, ondelettes) d'autres

se sont tournés vers des descripteurs plus élémentaires, basés sur des statistiques du premier ordre (eg :gradient, gradients cumulés).

On peut résumer ces approches du premier ordre en quatre grandes étapes :

1. Seuillage de la réponse du détecteur : gradient niveaux de gris, couleur, gradients accumulés,...
2. Regroupement par "smearing" ou algorithme morphologique
3. Détection des lignes de base des zones de texte et création des boîtes englobantes équivalentes
4. Filtrage des boîtes englobantes par des prédicats géométriques

Smith and Kanade [88] extraient par seuillage du gradient, les forts contours verticaux de l'image. Les contours isolés sont supprimés et les autres regroupés par des filtres morphologiques. Les boîtes englobantes équivalentes sont calculées et soumises à un certain nombre d'heuristiques (eg : taille, ratio hauteur/largeur, taux de remplissage des boîtes). L'algorithme est validé sur soixante images issues de journaux télévisés.

De nombreux travaux utilisent cette méthode de seuillage du gradient, celles-ci diffèrent par les pré-traitements appliqués à l'image, par l'algorithme de binarisation mis en place (global ou adaptatif) ainsi que par la complexité des post-traitements mis en oeuvre. On pourra citer Sato et al. [80] qui reprennent le même algorithme en ajoutant des modules de restauration pour rendre les zones de texte acceptables pour un OCR commercial. Hasan and Karam [27] proposent de prendre une image lissée par des opérateurs morphologiques comme pré-traitement à une détection de contours par gradient morphologique. Une amélioration par pré-détection grossière des zones candidates a été proposée par Byun et al. [10]. Plus récemment Chen et al. [15, 14] ont utilisé cette détection de fort gradient comme un module de pré-localisation des zones potentielles pour la recherche de textes en surimpression.

Du fait de la présence de frontières (hautes et basses) entre une zone de texte et le fond, certains auteurs ont proposé de coupler la détection de gradient, avec une détection des coins. Aussi Hua et al. [30] proposent dans le cadre de détection de texte en surimpression (Base de 90 images clés issus des journaux de C.N.N), une pré-localisation basée sur le détecteur de coin de SUSAN [89] suivie d'une décomposition des zones détectées par l'étude des projections des gradients verticaux et horizontaux. Lyu et al. [54] proposent (dans le cadre d'une localisation de texte multi-langue en surimpression) de calculer une carte des contours en utilisant une version modifiée du gradient de SOBEL favorisant les régions riches en coins ; cette carte est soumise à un seuillage adaptatif détectant si l'on se trouve sur un fond complexe ou non, puis les zones retenues sont découpées itérativement par projections horizontales et verticales pour former les zones de texte potentielles ; ce processus est reproduit pour différents sous-échantillonnages de l'image pour obtenir un système multi-échelle.

Lebourgeois [40] quant à lui propose d'intervertir la phase de groupement et de binarisation, il procède à l'accumulation des gradients sur une fenêtre glissante horizontale, l'image est ensuite découpée par suivi des profils de projections horizontaux et verticaux. Le système "HWDavid" classé deuxième de la compétition ICDAR 2003[53] ré-utilise cet outil comme première étape de leur chaîne de localisation. Enfin cette approche est complétée par Wolf and Jolion [112] qui mettent en place des heuristiques (géométriques et morphologiques) de sélection plus robustes et intègrent le support de la redondance temporelle pour la détection de texte dans les vidéos.

Dans les systèmes utilisant des descripteurs texturaux de plus haut niveau on pourra citer les travaux de Wu et al. [115, 116]. Ils proposent pour la localisation des zones de texte (sur 48 images comprenant des documents scannées et des images issus du web) de calculer en chaque point de l'image un vecteur d'attributs en réponse à des filtres du type "dérivées de gaussiennes". L'image des attributs est ensuite classifiée en $[Texte, fond, autre]$. L'image correspondant au label texte sert ensuite de masque à un processus classique de détection de contours puis d'agrégation. Cet algorithme est repris par Meyers et al. [61] qui l'utiliseront comme module de localisation, pour des images de

"posters" prises sous de nombreux angles de prises de vues (leur approche de correction de perspective est présentée en Section 4.2.5 p.33). Le système "Todoran" de la compétition ICDAR 2003[53] (classé dernier sur quatre participants), proposera une amélioration de ce système en calculant les descripteurs pour chaque canal de l'espace RGB et en modifiant l'algorithme de regroupement spatial pour la prise en compte de l'information colorimétrique.

Chen et al. [16] quant à eux utilisent une détection de contours multi-échelle basée sur des filtres LOG (Laplacian of Gaussian), pour la *reconnaissance* de signes multilingues dans les scènes[2]. Les contours non significatifs sont éliminés à l'aide d'heuristiques (basées sur leur intensité, la taille de l'ensemble qu'il représente et les différences de contraste de cet ensemble avec le fond). Les contours significatifs attenants sont ensuite fusionnés récursivement à l'aide des heuristiques précédentes. Les composantes connexes correspondant aux contours finaux sont ensuite regroupées sur des critères géométriques et colorimétriques.

4.2.4 Approches utilisant de manière intensive l'apprentissage

L'ensemble des systèmes proposés dans les sections précédentes sont entachés d'un trop grand nombre d'heuristiques, ce qui pose le problème de leur généralité. De plus même si l'on peut assimiler le texte à un certain type de texture, nous avions déjà souligné le problème de l'utilisation de descripteurs simples mais faiblement discriminants. C'est pourquoi des travaux utilisant massivement les algorithmes d'apprentissage ont été proposés pour contribuer conjointement à la réduction des fausses détections et l'amélioration de la généralisation.

Le concept général de ces systèmes est schématisé sur la Figure 4.1 : les distinctions entre systèmes apparaîtront aux étapes de sélection des attributs et aux classifieurs retenus.

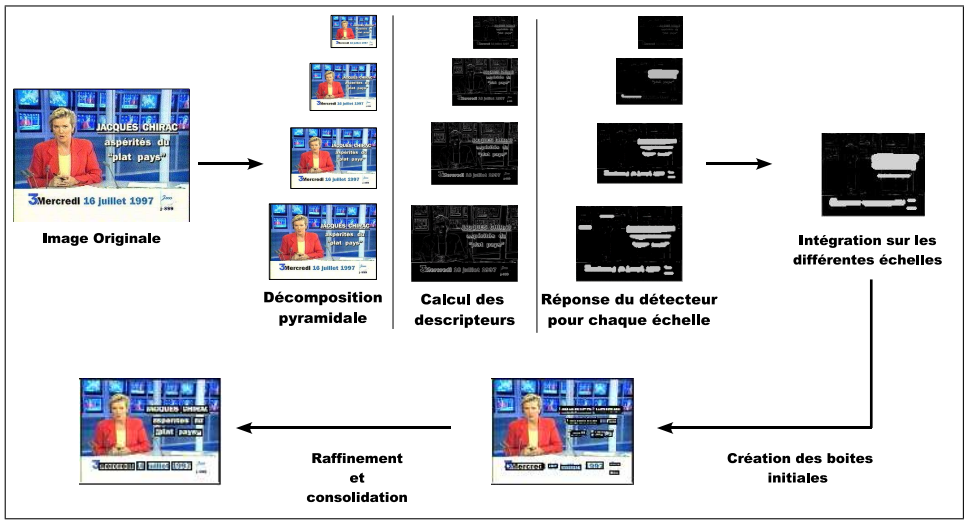


FIG. 4.1: Exemple didactique. Schéma algorithmique type d'un système de localisation de texte utilisant l'apprentissage

Les descripteurs sont calculés en chaque point de l'image sur des voisinages de taille fixe (i.e.fenêtres glissantes) intégrant ainsi l'invariance par translation. Ce processus est ensuite répété au travers d'un

[2]L'application visée étant la reconnaissance, les auteurs ne proposent pas de scores de localisations.

"espace d'échelles" pour intégrer l'invariance d'échelle.

Parmi les descripteurs les plus usités, on pourra citer :

- Mesure de la variance locale
- Estimation de la *force* des gradients Niveau de Gris ou couleur
- Coefficient d'ondelette
- Histogramme de la distribution des contours

On passe ensuite à la sélection des portions d'images candidates pour chaque échelle par l'intermédiaire d'un classifieur préalablement entraîné. On rétro-projette les zones candidates à l'échelle initiale et on crée les boîtes englobantes. On passe ensuite par différentes étapes de raffinement de ces boîtes.

Li et al. [45] sur une base de vidéo mêlant texte sur-imposé et texte enfoui, utilisent la décomposition en ondelettes de Harr. Une fenêtre glissante parcourt l'image et les coefficients de la décomposition sont soumis à un Perceptron Multi-Couche (*M*ulti*L*ayer *P*erceptron) qui labellise la zone en Texte/Non-Texte. Clark and Mirmehdi [19] utilisent également un M.L.P pour détecter de larges zones de texte enfoui dans des images "d'intérieur". Parmi les descripteurs utilisés on pourra citer l'histogramme des variances locales, la densité de gradients et la distribution de leur orientation. Lienhart and Wernicke [51] utilisent une image de la force des gradients couleur (dans les directions horizontales et verticales, somme des valeurs du gradient sur les trois canaux RGB) : les mesures prises sur une fenêtre glissante de 20*10 sont soumises à un M.L.P. Tang et al. [93] proposent une méthode non supervisée de détection des apparitions/disparitions des zones de texte en surimpression utilisant un Fuzzy Clustering Neural Network. Kim et al. [39] proposent de donner comme descripteurs à une Machines à Vecteurs de Support (*S*upport *V*ector *M*achines) les niveaux de gris sur un voisinage de type "Star Pattern" (ce type de pattern avait été proposé initialement dans [35]), un processus itératif d'identification des régions (dérivé de l'algorithme "Mean Shift") est ensuite utilisé pour sélectionner les zones d'intérêt sur l'image classifiée. Ce procédé est réutilisé par Jung et al. [36] comme module d'un système, traitant à la fois des documents couleur complexes et des textes surimposés, enfin Lee et al. [41] l'utilisent comme détecteur de texte surimposé dans les vidéos.

Wolf [110] propose un jeu de descripteurs comprenant la force des gradients accumulés (cf.Section 4.2.3 p.29) et l'estimation de la hauteur de la ligne de base potentielle pour la détection de texte en surimpression et de publicités. Toujours pour la détection de texte en surimpression Chen et al. [15, 14], Chen [11] sélectionnent tout d'abord des régions d'intérêt par un algorithme rapide du type "Approche dérivative". Ils calculent ensuite un certain nombre de descripteurs (Gradient, Carte des distances, Variance du gradient et coefficient DCT) sur des fenêtres glissantes, une comparaison des réponses d'un M.L.P et d'une S.V.M est proposée.

Enfin Ye et al. [118] proposent une approche itérative : dans une première passe, ils procèdent à une décomposition en ondelettes de Daubechies pour n échelles de l'image, les pixels candidats pour chaque échelle sont classifiés (Texte/Non Texte) sur l'intensité des coefficients haute fréquence de la décomposition. Des germes sont initialisés à l'endroit où la densité de pixels classifiés comme texte est suffisante, et un processus de croissance de régions basé sur la connexité et l'intensité des coefficients est réalisé pour créer les zones potentielles de texte. Ces zones pouvant être constituées de plusieurs lignes de texte, un algorithme de découpage basé sur les projections de profils est initialisé. On passe ensuite à une deuxième passe où pour chaque bloc retenu, on calcule un jeu de descripteurs basé d'une part sur des statistiques concernant les coefficients d'ondelette et d'autre part sur les passages par zéro

du gradient le long du bloc. Ce jeu est soumis à une S.V.M couplé à une méthode *bootstrap*[3]. Toutes les zones retenues pour chaque échelle sont ensuite rétro-projetées à l'échelle initiale.

Il est important de noter que nombre de ces algorithmes n'ont pas complètement rempli leurs objectifs, d'une part le nombre d'heuristiques pour ces systèmes reste important, d'autre part les algorithmes d'apprentissage sont extrêmement dépendants de leur base d'entraînement. Même s'ils permettent de réduire le nombre de fausses détections pour une application donnée, le problème de la généralisation des systèmes d'extraction de texte reste un problème ouvert.

4.2.5 Approches spécifiques pour le texte orienté :

La plupart des travaux en vidéo-texte se sont concentrés sur des textes "artificiels" mêlant texte en surimpression et publicités. 95% de ces textes sont plans et horizontaux. La hausse de complexité des systèmes pour récupérer 5% des instances manquées est considérée comme prohibitive et contre-productive pour un système d'indexation [48].

Même dans le cadre du texte enfoui, très peu d'auteurs ont mis en place des stratégies pour localiser des textes non plans et le plus souvent seule une tolérance sur l'orientation est proposée ou alors il s'agit de texte projeté sur des surface planes (plaques d'immatriculation, panneaux routiers, ...) et l'algorithme est décomposé en deux étapes, détection du support et correction géométrique puis localisation du texte. On pourra par exemple regarder les travaux de Wu et al. [117] pour la localisation temps-réel du texte sur des panneaux routiers.

Certains travaux spécifiques ont tout de même été proposés. Messelodi and Modena [58] proposent dans une application d'extraction de titres de livres, d'étudier l'alignement des composantes connexes obtenues par seuillage adaptatif de l'image. Ils associent à chaque composante soit un barycentre s'il s'agit d'un caractère ou n barycentres (si l'hypothèse de caractères connectés est retenue). Ils observent ensuite pour un ensemble d'angles donné la distribution des pentes des segments reliant chaque barycentre. La direction majoritaire du texte correspond à la direction d'entropie minimale. Meyers et al. [61] pour la correction d'orientation de texte présent sur des panneaux d'affichage combinent la détection des points de fuite du support et des lignes de texte. Enfin Clark and Mirmehdi [20] détectent directement les points de fuite de larges paragraphes de texte plongés dans une scène.

4.2.6 Les approches basées sur la compression

L'idée générique de ces approches repose sur différentes caractéristiques pré-supposées du texte : une ligne de texte composée de caractères doit fournir une réponse importante dans les harmoniques horizontales dûes au changement rapide d'intensité introduite par les alternances de vallées et de crêtes. On espère également détecter des variations dans les harmoniques verticales dues à l'espacement entre différentes lignes de texte. Toutes ces informations sont capturées et encodées par les algorithmes de compression et de transmission et sont donc disponibles sans calcul supplémentaire.

Zhong et al. [123] proposent un algorithme utilisant les coefficients DCT pour la détection de texte dans les images JPEG et vidéo MPEG. Ils sélectionnent par seuillage l'ensemble des blocs possédant une forte énergie horizontale et verticale puis procèdent à un nettoyage des zones détectées par des post-traitements morphologiques et une analyse en composantes connexes. Suivant le même schéma Zhang et al. [119] ajoutent une étape de vérification basée sur la redondance temporelle des zones de

[3] Attention nous ne parlons pas ici des méthodes de Efron and Tibshirani [24], il s'agit simplement ici d'une méthode itérative de sélection des ensembles texte/non texte

texte (voir 4.2.7 p.34). Et Crandall et al. [21] étendent cette étape de vérification à des zones de texte subissant des rotations et/ou des changements d'échelle.

Ces méthodes proposent des localisations grossières des zones de texte mais ont l'avantage d'être extrêmement rapides.

4.2.7 Apport de la redondance temporelle

Pour le texte présent dans les vidéos, on peut tirer parti de la redondance temporelle pour améliorer la phase de localisation. En effet chaque ligne de texte sera présente sur un nombre contigu de trames. Cette redondance est particulièrement utilisée dans le cas des textes en surimpression.

On peut l'exploiter de différentes manières :

1. Accroître la potentialité de détection, la même zone de texte apparaissant dans des configurations plus ou moins complexes le long d'une séquence.
2. Réduire le nombre de fausses détections et de zones de texte manquées en vérifiant la constance de la détection sur plusieurs trames.
3. Fournir pour un système d'indexation une seule détection intégrant l'aspect temporel

Dans la plupart des systèmes on va tout d'abord étudier "grossièrement" la vidéo puis affiner ce résultat : on n'extrait de la vidéo qu'une trame toutes les secondes ou toutes les demi-secondes. Pour cette trame on lance le module de localisation de texte. Si une instance de texte est trouvée pour une trame t donnée, on calcule une signature de cette instance (comprenant des caractéristiques de position, taille, information colorimétrique, ...) et on lance une étape de suivi de la zone de texte le long de n trames précédant et suivant la trame t. Pour ces n trames on cherche les zones de l'image ayant une signature similaire.

Deux cas peuvent apparaître :

1. Aucune instance de texte possédant une signature similaire n'est trouvée, on considère qu'il s'agit d'une fausse détection.
2. Des instances de texte possédant une signature similaire sont trouvées, on considère qu'il s'agit d'une seule et même zone de texte. Et on en conserve l'ensemble des occurrences.

Un exemple type de cette approche est proposé par Lienhart and Wernicke [51] : pour l'étape de suivi il utilise un algorithme de "block matching" prenant en compte le recouvrement des boîtes englobantes sur plusieurs trames et le calcul de correspondance entre les boîtes basé sur des signatures extraites des projections de lignes de profil.

Li et al. [45] proposent sous l'hypothèse de mouvement purement linéaire des zones de texte, d'utiliser un simple "block matching". Wolf [110], sur l'hypothèse d'un texte statique, propose de tester le recouvrement en hauteur, position et aire des boîtes englobantes ainsi qu'une mesure similaire à celle de Lienhart. Enfin Crandall et al. [21] proposent deux algorithmes de matching, le premier pour des textes au déplacement linéaire combine l'approche de Li et la prise en compte des vecteurs de mouvement issus du codage MPEG, le second pour des textes avec "effets spéciaux" comparent des signatures extraites de versions binarisées des zones de texte.

4.3 Restauration

Comme nous l'avions souligné à la Section 3.2 (p.18), les O.C.R obtiennent aujourd'hui d'excellentes performances pour la reconnaissance de caractères dans les documents scannés. Cependant beaucoup de chercheurs ne considèrent pas les O.C.R. suffisamment matures pour appréhender le

problème de la reconnaissance de texte en sur-impression ou enfoui et ne considèrent pas la sortie d'un O.C.R. comme une métrique valide. Aussi nombre de systèmes ne font que localiser les zones de texte.

Concernant les zones de texte en surimpression, la faible résolution des textes dans les vidéos couplée avec différentes dégradations (eg :effets d'anti-aliasing, de sous-échantillonnage et d'artefact de compression) impose de procéder en deux étapes :

- une phase de restauration comprenant la suppression des fonds complexes ainsi qu'une augmentation artificielle de la résolution.
- un module de segmentation déterminant la polarité[4] du texte et fournissant la meilleure segmentation possible pour un **O.C.R. donné**.

Concernant le texte enfoui, tout dépendra de la complexité du texte auquel on aura affaire.

Il est d'ailleurs important de noter que le problème de la reconnaissance de texte non contraint (eg :bruité, comportant des connections entre caractères, des lettres brisées, des fontes complexes) est toujours considéré comme un problème ouvert. D'ailleurs de nombreux C.A.P.T.C.H.A[5] utilisent le fossé entre les capacités humaines et les systèmes actuels de reconnaissance de caractères. On trouvera des exemples de ces travaux dans [5] et [18].

4.3.1 Restauration dans une image fixe :

4.3.1.1 Augmentation de la résolution :

Une des premières étapes de restauration est d'augmenter artificiellement la résolution des zones localisées. On procède à des interpolations sub-pixéliques (bilinéaires, bicubiques, ...) en conservant le ratio des zones de texte. Communément on prend comme référence la hauteur des boîtes englobantes que l'on étend dans un domaine de 40 à 100 pixels [48].

4.3.1.2 Restauration au niveau des caractères :

Certains auteurs ont aussi proposé des restaurations au niveau des caractères : Sato et al. [80] proposent quatre filtres directionnels visant à renforcer les traits des caractères, cette approche est réalisée une fois les zones localisées et pour une échelle donnée. Chen et al. [13] déplacent cette approche avant la phase de localisation et l'intègre sur différentes échelles.

4.3.2 Restauration dans les vidéos

Dans le cas de la vidéo on peut tirer également parti de la redondance temporelle pour supprimer le fond complexe entourant les caractères. On part de l'hypothèse que, le long de la séquence, les valeurs de gris des zones de texte sont relativement constantes par rapport à celles du fond, ceci se vérifiera dans le cas de texte statique ou de texte possédant un mouvement différent du reste de l'image.

Aussi pour du texte statique on peut simplement proposer un opérateur Min/Max temporel si l'on connaît la polarité du texte, median/moyenne sinon. Ce type d'opérateur a été initialement proposé par Sato et al. [80] pour du texte purement statique.

Pour le texte en mouvement, on collecte les différentes occurrences d'une même zone de texte trouvées grâce à l'étape de localisation (en s'aidant au besoin d'algorithme de type "block matching"

[4] texte claire sur fond sombre et inversement

[5] Completely Automated Public Turing Test To Tell Computers And Humans Apart `www.captcha.net`

voir 4.2.7 p.34), et on applique des opérateurs du type min/max/median/moyenne sur cette collection, voir [44, 43, 51, 110].

4.4 Extraction

La phase d'extraction consiste, en partant des boîtes englobantes localisées, à binariser et à nettoyer la zone de texte pour la rendre la plus "acceptable" possible par un module d'O.C.R. Elle est généralement proposée par les systèmes extrayant du texte en sur-impression.

De manière générale on part des boîtes englobantes localisées et dont l'intérieur a été *restauré* (cf Section précédente). Pour l'aspect segmentation, l'algorithme de maximisation de la variance interclasse (Ohtsu/Fisher[66]) est encore l'un des plus utilisés. Plus généralement ce sont des variantes locales appliquées à l'intérieur des boîtes englobantes qui sont mises en place pour prendre en compte le problème des fonds complexes. La phase de nettoyage comporte souvent des filtres morphologiques supprimant toutes les composantes touchant le bord de la boîte [54, 51].

D'autres méthodes ont été cependant proposées, un simple seuillage est par exemple utilisé par Sato et al. [80] après le réhaussement du contour des traits des caractères. Wolf [110] propose pour le texte dans les vidéos une variante de l'algorithme de Niblack [64] basée sur la maximisation du contraste local. Cette variante est comparée à d'autres algorithmes utilisés en analyse de documents [95].

Chen *et al.* [14, 11] proposent non pas un seuillage en deux classes mais multiclasse avec votes : des seuillages automatiques pour 2, 3 et 4 classes sont proposés, les images obtenues sont soumises à une analyse en composantes connexes et transmises à l'O.C.R, une modélisation de la réponse de celui-ci permet de choisir la chaîne résultat.

Remarques

Polarité des zones de texte : dans beaucoup d'applications, la polarité de la zone de texte n'est pas connue à l'avance, aussi elle est souvent déduite a posteriori par une analyse en composantes connexes des images binaires générées.

Les approches par segmentation et regroupement spatial (voir Section 4.2.1 p.26) n'ont généralement pas besoin de ce module, car la sortie de leur module de localisation comprend généralement la détection des caractères eux-mêmes.

4.5 Critère d'évaluation

Pour pouvoir valider ou invalider un algorithme de localisation et d'extraction de zones de texte dans les images et les vidéos, on doit pouvoir le soumettre à une évaluation la plus objective possible. Nous allons voir dans cette section que cette démarche est difficile à mettre en oeuvre.

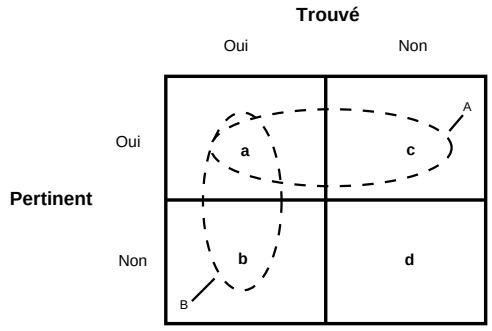
4.5.1 Rappels :

De manière générale un système de recherche d'information est évalué selon les mesures suivantes :

Définition 11 ***Critère d'évaluation :*** *L'évaluation de la qualité d'un système passe par le calcul de critères statistiques sur des bases dont le contenu est maîtrisé. Les critères les plus communément utilisés sont les critères dis de* RAPPEL *et de* PRÉCISION.

- *Le rappel correspond à la proportion de réponses pertinentes retrouvées par le système par rapport au nombre total d'instances pertinentes de la base.*
- *La précision est la proportion de réponses pertinentes retrouvées par rapport au nombre total d'instances retrouvées.*

En complément on pourra ajouter deux indicateurs le TAUX DE FAUX POSITIFS et le TAUX DE FAUX NÉGATIFS.



Correspondance :

· RAPPEL $= \frac{a}{a+c}$

· PRÉCISION $= \frac{a}{a+b}$

· TAUX DE FAUX POSITIFS $= \frac{b}{b+d}$

· TAUX DE FAUX NÉGATIFS $= \frac{c}{a+c}$

On peut également résumer la performance du système en utilisant la moyenne harmonique entre la PRÉCISION et le RAPPEL :

$$\text{MOYENNE HARMONIQUE} = \frac{2 \times \text{PRÉCISION} \times \text{RAPPEL}}{\text{PRÉCISION} + \text{RAPPEL}}$$

4.5.2 Application à la recherche de texte :

Pour la recherche de texte, on peut placer cette évaluation à deux endroits du système correspondant aux deux approches suivantes :

APPROCHE GÉOMÉTRIQUE : Elle prend en compte la phase de localisation, le plus souvent la qualité de recouvrement entre les boîtes englobantes de la vérité terrain et les boîtes englobantes trouvées par le système

APPROCHE CONTENU : elle prend en compte la phase de reconnaissance et doit pénaliser les instances de texte ratées ou ajoutées mais sans s'intéresser explicitement à leur localisation. La sortie d'un tel système sera l'ensemble des chaînes de caractères présents dans l'image ou la vidéo requête. On notera que cette évaluation fera intervenir le module externe d'O.C.R. et les performances intrinsèques de celui-ci pourront imposer de nouvelles limitations au système.

4.5.2.1 Evaluation pour l'approche géométrique

La définition de mesure de qualité d'un système de localisation de texte est à la base un problème "mal posé" ; ainsi la comparaison objective des algorithmes de localisation se heurte aux problèmes suivants :

– **Définition de la vérité terrain :** on devrait pouvoir définir pour la phase de localisation une base de connaissance qui devrait être indépendante de l'algorithme utilisé. C'est le plus souvent impossible, chaque système proposant une granularité de localisation différente (au niveau

caractères, mots, lignes, groupes de lignes). Aussi chaque système fournit une vérité terrain en adéquation avec cette granularité.
- **Mesure proposée :** la mesure proposée est fortement dépendante de la granularité de la vérité terrain.
- **Application recherchée :** en fonction de celle-ci, tout ou seulement un sous-ensemble des zones de texte doit être retrouvé.

On présentera ci-après la mesure de performance communément utilisée dans la littérature ([110, 53, 32]). Elle se base sur les pourcentages de recouvrement entre les boîtes englobantes de la vérité terrain et les boîtes englobantes détectées par le système :

$$\text{RAPPEL} = \frac{\sum_i Match_G(G_i)}{card(G)} \quad \text{PRÉCISION} = \frac{\sum_j Match_L(L_j)}{card(L)}$$

avec

$$Match_G(G_i) = \max_j \frac{2 \cdot |G_i \cap L_j|}{|G_i| + |L_j|} \quad Match_L(L_j) = \max_i \frac{2 \cdot |L_j \cap G_i|}{|L_j| + |G_i|}$$

où

$|.|$ est l'aire de la boîte (i.e. le nombre de pixels)

où $L = \{l_1, \cdots, l_N\}$, $G = \{g_1, \cdots, g_M\}$ représente respectivement l'ensemble des boîtes englobantes générées par le système et celles de la vérité terrain.

Cette métrique souffre de trois inconvénients majeurs (cf. [111]) :

Perception : Le taux de recouvrement des boîtes n'est pas une mesure perceptuellement valide. En effet pour la Figure 4.2, la mesure de précision et de rappel est la même pour les deux détections proposées en (a) et (b), alors que la détection (a) semble tout de même bien plus satisfaisante.

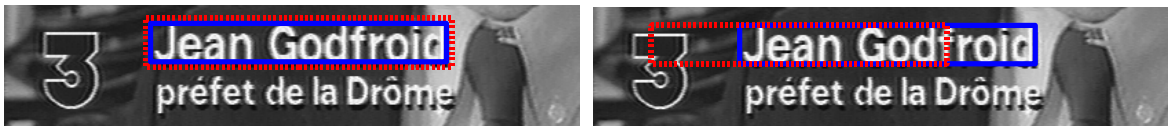
Ambiguïté : C'est une mesure ambiguë : si on prend par exemple un taux de RAPPEL de 50%, de manière abusive on pourrait conclure soit que 50% des boîtes de la vérité terrain ont été localisées parfaitement, soit que toutes les boîtes de la vérité terrain ont été localisées avec un recouvrement de 50%. De manière générale on se trouvera entre ces deux extrêmes mais on ne pourra pas réellement conclure ni sur la qualité de la localisation, ni sur le nombre de zones de texte "ratées",...

Granularité : Cette mesure ne propose que des correspondances "une boîte vers une boîte" (one-to-one). On peut très bien avoir un système proposant des détections partielles ou ne correspondant pas à la granularité de la vérité terrain. Dans les deux cas la métrique tendra à fortement pénaliser ce type de système. On consultera la Figure 4.3 pour des illustrations de ces problèmes pour des textes en sur-impression.

Le fait que cette mesure utilise un simple recouvrement des boîtes couplé au problème de la granularité en fait un outil peu adapté pour le texte enfoui où l'on peut trouver beaucoup de texte penché, non plan ou utilisant des effets de style.

Il existe différentes variantes plus ou moins permissives de cette métrique (voir [51],[11],[54]) mais aucune ne résout de manière univoque les problèmes présentés.

Possibles améliorations des métriques précédentes: Huiping [32] et HUA et al. [31] proposent d'estimer la *lisibilté* des zones de texte et ainsi de pondérer les résultats de leur systèmes en fonction de cette *lisibilité*. Malheureusement ces critères de lisibilité sont trop souvent corrélés aux descripteurs utilisés dans l'algorithme de localisation de texte.



(a) (b)

FIG. 4.2: En ligne continue la boîte de la vérité terrain, en pointillé la boîte détectée. Pour les deux cas (a) et (b) les mesures de PRÉCISION et RAPPEL sont équivalentes

(a) (b)

FIG. 4.3: En ligne continue la/les boîte(s) de la vérité terrain, en pointillé la/les boîte(s) détectée(s). Bien que dans les deux cas (a) et (b) le résultat de la détection soit perceptuellement correct, on dépend de la granularité de la vérité terrain. Dans les deux cas on obtient un mauvais score de localisation.

Une première amélioration tangible de ces métriques est de proposer des mesures "one to many" et "many to one" (voir Figure 4.3) entre boîtes pour s'affranchir au mieux possible des problèmes de granularité. Ce type d'approche a été proposée par Wolf([110, 111]).

4.5.2.2 Evaluation pour l'approche contenu :

En ce qui concerne l'approche contenu, il n'y a pas de consensus. Les trois mesures les plus couramment citées sont :

- **L'inspection visuelle :** une simple inspection visuelle de la qualité de la segmentation obtenue.
- **L'utilisation d'un O.C.R. :** une évaluation indirecte par la performance moyenne de l'O.C.R. (le plus souvent par comparaison de chaînes [104]), ce qui rend l'application dépendante de l'O.C.R. et de ses particularités
- **Erreur de probabilité (EP) :** elle nécessite une segmentation de référence ce qui, dans la plupart des cas, est impossible à définir. Elle est définie comme suit (cf. [48]) :

$$EP = P(O)\,P(B|O) + P(B)\,P(O|B)$$

où $P(B|O)$ et $P(O|B)$ sont les probabilités d'erreur de classifier du texte comme fond et réciproquement avec $P(O)$ et $P(B)$ les probabilités a priori texte/fond dans la base de test.

L'évaluation basée sur le contenu est considérée comme un problème ouvert, aucune des solutions précédentes n'étant satisfaisante.

Autres applications de l'extraction de texte

Nos recherches personnelles se focalisant particulièrement sur l'apport de l'extraction de texte pour l'indexation, c'est cette problématique qui a servi de fil conducteur à ce document. Le lecteur intéressé par d'autres applications pourra se reporter aux travaux de compilation de Jung et al. [36] et

Liang et al. [47]. Ces papiers traitent des applications suivantes : l'indexation (d'image et de vidéos) bien évidemment mais également

- **La reconnaissance de code** pour les véhicules de transport de marchandise.
- **La détection automatique de panneaux** pour l'aide à la circulation routière.
- **La reconnaissance de panneaux indicateurs** pour l'aide aux personnes malvoyantes.
- **La reconnaissance de plaques minéralogiques**
- **La reconnaissance de texte manuscrit** en ligne ou hors ligne.
- **La digitalisation d'ouvrages** à l'aide de caméra (à mettre en opposition avec l'utilisation du scanner).
- **La structuration de documents**

On pourra les compléter par les premiers travaux de détection de *spam basé sur des images* (voir pour exemple les travaux de [4]).

4.6 Conclusion

Le survol proposé des travaux d'extraction de texte dans les images et les vidéos a permis de mettre en lumière deux grandes catégories d'approches.

La première est l'approche dite *texture* (qui inclut les approches dérivatives et fréquentielles). Ces méthodes ont prouvé leur efficacité dans différents cadres applicatifs et particulièrement dans le cas du texte en sur-impression. Leur couplage avec les espaces d'échelles et les techniques d'apprentissage ont permis des avancées importantes. Cependant elles restent trop souvent dépendantes d'a priori importants sur les zones de texte (eg :longueurs minimales des chaînes de caractères, contrainte d'horizontalité) ce qui réduit leur capacité de généralisation.

La seconde est l'approche par *composantes connexes*. Historiquement, c'est la première approche étudiée. Sa capacité à traiter différents types de textes a permis d'obtenir de bons résultats sur des applications d'analyse de documents complexes. Cependant le nombre pléthorique d'heuristiques utilisé la contraint à des domaines applicatifs restreints et maîtrisés. Ces méthodes ne localisent pas correctement le texte en sur-impression lorsqu'on a affaire à des images de qualité médiocre, de faible résolution ou compression. Aujourd'hui, au travers des travaux de détection de texte dans les scènes, cette approche revient en compétition avec l'approche *texture*.

D'un point de vue général, nous avons souligné la difficulté de fournir aujourd'hui une méthode générique et complète (i.e. allant de la détection des zones jusqu'à la reconnaissance par un outil de reconnaissance de caractères) pour toutes les applications du fait des variations énormes de types de textes rencontrés. D'ailleurs en toute objectivité la problématique de l'extraction de texte dans son ensemble reste encore un problème ouvert. Les récents travaux autour des *C.A.P.T.C.H.A* [6] basés sur des problématiques (qui semblent basiques) de reconnaissance de texte sont là pour le rappeler (voir [5] et [18]). La non participation d'équipes de recherche aux compétitions I.C.D.A.R de 'reconnaissance de mots' et 'reconnaissance de caractères' de 2003 [53] et 2005 [52] également.

Un des derniers points illustré dans cette partie est la difficulté de juger les algorithmes existants. Chaque module d'un *S*ystème d' *E*xtraction de *T*exte peut être de complexité différente et comparer des modules sans prendre l'ensemble du système en considération n'a pas de sens. A ceci s'ajoute la quasi absence de base de tests commune, ainsi que la difficulté de proposer des méthodes d'évaluation non-ambiguës et génériques permettant de s'affranchir de notre perception subjective de la qualité.

[6] Completely Automated Public Turing Test To Tell Computers And Humans Apart, `www.captcha.net`

5 Premiers pas
Réflexion sur un existant

Dans ce court chapitre, nous introduirons, le point de départ de nos recherches. Il s'agit d'une réflexion sur une chaîne d'extraction de texte qui a été développée dans la thèse de Claire Hélène Demarty, soutenue au CMM en 2000 [22].

5.1 Introduction de la problématique

Dans [22], Demarty effectuait différentes analyses sur des séquences vidéos de journaux télévisés. Le but était une caractérisation sémantique de ces séquences.

Une des premières tâches de ce travail concernait la structuration temporelle (eg :détection des transitons, sélection des images clés[1]). Un fois cette tâche accomplie, une des premières possibilités est d'extraire des objets d'intérêt présents sur les images clés (eg : zones d'incrustation, détection du locuteur, détection de **zones de texte**). C'est bien évidemment cette dernière possibilité qui va nous intéresser par la suite.

Dans la section suivante, l'algorithme développé sera présenté ainsi que les hypothèses sous-jacentes.

5.2 Présentation de l'algorithme

5.2.1 Hypothèses de travail

Les hypothèses de l'algorithme sont classiques pour l'extraction de texte en sur-impression (voir Lienhart [48]) :

CONTRASTE : Les zones de texte doivent avoir un contraste suffisant vis-à-vis de leur fond propre.

CONTRAINTES GÉOMÉTRIQUES :

- Taille : La taille des caractères doit être *suffisante* pour que le spectateur (de la séquence) puisse lire le texte.
- Alignement : Les caractères en surimpression apparaissent en grappe, et on impose qu'il faut un minimum de trois caractères pour définir une zone de texte. Ils sont de plus alignés horizontalement.

[1] KeyFrame : C'est une image considérée comme représentative du contenu informationnel d'une entité (prise de vue ou sous ensemble de prise de vue). En théorie l'union de toutes les images clés d'une entité est suffisante pour la résumer sans perte d'information

- Distance inter-caractère/inter-traits : Ici Demarty considère que la taille intrinsèque des caractères d'une même ligne, la distance inter-trait, ainsi que la distance séparant les mots d'une même phrase peuvent être regroupés en un seul paramètre de taille (noté N). Nous avions vu dans la Section 3.2 qu'il s'agit d'une hypothèse très forte. Cependant dans le cadre de la base d'images traitée et en utilisant judicieusement ce paramètre, cette simplification s'avère utilisable en pratique.

Nous remarquerons que les contraintes géométriques découlent directement des propriétés typographiques des fontes romanes. La Figure 5.1 en présente les éléments de base.

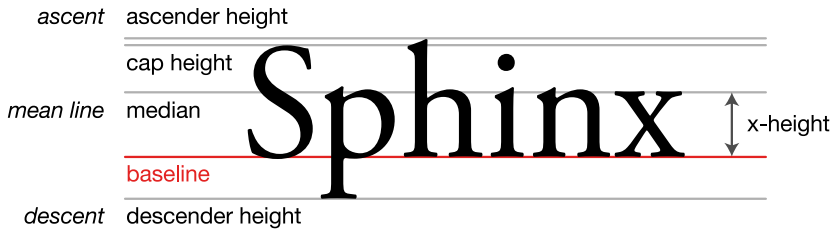


FIG. 5.1: Petit rappel de typographie : importance de la ligne de base dans les fontes romanes

Les hypothèses étant posées, passons à la description de l'algorithme proposé.

5.2.2 Descriptif de l'algorithme et pseudo-code

Le but de cet algorithme (voir 1) est de fournir une image très nettoyée, qui sera ensuite binarisée et soumise à un O.C.R.

Celui-ci est présenté en page 43. Il est définit pour la détection de caractères clairs sur fond sombre. Notons qu'un processus *dual* peut être mis en place si la polarité texte/fond est inversée.

Le descriptif de l'algorithme 1 est présenté ci-après. L'ensemble des étapes peut être suivi à l'aide de la Figure 5.2. Il se décompose en trois grandes étapes, dont la seconde est certainement la plus sensible :

1. Extraction des petits détails clairs : l.1 à l.3 : On extrait l'ensemble des petites structures claires et fines de l'image (sur le canal de luminance) à l'aide d'un chapeau haut de forme blanc de taille N (voir Figure 5.2(b)). Ici N est à choisir en fonction de *l'épaisseur* maximale des traits à extraire. Comme nous l'avons dit précédemment ce paramètre englobe la taille du trait et la taille de l'espace inter-caractère. Dans l'application visée, il a été fixé à une taille maximale de **10 pixels**, ce qui selon Demarty correspond à des caractères relativement larges pour des images issues de journaux télévisés en résolution 720*576 (i.e. **S**tandart **D**efinition **T**.**V**). Tout caractère dont l'épaisseur est supérieur à N ne pourra être récupéré par la suite.
2. Obtention d'une zone de détails horizontaux : l.3 à l.9 : C'est l'étape sensible de l'algorithme où le choix du paramètre N est crucial. Elle consiste à grouper les caractères sous forme de bandes horizontales. La méthodologie repose sur la propriété de *ligne de base* des fontes romanes et sur l'hypothèse d'horizontalité des zones de textes.

   Une fermeture horizontale de **taille** $2*N$ est appliquée, elle vise à connecter les caractères entre eux (voir Figure 5.2(c)), **ainsi que les mots d'une même phrase**. Puis une ouverture toujours horizontale de taille $3 * N$ est utilisée, elle vise à éliminer toutes les zones qui après l'étape de fermeture ne possèdent pas une taille suffisante (voir Figure 5.2(d)). Ceci permet de filtrer des petites composantes isolées et de regrouper les caractères en grappe. Les mots d'une taille

Algorithme 5.1 : Algorithme d'extraction de texte proposé par Demarty [22].

Prérequis : Polarité du texte considéré clair sur fond sombre. Le texte est horizontal. On a un a priori sur sa taille.

Input : $ImIn$:Image à teintes de Gris, N :Épaisseur/taille Maximale d'un caractère

Result : $ImOut$: Image nettoyée et **à binariser pour extraire les composantes connexes des lettres**

1 • ***Extraction des petits détails clairs :***

`// Extraction utilisant un chapeau haut de forme par reconstruction`

2 $imTopHatWhite \leftarrow$ Chapeau haut-de-forme blanc par reconstruction de $ImIn$, utilisant B élément structurant isotrope (i.e hexagonal) de taille N

3 • ***Obtention d'une zone de détails horizontaux :***

`// Connections des caractères par fermeture`

`// Notons ici que nous utilisons un segment de taille` $2*N$`, ceci permet à la fois de connecter les caractères entre eux, mais également les mots d'une même phrase séparés par un espace inférieur à cette taille`

4 $imClose \leftarrow$ Fermeture de $imTopHatWhite$, utilisant un segment horizontal de taille $2 * N$

`// Hypothèse d'au moins trois caractère alignés, qui ont été connectés lors de l'étape précédente`

5 $imOpen \leftarrow$ Ouverture de $imClose$, utilisant un segment horizontal de taille $3 * N$

`// Homogénéisation des niveaux de gris puis extension de la zone pour couvrir les jambages supérieur/inférieur et les majuscules`

6 $imFillHoles \leftarrow$ Bouchage de trous éventuels de $imOpen$

7 $im \leftarrow$ Fermeture par reconstruction de $imFillHoles$, utilisant un segment vertical de taille $N/2$

8 $imRescueBorder \leftarrow$ Dilatation de im, utilisant un élément structurant isotrope (i.e hexagone) de taille $N/2$

9 • ***Extraction des caractères clairs dans les zones horizontales extraites :***

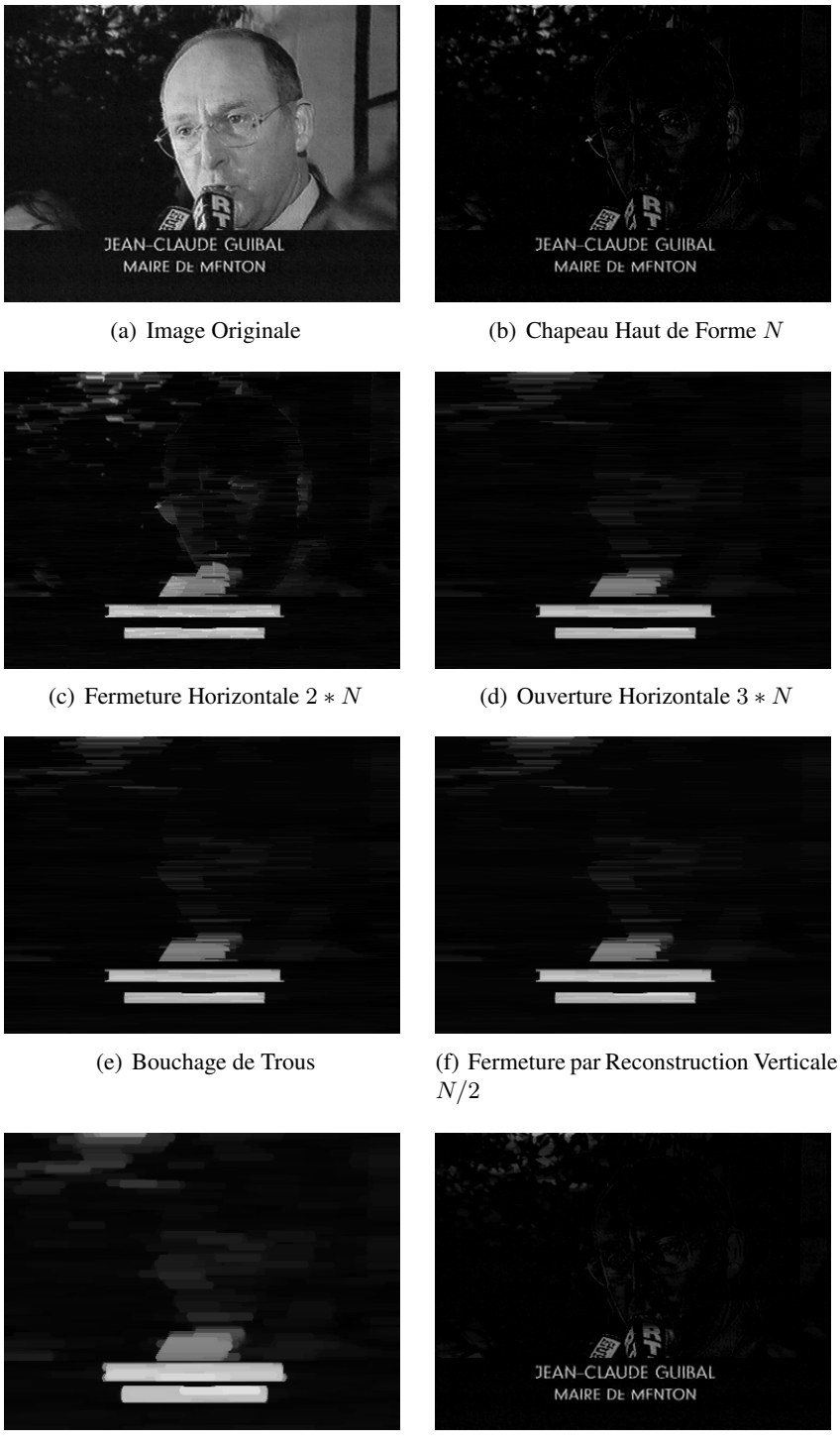
10 $ImOut \leftarrow$ Infimum($imRescueBorder$,$imTopHatWhite$)

inférieure à trois caractères sont ainsi éliminés, sauf s'ils ont été groupés avec d'autres mots plus longs de la même ligne.

L'extraction continue par un bouchage de trous éventuels dans chaque région (voir Figure 5.2(e)), puis on applique une fermeture par reconstruction verticale de taille $N/2$ visant à uniformiser les niveaux de gris au sein des régions (voir Figure 5.2(f)).

Puis une dilatation isotrope (ie utilisant un élément structurant hexagonal) de taille $N/2$ est appliqué. Elle permet de récupérer pour les régions couvrant des zones de texte, les portions de caractères présents au dessous/dessus de la ligne de base (voir Figure 5.2(g)).

3. Extraction des caractères : l.1 à l.3 : Enfin, nous avons à récupérer uniquement les petits détails clairs extraits par le chapeau haut de forme et qui sont compris dans les zones agrégées précédemment. L'infimum entre le chapeau haut de forme et le résultat de l'étape précédente permet cette récupération (voir Figure 5.2(h)). Il nous reste à binariser ce résultat.



(a) Image Originale

(b) Chapeau Haut de Forme N

(c) Fermeture Horizontale $2 * N$

(d) Ouverture Horizontale $3 * N$

(e) Bouchage de Trous

(f) Fermeture par Reconstruction Verticale $N/2$

(g) Dilatation Isotrope $N/2$

(h) Infimum

FIG. 5.2: Étapes de l'algorithme de détection de texte dans les images clés de Demarty [22].

(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

FIG. 5.3: Exemples de détections de texte pour différentes images issues de la base de Demarty [22]. De gauche à droite : image initiale, résultat de l'algorithme, seuillage *entropique*(cf Pun [69]) de ce résultat.

5.2.3 Forces et faiblesses de l'algorithme en l'état

La force de cet algorithme repose sur sa simplicité et sur le nombre très faible de paramètres dont il dépend. Pour une base d'images possédant des zones de textes aux propriétés similaires en terme de taille, contraste et alignement, il fournit des résultats pertinents comme l'ont montré les exemples précédents.

Cependant nous exposerons brièvement quelques critiques générales, toujours dans le cadre de l'extraction de texte dans les journaux télévisés :

1. Contraste du texte : l'utilisation d'un chapeau haut de forme en première étape de l'algorithme le rend sensible[2] aux variations locales du fond propre du texte . Ainsi sur l'exemple de la Figure 5.3(i), les mots "Godfroid" et "Drôme" se retrouvent à cheval sur des paliers de contraste différents, rendant par la même difficile l'étape d'extraction et de binarisation. Notons que ceci se produit assez fréquemment, le texte le long d'une séquence ne repose pas continuellement sur un fond coopératif.
2. La sortie de l'algorithme est une image nettoyée : le problème est que cet algorithme fait toute confiance à l'image clé sélectionné précédemment. Puis a un système d'O.C.R pour sélectionner/classifier les zones de textes des non-textes. Or comme nous l'avons souligné dans la partie introductive un tel pari est audacieux. Il faudrait pour plus de robustesse, intégrer les réponses sur une séquence d'images et procéder à une restauration temporelle et/ou spatiale (cf.Section 4.3.2 p.35)

Finalement, nous pourrions dire qu'il s'agit d'un algorithme quasi opérationnel (hormis l'étape de restauration avant soumission à un O.C.R) pour la détection de texte bien calibré dans les journaux télévisés.

Les réelles limitations de cet algorithme tiennent à sa difficulté de généralisation, que nous allons exposer dans la section suivante.

5.3 Discussion sur la généralisation de l'algorithme

Le but ici est de bien soulever les limitations à l'extension de l'algorithme précédent. Comme nous l'avons vu dans la section précédente (si les hypothèses sont vérifiées bien entendu), l'algorithme est simple, efficace et possède peu de paramètres. En d'autres termes, il répond à la problématique posée. Nous allons détailler ici pourquoi l'algorithme est difficilement généralisable en l'état.

5.3.1 Contrainte d'alignement :

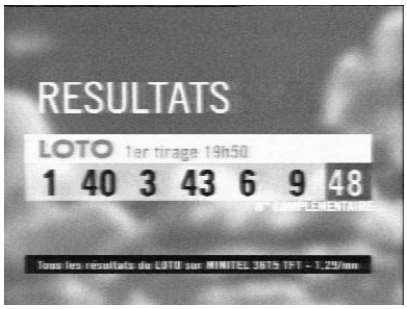
Une des premières remarques que l'on peut formuler, est la contrainte d'horizontalité extrêmement forte de cet algorithme [3]. Cette contrainte invalide la détection de textes penchés, même faiblement qui sont relativement courants si l'on traite des bases plus généralistes comme nous le verrons dans la suite du manuscrit.

5.3.2 Approche Multi-Résolution :

Cette seconde remarque est la plus importante : comment étendre l'algorithme précédant pour la prise en compte de plusieurs échelles de textes dans une même base d'images, ou au sein d'une même

[2] Notons cette sensibilité provient de la taille très faible d'ouverture utilisée
[3] Comme d'ailleurs la plupart des algorithme de détections de texte dans les vidéos

image ?. Les exemples de la Figure 5.4 rappelle cette contrainte tant pour les textes en sur-impression que pour les textes enfouis.



(a) Base Wolf [110]

(b) Base ICDAR [53]

(c) Base ICDAR [53]

FIG. 5.4: On ne peut pas se passer d'une approche multi-résolution pour prendre en compte la variabilité des textes dans les images. Pour les textes en surimpression au sein d'une même image en haut comme pour les textes enfouis issus d'une même base en bas.

Une première solution naïve serait de définir des *gammes/intervalles* : pour des textes possédant une taille entre T_1 et T_2, il faudrait utiliser le paramètre N_{12}. Ayant cette connaissance nous pourrions adjoindre des pré/post/traitement pertinents à la chaîne d'extraction. Cette solution échouera inévitablement, premièrement pour des problèmes d'effets de bord aux bornes des intervalles et deuxièmement elle ne fonctionnera que si la base d'images est relativement homogène (i.e les intervalles sont valables pour l'ensemble des images de la base).

Une solution plus réaliste serait de proposer une *vraie* approche multi-résolution utilisant des approches linéaires ou non-linéaires.

5.4 Conclusion

Des contraintes de généralisation que nous avons exposées précédemment, c'est principalement la difficulté de traiter des textes de différente taille qui a guidé en premier lieu nos travaux.

Conclusion

Nous avons inscrit nos travaux au sein du vaste domaine de la recherche d'image par le contenu. Nous avons souligné qu'un des défis actuels est de «franchir» le fossé sémantique (i.e comment à partir de descripteurs de bas niveau remonter à l'information sémantique véhiculée par l'image). Une des approches pour résoudre partiellement ce problème consiste à se tourner vers des descripteurs dits de "haut niveau" (i.e. véhiculant *directement* des informations sémantiques de l'image). Nous avons montré en fin du Chapitre 2 que le texte présent dans une image, qu'il soit enfoui ou en sur-impression, correspondait à une telle définition. Dans les Chapitres 3 et 4, nous avons d'une part décrit la variabilité de l'objet «texte» et passé en revue les méthodes actuelles de localisation et de reconnaissance de celui-ci tant dans les images que les vidéos. Au sein de cette revue, nous avons pu apprécier les grandes avancées récentes dans ce domaine, notamment le développement de systèmes complets pour l'extraction de texte dans les vidéos. Ces avancées ont été réalisées pour la plupart sur des applications où des hypothèses relativement fortes sur les zones de texte à localiser étaient permises.

Nous allons quant à nous traiter plus particulièrement le cas de bases extrêmement hétérogènes, mélangeant texte en sur-impression, enfouis sur des fonds photographiques diversifiés. Ces bases étant représentatives aujourd'hui d'une vraie demande des utilisateurs (notamment les détenteurs de grands fonds photographiques).

Nous avions souligné, en conclusion de notre revue des méthodes existantes, deux grandes familles d'approches : les approches dites "textures" et les "approches composantes connexes". Nous n'avons pas inscrit le point de départ de nos travaux directement au sein de ces approches. En effet, dans le Chapitre 5, nous avons présenté un algorithme pré-existant, défini pour la localisation de texte dans les vidéos et développé en 2000 au sein de notre laboratoire. Notre démarche première a été de mettre en évidence ses forces et ses limitations. Une de ces faiblesses les plus criantes concerne l'absence d'une approche multi-résolution pour la prise en compte de texte de taille variable au sein d'une même image et/ou d'une base d'images. Notre idée de départ pour intégrer cette approche est d'utiliser des opérateurs résiduels numériques introduits récemment par Beucher [8]. La partie suivante sera consacrée à l'étude de ces opérateurs.

Opérateurs Résiduels Numériques

A la fin du chapitre précédent nous avons mis en lumière le point bloquant de l'extension de l'algorithme de [22], qui est de nécessiter d'avoir une connaissance préalable de la *taille* du texte que l'on souhaite extraire. Une manière classique pour résoudre ce problème est d'utiliser des approches multi-résolution. Ce sont souvent des approches globales, une série de filtrage d'activité croissante est appliquée à l'image. Il permet d'extraire différents niveaux de détails (résolutions) de l'image.

Nous proposons ici l'utilisation des opérateurs résiduels numériques (plus précisément des récentes avancées proposées par [8]) pour effectuer cette approche multi-résolution. Un de ces opérateurs, l'ouvert ultime, sera tout particulièrement étudié. Celui-ci permet de mettre en lumière les structures dominantes en terme de contraste de l'image : plus précisément il permet de connaître en chaque point de l'image la taille de la structure dominante (i.e. la taille d'ouverture pour laquelle elle disparaît) ainsi que son contraste.

Cette partie sera scindée en trois chapitres. Premièrement, nous introduirons le formalisme des opérateurs résiduels numériques en nous focalisant plus particulièrement sur l'opérateur d'ouverture ultime. Ensuite, nous étudierons le comportement de cet opérateur au travers d'images réelles, ceci nous permettra de mettre en lumière ses forces et ses limitations. Enfin, nous proposerons des solutions de calcul efficaces de celui-ci, pour transformer cet outil d'étude en brique de base algorithmique.

6 Présentation, définition et illustration des opérateurs résiduels numériques

On ne se compose pas plus une sagesse en introduisant dans sa pensée les divers résidus de toutes les philosophies humaines qu'on ne se ferait une santé en avalant tous les fonds de bouteille d'une vieille pharmacie.

Tas de pierres (1901)
VICTOR HUGO

Ce chapitre commencera par un rappel du formalisme associé aux opérateurs résiduels numériques. Ensuite nous nous intéresserons plus particulièrement à l'opérateur d'ouverture ultime. Enfin nous étendrons cet opérateur avec l'utilisation d'autres familles de transformations que celles présentées dans la littérature.

6.1 Transformations Résiduelles Numériques

Dans cette section, nous partirons des définitions des transformations résiduelles dans le cas de la morphologie binaire et nous verrons dans quel cadre et sous quelles contraintes nous pouvons les étendre au cas numérique.

6.1.1 Les transformées résiduelles ensemblistes

Une transformation résiduelle θ appliquée à un ensemble A est définie à partir de deux familles de transformations ordonnées appelées primitives dépendant d'un paramètre λ (ψ_λ et ξ_λ avec $\psi_\lambda \geq \xi_\lambda$). Le résidu de taille λ adjoint à cette transformation est l'ensemble $r_\lambda = \psi_\lambda \backslash \xi_\lambda$ (où $\backslash$ désigne la différence ensembliste) . Et la transformation elle-même est définie comme $\theta = \bigcup_{\forall \lambda \geq 0} (r_\lambda)$.

Pour la suite et pour mieux dissocier la définition de la transformation et les images associées à cette transformation, on notera :

Définition 12 *Résidu et Image Transformée associée à θ pour un ensemble A*

Soit $A \subset E$ un ensemble et ψ_λ,ξ_λ deux familles de primitives avec $\psi_\lambda \geq \xi_\lambda$ dépendant d'un paramètre λ (avec $\lambda \geq 0$). On notera

$$r_\lambda(A) = \psi_\lambda(A) \backslash \xi_\lambda(A), \forall \lambda \geq 0 \tag{6.1}$$

le résidu de niveau λ.

L'image Transformée associée à θ :

$$R_\theta(A) = \bigcup_{\forall \lambda \geq 0} (r_\lambda(A)) \tag{6.2}$$

A R_θ on associera une image *Indicatrice* notée q_θ.

Définition 13 *Image Indicatrice q_θ*

L'image indicatrice, associée à θ informe l'occurrence de l'activité de θ sur un point. Elle est donc restreinte au support de R_θ et est définie de la manière suivante :

$$q_\theta(a) = \lambda + 1 : a \in r_\lambda \tag{6.3}$$

Pour assurer l'unicité de cette image, on doit choisir les familles de primitives telles que $\forall \lambda, \mu \ \lambda \neq \mu : r_\lambda \cap r_\mu = \phi$.

Les définitions précédentes étant très générales, prenons l'exemple de l'érosion ultime d'un ensemble A et définissons la en terme de transformation résiduelle :

Dans ce cadre ψ_λ est l'érosion ε_λ et ξ_λ est l'ouverture élémentaire par reconstruction $\gamma^\infty \circ \varepsilon_\lambda$.

6.1.2 Extension aux images numériques

Comme l'indique Beucher dans [7, 8] l'extension directe de ces transformations aux images à teintes de gris n'est pas forcément triviale.

Nous allons reprendre les définitions précédentes, les transposer dans le cadre numérique et détailler les aménagements à réaliser pour l'utilisation de ces transformations.

La redéfinition du résidu ne pose pas de réel problème, on peut se contenter de remplacer la différence ensembliste par une soustraction numérique comme suit :

Définition 14 *Résidu dans le cadre numérique*

Soit I une image à teintes de gris et deux familles de primitives ψ_λ ξ_λ avec $\psi_\lambda \geq \xi_\lambda$ dépendant d'un paramètre λ (avec $\lambda \geq 0$). On notera

$$r_\lambda(I) = \psi_\lambda(I) - \xi_\lambda(I), \forall \lambda \geq 0 \tag{6.4}$$

l'image associée au résidu de niveau λ.

Malheureusement ce *résidu n'est pas forcément unique*. Dans le cas binaire, on a restreint le choix des primitives de manière à ce que les résidus soient disjoints deux à deux. Or cette contrainte est très difficile à obtenir dans le cas numérique. Il est ainsi possible d'avoir plusieurs résidus pour chacun des points du support de l'image sur laquelle on applique la transformation résiduelle.

En première approche on pourra redéfinir l'image *Transformée* R_θ en utilisant les équivalences numériques entre opérateurs sur les ensembles et sur les fonctions :

Définition 15 *Proposition d'une définition de R_θ dans le cadre numérique*
Soit I une image à teintes de gris et deux familles de primitives ψ_λ ξ_λ avec $\psi_\lambda \geq \xi_\lambda$ dépendant d'un paramètre λ (avec $\lambda \geq 0$). L'image Transformée associée à l'opérateur résiduel θ :

$$R_\theta(I) = \bigvee(r_\lambda(I)), \forall\lambda \geq 0 \tag{6.5}$$

Cette non unicité du résidu rend impossible une définition triviale de l'image q_θ dans le cadre numérique.

6.1.2.1 Définition d'une nouvelle image *Résiduelle* dans le cadre numérique

Dans [7, 8] une des solutions proposées consiste à définir cette image telle qu'en tout point x de q_θ corresponde la valeur de λ pour laquelle $r_\lambda(x)$ est positif et maximal.

Définition 16 *Nouvelle définition de q_θ dans le cadre numérique*
L'image Indicatrice associée à la transformation θ :

$$q_\theta(x) = \lambda + 1 : \lambda \geq 0, r_\lambda(x) > 0 \textit{ et maximum} \tag{6.6}$$

Ce maximum n'est pas forcément unique. En première approche on prendra la plus élevée des valeurs λ pour laquelle ce maximum apparaît.

$$q_\theta(x) = \bigvee \lambda + 1 : \lambda \geq 0, r_\lambda(x) > 0 \textit{ et maximum} \tag{6.7}$$

Comme l'indique Beucher d'autres stratégies existent et nous en discuterons dans le cadre de l'ouverture ultime.

6.2 Un cas particulier, l'ouverture ultime :

L'opérateur ouvert ultime est une transformation résiduelle qui utilise comme primitives(ψ_λ et ξ_λ) deux ouvertures d'une même famille. On rejoint ainsi un opérateur de type «granulométrique» piloté par le paramètre λ.

Définition 17 *Ouverture Ultime*
Soient deux familles de primitives ψ_λ, ξ_λ avec $\psi_\lambda \geq \xi_\lambda$ dépendant d'un paramètre λ (avec $\lambda \geq 0$). On définit l'opérateur d'ouverture ultime ν comme une transformation résiduelle θ telle que :

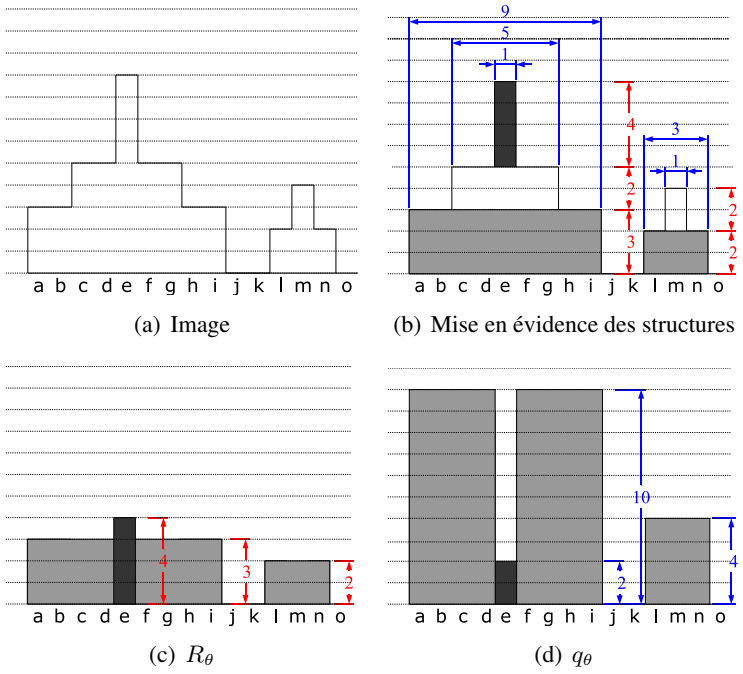
$$\begin{aligned} \psi_\lambda &= \gamma_\lambda \textit{ ouverture de taille } \lambda \\ \xi_\lambda &= \gamma_{\lambda+1} \textit{ ouverture de taille } \lambda + 1 \\ \theta &= \bigvee(\gamma_\lambda - \gamma_{\lambda+1}), \forall\lambda \geq 0 \end{aligned} \tag{6.8}$$

on pourra définir de façon similaire l'opérateur de fermeture ultime υ en utilisant ψ_λ et ξ_λ comme deux fermetures d'une même famille.
Les définitions des images R_θ et q_θ associées à ν découleront directement des définitions 15 et 16 (i.e. Équation 6.5,6.7).

Illustrons les définitions précédentes sur un profil de ligne (Figure 6.1(a)). Nous utiliserons γ_λ et $\gamma_{\lambda+1}$ comme des ouvertures par des éléments structurants plans dont la taille est paramétrée par λ. Les zones grisées de la Figure 6.1(b) illustrent les structures dominantes en terme de contraste de l'image.

Appliquons la définition 17 et construisons les images q_θ (Figure 6.1(d))) et R_θ (Figure 6.1(c)) en appliquant les ouvertures de taille croissante. Sur l'image (Figure 6.1(a)), on observe aux points $\{e, m\}$ des structures de taille 1 et de contrastes respectifs 4 et 2. Ces deux structures disparaîtront pour une ouverture de taille 2, on peut valuer $q_\theta\{e\} = 2$, $q_\theta\{m\} = 2$, $R_\theta\{e\} = 4$ et $R_\theta\{m\} = 2$. L'ouverture de taille 3 n'affecte aucune structure de l'image. L'ouverture de taille 4 par contre, va affecter les pixels $\{l, m, n\}$, les pixels $\{l, n\}$ n'ont pas encore été valués on affecte $q_\theta\{l, n\} = 4$ et $R_\theta\{l, n\} = 2$, le pixel $\{m\}$ a déjà une valuation, le résidu (saut courant en terme de contraste) est égal au précédent ($R_\theta\{m\} = 2$). Selon l'équation Eq.17 en cas d'égalité on conserve le dernier indice, d'où $q_\theta\{m\} = 4$ et $R_\theta\{m\} = 2$. Et on continue ainsi à appliquer des ouvertures de plus en plus grandes jusqu'à ce que l'image devienne uniforme.

Cet exemple nous permet de montrer l'intérêt de cet opérateur. Il met en lumière les structures dominantes en terme de contraste de l'image : plus précisément il permet de connaître en chaque point de l'image la taille de la structure dominante représentée par q_θ (i.e. la taille d'ouverture pour laquelle elle disparaît) ainsi que son contraste représenté par R_θ.



(a) Image (b) Mise en évidence des structures

(c) R_θ (d) q_θ

FIG. 6.1: Ouverture Ultime par des segments sur un profil de ligne

6.3 Extension de l'ouverture ultime à d'autres familles de critères :

6.3.1 Transformation morphologique par critère

6.3.2 Les ouvertures par critère

Nous ne traiterons que des critères rationnels et unidimensionnels (voir la définition 18), le lecteur pourra consulter [97] pour la présentation de critères «vectoriels». Selon les caractéristiques du critère utilisé, l'opérateur résultant sera une ouverture ou un amincissement. La définition et les propriétés des critères ainsi que les opérateurs morphologiques qui en découlent sont présentés dans les sous-sections suivantes.

6.3.2.1 Les ouvertures binaires par critère

Une des premières ouvertures par critère proposées dans la littérature est l'ouverture surfacique qui a été plus particulièrement étudiée par Vincent [103]. Le lecteur intéressé par les fondements théoriques de cet opérateur pourra se référer à [82] (p. 141–150).

Pour rappel l'ouverture surfacique binaire supprime toutes les composantes connexes d'une image binaire ayant une surface plus petite qu'un paramètre λ.

Dans [102], l'auteur montre l'équivalence entre l'ouverture surfacique et le supremum d'ouvertures avec la classe d'éléments structurants ayant une surface supérieure ou égale à λ. J.Breen and Jones [34] généralisent le concept en introduisant les opérateurs par critères et Walter [105] étend la propriété énoncée par *Vincent* à l'ensemble des opérateurs utilisant des critères croissants et planaires (voir Def.18).

Les critères Un critère assigne à un ensemble connexe $A \subset E$ une variable booléenne : une caractéristique (comme par exemple la surface, l'élongation maximale) de l'ensemble est extraite, et cette caractéristique est comparée avec une valeur de référence.

Définition 18 *Critère*
Soit $E \subset \mathbb{Z}^n$ et $\mathcal{C}(E)$ l'ensemble des sous-ensembles connexes de E. Un critère κ est une application de $\mathcal{C}(E)$ dans $\{0,1\}$ qui assigne à chaque ensemble connexe $A \in \mathcal{C}(E)$ un nombre $\kappa(A) \in \{0,1\}$. Si $\kappa(A) = 1$ on dit que A remplit le critère κ.

Un critère κ est dit croissant si pour deux ensembles connexes $A, B \in \mathcal{C}(E)$:

$$A \subset B \implies \kappa(A) \leq \kappa(B) \tag{6.9}$$

Les ouvertures et fermetures par critère Considérons maintenant des images binaires $X \in \mathcal{X}(E)$ et un critère croissant κ. Une ouverture par critère enlève toutes les composantes connexes de cette image binaire qui ne remplissent pas le critère κ.

Définition 19 *Ouverture par critère binaire Γ^κ*
Soit $X \subset E$ un ensemble et κ un critère croissant. Une ouverture Γ^κ par critère est l'union de toutes les composantes connexes de X qui remplissent le critère κ. Avec $\Gamma_x(X)$ la composante connexe de X contenant x si $x \in X$ et $\emptyset$ sinon, l'ouverture par critère s'écrit de la manière suivante :

$$\Gamma^\kappa(X) = \{x \in X \mid \Gamma_x(X) \textit{ remplit } \kappa\}, \textit{ avec } \kappa \textit{ croissant} \tag{6.10}$$

La fermeture par critère se définit par dualité (cf.A p.219) :

$$\Phi^{\kappa}(X) = [\Gamma^{\kappa}(X^c)]^c \tag{6.11}$$

Il a été montré [34] que l'opérateur défini dans la définition 19 est une ouverture, c'est-à-dire qu'il est idempotent, croissant et anti-extensif (cf. Annexe A).

Le théorème suivant, énoncé dans [105], montre le lien entre l'ouverture algébrique liée à un critère κ, et les ouvertures par adjonction (i.e. morphologiques).

Theorem 6.3.1 *Lien entre ouverture par critère et ouverture morphologique*
Soit $E \subset \mathbb{Z}^n$, $\mathcal{X}(E)$ l'ensemble des sous-ensembles de E et κ un critère croissant. L'ouverture par critère γ_κ est le supremum des ouvertures morphologiques avec la classe d'éléments structurants (B) qui remplissent le critère :

$$\Gamma^{\kappa} = \bigcup_{B \in E} \left\{ \gamma^B \mid \kappa(B) = 1 \right\} \tag{6.12}$$

La fermeture par critère peut s'écrire comme l'infimum de fermetures avec les éléments structurants qui remplissent le critère :

$$\Phi^{\kappa} = \bigcap_{B \in E} \left\{ \phi^B \mid \kappa(B) = 1 \right\} \tag{6.13}$$

Exemple de l'ouverture surfacique L'ouverture surfacique ne garde que les composantes connexes d'une image ayant une surface supérieure ou égale à λ. On peut l'écrire comme suit :

$$\Gamma_{\lambda}^{Surface}(X) = \{x \in X \mid Surface(\Gamma_x(X)) \geq \lambda\} \tag{6.14}$$

Mise en oeuvre La mise en oeuvre d'une ouverture par critère pour des images binaires est très simple. Elle se résume à une opération de labellisation de composantes connexes, suivie de la sélection des labels pour lesquels les composantes satisfont le critère.

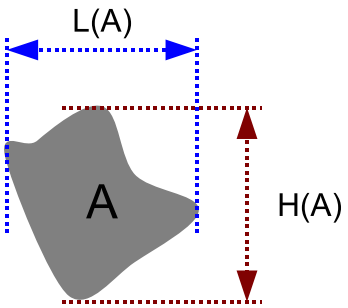
6.3.2.2 Les ouvertures/fermetures par critère pour des images à teintes de gris

Soit $I : D_I \subset \mathbb{Z}^2 \rightarrow T \subset \mathbb{Z}$ une image à teintes de gris.[1] Comme toute ouverture binaire, les opérateurs décrits dans 6.3.2.1 peuvent être appliqués également à I, en les appliquant aux ensembles dits sections I_t^+ (cf Annexe A) au niveau t et en empilant les résultats.

Caractéristiques des critères pour les images à teintes de gris Pour les images à teintes de gris, le critère κ peut également dépendre des valeurs de I, on parlera alors d'un *critère non-planaire* ; à l'inverse s'il ne dépend que d'une mesure μ sur les sections on parlera d'un *critère planaire*.

Ces considérations sont importantes au vu de la définition suivante :

[1] Pour la suite et sauf mention contraire nous restreindrons notre propos aux images à deux dimensions



- L'aire de l'ensemble connexe A.
- Le diamètre de Ferret vertical (qui se résume à la "Hauteur" de A)
  $H(A) = \bigvee_{p,q \in A} |p_y - q_y|$
- Le diamètre de Ferret horizontal (qui se résume à la "Largeur" de A)
  $L(A) = \bigvee_{p,q \in A} |p_x - q_x|$
- L'élongation maximale
  $\alpha(A) = \bigvee_{p,q \in A} |p - q|$
  avec $|\cdot|$ la distance discrète entre deux points $p = (p_x, p_y)$ et $q = (q_x, q_y)$
  tq : $|p - q| = |p_x - q_x| \vee |p_y - q_y|$
  L'élongation maximale peut être également définie comme $\alpha(A) = \bigvee(L(A), H(A))$

FIG. 6.2: Exemples de caractéristiques d'un ensemble connexe permettant la définition de critères croissants et planaires.

Définition 20 *Ouverture par critère sur une image à teintes de gris*
Soit $I : D_I \subset \mathbb{Z}^2 \to T \subset \mathbb{Z}$ une image à teintes de gris et κ un critère planaire et croissant. Une ouverture par critère s'écrit de la manière suivante :

$$[\Gamma^{\kappa}(I)](x) = \sup \left\{ t \leq I(x) \;\middle|\; \Gamma_x \left[I_t^+ \right] \text{ remplit } \kappa \right\} \tag{6.15}$$

et la fermeture par critère :

$$[\Phi^{\kappa}(I)](x) = \inf \left\{ t \geq I(x) \;\middle|\; \Gamma_x \left[I_t^- \right] \text{ remplit } \kappa \right\} \tag{6.16}$$

On peut toujours appliquer cette définition avec des critères non planaires ou non croissants mais le résultat ne sera pas une ouverture. Ainsi si le critère n'est pas croissant mais planaire, on obtient un opérateur idempotent et anti-extensif soit un amincissement (cf. [34]). Et si le critère n'est pas planaire, on ne peut pas garantir l'idempotence.

Exemples de caractéristiques des composantes connexes permettant la définition d'un critère croissant et planaire : La Figure 6.2 propose différents critères que nous utiliserons par la suite.

Mise en oeuvre : Une implantation naïve consisterait à appliquer la définition 20 mais ne serait pas efficace à moins de disposer d'une architecture dédiée. Des solutions algorithmiques dérivées de l'algorithme d'inondation par files d'attente [59] ont été proposées dans [34, 105]. D'autres implantations à base de «component-tree» ont été également proposées par Wilkinson and Roerdink [109]. Une discussion sur les performances respectives de ces implantations est présentée par Meijster and Wilkinson [57].

On rappellera ici brièvement le schéma algorithmique des fermetures par critère associées à un processus d'inondation à *niveau constant* ; pour le détail de cette implémentation on consultera [105] :

On part du ou des minima globaux de l'image à inonder. Pour ce premier niveau d'inondation s, on évalue la mesure associée au critère pour l'ensemble des lacs générés ; si un lac remplit le critère, l'image résultat prend la valeur s pour tous les points appartenant à ce lac (on dit que le lac est «gelé»). Puis, on incrémente s, les lacs existants sont étendus, de nouveaux minima locaux ayant la valeur s sont ajoutés, jusqu'à ce que tous les lacs pour ce nouveau niveau s soient déterminés. Si deux lacs se rencontrent, ils fusionnent, et la mesure associée au critère est évaluée pour le lac fusionné, comme s'il s'agissait d'un seul lac. Puis on continue le processus d'inondation avec évaluation de la mesure du critère à chaque changement de niveau s jusqu'à ce qu'il n'y ait plus de lacs qui ne remplissent pas le critère.

Pour la suite, et bien qu'il s'agisse d'un abus de langage, on appellera «valeur du critère κ», la mesure μ associée à un critère κ donné.

6.3.3 Ouverture ultime utilisant des familles d'ouvertures par critère

Nous avons introduit en début de cette note les critères et nous avons montré que pour tout critère croissant et planaire on peut définir une ouverture algébrique à partir de la définition 20. Nous rappelons ici qu'à un critère κ on associe une mesure μ, et que c'est la comparaison de cette mesure par rapport à un seuil donné λ qui indique si le critère est validé ou non pour une composante connexe donnée (cf. Ouverture Surfacique Eq. 6.14).

Nous pouvons maintenant introduire l'opérateur d'ouverture ultime par critère.

Ouverture ultime par critère : La définition (17) indique que l'on doit choisir ψ_λ, ξ_λ comme deux ouvertures d'une même famille avec $\psi_\lambda \geq \xi_\lambda$. Il suffit de choisir une famille d'ouvertures par critère et de faire évoluer le paramètre λ associé au critère.

Définition 21 *Ouverture ultime par critère*

Soient deux familles de primitives ψ_λ, ξ_λ avec $\psi_\lambda \geq \xi_\lambda$ dépendant d'un paramètre λ (avec $\lambda \geq 0$). On définit l'opérateur d'ouverture ultime par critère ν^κ comme une transformation résiduelle θ telle que :

$$\begin{aligned} \psi_\lambda &= \Gamma^\kappa \text{ ouverture par critère } \kappa \text{ associée à une mesure } \mu \text{ tq } \mu \geq \lambda \\ \xi_\lambda &= \Gamma^{\tilde{\kappa}} \text{ ouverture par critère } \tilde{\kappa} \text{ associée à une même mesure } \mu \text{ tq } \mu \geq \lambda + 1 \\ \theta &= \bigvee(\Gamma^\kappa - \Gamma^{\tilde{\kappa}}), \forall \lambda \geq 0 \end{aligned} \tag{6.17}$$

La croissance de la mesure μ n'est pas forcément unitaire mais on pourra se rapporter à ce cas par anamorphose[a].

On pourra définir de façon similaire l'opérateur de fermeture ultime par critère υ^κ. Les définitions des images R_θ et q_θ associées à ν^κ sont triviales.

[a] En faisant fi du formalisme : Une anamorphose transforme une valeur d'origine en une autre valeur d'arriver. Chaque valeur d'origine se voit attribuer une nouvelle valeur F(valeur entrée), où F est une fonction de transfert linéaire ou non, continue ou discontinue. On indique ici que l'on pourra associer à la croissance de la mesure μ, une nouvelle mesure $f(\mu)$ qui croîtra de manière unitaire

Dans le chapitre suivant nous allons illustrer l'action de l'opérateur sur des images non synthétiques.

7 Illustration de l'action de l'opérateur sur des images réelles

Idéal : modèle qu'on se compose, en vue de l'admirer et de l'imiter.
L'idéal est toujours nettoyé d'un peu de réalité qui ferait tache.

EMILE-AUGUSTE CHARTIER

Nous avons montré précédemment l'action l'opérateur d'ouverture ultime au travers d'un exemple sur une image synthétique. Nous allons voir son effet sur des images réelles. Nous regarderons quelles sont les familles de transformations les mieux adaptées à notre cadre applicatif.

Nous découperons cette étude en trois grandes étapes. Tout d'abord nous illustrons l'action de l'opérateur appliqué directement sur chaque polarité de l'image (structures claires et sombres), nous nommerons cette utilisation Approche Directe*. Cette étude nous permettra de mettre en lumière certaines myopies de l'opérateur. La description de celles-ci ainsi que des propositions pour y pallier nous donnera une meilleure connaissance du comportement de l'opérateur dans certains cas particuliers. Cette connaissance est indispensable pour une bonne compréhension (puis utilisation) de cet outil.*

Ensuite nous nous intéresserons à l'utilisation de l'opérateur sur le gradient de l'image. Nous confronterons les forces et faiblesses de cette Approche Gradient *avec la précédente.*

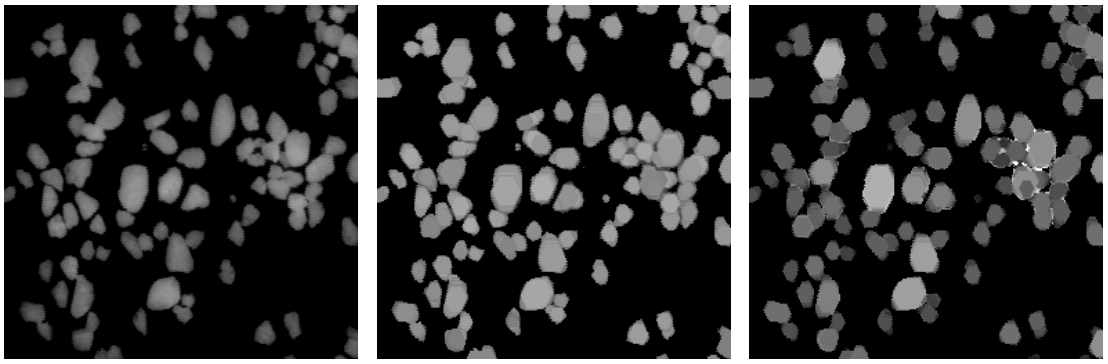
7.1 Utilisation de l'ouverture/fermeture ultime : approche directe

7.1.1 Rappel des travaux existants

Dans [8], Beucher utilise comme famille de primitives des ouvertures morphologiques par des *boules* de tailles croissantes. Ce travail a trouvé un intérêt applicatif fort dans le travail de doctorat de OUTAL [67]. Dans les deux cas une hypothèse géométrique quant aux structures d'intérêt est tacite : leur représentation sous forme de boules (ou hexagones équivalents) est suffisante pour les caractériser. Cette approche permet notamment de généraliser la notion de *boules critiques* dans un cadre numérique.

Comme l'indique explicitement Beucher, l'image initiale est remplacée par l'union des cylindres les plus significatifs[1] inclus dans le sous graphe de celle-ci (cette information est contenue dans R_θ). L'image associée q_θ quant à elle contient pour chaque site/pixel x de l'image initiale le rayon du plus grand cylindre significatif de la reconstruction partielle recouvrant x.

Ceci est illustré sur la Figure 7.1. L'image est grossièrement biphasée mais permet de montrer que l'opérateur se comporte dans le cadre numérique d'une manière proche de ce que l'on attend dans le cas binaire.



(a) Image Originale (b) R_θ, Correction $\gamma = 3$ (c) q_θ après normalisation

FIG. 7.1: Illustration de l'action de l'ouvert ultime, sur une image grossièrement biphasée. La famille de transformations utilisée est composée d'ouvertures morphologiques par des boules (i.e. hexagone) de tailles croissantes.

7.1.2 Utilisation de ces travaux pour la mise en lumière des caractéristiques des lettres

L'approche précédente peut être complétée en utilisant la notion de connexité. Si l'on se place dans un cadre binaire une ouverture morphologique peut scinder une composante connexe en plusieurs composantes connexes alors que les ouvertures par reconstruction la préservent ou l'éliminent intégralement.

On peut donc utiliser également des familles d'ouvertures par reconstruction pour définir un opérateur d'ouverture ultime dans le cas numérique.

Cependant que ce soit pour les approches avec ou sans reconstruction, il existe une dépendance *forte* entre l'information véhiculée par les images R_θ et q_θ et le choix de l'élément structurant. Finalement on ne caractérise pas vraiment la *taille* des structures mais une taille en relation avec la *forme* de l'élément structurant utilisé. Pour les mêmes raisons qu'avait souligné Vachier [98], dans le cadre des valeurs d'extinctions numériques, si l'on veut extraire des informations en taille et seulement en taille des structures d'images, il faudrait en toute rigueur considérer toutes les configurations possibles d'éléments structurants (ce qui est inenvisageable, techniquement parlant).

Nous allons voir au travers d'exemples que l'utilisation de familles d'ouvertures morphologiques n'est pas adaptée à notre cadre applicatif.

En première approximation, on pourrait considérer que les lettres appartenant à une même famille de polices (et à taille constante), partagent certaines caractéristiques simples comme :

[1] Un cylindre est significatif s'il est le plus *grand* et le plus haut cylindre pouvant être inclus dans la fonction initiale

1. Une *épaisseur* de trait commune
2. Une information d'élongation, liée à la taille des traits (permettant de les classer en sous-familles (e.g :minuscules, majuscules)).

On pourrait être tenté d'utiliser des familles d'ouvertures par reconstruction associées à un élément structurant circulaire pour caractériser l'épaisseur et le supremum d'ouvertures par reconstruction par des segments (dans un grand nombre de directions) pour caractériser la longueur des traits.

Cependant, si l'on regarde les lettres de la Figure 7.2, on observe clairement que même dans un cadre binaire, la réalité n'est pas si simple. On ne peut résumer l'épaisseur du trait d'un caractère à la plus grande boule maximale-inscrite dans celui-ci. D'autres exemples pourraient venir compléter cette figure montrant que l'utilisation de segments (comme élément structurant) est également peu robuste vis à vis de la variabilité des lettres rencontrées au sein d'une même phrase.

Nous reviendrons d'ailleurs sur la complexité de définir des facteurs de forme robuste des lettres dans la suite du manuscrit.

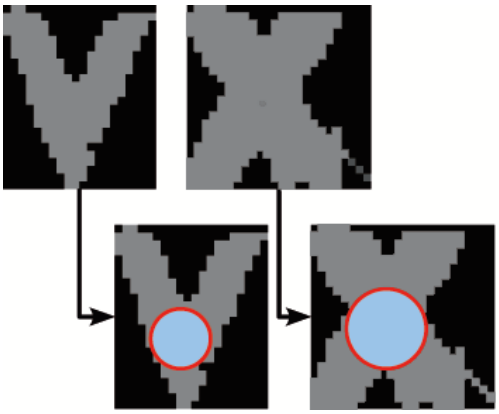


FIG. 7.2: L'utilisation d'ouvertures ultimes basées sur des familles d'ouvertures morphologiques (ici utilisation d'ouvertures par des boules de tailles croissantes) est mal adaptée à la caractérisation des lettres.

Les outils précédemment énoncés ne peuvent donner de mesures non ambiguës de caractérisation des lettres. Leur extension au cadre numérique est encore plus sujette à caution.

7.1.3 Conclusion

Le rappel de travaux déjà effectués à l'aide de cet opérateur nous a permis de montrer que l'approche initiale n'est pas forcément la plus adéquate dans le cadre de notre problématique. Par la suite nous introduisons l'utilisation d'autres familles de transformations. Celles-ci sont des approches dites par *critères*.

7.1.4 Utilisation de l'ouverture/fermeture ultime par critère

Pour discuter du choix du critère retenu pour la mise en lumière des lettres nous allons illustrer l'effet de l'opérateur sur des images réelles[2].

Un premier exemple de l'effet de l'opérateur pour différents critères est présenté sur la Figure 7.3.

Ce premier résultat nous permet d'illustrer deux points importants :

1. On peut montrer l'intérêt de cet opérateur pour la caractérisation des structures : si l'on observe les images q_θ(Figure 7.3(g), Figure 7.3(j), Figure 7.3(m), Figure 7.3(d)) on remarque

[2]Par la suite nous changerons d'images pour illustrer différents comportements ; en effet ceux-ci seront dépendants de certaines caractéristiques propres aux images étudiées. Les images étudiées sont issues de la base d'évaluation I.C.D.A.R [53]

(a) Image Originale (Base d'image [53])

(b) Canal de Luminance

(c) R_θ, Correction $\gamma = 3$

(d) q_θ

(e) q_θ labellisée

Ouverture ultime avec critère surfacique.

(f) R_θ, Correction $\gamma = 3$

(g) q_θ

(h) q_θ labellisée

Ouverture ultime avec critère d'élongation.

(i) R_θ, Correction $\gamma = 3$

(j) q_θ

(k) q_θ labellisée

Ouverture ultime avec critère de hauteur.

(l) R_θ, Correction $\gamma = 3$

(m) q_θ

(n) q_θ labellisée

Ouverture ultime avec critère de largeur.

FIG. 7.3: Présentation sur une image réelle de l'utilisation de l'ouverture ultime pour les critères surfacique, d'élongation, de hauteur et de largeur.

que chaque lettre est représentée par une composante connexe valuée respectivement par son élongation, sa hauteur, sa largeur et sa surface. Ceci peut permettre de premiers traitements efficaces de sélection des lettres sur des contraintes géométriques et propose de plus une pré-segmentation des lettres.

2. On observe dès à présent que dans le cas du critère surfacique, R_θ possède de très faibles valuations.

Ici nous avons volontairement sélectionné un cas «d'école» où l'opérateur se comporte exactement de la manière escomptée. En effet les lettres sont de quasi zones plates avec un contraste apparent (vis à vis du panneau sur lequel elles reposent) important. Nous verrons dans les sections suivantes les problèmes liés à l'utilisation de cet opérateur.

7.1.5 Discussions sur le choix du critère pour la détection des lettres

7.1.5.1 L'utilisation du critère surfacique est à proscrire

Ce titre pourrait sembler un peu abrupt, cependant dans le cadre de notre application (que nous pouvons sûrement étendre à d'autres), les cas où une ouverture ultime surfacique donne des résultats exploitables restent des exemples académiques.

Pour s'en convaincre, il faut comprendre pourquoi les valeurs de gris de l'image transformée sont si faibles sur la Figure 7.3(c). Lorsque nous réalisons une ouverture ultime, la série d'ouvertures surfaciques avec un pas **unitaire** évolue de manière progressive, donnant lieu à des résidus faibles entre deux étapes successives. Ce ne serait pas le cas si l'image d'entrée était composée d'objets aux contours parfaitement définis, cas idéal, qui ne correspond pas généralement aux propriétés des images réelles (eg : contours flous, zones de transition, bruit).

La Figure 7.4 illustre clairement les limitations de l'utilisation du critère surfacique. Malgré le fort contraste apparent des lettres sur l'image originale, l'image transformée (R_θ) a des valuations très faible ; en particulier les lettres sont valuées pour des valeurs de R_θ<7.

Remarque 6 *Dans le chapitre 8 traitant de l'implémentation efficace de l'ouvert ultime par un processus d'inondation, nous verrons que cette constatation est encore plus évidente.*

7.1.5.2 Critère retenu par la suite

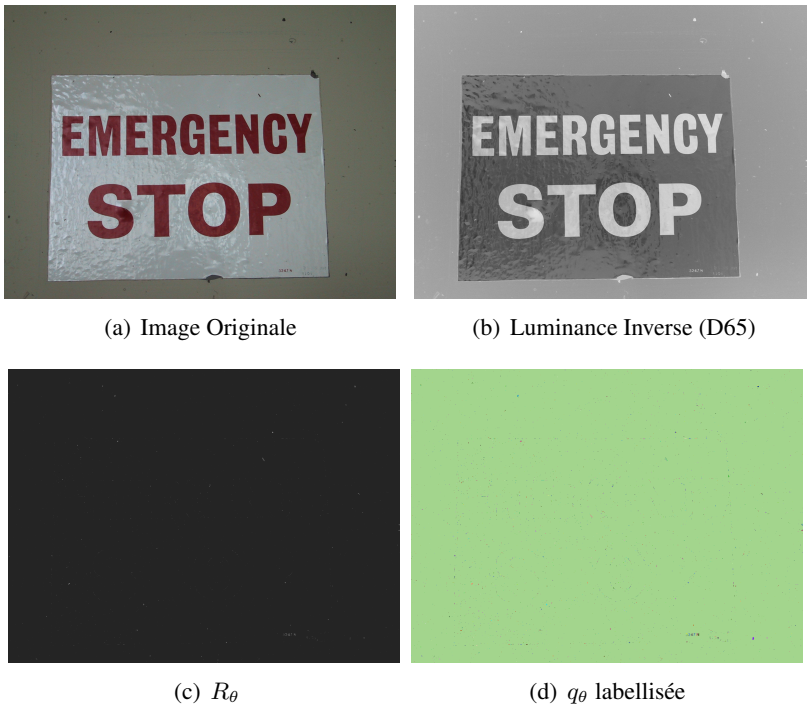
Les différents critères que nous avons présentés Hauteur, Largeur, Élongation sont certes dissemblables mais ils fourniront généralement des pré-segmentations des lettres *équivalentes*.

Bien sûr la valuation de q_θ est conditionnée au choix du critère mais nous sommes avant tout intéressés par **la segmentation obtenue**.

Dans nos applications, nous utiliserons en majorité le critère de **hauteur** n'ayant pas trouver de contre exemple flagrant nous poussant a en changer.

7.1.5.3 Conclusion

Nous venons d'exposer les raisons qui nous ont poussés à utiliser des familles d'ouvertures par critères (par opposition aux familles d'ouvertures morphologiques) pour l'utilisation de l'opérateur d'ouvert ultime dans notre cadre applicatif. Nous verrons par la suite que ce choix s'avérera pertinent dans les chaînes de traitement que nous proposerons.



(a) Image Originale (b) Luminance Inverse (D65)

(c) R_θ (d) q_θ labellisée

FIG. 7.4: L'utilisation du critère surfacique est à proscrire. Dans cet exemple R_θ et q_θ sont constantes (voire texte). Base d'image [53].

On notera toutefois que les champs de recherche en termes de critères et de transformations résiduelles restent à défricher. Ce travail permettra peut-être de proposer des familles de transformations plus pertinentes dans l'avenir.

Nous venons de discuter de l'intérêt de l'opérateur utilisant des familles d'ouvertures/fermetures par critères. Nous allons à présent mettre en lumière différentes myopies dont il souffre et proposerons différentes stratégies pour y pallier. La connaissance de ces myopies est indispensable pour une bonne compréhension (puis utilisation) de cet outil.

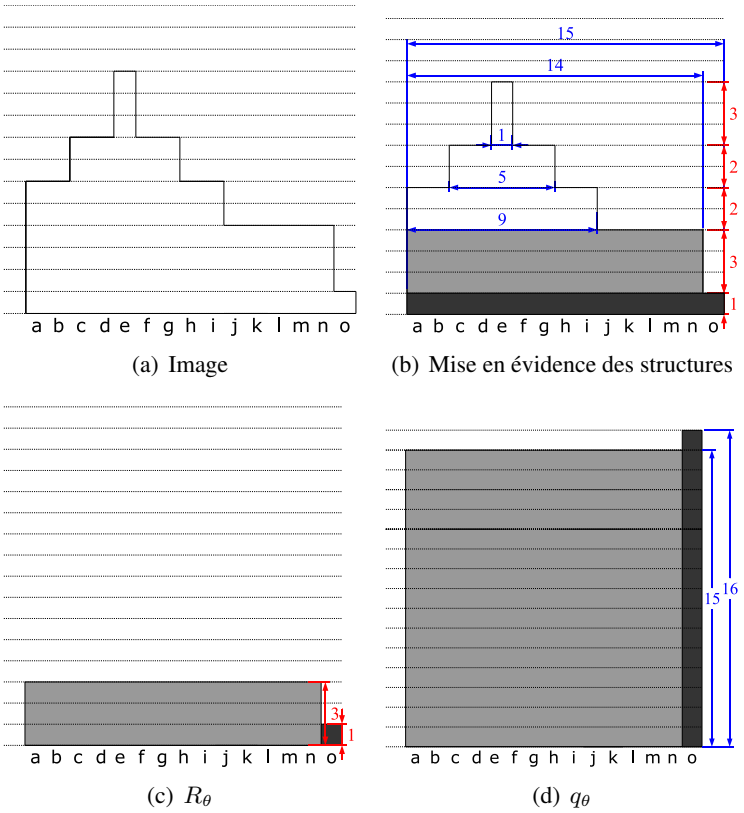
7.2 Myopie de l'opérateur pour des structures imbriquées

L'opérateur d'ouverture ultime dans sa première définition (cf. def 16) est **un opérateur «sans mémoire»**, il n'enregistre que le dernier résidu maximal rencontré. Ainsi l'utilisateur de cet opérateur devra d'abord se définir une échelle d'analyse dans laquelle l'action de cet opérateur aura un sens.

On pourra consulter pour exemple le profil de ligne présent sur la Figure 7.5(a), que l'on pourrait décomposer en deux structures imbriquées : une première structure *pyramidale* (points $\{a, \ldots, i\}$) reposant sur *un socle* (points $\{a, \ldots, n\}$). Sans sélection, seule la structure 2 (le socle) en gris sur la Figure 7.5(b) contribuera au résultat final en Figure 7.5(c) et Figure 7.5(d).

7.2.1 Stratégies pour résoudre cette myopie

1. On définit une valeur de taille d'ouverture au-delà de laquelle le processus est arrêté.



(a) Image (b) Mise en évidence des structures

(c) R_θ (d) q_θ

FIG. 7.5: Myopie de l'ouvert ultime pour des structures imbriquées

2. Une autre solution consisterait non pas à prendre le dernier résidu maximal rencontré mais les N sous-résidus maximaux rencontrés (ce type d'approche rejoint les travaux de Leite and Guimaraes [42]). On obtiendrait alors une «hiérarchisation» des résidus. Cette hiérarchie peut être difficile à analyser et la définition d'un N optimal reste inaccessible dans le cas général. Une discussion sur les hiérarchies est proposée en Section 7.2.1.2.

7.2.1.1 Illustration de cette myopie sur une image naturelle :

Pour exposer cette myopie dans un cadre réel, nous allons reprendre l'exemple de la détection de texte enfoui dans une image. Sur la Figure 7.6(a) on observe différentes chaînes de caractères disposées sur un panneau. Nous allons utiliser l'opérateur de fermeture ultime par hauteur pour mettre en lumière les lettres.

On procède tout d'abord à une fermeture ultime par hauteur sans valeur d'arrêt (i.e. nous appliquons des fermetures de tailles croissantes jusqu'a atteindre la taille de l'image) : les résultats sont présentés au travers des images 7.6(c), 7.6(d), 7.6(e). Le résultat est plus que décevant, les images q_θ et R_θ ne contiennent aucune information pertinente. L'explication en est simple, lors de l'application de l'opérateur on ferme les vallées de l'image et on calcule les résidus associés. Puisque l'on applique l'opérateur pour une valeur d'arrêt égale à la taille maximum admissible par l'image (ici λ_{max}=Hauteur de l'image), un dernier résidu ι_{last} est engendré pour l'ensemble de l'image lorsque l'on atteint le/les maxim(um/a) globaux de l'image. Si ce dernier résidu est supérieur à la valuation de

R_θ en tout point de l'image on obtient une image $R_\theta = \iota_{last}$ et $q_\theta = \lambda_{max}$. Certes ce(s) maxim(um/a) glob(al/aux), serr(a/ont) souvent des pics de bruit, et un faible filtrage pourrait les supprimer. Mais, ceci n'est pas évidemment une généralité. De plus il faudra être attentif à ce que le filtrage mis en place ne détériore pas des structures d'intérêt de l'image.

On procède ensuite à une fermeture ultime par hauteur avec une valeur d'arrêt égale à la moitié de la hauteur de l'image : les résultats sont présentés au travers des images 7.6(f), 7.6(g) et 7.6(h). Là encore le résultat est décevant, on observe que seule la lettre "d" est visible ! Pourquoi ? Le problème est simple : **les lettres sont imbriquées dans le panneau** et lorsque le panneau disparaît pour une taille de fermeture donnée, le résidu engendré est supérieur aux résidus respectifs des lettres à l'intérieur du panneau.

Pour détecter les lettres à l'aide de cet opérateur, on doit prendre comme valeur d'arrêt une taille inférieure à celle de la hauteur du panneau. Ceci est présenté sur les Figures 7.6(i), 7.6(j) et 7.6(k).

7.2.1.2 Discussion sur les hiérarchies de résidus

Nous avions signalé en début de section que l'utilisation des hiérarchies de résidus serait peut-être une piste intéressante pour la prise en compte de la myopie des opérateurs résiduels en présence de structures imbriquées.

Cependant on doit être attentif aux résultats proposés par ce type d'approche. Pour illustrer ceux-ci nous allons nous aider de la Figure 7.7 qui présente les deux premiers niveaux d'une hiérarchie de résidus utilisant des ouvertures ultimes surfaciques. Pour cet exemple nous considérerons que les deux structures qui nous intéressent se trouvent en $\{e\}$ et $\{h\}$. En étudiant les deux niveaux nous pouvons isoler deux problèmes :

1. **Niveaux de hiérarchies différents :** les deux structures d'intérêt apparaissent chacune dans un niveau de la hiérarchie. De plus *le socle* sur lequel elles reposent apparaît dans le premier niveau. Ceci implique que chaque structure doit être identifiée séparément, on se prive ainsi d'un possible couplage spatial qui pourrait aider leur identification.
2. **Structure *fantôme* :** Bien que la structure d'intérêt en $\{e\}$ apparaisse au premier niveau de hiérarchie, son *fantôme* apparaît dans le second niveau avec un contraste (valuation de R_θ) supérieur à celui de la structure en $\{h\}$ et **une valuation de q_θ correspondant à la taille du socle**. Cet artefact pourrait être simplement décelé dans ce cas en comparant la taille du support (ici 1) avec la valuation de q_θ (ici 12) ; **cependant il s'agit d'un exemple didactique mono-dimensionnel, la prise en compte d'images et de critères de dimensions supérieures pourra rendre complexe la recherche d'un tel fantôme**.

Conclusion sur les hiérarchies : Comme nous venons de le voir, l'utilisation des hiérarchies peut entraîner des résultats *singuliers*. Aussi leur utilisation devra être couplé dans un premier temps avec une connaissance a-priori des structures recherchées.

Chacun des niveaux pourrait être nettoyé d'un sous-ensemble de structures parasites (ou fantôme), en vérifiant que le support des composantes connexes et leur valuation sont concordantes.

Cependant cette approche n'est pas forcément adaptée aux problèmes de la recherche de texte : en effet pour agréger et valider une zone de texte nous devons disposer d'au moins trois composantes éligibles (ie composantes «proches» aux caractéristiques géométriques compatibles)[3]. Or rien n'indique que les lettres d'une même zone de texte se retrouvent au même niveau d'une hiérarchie.

[3]Hypothèse majoritairement admise, il faut un minimum de trois lettres pour former un mot,cf. Section 3.2.1 22

(a) Image Originale (Base d'image [53])

(b) Canal de Luminance

(c) R_θ

(d) q_θ

(e) q_θ labellisée

Fermeture ultime par hauteur, sans arrêt de l'algorithme.

(f) R_θ

(g) q_θ

(h) q_θ labellisée

Arrêt de l'algorithme avec λ égale à la moitié de la hauteur l'image.

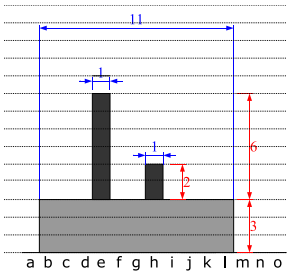
(i) R_θ

(j) q_θ

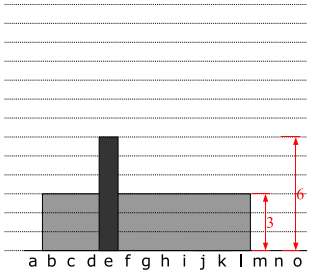
(k) q_θ labellisée

Arrêt de l'algorithme avec λ égale au quart de la hauteur de l'image.

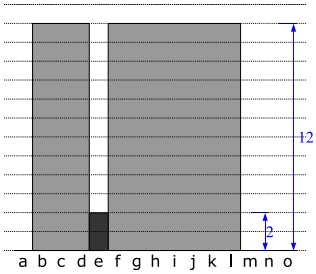
FIG. 7.6: Présentation sur une image réelle des conséquences de la myopie de l'opérateur pour les structures imbriquées



(a) Image

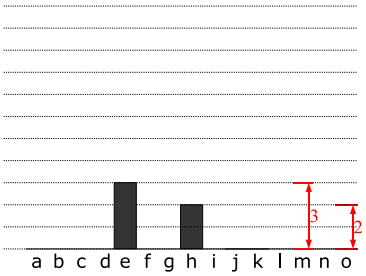


(b) R_θ

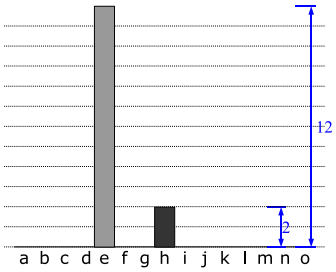


(c) q_θ

Premier niveau de hiérarchie



(d) R_θ



(e) q_θ

Second niveau de hiérarchie

FIG. 7.7: Illustration des deux premiers niveaux d'une hiérarchie de résidus utilisant des ouvertures ultimes surfaciques.

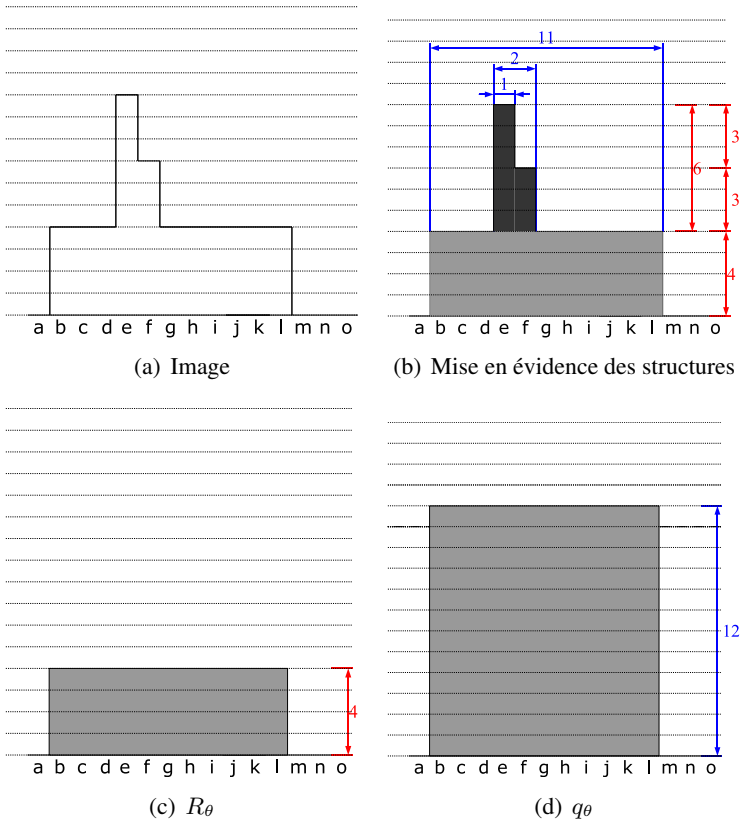
7.2.2 Conclusion sur les structures imbriquées :

Nous n'avons que partiellement résolu la myopie de l'opérateur pour les structures imbriquées. En effet, dans ce cas, pour que l'opérateur apporte des résultats satisfaisants il faut une connaissance a priori des structures de l'image, ce qui réduit (de manière importante) la pertinence de cet outil pour des applications généralistes. Ainsi on pourra s'intéresser à une approche hiérarchique pour ce type de myopie (voir Section 7.2.1.2). Bien sûr le choix des niveaux de hiérarchies à prendre en compte restera dans tous les cas un problème ouvert. Nous verrons par la suite que nous retrouverons cette myopie dans l'approche gradient.

7.3 Myopie de l'opérateur pour des zones dites de «transitions graduelles»

7.3.1 Zones dites de «transitions graduelles»

L'utilisation de la définition 16 pose également problème en cas de transition «graduelle». En effet si on regarde la Figure 7.8(b), on observe une «zone de transition graduelle» (au point $\{e, f\}$) reposant sur une large composante plus faiblement contrastée. Si l'on suit la définition 16, cette transition est «invisible» à l'opérateur. En effet la transition représentée par les pixels $\{l, m\}$ est de contraste 4 alors que la transition de $\{e\}$ à $\{g\}$ se fait en deux sauts de contraste de 3. Si nous considérions que les pixels $\{e, f, g\}$ sont une seule et même transition, elle aurait un contraste de $3 + 3 = 6$ et elle ne serait pas invisible à l'opérateur.



(a) Image (b) Mise en évidence des structures

(c) R_θ (d) q_θ

FIG. 7.8: Myopie de l'ouvert ultime pour des transitions graduelles

7.3.2 Stratégies pour parer à cette myopie

1. On pourrait non pas incrémenter la taille des ouvertures de un en un mais définir un *pas d'ouverture*. On devra bien évidemment définir ce pas en fonction de l'application.
2. On pourrait procéder à un redressement des frontières des régions avant l'application de l'opérateur. Il faudra bien évidemment veiller à ce que ce redressement n'altère pas les structures d'intérêt et qu'il ne forme pas de nouvelles structures «artificielles» dans l'image.

3. Une autre solution consisterait à suivre en tout point x du domaine de définition de l'image l'activité de l'opérateur ν : en effet on peut par exemple associer pour tout point x un tableau $\mathcal{M}$ qui conserve pour chaque valeur λ 1 si un résidu non nul a été engendré, 0 sinon (ceci a été proposé initialement par Leite and Guimaraes [42]). On dira par exemple que l'opérateur est actif en un point $\{e\}$ pour une taille λ si $\mathcal{M}_\lambda(e) = 1$. Ainsi si l'on reprend l'image 7.8(a) et que l'on regarde l'activité de l'opérateur ν aux pixels $\{e\}$ et $\{f\}$, on peut définir :

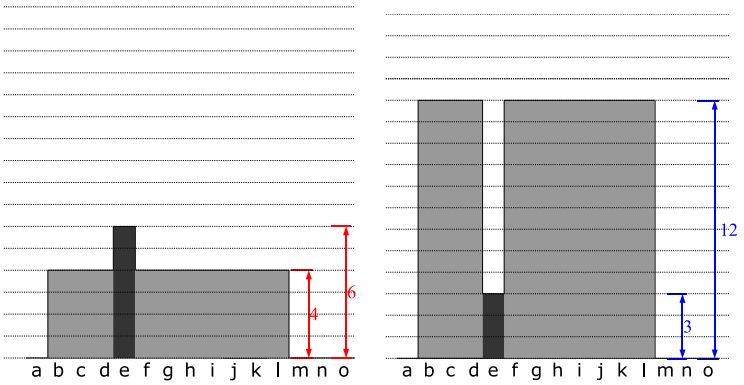
$$\mathcal{M}_\lambda(e) = \{0, 1, 1, 0, 0, 0, ...\} \text{ pour } \lambda = 0, 1, 2, 3, 4, 5, ...$$
$$\mathcal{M}_\lambda(f) = \{0, 0, 1, 0, 0, 0, ...\} \text{ pour } \lambda = 0, 1, 2, 3, 4, 5, ...$$

On observe que pour tout point x, on peut détecter la/les valeur(s) de λ pour laquelle ν arrête «temporairement» son activité sur x. Il suffit de regarder le passage de l'état 1 à l'état 0 de $\mathcal{M}_\lambda(x)$.

On peut proposer d'accumuler les résidus en un point x tant que l'activité ν en x est non nulle, c'est à dire pour toute série adjacente de 1, et de prendre en compte ces résidus accumulés pour le calcul de R_θ et q_θ.[4]

Ainsi si l'on reprend l'image 7.8(a) et que l'on regarde l'activité de l'opérateur ν au pixel $\{e\}$: l'opérateur devient actif sur e pour $\lambda = 1$, un résidu de 3 apparaît, ν est toujours actif pour $\lambda = 2$, un nouveau résidu de 3 apparaît, il est accumulé avec le précédent, pour $\lambda = 3$, ν arrête «temporairement» son activité ; on utilise le résidu accumulé en $\{e\}$ de 6 pour valuer $R_\theta\{e\}$ et $q_\theta\{e\}$.

Le résultat de cette stratégie pour l'image complète en utilisant un pas d'ouverture de 1 est présenté sur la Figure 7.9.



(a) R_θ (b) q_θ

FIG. 7.9: Correction partielle de la myopie de l'ouvert ultime pour des transitions graduelles

7.3.2.1 Illustration de l'effet de l'accumulation pour une image réelle :

Pour exposer l'intérêt des accumulations dans un cadre réel, nous allons reprendre l'application de détection de texte enfoui dans une image. Sur la Figure 7.10 nous proposons de comparer les résultats

[4] On peut coupler cette approche avec la première en faisant croître λ avec un *pas non unitaire*

obtenus pour des fermetures ultimes utilisant le critère d'élongation[5] avec et sans prise en compte des accumulations.

Nous pouvons observer que sans prise en compte de l'accumulation, seulement un sous-ensemble des lettres est détecté (cf Figure 7.10(e)) bien qu'elles semblent être d'égal contraste sur l'image de luminance.

L'utilisation des accumulations (cf Figure 7.10(h)) permet de détecter l'ensemble des lettres. Pour comprendre en détail son rôle nous allons suivre les résidus engendrés en deux points de l'image : nous placerons ceux-ci respectivement, au milieu des lettres "P" et "R" de la chaîne de caractères "PRIVATE". Les résultats sont fournis sur la Figure 7.11.

Si l'on regarde de plus près les résidus engendrés sur les Figures 7.11(c),7.11(e) on remarque que la valeur maximale de résidu engendrée aux points "P" et "R" pour des valeurs de λ compris entre 0 et 100 est inférieure, respectivement, supérieure, au dernier résidu engendré (cf Figures 7.11(d),7.11(f)). Ceci explique pourquoi la lettre "R" est correctement détectée (la valuation de q_θ aux points du support de la lettre est cohérente avec la taille «géométrique» de celle-ci) à l'inverse de la lettre "P" (pour laquelle la valuation de q_θ correspond à la taille du panneau). Ce comportement est contre- intuitif car les lettres semblent avoir le même contraste sur l'image de luminance.

Or en "R" et "P", non pas un, mais plusieurs résidus sont générés pour des valeurs de λ croissantes, on a donc pour chacune d'elles une transition graduelle. Si l'on prend l'accumulation des résidus pour chacune des lettres le long de cette transition (courbe grisée sur les Figures 7.11(c),7.11(e)) , l'accumulation est pour chacune d'elles (≈ 150) bien supérieure au dernier résidu engendré (≈ 50) (cf Figures 7.11(d),7.11(f)).

D'une certaine manière l'accumulation permet de lever l'ambiguïté de l'opérateur en présence de transitions graduelles de l'image.

7.3.2.2 Réflexion autour de l'accumulation :

Nous avons présenté dans le paragraphe précédent une solution au problème des transitions graduelles. Nous allons mettre en lumière les défauts de cette approche et en proposer une variante qui, à notre sens, *est plus naturelle*.

Reprenons le résultat de la Figure 7.9 issu de l'application de la stratégie précédente sur le profil de ligne en Figure 7.8(b). La transition composée des points $\{e, f, g\}$ a été prise en compte **mais seul** e **en a «profité»**. Deux problèmes apparaissent ici : le point f bien qu'ayant participé à l'accumulation n'apparaît pas dans le résultat et, plus gênant encore, $q_\theta\{e\}$ est *mal valué*. En effet on a sur l'image une structure de taille 1 (au point e), le résidu est engendré pour une fermeture de taille 3 (à cause de l'accumulation). La corrélation entre taille de la structure et valuation de q_θ n'est plus assurée.

Une variante de l'approche précédente serait de **propager** l'accumulation sur l'ensemble de la zone de transition. Le résultat de cette propagation est illustré sur la Figure 7.12 (à comparer avec le résultat de la Figure 7.9).

Algorithmiquement parlant, cela demande de connaître à tout moment les points faisant partie d'une/des zones de transition. Ceci est difficilement réalisable avec les algorithmes que nous proposerons dans la suite de ce chapitre. Dans les perspectives nous discuterons d'outils nous permettant d'envisager la mise en place de cette stratégie.

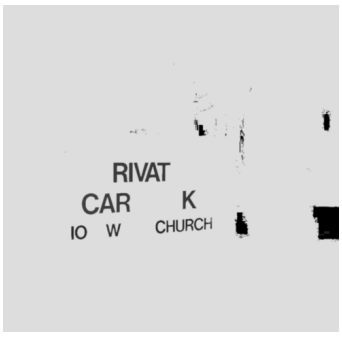
Attention le problème des structures imbriquées n'est pas pour autant résolu : La présentation de la prise en compte des transitions graduelles ne doit pas nous faire perdre de vue la myopie de

[5] L'expérience pourrait être ré-éditée pour les autres critères mais n'apporterait pas de résultat complémentaire.

(a) Image Originale Base d'image [53]

(b) Canal de Luminance

(c) R_θ



(d) q_θ

(e) q_θ labellisée

Fermeture ultime avec critère d'élongation sans arrêt de l'algorithme.

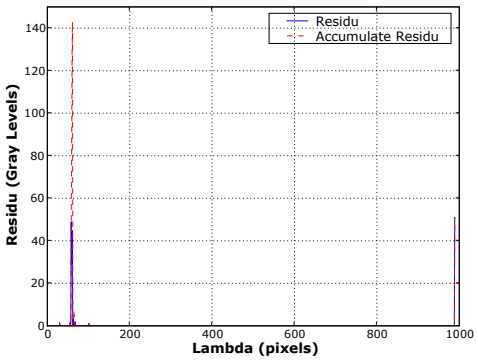
(f) R_θ

(g) q_θ

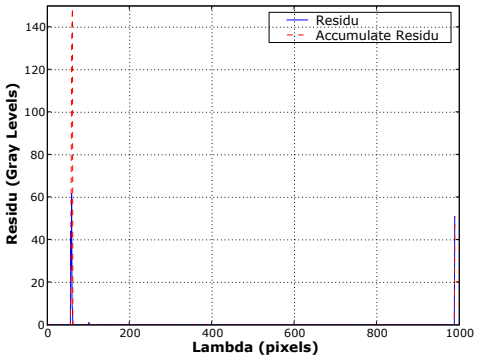
(h) q_θ labellisée

Fermeture ultime avec critère d'élongation sans arrêt de l'algorithme avec accumulation des résidus pour des transitions de tailles unitaires.

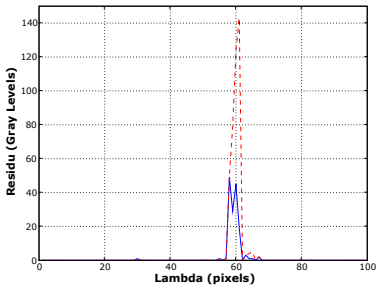
FIG. 7.10: Comparaison pour une image réelle de l'utilisation de la Fermeture Ultime pour le critère d'élongation avec et sans utilisation de l'accumulation. Les transitions prises en compte sont de tailles unitaires.



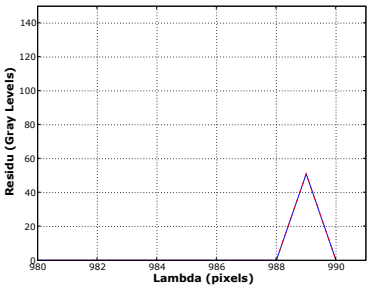
(a) Suivi des résidus dans la lettre P



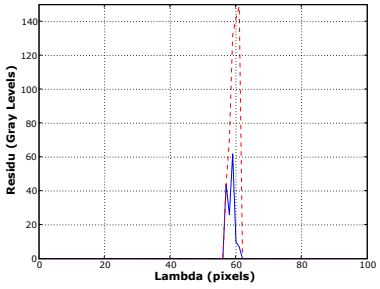
(b) Suivi des résidus dans la lettre R



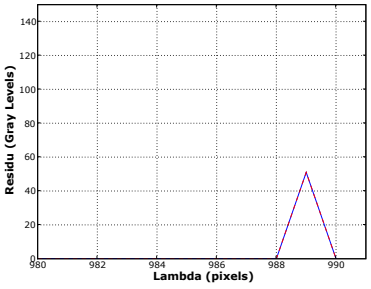
(c) Suivi des résidus dans la lettre P, zoom sur la première accumulation



(d) Suivi des résidus dans la lettre P, zoom sur la dernière accumulation

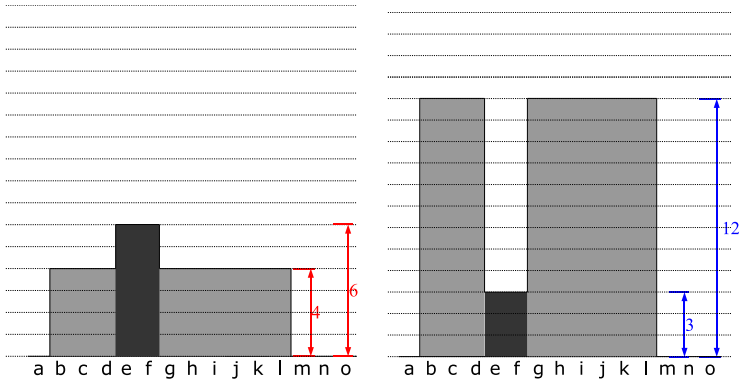


(e) Suivi des résidus dans la lettre R, zoom sur la première accumulation



(f) Suivi des résidus dans la lettre R, zoom sur la dernière accumulation

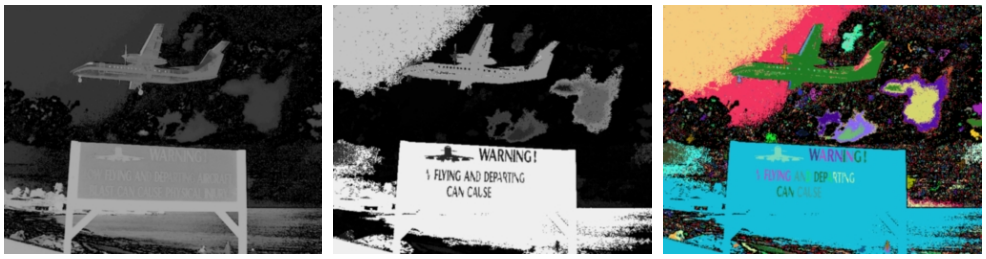
FIG. 7.11: Suivi des résidus dans les lettres "P" et "R" .



(a) R_θ (b) q_θ

FIG. 7.12: Seconde stratégie de correction de la myopie de l'ouvert ultime pour des transitions graduelles : on propage l'accumulation à l'ensemble de la zone de transition

l'opérateur face aux structures imbriquées. Pour ce faire nous proposons de reprendre l'exemple de la Figure 7.6(a), et d'arrêter l'algorithme pour un critère d'arrêt égal à la moitié de la hauteur de l'image. Le résultat est présenté en Figure 7.13, même avec accumulation, les résidus engendrés lors de la disparition du panneau (et d'une partie du fond de l'image) sont supérieurs pour un sous-ensemble aux résidus engendrés par les lettres lors de leur disparition dans le panneau.



(a) R_θ. Correction $\gamma = 3$ (b) q_θ (c) q_θ labellisée

FIG. 7.13: L'accumulation ne résout pas pour autant les problèmes de structures imbriquées

Conclusion sur les transitions graduelles : Nous venons de voir l'intérêt de la prise en compte des accumulations sur un cas réel. Les points suivants nous semblent importants à retenir :

1. La version avec accumulation apporte de nouvelles informations au travers de l'image R_θ. En effet l'accumulation des résidus le long des transitions progressives permet d'avoir une information plus pertinente sur le contraste «réel» d'une composante vis à vis du fond l'environnant.
2. Nous avons choisi ici de prendre en compte des transitions de taille unitaire mais cela ne sera pas toujours le choix le plus judicieux. Malheureusement comme pour la détermination de la valeur d'arrêt en présence de structures imbriquées, le choix de la taille de transition à prendre en compte dans un cadre générale reste un problème ouvert.

3. La deuxième stratégie proposée, basée sur la propagation de l'accumulation à l'ensemble de la zone de transition est à notre sens plus intéressante car elle véhiculera des informations à la fois de contraste au travers de R_θ mais également une information plus *intuitive /correcte* de la géométrie des structures au travers de q_θ. Nous en reparlerons dans les perspectives.

Dans les sections précédentes nous avons passé en revue les informations nécessaires à la connaissance de l'opérateur. La mise en lumière de ces myopies nous permet maintenant de mieux appréhender son comportement pour certaines configurations.

En particulier, la nécessité de définir une *échelle* de traitement (par l'arrêt de l'opérateur) est indispensable dans des applications réelles[6]. Dans la section suivante nous présentons quelques résultats de segmentation.

7.4 Quelques résultats satisfaisants de l'approche directe

Malgré les faiblesses soulevées précédemment, notons que cette approche (non paramétrique) fournie des résultats intéressants dans des situations variées (i.e. si l'on ne se trouve pas dans un cas pathologique identifié précédemment). La Figure 7.14 présente des résultats probants de cette approche. Notons qu'aucune sélection d'échelle n'a été effectuée (l'opérateur est arrêté pour une taille d'ouverture de la hauteur de l'image).

7.5 Conclusion de l'approche directe

Nous venons de proposer une analyse des comportements et résultats de l'opérateur pour une *approche directe*. Ceci nous a permis de mettre en lumière ces myopies et de proposer des solutions pour y pallier.

Nous allons nous intéresser par la suite à l'application de l'opérateur au travers de l'*approche gradient*. Nous retrouverons nombre des comportement préalablement évoqués auxquels nous adjoindrons quelques nouvelles configurations intéressantes. Ceci nous permettra ensuite une brève revue comparative des deux approches.

7.6 Utilisation de l'opérateur sur des images «réelles» : approche gradient

Une des raisons de s'intéresser à une approche gradient est que celle-ci permettrait de s'affranchir d'une connaissance a priori sur la polarité des structures d'intérêt.

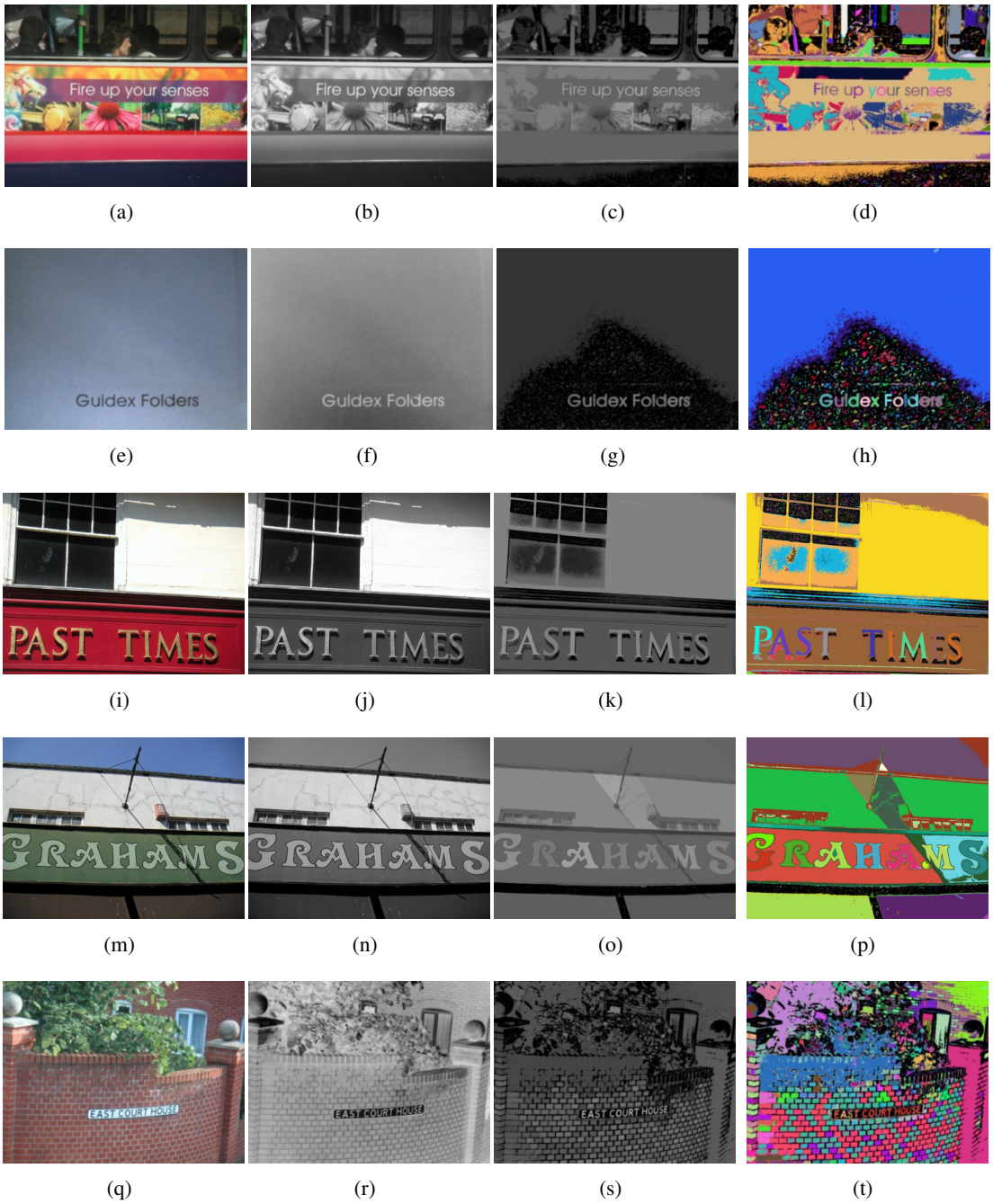
Remarque 7 *La couleur n'est pas toujours une information disponible, l'étude se voudra donc indépendante de cette information. Celle-ci pourra bien sur être utilisée pour certaines bases d'images.*

Nous commencerons notre analyse comme précédemment en regardant l'information véhiculée par l'opérateur sur des cas simples. Nous rappellerons seulement que nous arrêterons toujours l'algorithme avant le calcul du dernier résidu (cf. Section 7.2.1.1).

Nous verrons que les myopies présentées dans les sections précédentes sont toujours présentes.

Sur la Figure 7.15, on observe que dans un *cas simple* on obtient, comme pour l'approche directe, une pré-segmentation des lettres. Chacune d'elles étant valuées (dans q_θ) par une valeur de gris associée au critère sélectionné (Élongation, Hauteur, Largeur).

[6] L' autre stratégie utilisant des hiérarchies n'ayant pas été poursuivie, nous n'en ferons plus mention par la suite



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

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

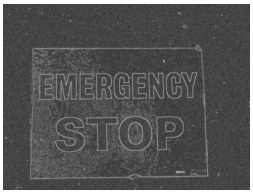
(i) (j) (k) (l)

(m) (n) (o) (p)

(q) (r) (s) (t)

FIG. 7.14: Approche Directe :Utilisation de l'opérateur de fermeture ultime par hauteur. Présentation de segmentation *satisfaisante* en l'absence de tout autre traitement. De gauche a droite : image originale, canal de luminance pour une polarité du texte, R_θ (Correction $\gamma = 3$), q_θ labellisée. Base ICDAR [53]

(a) Image Originale



(b) Gradient sur la luminance (D65). Correction $\gamma = 3$

(c) R_θ, Correction $\gamma = 3$

(d) q_θ

(e) q_θ labellisée

Fermeture ultime avec critère de d'élongation.

(f) R_θ, Correction $\gamma = 3$

(g) q_θ

(h) q_θ labellisée

Fermeture ultime avec critère de Hauteur.

(i) R_θ, Correction $\gamma = 3$

(j) q_θ

(k) q_θ labellisée

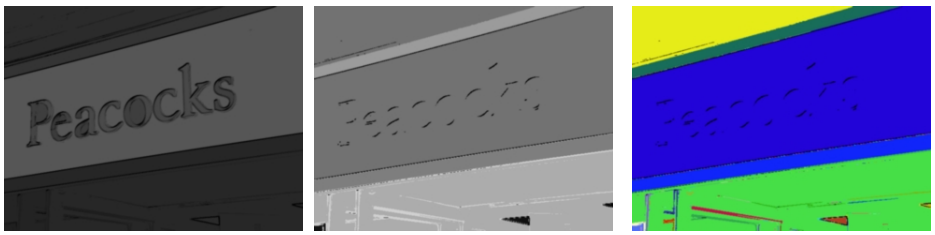
Fermeture ultime avec critère de Largeur.

FIG. 7.15: Opérateur de fermeture ultime pour différents critères. Application sur le gradient de luminance .

7.6.1 Comportement de l'approche gradient en présence de structures imbriquées

L'approche gradient souffre de la même myopie que l'approche directe. Si l'on regarde l'exemple de la Figure 7.16, on observe que les lettres sont incrustées dans un panneau. Le gradient entourant les lettres, pourtant de fort valuation, est inférieur à celui du panneau. Le résidu engendré lors de la fermeture du panneau est supérieur à celui de la fermeture des lettres et la segmentation obtenue est inexploitable.

(a) Image Originale (b) Gradient Morphologique sur la luminance (D65)



(c) R_θ (d) q_θ (e) q_θ labellisée

FIG. 7.16: Le problème des structures imbriquées est bien sûr toujours présent pour l'approche gradient. (critère utilisé : hauteur)

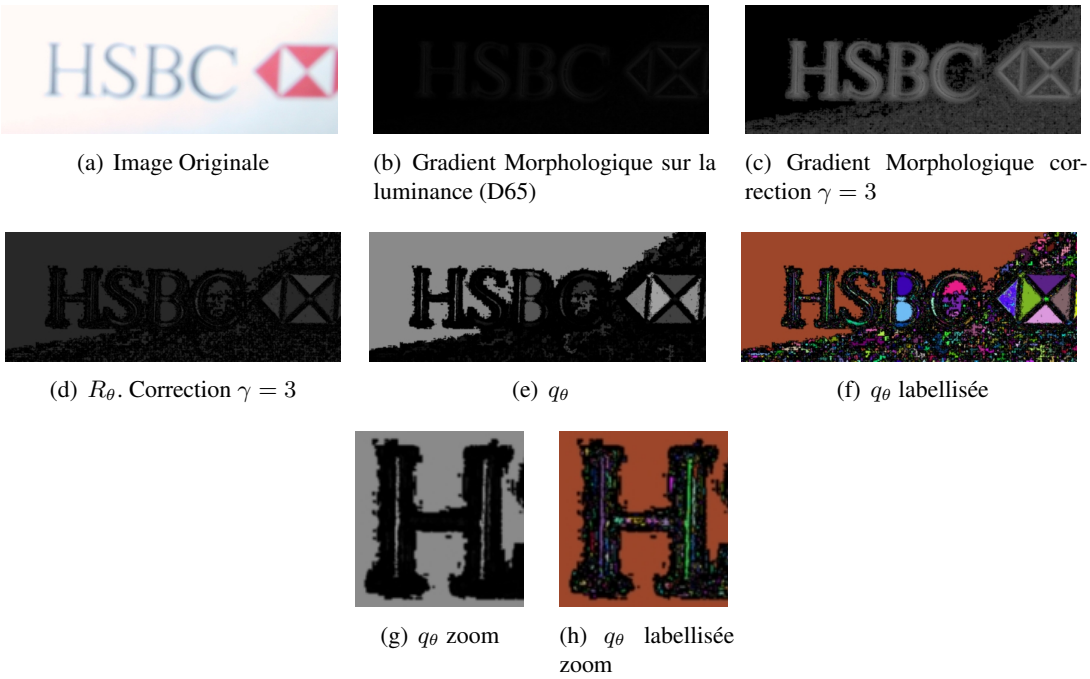
Les stratégies pour parer cette myopie sont les mêmes que l'approche directe (cf. Section 7.2.1). Nous rappelons également qu'une discussion sur les hiérarchies de résidus est présentée en Section 7.2.1.2.

7.6.2 Comportement de l'approche gradient en présence de frontières floues

Nous avions en Section 7.3 présenté la myopie de l'opérateur (dans le cas direct) en présence de transition graduelle. Pour le cas du gradient, le problème se pose différemment. On observe sur la Figure 7.17 le comportement de l'approche gradient en présence de *frontière floue*, le résultat est inexploitable. Ce qui est remis en cause ici n'est pas l'opérateur mais le choix du gradient.

En présence de frontière floue, une première solution serait d'utiliser des gradients morphologiques épais (cf. [75]). Le problème dans notre cas est que l'on se retrouve encore une fois devant *un paradoxe de l'oeuf et de la poule*, on ne sait pas avant d'avoir étudié l'image si l'on est dans ce cas.

Dans le cas de l'approche directe on peut utiliser la stratégie d'accumulation pour pallier partiellement cette myopie. Le résultat d'une telle approche est proposé en Figure 7.18. Le résultat est loin d'être parfait mais l'on obtient des marqueurs cohérents pour chacune des lettres.



(a) Image Originale

(b) Gradient Morphologique sur la luminance (D65)

(c) Gradient Morphologique correction $\gamma = 3$

(d) R_θ. Correction $\gamma = 3$

(e) q_θ

(f) q_θ labellisée

(g) q_θ zoom

(h) q_θ labellisée zoom

FIG. 7.17: En présence transition(i.e. frontière floue), l'application d'un gradient morphologique de taille unitaire suivi de l'utilisation de l'opérateur de fermeture ultime (ici utilisant un critère de hauteur) n'est pas appropriée.

7.6.3 Approche gradient et échelle de traitement

Un dernier point que nous souhaiterions souligner est la difficulté de généralisation de l'approche gradient pour des bases d'images comportement des zones de textes composées de *petites* lettres comme présenté sur la Figure 7.19. On pourrait dans ce cas utiliser l'approche directe ou d'autres types de gradient. Certes, mais là encore, il faudrait une connaissance a priori préalable des tailles des lettres rencontrées, ce qui n'est pas garanti comme nous le verrons dans le Chapitre 10, p.185.

7.7 Quelques résultats satisfaisants de l'approche gradient

Comme pour l'Approche Directe (cf Section 7.4 p.79), l'Approche Gradient peut être appliquée sans sélection d'échelle sur un sous-ensemble de cas et permet de se passer (partiellement) d'a priori sur la polarité du texte. La Figure 7.20 présente des résultats probants de cette approche.

7.8 Conclusion de l'approche gradient

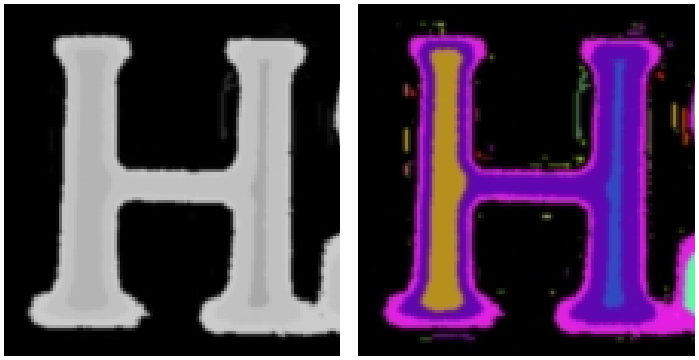
Dans les sections précédentes, nous avons passé en revue le comportement de l'opérateur au sein de l'*Approche Gradient*. Nous avons montré que cette approche souffre des mêmes myopies que l'*Approche Directe*. Notons que les problèmes rencontrés sont également imputables au type de gradient utilisé, i.e. nous n'avons testé l'opérateur que sur un *gradient de luminance*. L'utilisation de gradients plus *robustes* et/ou de pré-filtrage, ainsi que la possibilité d'intégrer partiellement des infor-

(a) Image Originale (b) Canal de luminance (D65)

(c) R_θ correction $\gamma = 3$ (d) q_θ (e) q_θ labellisée

Approche directe sans accumulation (Critère Hauteur).

(f) R_θ correction $\gamma = 3$ (g) q_θ (h) q_θ labellisée

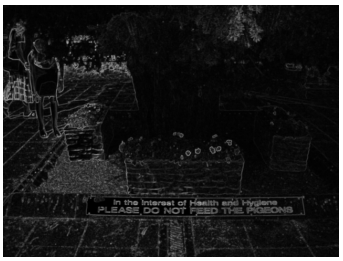


(i) q_θ zoom (j) q_θ labellisée zoom

Approche directe avec accumulation (Critère Hauteur).

FIG. 7.18: Dans le cas de transition graduelle. L'approche directe, utilisant la stratégie d'accumulation, est plus exploitable.

(a) Image Originale



(b) Gradient Canal de luminance (D65)

(c) Zoom sur une portion du gradient

(d) Zoom sur la portion de q_θ labellisée obtenue

FIG. 7.19: L'approche *Gradient* appliquée brutalement n'est pas adaptée aux *petites lettres* (Critère utilisé :Hauteur).

mations colorimétrique ou de texture par l'utilisation de gradient dédié seraient des pistes intéressantes à explorer.

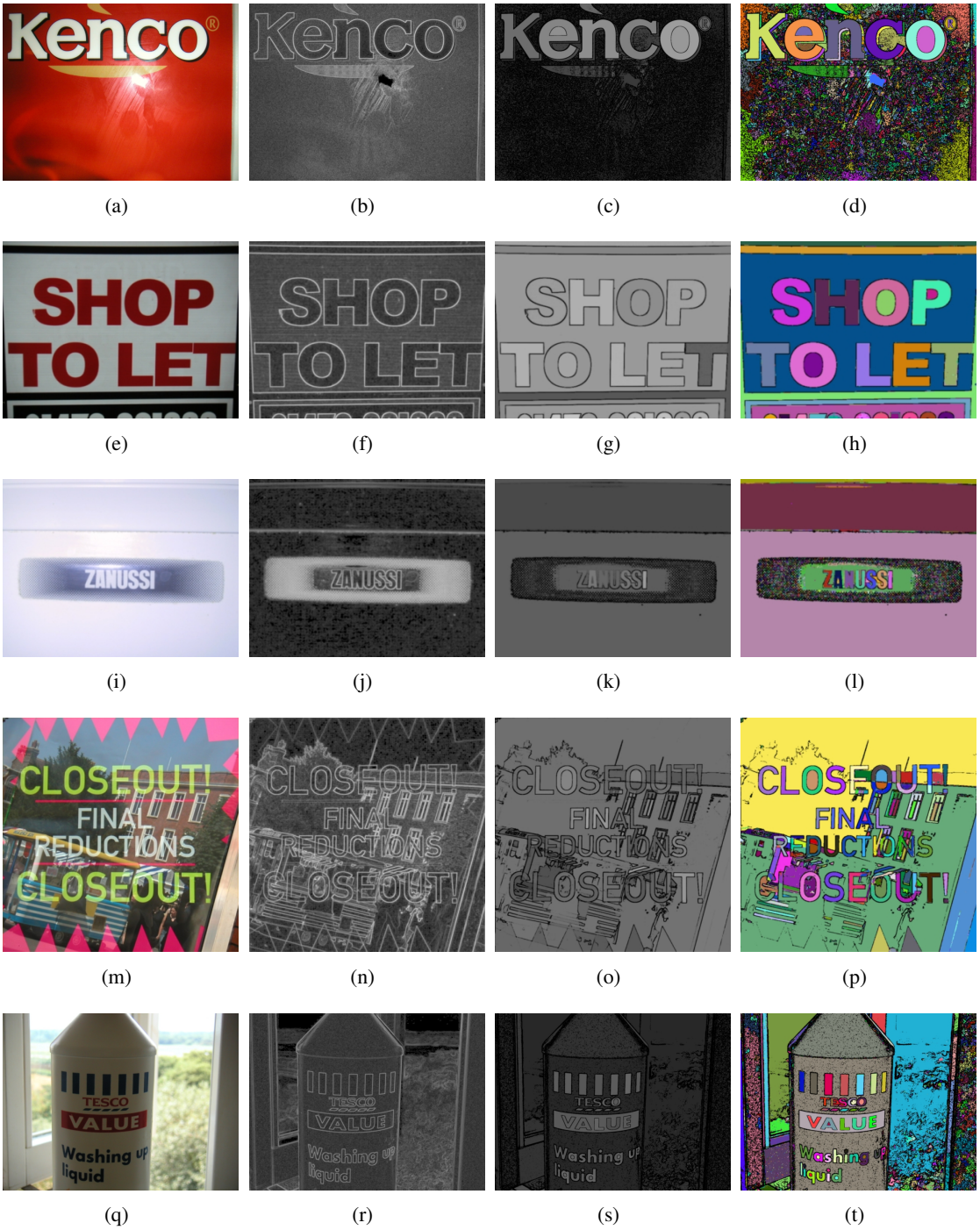
	APPROCHES	
	DIRECTE	GRADIENT
PETITES STRUCTURES	Adapté	Non
POLARITÉ	Dépendance	Indépendance
COULEUR	Indisponible	Gradient dédié
MYOPIE STRUCTURES IMBRIQUÉES	OUI	OUI
SOLUTIONS ?	Stop, Hiérarchies	idem
MYOPIE TRANSITIONS GRADUELLES	OUI	idem
SOLUTIONS ?	Accumulation	Indisponible

TAB. 7.1: Récapitulatif :forces et faiblesses des approches directe et gradient

7.9 Conclusion

Dans les sections précédentes, nous avons proposé une analyse des informations véhiculées par l'opérateur d'ouverture ultime sur des images réelles. Nous avons mis en lumière certaines de ces myopies et proposé différentes stratégies pour y pallier. Nous avons montré que, pour notre cadre applicatif, l'utilisation de familles d'ouvertures par critère semble mieux adaptée. Nous avons discuté des avantages et inconvénients des approches *Directe* et *Gradient*. La plupart des remarques importantes que nous avons formulées sont résumées dans le Tableau 7.1.

Le passage d'un outil d'étude, pour lequel les temps de traitement ne sont pas une préoccupation importante, à une brique algorithmique, nous impose maintenant de proposer des solutions de calcul efficaces.



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

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

(i) (j) (k) (l)

(m) (n) (o) (p)

(q) (r) (s) (t)

FIG. 7.20: Approche Gradient :utilisation de l'opérateur de fermeture ultime par hauteur. Présentation de segmentation *satisfaisante* en l'absence de tout autre traitement . De gauche à droite : image originale, gradient morphologique sur le canal luminance (Correction $\gamma = 3$), R_θ (Correction $\gamma = 3$), q_θ labellisée .Base ICDAR [53]

Dans le chapitre suivant nous allons proposer une implémentation *générique* et efficace de cet opérateur basée sur un processus d'inondation. Une variante de celle-ci prenant en compte la première stratégie (voir p. 73) pour résoudre la myopie de l'opérateur pour les transitions graduelles de l'image sera également décrite.

8 Implantation efficace de l'opérateur d'Ouverture Ultime

Premature optimization is the root of all evil.

D. KNUTH

On the other hand, we cannot ignore efficiency.

JON BENTLEY

Nous avons présenté dans les chapitres précédents, l'opérateur d'ouverture ultime, ses intérêts et ses myopies au travers d'exemples didactiques et d'images réelles. Cet opérateur comme bien d'autres en morphologie mathématique a été défini comme une combinaison de transformations plus élémentaires. Or cette combinaison est généralement coûteuse. Elle devient même prohibitive si l'on souhaite utiliser l'opérateur sous forme d'une routine. Si l'on omet une résolution de ce problème par la définition d'architectures dédiées on se doit de proposer des solutions de calcul efficaces.

Ce chapitre traitera des solutions algorithmiques que nous avons développées. Nous rappellerons tout d'abord l'implantation basique de l'opérateur qui sera utilisable quelle que soit la famille d'ouvertures envisagée. Puis nous proposerons un algorithme efficace dérivé de l'algorithme de l'inondation pour un sous-ensemble de critères. Une variante de celui-ci sera proposée pour résoudre en partie certaines myopies de l'opérateur. Enfin nous proposerons d'autres pistes de réflexion algorithmique pour de futurs développements.

8.1 Approche basique :

Nous rappelons avec l'algorithme 8.1 p.90 l'implémentation basique de l'ouverture ultime proposée par Beucher (Beucher [7]) qui est valable pour n'importe quelle famille d'ouvertures.

Sans parler de complexité au sens strict, on observe que le facteur limitant est bien évidemment le calcul d'une ouverture de taille λ (ligne.12) à chaque passe de l'algorithme. Ce coût devient vite prohibitif avec l'augmentation de la taille des éléments structurants pour les ouvertures morphologiques.

Algorithme 8.1 : Implantation générique de l'ouverture ultime

Data : $imRelief$ Image à traiter, $step$ (pas des tailles d'ouverture ; default=1), $stop$ (taille d'ouverture maximale)
Result : R_θ (image *Transformée*),q_θ (image *Indicatrice*)

```
1  begin
2      • Initialisation :
3      R_θ ⟵ 0
4      q_θ ⟵ 0
5      imOpenCurrent ⟵ 0 //Ouverture courante
6      imResCurrent ⟵ 0 //Image des résidus courants
7      imOpenPrec ⟵ imRelief //Ouverture précédente
8      λ ⟵ 0 //Taille de l'ouverture courante
9      • Boucle Principale :
10     while λ<stop do
11         λ=λ+step
           //Calcul du résidu courant
           //Remplacer γ par la famille d'ouverture choisie
12         imOpenCurrent ⟵ γ(imRelief,λ)
13         imResCurrent ⟵ imOpenPrec − imOpenCurrent
           //Si, au pixel x, un résidu non nul et supérieur ou égal au résidu
             précédent est généré R_θ(x) et q_θ(x) doivent être mis à jour (cf.
             définition 17)
14         foreach pixel x do
15             if imResCurrent(x)>=R_θ(x) then
16                 if imResCurrent(x)>0 then
17                     R_θ(x) ⟵ imResCurrent(x)
18                     q_θ(x) ⟵ λ
19         imOpenPrec ⟵ imOpenCurrent
20 end
```

Dans le cas des ouvertures /fermetures par critère, même si le coût de calcul d'une ouverture /fermeture de taille λ peut être ramené à une valeur proche d'une constante (Meijster and Wilkinson [57]) pour toute valeur de λ, cette constante n'est pas négligeable.

Comme nous l'avions souligné dans le chapitre précédent, c'est tout particulièrement la définition d'un algorithme efficace pour l'opérateur d'ouverture ultime utilisant des familles d'ouvertures par critères dont nous avons besoin.

8.2 Réflexion pour la conception d'un algorithme efficace pour l' ouverture/fermeture ultime par critère :

Nous proposons ci-après une implantation efficace de l'ouverture ultime par critère. Cette implantation n'a pas la prétention d'être optimale mais d'être générique pour un sous-ensemble de critères. Ce sous-ensemble sera lié aux propriétés de l'algorithme d'inondation à **niveau constant** dont cette

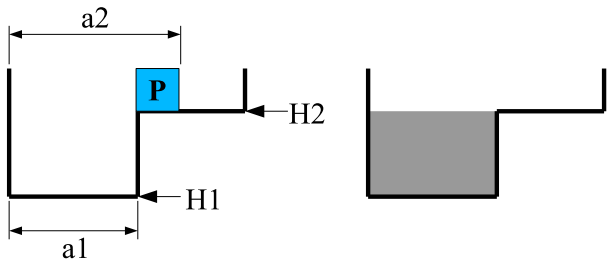
implantation dérive.

8.2.1 Concept

J.Breen and Jones [34] ont montré comment on peut calculer avec seulement une inondation une courbe granulométrique pour un critère croissant et planaire κ donné. En partant de la même approche nous proposons ici une version "locale" de cet algorithme.

Pour l'introduire nous nous intéressons à l'implémentation de l'ouverture/fermeture ultime *surfacique* :

Observons l'évolution de la surface d'un lac au cours du processus d'inondation. Nous pouvons remarquer (voir Figure 8.1) qu'au niveau $h1$ la surface du lac est $a1$. Cette surface ne sera pas modifiée jusqu'à ce que le lac incorpore le pixel p au niveau $h2$.



(a) Image inondée (b) Résidu engendré

FIG. 8.1: Détection d'un résidu engendré par des fermetures surfaciques de tailles croissantes

Si l'on regarde la Figure 8.1(a) sous l'angle de fermetures surfaciques successives de tailles croissantes, on voit qu'aucune fermeture surfacique de taille inférieure ou égale $a1$ ne modifiera les structures de l'image. Une fermeture de taille $a2$ engendrera un résidu non nul (égal à $h2$ -$h1$) c.f. Figure 8.1(b).

8.2.2 Application de ce concept pour le calcul de la fermeture ultime surfacique

En partant de l'observation précédente nous allons montrer à l'aide de la Figure 8.2 comment calculer au fur et à mesure de l'inondation les résidus engendrés par des fermetures surfaciques de tailles croissantes. On part du ou des minima globaux de l'image et pour chaque niveau d'inondation nous regardons les lacs qui ont grandi et nous mettons à jour les images R_θ et q_θ en respectant la définition 17.

On procède à une première phase d'initialisation où, d'une part on mesure la surface associée à chaque minimum local de l'image et, d'autre part on conserve leur niveau respectif. Ceci nous permettra pour la suite de détecter l'accroissement des lacs au cours de l'inondation et de valuer correctement R_θ et q_θ. Une fois cette initialisation réalisée, nous allons suivre les différentes étapes d'inondation.

Au niveau d'inondation trois (Étape 1), le lac $L_1 = \{b\}$ grandit. C'est pourquoi le premier résidu apparaît en b : $R_\theta\{b\} =$ Niveau Courant $-$ Niveau Précédent $= 3 - 2 = 1$ et $q_\theta\{b\} = 2$. En effet on prend la mesure surface de L_1 avant d'atteindre le niveau trois **plus un** : on ajoute un, car le résidu apparaîtrait pour une fermeture de taille supérieure à la surface du lac.

Quand le niveau d'inondation monte à quatre (Étape 2), c'est le lac $L_2 = \{f, g\}$ qui grandit. Les valeurs $R_\theta\{f, g\} = 4 - 1 = 3$ et $q_\theta\{f, g\} = 2 + 1 = 3$ lui sont associées.

On passe ensuite au niveau cinq (Étape 3), un résidu de 2 apparaît pour L_1, ce résidu est supérieur au précédent au point $\{b\}$, nous devons actualiser les deux images comme suit : $R_\theta\{b\} = 2$, $q\{b\} = 3$, les autres points sont également actualisés $R_\theta\{c\} = 2$, $q\{c\} = 3$, $R_\theta\{e, h\} = 1$, $q_\theta\{e, h\} = 5$. Pour $\{f, g\}$ le résidu courant est inférieur aux valuations de $R_\theta\{f, g\}$, il n'y a pas donc pas de modification.

Les lacs L_1 et L_2 sont fusionnés et le processus d'inondation continue. Au niveau six (Étape 4), un résidu de un apparaît. Sa valeur est inférieure à la précédente pour les points $\{b, c, f, g\}$, il n'y a pas de modification de R_θ et q_θ pour ces points. Elle est par contre égale pour les points $\{e, h\}$. Or si l'on se réfère aux équations 6.5, 6.7 l'image q_θ doit être mise à jour : $q_\theta\{e, h\} = 9$. Les points $\{d\}$ et $\{i\}$ n'ont pas encore été traités. On leur assigne les valeurs suivantes : $R_\theta\{d, i\} = 1$ et $q_\theta\{d, i\} = 9$.

Le processus d'inondation s'arrête au maximum de l'image, les points $\{a, i\}$ appartiennent à ce maximum, aucun résidu ne pourra y être engendré.

8.2.3 Généralisation aux critères bi-dimensionnels ou plus

Nous avons vu précédemment dans un cas unidimensionnel qu'il était relativement aisé de calculer les images R_θ et q_θ pour le critère surfacique le long du processus d'inondation. Nous allons voir que l'extension du concept précédent à des critères bi-dimensionnels ou plus n'est pas forcément triviale.

8.2.3.1 Critère surfacique :

Il n'y a pas de problème de généralisation du concept précédent pour des dimensions supérieures à un pour l'utilisation du critère surfacique.

En effet le changement du niveau d'inondation pour un lac donné (qu'il ait lieu lors d'une étape de croissance, ou lors d'une fusion) entraîne automatiquement une modification de la valeur de son critère.

8.2.3.2 Critère autre que la surface :

Si l'on généralise l'algorithme précédent à d'autres critères, il faut noter que **la modification du niveau d'un lac n'implique pas obligatoirement une modification de la valeur de son critère**.

On prendra pour exemple la Figure 8.3(a) où les images R_θ et q_θ (Figures 8.3(b),8.3(c) respectivement) sont engendrées par des fermetures de tailles croissantes utilisant un critère de "largeur" (voir 6.3.2.2 et Figure 6.2).

Lorsque nous atteignons le niveau d'inondation six, cela n'entraîne pas un changement de la valeur du critère du lac (sa largeur est toujours de trois). Si le résidu est actualisé à chaque niveau d'inondation les pixels à 1 sur l'image initiale auront une Transformée $R_\theta = 6 - 1 = 5$ alors que leur vrai résidu (ayant lieu pour une fermeture par largeur de taille 4) est de $8 - 1 = 7$

Nous en déduisons la contrainte suivante :

Contrainte 8.2.1 *Quand peut-on calculer les résidus ?*

Au cours du processus d'inondation, on ne calcule les résidus que pour les lacs qui, ayant changé de niveau d'inondation, changent aussi leur mesure. Si ce n'est pas le cas, les valuations de R_θ (et donc de q_θ) sont erronées.

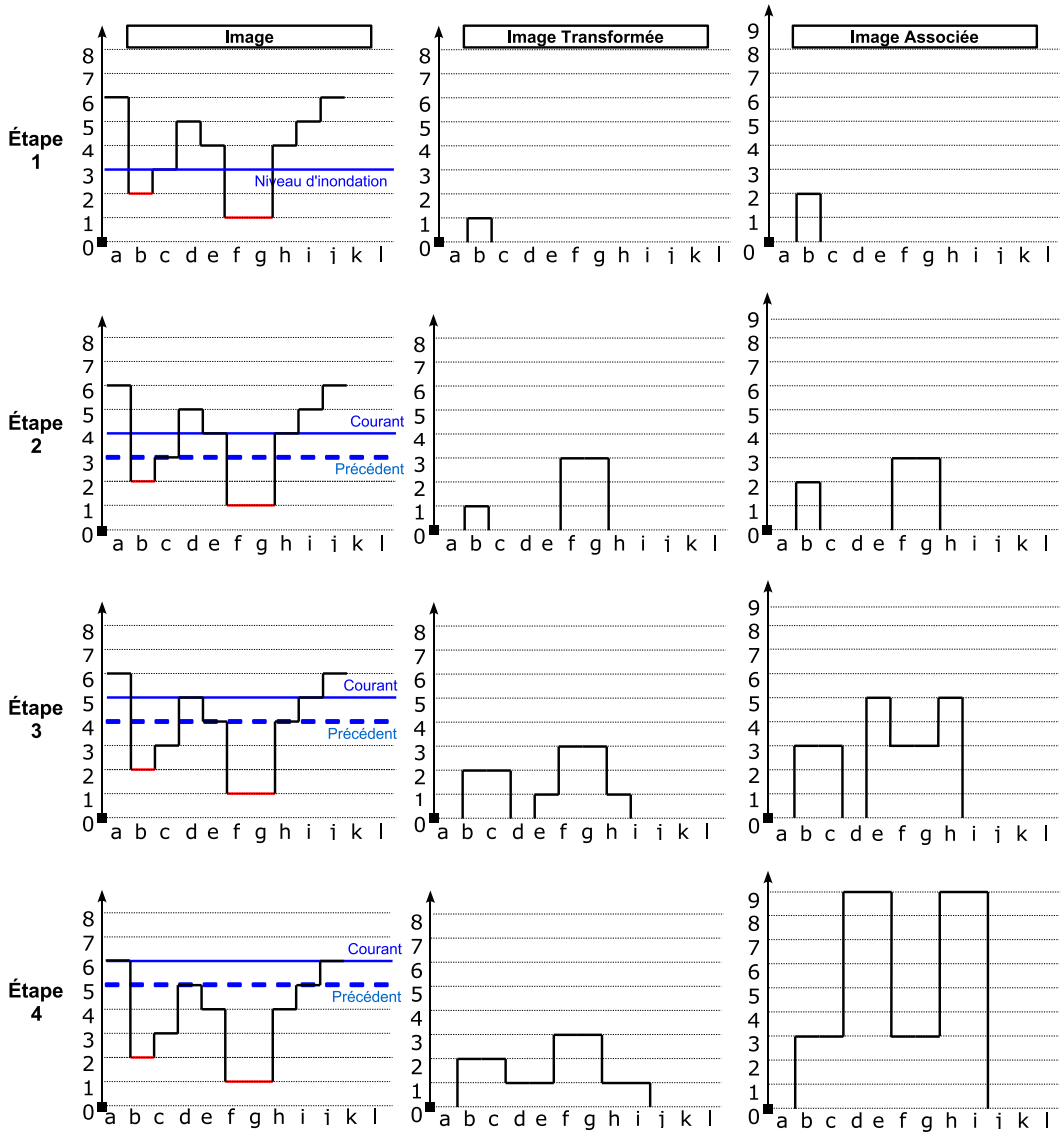
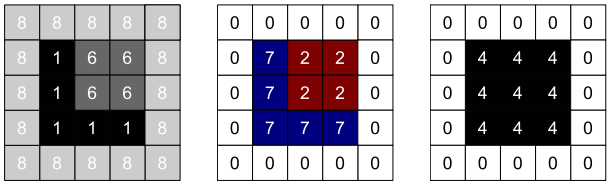


FIG. 8.2: Étapes de l'implémentation par inondation de la fermeture ultime par critère surfacique



(a) Image (b) R_θ (c) q_θ

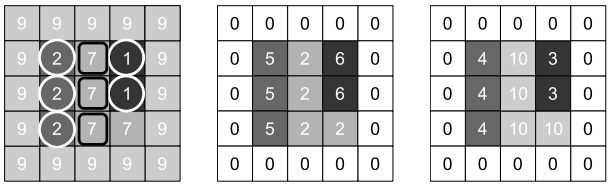
FIG. 8.3: Fermeture ultime utilisant le critère de largeur

8.2.3.3 Configurations particulières :

En partant de la contrainte 8.2.1, nous allons passer en revue quelques-unes des configurations qui justifieront l'implémentation proposée dans l'algorithme 8.3. Cette implantation se veut générique pour l'ensemble des critères définis en 6.3.2.2 et est extensible à d'autres critères croissants et planaires.

Dans les exemples suivants, les images seront inondées à niveau constant en partant de leurs minima globaux, la connexité sera définie par un voisinage 8 connexe, un ensemble donné de pixels $p, q, ...$ sera noté $\{p, q, ...\}$, le niveau d'inondation sera noté S et le niveau d'un lac donné sera noté s_L, enfin la mesure associée au critère ou par abus de langage la valeur du critère sera notée μ.

Valeur du critère à prendre en compte lors de la fusion de lacs : On applique à l'image de la Figure 8.4 une fermeture ultime surfacique (résultat en R_θ et q_θ).



(a) Image (b) R_θ (c) q_θ

FIG. 8.4: Fusion des lacs et fermeture ultime utilisant le critère de surface

Cette image comprend deux minima régionaux (points cerclés en blanc) formant deux lacs L_1 et L_2, de surface $\mu = 3$ et $\mu = 2$, et de niveau $s_{L_1} = 2$ et $s_{L_2} = 1$ respectivement. Les points cerclés de noir font partie de la ligne de partage des eaux entre les deux lacs et appartiennent ainsi "virtuellement" à ces deux lacs.

Si l'on suit le processus d'inondation les deux lacs vont fusionner en un lac L_3 lors de la rencontre du premier pixel de conflit au niveau $S = 7$. Le lac fusionné possède un niveau de $s_{L_3} = 7$ et une surface $\mu = 6$. Ceci valide la contrainte 8.2.1 pour les deux lacs L_1 et L_2, on peut donc calculer la mise à jour des images R_θ et q_θ.

La mise à jour de R_θ ne pose pas de difficulté, il suffit de calculer pour les pixels correspondant à chaque lac la différence entre le niveau du lac fusionné et les niveaux respectifs de chaque lac avant fusion, et de tester si cette différence est supérieure à la valuation courante de R_θ.

La modification de R_θ implique la mise à jour de q_θ. **La valeur du critère du lac fusionné n'est pas la taille de fermeture surfacique qui aurait fermé respectivement L_1 et L_2** . En effet, une fermeture surfacique de taille 3 (respectivement 4) aurait fermé L_2 (respectivement L_1). D'où q_θ n'est pas actualisée avec la mesure du lac fusionné mais avec la mesure de la surface du lac avant fusion **+1**,$q_\theta\{L_1\} = 4$ et $q_\theta\{L_2\} = 3$.

Cet exemple implique la contrainte suivante :

Contrainte 8.2.2 *Valuation du critère à prendre en compte lors d'une fusion*

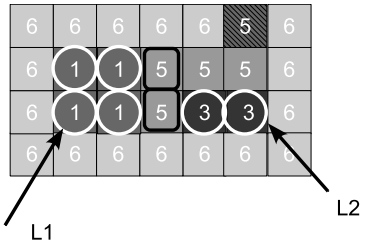
La valeur du critère du lac fusionné n'est pas équivalente aux tailles des fermetures par critère qui auraient fermé chacun des lacs.

Ainsi pour valuer correctement q_θ pour un lac donné, on doit connaître la valeur de son critère avant fusion et pouvoir calculer à partir de cette valeur la taille de la fermeture équivalente.

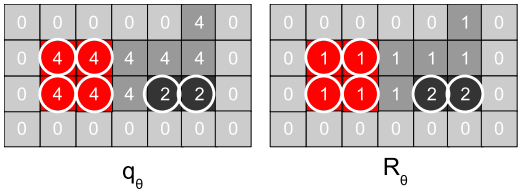
Nous venons de présenter les deux premières contraintes à respecter pour implémenter l'opérateur en suivant le processus d'inondation. Nous avons utilisé le critère surfacique pour lequel il n'y a pas de cas dit pathologique. Nous allons montrer par la suite que pour d'autres critères, de nouvelles contraintes vont apparaître.

La croissance d'un lac n'implique pas forcément une modification de la valeur de son critère : cette affirmation va engendrer deux nouvelles contraintes. Comme précédemment nous utiliserons des exemples pour les mettre en lumière.

Premier cas de figure : On applique à la Figure 8.5(a) une fermeture ultime utilisant un critère de "hauteur".



(a) Image



(b) Résultat erroné dû à la méconnaissance du niveau de gris des points des lacs

FIG. 8.5: Premier cas de figure : résultat erroné

Cette image comprend deux minima régionaux formant deux lacs L_1 et L_2, de mesure $\mu_1 = 2$ et $\mu_2 = 1$, et de niveau $s_{L_1} = 1$ et $s_{L_2} = 3$. Les lacs fusionnent en un lac L_3 de mesure $\mu_3 = 2$ au niveau d'inondation $S = 5$. La contrainte 8.2.1 est validée pour le lac L_2 on peut donc valuer $R_\theta(\{L_2\}) = 5-3 = 2$, $q_\theta(\{L_2\}) = 1+1 = 2$. **Attention la contrainte 8.2.1 n'est pas validée pour** L_1.

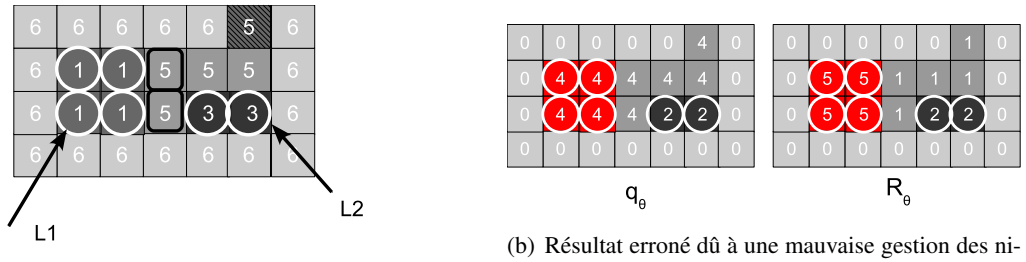
La configuration est la suivante : nous avons un lac fusionné L_3 de niveau $S = 5$. Le processus d'inondation continue jusqu'au niveau $S = 6$, le lac L_3 valide la contrainte 8.2.1, un résidu de $\iota = 6-5 = 1$ apparaît. La contrainte 8.2.1 est validée pour le lac L_3 on peut donc mettre à jour $R_\theta(\{L_3\})$ et $q_\theta(\{L_3\})$. Le résultat est présenté en Figure 8.5(b). **Ce résultat est complètement erroné** (voir Figure 8.7(b)) car lors de la fusion nous n'avons pas pris en compte le fait que la contrainte 8.2.1 n'avait pas été validée pour les points de L_1. De plus la seule connaissance du niveau du lac fusionné est insuffisante pour une évaluation correcte des résidus pour les points de L_1.

Partant de ce constat on se doit d'imposer la contrainte suivante :

Contrainte 8.2.3 *Connaissance de la valuation des points d'un lac La connaissance du niveau d'inondation d'un lac n'est pas suffisante pour le calcul des résidus en tous ces points. On se doit de connaître le niveau de gris de chaque point.*

Second cas de figure : partant du constat précédent nous reprendrons le même exemple mais en utilisant la contrainte 8.2.3.

Déroulons une nouvelle fois le processus. Cette image comprend deux minima régionaux formant deux lacs L_1 et L_2, de mesure $\mu_1 = 2$ et $\mu_2 = 1$, et de niveau $s_{L_1} = 1$ et $s_{L_2} = 3$. Les lacs fusionnent en un lac L_3 de mesure $\mu_3 = 2$ au niveau d'inondation $S = 5$. La contrainte 8.2.1 est validée pour le lac L_2 on peut donc calculer $R_\theta(\{L_2\}) = 5-3 = 2$, $q_\theta(\{L_2\}) = 1+1 = 2$.



(a) Image

(b) Résultat erroné dû à une mauvaise gestion des niveaux des lacs lors de la fusion

FIG. 8.6: Second cas de figure : Résultat erroné

On assigne aux points de L_2 la valeur 5. De nouveau la contrainte 8.2.1 n'est pas validée pour L_1.

La configuration est la suivante : les points qui **appartenaient** à L_1 ont un niveau propre de $s_{L_1} = 1$, les points qui appartenaient au lac L_2 ont un niveau propre de $s_{L_2} = 5$ et le lac fusionné L_3 est au niveau $S = 5$.

Le processus d'inondation continue jusqu'au niveau $S = 6$, le lac L_3 valide la contrainte 8.2.1, un résidu de $\iota = 6 - 5 = 1$ apparaît pour le sous-ensemble $\{L3\} - \{L1\} \cup \{L2\}$. On met à jour $R_\theta = 1$, $q_\theta = 4$ pour ce sous-ensemble. Pour les points de $\{L1\}$ un résidu de $\iota = 6 - 1 = 5$ apparaît, on met à jour $R_\theta\{L1\} = 5$,$q_\theta\{L1\} = 4$ (Voir Figure 8.6(b)).

Là encore le résultat est erroné (voir Figure 8.7(b)), en effet un résidu $\iota = 4$ aurait dû être généré pour une fermeture de taille 3 pour les points de $\{L1\}$. La prise en compte du niveau propre des pixels est insuffisante pour traiter correctement les points issus de $\{L1\}$. Donner à $\{L3\}$ le niveau d'inondation 5 ne nous a pas permis de détecter cette configuration. La contrainte suivante va nous le permettre.

Contrainte 8.2.4 *Niveau d'inondation à conserver lors d'une fusion*

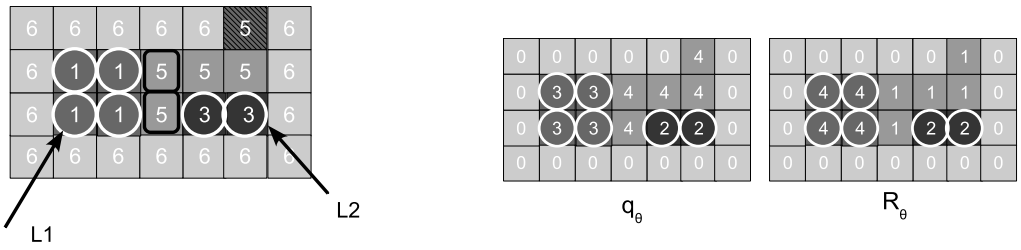
Lors de la fusion de deux lacs L_1 et L_2 de niveaux s_{L_1} et s_{L_2} en un lac L_3 au niveau d'inondation S, on ne doit pas modifier le niveau du lac qui n'a pas changé la mesure de son critère. Cela revient à affecter à L_3 non pas S mais le niveau du lac qui n'a pas changé la mesure de son critère. Si aucun des deux n'a changé de mesure on pourrait prendre arbitrairement l'un de ces niveaux. ***Nous allons retenir comme convention de prendre l'inf de ces niveaux. Ceci imposera de maintenir la hauteur courante d'inondation d'un lac même si celui-ci a été préalablement gelé (voir contrainte 8.2.6 p. 98).***

Application des deux nouvelles contraintes : Reprenons l'ensemble du processus en utilisant les contraintes 8.2.3 et 8.2.4.

1. Avant fusion on a : L_1 de niveau $s_{L_1} = 1$ et de mesure $\mu_1 = 2$ et L_2 de niveau $s_{L_2} = 3$ et de mesure $\mu_2 = 1$.
2. Lors de la fusion au niveau d'inondation $S = 5$, L_2 valide la contrainte 8.2.1, on met à jour $R_\theta\{L_2\} = 5 - 3 = 2$, $q_\theta\{L_2\} = 1 + 1 = 2$ et s_{L_2} à 5. **On assigne aux points de L_2 la valeur 5**. L_1 ne change pas de mesure. Il n'est donc pas actualisé suivant la contrainte 8.2.1.
3. On fusionne les lacs $\{L_3\} = \{L_1\} \cup \{L_2\}$, on calcule $\mu_3 = 2$ **et on impose** $s_{L_3} = Inf(s_{L_1}, s_{L_2}) = Inf(5, 3) = 3$.

4. Le lac L_3 grandit au niveau 5, **la contrainte 8.2.1 est validée lorsqu'il incorpore le pixel de valeur 5 hachuré sur la Figure 8.7(a)**. Un résidu de $\iota = 5 - 1 = 4$ apparaît pour $\{L_1\}$ et met à jour $R_\theta\{L_1\} = 4$ et $q_\theta\{L_1\} = \mu_1 + 1 = 3$. **On assigne aux points de L_3 la valeur 5.**
5. Le lac L_3 grandit au niveau 6, il valide la contrainte 8.2.1, un résidu de $\iota = 6 - 5 = 1$ apparaît, on met à jour $R_\theta\{L_3\}$, $q_\theta\{L3\}$.

Le processus est terminé et l'on obtient le résultat correct présent sur la Figure 8.7(b).



(a) Image

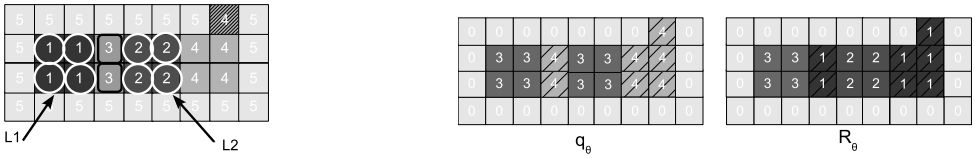
(b) Résultat attendu

FIG. 8.7: Résultat correct

Coût algorithmique de la contrainte 8.2.4 Cette contrainte peut être algorithmiquement lourde de conséquence :

1. Elle impose de connaître la mesure associée à un lac même quand celui-ci se propage sur un plateau. Pour les critères décrits en 6.3.2.2 p.61, ce coût est négligeable et n'a pas d'impact sur l'efficacité de l'algorithme. Cependant cette contrainte pourra devenir pénalisante pour d'autres critères dont la mesure ne peut pas être connue en temps constant.
2. On peut avoir à calculer les résidus lors des propagations sur des plateaux, ce qui pose des problèmes pour définir la complexité théorique de l'algorithme.

Peut-on lors de la fusion résoudre la contrainte 8.2.4 ? Sur la Figure 8.8(a) est présentée une configuration légèrement plus complexe que la précédente. On calcule une fermeture ultime utilisant le critère de "hauteur", dont le résultat est présenté sur la Figure 8.8(b). Nous verrons à l'aide de cet exemple que la contrainte 8.2.4 ne peut être résolue lors de la fusion des lacs.



(a) Image

(b) Fermetures ultimes utilisant un critère de hauteur

FIG. 8.8: Exemple de problèmes lors de la fusion de lacs pour le critère de hauteur

1. Avant fusion on a : L_1 de niveau $s_{L_1} = 1$ et de mesure $\mu_1 = 2$ et L_2 de niveau $s_{L_2} = 2$ et de mesure $\mu_1 = 2$.

2. Lors de la fusion au niveau d'inondation $S = 3$, **aucun** des lacs ne valide la contrainte 8.2.1. On ne peut pas mettre à jour R_θ et q_θ.
3. On fusionne les lacs, on calcule $\mu_3 = 2$ **et on impose** $s_{L_3} = inf(s_{L_1}, s_{L_2}) = 1$ conformément à la contrainte 8.2.4.
4. Le lac L_3 grandit au niveau $S = 3$, la contrainte 8.2.1 n'est pas validée en effet $\mu_3 = 2$.
5. Le lac L_3 grandit au niveau $S = 4$, L_3 valide la contrainte 8.2.1 quand il incorpore le pixel de valeur 4 hachuré sur la Figure 8.8(a). On peut mettre à jour R_θ, q_θ ($R_\theta\{L_3\} - \{L_1\} \cup \{L_2\} = 4 - 3 = 1$, $q_\theta\{L_3\} - \{L_1\} \cup \{L_2\} = 2 + 1 = 3$, $R_\theta\{L_2\} = 4 - 2 = 2$, $q_\theta\{L_2\} = 2 + 1 = 3$, $R_\theta\{L_1\} = 4 - 1 = 3$, $q_\theta\{L_1\} = 2 + 1 = 3$). Et on peut imposer $s_{L_3} = 4$
6. Le lac L_3 grandit au niveau $S = 5$, lors de cette croissance il valide la contrainte 8.2.1, un résidu de $S - s_{L_3} = 5 - 4 = 1$ est engendré, il est inférieur au résidu précédent sauf pour les points de niveau 3 et 4 de l'image 8.8(a). On met à jour R_θ et q_θ pour ces points. Et on obtient le résultat correct de la Figure 8.8(b).

Traitement du dernier niveau d'inondation ne vérifiant pas la contrainte 8.2.1 Comme nous l'avons vu précédemment, le calcul des résidus le long du processus d'inondation est régi par la contrainte 8.2.1. Quand l'inondation arrive au maximum global de l'image , il nous manque un niveau "virtuel" au-dessus de ce maximum pour traiter ce dernier niveau. Ceci se produit si l'on effectue une fermeture ultime par critère dont la valeur d'arrêt est supérieure ou égale à la valeur maximale admissible par l'image (par exemple pour le critère surfacique, la valeur d'arrêt est supérieure à la surface de l'image).

Contrainte 8.2.5 *Traitement du dernier niveau*
Pour traiter ce cas, on doit donc effectuer un post-traitement :

1. *L'image est complètement inondée et forme un seul lac L. Pour ce lac on connaît entièrement les pixels qui le composent et leur niveau.*
2. *On calcule la différence entre le maximum de l'image et les niveaux des pixels et on met à jour si nécessaire R_θ et q_θ.*

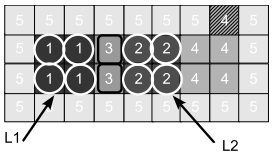
Arrêt de l'algorithme en cours d'inondation : Comme nous l'avons souligné à la Section 7.2, l'opérateur d'ouverture/fermeture ultime est un opérateur sans mémoire. Une solution à ce problème consiste à arrêter l'algorithme pour une certaine valeur λ, avant qu'une forte transition de l'image n'efface de R_θ et q_θ l'information d'intérêt.

Étudions maintenant comment gérer correctement l'arrêt de l'algorithme au cours de l'inondation. Comme précédemment nous allons formaliser ce cas sous la forme d'une contrainte :

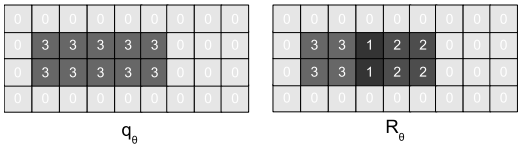
Contrainte 8.2.6 *Prise en compte d'une valeur d'arrêt au cours de l'inondation*
Les contraintes 8.2.2 et 8.2.4 impliquent que la contrainte 8.2.1 doit pouvoir être vérifiée , lors de la fusion entre lacs mais également lors de la propagation sur les plateaux.
Ceci impose que :

1. *Croissance d'un lac : on n'autorise la croissance d'un lac que si la valeur du critère associée à ce lac est strictement inférieure au seuil d'arrêt, sinon on "gèle" le lac.*
2. *Fusion : on ne teste la contrainte 8.2.1 pour chaque lac que si la valeur du critère associée à ce lac est inférieure ou égale au seuil d'arrêt.*

On pourra reprendre la Figure 8.9(a) et procéder à une fermeture ultime utilisant le critère de hauteur mais en arrêtant l'algorithme pour une valeur de $\lambda = 3$, le résultat est présenté sur la Figure 8.9(b). Dans cette situation le niveau 5 d'inondation n'est pas considéré car au niveau 4 tous les lacs de l'image ont atteint le critère d'arrêt $\lambda = 3$.



(a) Image



(b) Fermeture ultime utilisant un critère de hauteur, avec un seuil d'arrêt de l'algorithme de $\lambda = 3$

FIG. 8.9: Exemple d'arrêt de l'opérateur de fermeture ultime utilisant le critère de hauteur

8.3 Détails de l'implémentation proposée :

La description de l'implémentation est détaillée sous forme d'un pseudo-code (voir Section 8.3.2 p.100), complétée de sa description exhaustive en 8.3.3 p. 104. Celle-ci tendra le plus possible à relier le code proposé avec les différentes contraintes exposées au long de la Section 8.2.

Dans la section ci-après nous clarifierons quelques conventions de nommage utilisées au sein de l'algorithme.

8.3.1 Structures de données et conventions de nommage

Certaines des conventions de nommage algorithmique sont décrites p.221, nous les complétons ici. Nous introduisons les images (noms et *typage*) ainsi que quelques structures de données auxiliaires nécessaires.

Conventions Images

- imRelief : l'image inondée
- imlabels : les minima marqués de l'image ImRelief (s'il n'y a pas de minimum marqué, le statut du pixel est label-fond)
- imTravail : image indiquant le statut d'un pixel (traité, non-traité, dans-fah)
- imPropageLabels : image dans laquelle se propageront les labels
- R_θ : l'image Transformée
- q_θ : l'image Indicatrice

Structures de données

- pixel p : pour alléger l'écriture la dénomination pixel pourra signifier conjointement la position et/ou la valeur du pixel
- fah : file d'attente hiérarchique
- Equivalence : tableau d'équivalence, qui permettra de gérer la fusion des lacs
- NiveauCourantLac : tableau qui conserve le niveau courant de chaque lac
- ValeurPrecCritere : tableau qui conserve la mesure associée au critère pour chaque lac
- TabCritère : tableau d'*objet SupportCritère*. On associe un objet à chaque label. Il est chargé de maintenir à jour la mesure du critère d'un lac quand celui-ci grandit ou fusionne. Il est défini en

p.100. Par exemple pour le critère surfacique, *SupportCritère* est extrêmement simple : l'attribut C est un compteur, lors de l'appel de AjouterPixel(p) on incrémente C de 1, lors de l'union on additionne les valeurs des compteurs et l'appel de ValeurAssociéeAuCritère() renvoie la valeur de C.

– TabDynLac : On associe à chaque label une liste, celle-ci conservera la position (pos) et la valuation(val), de chaque pixel associé à un bassin versant (conformément à la contrainte 8.2.3). Ainsi TabDynLac[Label] pointera sur la liste contenant les points associés à Label.

Structure SupportCritère
Data : C Structure de données auxiliaires contenant les informations pour calculer la valeur du critère
`AjouterPixel(pixel P)`
→ Mise à jour de C suite à l'ajout du pixel P
`Union(SupportCritère)`
→ Mise en commun des C respectives
`ValeurAssociéeAuCritère()`
→ Calcule la valeur associée au critère et renvoie cette valeur

8.3.2 Pseudo-code

Algorithme 8.3 : Algorithme de fermeture ultime par critère :Initialisation

Data : $imRelief$ Image d'entrée, $imLabels$ Image des Labels, $MaxCriterion$(Plus grande étape de fermeture), $\mathcal{N}$ (Voisinage Considéré)

Result : R_θ (image *Transformée), q_θ (image Indicatrice)*

• ***Initialisation des structures de données*** **:**

fah `// Création de la file d'attente hiérarchique`

imPropageLabel ← imLabels

imTravail ← **non-traité**

$R_\theta \leftarrow 0$

$q_\theta \leftarrow 0$

• ***Initialisation*** **:**

`// On parcourt l'image de labels (`$Imlabels$`). A chaque label correspond un minimum (imposé ou non) qui forme le lac initial. Pour chaque lac on conserve son niveau, sa mesure associée au critère, et les pixels(position et valuation) le constituant`

forall *pixels p de imLabels* **do**

  if $p \neq$ ***label-fond*** **then**

    imTravail(p) ← **traité**

    NiveauCourantLac(p) ← imRelief(p)

    `// On doit connaître la valuation de chaque pixel, voir contrainte 8.2.3`

    TabDynLac[p] ← ajouter p

    TabCritère[p].AjouterPixel(p)

    `// Les voisins des labels sont introduits dans la fah`

    forall *pixels v* $\in \{\mathcal{N}_{(p)} \backslash p\}$ **do**

      if $imLabels(v) =$ ***label-fond*** **then**

        `// Imposition des marqueurs`

        fah ← ajouter (p,v) avec priorité ∨(imRelief(p),imRelief(v))

        imTravail(v)← **dans-fah**

`// On a fini de parcourir les minima, leurs pixels voisins sont dans fah`

for $i \leftarrow 1$ **to** *NumLabels* **do**

  `// L'inondation n'a pas débuté. On initialise` $ValeurPrecCritere$ `avec la valeur associée au critère des minima. Ceci permettra d'évaluer par la suite la contrainte 8.2.1 p.92`

  ValeurPrecCritere[i] ← TabCritère[i].ValeurAssociéeAuCritère()

  `// Aucune fusion. Chaque lac est son propre représentant`

  Equivalence[i] $= i$

`// Fin de l'initialisation`

Algorithme de fermeture ultime par critère : Propagation

21 • ***Propagation*** **:**
22 **while** *fah* $\neq$ *VIDE* **do**
    `// On extrait les pixels de la fah`
23 prio-ws=extraire priorité maximale de fah
24 pInondant, pInondé ← extraire paire de pixels au niveau prio-ws
    `// Le point est-il fermement assigné ?`
25 **if** *imWork(pInondé)* $\neq$ ***traité*** **then**
        `// On récupère le label du point inondant`
26 Label = imPropageLabel(pInondant)
        `// On propage ce label au point inondé`
27 imPropageLabel(pInondé) ← Label
        `// Le point devient fermement assigné`
28 imWork(pInondé) ← **traité**
        `// Récupération du lac représentant`
29 TopLabel1 = TrouverLacReprésentant(Label) 72
        `// Le Lac grandit, ceci peut entrainer une mise à jour des résidus`
30 CroissanceLac(prio-ws,TopLabel1,pInondé) : Goto 42
31 **forall** *pixels* $v \in \{\mathcal{N}_{(pInondé)} \setminus pInondé\}$ **do**
32 **if** *imWork(v)* = ***traité*** **then**
                `// Points de conflit : appartiennent-ils au même lac ?`
33 TopLabel2=TrouverLacReprésentant(imPropageLabel(v))
34 **if** *TopLabel1* $\neq$ *TopLabel2* **then**
                    `// Ils proviennent de lacs différents ! Les lacs sont fusionnés`
35 FusionDeLacs(prio-ws, TopLabel1, TopLabel2) : Goto 52
36 **else**
                `// Point non encore traité : Insertion dans la FAH`
37 **if** *imWork(v)*=***non-traité*** **then**
                    `// Imposition des marqueurs : on ajoute le point à la priorité imRelief(v), sauf si imRelief(v) est inférieur à la priorité maximale courante`
38 fah ← ajouter (pInondé,v) avec priorité $\vee$(imRelief(v),prio-ws)
39 imWork(v)← **dans-fah**

40 • ***Post-Traitement*** **:**
`// Lancement du post-traitement éventuel : cf Contrainte 8.2.5 p. 98`
41 Goto 75

CroissanceLac(prio-ws, Label, pixel)

```
   // Le lac n'est pas gelé (ie Le critère de stop n'est pas atteint).
42 if ValeurPrecCritere[Label]<MaxCriterion then
43 |  ListePixelLac ← TabDynLac[Label] // Liste des pixels du lac pointés par
   |      label
   |  // On doit connaître la valuation de chaque pixel, voir contrainte 8.2.3
44 |  ListePixelLac ← ajouter pixel
45 |  TabCritère[Label] ← ajouter pixel
46 |  ValeurCritere ← TabCritère[Label].ValeurAssociéeAuCritère()
   |  // On ne sait pas par avance quel point du plateau peut engendrer un
   |      calcul de résidus. On doit donc tester systématiquement les
   |      contraintes 8.2.1 et 8.2.6
47 |  if ( ValeurCritere>ValeurPrecCritere[Label] et ValeurCritere<MaxCriterion et
   |  p>NiveauCourantLac[Label]) ou (ValeurCritere=MaxCriterion) then
48 |  |  MiseAJourResidus(prio-ws,Label,ValeurCritere) : Goto 66
49 |  |  NiveauCourantLac[Label] ← prio-ws
   |  // On doit tenir systématiquement à jour la valeur associée au critère
50 |  ValeurPrecCritere[Label] ← ValeurCritere
51 else
   |  // Le lac est gelé, aucun résidu ne sera engendré par la suite. Cependant
   |      son niveau d'inondation doit être mis à jour pour que l'inondation se
   |      poursuive. Ceci est une conséquence collatérale de la contrainte 8.2.4
   |      p.96
52 |  NiveauCourantLac[Label] ← prio-ws
```

FusionDeLacs(prio-ws,Label1,Label2)

```
   // On récupère la valeur associée au critère avant fusion cf Contrainte
      8.2.2
53 ValeurCritère1 ← TabCritère[Label1].ValeurAssociéeAuCritère()
54 ValeurCritère2 ← TabCritère[Label2].ValeurAssociéeAuCritère()
   // On calcule la valeur associée au critère après fusion
55 TabCritère[Label1].Union(TabCritère[Label2])
56 ValeurCritèreFusion ← TabCritère[Label1].ValeurAssociéeAuCritère()
   // Contraintes 8.2.1, 8.2.2 p.92, 94 : S'il y a eu modification de la valeur
      du critère et changement de niveau, et que l'on n'a pas atteint la valeur
      d'arrêt de l'algorithme (Contrainte 8.2.6) , des résidus ont pu être
      engendrés pour les deux lacs
57 if ValeurCritereFusion > ValeurCritere1 et ValeurCritere1+1 ≤ ValeurdeStop et prio-ws >
   NiveauCourantLac[Label1] then
58 |  MiseAJourResidus(prio-ws,Label1,ValeurCritere1+1) : Goto 66
   |  // Le niveau du lac n'est mis à jour que si des résidus ont été
   |     engendrés.
59 |  NiveauCourantLac[Label1] ← prio-ws
60
61 Répéter les lignes 57 à 59 pour le lac 2 (pointé par Label2)

   // Mise à jour des structures de données après fusion
62 Equivalence[Label2] ← Label1
63 TabDynLac[Label1] ← TabDynLac[Label1] ∪ TabDynLac[Label2]
   // Application de la contrainte 8.2.4
64 NiveauCourantLac[lab1] ← ∧(NiveauCourantLac[lab1],NiveauCourantLac[lab2])
   // La valeur associé au critère est systématiquement mise à jour
65 ValeurPrecCritere[Label1] ← ValeurCritereFusion
```

MiseAJourResidus(prio-ws,Label,ValeurDuCritere)

```
66 ListePixelLac ← TabDynLac[Label] // Liste des points (position et valuation) du
      lac pointé par Label
67 foreach point p dans ListePixelLac do
68 |  résidu ← prio-ws - p // Résidu courant en p
69 |  p ← prio-ws // fermeture à la hauteur de traitement, voir contrainte 8.2.3
70 |  if (résidu ≥ R_θ(p) et résidu > 0) then
71 |  |  R_θ(p) ← résidu ;q_θ(p) ← ValeurDuCritere
```

8.3.3 Explication détaillée du pseudo-code précédent

Dans cette section nous allons dérouler le pseudo-code de l'algorithme 8 . 8.3 p.101 en faisant apparaître le plus clairement possible le rôle des contraintes 8.2.1, 8.2.2, 8.2.3, 8.2.4, 8.2.5, et 8.2.6. Les définitions des différentes structures de données et notations sont présentées de manière exhaustive en Section 8.3.1 p.99.

TrouverLacReprésentant(Label)

```
72 while Label≠ Equivalence[Label] do
73     Label ←Equivalence [Label]
74 renvoyer Label
```

Post-Traitement()

```
   // Fermeture au maximum global de l'image : Contrainte 8.2.5 p. 98
75 if MaxCriterion ≥ Valeur maximale admissible par ImRelief then
76     Max ← Valeur maximale de ImRelief
77     ValeurDuCritere ← Valeur maximale admissible par ImRelief
78     foreach List dans TabDynLac do
79         foreach point p dans List do
80             résidu ← Max - p // Dernier résidu en p
81             p ← Max // fermeture à la hauteur de traitement
82             if (résidu ≥ Rθ(p) et résidu > 0) then
83                 Rθ(p) ← résidu ; qθ(p) ← ValeurDuCritere
```

Il s'agit d'une modification d'un algorithme de détermination de la ligne de partage des eaux contraint par file d'attente hiérarchique. Nous reprendrons donc en grande partie le formalisme qui y est associé : aussi nous nous servirons d'une image de travail "imTravail" contenant les états des points à inonder (non-traité, dans-fah, traité) voir ci-après. Les labels propagés seront eux stockés dans l'image de labels "imPropageLabel".

Nous rappellerons ici quelques conventions liées aux états des pixels dans l'image de travail et l'image de labels :

Image de Travail :

non-traité : le point ne possède pas encore de label. Par ailleurs, le point ne se trouve dans aucun niveau de la file d'attente hiérarchique.

dans-fah : le point apparaît dans au moins un niveau de hiérarchie de la file d'attente hiérarchique.

traité : le point est labellisé, et son label ne changera pas par la suite. Il pourra servir à la détermination de l'état d'autres points. La gestion des labels eux-mêmes et des dépendances lors des fusions des bassins seront adressés par l'image "imPropageLabel" et une table d'équivalence.

Image de labels :

label-fond : état d'un point d'une image de labels, pour lequel aucun label n'a été assigné.

L'algorithme peut se scinder en trois grandes étapes distinctes, l'**Initialisation** (eg : des images, des structures de données auxiliaires), la **Propagation** des points qui simule le processus d'inondation et un **Post-traitement** éventuel. Nous nous efforcerons de souligner les modifications à apporter à l'algorithme *standard* permettant de produire ce nouvel algorithme.

I. Initialisation : Cette phase (ligne 1 à 20 p.101) se déroule en trois grandes étapes :

1. Initialisation des images et des structures de données :

   * **Initialisation des images l.1 à l.6** L'image de travail ne contient que des points marqués *non-traité* (i.e candidat) et l'image de labels à propager (i.e. les marqueurs) est copiée dans "imPropageLabel".

     Ajout par rapport à l'algorithme classique : les images de sortie R_θ, q_θ sont initialisées à zéro.

   * **File d'attente hiérarchique l.2** Nous créons une F.A.H. Elle nous permettra de traiter les points de l'image en fonction de leur hauteur topographique : les points de plus faibles valeurs topographiques seront ceux de priorité la plus élevée.

2. Mise en place du processus de propagation (l.7 à l.17) : le parcours de l'image de label, va nous permettre de mettre en place le processus d'inondation et d'initialiser l'ensemble des données auxiliaires (points le constituant, valeur du critère associé,...) définissant un lac.

   * **Reprise de l'algorithme *standard* :** lors du parcours d'un label en particulier, tous les points lui appartenant sont marqués comme "traité"(l.10). Pour chaque point labellisé les voisins (v) sont extraits (l.15). Ceux ne possédant pas encore de label peuvent participer au processus d'inondation : ils sont placés dans la "fah" à la priorité correspondante à la hauteur topographique de l'image "imRelief" [1]. Ces points sont marqués "dans-fah".

   * **En parallèle de l'algorithme *standard* :** lors du parcours d'un label en particulier et conformément à la contrainte 8.2.3 un tableau dynamique contenant l'ensemble des points le constituant est rempli (l.12). On conserve sa hauteur topographique (l.11) et on lui associe une structure "SupportCritère" (cf 8.2 p.100) (l.13) qui permettra de maintenir la valeur associée au critère au long du processus d'inondation.

3. Calcul et sauvegarde de la valeur du critère pour les étapes ultérieures et mise en place des structures assurant le suivi des fusions de lacs (l.18 à l.20) :

   * **Algorithme *standard* et table d'équivalence** Initialisation d'une table d'équivalence : elle permettra de suivre les fusions entres labels (i.e Bassins Versants). Aucune fusion n'ayant eu lieu, chaque label est son propre représentant.

   * **Sauvegarde des valeurs associées au critère pour les étapes ultérieures** Conformément à la contrainte 8.2.1, la mise à jour des résidus ne doit avoir lieu que si un changement de niveau d'inondation s'accompagne d'une modification de la valeur associée au critère. L'inondation n'ayant pas encore commencé, la valeur courante/précédente associée au critère est la même et est initialisée en (l.20).

Récapitulons :

Pour chaque label/marqueur (associé à un lac), nous connaissons sa hauteur topographique (dans "NiveauCourantLac"), l'ensemble des points le constituant (dans "TabDynLac"), nous avons initialisé une structure pour la mise à jour du critère (dans "TabCritère"). Nous avons propagé les labels aux pixels voisins des marqueurs et mis ceux-ci dans la "fah". Nous connaissons la valeur associée au critère de chaque marqueur (dans "ValeurPrecCritere").

Le processus d'inondation peut commencer.

[1] Plus précisément au suprémum des valuations entre $imRelief(v)$ et $imRelief(p)$, pour permettre l'imposition des marqueurs *à la volée*

II. Propagation Cette phase (lignes 21 à 39 p.102) est plus compliquée à scinder en sous-étapes clairement définies. Nous allons donc suivre le processus mis en place et comme précédemment nous soulignerons les modifications à apporter à l'algorithme *standard* permettant de produire ce nouvel algorithme.

Nous rentrons ici dans l'étape de propagation des labels. Ils suivront les priorités liées aux valeurs topographiques des pixels.

Remarque 8 *Les modifications propres au nouvel algorithme ont été factorisées dans des fonctions/procédures/sous-programmes annexes, le lecteur pourra/devra naviguer entre le coeur du processus de propagation p.102 et ces fonctions.*

II.a Traitement des points issus de "fah" : nous extrayons les points de priorité la plus importante (i.e de plus faibles valuations topographiques). Plus précisément nous ne récupérons pas un mais **deux point** à chaque étape d'extraction : le point dit inondé pInondé ainsi que le point

Commençons (l.22 à l.28) : si le point Inondé n'est pas marqué comme **traité** (l.25), il peut participer au processus d'inondation. On récupère le label du pixel Inondant et on le propage au pixel Inondé, ce point est considéré maintenant comme **traité**.

Maintenant nous pouvons commencer à souligner ce qui est propre au nouvel algorithme et ce qui appartient à l'algorithme *standard* d'inondation :

1. Nous devons en premier regarder si le nouveau point traité peut engendrer des résidus.
2. Nous soucier du traitement des voisins de ce pixel, pour assurer le processus d'inondation.

II.b Croissance d'un lac et mise à jour des résidus potentiels : nous récupérons le lac représentant l.29 (i.e s'il y a eu des fusions préalables de lacs, nous récupérons le label représentant du lac fusionné). Nous passons pInondé (avec sa priorité et le lac auquel il peut être assigné) à la fonction ad-hoc "CroissanceLac" (l.30). Cette fonction est définie en l.42 p.103. Intéressons-nous à la description de cette fonction :

CroissanceLac : "pInondé" peut maintenant être assigné à un lac donné et peut engendrer une mise à jour des résidus en celui-ci. On teste (l.42) si ce lac n'a pas déjà été gelé. Nous devons traiter deux cas :

Le lac n'est pas gelé : "pInondé" est assigné à ce lac (il est ajouté à la collection des points du lac conformément à la contrainte 8.2.3 et le critère du lac est mis à jour).

1. **On ne sait pas par avance quel point du plateau peut engendrer un calcul de résidus**. On teste donc systématiquement les contraintes 8.2.1 et 8.2.6. Si l'une d'elles est vérifiée on lance la fonction ad-hoc de calcul des résidus (l.48) en ce lac et on met à jour la hauteur courante du lac **après ce calcul**. La fonction de mise à jour des résidus est définie en page 108.
2. La valeur courante du critère est sauvegardée dans "ValeurPrecCriteria" pour assurer le test correct de la contrainte 8.2.1 par la suite.

Le lac est gelé : ceci indique qu'aucun nouveau point ne pourra lui être *physiquement* assigné et aucun résidu ne pourra y être engendré. Cependant **nous devons utiliser ce point pour la mise à jour de la hauteur courante du lac auquel il est donc *virtuellement* assigné**, ceci est imposé par la convention de la contrainte 8.2.4.

II.c Suite du processus de propagation : pour chaque point "pInondé" nous extrayons l'ensemble de ses voisins "v" (l.31), comme lors de la phase d'initialisation, ces voisins ont été classés selon leurs "états" (sauvegardés dans "imTravail"). Ces "états" vont engendrer deux cas de figure :

"v" est non-traité (l.37) : on le place dans la "fah" à la priorité correspondant à la hauteur topographique de l'image "imRelief"[2].

"v" est traité (l.32) :cela signifie qu'un label lui a déjà été assigné. On vérifie l.34 si ces deux labels ne référencent pas le même lac représentant. Si ce n'est pas le cas, **nous sommes en présence d'un point de conflit**. Ceci va entraîner la fusion des lacs pointés respectivement par $TopLabel1$ et $TopLabel2$ et la création potentielle de résidus. Cette tâche est sous-traité à la fonction **FusionDelacs** déclarée en l.52 p.103 et définie ci-après.

FusionDelacs :

Nous commençons par déterminer la valeur du critère qui sera associée au lac fusionné (l.52 à 56) ceci nous permet d'évaluer les contraintes 8.2.1 et 8.2.2 pour chacun des lacs en cours de fusion.

Le test des contraintes est présenté pour le premier lac en l.57. Si le test est positif on calcule les résidus pour ce lac et **on met à jour sa hauteur courante.**

Remarque 9 *Notons que l'on donne à la fonction de calcul des résidus la valeur du critère précédent **plus un** (cf l.58). Ceci est guidé par la contrainte 8.2.2. Cette valeur est correcte pour l'ensemble des critères utilisés dans cette note, il faudra par contre l'adapter pour d'autres types de critères (i.e définir un pas d'incrément unitaire pour chaque critère).*

Un test similaire est réalisé pour le second lac en l. 61.

On procède ensuite à la fusion **effective** des lacs, on optera pour la convention suivante : le premier lac absorbera le second et deviendra de fait le lac fusionné [3]. Le niveau courant du lac fusionné est **l'inf (voir contrainte 8.2.4 p.96)** des niveaux courants des deux lacs (l.64). La table d'équivalence est actualisée, la collection des points du lac absorbé est transmise au lac absorbant et l'on met à jour "ValeurPrecCriteria" pour une évaluation correcte de la contrainte 8.2.1 par la suite.

Fonction ad-hoc de calcul des résidus pour un lac donné (l.66) : comme nous l'avions mentionné, le calcul des résidus a été factorisé dans une fonction "ad-hoc". En effet pour calculer les résidus pour un lac donné, il suffit de connaître la hauteur topographique de fermeture (variable "val"), le lac (*indexé* par la variable "label"), et la valeur de fermeture par critère à laquelle cela correspond (variable "ValeurDuCritère").

Aussi l.66, on récupère l'ensemble des points du lac *indexé* par "label". Pour chaque point, on calcule le résidu courant **et on assigne au point la hauteur de fermeture (voir contrainte 8.2.3)**. Connaissant la valeur du résidu on peut mettre à jour R_θ et q_θ en ce point conformément à la définition 17 p.57.

[2] Plus précisément au supremum des valuations entre $imRelief(v)$ et $imRelief(p)$, pour permettre l'imposition des marqueurs *à la volée*

[3] Nous pourrons juste rajouter ici une considération algorithmique : en fonction de la structure de donnée sous-jacente, il pourra être intéressant que le lac *le plus peuplé* absorbe l'autre.

III. Post-traitement lié a la contrainte 8.2.5 p.98 Ceci est la dernière phase de l'algorithme (l.75 à l. 83) elle est l'exacte transcription de la contrainte 8.2.5 p.98 et est liée au traitement du dernier niveau d'inondation.

On récupère le maximum de l'image "imRelief", on parcourt l'ensemble des lacs, pour chaque lac on récupère les points le constituant et pour chaque point on effectue (au besoin) la mise à jour de R_θ et q_θ.

8.4 Modifications des algorithmes précédents pour la prise en compte des accumulations :

Dans la Section 7.3 p.73, nous avions souligné la myopie de l'opérateur en présence de transitions graduelles. Nous avions proposé différentes stratégies pour résoudre partiellement cette myopie. Dans les sections suivantes nous allons décrire les approches algorithmiques permettant d'implémenter la première stratégie [4].

8.4.1 Algorithme basique

Un algorithme dérivé de 8.1 p.90 prenant en compte l'accumulation des résidus sur les zones de transitions graduelles est proposé en 8.10 p.112. Les modifications apportées se trouvent de la ligne 14 à 23 : tant que l'opérateur est actif en un pixel x, les résidus engendrés par les ouvertures successives sont accumulés dans l'image de travail $imAccumulate$; quand l'opérateur pause "temporairement" son activité (l. 17), l'accumulation en ce point est terminée, on utilise le résidu accumulé dans $imAccumulate(x)$ pour mettre à jour $R_\theta(x)$ (l. 19). **Attention** pour la mise à jour de $q_\theta(x)$ on ne prend pas l'indice courant d'ouverture mais le dernier indice pour lequel l'opérateur a eu une activité sur x (l. 20) ; enfin on réinitialise l'image d'accumulation en ce point (l. 23) dans l'attente de nouveaux résidus.

Nous allons proposer par la suite un algorithme efficace dérivant comme précédemment de l'algorithme d'inondation.

8.4.2 Réflexion pour la conception d'un algorithme efficace prenant en compte l'accumulation

Nous allons illustrer comment prendre en compte l'accumulation le long du processus d'inondation. Les contraintes 8.2.1, 8.2.2, 8.2.3, 8.2.4, 8.2.5, 8.2.6 présentées précédemment restent valides dans ce cas.

Exemple de fermeture ultime surfacique avec accumulation : intéressons-nous au profil de ligne présent sur la Figure 8.10. Nous allons inonder cette image et montrer comment calculer la fermeture ultime surfacique avec accumulation. Par rapport à l'implémentation sans accumulation nous allons avoir besoin de conserver au cours de l'inondation deux nouvelles informations contenues dans les images I_Σ et I_{indice} qui conserveront respectivement en tout point les résidus accumulés, et la valeur associée au critère.

On parcourt les minima de l'image (Étape 0) et on calcule pour chacun leur surface. On renseigne l'image I_{indice} ($I_{indice}\{b\} = 1$).

Au niveau d'inondation trois (Étape 1), le lac $L_1 = \{b\}$ grandit $L_1 = \{b, c\}$. C'est pourquoi le premier résidu apparaît en b : Résidu$\{b\}$ = Niveau Courant − Niveau Précédent $= 3 - 1 = 2$. La surface actuelle du lac est $\mu_{L_1} = 2$; au point b, on ne sait pas si l'on est sur une zone de transition graduelle. On doit donc conserver cette valeur dans I_Σ et poursuivre l'inondation. On conserve la mesure du critère de L_1,$\mu_{L_1} = 2$ dans $I_{indice}\{b, c\} = 2$.

Quand le niveau d'inondation monte à cinq (Étape 2), le lac L_1 grandit et des résidus non nuls apparaissent en b et c. Plus précisément lorsque L_1 incorpore le pixel d, μ_{L_1} devient égale à 3 or $I_{indice}\{b, c\} = 2$ à l'étape 1. On ne sait toujours pas si l'on a fini de parcourir la zone de transition

[4] Nous reviendrons sur le cas de la deuxième stratégie dans la conclusion de ce chapitre

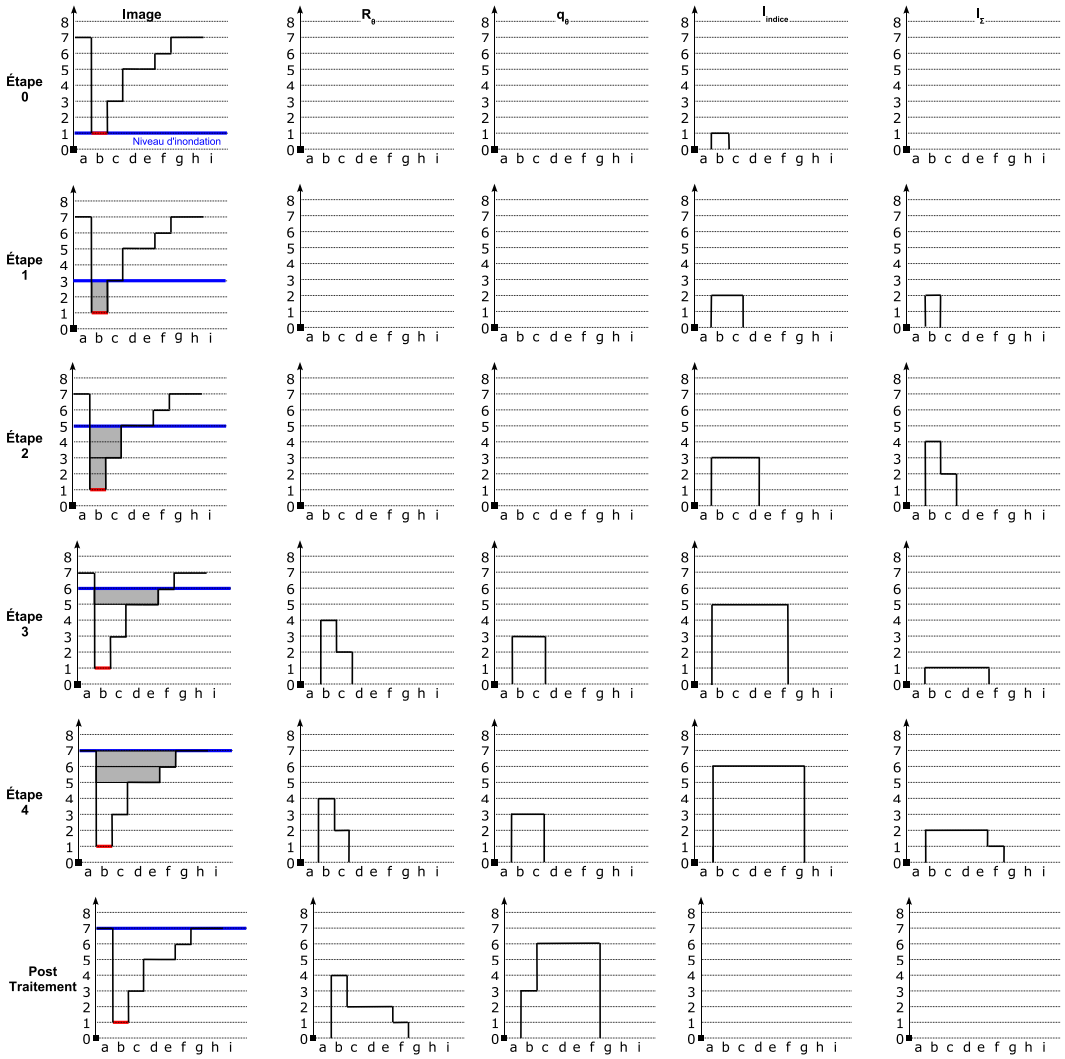


FIG. 8.10: Fermeture ultime surfacique (avec un pas d'ouverture unitaire) par inondation prenant en compte l'accumulation. De gauche à droite l'image inondée, R_θ, q_θ et les deux images auxiliaires permettant la prise en compte de l'accumulation I_Σ, I_{indice}. De haut en bas les différentes étapes de l'inondation (avec sur la première colonne : en trait gras le niveau d'inondation, et en grisé les structures de l'image pouvant participer à l'accumulation).

Algorithme 8.10 : Implantation générique de l'Ouverture Ultime Avec Accumulation

Data : $imRelief$ Image à traiter, $step$ (pas des tailles d'ouverture ; default=1), $stop$ (taille d'ouverture maximale)
Result : R_θ (image *Transformée*), q_θ (image *Indicatrice*)

```
1  begin
2      • Initialisation :
3      R_θ ⟵ 0
4      q_θ ⟵ 0
5      imOpenCurrent ⟵ 0 //Ouverture Courante
6      imResCurrent ⟵ 0 //Image des résidus courants
7      imOpenPrec ⟵ imRelief //Ouverture Précédente
8      imAccumulate ⟵ 0 //Image d'accumulation des résidus
9      λ ⟵ 0 //Taille de l'ouverture courante
10     while λ<stop do
11         λ=λ+step
           //Calcul du résidu courant
           //Remplacer γ par la famille d'ouverture choisie
12         imOpenCurrent ⟵ γ(imRelief,λ)
13         imResCurrent ⟵ imOpenPrec − imOpenCurrent
14         foreach pixel x do
               //Si un résidu non nul apparaît en x, M(x) = 1,on doit accumuler sa
                 valeur.
15             if imResCurrent(x)>0 then
16                 imAccumulate(x) ⟵ imAccumulate(x) + imResCurrent(x)
17             else
                   //Si il n'y a pas de résidu en x, l'accumulation en ce point est
                     terminée. M(x) est passé de l'état 1 à l'état 0, on doit
                     potentiellement mettre à jour R_θ(x) et q_θ(x)
18                 if imAccumulate(x)>=R_θ(x) then
19                     if imAccumulate(x)>0 then
                           //Mise à jour des images Transformée et Indicatrice
20                         R_θ(x)⟵ imAccumulate(x)
                           //Le dernier résidu non nul a eu lieu pour une taille
                             d'ouverture de λ - step
21                         q_θ(x)⟵ λ - step
22
23                     imAccumulate(x) ⟵ 0 //Remise à zéro de l'image d'accumulation
24     imOpenPrec ⟵ imOpenCurrent
25 end
```

graduelle. On accumule donc le résidu dans I_Σ et on met à jour I_{indice} ($I_\Sigma\{b\} = 4$, $I_\Sigma\{c\} = 2$, $I_{indice}\{b, c, d\} = 3$). Lorsque L_1 incorpore le pixel e, la mesure du lac évolue $\mu_{L_1} = 4$ mais sans changement de niveau d'inondation, il n'y a donc pas de génération de résidu, $\{e\}$ est simplement

ajouté à la collection des pixels de L_1.

On passe ensuite au niveau six (Étape 3), un résidu $\iota = 1$ apparaît pour les points b, c, d, e et L_1 incorpore le point f ($L_1 = \{b, c, d, e, f\}$). Concernant les points d, e, on ne peut pour l'instant rien conclure (i.e on ne sait pas si le point e est sur une zone de transition progressive, on conserve donc la valeur ι dans $I_\Sigma\{d, e\} = 1$). Par contre, **on peut détecter ici la fin de l'accumulation pour les pixels b et c : la surface du lac est de $\mu_{L_1} = 5$ alors que $I_{indice}\{b, c\} = 3$ à l'étape 2, on détecte que la mesure du critère a évolué de plus de un entre les deux étapes d'inondation**. R_θ en b et c et ι sont inférieurs au résidu accumulé dans I_Σ on peut donc valuer $R_\theta\{b\} = I_\Sigma\{b\} = 4$, $q_\theta\{b\} = I_{indice}\{b\} = 3$ et $R_\theta\{c\} = I_\Sigma\{c\} = 2$, $q_\theta\{c\} = I_{indice}\{c\} = 3$. **Et on conserve ι pour b, c dans I_Σ car il pourra contribuer à une prochaine accumulation**.

Au niveau sept (Étape 4), un résidu de $\iota = 1$ apparaît pour les points b, c, d, e, f. On ne sait pas si la zone de transition est terminée, on accumule dans I_Σ.

On est arrivé au maximum de l'image, l'inondation est terminée, il reste des résidus accumulés dans I_Σ. Pour leur prise en compte on doit faire une étape de post-traitement. On re-parcourt l'ensemble des pixels p du lac si la valuation de $I_\Sigma\{p\}$ est supérieure ou égale à $R_\theta\{p\}$, on value $R_\theta\{p\} = I_\Sigma\{p\}$ et $q_\theta\{p\} = I_{indice}\{p\}$.

Nouvelles contraintes : L'exemple précédent nous a donné les modifications à apporter à l'algorithme 8.3 pour la prise en compte des transitions progressives.

Nous avons deux contraintes supplémentaires par rapport à l'implantation sans accumulation :

Contrainte 8.4.1 *Conservation des résidus et de leur indice*

On doit pouvoir conserver pour tout point l'accumulation de ses résidus en cours ***et les indices correspondants***.

Contrainte 8.4.2 *Valeurs de résidus accumulées et non traitées*

A la fin de l'inondation ou si l'on arrête l'algorithme, on peut toujours avoir des points pour lesquels des résidus sont accumulés. On devra donc mettre en place un post-traitement (prenant en compte ces accumulations) pour valuer correctement R_θ et q_θ. Cette prise en compte des accumulations devra également être couplée au post-traitement 8.2.5.

Les sections ci-après expliciteront les modifications à apporter à l'algorithme pour la prise en compte des transitions.

8.5 Détails de l'implémentation proposée :

L'implémentation est détaillée sous forme d'un pseudo-code (algorithme.12 en p.115), comme précédemment elle sera complétée par une description exhaustive proposée en Section 8.5.2 p. 117. Celle-ci tendra le plus possible à relier le code proposé avec les différentes contraintes exposées au long de la Section 8.4.2.

Pour prendre en compte l'accumulation nous devons définir deux images de travail supplémentaires qui permettront de tester les contraintes 8.4.1 et 8.4.2 :

- I_Σ : image contenant en chaque point l'accumulation des résidus
- I_{indice} : image contenant en chaque point d'un lac la mesure du critère associé à ce lac.

L'algorithme est une extension du précédent avec quelques ajustements, aussi nous ne décrirons que ceux-ci. Cependant le lecteur devra être attentif aux subtilités introduites lors du calcul des résidus car c'est là que repose toute la mécanique explicitée dans la Section 8.4.2.

Nous attirons l'attention du lecteur sur un dernier point avant de commencer la description :

Remarque 10 *L'algorithme décrit ne prend en compte que des transitions progressives unitaires, mais son extension à d'autres types de transitions est triviale.*

Nous reprenons le découpage en trois grandes étapes, en détaillant les modifications de chacune d'elles.

8.5.1 Pseudo-code

Algorithme 8.11 : Algorithme de fermeture ultime par critère avec accumulation

Data : imRelief, imLabels, ValeurdeStop,$\mathcal{N}$ (Voisinage)
Result : R_θ (image *Transformée*), q_θ (image *Indicatrice*)

1 • ***Initialisation :*** **:**
`// Idem à l'algorithme 8.3.2 :initialisation`
2 $I_\Sigma \leftarrow 0$
3 $I_{indice} \leftarrow 0$
`// On a fini de parcourir les minima de l'image, on doit initialiser l'image des indices d'accumulation.`
4 **forall** *pixels p de imLabels* **do**
5 | **if** $p \neq$ *label-fond* **then**
6 | | I_{indice}(p) $\leftarrow$ ValeurPrecCritere[p]
7 • ***Propagation*** **:**
`// Idem à l'algorithme 8.3 :propagation`
`// On modifie le test de la l.57 pour détecter l'arrêt de l'algorithme. En effet, lorsque que l'on arrête l'algorithme on peut toujours avoir des résidus accumulés dans` I_Σ `qu'il faut traiter`
8 **if** *(ValeurCritereFusion* $\geq$ *ValeurCritere1**+1** et ValeurCritere1**+1*** $<$ *MaxCriterion et prio-ws* $>$ *NiveauCourantLac*$[$*Toplabel1*$]$*)* ***ou (ValeurCritereFusion*** $\geq$ ***ValeurCritere1+1 et ValeurCritere1+1*** $=$ ***MaxCriterion)*** **then**
9 | ...
10
11 Répéter les lignes 8 à 9 pour le lac 2 (pointé par Label2)
12 • ***PostTraitement*** **:**
`// Deux configurations possibles`
13 Goto 30

MiseAJourResidus(prio-ws,Label,ValeurDuCritere)

```
14 ListePixelLac ← TabDynLac[Label] // Liste des pixels du lac pointé par label
15 foreach point p dans ListePixelLac do
16     résidu ← prio-ws - p // Résidu courant en p
17     p ← prio-ws // fermeture à la hauteur de traitement voir contrainte 8.2.3
       // On est sur une zone de transition ? Si oui on doit accumuler les
          résidus
18     if ValeurDuCritere = I_indice(p)+1 et résidu≠ 0 then
19         I_Σ(p) ← I_Σ(p) + résidu
20     else
           // On regarde si le résidu accumulé est > résidu courant
21         if I_Σ(p) > résidu then
22             if I_Σ(p) > R_θ(p) et I_Σ(p) >0 then
23                 R_θ(p) ← I_Σ(p)
                   // On doit prendre la mesure du critère précédent
24                 q_θ(p) ← I_indice(p)
25         else
               // On retombe sur le cas sans accumulation
26             if résidu ≥ R_θ(p) et résidu > 0 then
27                 R_θ(p) ← résidu ;q_θ(p) ← ValeurDuCritere
           // On conserve le résidu courant dans l'image d'accumulation
28         I_Σ(p) ← résidu
       // On propage la valeur du critère dans I_indice
29     I_indice(p) ← ValeurDuCritere
```

Post-Traitement()

```
   // Contrainte 8.4.2 :Prise en compte de l'accumulation
30 if MaxCriterion≥ Valeur maximale admissible par ImRelief then
31     Max ← Valeur maximale de ImRelief
32     ValeurDuCritere ← Valeur maximale admissible par ImRelief
33     foreach List dans TabDynLac do
34         foreach point p dans List do
35             résidu ← Max - p // Résidu courant en p
36             p ← Max // fermeture à la hauteur de traitement
               // On regarde si le résidu accumulé est > résidu courant
37             if I_Σ(p) > résidu then
38                 if I_Σ(p) ≥ R_θ(p) et I_Σ(p) >0 then
39                     R_θ(p) ← I_Σ(p)
                       // On doit prendre la mesure du critère précédent
40                     q_θ(p) ← I_indice(p)
41             else
42                 if (résidu ≥ R_θ(p) et résidu > 0) then
43                     R_θ(p) ← résidu ;q_θ(p) ← ValeurDuCritere
44 else
45     foreach List dans TabDynLac do
46         foreach point p dans List do
               // Arrêt de l'algorithme : des résidus accumulés peuvent subsister
47             if I_Σ(p) > R_θ(p) et I_Σ(p)> 0 then
48                 R_θ(p) ← I_Σ(p) ;q_θ(p) ← I_indice(p)
```

8.5.2 Explication détaillée du pseudo-code précédent

Nous reprenons le découpage en trois grandes étapes, en détaillant les modifications de chacune d'elles.

I. Initialisation (l.1 à l.6 p.115) : on réalise dans un premier temps la même initialisation que précédemment. On y ajoute la prise en compte des images I_Σ et I_{indice} : il n'y a pas eu encore de résidu engendré donc I_Σ est initialisé à zéro, I_{indice} est également dans un premier temps initialisé à zéro, **puis pour chaque label (ici associé à un minimum) on lui assigne la valeur courante du critère** (voir l'exemple en 8.4.2 p.110).

II. Propagation (l.7 à l.11 p.115) : concernant la propagation proprement dite les seules modifications par rapport à l'algorithme précédent sont liées à la gestion de l'arrêt de l'algorithme en cours d'inondation. Il n'y a pas de modification à apporter lors de la croissance d'un lac donné **mais lors de l'étape de fusion**. Comme précédemment nous commençons par déterminer la valeur du critère qui sera associée au lac fusionné. Ceci nous permet d'évaluer les contraintes 8.2.1 et 8.2.2 pour chacun

des lacs en cours de fusion **auquel nous ajoutons le test d'arrêt de l'algorithme** (voir l.8). En effet des résidus peuvent être en cours d'accumulation pour ce lac et nous devons les traiter dès qu'un lac atteint le critère d'arrêt de l'algorithme. Le reste du processus de fusion se déroule comme dans le premier algorithme.

L'ensemble de la mécanique des accumulations a été *factorisé* dans la fonction ad-hoc de calcul des résidus, elle reprend la mécanique explicitée en 8.4.2 p.110 .

Fonction ad-hoc de calcul des résidus pour un lac donné (l.13) : comme précédemment pour calculer les résidus pour un lac donné, on doit connaître la hauteur topographique de fermeture (variable "val"), le lac que l'on souhaite fermer (*indexé* par la variable "label"), et la valeur de fermeture par critère à laquelle cela correspond (variable "ValeurDuCritère"). Dans le cas de l'accumulation, on doit connaître en plus la valeur du critère associée à chaque lac en tout point de celui-ci, cette information est véhiculée par I_{indice}.

On récupère l'ensemble des points du lac pointé par "label" (l.15). Pour chaque point "p", on calcule le résidu courant et on assigne à "p" la hauteur de fermeture (voir contrainte 8.2.3). Si un résidu non nul est engendré, deux cas de figure peuvent se présenter :

- **Accumulation en "p"** La valeur de fermeture ("ValeurDuCritère") est égale à la valeur précédente du critère plus un, **on se trouve sur une transition unitaire** (l.18). Le résidu courant est accumulé dans I_{Σ}.
- **Fin d'accumulation en "p"** si l'on ne se trouve pas sur une transition, on a fini l'accumulation des résidus en "p". On teste qui, du résidu courant et du résidu accumulé, est prédominant et on met à jour R_θ et q_θ en fonction (l.20 à l.27). Et dans les deux cas **le résidu courant est placé dans** I_{Σ} (l.28), car il pourra participer à une accumulation ultérieure.

Enfin quelle que soit la configuration, résidu nul/non-nul **on met à jour** I_{indice} **à la valeur du critère courant**. Ceci est indispensable pour détecter si un point doit être accumulé (cf test l.18).

III. Post-traitements (l.12 p.115) : pour cet algorithme, non pas un, mais **deux** cas de figure peuvent se présenter, le premier est une adaptation du post-traitement de l'algorithme sans accumulation, le second correspond à la gestion de l'arrêt en cours d'inondation. Ces adaptations découlent toutes du même problème, il nous manque un niveau d'inondation *virtuel* pour prendre en compte les dernières accumulations.

III.a Post-traitement lié au traitement du dernier niveau d'inondation (l.30 à .l43 p.117) : comme précédemment on calcule en chaque point le résidu associé à la fermeture au maximum de l'image $imRelief$, mais on doit également prendre en compte le fait que des résidus sont toujours accumulés (test l.37). On met à jour R_θ et q_θ en fonction, conformément à la définition 17 p.57.

III.b Post-traitement lié à l'arrêt de l'algorithme en cours d'inondation (l.44 à .l48 p.117) : si l'on a arrêté l'algorithme en cours d'inondation, des résidus peuvent être en cours d'accumulation. On force ici leur prise en compte conformément à la définition 17 p.57

8.6 Tests de performance et complexité

Pour mesurer les performances des algorithmes 8.3.2 et 8.5.1 nous allons reprendre les protocoles expérimentaux définis en [57]. Ceux-ci ont été testés sur une collection d'images synthétiques de

tailles et de nombres de minima variables. Les temps indiqués sont une moyenne sur un certain nombre de réalisations, calculés à l'aide d'un ordinateur de bureau [5].

8.6.1 Images Synthétiques

Génération des images de tests Pour tester la sensibilité des algorithmes vis à vis de la complexité de l'image, certaines images synthétiques de tailles et de nombres de minima variables ont été générées. Elles sont calculées en plaçant aléatoirement N_{min} points sur un fond noir, puis en calculant la carte de distance euclidienne associée.

Taille d'image fixe et variation du critère d'arrêt de l'algorithme On procède tout d'abord à l'étude de la dépendance des algorithmes au seuil d'arrêt. Pour cela, on va réaliser des fermetures ultimes utilisant les critères de surface et de hauteur sur des cartes de distance de taille 512x512 en faisant varier le seuil d'arrêt λ et le nombre de minima de l'image M. L'ensemble des résultats est fourni sur la Figure 8.11.

On peut observer que pour de faibles valeurs de λ, un nombre plus élevé de minima correspond à un temps de calcul plus important. Mais passé un certain seuil, c'est sur l'image ne comprenant que deux minima que les temps de calcul augmentent rapidement vers une tendance quadratique. Dans le cas présent la densité de minima conditionne le nombre de valeur de gris K de l'image. En effet, pour une même taille d'image, l'accroissement de M est proportionnel à N_{min} et le nombre de teintes de gris K lui est inversement proportionnel. Or au cours du processus d'inondation, on ne calcule les résidus que pour les lacs qui, ayant changé de mesure, changent aussi de niveau d'inondation. Ceci explique la forte dépendance de l'algorithme au nombre de valeur gris présent dans l'image.

On peut également observer que le coût engendré par la prise en compte des accumulations, bien que non nul, ne pénalise pas la «tendance» observée.

Dépendance de l'algorithme à la taille de l'image Pour mesurer la dépendance des algorithmes vis à vis de la taille de l'image N, des images synthétiques de taille croissante sont générées en utilisant la même densité α de minima par nombres de points de l'image ($\alpha = N_{min}/N = constante$). Le seuil d'arrêt de l'algorithme est toujours fixé à sa valeur maximale. Les résultats sont présentés sur la Figure 8.13.

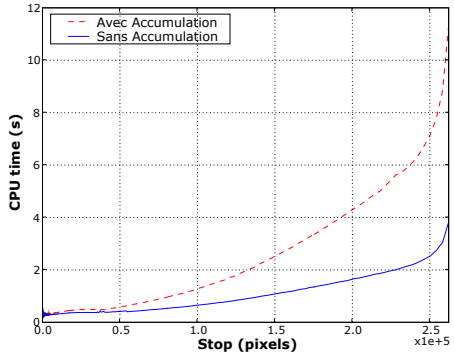
On peut remarquer que, quel que soit l'algorithme, on observe une dépendance proportionnelle à l'accroissement de la taille de l'image.

Comportement sur un jeu d'images réelles Pour appréhender les temps de calcul dans des conditions plus réalistes, une sélection des 38 images de taille 1600x1200 a été extraite de la base $ICDAR$. Pour chacune nous extrayons le gradient de luminance et mesurons les temps de calcul des fermetures ultimes pour les critères de hauteur et d'aire et ceci en faisant varier le critère d'arrêt de l'algorithme. A titre de comparaison, la même procédure est mise en oeuvre pour des fermetures par critères «simples» (l'algorithme utilisé est issu de [105]). La figure Figure 8.12 présente les temps moyennés sur les 38 images.

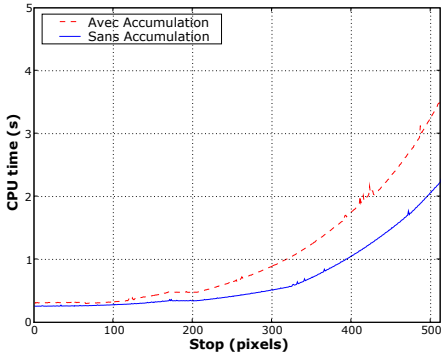
On peut observer que les temps d'exécution croissent avec la valeur d'arrêt de l'algorithme mais que cette croissance n'est pas quadratique comme on aurait pu le craindre.

A la lumière des exemples proposés, il semble que l'approche proposée dépende majoritairement du nombre de points de l'image N et du nombre de teintes de gris K. Ci-après nous nuancerons cette

[5] Pour information Intel© Pentium© 4 2.4Ghz, nombre de réalisations :10

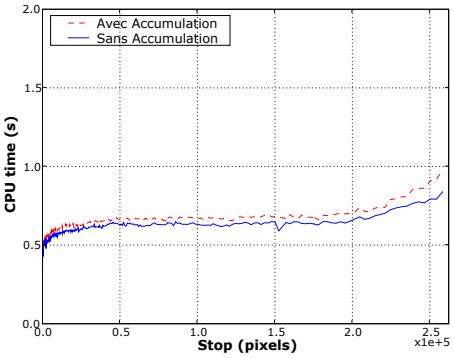


(a) Critère surfacique

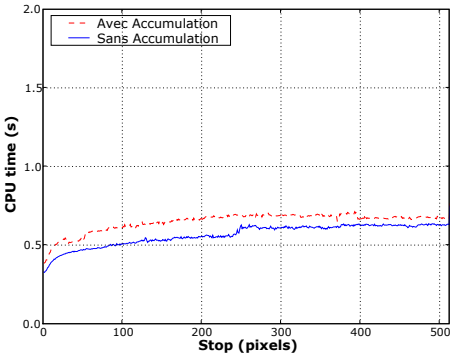


(b) Critère de hauteur

Temps de calcul : image 512x512, 2 minima, 457 niveaux de gris



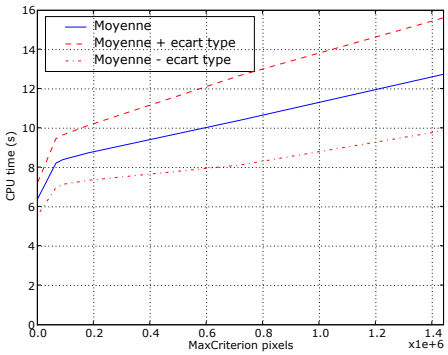
(c) Critère surfacique



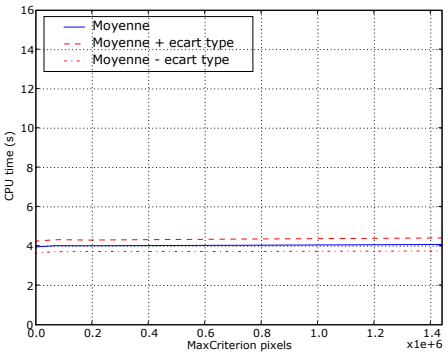
(d) Critère de hauteur

Temps de calcul : image 512x512, 2000 minima, 22 niveaux de gris

FIG. 8.11: Temps de calcul (images 512x512) des algorithmes en fonction du seuil d'arrêt sur les images synthétiques. Les temps des algorithmes avec et sans prise en compte de l'accumulation sont respectivement en pointillé et trait plein. Les figures a) et b) montrent le temps de calcul pour les critères de surface et de hauteur pour une image comprenant 2 minima et 457 teintes de gris. Les figures c) et d) pour une image comprenant 2000 minima et 22 niveaux de gris.

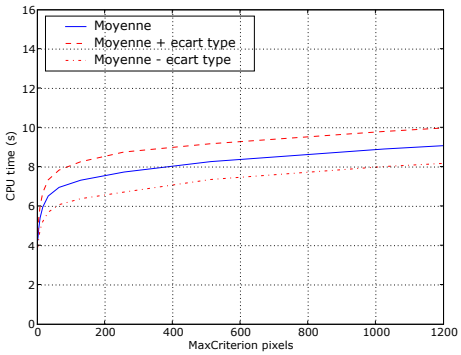


(a) Fermeture Ultime

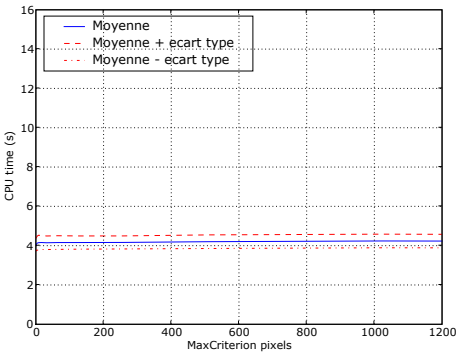


(b) Fermeture

Temps de calcul : image 1600x1200, Critère Surfacique



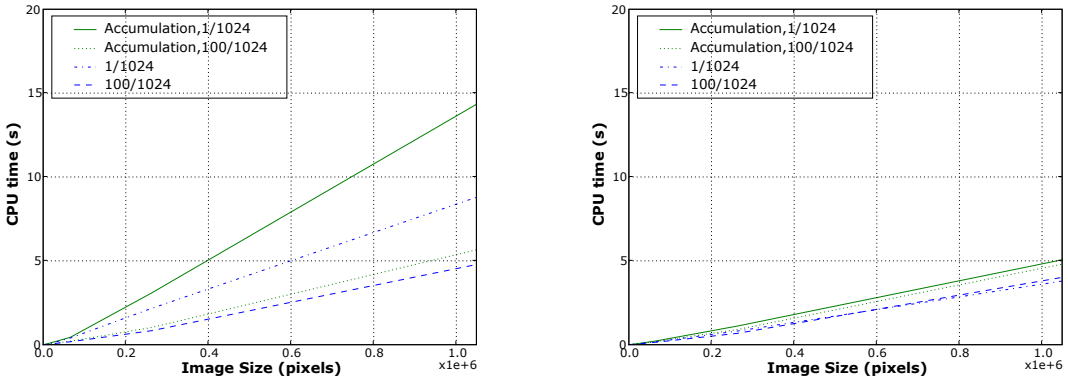
(c) Fermeture Ultime



(d) Fermeture

Temps de calcul : image 1600x1200, Critère de Hauteur

FIG. 8.12: Comparaison des temps de calcul des algorithmes fermetures ultimes et de fermetures par critère en fonction du seuil d'arrêt sur des images réelles. Le temps est moyenné sur 38 images (1600x1200) de la Base ICDAR et le calcul est effectué sur le gradient de luminance.



(a) Critère surfacique (b) Critère de hauteur

FIG. 8.13: Temps de calcul des algorithmes en fonction de la taille des images pour une même densité de minima. Le seuil d'arrêt de l'algorithme est toujours fixé à sa valeur maximale.

première analyse en discutant plus en profondeur les interdépendances entre N,M,K et le critère utilisé.

8.6.2 Brève réflexion sur la complexité théorique :

La complexité pour le «cas moyen» et le «cas le plus défavorables» des algorithmes 8.3.2 et 8.5.1 est difficile à déterminer. En effet, nous devons calculer des «complexités amorties» car le «pire des cas» de chaque sous-fonction ne peut pas apparaître à notre connaissance au sein d'un même exemple. Nous proposons par la suite quelques pistes de réflexion pour le calcul d'une borne algorithmique théorique.

Hypothèse pour le calcul d'une borne inférieure : M est relié à K et n'induit pas de coût «propre» L'étude sur les images synthétiques nous a permis de remarquer que le nombre de minima M *ne semble pas avoir* d'influence importante sur l'efficacité de l'algorithme. Rappelons également que la recherche des minima n'est pas incluse dans l'algorithme même, ce qui n'est pas le cas de l'algorithme de fermeture par critère proposé par J.Breen and Jones [34]. Si nous restreignons les paramètres à N le nombre de points de l'image, K le nombre de teintes de gris et λ la valeur d'arrêt de l'algorithme, nous pouvons rapidement proposer un cas pathologique : dans le cas du critère surfacique et en prenant comme valeur d'arrêt λ la plus grande valeur admissible (i.e. pour une image I, $\lambda = Surface(I)$), nous convergeons vers un algorithme qui ne dépend plus que de N et K. Si nous supposons que le coût associé au processus d'inondation par file d'attente hiérarchique est en O(N) (voir [101, 76]), le cas le plus défavorable intervient pour une configuration très simple présentée sur la Figure 8.14.

Si l'on suit le processus d'inondation, pour chaque changement de niveau k , le lac modifie la mesure de son critère. On doit donc mettre à jour R_θ et q_θ. Ainsi lors du passage du niveau $k_2 = K-2$ à $k_1 = K-1$, on devra calculer les résidus pour l'ensemble des pixels du niveau k_2 ce qui représente

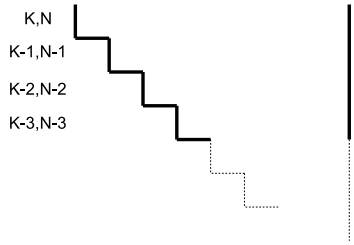


FIG. 8.14: Configuration pour laquelle les algorithmes 8.3.2 et 8.5.1 sont quadratiques

$N-2$ pixels. Si l'on développe ce schéma pour chaque niveau, on peut voir que l'on aura besoin de $K \bullet (N+1)/2$ opérations pour calculer le résultat de la fermeture ultime surfacique.

On peut donc résumer le coût de l'algorithme comme la somme du coût de l'inondation et du coût de traitement du calcul de R_θ et q_θ : $O(N) + O(K \bullet N)$, que l'on peut simplifier dans le pire des cas où $K = N$ en une complexité en $O(N^2)$.

Remise en cause de l'hypothèse Cependant il est faux de supposer que le nombre de minima M ne rentre pas «intrinsèquement» en ligne de compte. Car

1. M conditionne le nombre d'opérations de «fusion» à prendre en compte, «fusion» qui nécessite dans notre cas de maintenir une table d'équivalence (cf Procédure 9 p.105).
2. Lors de l'assignation d'un pixel à un lac (cf étape de propagation), nous devons retrouver à partir du label associé au pixel son lac représentant. Et donc parcourir la table d'équivalence.

Notons qu'il existe une solution optimale à cette gestion. C'est l'utilisation d'ensembles disjoints associés à l'algorithme de Tarjan [94]. La borne amortie de chaque opération de base[6] de cet algorithme pour N éléments est en $O(\alpha(N))$ (avec α, la fonction inverse d'Ackermann).

Aussi le coût de maintien de la table d'équivalence dépendra à la fois du nombre de points de l'image N et du nombre de minima M.

Ceci clôt, nos réflexions sur le calcul d'une borne à nos algorithmes et nous observons que cette borne est complexe à déterminer : elle dépendra de N, K et M **et de leur interdépendance**.

Complexité et dépendance du critère utilisé Enfin, un dernier point et non des moindres est la complexité associée au critère utilisé. Nous n'avons utilisé jusqu'à présent que des critères pour lesquels la mise à jour de leur valeur associée peut se faire en temps constant. Pour d'autres critères, il faudra prendre en compte leur complexité intrinsèque $O(\mu)$. Ceci est très pénalisant pour notre approche, car $O(\mu)$ intervient chaque fois qu'un point est associé à un bassin versant.

Conclusion Il serait certes intéressant de disposer d'une réelle étude théorique de la complexité de notre algorithme. Cependant nous pensons que des approches concurrentes utilisant des approches dites *d'arbre de composantes* seraient plus pertinentes. Nous y reviendrons dans les perspectives de ce chapitre.

[6] Sous l'angle de notre algorithme les opérations de base sont : *MakeSet*-un label est son propre représentant et est associé à un bassin versant, *Find*-étant donné un label, retrouver le lac représentant, *Union*-fusion de deux lacs, un des deux est choisi pour être le nouveau représentant.

8.7 Conclusion

Dans ce chapitre nous avons proposé une première implémentation «efficace» de l'opérateur d'ouverture ultime dérivant d'un algorithme d'inondation. Bien que l'algorithme semble avoir une complexité «sup-quadratique» dans le cas le plus défavorable, les tests effectuées sur des jeux d'images réelles indiquent que l'approche proposée est satisfaisante en l'état. Elle pourra servir d'implémentation de référence pour de possibles confrontations avec d'autres approches (cf Perspectives). Les implantations proposées (avec ou sans prise en compte des transitions graduelles), nous permettent d'utiliser maintenant cet opérateur *en routine*. Ce qui était le but premier de ce travail algorithmique.

8.8 Perspectives

Nous avons proposé deux algorithmes dont la complexité dans le cas le plus défavorable est probablement sup-quadratique. De plus cette complexité doit être pondérée par celle associée au calcul du critère. Or notre implantation souffre ici d'un inconvénient majeur car on doit recalculer la valeur associée à un critère pour chaque nouveau point. Aussi, la courbe de complexité de notre algorithme suivra également celle du critère choisi.

Une remarque plus générale concerne le nombre de fois que l'on doit re-parcourir l'ensemble des points d'un lac pour mettre à jour des résidus éventuels. Notre algorithme est aveugle pour un niveau d'inondation courant au niveau d'inondation suivant, aussi on peut parcourir de nombreuses fois un même lac sans raison autre que cet aveuglement.

Enfin nous avons mentionné qu'il y avait un minimum de deux alternatives à l'accumulation des résidus sur les zones de transition, l'implantation actuelle n'en propose qu'une et le support de la seconde serait complexe à mettre en oeuvre.

Les trois cas explicités précédemment nous poussent aujourd'hui à nous tourner vers des approches dites *d'arbre de composantes* (voir les travaux de Najman and Couprie [63] et Berger et al. [6] pour le "component-tree" et Salembier et al. [78] pour le "max-tree"). Celles-ci permettront de minimiser les coûts de parcours et de ne calculer les valeurs associées aux critères qu'en certains points clés. Ces approches nous permettront donc de réduire les coûts algorithmiques actuels et d'étendre les opérateurs présentés à des critères plus complexes que ceux explicités dans ce manuscrit, sans le payer par une augmentation trop importante de la complexité globale. Enfin, elles permettront d'intégrer «facilement» la prise en compte de hiérarchies de résidus.

Il nous semble que c'est finalement l'approche la plus naturelle et *élégante* à la résolution des problématiques algorithmiques que nous avons rencontrées.

Conclusion

Dans les chapitres précédents nous avons décrit l'opérateur d'ouverture ultime. Nous avons mis en lumière certaines de ses limitations et proposé différentes stratégies pour y pallier.

Ensuite nous en avons proposé une implémentation *efficace*, étape indispensable pour utiliser l'opérateur sous forme de *routine* algorithmique.

Cet opérateur véhicule deux types d'informations complémentaires : il permet de connaître en chaque point de l'image la taille de la structure dominante représentée par l'image indicatrice ainsi que son contraste représenté par l'image transformée.

La partie suivante traitera de l'utilisation de ces informations pour la création de chaînes de localisation de texte.

Chaînes de traitements et résultats

Nous avons acquis au travers de la partie précédente une bonne connaissance du comportement de l'opérateur d'ouverture ultime. La connaissance de ses forces et faiblesses nous permet maintenant de réfléchir à son intégration dans des chaînes globales de localisation de texte.

Dans le chapitre 9, nous utiliserons sa faculté à proposer des pré-segmentations satisfaisantes, pour la création de chaînes de traitement, nos travaux rejoindront ainsi la famille des approches par composantes connexes.

Les chaînes décrites seront ensuite évaluées dans le chapitre 10 sur deux projets : 1. la campagne d'évaluation d'algorithmes de traitement d'images $imagEval$; 2. la campagne d'évaluation ICDAR(International Conference on Document Analysis and Recognition), qui permettra de situer nos approches par rapport à l'état de l'art.

Notons que nos chaînes ont été intégrées sur un troisième projet : la réalisation d'un démonstrateur pour EADS sur le classement automatique d'images relatives à l'industrie aéronautique ou de défense. Pour des raisons de confidentialité, nous ne pourrons produire ici d'exemples ou de résultats de ce projet.

9 Chaînes de traitement

> Il faut supporter aussi bien que possible le lot que la destinée nous assigne et savoir qu'on ne peut lutter contre la force de la nécessité.
>
> ESCHYLE

Nous allons proposer dans ce chapitre différentes chaînes de localisation de texte. Elles reposeront chacune sur une première phase utilisant les transformations résiduelles présentées dans les chapitres précédents. Comme nous le verrons dans le chapitre 10 nous avons traité différentes bases d'images tant dans des domaines académiques qu'industriels.

Ces bases n'ont pas toutes les mêmes propriétés intrinsèques et la recherche de performance (en terme de qualité de localisation) nous a amené à développer différentes stratégies. Ces stratégies dépendent en premier lieu de l'utilisation des informations véhiculées par l'ouvert ultime i.e on pourra se baser sur l'image indicatrice, l'image transformée ou une combinaison des deux.

Quelle que soit la stratégie utilisée notre premier résultat se résume à une première segmentation de l'image comprenant les lettres mais aussi pléthore de composantes parasites. Nous devons donc greffer des modules de sélection/filtrage pour réduire le nombre de non lettres, puis procéder à une étape d'agrégation des composantes pour former nos résultats de localisation.

La plupart seront paramétriques, aussi nous tenterons dans la mesure du possible de les justifier et de quantifier leur apports. On soulignera, ainsi, l'interdépendance des modules et la difficulté de proposer des métriques intermédiaires, et par là-même, le risque d'optimiser un module donné sans prendre en compte l'optimisation globale *d'un système.*

Avant de rentrer dans le vif du sujet, nous proposons dans la Section 9.1 de revenir sur les différentes hypothèses que nous avons adoptées.

9.1 Hypothèses

Dans les bases que nous allons traiter, les zones de texte à localiser sont très variées. Ceci restreint de manière drastique le nombre des hypothèses sur lesquelles nous pouvons nous appuyer pour

bâtir un système de localisation de texte pour l'ensemble des bases. Nous rappelons dans la suite du paragraphe celles retenues dans notre cas [1].

CONTRASTE : La plupart des textes en surimpression ou enfouis sont *logiquement* définis pour être lisibles pour un observateur humain. On se basera donc sur un contraste significatif du texte vis-à-vis de son fond propre.

CONTRAINTES GÉOMÉTRIQUES :

- Taille : Bien que la taille des caractères puisse être extrêmement variable, des caractères d'une même zone de texte possèdent des caractéristiques *comparables*.
- Alignement : On considérera que les caractères apparaissent en grappes. Typiquement on considérera qu'il faut un minimum de trois caractères pour définir une zone potentielle de texte. On ne pourra pas s'appuyer sur de trop fortes contraintes d'horizontalité des zones de texte : nombre de textes enfouis auront subi des déformations géométriques et/ou de perspectives. Les contraintes de regroupement sur l'alignement des caractères devront ainsi être relâchées.
- Distance inter-caractère : On considérera généralement que la distance inter-caractère pour un même mot ou une ligne de texte est proche d'une constante.

Types de textes non traités : Nous décrivons ci-après, les *types de textes* qui ne seront pas pris en compte par nos approches. Soit par choix de notre part, soit parce qu'ils ne peuvent tout simplement pas être traités par notre méthode. La Figure 9.1 résume en images quelques uns de ces cas.

- Zones de texte vertical : il est souvent admis que la recherche de textes verticaux (cf Lienhart [48]), conjointement aux *autres textes*, entraîne une baisse très importante de l'efficacité des systèmes de localisation. Le nombre de zones de texte vertical étant de plus marginal, aucun outil ne sera mis en place pour les prendre en compte.
- Texte Manuscrit : La recherche de texte manuscrit faisant l'objet de recherche spécifiques (séparés de la recherche d'autres types de textes), nous n'avons pas développé d'outils spécifiques pour ceux-ci.
- Lettres "attachées" : Il s'agit ici d'une limitation inhérente à notre approche. En effet, nos systèmes sont basés sur une première étape d'extraction de composantes connexes que nous filtrerons pour ne conserver que les lettres. Aussi, si lors de la première étape, nous ne récupérons pas des lettres dissociées, mais une grappe, tout concourra dans les modules que nous présenterons à classifier ce cas comme un faux positif.
- Fond *complexe* et résolution : Il s'agit une nouvelle fois de limitation inhérente à notre approche. Si nous ne pouvons extraire de composantes satisfaisantes lors de la première étape de segmentation(du fait de fond complexe, de texte *dégradé*,...), là encore nos stratégies échoueront.

9.2 Discussion sur les pré-filtrages possibles

Jusqu'à présent nous n'avons qu'appliqué l'opérateur. Le lecteur aura remarqué qu'il n'est jamais question de filtrage préalable. Pourtant les pré-filtrages sont la plupart du temps inhérents aux processus de segmentation morphologique, comme spécifié dans le paradigme de la segmentation([60]).

[1] Pour rappel le chapitre 3 p.17, expose les hypothèses les plus souvent usitées dans la littérature et ceci pour divers cas applicatifs

(a) Texte Manuscrit

(b) Caractères collés

(c) Textes Verticaux

(d) Fond non-coopératif

(e) Zoom Figure 9.1(d)

(f) Zoom Figure 9.1(b)

FIG. 9.1: Exemple type de texte non prise en compte par nos approches. Base $imagEval$

Certes, mais nous travaillons sur des bases d'images généralistes où la définition d'un processus de pré-filtrage commun à toutes les images est difficile à envisager. Ainsi on ne peut pas utiliser de filtres *forts*, car ils détruiront les lettres presque aussi vite qu'ils réduiront le bruit. En effet des lettres peuvent être présentes à toutes les *échelles*, il est difficile de concilier réduction du bruit, préservation des contours, préservation de la topologie (i.e. ne pas boucher des lettres présentant des trous caractéristiques). On se retrouve devant un problème classique du *Paradoxe de l'oeuf et de la poule*. Il faut une connaissance a priori sur la taille des structures d'intérêt pour définir des filtres appropriés.

Remarque 11 *Il est cependant bien évident que dans un cadre applicatif moins générique où les images présenteraient certaines caractéristiques communes (eg :taille de fontes minimales, a priori colorimétriques), il serait plus que souhaitable de faire appel à des pré-filtrages pour réduire le bruit et/ou renforcer les contours des structures d'intérêt.*

Nous verrons dans le chapitre 10, p.185, que les bases que nous avons eues à traiter ne permettaient guère d'hypothèses fortes sur les «propriétés» des images.

Présentation des chaînes de traitement : Nous venons de formaliser les hypothèses qui guideront nos développements ; nous pouvons passer à la description des chaînes de traitement. On trouvera sur la Figure 9.2, l'organigramme des chaînes nommée "Transformée"(ou "Approche Transformée") et "Indicatrice"(ou "Approche Indicatrice"). Elles utilisent toutes deux l'ouvert ultime mais selon l'image utilisée pour l'obtention de la première étape de segmentation.

Nous découperons la description des modules les constituant en trois grandes étapes :

1. Modules de l'Approche Transformée : il s'agit des outils nécessaires pour 1.segmenter l'image de *contraste* issue de l'ouvert ultime (cf Section 7.4, p.79) ; 2.effectuer une première passe de filtrage rapide des composantes non probables.
2. Modules de l'Approche indicatrice : si l'indicatrice est directement utilisable comme image presegmentée. Nous proposons également quelque filtrages simples pour diminuer sensiblement le nombre de composantes non probables.
3. Modules hybrides : il s'agit des modules de sélection des composantes par apprentissage et de regroupement itératif des composantes. Ils interviendront quelle que soit l'approche utilisée.

On accordera une attention toute particulière à leurs forces, faiblesses et inter-dépendances. Notons que nombre de remarques négatives se veulent constructives pour des travaux ultérieurs. Les scores présentés dans le chapitre 10, p. 185, valideront les approches proposées en l'état de nos travaux.

9.3 Modules de l'Approche Transformée

9.3.1 Seuillage "adaptatif" de l'image R_θ

La première étape utilisant l'opérateur d'ouverture ultime nous a permis d'assigner à différentes composantes de l'image, un contraste apparent (image Transformée R_θ), ainsi qu'une taille (image indicatrice q_θ). Il arrive que la segmentation obtenue au travers de l'indicatrice ne soit pas exploitable, ie les segmentations obtenues ne permettent pas de récupérer une composante connexe par lettre. Dans ce cas il est intéressant de regarder l'information véhiculée par l'image transformée.

Celle-ci peut ainsi être vue comme une image à niveaux de gris que l'on doit segmenter pour extraire les structures d'intérêts. Certes mais comment ? Nous pourrions nous tourner vers des méthodes de binarisation automatique utilisant l'histogramme de l'image, mais :

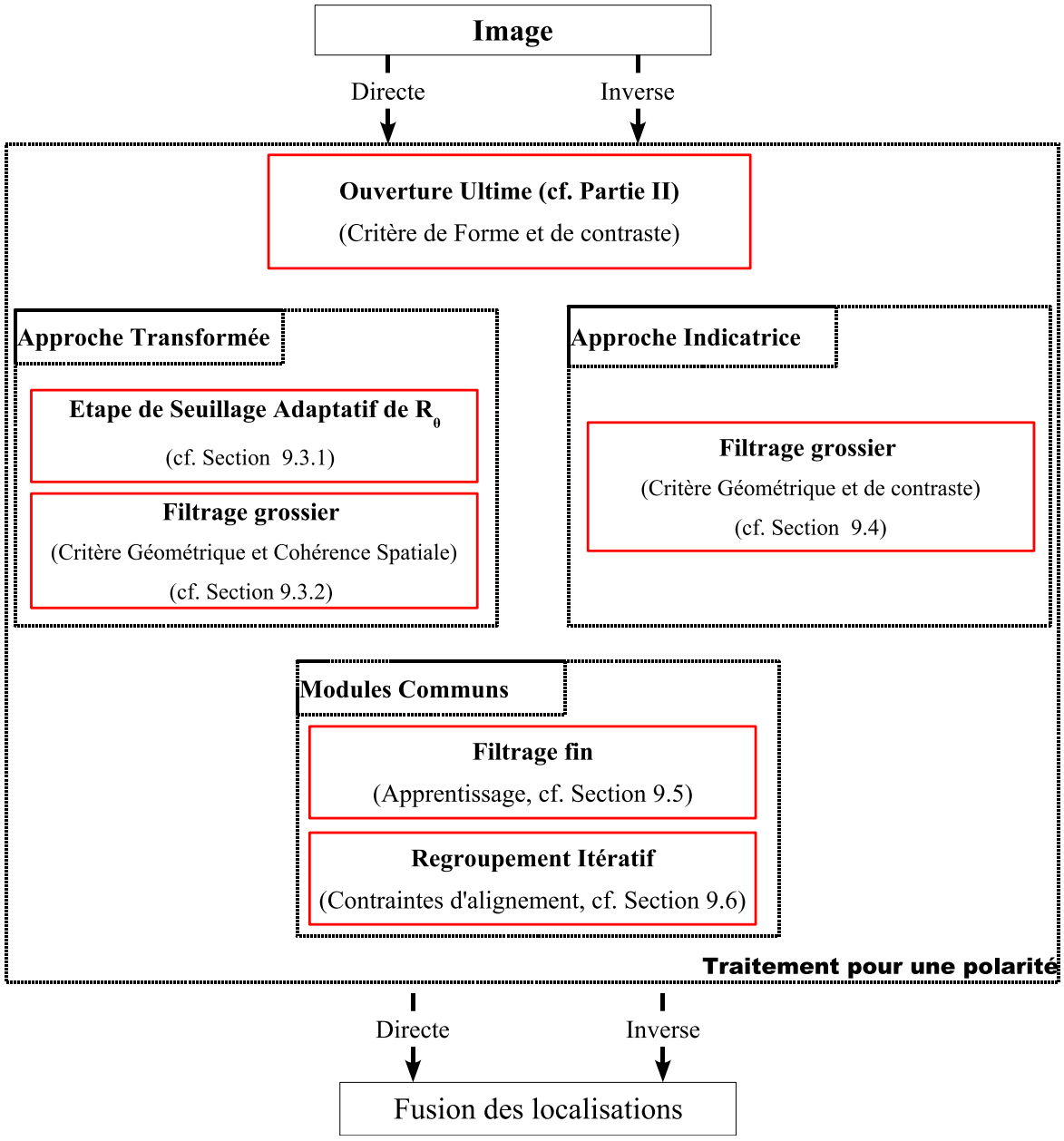


FIG. 9.2: Présentation des chaînes de traitement Transformée et Indicatrice

1. Cas des approches globales : Une des hypothèses majoritaires de ces méthodes est qu'il existe un *mode* représentatif des structures à extraire. Or dans notre cas rien n'indique que les lettres formeront un tel mode dans l'histogramme.
2. Cas des approches locales : Une solution consiste à appliquer ces méthodes de binarisation non sur la totalité de l'image, mais au sein d'une fenêtre glissante, permettant ainsi de s'adapter aux variations locales de l'image. Cependant ces approches nécessitent la connaissance d'une échelle d'analyse (ici la taille de la fenêtre glissante), ce qui nous est inconnu.

Aussi il ne nous est pas apparu de méthode simple et élégante pour cette étape. Nous soulignerons toutefois que le nombre de méthodes proposées dans la littérature est considérable (voir pour exemple la revue des méthodes de Sezgin and Sankur [83]). Aussi, nous ne pouvons affirmer qu'il n'existe pas de méthode automatique qui résolve notre problème.

Nous nous sommes tournés vers une approche paramétrique résumée sur la Figure 9.3.

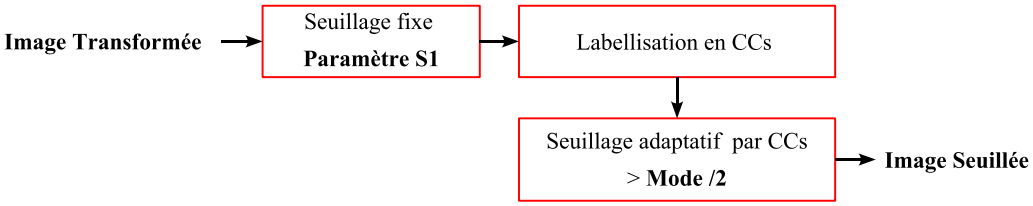


FIG. 9.3: Module de seuillage de l'image transformée R_θ

Nous sélectionnons à l'aide **d'un seuil**(S) les structures d'un contraste minimal qui pourraient être des lettres. L'application d'un seuil paramétrique étant toujours une étape délicate, nous appliquerons premièrement un seuillage "bas" pour ne pas perdre de lettres puis un seuillage adaptatif pour affiner le résultat précédent.

Après l'application du seuil bas, deux cas se présentent :

1. L'application du seuil a permis d'obtenir une composante connexe par lettre. Exemple en Figure 9.4. On notera que l'indicatrice (cf Figure 9.4(c)) n'est guère utilisable.
2. Le seuil est trop bas, nous avons agrégé les composantes lettres avec une portion du fond. Exemple en Figure 9.5 L'application du seuil adaptatif permet de récupérer une composante par lettre.

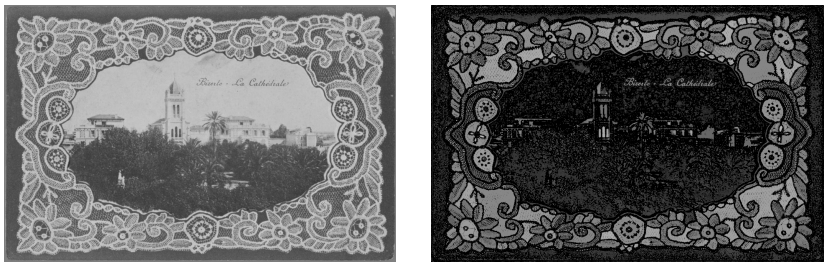
Pour résoudre le problème de fusion, nous labellisons les composantes issues du seuil bas et nous appliquons un seuil adaptatif dans chacune des composantes. Pour cela nous regardons la distribution des valeurs de gris dans chaque composante, tous les pixels (d'une composante) dont l'intensité est inférieure au mode de distribution divisé par deux sont mis à zéro.

L'hypothèse est que lorsque nous agrégeons les lettres avec une portion du fond, ce sont d'une part celles-ci qui contribuent le plus à la distribution obtenue et d'autre part celles de plus fort contraste. De plus si nous avions déjà une composante isolée après l'application du seuil bas, ce type de seuillage ne sera pas destructif.

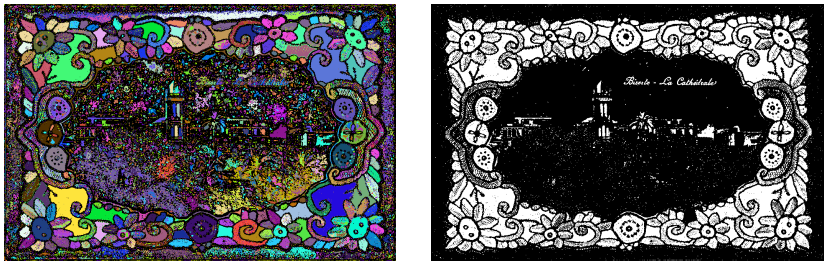
Comme nous l'avons souligné, cette étape est discutable et de nombreux contre-exemples peuvent être proposés. Cependant il ne nous est pas apparu de solution *plus élégante*.

Remarque 12 *Validation du module de seuillage ?*

L'utilisation d'une méthode paramétrique au tout premier niveau d'une chaîne de traitement étant discutable, nous donnerons, par la suite, systématiquement l'influence du choix de paramètre de seuil



(a) Image originale (b) R_θ, Correction $\Gamma = 3$



(c) q_θ labellisée (d) R_θ, Seuil fixe : Valeur des pixels>8

(e) Zoom R_θ seuillée (f) Zoom q_θ

FIG. 9.4: Illustration du seuillage de la transformée.Ouverture ultime par hauteur avec arret au tiers de l'image. Cas 1 : l'indicatrice n'est pas utilisable en l'état. Le seuillage «bas» la transformée nous permet de récupérer des composantes connexes exploitables.Base ImagEval

sur la qualité *de la localisation obtenue. Pour cela nous ferons varier ce seuil dans une plage donnée et nous tracerons l'évolution de la précision et du rappel. Ceci nous permettra de trouver un seuil* optimal[2] *pour une base donnée et de mettre en lumière la sensibilité de la chaîne à celui-ci.*

9.3.2 Méthode grossière pour un premier filtrage des composantes non lettres

Nous allons proposer ici deux étapes de filtrage simples qui permettront de réduire de manière drastique le nombre de composantes non probables (i.e non lettres)

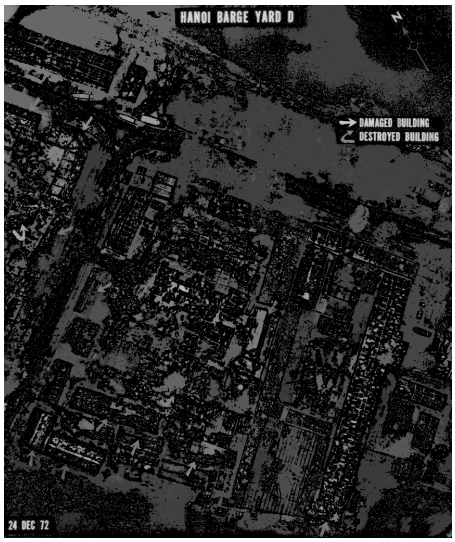
La première aura trait à la suppression de zones de l'image ne possédant pas des propriétés de *cohérence spatiale des zones de texte* (eg : zone fortement texturée) et la seconde aux filtrages de composantes non lettres sur des critères géométriques simples.

9.3.2.1 Filtrage basé sur la "cohérence spatiale" des zones de texte

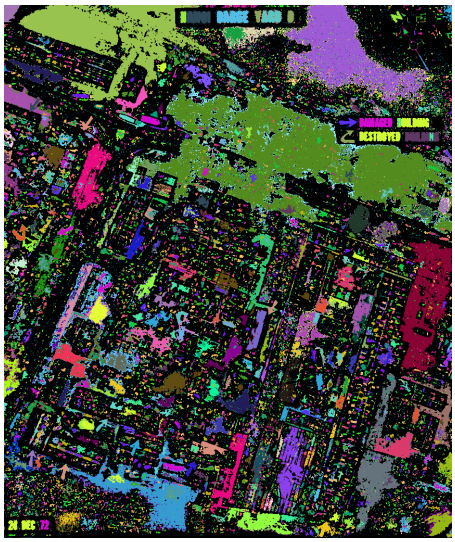
Dans les étapes précédentes l'ouvert ultime a extrait les structures d'intérêt d'une image en leur associant une taille et un contraste donnés. L'étape de seuillage précédente a permis une première sélection sur un critère de contraste. Parmi les structures restantes se trouvent les caractères que nous cherchons mais aussi de nombreuses *autres* composantes, notamment dans les zones de textures. Nous

[2]Optimal au sens où les autres paramètres seront quant à eux fixés, il ne s'agira donc pas de l'optimum global atteignable

(a) Image originale



(b) R_θ, Correction $\Gamma = 3$

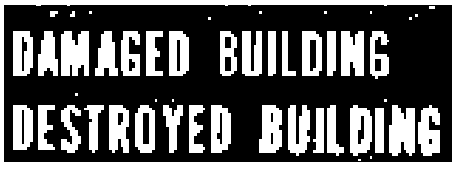


(c) q_θ labellisée

(d) R_θ, Seuil fixe Valeur du Pixel >8

(e) Zoom R_θ seuillée

(f) Zoom q_θ



(g) Zoom R_θ après application du seuil «bas», et du seuil adaptatif

FIG. 9.5: Illustration du seuillage de la transformée.Ouverture ultime par hauteur avec arrêt au tiers de l'image. Cas 2 : l'indicatrice n'est pas utilisable en l'état. Le seuillage «bas» connecte les lettres avec le fond. L'utilisation du seuillage adaptatif permet de récupérer des composantes connexes exploitables. Base ImagEval

avons constaté un grand nombre de composantes connexes très proches les unes des autres dans ces dernières. A l'inverse les zones de texte reposent le plus souvent sur des *Supports coopératifs*, elles sont proches entres elles mais distantes d'autres composantes de l'image (comme présenté sur la Figure 9.7(b)).

Pour décider si une composante donnée ne fait pas partie d'une zone de texte, nous allons regarder le nombre de composantes connexes regroupées par une simple dilatation de taille $SizeBall$. Si ce nombre est élevé nous considérerons qu'il ne s'agit pas d'une zone de texte et nous l'éliminerons.

Plus formellement on définira le nombre de composantes connexes regroupées $NbCC_{Ri}$ en une région Ri par une étape de dilatation δ de taille $SizeBall$. Si $NbCC_{Ri}$ dépasse un certain seuil $Tresh_{NB_{CC}}$ pour Ri nous considérons que cette zone ne peut être une zone de texte et nous la supprimons.

En première approche les paramètres $SizeBall$ et $Tresh_{NB_{CC}}$ ont été fixés expérimentalement à :

$$SizeBall_{CC} = 1 \quad Tresh_{NB_{CC}} = 500$$

Cette première étape de filtrage est résumée dans le premier bloc de la Figure 9.6.

Discussion sur quelques cas problématiques Cette étape simple a l'avantage d'éliminer rapidement et simplement un grand nombre de faux positifs, et ainsi d'accélérer la suite des traitements. En contrepartie, nous avons quelques faux négatifs notamment dans les cas où :

- Certaines lettres sont proches d'une zone texturée, cas de la Figure 9.7 en haut. Même en utilisant $SizeBall = 1$ (i.e une dilatation unitaire), une partie du texte est agrégée à la texture proche et donc supprimée par ce module.
- L'ensemble de la zone de texte repose sur une zone texturée , cas de la Figure 9.7 en bas.

On pourra également arguer qu'il s'agit d'un module paramétrique, les valeurs de $SizeBall$ et $Tresh_{NB_{CC}}$ ayant été fixées empiriquement, et que la pertinence de cette étape sera fortement dépendante de la qualité de l'étape précédente (i.e détermination des composantes connexes). Dans la Section 9.3.2.4 nous discuterons de la quantification de l'apport de ce module.

9.3.2.2 Filtrage basé sur les propriétés attendues des traits des caractères

Une des caractéristiques simples que l'on peut attendre des caractères *plans* est que l'épaisseur de leur trait est sensiblement inférieure à leur hauteur. Nous proposons donc de supprimer toute composante connexe pour laquelle :

$$\frac{\text{Épaisseur}_{CC} * 2}{\text{Hauteur}_{CC}} > 1$$

Obtenir un estimateur efficace et non biaisé de l'épaisseur d'un caractère est complexe (nous y reviendrons dans la Section 9.5.2). Nous désirons ici supprimer le maximum de composantes non probables en minimisant la perte de lettres, nous avons donc opté pour un estimateur qui fournira soit une estimation proche de l'épaisseur réelle, soit qui aura tendance a sous-estimer celle-ci.

Il est calculé comme l'infimum de l'épaisseur estimé à l'aide des fonctions distances horizontales et verticales.

Cette seconde étape de filtrage est résumée dans le second bloc de la Figure 9.6. Comme le module précédent, ceci nous permet d'éliminer rapidement des composantes non lettres avec une faible probabilité de faux négatifs.

Remarque 13 *Notons ici qu'il est préférable que ce module soit situé après le module basé sur la cohérence spatiale. En effet, le présent module pourrait supprimer (à juste titre d'ailleurs) des composantes connexes dans des zones texturées et par là même amoindrir l'efficacité du premier module.*

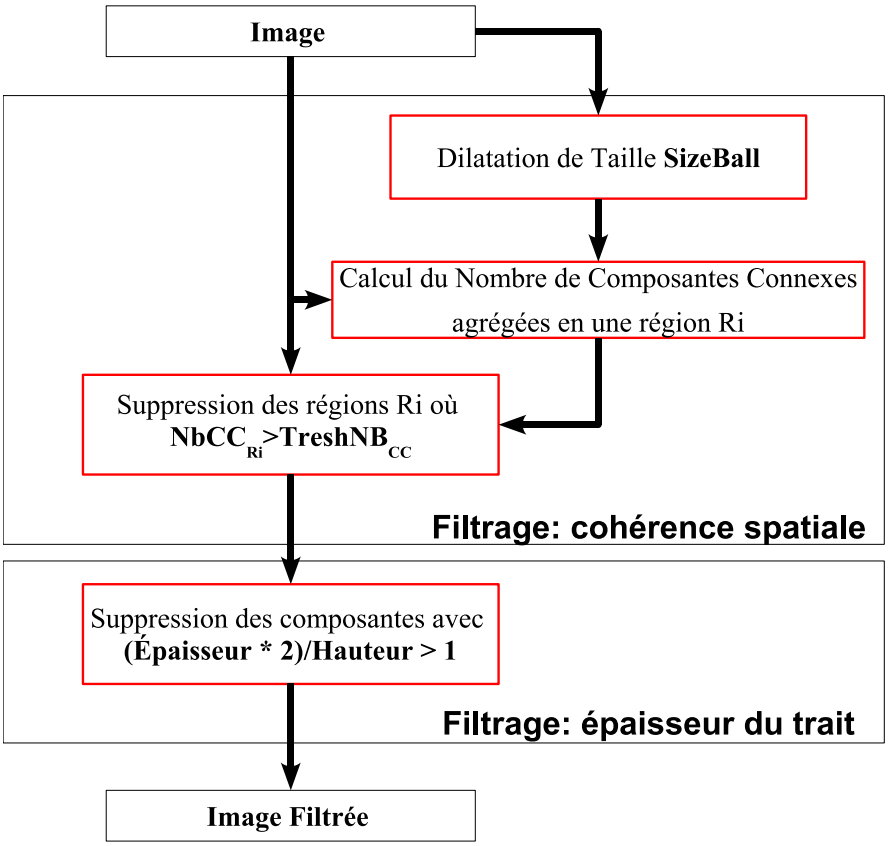


FIG. 9.6: Architecture du module de filtrage grossier.

Discussion sur quelques cas problématiques : Nous allons soulever principalement deux cas problématiques :

1. L'estimateur de l'épaisseur du trait utilisé est sensible aux problèmes de digitalisation et il aura tendance à sous-estimer l'épaisseur réelle de la composante. Même si des efforts importants sont apportés à la définition d'un meilleur descripteur cette seule contrainte du ratio épaisseur sur hauteur est mince pour discriminer lettres et non lettres. Aussi des composantes *non lettres* vont passer au travers de ce tamis, bien qu'elles ne semblent pas avoir les propriétés requises.

2. Si un caractère était déjà détérioré, soit parce qu'il l'est intrinsèquement dans l'image, soit parce qu'une partie de l'information qu'il véhicule a été perdue lors d'une étape précédente ; celui-ci sera supprimé par ce module. Il est à noter que ce problème n'est pas simple à résoudre car une telle composante n'aura pas les caractéristiques attendues d'un caractère et sera de toute façon supprimée par le module de filtrage fin des composantes basées sur l'apprentissage (voir Section 9.5).

9.3.2.3 Exemple de filtrage sur un cas réel

On trouvera sur la Figure 9.8 un exemple de filtrage obtenu à l'aide du module de filtrage grossier. On notera le grand nombre de composantes non lettres supprimées par les deux sous-modules proposés.

9.3.2.4 Quantification de l'apport du filtrage grossier

Les deux étapes précédentes ont supprimé pléthore de faux positifs en minimisant le nombre de faux négatifs, certes mais de combien ?

Ici se pose le problème de mesurer de manière non ambiguë l'apport de ce filtrage, or plusieurs problèmes surviennent :

1. Dépendance au module précédent : c'est le plus facile à appréhender, quand on applique ce traitement on est à un endroit "i" de la chaîne. Rien ne nous indique que n'avons pas perdu certaines zones préalablement.
2. Métrique ? : c'est le problème le plus flagrant. Celle-ci est basée sur les boîtes englobantes finales entourant les zones de texte localisées. Or nous tentons ici **de supprimer des composantes connexes**. Une solution serait de mesurer le nombre de composantes connexes *correctement filtrées*. En première approximation on pourrait nommer *correctement filtrées* l'ensemble des composantes connexes filtrées par ce module et *n'étant pas inclues dans les espaces délimités par la vérité terrain*. En substance :

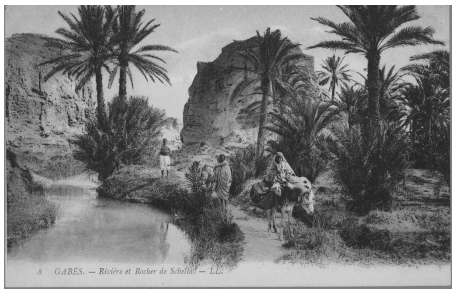
$$\text{PRÉCISION} = \frac{\text{Nombre de composantes } \textit{correctement} \text{ filtrées}}{\text{Nombre de composantes filtrées}}$$
$$\text{RAPPEL} = \frac{\text{Nombre de composantes } \textit{correctement} \text{ filtrées}}{\text{Nombres de composantes } \textit{à} \text{ filtrer}}$$

   Cependant deux biais surviennent, qui sont liés à :

   a) la granularité de la vérité terrain : plus la granularité est épaisse[3] (mot, puis phrases/paragraphe) plus le compte des composantes est biaisé et tendrait à être surévalué, il faudrait pousser la granularité à une finesse extrême (ie. au niveau lettre).

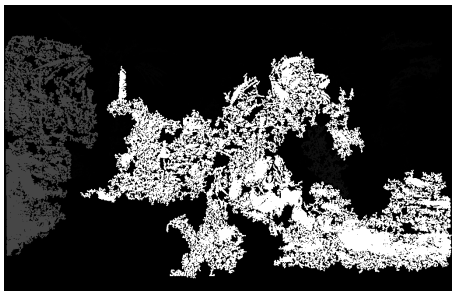
   b) la «polarité du texte» : si la chaîne de traitement agit indépendamment sur les deux polarités du texte, cette métrique induit inévitablement un biais. En effet prenons l'exemple d'un texte noir sur fond clair et plaçons-nous dans le cas où le système traite la polarité inverse, le module peut filtrer des composantes qui seront considérées comme des faux positifs lors du calcul de la métrique.

[3] Le problème se pose également si le texte est penché, l'annotation est sous forme de boîte englobante et non sous forme du plus petit parallélogramme englobant



(a) Image originale

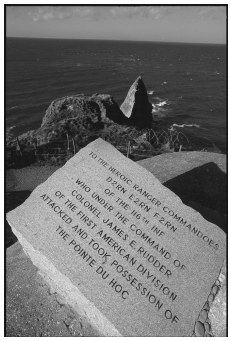
(b) Image seuillée (polarité noire du texte)



(c) Zone non probable en blanc

(d) Résultat du filtrage basé sur la cohérence spatiale

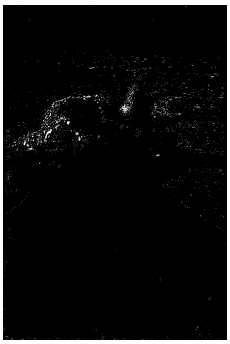
Perte d'information pour des zones de texte proches de zone texturés de l'image.



(e) Image Originale

(f) Image seuillée issue des étapes précédentes

(g) Zone non probable en blanc



(h) Résultat du filtrage basé sur la cohérence spatiale

Perte d'information pour des zones de texte présentes sur un fond texturé.

FIG. 9.7: Exemple de cas pathologiques où le filtrage grossier basé sur la cohérence spatiale supprime des zones de texte. Base $imagEval$

(a) Image Originale (b) Image de composantes connexes

(c) Résultat du filtrage basé sur la cohérence spatiale (d) Résultat du filtrage basé sur l'épaisseur du trait

FIG. 9.8: Exemple de l'apport du filtrage grossier sur une image réelle. Base $imagEval$.

L'idéal serait de disposer d'une *segmentation de référence*, mais comment rendre cette segmentation indépendante d'un système donné ? Aussi nous garderons cette métrique malgré les biais que nous avons soulignés. Les définitions des différents cas de figure sont explicitées sur la Figure 9.9.

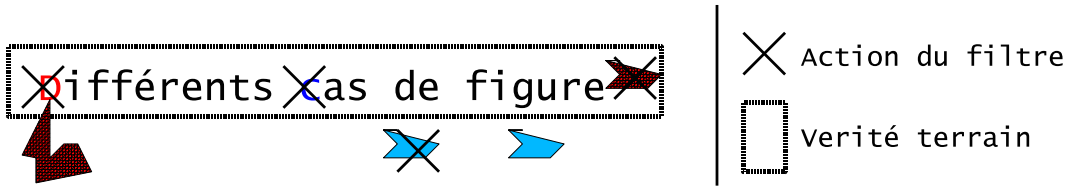


FIG. 9.9: Métrique pour le filtrage grossier : En noir les composantes lettres qui ne sont pas détruites. En bleu les cas correctement traités : la lettre supprimée est comptée comme un faux positif, la composante non lettre et non supprimée est comptée comme un faux négatif. En rouge les biais de la métrique : la composante non lettre est comptée comme un faux positif, la lettre agrégée avec une autre composante n'est pas comptabilisée.

Nous avons testé ces filtres sur la base d'image Officiel d'ImagEval comportant 500 images, la vérité terrain étant fournie par le comité d'organisation. Les résultats sont présentés dans le Tableau 9.1.

Nous obtenons pour les deux filtres une précision proche de 1, ce qui indiquerait qu'aucune lettre

n'aurait été perdue[4]. Le rappel du premier module est faible car un fort bruit persiste après l'application de celui-ci (cf Figure 9.8(c)), bruit qui est généralement filtrée par le second module dont le rappel est bien plus élevé.

Filtrage basé sur la «cohérence spatiale»		
Précision	Rappel	Moyenne Harmonique
100%	10%	14%
Filtrage basé sur «l'épaisseur»		
Précision	Rappel	Moyenne Harmonique
99.9%	91.3%	95.3%

TAB. 9.1: Approche Transformée. Quantification des modules de filtrage *grossier*. .

9.3.3 Conclusion et Perspectives

Conclusion sur le module de filtrage : le module présenté est simple et permet de manière rapide et efficace de supprimer de nombreuses composantes non caractères. Ceci permettra de proposer par la suite des outils plus spécifiques mais malheureusement plus coûteux pour discriminer plus finement lettres et non lettres. Il s'agit également d'un module paramétrique, les valeurs fournies ayant été fixées expérimentalement pour un meilleur rapport filtrage sur faux négatifs. Malgré ces défauts, le bilan est positif , et ce module peut être utilisé en l'état. Le paragraphe suivant détaillera les pistes d'amélioration et de consolidation à mettre en place.

Perspectives sur le module de filtrage : nous scinderons ici les perspectives en deux catégories complémentaires :

1. Renforcement et validation de la méthode : un des problèmes de ce module est le biais de la métrique mesurant son action. Il faudrait, il nous semble définir une vérité terrain au niveau composante connexe. C'est une étape un peu artificielle car il est bien évident que ceci suggère une segmentation de *référence* qui par essence n'existe pas (cf Section 4.5.2.2). Cependant rien n'empêche d'en proposer une tant qu'elle permet d'augmenter *de manière transparente* la qualité d'un système donné, en d'autres termes on renforcerait la dépendance entre les modules pour obtenir de meilleurs résultats. Une vérité terrain étant disponible, on pourrait d'une part fixer de manière non arbitraire les paramètres des modules existants mais également en proposer de nouveaux. Dans l'absolue, il faudrait que cette vérité soit également fournie par *des tiers*.
2. Multiplication des filtres : nous travaillons au niveau *caractère*, aussi définir des filtres simples pour valider ou invalider un caractère est plus *sensible* que dans le cadre de la validation de *zones de texte*. Cependant si une vérité terrain est disponible (et bien sûr, en prenant garde aux problèmes de généralisation), nous pourrions proposer d'autres filtres (eg : critère de solidité (Aire/Aire Enveloppe Convexe) et de convexité (Périmètre de l'enveloppe convexe/périmètre), ratio des boites englobantes) et également définir un système sous la forme d'une cascade de classifieurs (eg :[17]). Ceci rejoindra les perspectives du module de filtrage par apprentissage décrit en Section 9.5.

[4]En fait nous en perdons quelques-unes, mais ce nombre est extrêmement faible vis-à-vis du nombre de composantes *autres supprimées*

9.4 Modules de l'Approche Indicatrice

9.4.1 Filtrages sur les valuations de l'image indicatrice

Dans cette section, nous allons proposer trois filtrages simples pour l'approche Indicatrice :

1. Le premier correspond aux filtrages de zones que l'on considérera comme *mal valuées*, et que S.Beucher nomme *valeurs perchées* dans [7]. Ce sont des composantes de l'indicatrice dont la valuation et la taille réelle ne concordent pas.
2. Le second (très pragmatique), tendra à réduire les composantes issues du bruit présent dans l'image.
3. Le troisième, pragmatique également, tendra à supprimer les composantes ne possédants pas un contraste suffisant.

9.4.2 Filtrage des valeurs perchées

Nous allons reparler ici de certains problèmes de valuations de l'image q_θ. On pourra les différencier en deux familles, bien que fondamentalement, elles correspondent toutes deux à ce que Beucher ([7]) nomme des *valeurs perchées*.

1. Ce sont des régions dont la valeur de gris est **bien supérieure** à leur taille réelle. Il suffit de reprendre les exemples de la Section 7.2, en considérant que lorsque le plateau sur lequel repose les structures disparaît, le résidu est supérieur non pas en tout point mais en quelques sites seulement. On retrouve alors des valuations complètement incohérentes en ces sites.
2. La seconde correspond à ce que nous nommerons un effet de couronne. Les zones de transition autour des lettres sont valuées pour une taille légèrement supérieure. Deux configurations peuvent apparaître voir Figure 9.10(a) et Figure 9.10(b). Dans la première, les zones sont clairement *mal valuées*, on peut donc facilement les supprimer, dans le second cas bien qu'il s'agisse pour nous d'une structure parasite, sa valuation est correcte et nous n'avons pas de solution géométrique pour la supprimer.

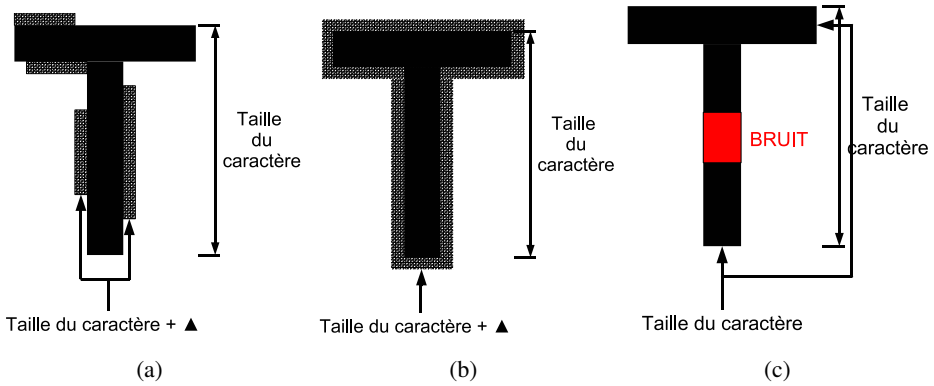


FIG. 9.10: Différents types de valeurs perchées de l'image q_θ.

9.4.3 Filtrage proposé

Nous pourrions nous dire que la connaissance de ces comportements pourrait nous permettre de *simplement* filtrer une partie de q_θ. Il suffirait de supprimer les portions d'image de q_θ pour lesquelles, il n'y a pas concordance entre la valuation et la taille intrinsèque d'une région. Un exemple d'un tel filtrage est proposé sur la Figure 9.11(e)

Remarque 14 *Il ne faut pas oublier que lorsque du bruit est présent dans l'image et/ou pour de petites lettres, ce sont les lettres elles-mêmes qui peuvent se retrouver scindés en plusieurs fragments. Ceci est illustré sur la Figure 9.10(c). On devra donc être conscient que ce filtrage pourra nous faire perdre quelques cas.*

9.4.4 Suppression des *petites composantes*

La présence de bruit de type "poivre et sel" et plus généralement de texture *fine* entraîne la création de "petites" composantes connexes au sein de l'indicatrice. Elles se présentent comme de fins et hauts pics[5] au sein du relief topographique représentant l'image. Elles disparaissent pour de faibles tailles d'ouvertures, mais en engendrant des résidus «importants». Aussi elles persistent au sein de q_θ.

Une solution très simple et pragmatique consiste à utiliser un petit filtre aréolaire d'une taille (S_{area}) inférieure à celle des plus petites de lettres présentes dans la base d'images. [6]

Un exemple de filtrage obtenu est présenté en Figure 9.11(h).

9.4.5 Filtrage sur le contraste

A l'instar du module de seuillage de l'approche transformée (cf. Section 9.3.1), nous allons appliquer un seuil bas et paramétrique pour sélectionner les structures de contraste minimal. Notre but, ici, n'est pas d'utiliser ce seuil pour une segmentation de l'image mais bien de filtrer les zones faiblement contrastées.

Ce seuil fixe (S) est appliqué sur l'image transformée ; nous obtenons ainsi une image binaire qui va nous servir de masque. En effet, nous classifierons comme «fond» toutes zones de l'image indicatrice non incluses dans ce masque.

Le résultat d'un tel filtrage est présenté sur la Figure 9.11(j).

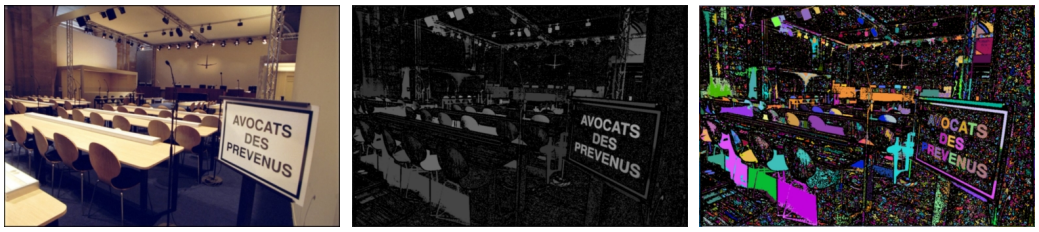
Notons qu'ici le choix du seuil est moins problématique que pour le module de seuillage de l'approche transformée. Le but recherché est toujours de supprimer de manière rapide le maximum de composantes non lettres. Cependant le choix d'un seuil fixe restant toujours problématique, nous veillerons par la suite à mesurer systématiquement son influence sur la performance globale de nos systèmes.

9.4.6 Quantification de l'apport de ces filtrages ?

Comme pour les modules de filtrage de l'approche transformée, les filtrages proposés ont supprimé pléthore de faux positifs. La métrique définie précédemment (cf Section 9.3.2.4), n'est pas transposable ici. En effet, nous ne travaillons plus avec des «blobs» (Ie composantes connexes blanches sur fond noir) mais au niveau des zones plates. Et le simple test d'inclusion proposé préalablement n'a plus de sens.

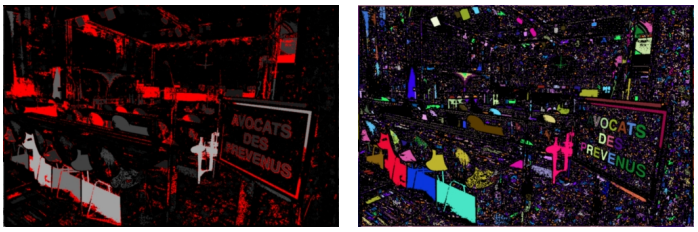
[5] Haut fait référence ici à leur dynamique vis-à-vis de leur fond propre

[6] Une taille typique de 5 à 6 pixels a été utilisée par la suite



(a) Image Originale (b) R_θ Correction $\Gamma = 3$ (c) q_θ labellisée

Ouverture Ultime par hauteur avec arrêt de l'opérateur au tiers de l'image.

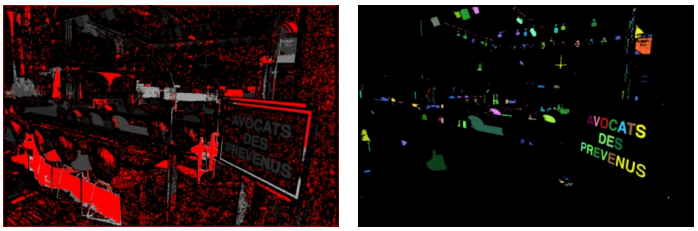


(d) Mise en lumière des valeurs perchées (en rouge) sur l'indicatrice (e) q_θ labellisée après suppresion des valeurs perchées

Illustration de la suppression des valeurs perchées

(f) Mise en lumière des petites composantes de bruit (en rouge) sur l'indicatrice (g) Zoom sur la Figure 9.11(f) (h) q_θ labellisée après suppression du bruit

Illustration de la suppression des petites composantes assimilées à du bruit



(i) Mise en lumière des zones de contraste faible(en rouge) sur l'indicatrice, Seuil sur l'image transformée : $S \leq 9$ (j) q_θ labellisée après filtrage sur le contraste

Illustration de la suppression de composantes de trop faible contraste

FIG. 9.11: Illustrations des modules de filtrage de l'indicatrice. Module de suppression des valeurs perchées, filtrage des petites composantes parasites et filtrage sur le contraste.

Aussi, nous ne pouvons proposer pour le moment de quantification de l'apport de ce module. Comme précédemment et même si cela est discutable, il faudrait disposer d'une segmentation de référence.

Nous allons maintenant passer à la description des modules hybrides. Il s'agit de modules qui interviendront quelle que soit l'approche utilisée. Après les étapes de segmentation et de filtrages grossiers il nous reste :

1. A supprimer les dernières composantes «non lettres».
2. A produire nos résultats de localisation sous formes de boîtes englobantes car c'est à cette échelle que seront évalués les algorithmes.

Le dernier filtrage entre lettres et non lettres, sera basé sur la définition d'un classifieur séparant ces deux populations. Il est décrit en Section 9.5.

La création des boîtes englobantes sera assurée par un module de regroupement itératif des composantes décrit en Section 9.6.

9.5 Méthode de filtrage par apprentissage

Organigramme Les chaînes présentées utilisent des *filtres grossiers* permettant de déterminer facilement et de manière *efficace* un grand nombre de composantes non lettres. A la sortie de ces filtres nous n'avons pas/plus de moyen simple pour filtrer les composantes non lettres restantes.

Nous devons donc développer un module de classification qui va nous permettre de discriminer lettres et non lettres.

Ce module, va passer d'une part par la détermination de caractéristiques et d'autre part par le choix d'une méthode d'apprentissage supervisée.

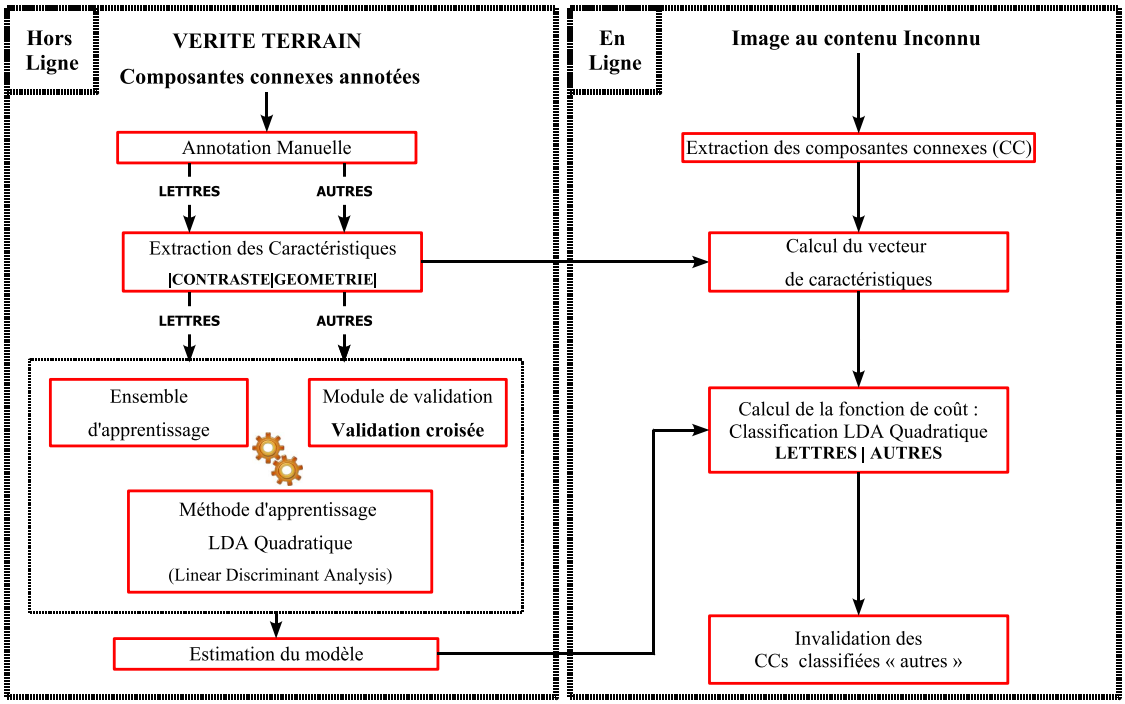


FIG. 9.12: Architecture du module de sélection des composantes par apprentissage.

On trouvera, sur la Figure 9.12 le schéma descriptif de l'approche. Les différentes caractéristiques retenues, ainsi que la méthode d'apprentissage sélectionnée, seront explicitées et commentées dans les sous-sections suivantes.

9.5.1 Annotation et vérité terrain

Concernant la discrimination entre lettres et non lettres, il ne nous a pas semblé judicieux de partir de bases de caractères pré-annotées ou de lister un ensemble de polices de caractères à différentes échelles. En effet, notre but est de différencier les lettres des non-lettres dans notre cadre applicatif. Or le fossé de *qualité* entre des lettres piochées dans des bases polices et les composantes connexes extraites des images de textes enfouis est parfois abyssal.

Nous avons donc opté pour une annotation en adéquation avec notre cadre applicatif. 177000 composantes connexes ont été ainsi extraites de 350 images de la base de test à blanc de la campagne d'évaluation $imagEval$. Ceci sous-tend que nous avons utilisé une sortie de l'une de nos chaînes de traitement. La généralisation à d'autres bases validera cette approche.

Elles ont été classifiées manuellement en deux classes : $lettres$ et $non - lettres$. Cette annotation est entachée d'un fort déséquilibre entre la taille de la population des "lettres" et celle des "non-lettres" de respectivement 5% et 95% que l'on devra prendre en compte lors de la phase d'apprentissage proprement dite.

Importance de cette annotation : Par la suite, quand nous ferons référence aux classifieurs mis en place et quelle que soit la base d'image traitée,ceux-ci auront toujours été *entrainés* sur ce jeu de données.

9.5.2 Caractéristiques sélectionnées

Avant de commenter les différentes caractéristiques que nous avons sélectionnées, nous attirons l'attention du lecteur quant aux variations très importantes de taille de lettres que nous avons traitées. Celles-ci pourront avoir une hauteur de quelques pixels à plusieurs dizaines de pixels, voir plusieurs centaines de pixels[7]. Ces variations d'échelle couplées aux problèmes de digitalisation font qu'il est difficile de trouver des descripteurs robustes et non ambigus. Nous avons donc opté pour la multiplication de descripteurs de forme simples[8] ; au détriment de descripteurs communs dans la littérature (eg : descripteur de Fourier, moments, histogrammes de projection cf. [96]). Leur couplage au sein de l'apprentissage en feront des mesures discriminantes. Nous nous efforcerons donc pour la suite d'expliciter pour chaque caractéristique son intérêt et ses limitations.

Familles de caractéristiques Pour discriminer les lettres des non lettres, nous nous sommes concentré sur quatre grandes familles de caractéristiques :

1. Celles concernant l'épaisseur du trait et la cohérence de celle-ci sur l'ensemble de la composante connexe.
2. Celles concernant des "propriétés" géométriques globales des lettres.
3. Celles concernant les propriétés de régularité des lettres.
4. Celle concernant le contraste entre la lettre et son fond.

[7] Pour des tailles d'images allant du format CIF à des résolutions de plus de 1600x1200 pixels

[8] Ne respectant forcément pas les critères d'invariance d'échelle, de rotation et/ou de translation

Ces caractéristiques sont issues des travaux de séparation texte/fond de l'analyse de documents ([121]) et de différents travaux de localisation de texte dans les images basés sur des approches en composantes connexes (eg : [50], [124], *Ashida System* [53]).

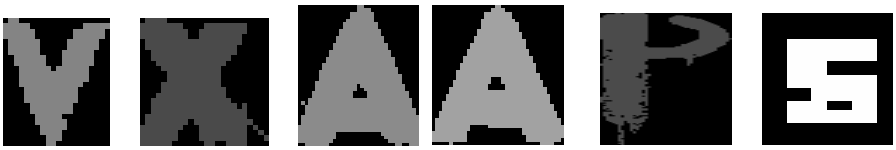
9.5.2.1 Estimation de l'épaisseur du trait :

Une caractéristique essentielle d'une lettre qui la distingue d'autres composantes connexes est de considérer qu'elle est constituée de traits. Ces traits sont caractérisés par une épaisseur (ou Graisse) qui certes varie entre deux extrêmes, le "délié" et le "plein", mais que l'on considérera comme uniforme pour notre application. On peut facilement concevoir que l'épaisseur moyenne des traits d'une lettre est relativement faible vis-à-vis d'autres caractéristiques géométriques et on peut également attendre que cette épaisseur soit relativement constante le long de la lettre à l'inverse d'autres composantes.

Nous allons proposer et expliciter ici les différentes mesures de l'épaisseur retenues : en effet aucune des mesures suivantes n'est suffisamment robuste pour en donner une estimation non ambiguë. Ceci étant surtout dû aux problèmes de digitalisation et d'échelle dans le cadre numérique.

- Fonction Distance : une première mesure intuitive de l'épaisseur d'une composante est de calculer le maximum de la fonction distance (voire [90]) (Noté $Max_{DistFunc}$) au sein de celle-ci. On espère ainsi pour une lettre obtenir la mesure de l'épaisseur du trait plein.

  Cependant comme illustré sur les images de la Figure 9.13 le max de la fonction distance ne se place pas toujours au sein du trait du caractère (cas 9.13(a) et 9.13(b)), la présence de trous parasites au sein de la composante 9.13(c) peut rendre son estimation caduque et elle donnera un résultat trivial pour des structures de petites tailles 9.13(d).



(a) Le maximum ne ce place pas dans le trait. Mais a la jonction

(b) Instabilité pour une même lettre en fonction de la position du trou

(c) Mauvaise estimation en présence de bruit

(d) Lettre trop petite

FIG. 9.13: Instabilité du maximum de la fonction distance comme estimation de l'épaisseur sur quelques lettres issues de l'annotation de la base ImagEval.

- Longueurs de plages : l'étude de la distribution des longueurs de plages dans la direction d'écriture d'une zone de texte est un moyen simple d'estimer l'épaisseur moyenne des traits dans le cadre de l'analyse de documents (eg :[121]).

  Nous proposons ici de l'utiliser au sein même des composantes connexes (comme proposé dans le système Ashida[53]), cependant cette restriction va rendre beaucoup plus sensible cette mesure aux problèmes de digitalisations et déformations géométriques. Aussi nous ne pouvons déterminer par avance la meilleure *statistique* à extraire de ces distributions.

  Nous avons opté pour la multiplication des descripteurs en conservant pour chaque composante : la moyenne et la médiane de la distribution des longueurs de plages dans les directions horizontales et verticales (respectivement RLE_{MedX},RLE_{MoyX},RLE_{MedY} et RLE_{MoyY}).

9.5.2.2 Estimation de la consistance de l'épaisseur du trait :

Comme nous l'avons mentionné précédemment, une lettre est composée de traits et si l'on passe outre les cas pathologiques de certaines fontes, on peut s'attendre à ce que l'épaisseur de ces traits soit relativement constante pour l'ensemble de la lettre, ce qui n'est pas garanti pour d'autres composantes connexes.

Pour mesurer cette consistance nous calculons les distributions des longueurs de plages et des différences des longueurs de plage (consécutivement) dans les directions horizontales et verticales. Nous extrayons des distributions «des différences», leur moyenne et leur variance (respectivement $RLE_{Delta_{MoyX}}$,$RLE_{Delta_{VarX}}$,$RLE_{Delta_{MoyY}}$,$RLE_{Delta_{VarY}}$) qui tendraient à être faibles pour les lettres et les variances des distributions RLE_{VarX}, RLE_{VarY} qui elles aussi tendraient à être faibles. On pourra souligner que comme pour la mesure de l'épaisseur seule, il s'agit de caractéristiques fragiles, sensibles aux problèmes de digitalisation et non invariantes par rotation.

FIG. 9.14: Exemple de lettre pour lesquelles les estimations de l'épaisseur et de la cohérence de celle-ci sont problématiques (Voire Tableau 9.2).

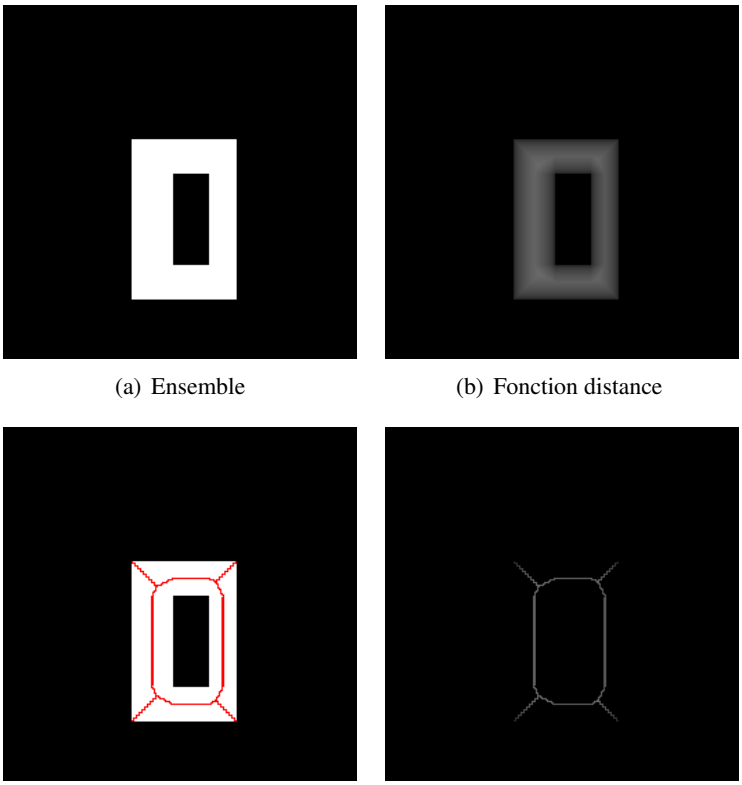
On trouvera sur la Figure 9.14 des exemples de lettres, où les descripteurs issus des longueurs de plages montrent de fortes limitations pour la mesure de l'épaisseur et de la consistance de celle-ci.

Utilisation du squelette pour la caractérisation de l'épaisseur et de la consistance de celle-ci Parallèlement aux différents estimateurs de l'épaisseur (et de la cohérence de celle-ci) vus précédemment, nous nous sommes intéressé à l'utilisation du squelette d'un ensemble pour extraire ces caractéristiques. Pour ce faire nous avons utilisé le squelette par points d'ancrage de Vincent [100], en utilisant comme base les maxima locaux de la fonction distance (cf. Figure 9.15(b)) au sein de la composante. Une fois le squelette obtenu on peut le valuer par la fonction distance obtenue préalablement. La moyenne et la variance (notée respectivement Moy_{skel} et Var_{skel}) le long du squelette valué (cf. Figure 9.15(d)) va nous fournir respectivement une estimation de l'épaisseur et de la variation de celle-ci le long de la composante.

Une première remarque tirée de la Figure 9.15, est que cette approche tend à sous-estimer l'épaisseur et à sur-estimer la variabilité de celle-ci. Ceci est dû d'une part à la nature du squelette lui même[9] et d'autre part aux problèmes bien connus de points *résiduels* (voire [90]) entraînant la formation de barbules.

Comparaison des mesures proposées Comme nous l'avons souligné précédemment, aucun des descripteurs précédents n'est exempt de reproche. Nous proposons dans les tableaux 9.2 et 9.3, une comparaison entre l'estimation psychovisuelle de l'épaisseur du trait de caractère et l'estimation, fournie par nos différents descripteurs. Le premier tableau 9.2 regroupe un ensemble de lettres *difficiles*, le second 9.3 concerne des lettres mieux définies. Tous les exemples fournis proviennent de notre annotation manuelle.

[9] Sur la Figure 9.15 la prise en compte des *coins* de la forme est liée à la propriété du squelette mais ne correspond pas à une *décomposition en trait* de la forme



(a) Ensemble (b) Fonction distance

(c) Squelette surimposé (d) Squelette valué

FIG. 9.15: Utilisation du squelette par points d'ancrage (4-connexe) pour la détermination de l'épaisseur et de la consistance de l'épaisseur d'un ensemble.

On observe bien que les descripteurs ne renvoient pas toujours une estimation non biaisée, et que sur l'ensemble des jeux de données, c'est la mesure issue du squelette qui tire le mieux son épingle du jeu.

9.5.2.3 Propriétés géométriques globales :

Ce sont des propriétés géométriques relativement simples, rapides à calculer et qui, couplées, peuvent facilement éliminer nombre de faux candidats. On calculera ainsi pour chaque composante sa hauteur h, sa largeur w, sa surface A, la surface de sa boîte englobante A_{Box}. Ces caractéristiques auront deux intérêts majeurs :

1. Elles serviront de facteur d'échelle pour pondérer d'autres caractéristiques
2. Leurs ratios sont utilisés comme des outils de filtrage pour discriminer les lettres des non-lettres pour des problématiques applicatives restreintes (eg : [50, 124]).

On notera que leur aspect discriminatoire devrait chuter si les lettres ont subi des déformations importantes de type : rotation, inclinaison et projection, ce qui justifie leur utilisation sous forme de caractéristiques et non de filtres.

Ensemble	Estimation Psycovisuelle	$Max_{DistFunc} * 2$	RLE_{MedX}	RLE_{MedY}	RLE_{MoyX}	RLE_{MoyY}	$Moy_{skel} * 2$
	1 et 5	6	4,5	2	4,12	3,45	3,26
	2	4	3,5	3	3,5	3,71	2,93
	5 et 2	6	5	2	4,27	5,88	3,84
	2 et 7	8	4	8	6,46	7,57	5,34
	4	6	5	8	4,81	6,59	4,29
	10 et 17	16	12	9	9,98	17,93	8,05
	7	12	9	8,5	9,54	10,38	7,33
	6	10	6	12	6,58	10,33	5,92
	10	14	24	11	19,67	14,84	9,95
	12	18	12	34,5	14,42	27,56	11,78

TAB. 9.2: Comparaison des estimateurs de l'épaisseur sur des lettres *complexes* de la base annotée.

Ensemble	Estimation Psycovisuelle	$Max_{DistFunc} * 2$	RLE_{MedX}	RLE_{MedY}	RLE_{MoyX}	RLE_{MoyY}	$Moy_{skel} * 2$
	3	6	6	4	8,96	4,07	3,94
	3/4	6	4	6	5,17	10,78	4,34
	4	6	4	4	6,74	5,33	3,94
	4	6	5	6	7,23	6,4	4,64
	7	12	13	9	19,06	10,72	9,33
	9	14	42	9	27,77	11,86	9,03
	10	12	10	9	14,49	16,62	8,4
	8	14	11	13	15,34	13,12	9,28
	8	12	19	9	25,77	10,59	9,98
	10	12	10	8	17,27	14,32	8,98
	9	10	12	9	16,75	12,17	10,01

TAB. 9.3: Comparaison des estimateurs de l'épaisseur sur des lettres *de bonne qualité* de la base annotée

9.5.2.4 Propriétés de régularité des lettres :

Une des propriétés que l'on peut attendre des lettres est qu'elles possèdent des formes plus "régulières" que d'autres composantes extraites. Pour mesurer cette régularité, nous proposons de calculer :

- La caractéristique d'Euler : calculée pour chaque composante connexe, elle renseignera sur le nombre de trous de celle-ci. Pour des lettres et en l'absence de bruit ce nombre tendra à être faible.
- La mesure de la compacité $CC = A/\text{Périmètre}^2$ (qui est habituellement utilisé comme mesure de circularité) nous renseigne sur le degré de compacité d'une forme. Les lettres bien que de formes multiples tendront à avoir une CC plus forte que nombre de non-lettres aux formes très irrégulières.
  Nous calculons également la complexité CX=Périmètre$/A$ qui informe si la composante connexe possède une frontière "torturée" : en effet pour une mesure d'aire constante, plus le périmètre est élevé plus la frontière de l'objet est complexe.[10] Ne sachant pas à l'avance si les composantes connexes seront fortement bruitées ou non nous calculons également ces caractéristiques sur les composantes après une opération de bouchage de trous ($CC_{FillHoles}$ et $CX_{FillHoles}$ respectivement).
- Le taux de remplissage/d'occupation (A/A_{Box}), que nous avons obtenu en couplant deux informations géométriques est également un paramètre de régularité. Pour des lettres horizontales il tendra à être important vis-à-vis de composantes moins régulières.

9.5.2.5 Contraste :

Enfin, une dernière propriété que l'on peut attendre d'une lettre est qu'elle ait un contraste significatif vis-à-vis de son fond propre (i.e pour une zone de quelques pixels l'environnant). Pour estimer ce contraste, nous calculons, pour chaque composante connexe, sa boîte englobante, nous étendons ensuite cette boîte de 10% dans chacune des directions. A l'intérieur de cette boîte, nous appliquons l'algorithme de la maximisation de la variance interclasse (M.V.I) pour deux classes. Il a été montré (Fukunaga [26]) que plus la MVI est importante plus les classes sont *correctement* séparées [11]. Nous conservons donc la valeur de la M.V.I comme caractéristique. De plus nous allons utiliser la propriété de conservation de la variance (Variance Totale= Variance Inter Classe + Variance Intra Classe) pour normaliser cette caractéristique. Nous définissons donc les deux caractéristiques de contraste suivantes :

$$C_1 = M.V.I \text{ et } C_2 = \frac{M.V.I}{\text{Variance Totale}}$$

Récapitulatif : nous avons exposé dans les paragraphes précédents les différentes caractéristiques retenues :

- Celles concernant l'épaisseur du trait et la cohérence de cette épaisseur sur l'ensemble de la composante connexe.

[10] Par souci de simplicité et de rapidité, le périmètre est estimé comme le nombre de pixels de la frontière extérieure de l'objet pour la quatre connexité. De meilleurs estimateurs pourraient être utilisés : Freeman "chain-code" ou "Formule de Crofton".

[11] Nous rappelons que ceci se fait sous l'hypothèse de distribution gaussienne des deux classes possédant une variance du même ordre.

- Celles concernant des "propriétés" géométriques globales des lettres.
- Celles concernant les propriétés de régularité des lettres.
- Celle concernant le contraste entre la lettre et son fond.

Prises une par une, elles n'ont qu'un faible pouvoir de discrimination et beaucoup d'entre elles sont sensibles au problème d'échelle, de rotation et de digitalisation. Cependant elles apporteront chacune des informations qui, couplées, permettront de pallier leurs insuffisances respectives.

Nous allons maintenant aborder la méthode d'apprentissage sélectionnée et présenter les résultats chiffrés de classification pour différents jeux de paramètres.

9.5.3 Méthode d'apprentissage sélectionnée

Dans le très large éventail des méthodes classifications disponibles, nous avons choisi une démarche pragmatique, en nous tournant vers une méthode *classique et robuste* de classification supervisée. Il s'agit de l'analyse linéaire discriminante (ALD)[12]. Dans les paragraphes suivants nous rappellerons brièvement les aspects techniques de cette méthode et la méthode d'entraînement que nous avons utilisé. Le lecteur pourra consulter [29] pour une description en profondeur des concepts théoriques sous-jacents.

Dans notre cas, nous n'avons pas de problème sur le nombre de classes à traiter. En effet nous sommes dans un cas «simple» à deux classes (lettres et non lettres) et nous devons trouver un *plan séparateur entre celle-ci*. Par contre nous devrons être attentif au fort déséquilibre des tailles des populations de chacune d'elles, la balance penchant fortement du coté de la classe non texte.

Rappels généraux sur le principe sur l'ALD Il s'agit d'une méthode d'analyse de type supervisé : l'analyse linéaire discriminante (*Linear Discriminant Analysis - LDA*) part de la connaissance de la partition en classes des ensembles. En d'autres termes, on a identifié pour chacune des classes, un ensemble d'individus appartenant à celle-ci.

Son objectif est de créer de nouvelles variables qui sont particulièrement efficaces pour séparer les classes. Ces nouvelles variables sont obtenues comme combinaisons linéaires des variables originales (i.e. caractéristiques ou descripteurs).

L'ALD consiste à déterminer des facteurs, combinaisons linéaires des variables descriptives, qui prennent des valeurs les plus proches possible pour des éléments de la même classe, et les plus éloignées possible entre éléments de classes différentes. Cela revient à décomposer la variance totale comme la somme des variance V intra-classe (moyenne des variances) et inter-classe (variance des moyennes) et à chercher à maximiser le rapport $\frac{V_{inter}}{V_{intra}}$.

Cette approche s'appuie sur l'hypothèse restrictive selon laquelle les classes ont des distributions normales. Quand cette hypothèse est convenablement vérifiée, l'ALD permet non seulement de générer des règles d'affectation, mais de plus de calculer les probabilités, pour chaque observation, d'appartenance à chacune des classes (probabilités dites "a posteriori").

Il est couramment admis (cf. [29]) que l'A.L.D donne des résultats *satisfaisants* même si l'hypothèse restrictive sur la distribution des classes n'est pas vérifiée. Bien sûr, il faudra le vérifier en pratique, en validant le *classifieur* généré.

[12] Nommée également de **Fisher**

9.5.3.1 Présentation formelle de l'ALD

Description et concept Considérons un ensemble d'observation x (nommé "features", caractéristiques, variables ou mesures) pour chaque échantillon d'un objet ou évènement dont nous connaissons la classe d'appartenance y. L'ensemble de ces échantillons est appelé *ensemble d'entraînement* (i.e Training set).

Le problème de classification se résume à trouver un *prédicteur performant* d'appartenance à une classe y pour tous les échantillons d'une même distribution (i.e. on ne veut pas juste traiter l'ensemble d'entraînement mais permettre la généralisation du système) en ne connaissant que les observations x.

Dans un cadre général, nous sommes en présence de n observations d'un couple (Y, X). Pour la i^e observation notée (Y_i, X_i), Y_i est un label qui dénote l'appartenance à un groupe/classe $j \in 1, ...g$ et $X_i \in \mathbb{R}^p$ est un ensemble de variables/features/caractéristiques explicatives de l'appartenance à un groupe (ie Y). Pour une nouvelle observation dont, nous pouvons mesurer l'ensemble de ses variables explicatives, $x_0 \in \mathbb{R}^p$, nous souhaitons connaître son groupe/classe y_o. Nous ne connaissons pas avec certitude le groupe y_o, nous devons donc modéliser cette incertitude par des probabilités d'appartenance à tel ou tel groupe. Ces probabilités sont évaluées grâce au théorème de Bayes.

Theorem 9.5.1 *Rappel théorème du Bayes : soit A et B deux évènements, le théorème de Bayes permet de déterminer la probabilité de A sachant B, si l'on connaît les probabilités de A, de B et de B sachant A.*

$$\mathbb{P}(A|B) = \frac{\mathbb{P}(B|A)\mathbb{P}(A)}{\mathbb{P}(B)}$$

Usuellement chaque terme de l'équation précédente se voit assigner la dénomination suivante :

$\mathbb{P}(A)$ *et* $\mathbb{P}(B)$ *sont les probabilités a priori de A et B*

$\mathbb{P}(A|B)$ *est appelée la probabilité a posteriori de A sachant B*

$\mathbb{P}(B|A)$*, pour un B connu, est appelée la fonction de vraisemblance de A*

Dans notre cas, l'application du théorème de Bayes (pour le cas continu) donne :

$$\mathbb{P}\left(Y = j|X = x_0\right) = \frac{f\left(x_0|y = j\right)\mathbb{P}\left(Y = j\right)}{\sum_{j'=1}^{g} f\left(x_0|y = j'\right)\mathbb{P}\left(Y = j'\right)} \forall j \in \{1, ..., g\} \tag{9.1}$$

Les probabilités *a priori* des groupes/classes j, notée $\mathbb{P}(Y = j)$ sont *connues*. D'une manière générale nous n'avons pas une certitude sur celles-ci, par contre nous pouvons proposer de l'estimer à partir des fréquences de chaque groupe dans les observations $\{Y_i\}_i^n$[13].

Afin de spécifier le modèle de discrimination normale, nous allons supposer l'hypothèse de normalité ci après. Pour le cas linéaire (i.e. pour l'ALD), on suppose que la densité des variables explicatives dans chaque groupe suit une loi multi-normale de même matrice de variance Σ dans chacun des groupes :

$$f(x|y = j) \sim \mathcal{N}(\mu_j, \Sigma)$$

Une fois estimés les différents paramètres des lois normales, l'équation 9.1 nous permettra de connaître les probabilités d'affectation d'une nouvelle observation aux différents groupes. La prévision quant à elle sera logiquement donnée par le groupe le plus probable, c'est à dire :

$$j_0 = \underset{j \in \{1,...,g\}}{\arg\max}\, \mathbb{P}(Y = j|X = x_0) = \underset{j \in \{1,...,g\}}{\arg\max}\, f(x_0|y = j)\mathbb{P}(Y = j)$$

[13] Si les classes sont équilibrées, on peut définir les groupes comme équiprobable $\mathbb{P}(Y = j) = 1/g$

9.5.3.2 ALD et scores de classifications en pratique :

La construction du système doit nous permettre d'estimer les probabilités d'affectation d'une nouvelle observation aux différents groupes. La prévision d'appartenance serait logiquement donnée par le groupe le plus probable. Il faut se demander si nous sommes proches ou éloignés d'une frontière de décision entre classes.

Pour deux classes l et k, la frontière de décision est l'ensemble pour lequel les probabilités a posteriori $\mathbb{P}(Y = l|X = x)$ et $\mathbb{P}(Y = k|X = x)$ sont égales. Ce qui nous intéresse d'un point de vue de la classification est de savoir notre distance à la frontière de décisions et de quel coté nous sommes. Intéressons nous au cas où un individu sera classé dans la classe l i.e.

$$\mathbb{P}(Y = l|X = x) > \mathbb{P}(Y = k|X = x)$$

En réintroduisant l'équation 9.1 et en passant au log on obtient

$$x^T\Sigma^{-1}(\mu_l - \mu_k) + \log(\mathbb{P}(Y = l)) - \log(\mathbb{P}(Y = k)) - \frac{1}{2}\mu_l^T\Sigma^{-1}\mu_l + \frac{1}{2}\mu_k^T\Sigma^{-1}\mu_k > 0$$

Que l'on peut ré-écrire comme :

$$S(x) > s,$$

avec

$$S(x) = x^T\Sigma^{-1}(\mu_l - \mu_k)$$

et

$$s = \log(\mathbb{P}(Y = k)) - \log(\mathbb{P}(Y = l)) + \frac{1}{2}\mu_l^T\Sigma^{-1}\mu_l - \frac{1}{2}\mu_k^T\Sigma^{-1}\mu_k$$

$S(x)$ est la fonction score de l'analyse discriminante et s le seuil de discrimination. En première approche pour une observation x_0, si $S(x_0) > s$ alors son groupe d'appartenance sera l, k sinon.

Nous allons estimer les paramètres des lois normales ($\hat{\mu_j}$ et $\hat{\Sigma}$) à l'aide d'un ensemble d'entraînement ("training set") :

les moyennes par groupes $\hat{\mu_j}$ sont estimées par le centre de gravité de chacun des groupes

$$\hat{\mu_j} = \frac{1}{n_j}\sum_{i\in J} X_i,$$

où J est l'ensemble des indices d'observations qui sont dans le groupe j et n_j le nombre d'observations dans le groupe j (i.e.le cardinal de J).

la matrice de variance-covariance par

$$\hat{\Sigma} = \frac{1}{n-g}\sum_{j=1}^{g}\sum_{i\in J}(X_i - \hat{\mu_j})(X_i - \hat{\mu_j})^T$$

Les probabilités a priori $\mathbb{P}(Y = l)$ et $\mathbb{P}(Y = k)$ seront quant à elles estimées à partir des fréquences de chaque groupe dans les observations.

9.5.3.3 Expansion quadratique

Classiquement dans les méthodes d'apprentissage, l'*astuce du noyau* est un moyen de convertir un algorithme de classification linéaire en un non linéaire en projetant les données initiales dans un espace de dimension plus grand. Ainsi la classification linéaire dans l'espace agrandi est assimilable à une classification non linéaire dans l'espace d'origine. Un des exemples les plus usités aujourd'hui concerne les machines à vecteurs de support (i.e. M.V.S). Toutefois nous pouvons tout à fait l'utiliser pour tout type de classifieur linéaire.

Nous utiliserons, quant à nous, une «simple» **expansion quadratique** : pour chaque observation X, nous avons un vecteur de caractéristiques de taille p, nous étendons ce vecteur de p à $p*(p+1)/2$ en intégrant le carré et le produit croisé des caractéristiques de bases. Ainsi dans le cas de l'A.L.D, les fonctions de séparations linéaires dans l'espace étendu correspondront a des fonctions de séparations quadratiques dans l'espace de départ.

L'intérêt d'une telle extension sera présentée au travers de nos scores de classifications (cf. Section 9.5.5).

9.5.4 Entraînement et validation du système

D'une manière générale, la procédure de validation consiste à séparer de manière aléatoire les données en deux parties distinctes : d'apprentissage (y_a, X_a) et de test (y_v, X_v). Le modèle est construit avec le jeu d'apprentissage (voir Figure 9.16). Ensuite en utilisant le modèle et les variables explicatives X_v, on prédit les classes d'appartenances $\hat{y}_v$.

Pour estimer la qualité du modèle construit, on mesure l'*erreur* prévisions et observations sur ce sous jeu. Pour cela nous allons nous donner un critère. Un des plus classiques est de mesurer le nombre de mal classés (MC) :

$MC = \|\hat{y}_v - y_v\|_1$ où $\|x\|_1 = \Sigma_i |x_i|$ est la normeL_1
$\hat{y}_v$ valeur prédite sur le set de test
y_v valeur **vraie** (ie annotée)

Notons que l'erreur ainsi calculée est une surestimation de l'erreur que nous aurions pu produire en utilisant toutes les données disponibles pour l'entraînement, mais celle-ci n'aurait pas pu être mesurée.

Maintenant se pose la problématique de la faculté de *généralisation* du modèle. En effet, pour que cette démarche (seule) soit valide, il faut, d'une part, un nombre très important d'observations (vis-à-vis du nombre de paramètres à estimer) et, d'autre part, être certain que le découpage en ensemble d'entraînement et de test n'induise pas de biais. Un moyen bien connu pour pallier ces problèmes est d'utiliser la validation croisée.

9.5.4.1 Validation du système par «Validation croisée» :

La validation croisée (*Cross Validation*) n'est pas un outil d'analyse statistique à proprement parler mais une approche pour partager les données en sous-groupes d'entraînement (train) et de test. Le système d'apprentissage proposé est entraîné sur le sous-groupe d'entraînement puis l'erreur est calculée sur le sous groupe de test. Il faut donc veiller à ce que les deux groupes soient parfaitement indépendants.

L'ensemble des données est découpé en m sous-groupes (typiquement 10). Ils doivent être choisis de façon à ce que chaque sous groupe contienne le même nombre de représentants de chaque classe.

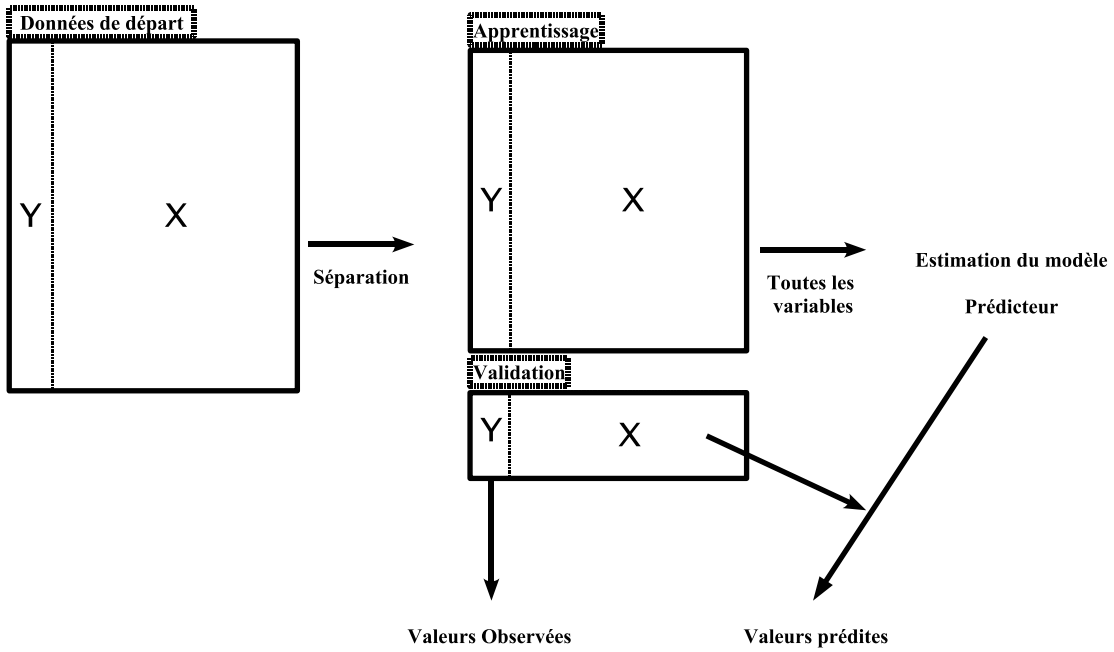


FIG. 9.16: Procédure d'apprentissage et validation.

Une certaine fraction des données est utilisée pour entraîner et le reste pour tester . Un des sous-groupes est écarté en vue d'être la base de test et le système est entraîné sur les $m-1$ autres groupes. Puis *l'erreur* commise sur le sous-groupe de test est évalué. Cette approche est répétée pour chacun des m groupes qui sert de test une et une seule fois.

Remarque 15 *Tous les systèmes présentés seront* validés *en utilisant le principe de validation croisée. Les tables de confusion présentées par la suite seront issues d'une des étapes de validation prise au hasard.*

9.5.4.2 Courbe Roc

Nous avons défini dans la Section 9.5.3.2, la fonction de score S et le seuil de discrimination s pour un ensemble de variables explicatives X. Nous pouvons donc avoir un renseignement sur le pouvoir discriminant d'un score, pour un seuil donné en définissant :

$$\alpha = \mathbb{P}(S(X) > s | Y = k) \text{ prévoir l alors que } Y = k$$
$$\beta = \mathbb{P}(S(X) <= s | Y = l) \text{ prévoir k alors que } Y = l$$

La courbe ROC (Receiver Operating Characteristic) est une courbe paramétrique ayant en abscisse $\beta(s)$ et en ordonnée $1-\alpha(s)$. Très basiquement, pour un s donné, plus $\beta(s)$ est faible et $1-\alpha(s)$ est fort, plus le résultat est satisfaisant.

Remarque 16 *Aussi un des intérêts est de pouvoir moduler pour un classifieur donné l'importance des faux négatifs vis-à-vis des faux positifs.*

9.5.5 Jeux de paramètres et scores de classification

Dans les paragraphes précédents, nous avons présenté les différents descripteurs, la méthode de classification utilisée ainsi que le processus d'entraînement. Nous présentons ci-après les résultats pour différents jeux de descripteurs.

Description des jeux de descripteurs Dans la Section 9.5.2, nous avons présenté et discuté des avantages et des inconvénients des descripteurs retenus pour la discrimination lettre/non lettres. Nous allons ici en sélectionner des sous-ensembles pour la réalisation de trois classifieurs.

Le premier à être mis en place comporte 27 descripteurs de base (406 après expansion quadratique, il est défini dans le Tableau 9.4. La table de confusion du système avec et sans expansion quadratique est présenté dans le Tableau 9.7. On observe un taux moyen d'erreur de l'ordre de 10%. Nous allons définir deux autres jeux de descripteurs et voir si nous obtenons de meilleurs résultats.

Le second jeu de descripteurs suit les remarques sur l'ambiguïté des mesures de l'épaisseur des traits (cf Section 9.5.2). Nous pensions que l'utilisation d'un squelette serait plus pertinente, nous avons donc remplacé, les différentes mesures basées sur les longueurs de plages par celles issues du squelette. La description de ce jeu est présenté dans le Tableau 9.5. La table de confusion (cf. Tableau 9.8) donne de moins bons résultats que le premier jeu de paramètres, indiquant que les mesures basées sur les longueurs de plages sont plus discriminantes bien que plus biaisées.

Enfin, un dernier jeu de paramètres, correspondant à l'union des deux jeux précédents est défini (cf Tableau 9.6). Bien que les taux de classifications soient meilleurs (cf. Tableau 9.9) que le jeu utilisant les mesures issues du squelette, ils sont inférieurs[14] l'erreur commise à ceux obtenus avec le premier jeu de paramètres.

# Nombre	Type	Descripteur
11	Caractéristiques des traits	$Max_{DistFunc}$,RLE_{MedX},RLE_{MedY} RLE_{MoyX},RLE_{MoyY},RLE_{VarX},RLE_{VarY} RLE_{MoyX},RLE_{MoyY},RLE_{VarX},RLE_{VarY} $RLE_{Delta_{MoyX}}$,$RLE_{Delta_{MoyY}}$ $RLE_{Delta_{VarX}}$,$RLE_{Delta_{VarY}}$
7	Géométrie	h, w, A (Hauteur, largeur Aire de la composante) **et leur inverse** surface de la boîte englobante A_{Box}
7	Régularité	E (Nombre d'Euler), Périmètre de la composante Périmètre de la composante après bouchage de trous CC, CX, $CC_{FillHoles}$ et $CX_{FillHoles}$
2	Contraste	C_1,C_2
27	Total	

TAB. 9.4: Description du jeu de paramètres I.

Dans la sous-section suivante, nous allons réaliser une étude plus fine du «meilleur» jeu de descripteurs.

[14]Il faudrait regarder les taux d'erreurs sur le jeu d'entraînement, pour vérifier si nous ne somme pas dans un cas de «sur-apprentissage».

# Nombre	Type	Descripteur
3	Caractéristiques des traits	$Max_{DistFunc}$, Moy_{skel} Var_{skel}
7	Géométrie	Idem au jeu I
7	Régularité	Idem au jeu I
2	Contraste	Idem au jeu I
19	Total	

TAB. 9.5: Description du jeu de paramètres II.

# Nombre	Type	Descripteur
13	Caractéristiques des traits	∪(Jeu I,Jeu2)
7	Géométrie	Idem au jeu I
7	Régularité	Idem au jeu I
2	Contraste	Idem au jeu I
29	Total	

TAB. 9.6: Description du jeu de paramètres III.

Détection / Vérité Terrain	lettres	Non Lettres
lettres	83.1 %	16.9 %
Non Lettres	13.3%	86.7%
Avec expansion quadratique		
Détection / Vérité Terrain	lettres	Non Lettres
lettres	89.1%	10.9%
Non Lettres	9.7 %	90.3 %

TAB. 9.7: Table de confusion du système d'apprentissage : jeu de paramètres I.

Détection / Vérité Terrain	lettres	Non Lettres
lettres	77.3%	22.7%
Non Lettres	27.6%	72.4%
Avec expansion quadratique		
Détection / Vérité Terrain	lettres	Non Lettres
lettres	82.5 %	17.5 %
Non Lettres	12.6 %	87.4%

TAB. 9.8: Table de confusion du système d'apprentissage : jeu de paramètres II.

Détection / Vérité Terrain	lettres	Non Lettres
lettres	78.7%	21.3%
Non Lettres 27.1	%	72.9%
Avec expansion quadratique		
Détection / Vérité Terrain	lettres	Non Lettres
lettres	83.4 %	16.6 %
Non Lettres	10.8 %	89.2 %

TAB. 9.9: Table de confusion du système d'apprentissage : jeu de paramètres III.

9.5.6 Étude du jeu de descripteurs 1

Pour compléter les scores de classifications proposés en Tableau 9.7, nous allons à présent étudier plus finement le jeu de descripteur 1. Nous réaliserons dans un premier temps une étude de l'inter-corrélation des descripteurs et de leur contribution à la variance de l'ensemble du jeu de données (i.e. lettres/non lettres mélangées) puis nous nous intéresserons à leur contribution en terme de classification).

Remarque 17 *Prise en compte de la balance des classes : l'ensemble du jeu de données étant constitué de 5% de «lettre» pour 95% de «non lettre», les études qui suivent sont réalisées en utilisant toujours toutes les données «lettre» et un ensemble de même taille sélectionné aléatoirement dans l'ensemble des données «non lettre».*

9.5.6.1 Contribution des descripteurs pour la caractérisation du jeu de données

Pour analyser cette contribution nous allons d'une part regarder les corrélations existantes entre les différents paramètres puis réaliser une analyse en composantes principales du jeu de données.

Matrice de corrélation Cette matrice calculée pour les 27 descripteurs de bases (i.e. sans expansion quadratique) est présentée en Figure 9.17.

On peut en extraire quelques grandes tendances :

- On retrouve, bien évidemment les corrélations inhérentes aux descripteurs utilisés : ainsi on observe bien le lien linéaire entre les paramètres de forme CX et CC et les mesures les constituant (i.e. périmètre et aire).
- On observe une corrélation entre le périmètre et les descripteurs géométriques (i.e. Hauteur, Largeur, Aire et Aire de la boîte englobante).
- Les descripteurs qui ont été calculés avec et sans bouchage de trous, i.e. Périmètre et facteur de forme CX et CC sont fortement corrélés
- Les mesures issues des différences de longueurs de plages et les paramètres de contrastes ne sont pas corrélées avec d'autres descripteurs.

Enfin, si l'on regarde la matrice de corrélation dans sa globalité et si l'on passe outre les mises en lumière de corrélations triviales entre descripteurs, on peut observer que nos descripteurs ne semblent pas trop corrélés (valeurs de 0.5 à -0.5). Ces constatations sont d'ordre général et ne nous permettent pas de conclure sur la pertinence de tel ou tel paramètre pour la classification entre «lettre» et «non lettre».

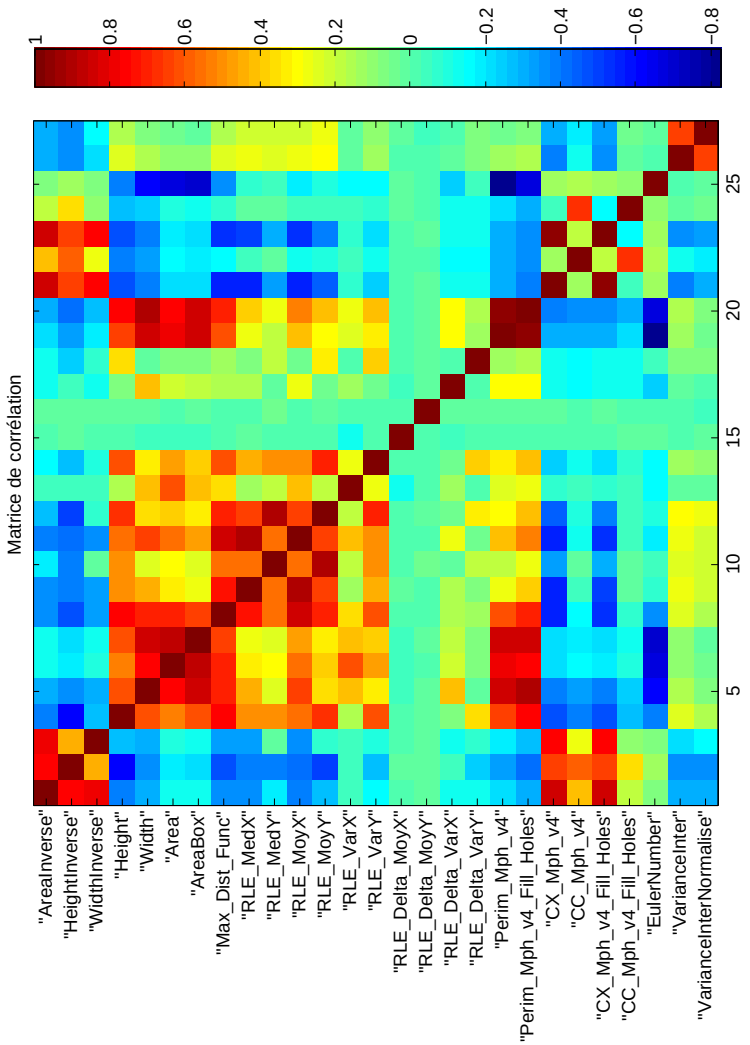


FIG. 9.17: Matrice de corrélation pour le jeu de descripteur 1

Étude plus fine des corrélations entre descripteurs : Existe-t-il un couple de descripteurs simples permettant de séparer les classes lettre et non lettre ?

Pour le vérifier nous avons projeté les données lettres/non lettres pour un sous-ensemble de couples de descripteurs. Précisément nous avons tracé les descripteurs de l'épaisseur et de la cohérence de celle-ci, ceux de contraste et de régularités en fonction des descripteurs géométriques globaux (i.e. Hauteur, Largeur, Aire et Aire de la boite englobante)[15]. Ces tracés sont synthétisés sur les figures 9.18, 9.19, 9.20 et 9.21.

Là encore, nous pouvons proposer quelques remarques d'ordre général. On retrouve fort logiquement la dépendance du périmètre (cf. Figure 9.20) vis-à-vis des descripteurs géométriques globaux (le périmètre aura tendance à croître en fonction de la «taille» de la composante étudiée). Aucun des descripteurs reliés à l'estimation de l'épaisseur et aux propriétés de régularité des lettres ne peut être simplement couplé avec un descripteur géométrique global pour séparer les classes lettres et non lettres (cf. Figures 9.18, 9.19, 9.20, 9.21). Ce qui ne remet pas en cause l'utilisation des descripteurs géométriques globaux comme facteur de normalisation. Enfin, un descripteur se détache du lot, il s'agit de la variance interclasse normalisée (i.e. descripteur de contraste $C2$, cf. Figure 9.21), pour lequel il semble apparaître une frontière de séparation entre «lettre» et «non lettre».

Analyse en composantes principales : Nous avons calculé une A.C.P sur l'ensemble de notre jeu de données. Le premier graphique (c.f. Figure 9.22) nous permet d'estimer la proportion de la variance totale contenue dans chaque axe. Nous observons que le premier axe contient moins de 35% de la variance totale, et qu'il faut additionner la contribution des 15 premiers axes pour atteindre 95% de celle-ci. Ceci permet de confirmer notre première analyse montrant que nos paramètres sont relativement indépendants les uns des autres.

Pour une étude plus fine, nous présentons sur la Figure 9.23, la contribution de chaque paramètre à la variance contenue dans chacun des axes de l'A.C.P. De manière générale on peut observer que tous les descripteurs contribuent à la variance dans un facteur de 1 à 2.

En conclusion. Nous avons mis en évidence une relative faible interdépendance entre nos descripteurs. Nous avons également remarqué : 1. la forte corrélation des descripteurs que nous avons calculés avec et sans bouchage de trous (i.e. Périmètre, CX,CC) ; 2. la capacité de discrimination du descripteur de contraste $C2$ (i.e. la variance interclasse normalisée).

Dans la section suivante, nous nous intéresserons à la contribution des descripteurs en termes *de classification*.

9.5.6.2 Contribution des descripteurs pour la classification

Quels sont les descripteurs qui participent le plus à la faculté de discrimination du jeu de paramètres que nous avons sélectionné ?

Pour réaliser celle-ci, nous allons regarder la contribution de chacun à la fonction de prédiction S (cf. Section 9.5.3.2) de notre classifieur . Celui-ci a été réalisé à l'aide du jeu de paramètres I et en utilisant l'expansion quadratique, ce qui représente un total de 406 descripteurs. Pour faciliter l'analyse des résultats, pour chacun des 27 descripteurs, nous avons sommé les contributions de tous les produits dans lesquels il est impliqué.

La Figure 9.24 présente les résultats de cette intégration. On observe tout d'abord que plusieurs paramètres contribuent fortement à la discrimination entre lettre et non lettre. Cela confirme à la fois la pertinence des descripteurs choisis et la non trivialité du problème de reconnaissance des caractères.

[15] La description des familles de descripteurs est proposée en Section 9.5.2 p.149

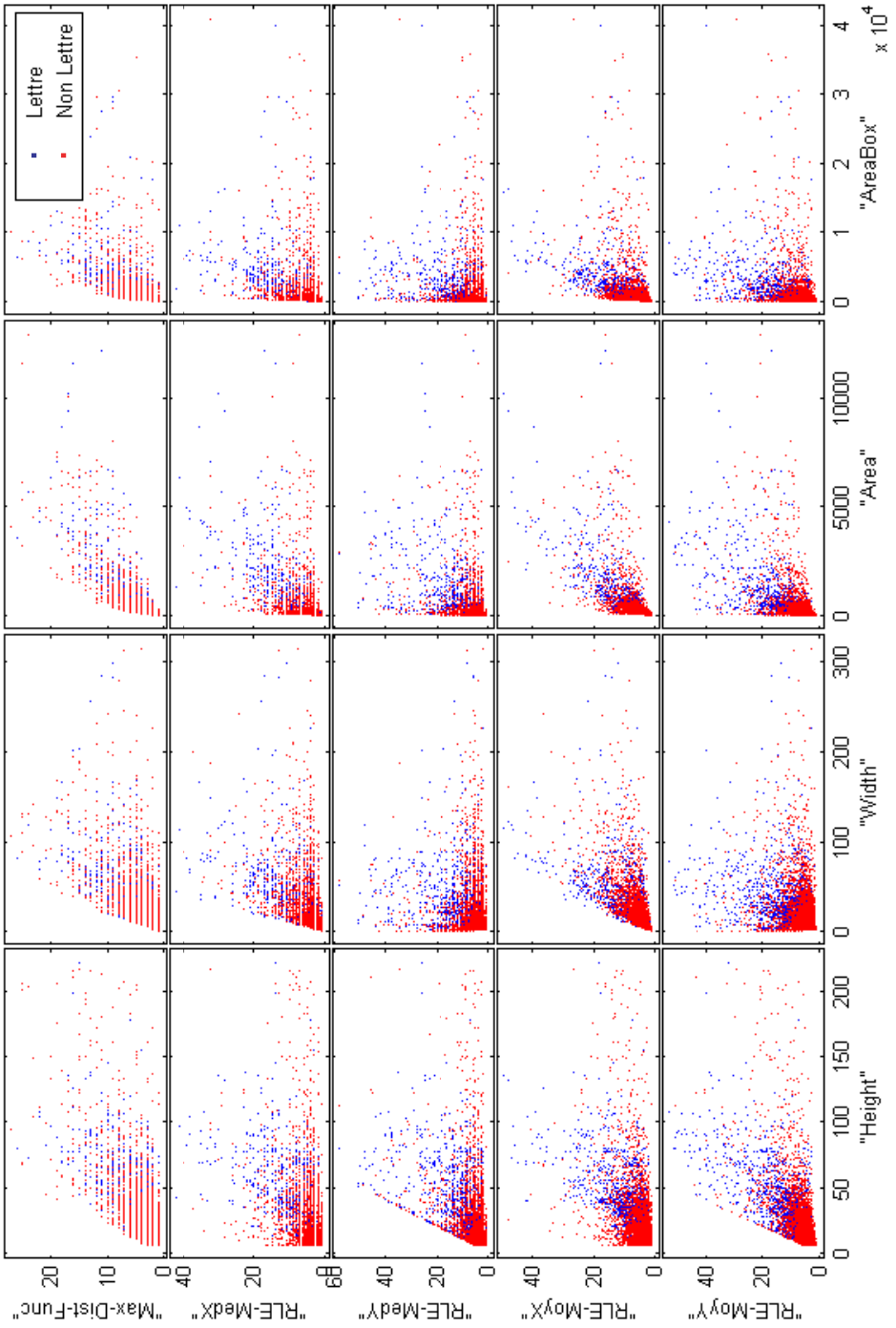


FIG. 9.18: Projection des données «lettre», «non lettre» par couple de descripteurs (1/4 : première figure sur quatre).

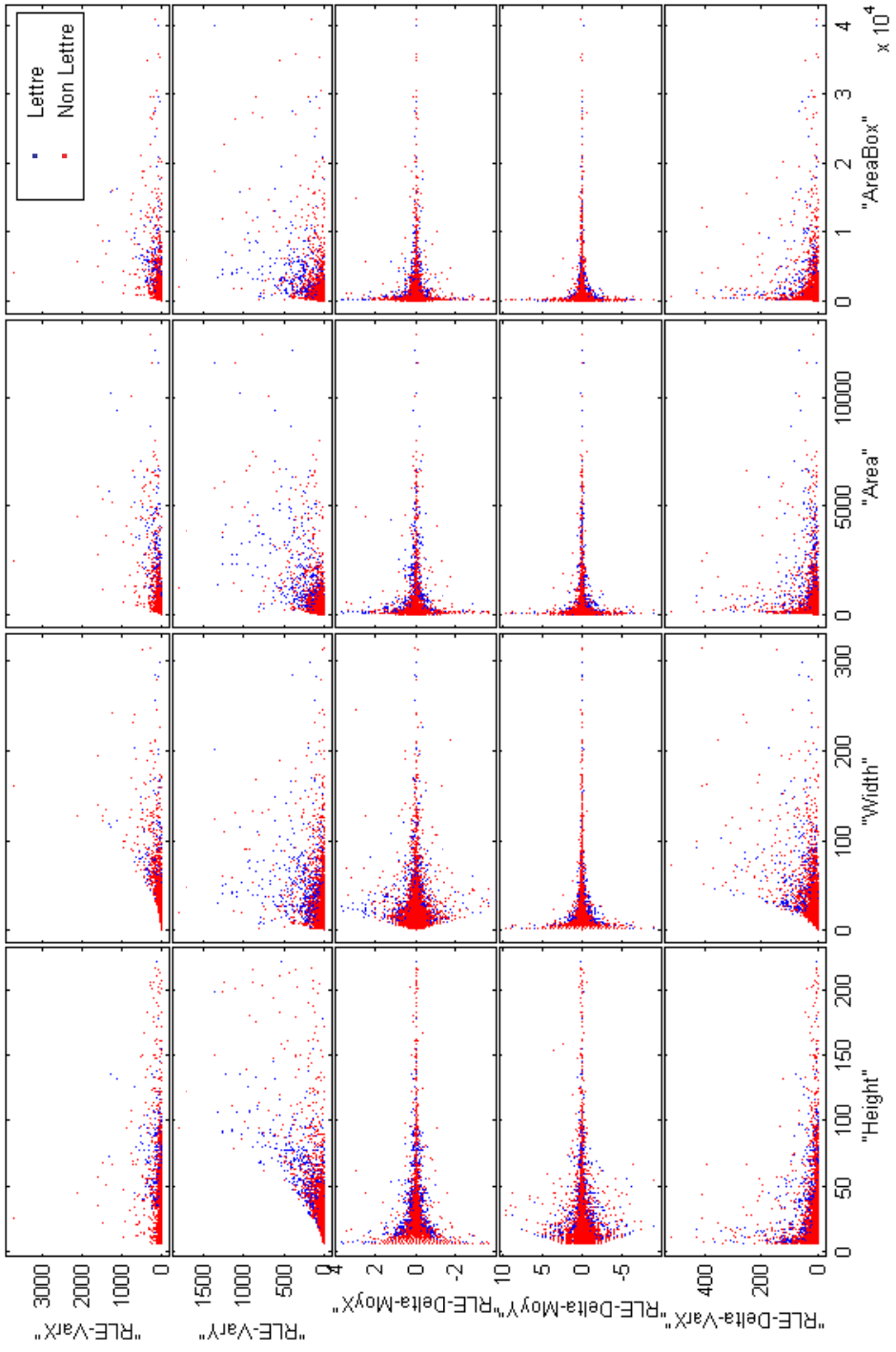


FIG. 9.19: Projection des données «lettre», «non lettres» par couple de descripteurs (2/4).

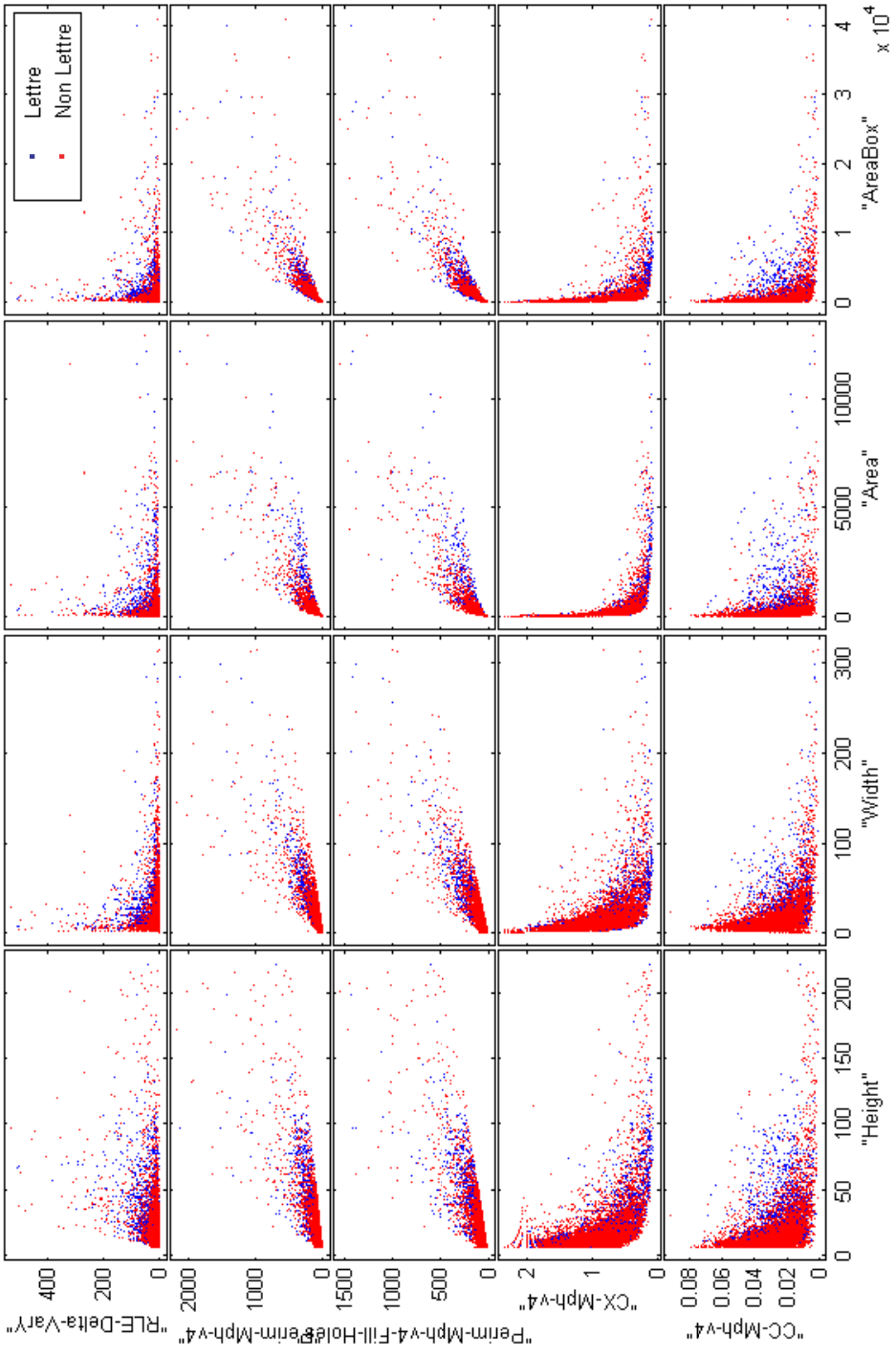


FIG. 9.20: Projection des données «lettre», «non lettre» par couple de descripteurs (3/4).

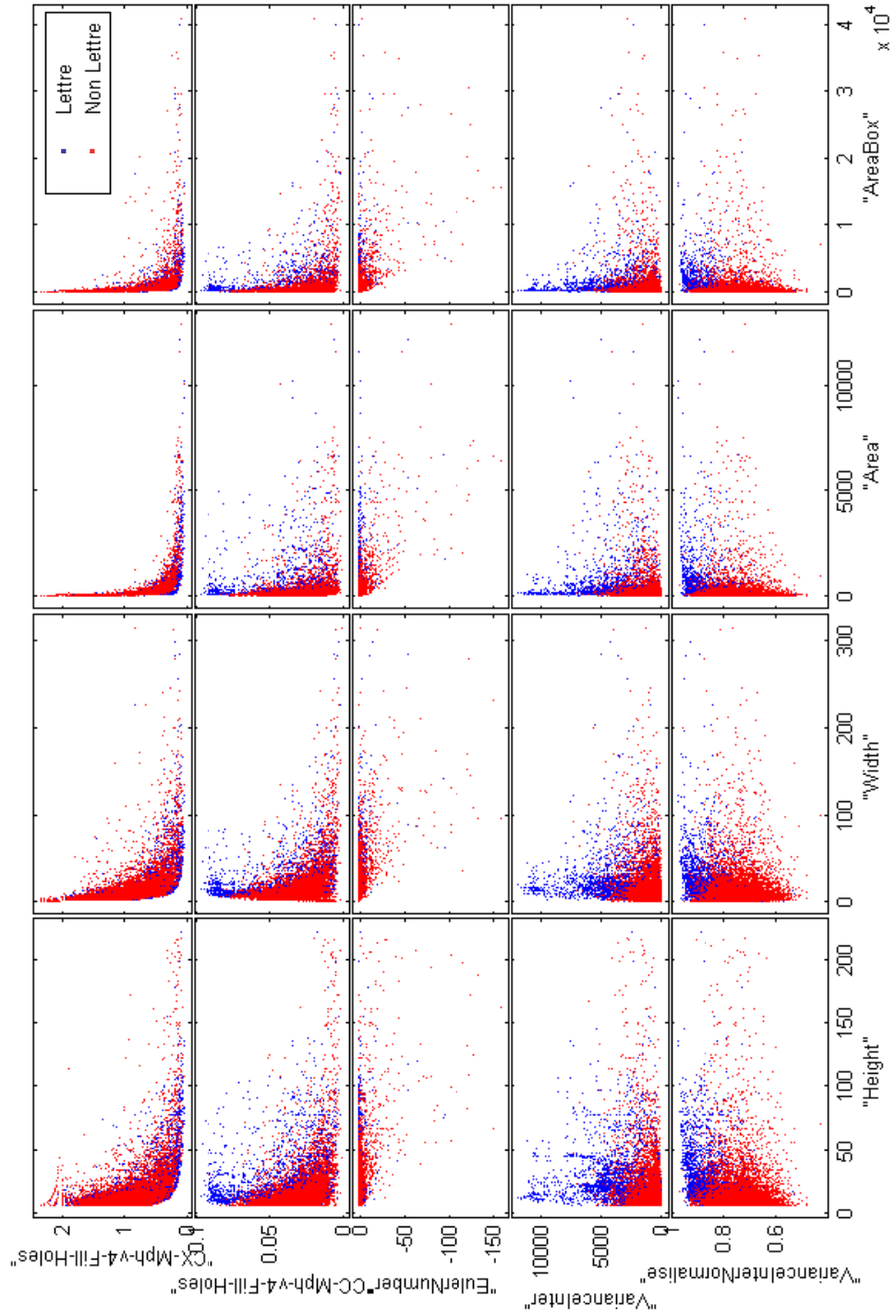


FIG. 9.21: Projection des données «lettre», «non lettre» par couple de descripteurs (4/4).

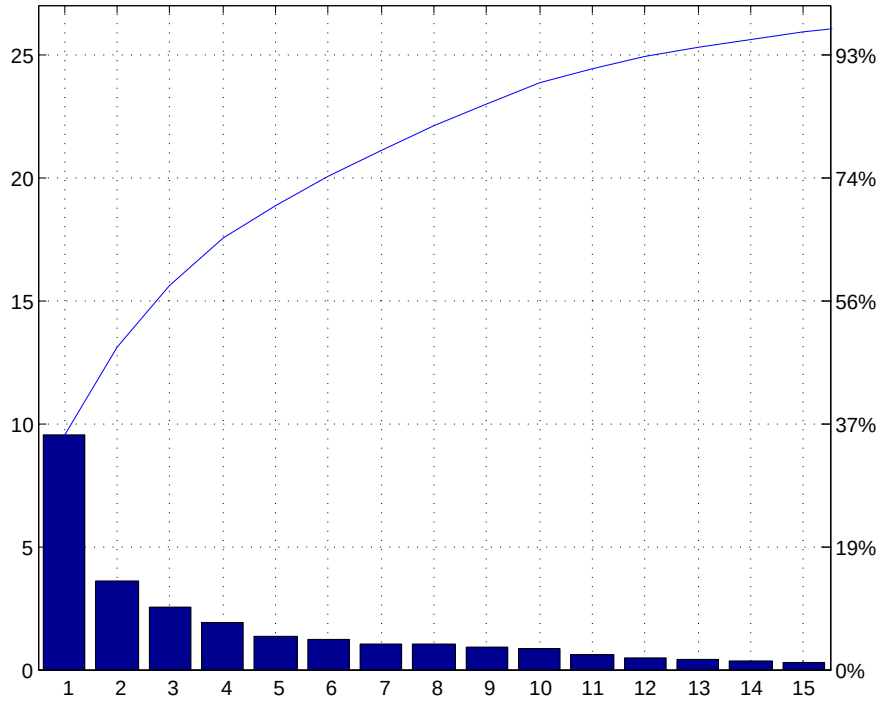


FIG. 9.22: Contribution de chaque axe de l'A.C.P à la variance totale du jeu de données

On observe que dans les descripteurs contribuant le plus, la variance interclasse normalisée n'est pas celle qui est la plus importante. En effet, les facteurs de forme CX et CC avec et sans bouchage de trous présentent la plus forte contribution.

Pour mieux comprendre ce phénomène, nous avons projeté les données lettres et non lettres pour les 5 descripteurs les plus discriminants (précisément nous avons projeté les données par couples de descripteurs). Les graphiques produits sont présentés en Figure 9.25.

On observe clairement la faculté de discrimination de la variance interclasse normalisée, mais elle ne contribue pas seule à la faculté de discrimination du classifieur. Fait très intéressant, le couple formé par le calcul de CX avec et sans bouchages de trous pour lequel nous avions mesuré une forte corrélation dans la section précédente, s'avère finalement avoir un fort pouvoir de discrimination.

Il serait intéressant de remonter à l'annotation que nous avons réalisé et regarder si il existe des explications intuitives à ce pouvoir de discrimination.

9.5.7 Courbe ROC du Jeu de descripteurs I

Pour le «meilleur» jeu de descripteurs nous proposons en Figure 9.26, la courbe ROC [16] du classifieur obtenu. Nous pourrions utiliser celle-ci pour favoriser le taux de lettres bien classées au détriment d'un plus grand nombre de faux négatifs (et inversement).

Cependant, nous ne devons pas oublier les données sur lesquelles nous travaillons. En effet nous n'avons pas simplement à filtrer les non lettres. Nous devons ensuite regrouper les lettres pour former les boîtes de texte. Nous allons discuter dans la Section 9.5.8, les différents cas de figure qui se présenteront lors de l'application de ce filtrage.

[16] définition en Section 9.5.4.2, p. 160

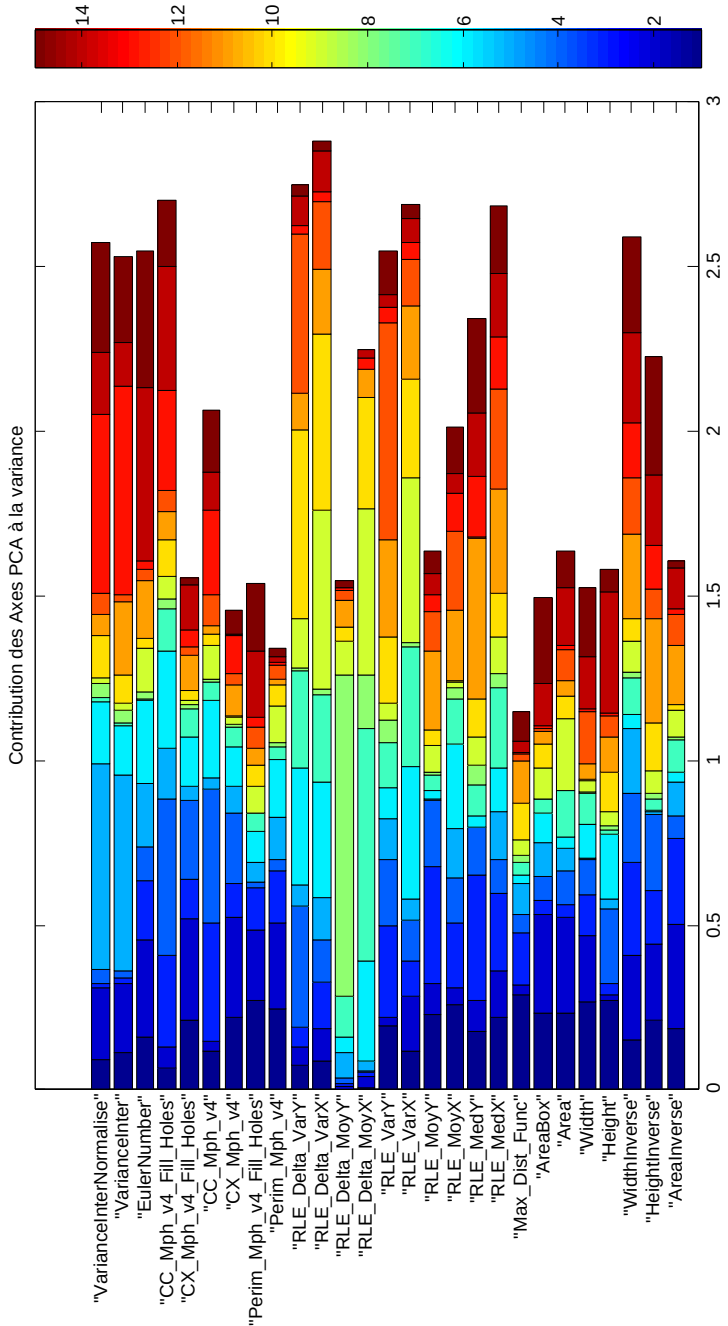


FIG. 9.23: Contribution des paramètres à la variance contenue dans les 15 premiers axes de l'A.C.P.

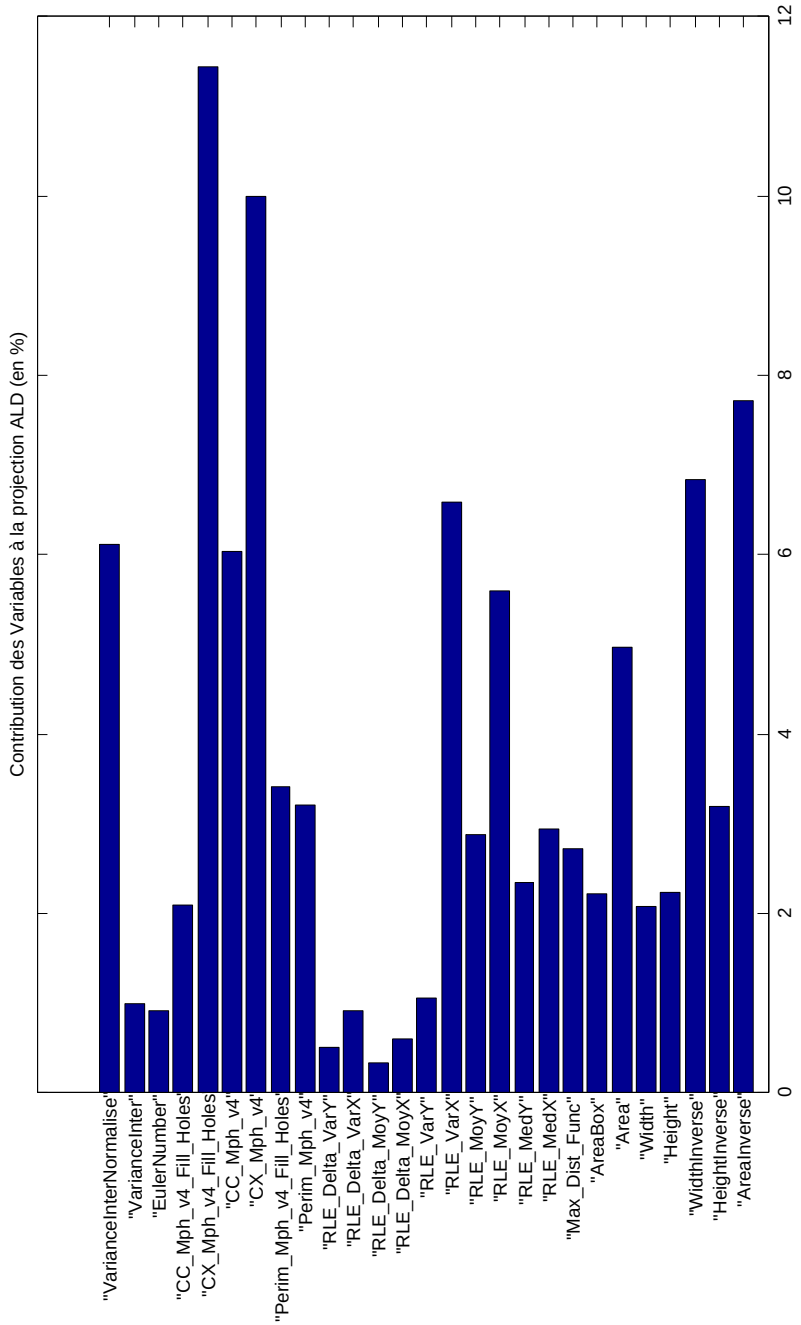


FIG. 9.24: Contribution des descripteurs dans la fonction de prédiction du classifieur, pour le jeu de paramètres I, en utilisant l'expansion quadratique.

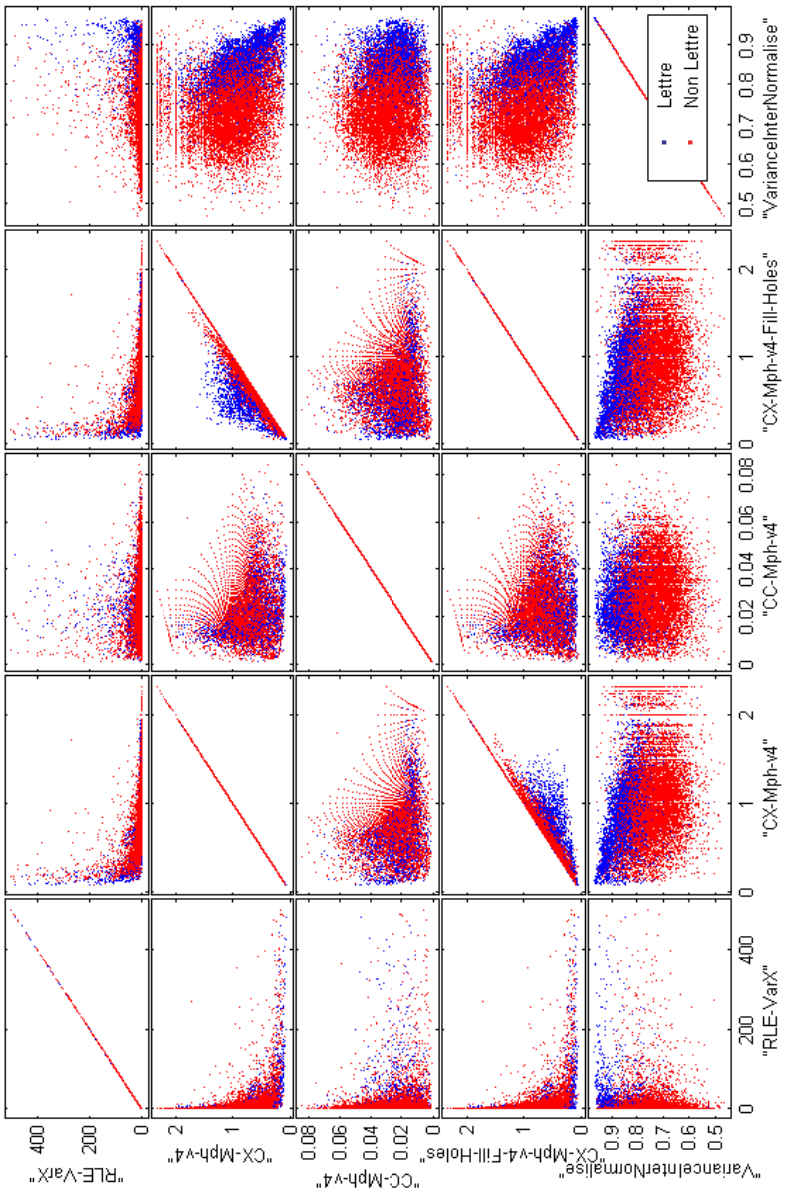


FIG. 9.25: Projection des données «lettre», «non lettres» par couples pour les cinq descripteurs contribuant le plus à la fonction de discrimination.

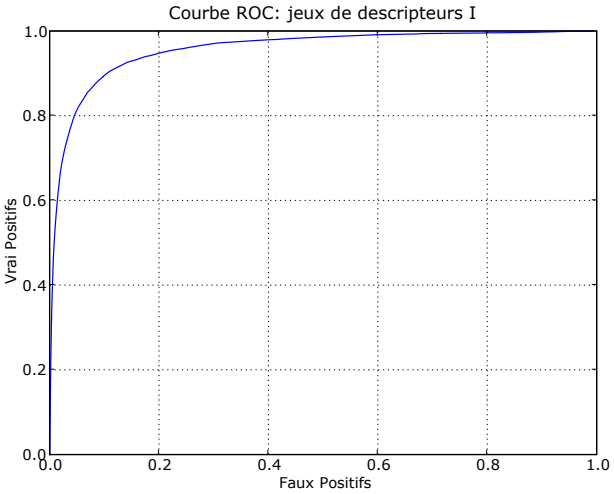


FIG. 9.26: Score du meilleur classifieur. Jeux de paramètres I avec expansion quadratique .

9.5.8 Quelques résultats en images du module de filtrage par apprentissage

Sur la Figure 9.27 sont représentés différents types de résultats obtenus grâce à ce module :

1. Cas le plus favorable (cf Figure 9.27(c)) : bien qu'il reste quelques faux positifs, le module d'apprentissage a conservé tous les caractères en supprimant un grand nombre de composantes connexes.

2. Deux cas problématiques :

   a) Faux Positifs (cf Figure 9.27(c)) : l'image ne contient aucun caractère, cependant le module d'apprentissage a conservé un nombre important de faux positifs.

   b) Faux Négatifs (cf Figure 9.27(c)) : le module a bien filtré l'image mais a malheureusement invalidé le "I" de la chaîne "OUI"

Dans le cas le plus favorable un module de regroupement spatial (voir Section 9.6) permettra de regrouper les caractères pour former des mots/phrases et invalidera les dernières composantes parasites.

Les deux cas problématiques sont à relier aux scores de classification. En effet le classifieur généré tendra à conserver des *non-caractères*, et à supprimer certaines lettres avec un taux de 10% dans chacun des cas. On pourrait induire que cette erreur est connue et donc partiellement maitrisable. Cependant la remarque suivante tempère nos espérances :

Les mauvaises classifications n'ont pas un impact toujours prévisible. En effet ; nous devons par la suite, regrouper les caractères en mots/phrases, car, c'est à cette échelle que l'efficacité des systèmes de localisation est mesurée. Ainsi l'impact de la perte d'une ou plusieurs lettres ou la présence de faux positifs n'est pas corrélée **linéairement** au résultat final.

Généralement on peut s'attendre (comme résultats de ce module) à un mélange des trois cas de figures précédents : un filtrage efficace d'un grand nombre de composantes non lettres avec en contre partie quelques faux positifs et faux négatifs.

(a) Image Originale

(b) Image des composantes avant filtrage

(c) Image des composantes après filtrage

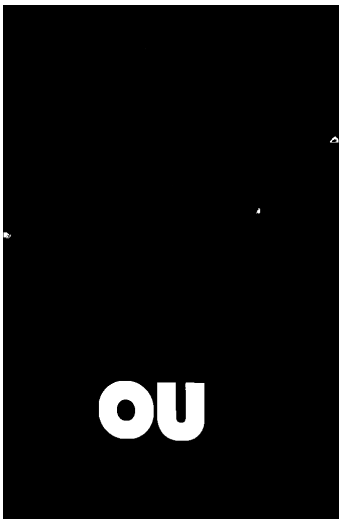
(d) Image Originale

(e) Image des composantes avant filtrage

(f) Image des composantes après filtrage

(g) Image Originale

(h) Image des composantes avant filtrage



(i) Image des composantes après filtrage

FIG. 9.27: Présentation de l'apport du filtrage par apprentissage (Premier jeu de descripteurs). Présentation de différents cas de figure : en haut filtrage satisfaisant, au milieu des faux positifs persistants, en bas un faux négatif. (Images issues de la campagne *Test à blanc* $imagEval$).

9.5.9 Conclusion et perspectives

Conclusion Nous venons de définir un module de classification lettres/non lettres utilisant une A.L.D. Malgré la très grande variabilité au sein du jeu de donnée nous obtenons des résultats de discriminations satisfaisants, avec un taux d'erreur moyen de 10%. Ce classifieur sera utilisé sur différentes bases d'images dans le Chapitre 10, p.185, ce qui permettra de valider «applicativement» sa capacité de généralisation.

De nombreuses améliorations et/ou refontes de ce modules sont envisageables :

1. Variables explicatives :

   Il semblerait curieux que les descripteurs que nous avons présentés soient pertinents quelle que soit l'«échelle» des lettres . Nous devrions d'une part définir des descripteurs spécifiques aux cas des petites lettres et d'autre part essayer des facteurs de formes plus classiques pour les «grandes lettres» (eg : descripteur de Fourier, moments).

2. Autres systèmes d'apprentissage :

   Nous avons décrit un système basé sur une A.L.D quadratique qui a l'avantage d'être une approche simple et robuste. Mais nous pourrions nous tourner vers des systèmes plus complexes comme les M.V.S et regarder si nous n'augmentons pas notre qualité de discrimination. Nous pourrions également définir notre système sous la forme d'une cascade de classifieurs (cf [17]) dans le but cette fois d'accélérer les temps de traitement.

3. Prise en compte de la relation spatiale :

   Enfin un axe important de progrès serait la prise en compte du «contexte». En effet nous nous contentons pour le moment de classifier lettre et non lettre, sans exploiter la cohérence spatiale des mots ou phrases constituant une zone de texte. Or comme nous l'avons montré nous ne pouvons pas prédire l'impact d'une mauvaise classification sur le résultat final de la chaîne de traitement.

9.6 Regroupement itératif des composantes

A la sortie du module de filtrage par apprentissage, seul un sous-ensemble des composantes a été conservé. Dans ce module nous allons les regrouper pour former les zones de texte.

Une des hypothèses les plus communes de la littérature, suppose que les lettres composant les mots viennent en *grappes* et qu'il faut un minimum de trois lettres. Nous ne dérogerons pas à cette règle.

9.6.1 Description détaillée du module

Pour effectuer cette étape d'agrégation des composantes, nous allons utiliser un traitement itératif. En effet, le module de filtrage par apprentissage a conservé des composantes qui ne sont pas des lettres et invalidé certaines lettres (un taux d'erreur de 10% pour chacun des cas cf Section 9.5).

Nous allons donc procéder à une première étape très restrictive de regroupement des composantes pour la formation de zones de texte embryonnaires (cette approche par "germes" s'inspire des travaux de [108]) puis nous relâcherons les contraintes pour former les zones de texte finales.

Les différentes étapes sont explicitées ci-dessous, un schéma récapitulatif est proposé sur la Figure 9.28 et illustré en image sur la Figure 9.31.

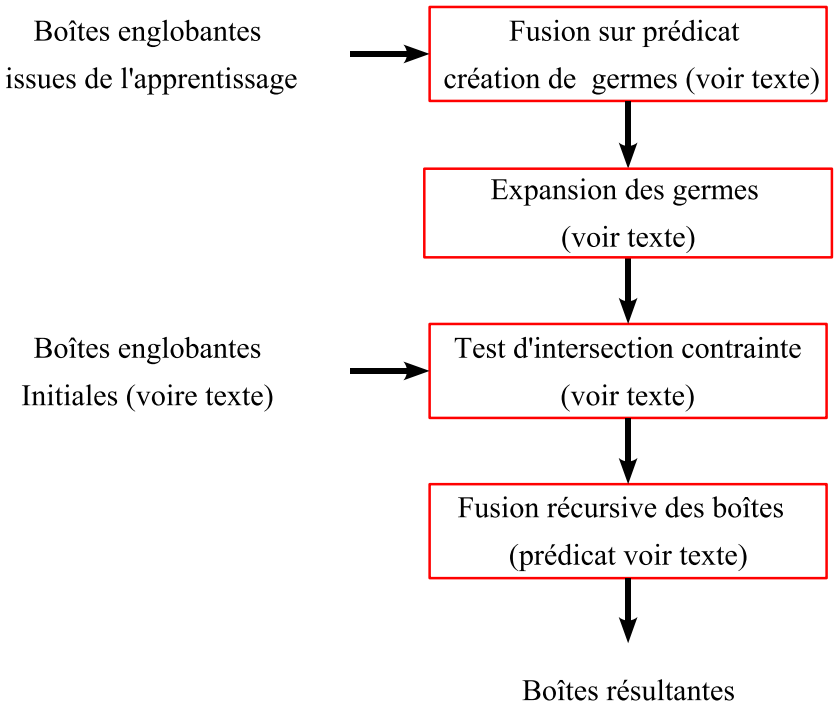


FIG. 9.28: Schéma récapitulatif du module itératif de regroupement des composantes connexes.

– Nous faisons l'hypothèse qu'une zone de texte est constituée d'au moins trois caractères ayant des contraintes de relation spatiales communes. Nous partons de la liste des boîtes englobantes validées par l'apprentissage :

1. Nous commençons par définir le critère de fusion entre deux boîtes. C'est une étape délicate car on ne peut pas imposer une contrainte d'horizontalité forte des zones de texte. Nous le définirons comme suit en nous aidant de la Figure 9.29 :

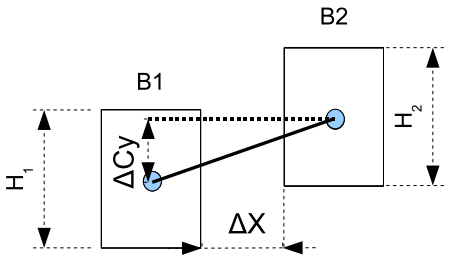


FIG. 9.29: Caractéristiques géométriques utilisées pour la fusion de boîtes

Deux boîtes peuvent fusionner si :

a) Leurs dimensions verticales sont cohérentes

$$\frac{|H_1 - H_2|}{\wedge(H_1, H_2)} < S_1 \text{ avec } S_1 = 1.1$$

ici on n'autorise pas la fusion entre une boîte et une autre plus de deux fois plus grande.

b) Elles respectent un alignement relatif

$$\frac{\Delta C_Y}{\wedge(H_1, H_2)} < S_2 \text{ avec } S_2 = 0.7$$

Les lettres ne sont pas forcément alignées horizontalement. Nous optons donc pour une distance verticale entre les centres des boîtes qui n'est pas trop stricte.

c) La distance horizontale entre les lettres est cohérente

$$\frac{\Delta X}{\wedge(H_1, H_2)} < S_3 \text{ avec } S_3 = 1$$

Par contre nous imposons que la distance horizontale entre les caractères ne soit pas plus importante que la plus petite des hauteurs de chacune des boîtes.

Les paramètres S_1,S_2 et S_3 ont été fixés expérimentalement.

2. Nous testons ce prédicat de fusion sur l'ensemble des boîtes englobantes issues de l'apprentissage. Pour qu'une boîte soit conservée elle doit valider ce prédicat avec au minimum deux autres boîtes. Si tel est le cas, la boîte testée, ainsi que les boîtes participant à la fusion, sont conservées. La Figure 9.31(a) illustre les germes formés.

– Les zones formées précédemment ont une forte probabilité d'être des zones de texte. Elles vont servir de germes pour cette seconde étape. La Figure 9.30 va nous permettre de mieux saisir la stratégie adoptée. Une voie est d'étendre ces germes d'un certain facteur (nommé ici $Size_{Expand}$) et de regarder les composantes connexes qui intersectent ces zones.
Ces composantes connexes sont potentiellement des caractères qui ont été perdus lors du filtrage par apprentissage ou lors de la création des germes. Un schéma simple consisterait à re-valider ces zones par un test d'intersection puis d'appliquer un prédicat similaire au précédent[17], pour tester l'adéquation des géométries des germes et des composantes connexes re-validées. Cette première manière est correcte dans le cas où les germes sont composés de lettres horizontales, cette configuration est présentée sur la Figure 9.30(a).
Mais nous pouvons avoir agrégé un germe de lettres possédant une certaine inclinaison (cf Figure 9.30(b)), en ce cas le germe n'est pas représentatif des boîtes qu'il contient. La stratégie précédente échoue doublement : 1) les zones de lettres ne sont pas récupérées 2) une zone de non lettres et n'ayant absolument pas les propriétés géométriques ayant amené à la formation du germe est validée !
Pour résoudre ce problème, nous avons adopté une approche *pragmatique*. L'erreur initiale est de considérer que la géométrie du germe est caractéristique des lettres qu'il contient. Pour corriger cette erreur nous proposons de calculer pour chaque germe la hauteur moyenne des boîtes le constituant. Puis nous partons des boîtes situées aux extrémités du germe auxquelles nous affectons cette hauteur moyenne et que nous étendons du facteur $Size_{Expand}$. L'ensemble de cette démarche est présenté sur la Figure 9.30(c).

1. Toutes les composantes connexes[18] qui intersectent et qui valident le prédicat de fusion avec les boîtes étendues sont validées.

[17] mais ne testant pas la distance entre les lettres
[18] Comprenant celles invalidées par l'apprentissage

2. L'étape précédente a pu re-valider des composantes connexes isolées. Nous appliquons donc le prédicat de fusion entre les boîtes re-validées. Nous obtenons un nouvel ensemble de zones de texte potentielles. Celui-ci est simplement ajouté à la liste des germes.

– Enfin une dernière étape de fusion récursive des boîtes résultantes peut être appliquée. Il existe principalement deux options qui dépendront en grande partie de la *granularité* de la vérité terrain des bases traitées.

1. Un prédicat simple d'inclusion, pour finir de former les zones de texte et éviter par là même de faire chuter les scores de localisation [19].
2. Un prédicat spécifique proposé par les détenteurs de la bases d'images traitée. C'est notamment le cas pour la campagne d'évaluation $imagEval$.

– Certaines chaînes de traitement traitant séparément les deux polarités de texte, une dernière étape peut être nécessaire : s'il existe une intersection non nulle pour des boîtes issues de polarités différentes elles sont simplement fusionnées.

La liste finale de boîtes englobantes constitue le résultat de l'algorithme. Cette liste sera comparée à une vérité terrain. Les résultats de cette confrontation fourniront les résultats quantitatifs d'une chaîne de localisation.

La Figure 9.31 illustre l'ensemble de ces étapes pour un cas réel, nous constatons que 1) la création des germes de trois caractères a permis de supprimer nombre de composantes non lettres 2) que le module de re-validation nous permet de récupérer la chaîne "17" sans pour autant créer de faux positifs.

9.6.2 Discussion sur quelques cas problématiques

Comme pour les modules précédents nous allons détailler ici quelques faiblesses de l'approche proposée ; des pistes d'amélioration seront proposées dans la section Section 9.6.3.

La faiblesse la plus flagrante est la dépendance de ce module à la qualité du filtrage par apprentissage précédent (cf Section 9.5). Deux cas antinomiques peuvent se produire :

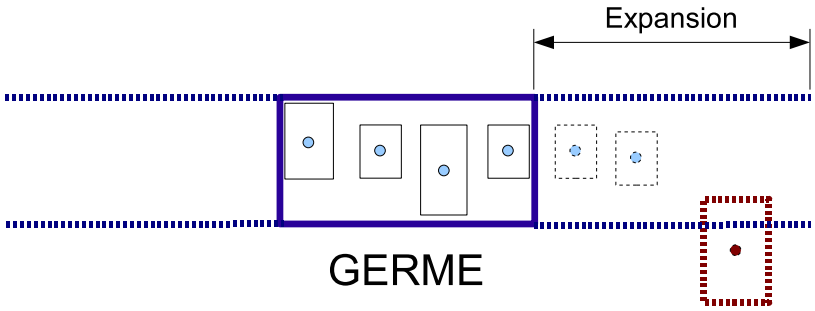
1. Le filtrage n'a pas validé suffisamment de composantes texte pour créer les germes. Ceci est illustré sur la Figure 9.27(i). Le module de filtrage n'a conservé que deux des lettres de la chaîne "OUI", ceci est insuffisant pour la création d'un germes de trois lettres. En ce cas le système ne fournira aucun résultat.
2. Le filtrage n'a pas invalidé suffisamment de composantes non-lettres pour éviter la création de germes parasites. Ceci est illustré sur la Figure 9.27(f). Le module de filtrage a validé un certain nombre de composantes non lettres. Celles-ci vérifient parfaitement les critères géométriques nécessaires à la création d'un germe. De plus ce germe pourra re-valider d'autres composantes non lettres augmentant par la même le taux de faux positifs.

Dans le cadre général, nous obtiendrons un résultat intermédiaire entre ces deux cas.

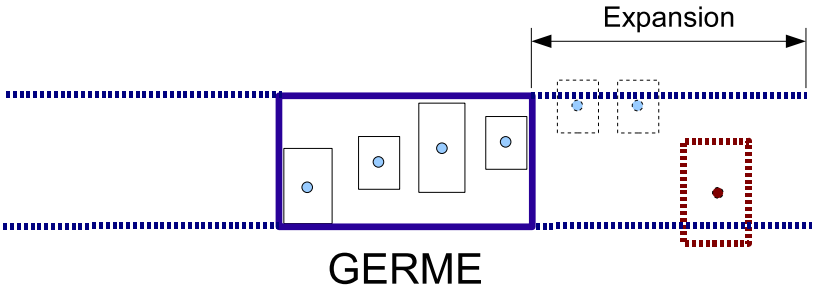
9.6.3 Conclusion et perspectives

Conclusion Nous avons exposé ici un module relativement commun dans les systèmes de localisation de texte. Il s'agit du regroupement itératif de localisations partielles issues d'étapes préalables. Celui-ci se retrouve sous deux formes dans la littérature : premier cas, les approches zones de texte,

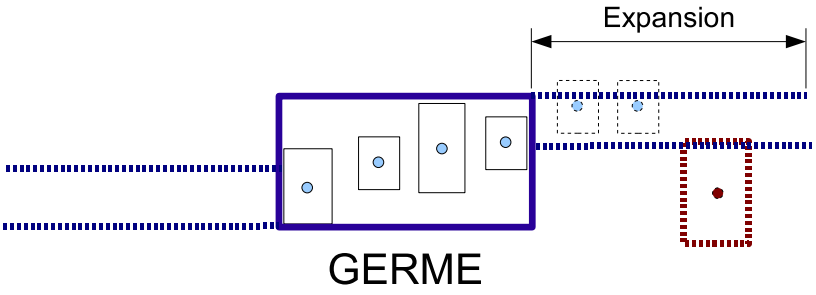
[19] Les méthodes d'évaluation utilisant généralement l'appariement de boîtes, et certaines n'utilisent qu'un simple test *one to one*, ainsi toute boite surnuméraire fera chuter la précision du système



(a) Cas 1 : Le texte est horizontal , on peut facilement tester en partant du germe, quelles composantes peuvent être agrégées. Les deux composantes lettres en pointillé bleu seront récupérées.



(b) Cas 2 : Le texte comporte une inclinaison ou nous avons placé dans le germe des zones de deux lignes de textes différentes. L'expansion du germe et le test d'agrégation sont biaisés. On va récupérer la boite rouge (non lettres) et ne pas récupérer les deux composantes lettres.



(c) Proposition de résolution du cas 2 qui reste valable pour le cas 1 : on va partir du germe et définir une boîte *moyennée* et faire l'expansion de celle-ci à partir de la boite la plus à gauche et de la boite la plus à droite. Ceci va bien nous permettre de récupérer les deux boîtes lettres et d'exclure la boîte non lettre.

FIG. 9.30: Justification de l'étape d'agrégation de nouvelles composantes à partir des premiers germes agrégés (voir texte) : le carré bleu représente le germe, les boîtes englobantes en trait pointillé bleu représentent des boîtes englobantes de lettres que l'on souhaite agréger, les boîtes englobantes en trait pointillé rouge représentent des boîtes non lettres qui ne doivent pas être agrégées.

(a) Création d'un germe d'au moins trois composantes

(b) Expansion du germe, en prenant en compte une *boîte moyenne* voir texte.

(c)

(d) Image issue de la base de Test à Blanc *ImagEval*

(e) Sortie du processus d'agrégation après application du prédicat *ImagEval*

FIG. 9.31: Illustration de différentes étapes du module de regroupement itératif des composantes.

elles vont regrouper des «zones» (mots ou portions de mots) sur différents prédicats (eg :Wolf and Jolion [112]), second cas, approche composante connexe, on doit regrouper les lettres pour former les mots puis la zone de texte (eg :Hase et al. [28], Wang and Kangas [108]).

Il s'agit souvent d'un module paramétrique s'appuyant sur des hypothèses fortes (sur l'horizontalité des textes par exemple). Nous avons dû également définir de manière empirique les différents tests d'agrégation pour nous conformer au mieux à la variabilité des textes que nous avons eus à traiter (cf Chapitre 10, p.185). Ne pouvant pas nous appuyer sur des hypothèses fortes, nous avons proposé un module itératif, passant par la création de germes (ayant de fortes probabilités d'être des zones de texte), puis en relâchant nos contraintes, pour agréger au mieux les lettres ne faisant pas partie des germes initiaux. Ce module offre des résultats satisfaisants dans notre cadre applicatif mais est entaché de faiblesses :

1. Il ne s'appuie que sur la géométrie des boîtes englobantes
2. La stratégie utilisant des germes permet de former des premières zones potentielles certes mais celle-ci est fortement dépendante des filtrages précédents. Nous pouvons isoler deux cas problématiques majeurs : premièrement le module précédent n'a pas conservé suffisamment de composantes lettres pour former le germe, ce qui entraîne une absence totale de localisation, deuxièmement, le module précédent a conservé un certain nombre de composantes non lettres qui sont agrégées en germes, ce qui se traduit inévitablement par de faux positifs.

Nous proposons ci-après quelques pistes de réflexion pour améliorer ce module.

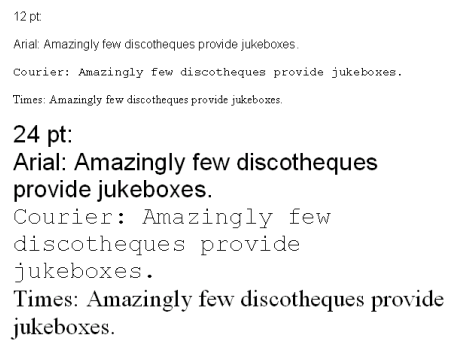
Perspectives

1. Une première piste serait de définir le problème d'agrégation sous forme d'une *fonctionnelle* à minimiser comme l'ont proposé Zhang and Chang [120], Park et al. [68] et Hase et al. [28]. Ceci nous permettrait d'agréger de proche en proche les composantes et de vérifier l'impact de cette agrégation vis-à-vis d'un modèle de zone de texte. Les modèles de zones pourraient être : des lignes avec différents angles ou en ellipse.
2. Un autre point important serait l'intégration d'autres caractéristiques que de simples considérations géométriques lors de l'agrégation. En effet si pour des textes plans et horizontaux celles-ci peuvent sembler suffisantes, il serait intéressant d'intégrer d'une part des informations colorimétriques (tester l'adéquation des *distributions couleur* des composantes avant agrégation comme proposé par Chen et al. [16]) et d'autre part des informations caractéristiques des lettres (avec l'intégration d'une partie des descripteurs de l'apprentissage dans les tests d'agrégation).
3. Enfin, un module additionnel de validation des boîtes formées pourrait êtres mis en place. On pourrait essayer d'une part de caractériser ce qu'est une **zone de texte**, on bouclerait ici sur les approches textures (eg : [11]) et d'autre part de faire intervenir un O.C.R comme arbitre final de validation.[20]

9.7 Intégration d'un OCR ?

Les systèmes de localisation de textes que nous avons développé utilisent l'approche *composante connexe* (cf Section 4.2, p.26). Aussi en sus des boîtes englobantes détectées qui permettent

[20]Rappelons que de nombreuses étapes (eg : restauration d'échelles, corrections de perspectives) sont souvent nécessaires pour utiliser cette approche



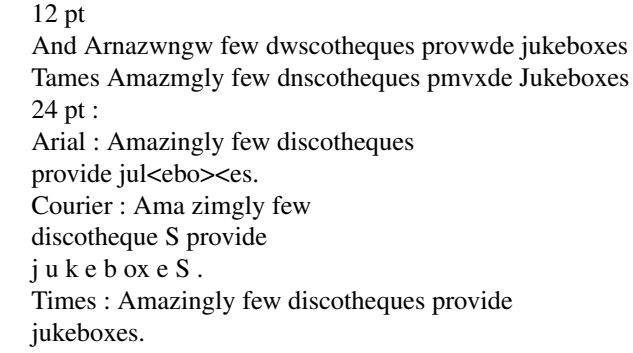


FIG. 9.32: Non robustesse d'un O.C.R donné sur une image de référence. Texte blanc sur fond noir pour trois fontes et deux tailles. O.C.R utilisé *Tesseract*©

l'évaluation de nos résultats vis-à-vis de vérité terrain, nous disposons des composantes connexes sous-jacentes qui ont permis la formation des boîtes englobantes.

A l'inverse des approches où une étape de seuillage à l'intérieur des boîtes détectées est nécessaire pour récupérer les lettres, nos systèmes proposent intrinsèquement une pré-segmentation des lettres.

Attention, notre segmentation n'est pas exempte de reproche ; en effet les composantes connexes que nous avons conservées peuvent être relativement éloignées de ce qu'un O.C.R peut traiter. Comme nous l'avons souligné dans la Section 4.4, p.36, pour tout T.I.E une étape importante de restauration précède généralement l'étape de reconnaissance.

De plus, une segmentation donnée même de *bonne qualité* ne correspondra pas forcément à ce que l'O.C.R visé est *capable de traiter*. Pour illustrer très simplement ce propos, il suffit de s'intéresser à la Figure 9.32, celle-ci illustre les sorties fournies par l'O.C.R *Tesseract*©[21] pour un texte blanc sur fond noir. On observe que le système n'est pas robuste vis-à-vis des couples fontes/tailles proposés.

Aussi, les exemples de reconnaissance que nous présenterons dans le Chapitre 10, p.185, tiendront plus de perspectives que de résultats.

9.8 Conclusion

Au cours de ce chapitre nous avons décrit deux chaînes de traitement ; utilisant différemment les informations véhiculées par l'opérateur d'ouverture ultime. Elles reposent toutes les deux sur une première étape de segmentation (i.e la première utilise l'image transformée comme image à segmenter alors que la seconde utilise directement l'indicatrice comme image pré-segmentée). Notre méthodologie s'inscrit donc dans les approches dites par «composantes connexes». Ces approches reposent «classiquement» sur une deuxième phase de filtrage et de sélection des composantes «lettre» suivie d'une phase d'agrégations de celles-ci pour former le résultat final de l'algorithme sous forme d'une liste de boîtes localisées par image.

Les modules décrits sont pour la plupart paramétriques et nous avons veillé à justifier au mieux ce fait. Nous avons également mis en lumière la difficulté de proposer des métriques intermédiaires permettant une optimisation *locale* de leur performance respective.

Les chaînes présentées seront évaluées et validées dans le chapitre suivant au travers de trois projets distincts : deux campagnes d'évaluation d'algorithmes et la réalisation d'un démonstrateur

[21]Qui sera l'OCR utilisé par la suite. O.C.R licencié sous "the Apache License, Version 2.0" voir `http://code.google.com/p/tesseract-ocr/`

pour EADS.

10 Présentation des bases traitées et résultats

Ne fais pas attention à ce que l'on écrit sur toi. Contente-toi de le mesurer.

The New York Times
ANDY WARHOL

Nous présenterons dans ce chapitre les résultats quantitatifs et qualitatifs de nos travaux sur deux "grandes" bases images.

*Une première section sera dédiée aux méthodes d'évaluation utilisées. Puis nous détaillerons nos résultats. Premièrement, au sein de la campagne d'évaluation d'algorithmes de traitement d'images $imagEval$, comportant une catégorie "localisation de texte". Deuxièmement en produisant des résultats avec la base d'évaluation "ICDAR" (*International Conference on Document Analysis and Recognition*). Cette dernière étude nous permettra de situer notre approche vis-à-vis de l'état de l'art.*

Nous détaillerons pour chacune des bases, leurs propriétés*, et la qualité de nos localisations. Nous veillerons à discuter des forces et faiblesses de notre approche pour chacune. Enfin nous terminerons ce chapitre sur une parenthèse concernant la reconnaissance des textes localisés.*

10.1 Rappels sur les méthodes d'évaluation

Au début de ce manuscrit (voir Section 4.5) nous avions passé en revue les différentes méthodes d'évaluation des Systèmes d'Extraction de Texte. Nous avions convenu de classer ces méthodes en deux grandes familles :

1. Les approches "géométriques", qui prennent en compte la phase de localisation. On mesure une qualité de recouvrement (eg :au niveau pixel, au niveau boîte englobante) entre une vérité terrain et la sortie d'un système. On ne se soucie pas de la reconnaissance.
2. Les approches "contenu", qui prennent en compte la phase de reconnaissance. On compare les chaînes de caractères annotées (les mots/phrases présentes dans les images) avec celles reconnues par le système. La qualité de localisation ne rentre pas en ligne de compte.

Pour les bases ImagEval et ICDAR les vérités terrain sont fournies, sous la forme d'une liste de boîtes englobant les textes à localiser.

Nous nous situons donc, dans l'évaluation du type **approche géométrique**. Plus précisément nous devons fournir à la sortie de nos algorithmes un ensemble de boîtes qui seront ensuite comparées de manière **géométrique** à une vérité terrain.

Nous avions identifié trois écueils à la comparaison objective des algorithmes de localisation :

- **Définition de la vérité terrain :** Comment mesurer le recouvrement d'une vérité terrain donnée par une annotation humaine dont la granularité peut être au niveau des caractères, des mots, des lignes, etc... avec un système de localisation de texte donné qui lui même peut fournir des résultats à une granularité qui lui est propre.
- **Mesure proposée :** la pertinence de la mesure proposée est toujours reliée à l'annotation réalisée.
- **Application recherchée :** en fonction de celle-ci, tout ou seulement un sous-ensemble des zones de texte peut être annoté.

10.1.1 Limitation de l'approche géométrique «simpliste»

Une des métriques de référence, pour les approches géométriques est celle définie pour la campagne d'évaluation ICDAR.

Pour rappel elle se définit comme suit :

$$\text{RAPPEL} = \frac{\sum_i Match_G(G_i)}{card(G)} \quad \text{PRÉCISION} = \frac{\sum_j Match_L(L_j)}{card(L)}$$

avec

$$Match_G(G_i) = \max_j \frac{2 \cdot |G_i \cap L_j|}{|G_i| + |L_j|} \quad Match_L(L_j) = \max_i \frac{2 \cdot |L_j \cap G_i|}{|L_j| + |G_i|}$$

où

$|.|$ est l'aire de la boîte (i.e. le nombre de pixels)

où $L = \{l_1, \cdots, l_N\}$, $G = \{g_1, \cdots, g_M\}$ représentent respectivement l'ensemble des boîtes englobantes générées par le système et celles de la vérité terrain.

Cette métrique souffre de trois inconvénients majeurs, comme souligné en Section 4.5 :

Perception : Le taux de recouvrement des boîtes n'est pas une mesure perceptuellement valide.

Ambiguïté : C'est une mesure ambiguë : si on prend par exemple un taux de RAPPEL de 50%, de manière abusive on pourrait conclure soit que 50% des boîtes de la vérité terrain ont été localisées parfaitement, soit que toutes les boîtes de la vérité terrain ont été localisées avec un recouvrement de 50%.

Granularité : Cette mesure ne propose que des correspondances "une boîte vers une boîte" ("one-to-one")

Le fait que cette mesure utilise un simple recouvrement des boîtes couplé au problème de la granularité en fait un outil peu adapté pour le texte enfoui où l'on peut trouver beaucoup de texte penché, non plan ou utilisant des effets de style.

Aussi parallèlement à la métrique I.C.D.A.R, nous utiliserons une métrique proposant des appariements "one-to-many" et "many-to-one".

10.1.2 Métrique prenant en compte les problèmes de granularité

Pour pallier aux biais de la métrique précédente, nous avons opté [1] pour une métrique prenant en compte des appariements de boîtes "one-to-many" et "many-to-one". Celle-ci est issue des travaux de Wolf and Jolion [111]. Nous utiliserons le nom du premier auteur comme référence par la suite.

Le but de cette métrique est de se rapprocher le plus possible, d'une définition intuitive des termes de précision(P_{OB}) et de rappel(R_{OB}), définis comme suit :

$$\begin{aligned} R_{OB} &= \frac{\text{Nombre de boîtes } \textit{correctement} \text{ localisées}}{\text{Nombre de boîtes de la vérité terrain}} \\ P_{OB} &= \frac{\text{Nombre de boîtes } \textit{correctement} \text{ localisées}}{\text{Nombre total de boîtes localisées}} \end{aligned} \tag{10.1}$$

Le principal problème est de mesurer de la manière la moins ambiguë possible, le nombre de boîtes *correctement* localisées. Pour cela, Wolf and Jolion [111], redéfinissent entièrement la chaîne de mesure.

10.1.2.1 Définition et procédé

Nous définissons toujours des taux de recouvrement entre boîtes, qui définiront la précision et le rappel entre une boîte de la vérité terrain et une boîte localisée :

$$\begin{aligned} \text{RAPPEL}_{AR} &= \frac{|G_i \cap L_j|}{|G_i|} \\ \text{PRÉCISION}_{AR} &= \frac{|G_i \cap L_j|}{|L_i|} \end{aligned}$$

Où, RAPPEL_{AR} et PRÉCISION_{AR} sont des mesures de *qualité* de recouvrement en terme de surface.

Là où la métrique ICDAR, se contentait de prendre le plus grand taux de recouvrement, nous allons créer ici des «matrices» de recouvrement ie :

$$\sigma_{ij} = \text{RAPPEL}_{AR}(G_i, D_j) \quad \tau_{ij} = \text{PRÉCISION}_{AR}(G_i, D_j)$$

Ces matrices permettent de déterminer les correspondances entre deux listes de boîtes. Une valeur non nulle pour un élément d'indices (i, j) indique qu'il y a bien un recouvrement entre une boîte de la vérité terrain G_i et une boîte localisée L_j. Cependant ce recouvrement n'est pris en compte que si celui-ci est au-dessus d'un **seuil de décision** de *qualité de recouvrement*.

Formellement, on définira :

$$\sigma_{ij} \geq tr \tag{10.2}$$

$$\tau_{ij} \geq tp \tag{10.3}$$

où $tr \in [0, 1]$ est la contrainte sur le rappel et $tp \in [0, 1]$ est la contrainte sur la précision

Dans la pratique nous détaillons ci-dessous les différents cas pris en compte :

Appariement "one-to-one" : On considère la mise en correspondance d'une boîte de la vérité terrain Gi avec une boîte localisée Lj si la ligne i de chaque matrice[2] contient juste un élément satisfaisant les contraintes des équations 10.2, 10.3 et si la colonne j de chaque matrice contient juste un élément satisfaisant les contraintes des équations 10.2, 10.3. Ce cas est présenté sur la Figure 10.1(a).

[1] et fait adopter dans la campagne ImagEval

[2] i.e. représentant σ_{ij} et τ_{ij}

Appariement "one-to-many" (i.e. détection morcelée) : une boîte de la vérité terrain Gi est mise en correspondance avec un ensemble So de boîtes localisées Lj tel que $j \in So$ si

- une portion significative de la boîte de la vérité terrain a été localisé (cf. conditions de l'équation 10.2 mais de manière morcelé : $\sum_{j \in So} \sigma_{ij} > tr$ **et**
- Pour qu'un de ces *morceaux* soient pris en compte, son taux de recouvrement doit satisfaire l'équation 10.3 : $\forall j \in So, \tau_{ij} \geq tp$

Ce cas est présenté sur la Figure 10.1(b).

Appariement "many-to-one" (i.e. fusion) : cas dual du précédent, une boîte localisée Lj est mise en correspondance avec un ensemble Sm de boîtes de la vérité terrain. Si

- chaque boîte de la vérité terrain qui participe à l'appariement doit avoir une portion significative à l'intérieur de la boîte localisée (cf. conditions de l'équation 10.2) : $\forall i \in Sm, \sigma_{ij} \geq tr$ **et**
- La précision de recouvrement lors de l'appariement est suffisante (cf. conditions de l'équation 10.3) : $\sum_{i \in Sm} \tau_{ij} > tp$. Autrement dit, la surface de la boîte localisée est bien couverte par l'union des boîtes de la vérité terrain.

Ce cas est présenté sur la Figure 10.1(c).

(a) Appariement "one-to-one" (b) Appariement "one-to-many" (c) Appariement "many-to-one"

FIG. 10.1: Les différents types d'appariement entre boîtes issues de la vérité terrain (ligne pointillée) et boîtes localisées (ligne pleine). Illustration provenant de [111], utilisant un panagramme bien connu.

Partant de ces nouvelles stratégies d'appariement, nous pouvons définir la précision et le rappel de l'équation 10.1 :

$$R_{OB}(G, L, tr, tp) = \frac{\sum_i Match_G(G_i, L, tr, tp)}{card(G)}$$
$$P_{OB}(G, L, tr, tp) = \frac{\sum_j Match_L(L_j, G, tr, tp)}{card(L)}$$

avec $Match_G$ et $Match_L$ des fonctions prenant en compte les différents types d'appariement.

$$Match_G(G_i) = \begin{cases} 1 \text{ Si } G_i \text{ s'apparie avec une seule boîte localisée} \\ 0 \text{ Si } G_i \text{ ne s'apparie avec aucune boîte localisée} \\ f(k) \text{ Si } G_i \text{ s'apparie avec un nombre } k \text{ de boîtes localisées} \end{cases}$$

$$Match_L(L_j) = \begin{cases} 1 \text{ Si } L_j \text{ s'apparie avec une seule boîte de la vérité terrain} \\ 0 \text{ Si } L_j \text{ ne s'apparie avec aucune boîte de la vérité terrain} \\ f(k) \text{ Si } L_j \text{ s'apparie avec un nombre } k \text{ de boîtes la vérité terrain} \end{cases}$$

$f(k)$ permet de moduler la précision et le rappel en cas d'appariement "many-to-one" et "one-to-many". Il sera généralement fixé à une constante (dans notre cas 0.8).

10.1.2.2 Passage de l'image à une liste d'images

Il nous reste à *intégrer* nos résultats pour une liste d'images : Nous avons N images, donc N listes de boîtes provenant de la vérité terrain ainsi que N listes de boîtes localisées. Soit :

$$\overline{F} = \{G^k\}, k = 1..N$$
$$\overline{L} = \{L^k\}, k = 1..N$$

avec

G_i^k boîte i de la vérité terrain dans l'image k

L_i^k boîte j localisée dans l'image k

$$R_{OB}(\overline{F}, \overline{L}, tr, tp) = \frac{\Sigma_k \Sigma_i Match_G(G_i^k, L^k, tr, tp)}{\Sigma_k |G^k|}$$
$$P_{OB}(\overline{F}, \overline{L}, tr, tp) = \frac{\Sigma_k \Sigma_i Match_L(L_i^k, G^k, tr, tp)}{\Sigma_k |L^k|}$$

10.1.2.3 Rôle des paramètres

Nous venons de rappeler les définitions utilisées par [111]. Celles-ci font intervenir notamment deux paramètres tr et tp qui imposent des contraintes sur la *qualité* des localisations.

Pour bien appréhender la performance d'un système, deux diagrammes doivent être générés :

1. On fixe le paramètre tr, et l'on regarde la variation en terme de précision/rappel d'un système vis-à-vis du paramètre tp
2. Le cas dual (tp fixe, tr variable)

Nous reviendrons sur l'interprétation de ces diagrammes par la suite (voir discussion en Section 10.2.3.2 p.195).

La création de ces diagrammes impose de fixer l'un des deux paramètres. Ainsi se pose la question d'une valeur de référence pour chacun d'eux. Comme l'indiquent les auteurs dans [111] ces valeurs ont été fixées certes de manière empirique mais dans le but de plus ou moins pénaliser les cas d'appariement partiels.

Aussi ils ont décidé en première approche de fixer $tr = 0.8$ et $tp = 0.4$. Ce qui implique qu'un système donnant beaucoup de résultats «morcelés» est plus «pénalisé» qu'un système renvoyant systématiquement un résultat plus grand que l'annotation (ie une boîte localisée par le système est plus grande mais recouvre entièrement la boîte de la vérité terrain).

Remarque 18 *Le paramètre tp est-il trop faible ?*

*Sa valeur est fixée à 0.4 et pourrait sembler faible. Mais il ne faut pas oublier que les taux d'appariement sont calculés sur les **surfaces** de recouvrement entre boîtes.*

Remarque 19 *Le paramètre tr est-il trop élevé ?*

Le choix de pénaliser fortement le morcellement d'un résultat dépend avant tout de l'adéquation entre la sortie d'un algorithme et la granularité de la vérité terrain. Ainsi, en cas de très forte disparité, on pourra être amené à modifier à la baisse ce paramètre. Nous y reviendrons par la suite.

10.1.2.4 Synthèse de l'évaluation

L'utilisation des deux diagrammes (i.e. variation de tp alors que tr est fixé, et inversement) permet d'apprécier *finement* les performances d'un système donné. Cependant il est souvent commode de déterminer *une* valeur de performance globale d'un système, pour permettre une comparaison rapide avec d'autres approches ou permettre l'optimisation d'un système.

Comme l'indiquent les auteurs dans [111], l'utilisation d'*une mesure* résumant de manière non ambiguë la définition du rappel et de la précision de l'équation 10.1 en Section 10.1.2 est complexe à définir.

Cette mesure doit pouvoir prendre en compte la variation des paramètres tr et tp pour ne pas pénaliser arbitrairement tel ou tel système. Pour pallier cette dépendance, les auteurs proposent de mesurer l'aire des sous-graphes des diagrammes, ce qui serait sensiblement équivalent à une valeur moyenne de performances sur l'ensemble des variations possibles des contraintes tr et tp.

Formellement les auteurs proposent de calculer :

$$R_{OV} = \frac{1}{2T}\sum_{i=1}^{T} R_{OB}(\overline{G}, \overline{L}, \frac{i}{T}, tp) + \frac{1}{2T}\sum_{i=1}^{T} R_{OB}(\overline{G}, \overline{L}, tr, \frac{i}{T})$$

$$P_{OV} = \frac{1}{2T}\sum_{i=1}^{T} P_{OB}(\overline{G}, \overline{L}, \frac{i}{T}, tp) + \frac{1}{2T}\sum_{i=1}^{T} P_{OB}(\overline{G}, \overline{L}, tr, \frac{i}{T})$$

avec T un paramètre contrôlant le rapport entre temps de calcul
et précision de l'approximation, en pratique $T = 20$.

La synthèse de l'évaluation correspondra à la moyenne harmonique de R_{OV} et P_{OV}.

10.1.3 Conclusion

Pour la suite, nous validerons systématiquement nos résultats à l'aide des métriques ICDAR et WOLF[3]. Ceci nous permettra de voir la pertinence d'une métrique prenant en compte différents types d'appariements pour des bases aux granularités antinomiques. Comme nous le verrons par la suite, la base ICDAR est annotée au niveau mots voire lettres, alors que le comité d'ImagEval annote au niveau phrases, voire paragraphes.

Pour la suite on notera :

- $ICDAR_{Prec}$, $ICDAR_{Rec}$, $ICDAR_{Hmean}$, respectivement la précision, le rappel et la moyenne harmonique pour la métrique $ICDAR$ (voir définition en p.186).
- $WolfPonctual_{Prec}$, $WolfPonctual_{Rec}$, $WolfPonctual_{Hmean}$, respectivement la précision, le rappel et la moyenne harmonique *ponctuelle* pour la métrique $WOLF$, i.e. pour des valeurs fixes de tr et tp (voir définition en p.187) .
- $WolfOverall_{Prec}$, $WolfOverall_{Rec}$, $WolfOverall_{Hmean}$, respectivement la précision, le rappel et la moyenne harmonique pour la métrique $WOLF$, intégrant les variations de tr et tp (voir définition en p.190).

Nous venons de décrire les outils métriques qui vont nous servir par la suite à mesurer de manière *quantitative* la performance des systèmes de localisation de texte. Nous pouvons désormais présenter les différentes bases traitées ainsi que les résultats, tant quantitatifs que qualitatifs, que nous avons obtenus.

[3]Pour le calcul de celles-ci nous utiliserons le logiciel DetVal 1.01 disponible sur `http://liris.cnrs.fr/christian.wolf/software/deteval/index.html`

10.2 Campagne d'évaluation ImagEval

10.2.1 Finalité de cette étude

Le projet $imagEval$[4] s'intéresse à l'évaluation de technologies de filtrage et d'indexation d'images, de recherche d'images par le contenu et de description automatique d'images dans des bases de données de gros volume. L'objectif est double : organiser des évaluations importantes à partir de besoins concrets et évaluer des technologies détenues par des équipes de recherche françaises et étrangères ainsi que des solutions logicielles.

Dans ce cadre le consortium a mis en place diverses tâches[5] : 1. La reconnaissance d'images transformées ; 2. La Recherche combinée texte/image ; 3. **La détection de zones de texte** ; 4. La détection d'objets ; 5. La reconnaissance d'attributs.

Nous avons participé à la tâche 3 qui comportait au départ du projet une phase d'extraction et de reconnaissance de texte. Cette tâche ayant ensuite été abandonnée faute de participants.

La campagne d'évaluation s'est déroulée en deux grandes étapes :

1. Soumission d'une base de test à blanc de 500 images, **le comité ne fournissant aucune vérité terrain**. Évaluation des algorithmes à huis-clos, sans divulgation publique des résultats.
2. Soumission d'une base de test officielle de 500 images, le comité fournissant la vérité terrain du test à blanc. Évaluation des algorithmes et divulgation publique des résultats (voir `http://www.imageval.org/e_resultats.html`).

Aussi par la suite, nous nous permettrons d'illustrer nos approches tant sur des images provenant du test à blanc que du test officiel mais nous ne fournirons des résultats quantitatifs que pour la base de test officielle. En effet, les considérations pratiques quant à la consistance de l'annotation n'ont été fixées que pour celle-ci.

10.2.2 Présentation de la base officielle

Les images sont issues de différents fonds photographiques dont : HFM Group (Hachette Filipacchi Médias), Niepce Museum, Archiv'Images, Réunion des Musées Nationaux, RENAULT. Aussi on se retrouve devant des pannels d'images différents mélangés au sein d'une même base.

10.2.2.1 Détails de la base officielle

La base officielle est composée de 500 images. Chaque image peut contenir du texte en surimpression (eg :des légendes de cartes postales) comme du texte enfoui mélangeant texte manuscrit, cachet de timbre et des textes présents au sein de l'image.

Synthétiquement elle est constituée de

- 190 Cartes postales anciennes (Noir et Blanc, colorisée ou en couleur), comportant majoritairement des légendes
- 206 Photographies Couleur, comportant majoritairement du texte enfoui *classique*
- 104 Photographies Noir et Blanc d'archives comportant majoritairement du texte enfoui

Notons que 90 images (18%) ne contiennent aucun texte : 40 cartes postales anciennes, 47 Photographie Couleur, 3 Photographie Noir et Blanc d'archives.

[4] Financé au sein du programme "Techno-Vision" du ministère délégué à la recherche. Détail sur `http://www.recherche.gouv.fr/appel/2004/technovision.htm`

[5] Une présentation plus détaillée est disponible sur `http://www.imageval.org/`

La Figure 10.2 mêlant images de la base officielle et images de la base de test à blanc illustre la variabilité des textes à localiser.

FIG. 10.2: Présentation de la variabilité des types de textes rencontrés au sein de la campagne d'évalution ImagEval

10.2.2.2 Illustration des annotations et remarques concernant les métriques

Comme nous l'avons indiqué précédemment le comité d'évaluation d'$imagEval$ a annoté la base d'images avec une granularité relativement faible (i.e. grossière : elle est *au niveau* ligne de texte voire paragraphe), comme présenté sur la Figure 10.3.

Critère de fusion d'ImagEval : Pour justifier ce choix le comité indique qu'une première annotation au niveau *mots* peut être réalisée et, partant de celle-ci l'application récursive d'un critère de fusion défini par le comité permet de remonter à l'annotation finale.

Ce critère que nous nommerons $MergeCriterionImageVal$ est défini sur la Figure 10.4.

Métriques : Des résultats seront fournis pour les métriques $ICDAR$ et $Wolf$. Notons que deux aménagements ont été réalisés par le comité d'évaluation du fait de la grande *difficulté* de la base :

- Prise en compte des images ne possédant pas de texte : si un système donné ne renvoie aucune localisation pour des images ne contenant pas de texte, **la précision et le rappel seront mis à 1.**
- Baisse des paramètres de sensibilité de la métrique de $Wolf$: le paramètre tr est fixé à 0.6 (et non 0.8, cf Section 10.1.2.3, p.189), pour pallier *aux problèmes* de granularité de l'annotation.

FIG. 10.3: Présentation de la granularité de l'annotation fournie par le comité d'$imagEval$.

10.2.3 Résultats

10.2.3.1 Chaînes de traitement et paramètres utilisés

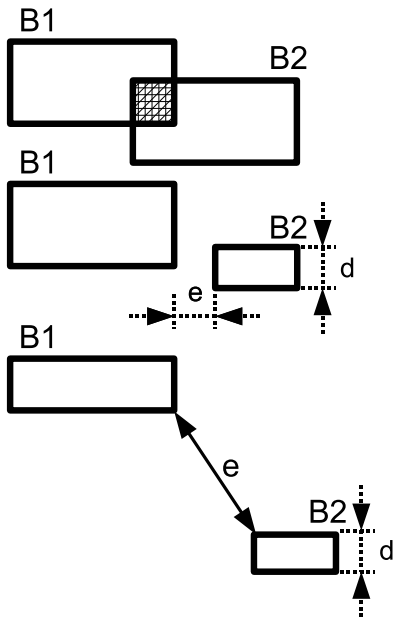
Dans le cadre de l'évaluation du test officiel les deux chaînes de traitement : "Approche Transformée" et "Approche Indicatrice" (cf. Figure 9.2, Chapitre 9, p.135) ont été utilisées.

Nous donnons dans les Tableaux 10.1 et 10.2, le détail des différents paramètres utilisés, respectivement pour l'approche transformée et l'approche indicatrice.

Ces paramètres ont été fixés expérimentalement lors de la campagne de test à blanc. Nous avons discuté dans le chapitre 9 des forces et faiblesses des différents modules que nous avons mis en place, et de la difficulté d'optimiser un module indépendamment des autres.

Cependant trois paramètres réclament une attention toute particulière :

1. Le *Stop* pour les deux chaînes de traitement. Nous arrêtons l'opérateur d'ouvert ultime au tiers de l'image, ceci pour contrer la myopie de l'opérateur en présence de structures imbriquées. Cette valeur a été fixée au-dessus des plus grandes lettres présentes dans la base mais ne palliera



Le critère considère que deux boîtes doivent être fusionnées, si elles sont *proches proportionnellement à leurs dimensions. Ceci implique l'utilisation de paramètres :*
Soit B_1 et B_2 deux boîtes :
- *Si $B_1 \cap B_2 \neq \varnothing$*
  → *fusion des boîtes*
  - *Sinon, calcul de e et d tel que :*
    - *e soit la distance euclidienne minimale entre B_1 et B_2*
    - *d soit la plus petite dimension de B_1 et B_2. Soient H_1, L_1, H_2, L_2, les hauteurs et largeurs de B_1 et B_2. $d = \wedge(H_1, L_1, H_2, L_2)$*
  - *Si $\frac{e}{d} \leq 2$*
    → *fusion des boîtes*

FIG. 10.4: Critère de fusion de boîtes défini par le comité d'$imagEval$, pour la création de la vérité terrain.

Paramètre	Description	Valeur Retenue
$Stop$	Arrêt de l'ouvert ultime	0.3*Hauteur de l'image
S	Seuil bas de la transformée (cf. Section 9.3.1)	9
Filtrage basé sur la «cohérence spatiale» (cf.Section 9.3.2.1)		
$SizeBall$	Taille Dilatation	1
$Tresh_{NB_{CC}}$	Nombre de composantes	500
Filtrage basé sur «l'épaisseur» (cf.Section 9.3.2.2)		
$S_{epaisseur}$	Seuil de tolérance pour une CC	1
Filtrage fin utilisant l'apprentissage (cf.Section 9.5)		
Classifieur sélectionné : jeu de paramètre I, avec expansion quadratique (cf. Section 9.5.5)		
Module de regroupement spatiale (cf. Section 9.6)		
$Nb_{CCgermes}$	Nombre minimal de composantes pour former un germe	3
$Size_{Expand}$	Extension des germes en pixels	200
S_1	Seuil d'appariement de boîtes sur la hauteur	1.1
S_2	Seuil d'appariement de boîtes sur l'alignement	0.7
S_3	Seuil d'appariement de boîtes sur la distance	0.7
Prédicat final de fusion des boîtes		
Applications récursive du critère de fusion ImagEval (cf. Figure 10.4)		

TAB. 10.1: Paramètres utilisés pour la chaîne transformée. Campagne de test officiel d'ImagEval.

Paramètre	Description	Valeur Retenue
$Stop$	Arrêt de l'ouvert ultime	0.3*Hauteur de l'image
Filtrage Grossier de l'indicatrice (cf.Section 9.4.1)		
S_{area}	Taille du Filtre Aérolaire	5
S	Seuil bas de la transformée (cf. Section 9.4.5)	9
Filtrage fin utilisant l'apprentissage cf.Section 9.5		
Idem Tableau 10.1		
Module de regroupement spatial (cf. Section 9.6)		
Idem Tableau 10.1		
Prédicat final de fusion des boîtes		
Idem Tableau 10.1		

TAB. 10.2: Paramètres utilisés pour la chaîne indicatrice. Campagne de test officiel d'ImagEval .

pas toujours la myopie de l'opérateur.

2. Le seuil S, qui est un seuillage **fixe** de l'image transformée.
3. La taille de l'extension des germes, $Size_{Expand}$.

Nous réaliserons une étude de la dépendance des chaînes de traitement à ces trois paramètres dans la Section 10.2.3.3.

10.2.3.2 Résultats quantitatifs

Les tableaux 10.3 et 10.4 résument les performances *globales* de nos chaînes de traitement sur la base officielle. Une première constatation est que les chiffres globaux sont relativement proches (au centième près) et dénoterait une capacité de nos chaînes à localiser *correctement* 60% des textes, ce qui est un score honorable au vu de la difficulté de la base.

Précision	Rappel	M-Harmonique
$ICDAR$		
0.48	0.57	0.52
$WolfPonctual$		
0.56	0.58	0.57
$WolfOverall$		
0.51	0.53	0.52
Nb de Boîtes Vérité Terrain		580
Nb de Boîtes Localisées		745

TAB. 10.3: Performances globale de la chaîne transformée ($tr = 0.6$; $tp = 0.4$)

Précision	Rappel	M-Harmonique
$ICDAR$		
0.49	0.54	0.51
$WolfPonctual$		
0.57	0.54	0.55
$WolfOverall$		
0.52	0.50	0.51
Nb de Boîtes Vérité Terrain		580
Nb de Boîtes Localisées		706

TAB. 10.4: Performances globale de la chaîne indicatrice ($tr = 0.6$; $tp = 0.4$)

Pour analyser plus finement ces résultats globaux, nous allons étudier les scores par *familles* d'images (i.e. cartes postales anciennes, photographies couleur et photographies noir et blanc).

Un des premiers indicateurs est de regarder la pertinence de nos approches sur les images ne contenant aucun texte. Pour rappel 90 images (18%) de la base ne contiennent aucune annotation :

40 cartes postales anciennes ; 47 photographies couleur ; 3 photographie noir et blanc d'archives. On observe un bon comportement de nos approches :

- **Cartes Postales** : respectivement 25 et 29 sont *classifiées vides* (i.e. nous ne fournissons pas de résultats de localisation) par la chaîne transformée et la chaîne indicatrice
- **Photographies Couleur et Noir Et Blanc** : respectivement 40 et 38

Ce qui indique que nos approches n'ont pas tendance à générer beaucoup de faux positifs.

Comme second indicateur nous proposons dans le Tableau 10.5, le nombre d'images (totales et par catégorie) pour lequel la moyenne harmonique ponctuelle est supérieure à 0.8 ; donc le nombre d'images, qui même avec un certain morcellement sont considérées comme *parfaitement localisées*. Nous pouvons croiser ces informations avec les tableaux 10.3 et 10.4. La chaîne transformée semble se comporter de meilleure manière sur les classes de cartes postales anciennes et de photographies noir et blanc ; en effet sur ce sous-ensemble, il est plus difficile d'obtenir des *segmentations correctes* à l'aide de l'indicatrice, car les fonds sont moins coopératifs et une grande partie des textes sont des légendes avec des textes de *petite taille*. On notera d'ailleurs que, d'une manière générale, l'approche indicatrice localise moins de zones que l'approche transformée (respectivement 706 et 745 boîtes localisées pour 580 boîtes annotées).

Pour les deux chaînes le nombre de boîtes *parfaitement localisées* pourrait paraître faible, cependant cette interprétation dépend de la classe d'image traitée :

- **Cartes Postales** ce nombre est avant tout relié aux problèmes de morcellement de nos localisations.
- **Photographies Couleur** ici il s'agit plus de *non localisations* de nos systèmes. En effet, ces images comportent généralement de une à deux zones de textes qui comportent généralement un faible nombre de caractères.
- **Photographies N&b** Plus difficile à analyser cette classe mélange les deux types précédents.

Chaîne	Nb Images avec $WolfPonctual_{Hmean} \geq 0.8$	Cartes postales anciennes	Photographies Couleur	Photographies N&B
Transformée	226(/500)	92(/190)	77(/206)	57(/104)
Indicatrice	214(/500)	96(/190)	73(/206)	45(/104)

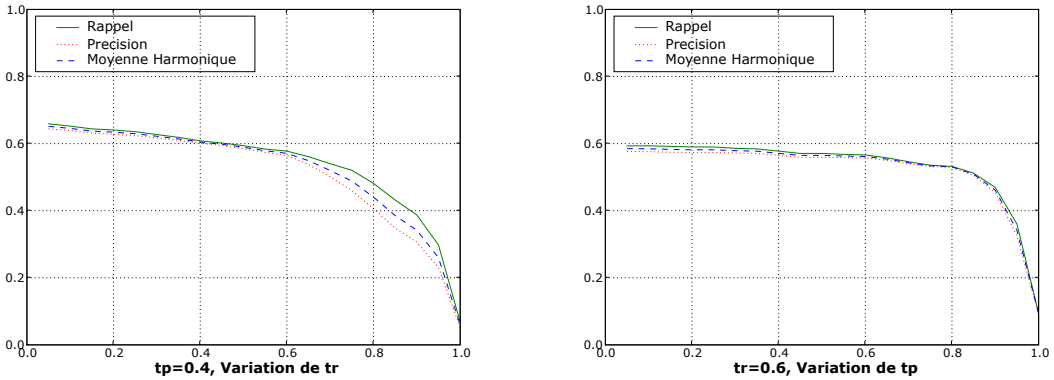
TAB. 10.5: Distribution du nombre d'images localisées avec $WolfPonctual_{Hmean} \geq 0.8$

Enfin, notons que pour pleinement discuter des résultats, il faudrait regarder chaque image séparément et regarder pour chacune la qualité psychovisuelle de la localisation obtenue et les scores fournis par les métriques. Ce qui est bien sur inenvisageable. Nous proposons ce type d'étude, sur un sous-ensemble dans la Section 10.2.3.4 p.200.

Diagrammes de performances : Les diagrammes de performances pour les deux chaînes sont présentés au sein de la Figure 10.5. Les deux paramètres tr et tp permettent respectivement de mesurer si une boîte de la vérité terrain est suffisamment *recouverte* par la/les boîtes localisées et si la localisation proposée est *précise* ou non.

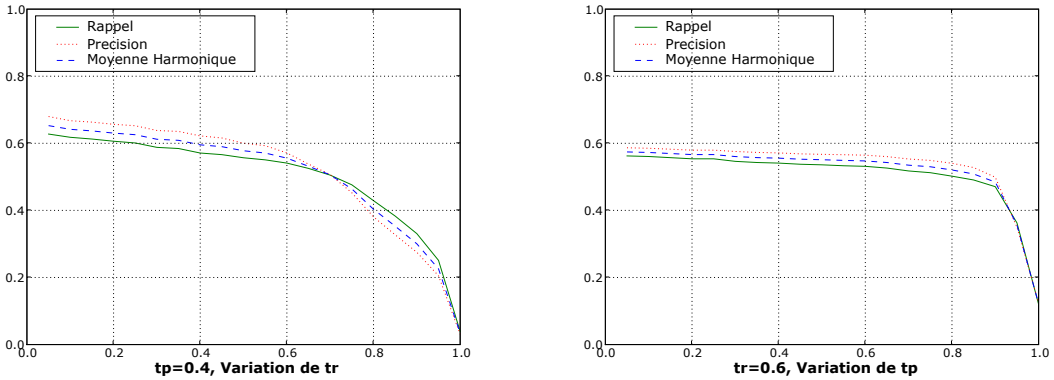
Si l'on regarde les graphiques de la colonne de gauche de la Figure 10.5, on observe que les performances de nos systèmes sont sensibles au seuil de rappel. Plus l'on pousse ce seuil vers 1, plus les performances du système chutent. Ceci est lié à nos problèmes avec la granularité de la vérité terrain. Le fait que nous fournissions dans de nombreux cas des résultats morcelés pénalise nos scores globaux de localisations.

Si l'on regarde les graphiques de la colonne de droite de la Figure 10.5, on observe que les performances de nos systèmes sont beaucoup moins dépendants du seuil de précision, ce qui indique que les boîtes que nous fournissons restent *inscrites* dans les boîtes de la vérité terrain. Il faut pousser relativement loin le paramètre pour faire chuter brutalement les performances de nos systèmes.



(a) (b)

Chaîne transformée



(c) (d)

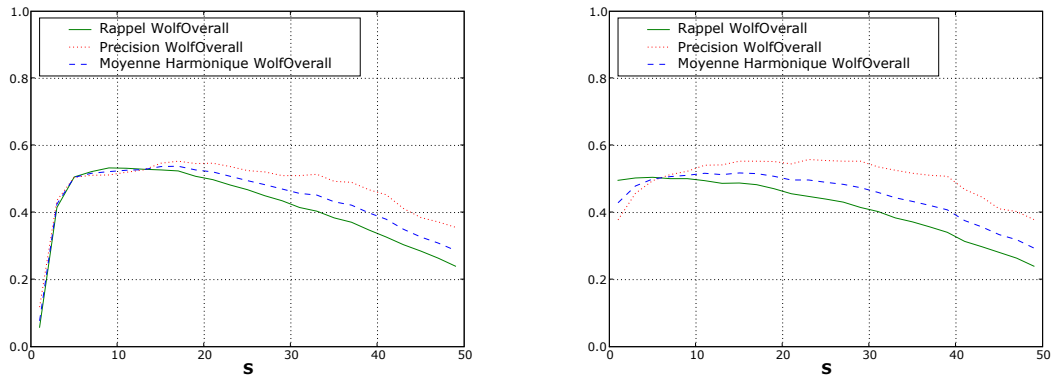
Chaîne Indicatrice

FIG. 10.5: Diagramme de performance sur la base officielle d'ImagEval. Colonne de gauche : variation de la contrainte de rappel tr (avec tp constant et égal à 0.4) ; Colonne de droite : variation de la contrainte de précision tp (avec tr constant et égal à 0.6).

10.2.3.3 Influence des paramètres

Un des paramètres le plus sensible de nos chaînes de traitement est l'utilisation d'un **seuil fixe** sur l'image transformée : 1. Comme paramètre effectif de seuillage pour la chaîne transformée ; 2. Comme simple paramètre de filtrage dans le cas de la chaîne indicatrice.

La Figure 10.6 permet d'apprécier l'influence de ce paramètre :



(a) Chaîne Transformée (b) Chaîne Indicatrice

FIG. 10.6: Influence du paramètre S, Seuil bas de la transformée. Pour la chaîne "Transformée" et la chaîne "Indicatrice". Métrique utilisé $WolfOverall\ tr = 0.6\ tp = 0.4$ (cf. 190).

Concernant la chaîne Indicatrice (cf Figure 10.6(b)), on observe clairement que ce seuil agit bien comme un simple filtrage au début de la courbe. Par contre si l'on pousse ce seuil au dessus de 20 on commence à perdre certaines de nos localisations.

Pour la chaîne transformée, la dépendance est bien plus forte, car de ce seuil dépend l'extraction des composantes connexes. Ainsi le système n'est *performant* que pour une gamme de valeurs données (ici de 5 à 15).

Ainsi pour des résultats quasi-similaires, il semble préférable de se tourner vers la chaîne Indicatrice, dont la dépendance vis-à-vis de ce paramètre est la plus faible.

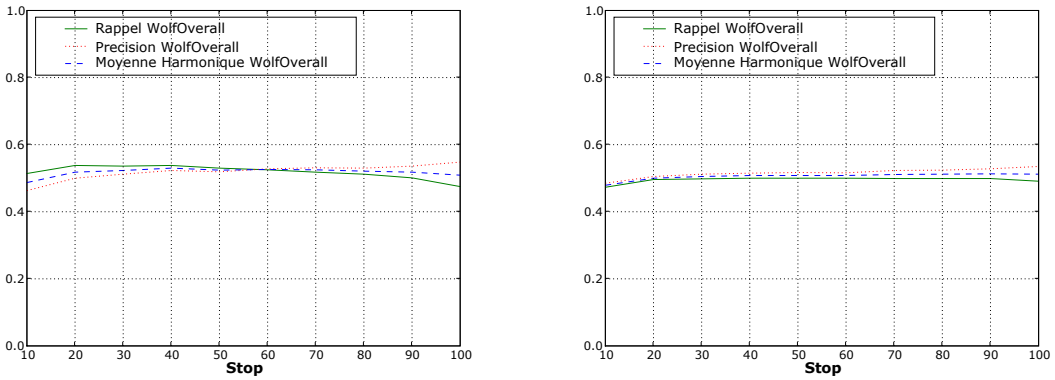
Il s'agissait à notre sens du paramètre le plus critique. Deux autres paramètres méritent notre attention :

1. La valeur du $Stop$, i.e. la valeur d'arrêt de l'ouverture ultime.
2. La taille d'expansion des germes $Size_{Expand}$

Leur influence est présentée respectivement sur les figures 10.7 et 10.8.

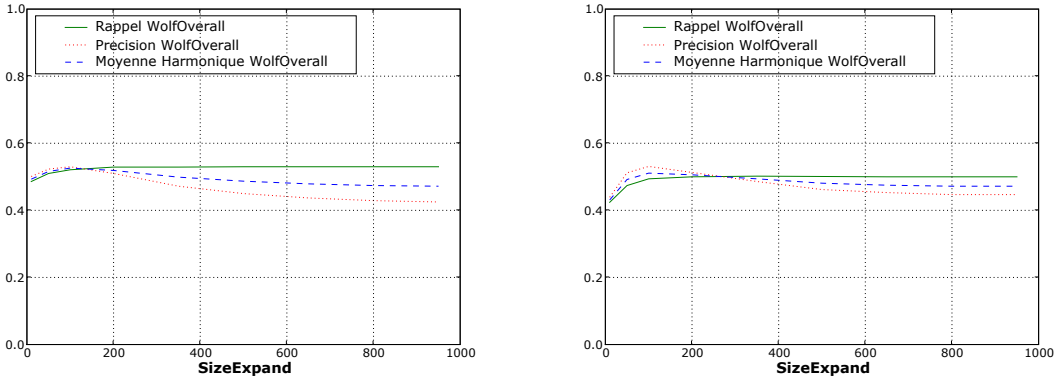
Lors de nos expérimentations la valeur du $Stop$ avais été fixée au tiers de la hauteur de l'image ; ceci pour contrer la myopie de l'opérateur en présence de structures imbriquées. Cette valeur correspondait à la hauteur des plus grandes lettres présentes dans la base. On observe sur la Figure 10.7, que la dépendance de nos chaînes de traitement vis-à-vis de ce paramètre est finalement relativement faible. Nous avons bien besoin de définir une valeur égale ou supérieure à 30% pour obtenir les meilleurs résultats ; i.e. pour prendre en compte toutes les tailles de lettres présentes dans la base. Passée cette limite, on observe une relative statibilité des performances moyennes.

La taille d'expansion des germes $Size_{Expand}$ avait elle aussi été fixée expérimentalement. On observe au travers de la Figure 10.8, une relative stabilité des résultats pour des valeurs de 75 à 250 pixels. En dessous de 75, les résultats sont plus faibles, car justement notre stratégie d'agrégation repose en partie sur cette étape d'extension. Pour des valeurs supérieures à 250 on observe nettement pour les deux chaînes de traitement, une baisse de la performance globale liée à la baisse de la précision. En effet, plus l'expansion sera importante plus l'on aura tendance à «récupérer» des composantes non-texte, le nombre de faux positifs augmentera au détriment de la précision de nos systèmes.



(a) Chaîne Transformée (b) Chaîne Indicatrice

FIG. 10.7: Pour S fixé à 9, influence du paramètre $Stop$, la valeur d'arrêt de l'ouvert ultime exprimée en pourcentage de la taille de l'image. Pour la chaîne "Transformée" et la chaîne "Indicatrice". Métrique utilisée $WolfOverall\ tr = 0.6\ tp = 0.4$ (cf. 190).



(a) Chaîne Transformée (b) Chaîne Indicatrice

FIG. 10.8: Pour S et $Stop$ fixés respectivement à 9 et 30% ; influence du paramètre $Size_{Expand}$, extension des germes en pixels. Pour la chaîne Transformée et la chaîne Indicatrice. Métrique utilisée $WolfOverall\ tr = 0.6\ tp = 0.4$ (cf. 190).

Dans ce paragraphe, nous avons montré que la dépendance de nos chaînes de traitement aux paramètres S, $Stop$ et $Size_{Expand}$ n'est pas aussi critique qu'on pouvait le craindre (tout du moins sur la base de la campagne officielle d'$imagEval$). Cependant nous nous devons de rappeler que les courbes présentées représentent des «moyennes», qui n'indiquent pas si les résultats obtenus en faisant varier nos paramètres seront psychovisuellement satisfaisants pour un utilisateur donné.

10.2.3.4 Résultats qualitatifs

La Figure 10.9 permet d'observer qualitativement nos résultats de localisations sur un jeu d'images issu du test officiel et utilisant la chaîne indicatrice. Nous fournissons pour chacune, les *scores* obtenus à l'aide des métriques $ICDAR$ et $WolfPontual$.

Nous observons que les résultats sont satisfaisants. Les faux positifs observés sont généralement dus à la classification comme texte de zones structurées, comme des barrières (cf. Figure 10.9(g)), des grilles de balcon (cf. Figure 10.9(f)), des alignements de fenêtres (cf. Figure 10.9(c)).

Il existe également des faux négatifs : nos systèmes, basés sur une première approche de segmentation, ne sont pas à même de localiser des textes *sur des fonds non coopératifs* ; des mots aux caractères *coller* ; ou des textes *manuscrit* et notre module de regroupement spatial ne prend pas en compte les textes verticaux. Les images illustrant ces cas sont présentées sur la Figure 9.1 p.133.

La Figure 10.9 permet une étude comparative sur la pertinence des métriques. On observe ainsi toutes les ambiguïtés de la métrique $ICDAR$ vis-à-vis de l'annotation de la base. Ainsi pour les Figures 10.9(a) et 10.9(b), les scores de la métrique $ICDAR$ sont *particulièrement* faibles, ce qui est dû simplement à un faible morcellement de nos localisations. La métrique proposée par Wolf and Jolion [111] donne des évaluations plus proches de notre impression psychovisuelle. Cependant la Figure 10.9(b), illustre un cas extrême où la différence de granularité est trop importante ; différence qui ne peut être prise en compte par la métrique de $Wolf$. Notons 1. qu'il faudrait abaisser si drastiquement les paramètres tr et tp que la métrique proposée perdrait tout son sens ; 2. qu'il ne s'agit pas d'un cas isolé, on retrouve ce problème pour quelques cartes postales où toute une légende a été annotée et où nous ne localisons qu'un sous-ensemble de celle-ci.

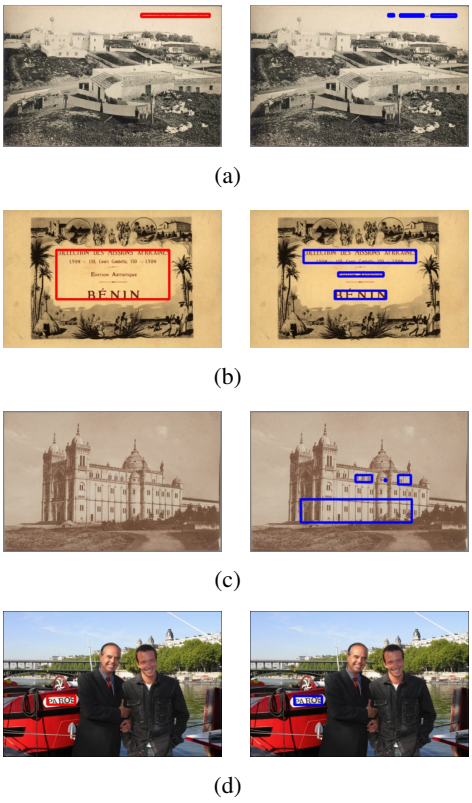
On pourra simplement conclure, qu'on peut ne pas toujours découpler un algorithme de localisation de l'annotation (au niveau géométrique) de la base sur laquelle il est évalué.

Nous présentons enfin sur la Figure 10.10 quelques résultats complémentaires de localisations sur la base de test à blanc et utilisant l'approche Transformée.

10.2.3.5 Temps de traitement

Nous proposons au travers des tableaux 10.6 et 10.7, le temps de calcul actuel de nos chaînes de traitement pour les paramètres explicités en Section 10.2.3.1. Les chaînes sont actuellement à l'état de prototype de recherche, aussi les temps indiqués restent prohibitifs (nous y reviendrons dans la Section 10.4 où nous donnerons des pistes simples pour diminuer substantiellement les temps de calcul).

Nous observons un temps moyen de 7 à 8 secondes par image dont une grande partie est consommée par l'opérateur d'ouvert ultime et le module de filtrage fin basé sur l'apprentissage. C'est sur ceux-ci qu'il faudra se concentrer pour améliorer les vitesses de traitement. On aura noté que les modules de «filtrages grossiers» bien que très simples, prennent un temps conséquent, il s'agit ici d'un faux problème lié à l'utilisation de fonctions non optimisés pour une gestion «rapide» d'image bidimensionnelle.



(a)

(b)

(c)

(d)

Précision	Rappel	M-Harmonique
$ICDAR$		
0.349	0.481	0.404
$Wolf$		
0.8	0.8	0.8
Précision	Rappel	M-Harmonique
$ICDAR$		
0.211	0.420	0.281
$Wolf$		
0.0	0.0	0.0
Précision	Rappel	M-Harmonique
$ICDAR$		
0.0	0.0	0.0
$Wolf$		
0.0	0.0	0.0
Précision	Rappel	M-Harmonique
$ICDAR$		
0.961	0.961	0.961
$Wolf$		
1.0	1.0	1.0

Exemples de résultats de localisation utilisant l'approche Indicatrice (Première partie). De gauche à droite : vérité terrain, nos résultats de localisation, évaluation des performances (Métrique ICDAR (cf. p. 186), $WolfPonctual$ (cf. p. 187))

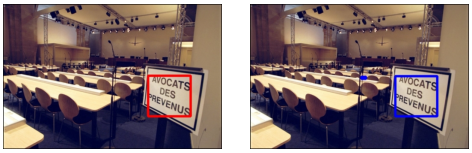
Temps de calcul par module en seconde	
Ouvert Ultime	3
Seuillage adaptatif	0.35
Filtrage Grossier	2.11
Filtrage Fin	2.41
Regroupement Itératif	0.37
Temps moyen cumulé	
8.2	

TAB. 10.6: Chaîne Transformée

Temps de calcul par module en seconde	
Ouvert Ultime	3
Filtrage Grossier	1
Filtrage Fin	2.51
Regroupement Itératif	0.51
Temps moyen cumulé	
6.9	

TAB. 10.7: Chaîne Indicatrice

TAB. 10.8: Temps de calcul de nos chaînes de traitement : les résultats sont moyennés sur une vingtaine d'images (taille moyenne 1024*655) de la base officielle $imagEval$. Le test a été réalisé sur un Pentium©Core Duo cadencé à 2.4 Ghz et disposant de 2 Go de RAM.

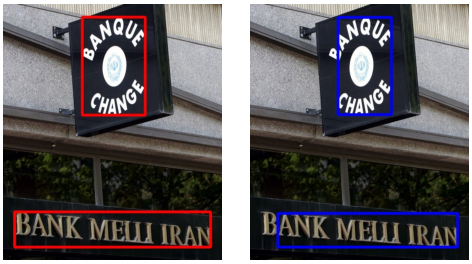


(a)

Précision	Rappel	M-Harmonique
$ICDAR$		
0.493	0.987	0.658
$Wolf$		
0.5	1.0	0.667

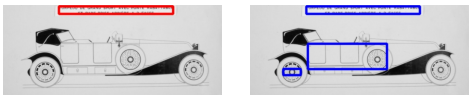
(b)

Précision	Rappel	M-Harmonique
$ICDAR$		
0.615	0.367	0.460
$Wolf$		
0.8	0.8	0.8



(c)

Précision	Rappel	M-Harmonique
$ICDAR$		
0.925	0.925	0.925
$Wolf$		
1.0	1.0	1.0



(d)

Précision	Rappel	M-Harmonique
$ICDAR$		
0.332	0.997	0.498
$Wolf$		
0.333	1.0	0.5

Exemples de résultats de localisation utilisant l'approche Indicatrice (Seconde partie). De gauche à droite : vérité terrain, nos résultats de localisation, évaluation des performances (Métrique ICDAR (cf. p.186), $WolfPonctual$ (cf. p. 187))

(b)

Précision	Rappel	M-Harmonique
$ICDAR$		
0.476	0.953	0.635
$Wolf$		
0.5	1.0	0.67

(c)

Précision	Rappel	M-Harmonique
$ICDAR$		
0.869	0.869	0.869
$Wolf$		
1.0	1.0	1.0

(d)

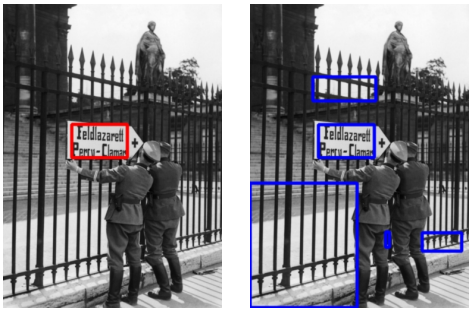
Précision	Rappel	M-Harmonique
$ICDAR$		
0.985	0.985	0.985
$Wolf$		
1.0	1.0	1.0

(e)

Précision	Rappel	M-Harmonique
$ICDAR$		
0.591	0.662	0.664
$Wolf$		
0.8	0.8	0.8

(f)

Précision	Rappel	M-Harmonique
$ICDAR$		
0.0	0.0	0.0
$Wolf$		
0.0	0.0	0.0



(g)

Précision	Rappel	M-Harmonique
$ICDAR$		
0.198	0.992	0.331
$Wolf$		
0.2	1.0	0.333

FIG. 10.9: Exemples de résultats de localisation utilisant l'approche Indicatrice (Troisième partie). De gauche à droite : vérité terrain, nos résultats de localisation, évaluation des performances (Métrique ICDAR (cf. p.186), $WolfPonctual$ (cf. p. 187))

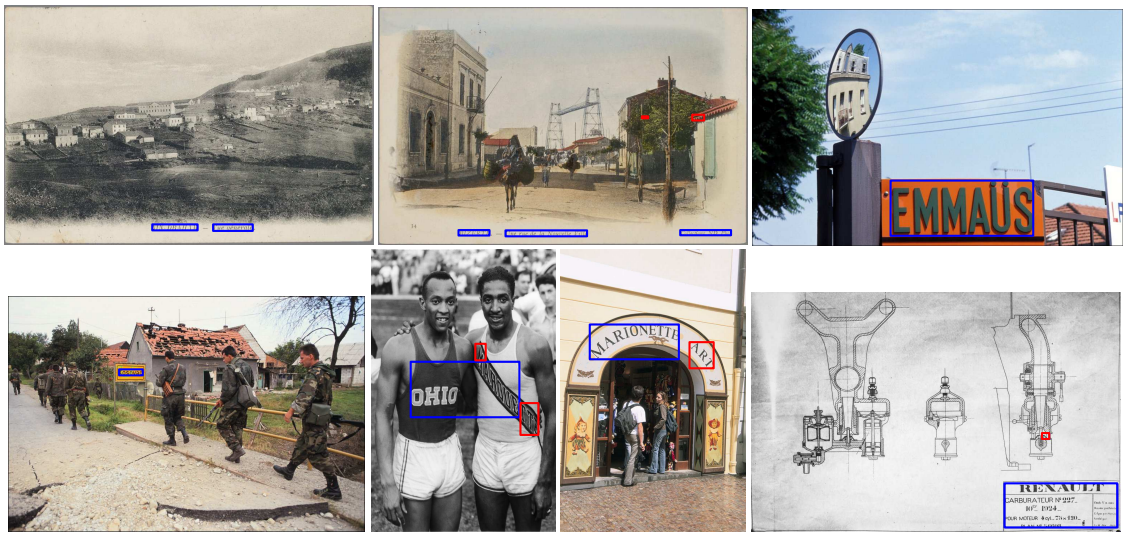


FIG. 10.10: Résultats complémentaires de localisation. Images issues de la base de test à blanc ImagEval. En bleu nos résultats de localisation, en rouge mise en lumière des faux positifs et faux négatifs

Nous venons de décrire de manière fine les résultats que nous avons obtenus au sein de la campagne $imagEval$. Nous proposerons dans la sections suivante des résultats complémentaires sur une autre base pour valider nos approches.

10.3 Position de nos travaux, Traitement de la base ICDAR

10.3.1 Finalité de cette étude

Nous venons d'exposer nos résultats au travers de la campagne d'évaluation $imagEval$,obtenant des résultats satisfaisants. Cependant cette base n'est pas librement accessible pour la communauté des chercheurs en *détection de texte*. Dans la section suivante nous proposons d'étudier le comportement de nos algorithmes sur une base libre de droit et faisant office de référence pour la localisation de texte enfoui.

10.3.2 Présentation de la base

La base ICDAR est décomposée en deux grands ensembles :

1. Une base librement accessible[6] ; composée de deux sous-ensembles d'apprentissage et de validation, de respectivement 258 et 251 images. Les images ayant été annotées par le comité ICDAR.
2. Une base non diffusée, permettant l'évaluation des algorithmes.

Deux campagnes d'évaluation ont été réalisées en 2003 et 2005, les résultats ont été respectivement publiés dans [53] et [52].

Ces campagnes comprenaient plusieurs volets : 1. évaluation des techniques de localisation ; 2. évaluation de la reconnaissance de mots ; 3. évaluation de la reconnaissance de caractères ; plus une «méta-tâche» permettant de proposer un système complet. Notons que seul le premier volet a reçu des contributions.

La base est constituée quasi-exclusivement de texte enfoui : panneaux de signalisation, noms de rues, affiches publicitaires, et de photos de couvertures de livres et de magazine. La plupart des textes sont sur des supports planaires et sont «horizontaux».

10.3.3 Annotation et métriques

10.3.3.1 Illustration des annotations

A l'opposé des bases précédentes, la vérité terrain fournie par le comité d'organisation ICDAR est relativement fine : la granularité se situe au niveau «mot», voire lettres quand celles-ci sont isolées. Sur la Figure 10.11, on peut ainsi observer que le comité a annoté des lettres isolées, et que certaines boîtes de la vérité terrain peuvent être incluses les unes dans les autres.

10.3.3.2 Métriques

Aucun aménagement particulier des métriques n'a été réalisé.

10.3.4 Résultats

10.3.4.1 Chaînes de traitement et paramètres utilisés

N'ayant pas accès à la base servant à l'évaluation finale des algorithmes ; les résultats proposés par la suite sont fournis sur la base *librement accessible*. Plus précisément nous avons fusionné les

[6] Les images et les vérités terrains sont disponibles sur `http://algoval.essex.ac.uk/icdar/Datasets.html`

FIG. 10.11: Présentation de la base ICDAR et de la finesse de son annotation.

deux sous-ensembles d'apprentissage et de validation (de respectivement 258 et 251 images) fournis par le comité I.C.D.A.R.

Les chaînes de traitement sont essentiellement les mêmes que celles utilisées dans le cadre de la campagne ImagEval. La seule modification concerne le module de regroupement spatial : le prédicat de fusion d'$imagEval$ (cf Figure 10.4, qui se justifiait par la granularité grossière de la vérité terrain) est remplacé par un simple prédicat d'inclusion.

10.3.4.2 Résultats quantitatifs

Les tableaux 10.9 et 10.10 résument les performances *globales* de nos chaînes de traitement sur l'ensemble de la base.

Comme précédemment nous pouvons extraire de ces tableaux deux remarques importantes :

1. Pour les deux chaînes, on observe que le nombre de boîtes localisées est bien inférieur au nombre de boîtes annotées. Ce qui indique que nous en *ratons* un grand nombre. Ce nombre représente de "vrai" faux négatifs mais également les problèmes que nous rencontrons avec la finesse de l'annotation d'ICDAR des zones de textes.
2. A l'instar du projet $imagEval$, il n'existe pas de différence notable en termes de performances globales pour les deux chaînes

Le premier point est à relier principalement à l'annotation fine de la base. En effet d'une part nous «raterons» systématiquement les zones de textes composées de moins de trois caractères et, d'autre part, le remplacement du prédicat de fusion d'$imagEval$ au profit d'un simple prédicat d'inclusion (cf Section 10.3.4.1) ne nous permettra de fournir des boîtes localisées en accord avec la granularité de la vérité terrain. Ceci est clairement illustré par la comparaison de la Figure 10.11 et de la Figure 10.13 représentant respectivement l'annotation de la base pour un sous-ensemble d'images et nos localisations sur celles-ci.

Le second point indique que la base comporte peu d'images pour lesquelles la présence de bruit est un obstacle à l'utilisation directe de l'indicatrice comme première segmentation de référence.

Le but principal de cette étude était de valider nos approches sur une base libre et commune à la communauté des chercheurs et ainsi de positionner nos travaux vis-à-vis de l'état de l'art. Comme nous l'avions mentionné précédemment les seules données disponibles (en termes de performances globales) sont issues des papiers de Lucas et al. [53] et Lucas [52]. Les évaluations ont été réalisées sur une base *non diffusée*, mais dont les caractéristiques doivent être proches de la base servant à l'entraînement et paramétrage des systèmes (i.e. celle sur laquelle nous avons réalisé nos évaluations). Aussi, et bien que cela soit discutable, nous pourrions extrapoler nos résultats sur la campagne d'évaluation officielle ce qui nous placerait dans le «peloton de tête» : pour rappel les deux meilleurs systèmes de 2003 obtiennent respectivement des performance globales de $ICDAR_{Hmean} = 0.45$ et $ICDAR_{Hmean} = 0.50$ et les deux meilleurs de 2005, respectivement $ICDAR_{Hmean} = 0.58$ et $ICDAR_{Hmean} = 0.62$ (la performance des autres systèmes étant inférieure à 0.4).

Il faudrait bien sûr disposer de la base d'évaluation pour confirmer cette assertion. On notera de plus qu'un effort de notre part en termes de vitesse de traitement doit être réalisé pour rivaliser pleinement avec les systèmes proposés (pour information les temps de calculs présentés, indique un temps de traitement moyen par image de respectivement 0.35s et 14.4s pour les deux systèmes les plus performants.

En conclusion, nous pouvons cependant dire que l'objectif est atteint ; nous obtenons des résultats satisfaisants sur cette base, alors qu'aucun travail spécifique n'a été réalisé pour améliorer les résultats.

Précision	Rappel	M-Harmonique
$ICDAR$		
0.53	0.41	0.46
$WolfPonctual$		
0.51	0.51	0.51
$WolfOverall$		
0.50	0.48	0.49
Nb de Boîtes Vérité Terrain		2263
Nb de Boîtes Localisées		1346

TAB. 10.9: Performances globales de la chaîne transformée sur la base ICDAR ($tr = 0.8$; $tp = 0.4$)

Précision	Rappel	M-Harmonique
$ICDAR$		
0.53	0.41	0.46
$WolfPonctual$		
0.50	0.50	0.50
$WolfOverall$		
0.48	0.48	0.48
Nb de Boîtes Vérité Terrain		2263
Nb de Boîtes Localisées		1378

TAB. 10.10: Performances globales de la chaîne indicatrice sur la base ICDAR ($tr = 0.8$; $tp = 0.4$)

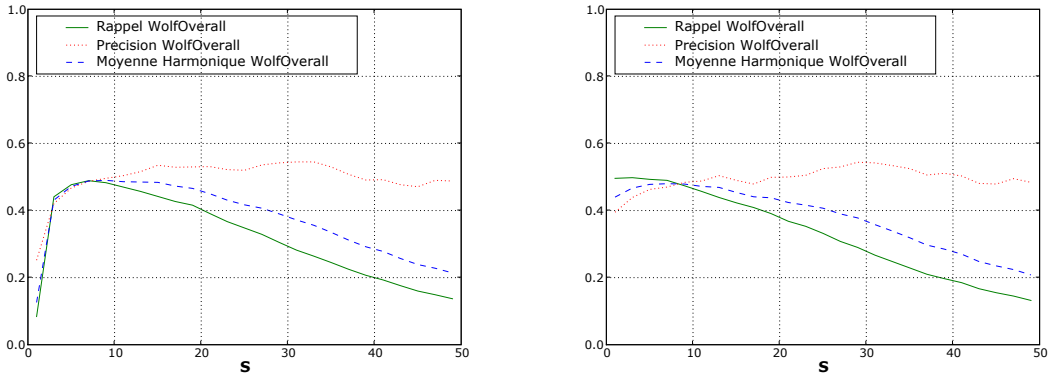
10.3.4.3 Influences des paramètres

Le choix d'un seuil paramétrique étant discutable nous proposons une nouvelle fois de regarder son influence sur les résultats quantitatifs de nos approches. Celle-ci est présentée sur la Figure 10.12.

Comme pour les deux bases précédentes on retrouve logiquement un écart de sensibilité. L'influence de S est relativement faible pour la chaîne indicatrice, en début de courbe pour des valeurs de 0 à 15 on observe une relative stabilité de la performance globale du système (i.e. Moyenne Harmonique). La dépendance de la chaîne transformée est plus importante ; de très faibles scores en début de courbe puis une stabilisation pour des valeurs de 9 à 15.

Pour des valeurs plus importantes, les deux systèmes conservent une bonne précision, mais leur rappel chute, entraînant avec lui la performance globale du système. Ce qui indique que lorsque le seuil augmente nous perdons au fur et à mesure des boîtes correctement localisées.

En conclusion et à l'instar de nos résultats sur la base $imagEval$, il semble préférable de se tourner vers la chaîne Indicatrice, dont la dépendance aux paramètres S est la plus faible.



(a) Chaîne Transformée (b) Chaîne Indicatrice

FIG. 10.12: Base ICDAR : Influence du paramètre S, Seuil bas de la transformée. Pour la chaîne Transformée et la chaîne Indicatrice. Métrique utilisée $WolfOverall\ tr = 0.8\ tp = 0.4$ (cf Section 10.1.2.4).

10.3.4.4 Résultats qualitatifs

La Figure 10.13 permet d'observer qualitativement nos résultats de localisation sur le jeu d'images proposé en Figure 10.11. Les résultats présentés ont été obtenus au travers de la chaîne Indicatrice (les paramètres sont explicités en p.206).

On observe clairement la différence de granularité entre les résultats que nous proposons et l'annotation réalisée. Quand les zones de textes sont «proches» et/ou dans le cas d'inclusion, notre système renvoie quasi-systématiquement une boîte localisée regroupant plusieurs boîtes de la vérité terrain. Nous pouvons également observer une absence totale de localisation pour les caractères isolés et la tendance de nos systèmes à favoriser la précision au détriment du rappel (pour l'échelle de nos localisations). On retrouve classiquement quelques faux positifs dans les zones structurées de l'image, et la présence de faux négatifs pour des images où les zones de textes sont *mal définies* (Image de la poubelle et de l'étiquette du jeans).

FIG. 10.13: Résultats qualitatifs de localisation sur la base ICDAR

10.4 Discussion Générale

Dans les sections précédentes nous avons présenté les résultats de nos travaux sur deux grandes bases d'images issues de différents projets : 1. $imagEval$; 2. base "ICDAR" pour nous positionner vis-à-vis de l'état de l'art.

Un bilan général positif

Les chaînes présentées ont été mises en place pour la campagne d'évaluation $imagEval$. Il s'agissait de traiter des bases extrêmement hétérogènes (eg : texte à différentes échelles, présence de texte non horizontal et/ou courbe (i.e. "WordArt"), légende et texte enfoui, support non plan) mais représentatives des attentes actuelles des détenteurs de grands fonds photographiques. Nos systèmes ont pu prendre en compte cette hétérogénéité et fournir des résultats très satisfaisants. Ceux-ci feront office de référence pour les futures campagnes.

Ensuite nous les avons appliqués sur la base d'évaluation d'ICDAR pour nous positionner vis-à-vis de l'état de l'art. Nous avons également obtenu des résultats très prometteurs ; en effet nous n'avons pas tiré parti des fortes hypothèses d'horizontalité des textes de cette base et là encore aucun apprentissage dédié, ni optimisation globale n'ont été réalisés. De plus, nous avons souligné (comme d'autres avant nous) les problèmes de finesse d'annotation de cette base. Un ajustement du module de regroupement spatial pourrait être envisagé, pour améliorer les scores quantitatifs de nos approches.

A la lumière de ces résultats, nos chaînes ont prouvé leur capacité de généralisation.

Ce bilan, certes satisfaisant, ne nous dispense pas de réexaminer nos méthodes d'un oeil critique et de proposer des pistes d'amélioration.

1. Paramètres : Les chaînes que nous avons présentées sont paramétriques. Cependant le nombre total de paramètres n'est pas prohibitif au vue de la variabilité des textes que nous avons eu à traiter. Bien que nous ayons identifié un paramètre "sensible" : le seuil actuellement *fixe* de la transformée, les autres paramètres pourraient être appris/optimisés notamment pour des bases moins hétérogènes. Dans le chapitre 9, nous avons discuté de la difficulté de juger de la pertinence de chaque module séparément et du besoin de définir des vérités terrain à différentes "échelles" (i.e. il faudrait une complémentarité entre une annotation au niveau boîte et au niveau composante connexe).
2. Précision Versus Rappel : Nos systèmes privilégient généralement la précision au détriment du rappel. On peut considérer cette stratégie comme un atout ou un défaut en fonction de l'application visée. En effet, bien qu'il s'agisse d'une qualité dans le cadre du démonstrateur réalisé pour EADS, cela est plus discutable dans un contexte d'indexation au sens large. On pourrait en effet préférer un système privilégiant dans un premier temps le rappel puis lui adjoindre des modules additionnels pour réduire progressivement le nombre de faux positifs. Notons que nous pourrions modifier nos propres systèmes en ce sens. On pourrait notamment relaxer les contraintes actuelles du module de regroupement spatial et modifier la «fonction de score» de notre module d'apprentissage pour diminuer le nombre de faux négatifs. Il faudrait ensuite proposer de nouveaux modules de filtrage qui pourraient se situer à deux niveaux. Premièrement au niveau boîte (i.e. la boîte que je viens de localiser possède-t-elle les propriétés d'une zone de texte ?), deuxièmement au niveau des composantes connexes où nous pourrions faire intervenir un O.C.R comme module de validation. Nous discuterons de ce point dans la Section 10.5.
3. Limitations des approches par segmentation : limitations inhérentes aux systèmes utilisant l'ap-

proche par composantes connexes, nos systèmes sont dépendants d'une première phase de segmentation de l'image. Celle-ci peut échouer dans le cas d'images fortement bruitées, compressées et/ou de trop faible résolution. Notons que nous travaillons sur des images fixes, ce qui nous prive des techniques généralement utilisées pour la recherche de texte dans les vidéos, qui tirent parti de la redondance temporelle (moyennage temporel, super-résolution). Toutefois pour le cas de la recherche de texte de faible résolution dans des images fixes il semble plus sage de se tourner vers des approches dites "textures", qui généralement déplacent le problème de segmentation après une première phase de localisation. Nous en reparlerons dans les perspectives générales de ce manuscrit.

4. Temps de traitement : les temps de traitement actuels restent prohibitifs pour une utilisation "intensive" de nos approches. Il s'agit ici d'un «faux» problème dans le sens où nos travaux restent à l'échelle d'un prototype de recherche. En effet, un effort d'ingénierie devra être effectué pour réduire les coûts algorithmiques. Des gains substantiels sont facilement réalisables par des méthodes «classiques» : calcul au sein d'une même "passe" des descripteurs, réduction drastique du nombre d'images intermédiaires, réduction de l'application de l'algorithme à un sous-ensemble de l'image préalablement identifié comme région d'intérêt . . . Enfin, notons que nos algorithmes ont été définis au sein d'une librairie propre à notre laboratoire (Morph-M) privilégiant la généricité (support d'image à N dimensions, pour tous types de données) à la vitesse d'exécution. La spécialisation de certaines fonctions à des images bi-dimensionnelles et pour des types scalaires "simples" (images sur 8 bits et 16 bits) permettrait d'atteindre des gains importants. Cependant il semble difficile de descendre ces coûts au niveau des meilleurs systèmes présentés lors des évaluations de la campagne ICDAR. Pour rappel lors de la campagne de 2005, le système "Alex Chen" a obtenu une performance globale de 0.58 avec un temps de traitement moyen par image de 0.35 seconde. Ce dernier est d'ailleurs proposé sous la forme d'un service web à l'adresse suivante `http://algoval.essex.ac.uk:8080/textloc/upload.html`.

Nous venons de faire le tour des trois grandes bases que nous avons eu à traiter. Nous avons discuté des qualités et faiblesses de nos systèmes pour chacune. Nous aimerions pour finir ouvrir une parenthèse sur des premiers résultats de reconnaissance de caractères.

10.5 Quelques résultats préliminaires de reconnaissance

Nous proposons dans cette section quelques résultats préliminaires de reconnaissance de texte. Comme nous l'avons souligné dans la Section 9.7, il s'agit davantage d'entamer ici une discussion sur l'intérêt et les défis que soulève l'utilisation d'un logiciel de reconnaissance de caractères en sortie de nos chaînes de traitement.

10.5.1 But et limitations actuelles

L'intégration d'un O.C.R a deux intérêts :

1. Passer de la simple localisation à la reconnaissance, et donc proposer un système «complet»
2. Réduire un certain nombre de fausses alarmes par l'étude des sorties de l'OCR

La Figure 10.14 propose des exemples de reconnaissances sur la base $imagEval$. Notons qu'aucun effort n'a été réalisé pour améliorer les résultats de reconnaissance. Sur les Figures 10.14(a), 10.14(d), 10.14(e), 10.14(h), nous pouvons effectivement observer des premiers résultats satisfaisant indiquant que nous pouvons effectivement déjà extraire des premiers résultats de reconnaissance.

Les Figures 10.14(b), 10.14(f), quant à elles, sont des faux positifs de notre part, nous détectons des séries de i ou de l alors qu'il s'agit de fenêtres ou de grilles. Nous pourrions espérer dans ce cas, en regardant les scores de confidences de l'OCR vis-à-vis des chaînes extraites et par l'intégration de «grammaires» filtrer ces faux positifs.

Cependant les Figures 10.14(c), 10.14(g) nous rappellent à juste titre que les composantes connexes que nous fournissons et/ou les capacités de reconnaissance de l'OCR ne sont pas en adéquation.

10.5.2 Conclusion

Nous obtenons certes de premiers résultats de reconnaissance intéressants. Mais le chemin est encore long pour proclamer que nous avons construit un système complet. En effet aucun effort de notre part n'a été réalisé pour *restaurer* les caractères avant l'étape de reconnaissance et il serait inapproprié de rejeter l'ensemble des erreurs de reconnaissance sur l'O.C.R utilisé. La mise en place d'un module de restauration (comprenant eg :correction d'échelle, de perspectives) doit être réalisée, ainsi que la mise en compétition de différents O.C.R et/ou un apprentissage dédié de ceux-ci.

(a)

OCR Output	
White	
Black	MY CARTHAGE Vue Generals

(b)

OCR Output	
White	WH IM we
Black	Uu L va \TT\ I

(c)

OCR Output	
White	YEMOIGNAGE PMOTQGRAPHIQIIE
	[ii IQBEITI CAPA Sill
	[A GUERRE CIVIL ! 55 AGNOLF
Black	

(d)

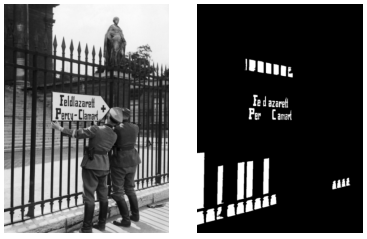
OCR Output	
White	
Black	HAUR BE D PEACE PEACE

(e)

OCR Output	
White	TERRITORIO LIBRE DE AMERICA
Black	REPUBLICA DE CUBA

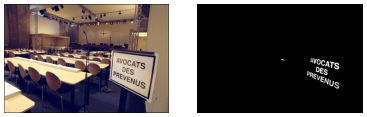
(f)

OCR Output	
White	tll I
Black	



(g)

OCR Output	
White	IIÍ¨ ?. I‘ ! llllll ma
Black	In cl BZBFEH PB ?’ [Emit‘ !



(h)

OCR Output	
White	
Black	AvocATS *ÕES PREVENUS

FIG. 10.14: Quelques résultats de *Tesseract*©sur la base *ImagEval* ; Gauche :Image Originale ; Milieu : ensemble des composantes connexes annotées texte (Pour les deux polarités du texte) par une chaîne de traitement, Droite : Résultat de l’O.C.R sous forme de chaînes de caractères.

11 Conclusion Générale

Le premier objectif de ce travail était de fournir aux systèmes de reconnaissance d'image par le contenu un nouveau descripteur de haut niveau : le texte présent dans l'image. La difficulté de cette tâche sur des bases généralistes a restreint cette motivation à la «seule» tâche de localisation. Nous avons pu apprécier le long de ce manuscrit que le terme «seule» est relativement réducteur au vu des défis à relever.

Nous aimerions souligner ce que nous considérons comme les deux apports principaux de nos travaux.

11.1 Apport de cette Thèse

11.1.1 Opérateurs Résiduels Numériques

Dans la seconde partie de ce manuscrit, une analyse approfondie sur l'opérateur d'ouverture ultime a été réalisée. Il s'agit d'un outil puissant et non paramétrique permettant de se passer d'a priori sur la taille des objets d'étude en privilégiant l'information de contraste des structures d'intérêt. Il permet notamment de réaliser des études granulométriques d'une image à teintes de gris sans passer par une étape préalable de binarisation (cf. travail de doctorat de OUTAL [67]), mais également de proposer des segmentations "automatiques" des images.

S'agissant d'un opérateur relativement récent, son application sur des images complexes nous a permis de mieux cerner ses forces et ses faiblesses. Cette analyse nous a appris beaucoup sur son comportement, nous avons notamment mis en lumière certaines de ses myopies (en présence de structures imbriquées et de transitions graduelles) et nous avons proposé différentes stratégies pour y pallier.

Nous avons également montré l'intérêt de son extension à d'autres familles d'ouvertures, notamment par critère, dans le cas où nous ne pouvons pas utiliser une hypothèse sur une «convexité» des objets étudiés. Enfin nous avons ouvert diverses pistes de réflexion concernant ses possibles extensions (eg : utilisation sur le gradient des images permettant l'exploitation de l'information couleur, de hiérarchies de résidus). Nous en reparlerons dans les perspectives de nos travaux.

Parallèlement à cette étude théorique et pratique nous avons proposé une première implémentation efficace de celui-ci (pour un sous-ensemble de cas), permettant de l'intégrer aujourd'hui comme «brique algorithmique», lui permettant de rejoindre la boîte à outils des méthodes de segmentation morphologique. Cette implémentation a donné lieu à la publication suivante : [72].

11.1.2 Localisation de textes dans des bases diversifiées

Le coeur de notre travail prenait place au sein de la campagne d'évaluation ImagEval. Il s'agissait de traiter une base d'images très hétérogène (eg : texte à différentes échelles, présence de "WordArt"

(i.e.texte non horizontal ou courbe), légende et texte enfoui, support non plan) mais représentative des attentes actuelles des grand détenteurs de fonds photographiques.

Nous avons bâti notre approche sur un nouvel opérateur, l'ouvert ultime, étudié en profondeur dans la deuxième partie de ce manuscrit. Malgré sa puissance, il ne résout pas à lui seul tous les problèmes. C'est pourquoi nous avons mis en place un ensemble d'outils complémentaires nécessaires pour le bon déroulement de l'application. Parmi ces outils, nous pouvons citer le module d'apprentissage qui, avec des descripteurs simples, donne un taux d'erreur de l'ordre de 10% (aussi bien de faux positifs que de faux négatifs). Le module de regroupement des caractères, qui joue aussi un rôle important pour la réduction de faux positifs. Celui-ci, tout en relâchant les contraintes pour détecter des zones de texte non horizontales (WordArt, effet de perspective, support non plan) est assez restrictif (au moins trois caractères validés par l'apprentissage) pour éviter un nombre important de faux positifs. Nous avons proposé deux chaînes de traitement, une basée sur la transformée (information de contraste issue de l'ouvert ultime) et une autre basée sur l'indicatrice (segmentation avec une information de taille fournie aussi par l'ouvert ultime). Celles-ci ont pu prendre en compte cette hétérogénéité (à l'exception des textes "verticaux" et manuscrits) et les résultats que nous proposons sont très satisfaisants et feront office de référence pour les campagnes à venir. Notons que chacune d'elles a fait l'objet d'une publication : [71] pour la chaîne transformée, [70] pour la chaîne indicatrice.

Ces chaînes ont montré leur capacité de généralisation sur deux autres projets : le développement d'un démonstrateur pour EADS, et le traitement de la base de référence d'I.C.D.A.R au prix d'ajustements minimes de notre part.

L'approche indicatrice est plus simple et moins paramétrique que l'approche transformée, mais elle «résiste» moins bien à la présence de bruit ou à la faible résolution des images. Cette constatation provient des travaux réalisés pour la société EADS qui n'ont pas pu être intégrés au corps de ce manuscrit.

Nos chaînes actuelles sont un nécessaire compromis entre qualité de localisation, temps de réalisation et vitesse de traitement. Nous avons volontairement opté pour une approche modulaire au détriment de la vitesse de nos systèmes. Cette approche permettra de changer à loisir les différents modules proposés pour d'autres plus performants en fonction d'une application donnée. Ces modules pourront tirer profit d'un travail d'ingénierie plus poussé permettant de réduire de manière substantielle le temps de calcul et d'ouvrir la voie à des utilisations "intensives" comme le filtrage de bases de données de grand volume.

11.2 Perspectives

Moyen terme

Consolider l'existant

Un des premiers axes de progrès vise à la «consolidation» en termes de performances et de fiabilité des modules que nous avons réalisés.

Premièrement, il faudrait remplacer le module de seuillage «fixe» retenu actuellement par une méthode *automatique*. Certes nous pourrions *apprendre* un seuil *optimal* pour une base donnée, mais ceci restreindrait notre capacité de généralisation ; de plus cette stratégie ne permettrait pas de rendre nos approches robustes à des variations globales ou locales d'illumination sur une même base d'images. En ce sens une étude approfondie des méthodes de seuillage automatique doit être envisagée.

Deuxièmement, nous avons étudié la sensibilité de nos systèmes aux différents paramètres que nous avions, dans un premier temps, fixés expérimentalement. Le jeu de paramètres retenus s'est fi-

nalement avéré robuste pour les trois projets auxquels nous avons participé. Cependant nous avons souligné qu'invariablement nous n'avons pu fournir que des optimums *locaux* de performances de nos systèmes. Pour améliorer ce fait, il faudrait soit optimiser *globalement* nos systèmes (notons que la combinatoire s'avérerait sans doute importante), soit proposer des métriques intermédiaires pertinentes (i.e. donc décorrélées les unes des autres) pour optimiser séparément chacun des modules. Nous pourrions ainsi à loisir remplacer des modules existants par d'autres, proposer des stratégies en cascades de filtres etc... Cette approche pourrait être envisageable dans un cadre applicatif plus restreint, pour lequel la définition de vérité terrain pour différentes granularités (i.e complémentarité d'une annotation au niveau boîtes et au niveau composante) s'avérerait pertinente. Nous pouvons également supposer que les performances du module de filtrage fin basé sur l'apprentissage et le module itératif de regroupement spatial pourraient être améliorés sur des bases au contenu plus contraint. Idéalement ces deux modules devraient être fusionnés, permettant au module d'apprentissage de bénéficier de la cohérence spatiale du texte.

Notons enfin que la métrique *la moins biaisée*, reste celle associée à la lecture automatique des chaînes de caractères présentes dans l'image. Elle nécessite de franchir le cap de la reconnaissance du texte et de valider nos systèmes à cette échelle.

Franchir le cap de la reconnaissance

A la fin du Chapitre 10, exposant le résultats de nos travaux, nous avons présenté quelques résultats de reconnaissance de texte. Utilisant une approche par composante connexe, nous disposons de tout ou partie des lettres qui ont formé une boîte localisée. L'utilisation *directe* d'un O.C.R nous a fourni de premiers résultats encourageants. Cependant, il serait très prématuré d'affirmer que la création d'une chaîne complète est *à portée de main*. Un travail approfondi concernant la restauration des caractères doit être entrepris. Cependant sans aller jusqu'à la reconnaissance, l'O.C.R pourrait dans un premier temps fournir une méthode de filtrage des faux positifs : quand celui-ci nous renvoie une série de "i" ou de "l" dans une zone de texte, il s'agit plus vraisemblablement d'une rambarde ou d'une grille que véritablement du texte.

Utilisation de nos travaux sur les opérateurs résiduels

Dans la seconde partie de ce manuscrit, une analyse approfondie de l'opérateur d'ouverture ultime a été réalisée. Nombre de nos observations n'ont pas trouvé place par la suite dans nos chaînes de traitement.

Aussi, nous n'avons pas tiré parti d'une approche utilisant *un gradient* de l'image, or ce type de représentation pourrait nous permettre de traiter en une fois les textes sombres sur fond clair et les textes clairs sur fond sombre ; elle pourrait de même faciliter le traitement des textes couleur ou sur des fonds texturés par l'emploi de gradients adaptés.

Lors de l'exposition des myopies de l'opérateur en présence de structures imbriquées et de transitions graduelles, nous avons discuté de différentes stratégies pour les pallier. Nous avons notamment souligné premièrement l'intérêt *d'accumuler* les résidus le long des zones de transition pour refléter plus fidèlement le contraste *réel* des composantes et deuxièmement proposé l'utilisation de *hiérarchies* de résidus pour analyser l'imbrication de structures à différentes échelles.

Long terme

Tout au long de ce manuscrit, l'objet "texte" a montré une grande faculté de polymorphisme et mis en lumière s'il y en avait encore besoin, l'extraordinaire faculté de généralisation du cerveau humain.

En effet, nous avons pu apprécier, dans notre partie introductive puis lors de la présentation des bases que nous avons traitées, qu'il n'existe pas un mais des textes. Aussi, plutôt que de rentrer dans un jeu d'oppositions d'approches, il serait souhaitable de faire collaborer différentes stratégies de manière parallèle ou séquentielle. C'est une approche pragmatique que l'on retrouve classiquement dans le domaines des O.C.R.

Ainsi lors de la campagne d'évaluations ICDAR de 2003 (cf. Lucas et al. [53]), un «méta-moteur» de localisation a été mis en place combinant les apports des trois algorithmes les plus performants, celui-ci a finalement obtenu les meilleurs résultats globaux.

A Glossaire

Abréviation et Acronyme

$ICDAR$: Abréviation de "International Conference on Document Analysis and Recognition". Lorque nous l'utiliserons, nous ferons référence d'une part à la conférence en tant que telle et d'autre part à la base d'images annotée de la tâche de localisation de texte disponible à l'adresse suivante `http://algoval.essex.ac.uk/icdar/Datasets.html`, dans la rubrique "Robust Reading Competitions".

$imagEval$: Le projet ImagEVAL a trait à l'évaluation des technologies de filtrage, reconnaissance par le contenu et l'annotation automatique d'images sur de grandes bases de données généralistes. La description de l'ensemble du projet est disponible à l'adresse `http://www.imageval.org/`.

Notation

Images et treillis associées :

Image Binaire : Soit E un sous-ensemble compact de $\mathbb{Z}^2$.Tout sous-ensemble compact $A \subset E$ sera appelé image binaire. On notera $\mathcal{X}(E)$ l'ensemble des sous-ensembles de E (i.e. l'ensemble des images binaires).

Image à teintes de gris : Soit $T = [t_{min}, \ldots, t_{max}]$ un ensemble de nombres réels appelés valeurs de gris . Une image à teintes de gris I est une application $I : E \to T$. On notera $\mathcal{F}(E)$ l'ensemble des images à teintes de gris sur E.

Soit $I \in \mathcal{F}(E)$ une image et $x \in E$, $I(x)$ sera appelé altitude de x (pour I). On notera $I_t^+ = \{x \in E | I(x) \geq t\}$ avec $t \in T$ la section de I au niveau t , et $I_t^- = \{x \in E | I(x) < t\}$ la section du fond de I.

Une autre définition de l'altitude d'une image à teintes de gris permet le passage d'un opérateur binaire à un opérateur qui s'applique aux images de teintes de gris. On regardera une image comme un empilement d'ensembles décroissants, c'est-à-dire comme un empilement de sections : $I(x) = \sup\{t \mid x \in I_t^+\}$, $I(x)$ est le supremum de toutes les valeurs t, pour lesquelles la section de I au niveau correspondant contient x.

Structure de treillis associés aux images On associera aux images binaires la structure de treillis ensembliste et aux images à teintes de gris la structure de treillis des fonctions. On se reportera aux ouvrages [81, 82] pour une discussion complète sur les différentes structures de treillis associés aux images, aux partitions,...

B	Element Structurant (E.S)
	Les définitions suivantes ne sont correctes que si B est un élément structurant symétrique ,invariant par trans
g_B^-	Gradient par érosion utilisant l'élément structurant B
g_B^+	Gradient par dilatation utilisant l'élément structurant B
g_B	Gradient symétrisé utilisant l'élément structurant B
κ	un critère que l'on considérera comme croissant et planaire, sauf si cela est démenti dans le corps du texte
Γ^κ	Ouverture par critère
Φ^κ	Fermeture par critère
Γ_x	Ouverture connexe ponctuelle en x (Binaire)
γ_B	Ouverture morphologique unitaire utilisant l'E.S B
γ_λ	Ouverture morphologique de taille λ
φ_B	Fermeture morphologique unitaire utilisant l'E.S B
φ_λ	Fermeture morphologique de taille λ
γ^∞	Ouverture par reconstruction
φ^∞	Fermeture par reconstruction
υ	Fermeture Ultime au sens de Beucher [7]
ν	Ouverture Ultime au sens de Beucher [7]
υ^κ	Fermeture Ultime par critère
ν^κ	Ouverture Ultime par critère
$g \circ f$	Composition des applications g et f : $g \circ f(x) = g(f(x))$

Transformations Morphologiques

Propriétés des opérateurs morphologiques :

Extensivité Une transformation ψ est extensive, si le résultat que l'on obtient en l'appliquant est plus grand que l'image originale :

$$\forall A \in \mathcal{X}(E),\ A \subseteq \psi(A) \quad \text{ou} \quad \forall f \in \mathcal{F}(E),\ f \leq \psi(f)$$

On définira l'**anti-extensivité** par dualité.

Croissance Une transformation morphologique ψ est croissante si elle préserve l'ordre, c'est-à-dire si la relation d'ordre entre deux images tient toujours après l'application de l'opérateur :

$$\forall A, B \in \mathcal{X}(E),\ A \subseteq B \Rightarrow \psi(A) \subseteq \psi(B) \quad \text{ou} \quad \forall f, g \in \mathcal{F}(E),\ f \leq g \Rightarrow \psi(f) \leq \psi(g)$$

Idempotence Une transformation morphologique ψ est dite idempotente, si le résultat ne change pas si l'on applique la transformation une seconde fois :

$$\forall A \in \mathcal{X}(E),\ \psi(A) = \psi\psi(A) \quad \text{ou} \quad \forall f \in \mathcal{F}(E),\ \psi(f) = \psi\psi(f)$$

Dualité Deux transformations ψ_1 et ψ_2 sont appelées duales, si le complément du résultat que l'on obtient en appliquant ψ_1 est égal au résultat que l'on obtient en appliquant ψ_2 au complément de l'image originale :

$$\forall A \in \mathcal{X}(E),\ \psi_1(A^c) = (\psi_2(A))^c \quad \text{ou} \quad \forall f \in \mathcal{F}(E),\ \psi_1(f^c) = (\psi_2(f))^c$$

Terminologie algorithmique :

fifo : First In, First Out (premier entré, premier sorti) sera utilisé souvent dans les files d'attente. Signifie que le contenant conserve l'ordre d'entrée de ses éléments lors de la sortie.

fah : File d'attente hiérarchique. File d'attente dotée d'une relation d'ordre. L'extraction des éléments de la file respecte cette relation d'ordre.

Voisinage

$\mathcal{N}$ un voisinage quelconque
$\mathcal{N}_{(x)}$ un voisinage quelconque centré sur x
$\mathcal{N}_{(x)} \backslash x$ un voisinage quelconque centré sur x, privé de son centre

Références

[1] L.V. Ahn, M. Blum, N. Hopper, and J. Langford. Captcha : Using hard AI problems for security. In *Advances in Cryptology - EUROCRPYT 2003 : International Conference on the Theory and Applications of Cryptographic Techniques*, pages 294–311, Warsaw, POLAND, may 4-8 2003.

[2] P. Aigrain, H. Zhang, and D. Petkovic. Content-based representation and retrieval of visual media : A state-of-the-art review. *Multimedia Tools and Applications*, 3(3) :179–202, 1996.

[3] S. Antani and R. Kasturi. A survey on the use of pattern recognition methods for abstraction, indexing and retrieval of images and video. *Pattern Recognition*, 35(4) :945–965, April 2002.

[4] H.B. Aradhye, G.K. Myers, and J.A. Herson. Image analysis for efficient categorization of image-based spam e-mail. In *International Conference on Document Analysis and Recognition*, pages 914–918, Seoul, Korea, August 2005.

[5] H.S. Baird, A.L. Coates, and R.J. Fateman. Pessimalprint : a reverse Turing test. *International Journal of Document Analysis and Recognition (IJDAR)*, 5(2-3) :158–163, April 2003.

[6] Christophe Berger, Thierry Géraud, Roland Levillain, and Nicolas Widynski. Effective component tree computation with application to pattern recognition in astronomical imaging. In *IEEE International Conference on Image Processing, San Antonio, Texas, USA*, pages 16–19, September 2007.

[7] Serge Beucher. Transformations résiduelles en morphologie numérique. Technical report, Centre de Morphologie Mathématique / École des mines de Paris, Décembre 2003.

[8] Serge Beucher. Numerical residues. In C. Ronse, L. Najman, and E. Decencière, editors, *Mathematical Morphology : 40 Years On*, volume 30 of *Computational Imaging and Vision*, pages 23–32. Springer-Verlag, Dordrecht, 2005.

[9] M.C. Burl, M. Weber, and P. Perona. A probabilistic approach to object recognition using local photometry and global geometry. *Lecture Notes in Computer Science*, 1407 :628–641, 1998.

[10] H. Byun, I. Jang, and Y. Choi. Text extraction in digital news video using morphology. In *DAS'02 : Proceedings of the 5th International Workshop on Document Analysis Systems V*, pages 341–352, London, UK, 2002. Springer-Verlag.

[11] D. Chen. *Text detection and recognition in images and video sequences*. PhD thesis, École Polytechnique Fédérale de Lausanne, Aug. 2003.

[12] D. Chen and J. Luettin. A survey of text detection and recognition in images and videos. IDIAP-RR-00 38, IDIAP, 2000.

[13] D. Chen, K. Shearer, and H. Bourlard. Text enhancement with asymmetric filter for video OCR. In *International Conference on Image Analysis and Processing*, pages 192–197, 2001.

[14] D. Chen, J.M. Odobez, and H. Bourlard. Text detection and recognition in images and video frames. *Pattern Recognition*, 37(3) :595–608, March 2004.

[15] D. Chen, J.M. Odobez, and J.P. Thiran. A localization/verification scheme for finding text in images and video frames based on contrast independent features and machine learning methods. *Signal Processing : Image Communication*, 19(3) :205–217, March 2004.

[16] X. Chen, J. Yang, J. Zhang, and A. Waibel. Automatic detection and recognition of signs from natural scenes. *IEEE Transactions on Image Processing*, 13(1) :87–99, January 2004.

[17] Xiangrong Chen and Alan L. Yuille. A time-efficient cascade for real-time object detection : With applications for the visually impaired. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops*, page 28, Washington, DC, USA, 2005. IEEE Computer Society.

[18] M. CHEW, H.S. BAIRD, K. Tapas, S.E.H. Barney, J. Hu, and P.B. Kantor. Baffletext : A human interactive proof. In *Proceedings of SPIE-IS&T Electronic Imaging, Document Recognition and Retrieval, Conference No10*, volume 5010, pages 305–316, Santa Clara CA , USA, january 2003.

[19] P. Clark and M. Mirmehdi. Recognising text in real scenes. *International Journal on Document Analysis and Recognition*, 4(4) :243–257, August 2002. ISSN 1433-2833.

[20] P. Clark and M. Mirmehdi. Rectifying perspective views of text in 3d scenes using vanishing points. *Pattern Recognition*, 36(11) :2673–2686, November 2003.

[21] D. Crandall, S. Antani, and R. Kasturi. Extraction of special effects caption text events from digital video. *International Journal of Document Analysis and Recognition (IJDAR)*, 5(2-3) : 138–157, April 2003.

[22] C.H. Demarty. *Segmentation et structuration d'un document vidéo pour la caractérisation et l'indexation de son contenu sémantique : Application aux journaux télévisés*. Thèse de doctorat en morphologie mathématique, École Nationale Supérieure des Mines de Paris, 2000.

[23] N. Dimitrova, L. Agnihotri, C. Dorai, and R.M. Bolle. Mpeg-7 videotext description scheme for superimposed text in images and video. *Signal Processing :Image Communication*, 16(1-2) : 137–155, September 2000.

[24] B. Efron and R. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall, New York, 1993.

[25] D.A. Forsyth and M.M. Fleck. Identifying nude pictures. In *3rd IEEE Workshop on Applications of Computer Vision*, Sarasota, Florida, December 1996.

[26] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, 2 edition, 1990.

[27] Y.M.Y. Hasan and L.J. Karam. Morphological text extraction from images. *IEEE Trans. Image Processing*, 9(11) :1978–1983, November 2000.

[28] H. Hase, T. Shinokawa, M. Yoneda, and C.Y. Suen. Character string extraction from color documents. *Pattern Recognition*, 34(7) :1349–1365, July 2001.

[29] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning : Data mining, inference, and prediction*, chapter 4. Series in Statistics. Springer, New York, 1st ed. 2001. corr. 3rd printing edition, 2003.

[30] X.S. Hua, X.R. Chen, L. Wenyin, and H.J. Zhang. Automatic location of text in video frames. In *MULTIMEDIA'01 : Proceedings of the 2001 ACM workshops on Multimedia*, pages 24–27, New York, NY, USA, 2001. ACM Press.

[31] X.S. HUA, L. Wenyin, and H.J. ZHANG. An automatic performance evaluation protocol for video text detection algorithms. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(4) :498–507, April 2004.

[32] L. Huiping. *Automated Processing and Analysis of Text in Digital Video*. PhD thesis, Language and Media Processing Laboratory : University of Maryland, 2000.

[33] A.K. Jain and B. Yu. Automatic text location in images and video frames. In *Pattern Recognition, 1998. Proceedings. Fourteenth International Conference on*, volume 2, pages 1497–1499, Brisbane, Qld., Australia, August 1998.

[34] Edmond J.Breen and Ronald Jones. Attribute openings, thinnings and granulometries. P.MARAGOS, R.SCHAFER, M.BUTT *: Mathematical Morphology and its applications to image and signal processing*, 64(3) :377–389, Nov 1996.

[35] K. Jung. Neural network-based text location in color images. *Pattern Recognition Letters*, 22 (14) :1503–1515, December 2001.

[36] K. Jung, K.I. Kim, and A.K. Jain. Text information extraction in images and video : a survey. *Pattern Recognition*, 37(5) :977–997, May 2004.

[37] K. Karu, A.K. Jain, and R.M. Bolle. Is there any texture in the image ? *Pattern Recognition*, 29(9) :1437–1446, September 1996.

[38] H.K. Kim. Efficient automatic text location method and content-based indexing and structuring of video database. *Journal of Visual Communication and Image Representation*, 7(4) :336–344, December 1996.

[39] K.I. Kim, K. Jung, and J.H. Kim. Texture-based approach for text detection in images using support vector machines and continuously adaptive mean-shift algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(12) :1631–1639, December 2003.

[40] F. Lebourgeois. Robust multifont OCR system from gray level images. In *4th International Conference Document Analysis and Recognition (ICDAR'97), 2-Volume Set, August 18-20, 1997, Ulm, Germany*, pages 1–5, 1997.

[41] C.W Lee, K. Jung, and H.J. Kim. Automatic text detection and removal in video sequences. *Pattern Recognition Letters*, 24(15) :2607–2623, November 2003.

[42] N.J. Leite and S.J.F. Guimaraes. Morphological residues and a general framework for image filtering and segmentation. *Journal on Applied Signal Processing*, 2001(4) :219–229, December 2001.

[43] H. Li and D. Doermann. Superresolution-based enhancement of text in digital video. In *International Conference on Pattern Recognition*, volume 1, pages 847–850, Los Alamitos, CA, USA, 2000. IEEE Computer Society.

[44] H. Li and D. Doermann. Text enhancement in digital video using multiple frame integration. In *MULTIMEDIA'99 : Proceedings of the seventh ACM international conference on Multimedia (Part 1)*, pages 19–22, New York, NY, USA, 1999. ACM Press.

[45] H. Li, D. Doermann, and O. Kia. Automatic text detection and tracking in digital video. *IEEE Transactions on Image Processing - Special Issue on Image and Video Processing for Digital Libraries*, 9(1) :147–156, January 2000.

[46] J. Li and R. Gray. Context based multiscale classification of images. In *IEEE International Conference on Image Processing*, volume 3, pages 566–570, October 1998.

[47] J. Liang, D. Doermann, and H. Li. Camera-based analysis of text and documents : a survey. *International Journal on Document Analysis and Recognition*, 7(2-3) :84–104, July 2005.

[48] R. Lienhart. Video OCR : A survey and practitioner's guide. In *VideoMining, Chapter 6, A. Rosenfeld and D. Doermann and D. DeMenthon*. Kluwer Academic Publishers, 2003.

[49] R. Lienhart. Automatic text recognition for video indexing. In *MULTIMEDIA'96 : Proceedings of the 4th ACM international conference on Multimedia*, pages 11–20, New York, NY, USA, 1996. ACM Press.

[50] R. Lienhart and W. Effelsberg. Automatic text segmentation and text recognition for video indexing. *Multimedia Syst.*, 8(1) :69–81, 2000.

[51] R. Lienhart and A. Wernicke. Localizing and segmenting text in images and videos. *IEEE Transactions on Circuits and Systems for Video Technology*, 12(4) :256–268, April 2002.

[52] S.M. Lucas. Text locating competition results. In *International Conference on Document Analysis and Recognition*, volume 1, pages 80–85, Seoul, Korea, 2005.

[53] S.M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, R. Young, K. Ashida, H. Nagai, M. Okamoto, H. Yamamoto, H. Miyao, J. Zhu, W. Ou, C. Wolf, J.M. Jolion, L. Todoran, M. Worring, and X. Lin. Icdar 2003 robust reading competitions :entries, results, and future directions. *International Journal on Document Analysis and Recognition*, 7(2-3) :105–122, July 2005.

[54] M.R. Lyu, J. Song, and M. Cai. A comprehensive method for multilingual video text detection, localization, and extraction. *IEEE Transactions Circuits and Systems for Video Technology*, 15 (2) :243–255, February 2005.

[55] V. Mariano. *Video Object Detection and Matching*. PhD thesis, Pennsylvania State University, Juin 2003.

[56] V.Y. Mariano and R. Kasturi. Locating uniform-colored text in video frames. In *International Conference on Pattern Recognition*, volume 4, pages 539–542, 2000.

[57] A. Meijster and M.H.F. Wilkinson. A comparison of algorithms for connected set openings and closings. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4) :484–494, April 2002.

[58] S. Messelodi and C.M. Modena. Automatic identification and skew estimation of text lines in real scene images. *Pattern Recognition*, 32(5) :791–810, May 1999.

[59] Fernand Meyer. Un algorithme optimal de ligne de partage des eaux. pages 847–857, Lyon,France, November 1991.

[60] Fernand Meyer and Serge Beucher. Morphological segmentation. *Journal of Visual Communication and Image Representation*, 1(1) :21–46, 1990. EX N-11/90/MM.

[61] G. Meyers, R. Bolles, Q.T. Luong, and J. Herson. Recognition of text in 3d scenes. In *4th Symposium on Document Image Understanding Technology, Columbia, Maryland*, April 2001.

[62] G. Nagy, T.A. Nartker, and S.V. Rice. Optical character recognition : an illustrated guide to the frontier. In *Document Recognition and Retrieval VII, SPIE Vol. 3967, San Jose*, pages 58–69, January 2000.

[63] L. Najman and M. Couprie. Building the component tree in quasi-linear time. *IEEE Transactions on Image Processing*, 15(11) :3531–3539, November 2006.

[64] W. Niblack. *An introduction to digital image processing*, pages 113–116. Strandberg Publishing Company, Birkeroed, Denmark, 1985.

[65] J. Ohya, A. Shio, and S. Akamatsu. Recognizing characters in scene images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16(2) :214–220, 1994.

[66] N. Otsu. A threshold selection method from grey-level histograms. *IEEE Trans. Systems, Man and Cybernetics*, 9(1) :62–66, January 1979.

[67] Souhaïl OUTAL. *Quantification par analyse d'images de la granulométrie des roches fragmentées : amélioration de l'extraction morphologique des surfaces,amélioration de la reconstruction stéréologique*. Thèse de doctorat en morphologie mathématique et géosciences, ENSMP, 2006.

[68] H.C. Park, S.Y. Ok, Y.J. Yu, and H.G. Cho. A word extraction algorithm for machine-printed documents using a 3d neighborhood graph model. *International Journal of Document Analysis and Recognition (IJDAR)*, 4(2) :115–130, 2001.

[69] T. Pun. A new method for grey-level picture thresholding using the entropy of the histogram. *Signal Processing*, 2 :223–237, July 1980.

[70] T. Retornaz and Marcotegui B. Workshop imageval : Scene-text localization based on ultimate opening.application on imageval database campaign., July 2007. URL `http://www.imageval.org/Workshop/ARMINES_CMM_ImagEVAL06_p.pdf`.

[71] T. Retornaz and Marcotegui B. Scene-text localization based on ultimate opening. In *International Symposium on Mathematical Morphology ISMM'07.*, Rio de Janeiro. Brasil, October 2007.

[72] T. Retornaz and Marcotegui B. Ultimate opening implementation based on a flooding process. In *ICS XII, The 12th International Congress for Stereology*, Saint-Etienne, France, September 2007.

[73] S.V. Rice, F.R. Jenkins, and T.A. Nartker. The 5th annual test of OCR accuracy. Technical Report 96-01, Information Science Research Institute, University of Nevada, Las Vegas, April 1996.

[74] V. Risson. *Application de la Morphologie Mathématique à l'analyse des conditions d'éclairage des images couleur*. Thèse de doctorat en morphologie mathématique, École Nationale Supérieure des Mines de Paris, December 2001.

[75] Jean-Francois Rivest, Pierre Soille, and Serge Beucher. Morphological gradients. *Journal of Electronic Imaging*, 2(4) :326–336, 1993.

[76] Roerdink and Meijster. The watershed transform : Definitions, algorithms and parallelization strategies. *FUNDINF : Fundamenta Informatica*, 41, 2000.

[77] Y. Rui, T. Huang, and S. Chang. Image retrieval : current techniques, promising directions and open issues. *Journal of Visual Communication and Image Representation*, 10(4) :39–62, April 1999.

[78] P. Salembier, A. Oliveras, and L. Garrido. Anti-extensive connected operators for image and sequence processing. *IEEE Transactions on Image Processing*, 7(4) :555–570, 1998.

[79] G. Salton. *Automatic text processing : The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley Longman Publishing Co., Inc., 1988.

[80] T. Sato, T. Kanade, E.K. Hughes, M.A. Smith, and S.I. Satoh. Video OCR : indexing digital new libraries by recognition of superimposed captions. *Multimedia Systems*, 7(5) :385–395, 1999.

[81] Jean Serra. *Image analysis and mathematical morphology*, volume 1. Academic Press, 1982.

[82] Jean Serra. *Image Analysis and Mathematical Morphology - Theoretical Advances*, volume 2. Academic Press, 1988.

[83] Mehmet Sezgin and Bulent Sankur. Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging*, 13(1) :146–168, 2004.

[84] J.C. Shim, C. Dorai, and R.M. Bolle. Automatic text extraction from video for content-based annotation and retrieval. In *14th International Conference on Pattern Recognition*, volume 1, pages 618–620, 1998.

[85] B.K. Sin, S.K. Kim, and B.J. Cho. Locating characters in scene images using frequency features. In *16th International Conference on Pattern Recognition (ICPR'02) - Volume 3*, pages 489–492, Quebec, Canada, 2002.

[86] A.W.M. Smeulders, M.L. Kersten, and T. Gevers. Crossing the divide between computer vision and databases in search of image databases. In *Proc. 4th Working Conf. Visual Database Systems*, pages 223–239, 1998.

[87] A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12) :1349–1380, December 2000. ISSN 0162-8828.

[88] M. Smith and T. Kanade. Video skimming for quick browsing based on audio and image characterization. Technical Report CMU-CS-95-186, Computer Science Department, Carnegie Mellon University, Pittsburgh, PA, July 1995.

[89] S.M. Smith and J. M. Brady. SUSAN – A new approach to low level image processing. Technical Report TR95SMS1c, Chertsey, Surrey, UK, 1995.

[90] Pierre Soille. *Morphological Image Analysis : Principles and Applications*. Springer-Verlag Berlin, Heidelberg, New York, 1999.

[91] A.L. Spitz. Determination of the script and language content of document images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(3) :235–245, March 1997.

[92] M. Szummer and R.W. Picard. Indoor-outdoor image classification. In *IEEE Intl. Workshop on Content-Based Access of Image and Video Databases, CAIVD*, pages 42–51, Bombay, India, 1998.

[93] X. Tang, X. Gao, J. Liu, and H. Zhang. A spatial- temporal approach for video caption detection and recognition. *IEEE Transactions on Neural Networks*, 13(4) :961–971, April 2002.

[94] Robert Endre Tarjan. Efficiency of a good but not linear set union algorithm. *J. ACM*, 22(2) : 215–225, 1975.

[95] O.D. Trier and A.K. Jain. Goal-directed evaluation of binarization methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(12) :1191–1201, 1995.

[96] O.D. Trier, A.K. Jain, and T. Taxt. Feature-extraction methods for character-recognition : A survey. *Pattern Recognition*, 29(4) :641–662, April 1996.

[97] Erik R. Urbach, Niek J. Boersma, and Michael H.F. Wilkinson. Vector-attribute filters. In C. Ronse, L. Najman, and E. Decencière, editors, *Mathematical Morphology : 40 Years On*, volume 30 of *Computational Imaging and Vision*, pages 95–104. Springer-Verlag, Dordrecht, 2005.

[98] C. Vachier. *Extraction de caractéristiques, Segmentation d'Image et Morphologie Mathématique*. Thèse de doctorat en morphologie mathématique, École Nationale Supérieure des Mines de Paris, December 1995.

[99] A. Vailaya, A. Jain, and H.J. Zhang. On image classification : city images vs. landscapes. *Pattern Recognition*, 31(12) :1921–1935, December 1998.

[100] L. Vincent. Efficient computation of various types of skeletons. In M.H. Loew, editor, *Proc. SPIE, Medical Imaging V : Image Processing*, volume 1445, pages 297–311, San Jose, CA, February 1991.

[101] L. Vincent and P. Soille. Watersheds in digital spaces : An efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6) : 583–598, June 1991.

[102] Luc Vincent. Morphological area openings and closings for grayscale images. In *NATO :Shape in picture workshop, Driebergen*, pages 197–208. Springer-Verlag, Sep 1992.

[103] Luc Vincent. Morphological area openings and closings, their efficient implementation and applications. JEAN SERRA, P.SALEMBIER EDITORS *: Mathematical Morphology and its applications to signal processing*, pages 22–27, May 1993.

[104] R.A. Wagner and M.J. Fischer. The string-to-string correction problem. *Journal of the Association for Computing Machinery*, 21(1) :168–173, 1974.

[105] Thomas Walter. *Application de la Morphologie Mathématique au diagnostic de la Rétinopathie Diabétique à partir d'images couleur*. Thèse de doctorat en morphologie mathématique, École Nationale Supérieure des Mines de Paris, September 2003.

[106] J.Z. Wang, J. Li, G. Wiederhold, and O. Firschein. System for classifying objectionable websites. In *Interactive Distributed Multimedia Systems and Telecommunication Services : 5th International Workshop,Oslo, Norway*, volume 1483, pages 113–124, September 1998.

[107] J.Z. Wang, J. Li, and G. Wiederhold. Simplicity : Semantics-sensitive integrated matching for picture libraries. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(9) : 947–963, 2001. ISSN 0162-8828.

[108] K. Wang and J.A. Kangas. Character location in scene images from digital camera. *Pattern Recognition*, 36(10) :2287–2299, October 2003.

[109] M.H.F. Wilkinson and J. Roerdink. Fast morphological attribute operations using Tarjan 's union-find algorithm. In *Mathematical Morphology and its Applications to Image and Signal Processing, Kluwer*, pages 311–320, 2000.

[110] C. Wolf. *Text Detection in Images taken from Videos Sequences for Semantic Indexing*. PhD thesis, INSA de Lyon, December 2003.

[111] C. Wolf and J.M. Jolion. Object count/area graphs for the evaluation of object detection and segmentation algorithms. *International Journal of Document Analysis and Recognition (IJDAR)*, 8(4) :280–296, 2006.

[112] C. Wolf and J.M. Jolion. Extraction and recognition of artificial text in multimedia documents. *Pattern Analysis and Applications*, 6(4) :309–326, 2004.

[113] E.K. Wong and M. Chen. A robust algorithm for text extraction in color video. In *IEEE International Conference on Multimedia and Expo (II)*, pages 797–799, 2000.

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

[115] V. Wu, R. Manmatha, and E.M. Riseman. Finding text in images. In *DL'97 : Proceedings of the second ACM international conference on Digital libraries*, pages 3–12, New York, NY, USA, 1997. ACM Press.

[116] V. Wu, R. Manmatha, and E.M. Riseman. Textfinder : An automatic system to detect and recognize text in images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21 (11) :1224–1229, 1999.

[117] W. Wu, X. Chen, and J. Yang. Detection of text on road signs from video. *IEEE Transactions on Intelligent Transportation Systems*, 6(4) :378–390, December 2005.

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

[119] D. Zhang, B. Tseng, and S.F. Chang. Accurate overlay text extraction for digital video analysis. In *International Conference on Information Technology : Research and Education*, pages 233–237, August 2003.

[120] D.Q. Zhang and S.F. Chang. Learning to detect scene text using a higher-order MRF with belief propagation. In *IEEE Workshop on Learning in Computer Vision and Pattern Recognition, in conjunction with CVPR (LCVPR),Washington DC*, page 52, June 2004.

[121] Yefeng Zheng, Huiping Li, and David Doermann. Machine printed text and handwriting identification in noisy document images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(3) :337–353, 2004.

[122] Y. Zhong, K. Karu, and A.K. Jain. Locating text in complex color images. *Pattern Recognition*, 28(10) :1523–1535, 1995.

[123] Y. Zhong, H.J. Zhang, and A.K. Jain. Automatic caption localization in compressed video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(4) :385–392, April 2000.

[124] J. Zhou and D.P. Lopresti. Locating and recognizing text in WWW images. *Information Retrieval*, 2(2/3) :177–206, 2000.

[125] J. Zhou, D.P. Lopresti, and T. Tasdizen. Finding text in color images. In *Document Recognition V SPIE , San Jose*, pages Vol 3305 : 130–140, 1998.