# Apports de nouveaux outils de traitement d'images et de programmation pour le relevé automatique de dégradations sur chaussées

Wissam Kaddah

▶ **To cite this version:**

Wissam Kaddah. Apports de nouveaux outils de traitement d'images et de programmation pour le relevé automatique de dégradations sur chaussées. Traitement du signal et de l'image [eess.SP]. Université de Bretagne occidentale - Brest, 2018. Français. NNT : 2018BRES0102 . tel-02130479

**HAL Id: tel-02130479**

**https://theses.hal.science/tel-02130479**

Submitted on 15 May 2019

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNIVERSITE BRETAGNE LOIRE / MATHSTIC

UBO Université de Bretagne Occidentale

# THESE DE DOCTORAT DE

L'UNIVERSITE
DE BRETAGNE OCCIDENTALE
COMUE UNIVERSITE BRETAGNE LOIRE

ECOLE DOCTORALE N° 601
*Mathématiques et Sciences et Technologies de l'Information et de la Communication*
Spécialité : *Signal, Image, Vision*

Par
**Wissam KADDAH**

## Apports de nouveaux outils de traitement d'images et de programmation pour le relevé automatique de dégradations sur chaussées

**Thèse présentée et soutenue à Yncréa-Ouest (ISEN-Brest), le 19/12/2018**

**Unité de recherche : Département VISION / L@bISEN à Yncréa-Ouest et la société ACTRIS**

**Rapporteurs avant soutenance :**

**Denis HAMAD**, Professeur, Université de Littoral Côte d'Opale

**Naveen Kumar NISHCHAL**, Associate professor, Indian Institute of Technology Patna

**Composition du Jury :**

**Christian BROSSEAU**, Professeur, Université de Bretagne Occidentale / *Président*

**Denis HAMAD**, Professeur, Université de Littoral Côte d'Opale / *Rapporteur*

**Naveen Kumar NISHCHAL**, Associate professor, Indian Institute of Technology Patna / *Rapporteur*

**Fethi BEN OUEZDOU**, Professeur, Directeur scientifique de Vedecom / *Examinateur*

**Ayman ALFALOU**, Professeur, Yncréa-Ouest / *Directeur de thèse*

**Marwa ELBOUZ**, Enseignante-chercheuse, Yncréa-Ouest / *Encadrante de thèse*

**Invités :**

**Yousri OUERHANI**, Docteur, ACTRIS / *Encadrant de thèse*

**Vincent BALTAZART**, Chargé de recherche, IFSTTAR / *Co-encadrant de thèse*

# Remerciements

C'est avec une certaine émotion et beaucoup de sincérité que je voudrais remercier toutes les personnes qui m'ont aidées pendant l'élaboration de ma thèse Cifre.

En premier lieu, j'aimerais remercier vivement mon directeur de thèse, M. Ayman Alfalou, professeur, directeur général de la recherche et directeur du laboratoire VISION à Yncréa-Ouest (ISEN-Brest), pour la confiance qu'il m'a accordée. Je lui remercie également pour son intérêt, ses nombreux conseils, ses qualités pédagogiques et sa sympathie qui m'ont apportés le goût pour les sciences. J'ai beaucoup appris à ses côtés et je lui adresse ma gratitude pour tout cela.

J'adresse de chaleureux remerciements à mes encadrants de thèse, Mme Marwa Elbouz, enseignante-chercheuse à Yncréa-Ouest (ISEN-Brest) et M. Yousri Ouerhani, docteur en traitement du signal et d'images chez la société ACTRIS à Brest, pour leurs remarques précieuses, leurs attentions de tout instant sur mes travaux, leurs conseils avisés et leurs écoutes qui ont été prépondérants pour la bonne réussite de cette thèse. Ils m'ont apporté et m'apportent encore beaucoup, à la fois humainement et professionnellement.

Merci à mon co-encadrant de thèse, M. Vincent Baltazart, docteur en traitement du signal à l'IFSTTAR de Nantes, pour m'avoir suivi toute la durée du mon travail et m'avoir accueilli au sein du laboratoire SII à l'IFSTTAR. Je lui remercie pour le temps qui m'a consacré surtout pendant les réunions téléphoniques où nous avons échangé des idées intéressantes ce qui a enrichi ce travail.

Je tiens aussi à témoigner toute ma reconnaissance aux professeurs Naveen Kumar Nishchal et Denis Hamad pour l'honneur qu'ils m'ont fait en acceptant d'être rapporteurs de ce travail de recherche et membres du jury.

Je souhaite également remercier les professeurs Christian Brosseau et Fethi Ben Ouezdou qui ont accepté d'associer leurs expertises respectives à l'évaluation de cette thèse.

Un grand merci à M. Philippe Raillat, directeur général de la société DIAGWAY, pour son implication dans le projet, mais aussi pour sa participation financière dans cette étude.

Je remercie tout particulièrement M. Marc Desthieux, co-dirigeant de la société ACTRIS pour son implication dans le projet, son soutien financier, sa disponibilité malgré la surcharge de son planning, sa bienveillance et son extrême gentillesse.

Je remercie également l'ensemble de personnels de Yncréa-Ouest dirigé par M. Marc Faudeil et particulièrement l'équipe VISION de m'avoir accueilli chaleureusement au sein de leur laboratoire. Plus précisément, je remercie mes deux camarades, docteur Djamel-Eddine Benarab et docteur Angel Bleunven, pour le temps fou que nous avons passé ensemble, notamment pendant les séances de *push-up* et de *ping-pong*. Nous avons partagé d'agréables moments et j'en garderai de très bons souvenirs.

Je pense aussi à tous mes amis doctorants en première et deuxième année, avec qui j'ai partagé amicalement une très belle expérience.

Au terme de ce parcours, j'associe à ces remerciements celles et ceux qui me sont chers et que j'ai quelques peu délaissés ces derniers mois pour achever ma thèse. Leur présence inconditionnelle dans ma vie, leurs attentions et encouragements m'ont accompagnée tout au long de ces années.

Finalement, mes plus profonds remerciements vont à ma famille qui a toujours été présente à côté de moi, en particulier mon père, ma mère et ma sœur, dont leur amour, leur soutien moral et leur confiance indéfectible dans mes choix, m'ont été d'un grand réconfort et ont contribué à l'aboutissement de ce travail.

# Table des matières

**Liste des figures** **xiii**

**Liste des tableaux** **xxiii**

**Introduction générale** **1**

**1 : État de l'art** **5**

1.1 Introduction . . . . . 7

1.2 Problématiques . . . . . 7

1.3 Dégradations de surface de chaussées . . . . . 8

1.3.1 Généralités . . . . . 8

1.3.2 Types de dégradations routières . . . . . 8

1.3.3 Méthodologie pour le relevé de dégradations . . . . . 10

1.3.3.1 Relevé visuel in situ . . . . . 11

1.3.3.2 Relevé visuel sur écran et systèmes imageurs à grand rendement . . . . . 11

1.3.3.3 Relevé automatique . . . . . 12

1.4 Méthodes de détection des marquages routiers . . . . . 13

1.4.1 Caractérisation optiques des marquages routiers . . . . . 13

1.4.2 Spécification des différents types de marquages . . . . . 13

1.4.3 Propriétés optiques des marquages routiers . . . . . 14

1.4.4 Approches existantes pour la détection du marquage au sol . . . 16

1.4.4.1 Méthodes de détection de lignes . . . . . 16

1.4.4.2 Méthodes orientées contours . . . . . 19
1.4.4.3 Méthodes orientées régions . . . . . 21
1.4.5 Outils de pré-traitement dédiés à l'extraction du marquage au sol 22
1.4.5.1 Transformée perspective inverse (IPM) . . . . . 22
1.4.5.2 Morphologie mathématique . . . . . 25
1.4.5.3 Méthodes de seuillage . . . . . 26
1.4.6 Extracteurs du marquage au sol basés sur la forme . . . . . 27
1.4.6.1 Opérateur par variation de luminosité . . . . . 27
1.4.6.2 Opérateur par recherche des gradients symétriques . . 28
1.4.6.3 Opérateur par patron . . . . . 29
1.5 Méthodes de détection de fissures . . . . . 30
1.5.1 Hypothèses . . . . . 30
1.5.2 Synoptique de la détection des fissures . . . . . 31
1.5.3 Méthodes de segmentation de fissures existantes . . . . . 32
1.5.3.1 Approches par seuillage . . . . . 32
1.5.3.2 Approches par morphologie mathématique . . . . . 34
1.5.3.3 Approches par apprentissage . . . . . 35
1.5.3.4 Approche par filtre adaptatif . . . . . 35
1.5.3.5 Approche par transformation en ondelettes continue . 36
1.5.3.6 Approche par modélisation markovienne . . . . . 36
1.5.3.7 Approche heuristique . . . . . 37
1.5.3.8 Approches par chemins minimaux . . . . . 38
1.5.3.9 Discussion . . . . . 41
1.6 Systèmes d'imagerie utilisés . . . . . 42
1.6.1 Systèmes spécialisés d'imagerie de la route . . . . . 42
1.6.1.1 Appareil Aigle-RN . . . . . 42
1.6.1.2 Système LCMS . . . . . 43

1.6.2 Système d'imagerie $VIAPIX^{®}$ . . . . . 44
1.7 Conclusion . . . . . 45

**2 : Extraction des dégradations de la signalisation horizontale 48**

2.1 Introduction . . . . . 49
2.2 Approche proposée . . . . . 50
2.2.1 Détection de la signalisation horizontale . . . . . 50
2.2.1.1 Solution proposée . . . . . 52
2.2.1.2 Détection de la route . . . . . 52
2.2.1.3 Segmentation couleur . . . . . 66
2.2.2 Identification des marquages détectés . . . . . 68
2.2.2.1 Méthode de reconnaissance géométrique . . . . . 68
2.2.2.2 Méthode de reconnaissance optique . . . . . 71
2.2.3 Géolocalisation du marquage identifié . . . . . 75
2.2.4 Quantification de l'usure du marquage routier . . . . . 76
2.3 Évaluation des performances de la méthode sur des images $VIAPIX^{®}$ 78
2.4 Conclusion . . . . . 82

**3 : Détection de fissures par sélection de chemins minimaux 84**

3.1 Introduction . . . . . 86
3.2 Synoptique de la méthode MPS . . . . . 86
3.2.1 Étape 1 : Sélection automatique de points amorces . . . . . 87
3.2.2 Étape 2 : Calcul des plus courts chemins entre paires de points amorces . . . . . 89
3.2.2.1 Stratégie de balayage de l'image . . . . . 89
3.2.2.2 Algorithme de calcul du plus court chemin . . . . . 90
3.2.2.3 Illustration . . . . . 90
3.2.3 Étape 3 : Sélection automatique des chemins de coût minimal . 91

3.2.4 Étape 4 : Raffinage de la segmentation du squelette de fissuration (Post-traitement) . . . 93
3.2.4.1 Principe de l'approche basée sur l'analyse de la topologie du squelette . . . 93
3.2.4.2 Suppression des objets de petite taille . . . 94
3.2.4.3 Suppression des ressemblances et des parties communes entre les chemins adjacents . . . 95
3.2.4.4 Mise à jour des points caractéristiques du squelette . . . 96
3.2.4.5 Mise à jour des chemins . . . 98
3.2.4.6 Sélection et filtrage des ramifications superflues . . . 99
3.2.4.7 Sélection et filtrage des boucles superflues . . . 101
3.2.4.8 Segmentation finale du squelette de fissuration . . . 103
3.2.5 Étape 5 : Estimation de l'épaisseur locale de la fissure . . . 105
3.3 Méthodologie d'évaluation des performances de MPS . . . 106
3.3.1 Pseudo-vérité terrain (PVT) . . . 106
3.3.2 Critères d'évaluation . . . 107
3.4 Illustration des performances de la méthode MPS . . . 110
3.5 Conclusion . . . 119

**4 : Optimisations de la sélection des chemins minimaux 122**

4.1 Introduction . . . 124
4.2 Améliorations proposées de la méthode MPS . . . 124
4.2.1 Analyse fonctionnelle . . . 124
4.2.2 Amélioration de l'étape 1 : Réduction des points amorces . . . 125
4.2.2.1 Modélisation de l'image et la fissure . . . 125
4.2.2.2 Sélection des points amorces par analyse d'anisotropie locale : MPS-V1 . . . 131
4.2.2.3 Sélection des points amorces par fusion des résultats : MPS-V2 . . . 135

4.2.3 Amélioration de l'étape 2 : Calcul optimisé des plus courts chemins entre paires de points amorces . . . 139
4.2.4 Amélioration de l'étape 3 : Mise à jour du paramétrage . . . 141
4.2.4.1 Répartition des valeurs de coûts des nouveaux chemins calculés . . . 142
4.2.4.2 Analyse de la sensibilité . . . 143
4.2.5 Post-traitement . . . 144
4.3 Étude de cas sur des images de chaussée . . . 145
4.3.1 Base de données testées . . . 145
4.3.2 Tests et validation . . . 147
4.3.3 Résultats statistiques . . . 150
4.4 Conclusion . . . 151

**5 : Détection de fissures par sélection de filaments sombres 154**

5.1 Introduction . . . 155
5.2 Approche proposée . . . 155
5.3 Synoptique générale . . . 156
5.3.1 Pré-traitement . . . 156
5.3.1.1 Nettoyage de l'image originale . . . 157
5.3.1.2 Sélection des points amorces . . . 161
5.3.2 Traitement . . . 169
5.3.2.1 Calcul des filaments sombres . . . 170
5.3.2.2 Sélection des filaments de coût minimal . . . 170
5.3.3 Post-traitement . . . 171
5.3.3.1 Raffinage du squelette de fissuration . . . 171
5.3.3.2 Estimation de l'épaisseur de fissure . . . 186
5.4 Tests et validation . . . 186
5.4.1 Tests sur des images Aigle-RN . . . 187
5.4.2 Tests sur des images $VIAPIX^{®}$ . . . 191

5.4.3 Tests sur des images LCMS "élévation" . . . . . 194
5.5 Conclusion . . . . . 197

**6 : Conclusion et perspectives** **199**
6.1 Conclusion . . . . . 200
6.2 Perspectives . . . . . 201

**Bibliographie** **205**

**Annexes** **220**

**A : Opérations élémentaires de la morphologie mathématique** **221**
A.1 Opérateurs de la morphologie mathématique . . . . . 222
A.1.1 Dilatation morphologique . . . . . 222
A.1.2 Érosion morphologique . . . . . 222
A.1.3 Ouverture morphologique . . . . . 223
A.1.4 Fermeture morphologique . . . . . 224

**B : Données techniques sur le système $VIAPIX^{®}$** **226**
B.1 Système $VIAPIX^{®}$ . . . . . 227
B.1.1 $VIAPIX^{®}$ acquisition . . . . . 227
B.1.2 $VIAPIX^{®}$ exploitation . . . . . 228

**C : Algorithme de Dijkstra appliqué à la segmentation de fissures** **231**
C.1 Introduction : les problématiques du plus court chemin dans le cadre de la thèse . . . . . 232
C.2 Utilisation de l'algorithme de Dijkstra dans le cadre de la thèse . . . . 232
C.3 Illustration du fonctionnement de l'algorithme de Dijkstra . . . . . 233
C.3.1 Cas de la problématique SPSP . . . . . 233
C.3.2 Cas de la problématique SSSP . . . . . 238

**Publications scientifiques de l'auteur** **240**

Articles de journaux . . . . . 240

Articles de conférences internationales . . . . . 240

**Résumé/Abstract** **243**

# Liste des figures

**1 : État de l'art**


1.1 Structure et couches de la chaussée . . . . . 8

1.2 Exemple d'images illustrant l'effacement de plusieurs types de marquages au sol . . . . . 10

1.3 Exemple d'images de chaussées illustrant plusieurs types de fissures . . 10

1.4 Illustration d'autres types de marquages au sol . . . . . 14

1.5 Zones de chromaticité pour les marquages jaunes et blancs en Europe . 15

1.6 Application de la technique de thinning sur une image binaire . . . . . 16

1.7 Principe de la détection de lignes en utilisant la transformée de Hough 18

1.8 Ajustement d'une ligne située dans un plan $2D$ à une série d'observations en appliquant l'algorithme RANSAC . . . . . 18

1.9 Application projective . . . . . 22

1.10 Représentation géométrique des repères utilisés dans la transformée perspective inverse . . . . . 23

1.11 Exemple d'application de la transformée perspective inverse . . . . . 25

1.12 Exemple du calcul du résidus . . . . . 27

1.13 Exemple de la détection de contours . . . . . 28

1.14 Variation des valeurs de niveaux de gris pour un marquage au sol . . . 28

1.15 Schéma classique pour détecter les fissures sur des images de chaussées 31

1.16 Courbe représentative illustrant la fonction S de Zadeh . . . . . 33

1.17 Exemple d'une segmentation suivant une approche morphologique . . . 34

1.18 Configurations pour modéliser les relations spatiales des sites . . . . . 37

1.19 Comparaison entre un chemin minimal calculé par l'algorithme de Dijkstra (en vert) et par l'algorithme de Fast Marching (en rouge) . . . . . 39

1.20 Système Aigle-RN . . . . . 42

1.21 Système LCMS . . . . . 43

1.22 Principe de triangulation laser . . . . . 44

1.23 $VIAPIX^{®}$ acquisition . . . . . 45

**2 : Extraction des dégradations de la signalisation horizontale**

2.1 Les trois catégories des marques à traiter dans le cadre de cette thèse . 49

2.2 Synoptique général de l'approche proposée . . . . . 50

2.3 Exemple d'une image $VIAPIX^{®}$ de taille $1216 \times 1600$ pixels illustrant un passage piéton détérioré . . . . . 50

2.4 Détection des objets brillants dans une image en appliquant une technique de segmentation colorimétrique . . . . . 51

2.5 Exemples des fausses alarmes lors de l'application du seuillage adaptative pour la détection de la couleur blanche dans une image . . . . . 51

2.6 Étapes principales pour la détection de la signalisation horizontale . . . 52

2.7 Étapes réalisées pour la détection de la route . . . . . 52

2.8 Simplification des images $VIAPIX^{®}$ en appliquant la transformée perspective inverse (IPM) . . . . . 53

2.9 Choix des blocs pour détecter la route devant la voiture . . . . . 54

2.10 Sélection des blocs de taille $m \times n$ pixels (m=n=12 pixels) dans une image $VIAPIX^{®}$ après avoir fait sa transformée perspective inverse . 55

2.11 Blocs restants après avoir comparé la valeur $diff$ à la valeur du seuil $(S = 50)$ . . . . . 56

2.12 Exemple des blocs sur lesquels nous allons appliquer le filtre médian . . 57

2.13 Illustration du principe théorique du filtre médian . . . . . 57

2.14 Blocs restants après avoir appliqué le filtre médian sur les valeurs moyennes des blocs conservés dans la figure 2.11 . . . . . 58

2.15 Résultat de la détection de la route après le seuillage $S_{route}$ . . . . . 59

2.16 Illustration de quelques exemples montrant le résultat de la détection de la route sur des images perspectives inverses . . . . . 60

2.17 Coordonnées de la caméra dans chaque image perspective inverse . . . 61

2.18 L'image orthophoto obtenue après la fusion des 15 images perspectives inverses présentées dans la figure 2.17 . . . . . 62

2.19 Présentation de quelques exemples illustrant la construction des images orthophotos . . . . . 63

2.20 Résultat de la détection globale de la route . . . . . 64

2.21 Détection de la route dans chaque image après avoir désassemblé l'image orthophoto présentée dans la figure 2.20.c (troisième exemple) . . . . . 65

2.22 Détection du marquage routier dans chacune des images de la figure 2.21 67

2.23 Résultats de la détection du marquage au sol en testant d'autres images 68

2.24 Illustration de la méthode des intervalles horizontaux pour reconnaitre les bandes d'un même passage piéton . . . . . 69

2.25 Mauvaise identification d'un passage piéton oblique formé de 5 bandes . 69

2.26 Bonne identification du passage piéton oblique présenté dans la figure 2.25 70

2.27 Reconnaissance d'un passage piéton sur trois images successives . . . . 71

2.28 Principe de la corrélation optique . . . . . 72

2.29 Illustration de certaines images références utilisées pour la reconnaissance du mot BUS et du pictogramme-vélo . . . . . 73

2.30 Reconnaissance du mot BUS et du pictogramme-vélo . . . . . 74

2.31 Illustration de quelques exemples montrant la bonne identification du mot BUS et du pictogramme-vélo . . . . . 74

2.32 Géolocalisation du marquage routier (passage piéton) . . . . . 75

2.33 Visualisation du passage piéton de la figure 2.32 sur Google Maps . . . 76

2.34 Exemple d'un passage piéton identifié constitué de sept bandes juxtaposées 76

2.35 Illustration de quelques exemples montrant la bonne identification du passage piéton . . . . . 79

2.36 Exemple 1 d'un passage piéton quasiment effacé sur la chaussée . . . . 80

2.37 Exemple 2 d'un passage piéton en mauvaise état . . . . . 80

2.38 Exemple 3 d'un passage piéton ayant quelques bandes détériorées . . . 81

2.39 Exemple 4 d'un passage piéton en moyen état . . . . . 81

2.40 Exemple 5 d'un passage piéton en bon état . . . . . 81

**3 : Détection de fissures par sélection de chemins minimaux**

3.1 Synoptique de la méthode MPS . . . . . 86

3.2 (a) Image originale de taille $256 \times 256$ pixels - (b) Sélection des minimaux locaux (ROI de taille $8 \times 8$ pixels) . . . . . 87

3.3 (a) Histogramme des niveaux de gris de l'image originale et seuil $T_e$ pour $K_e$=1 - (b) Sélection des points amorces (pixels jaunes) . . . . . 88

3.4 Illustration de la stratégie utilisée pour le calcul des chemins les plus courts dans l'image . . . . . 89

3.5 (a) Points amorces sélectionnés à l'étape 1 - (b) Chemins calculés entre les points amorces par l'algorithme de Dijkstra . . . . . 91

3.6 (a) Histogramme des coûts des chemins et position du seuil $T_c$ pour $K_c = 1$ - (b) Sélection des chemins de coût minimal . . . . . 92

3.7 Synoptique du post-traitement du squelette de fissuration . . . . . 94

3.8 (a) Squelette de fissuration à l'issue de l'étape 3 de l'algorithme - (b) Squelette de fissuration après l'élimination des chemins de longueur $< T_s$ 95

3.9 Suppression des parties communes entre les chemins adjacents . . . . . 96

3.10 (a) Définition d'un pixel extrémité $e_i$ - (b) Définition d'un pixel rond point $r_i$ . . . . . 96

3.11 Mise à jour des points amorces $a_i$ en jaune (points extrémités $e_i$ en rouge et rondpoints $r_i$ en vert) . . . . . 97

3.12 Illustration du filtrage d'un point extrémité isolé . . . . . 97

3.13 Mise à jour des points amorces : points extrémités (en rouge) et points de jonctions (en vert) . . . . . 98

3.14 (a) Chemins d'une ramification - (b) Chemins d'une boucle normale - (c) Chemins d'une boucle fermée . . . . . 99

3.15 Ramifications après le seuillage $T_c$ . . . . . 100

3.16 (a) Ramifications sélectionnées en bleu (entre un point de jonction en vert et un point extrémité en rouge) dans le squelette de fissuration - (b) Ramifications restantes après le seuillage par $T_c$ . . . . . 101

3.17 (a) Boucle “normale” formée de deux chemins ayant deux $r_i$ - (b) Boucle “fermée” formée d’un seul chemin où se confondent le point source et le point destination . . . 101

3.18 (a) Boucle “normale” après le seuillage $T_c$ - (a) Boucle “fermée” après le seuillage $T_c$ . . . 102

3.19 Boucles “normales” (a) et “fermées” (b) sélectionnées dans le squelette de fissuration (en bleu) - Boucles “normales” (c) et “fermées” (d) restantes (en vert) et éliminées (en rouge) après le seuillage par $T_c$ . . . 103

3.20 Affinage du squelette de fissuration après une première itération de l’étape 4 (le squelette initial est représenté à la figure 3.8.b) . . . 104

3.21 Squelette de fissuration final après quelques itérations de l’étape 4 . . . 104

3.22 Estimation de l’épaisseur ($K_w$=0.6) . . . 105

3.23 Évaluation des performances de la méthode MPS . . . 109

3.24 Images en niveaux de gris de taille 462×991 pixels acquises par le système Aigle-RN . . . 112

3.25 Images pseudo-vérité terrain correspondantes aux images de chaussée de la figure 3.24 . . . 113

3.26 Résultats de segmentation en appliquant la méthode markovienne sur les images de la figure 3.24 . . . 114

3.27 Résultats de segmentation en appliquant la méthode Geodesic Contour (GC) sur les images de la figure 3.24 . . . 115

3.28 Résultats de segmentation en appliquant la méthode Free-Form Anisotropy (FFA) sur les images de la figure 3.24 . . . 116

3.29 Résultats de segmentation en appliquant la méthode Minimal Path Selection (MPS) sur les images de la figure 3.24 . . . 117

3.30 Détection de faux positifs (fausses alarmes) dans une image de texture sans fissures . . . 120

**4 : Optimisations de la sélection des chemins minimaux**

4.1 Exemples des graphes valués . . . 127

4.2 Voisinages directionnels . . . 128

4.3 Exemples des chemins dans des graphes déterminés par des voisinages directionnels . . . 129

4.4 Illustration des quatre chemins minimaux dans quatre orientations différentes passant par un point $l$ . . . 130

4.5 Synoptique de la méthode MPS-V1 . . . 131

4.6 Pourcentage des VP des points amorces calculés en variant les deux paramètres $d$ et $K_{FFA}$ . . . 133

4.7 Exemple de la sélection des points amorces par la méthode FFA . . . 134

4.8 Réduction du nombre des points amorces à l'étape 1 de l'algorithme en utilisant la méthode FFA . . . 134

4.9 Synoptique de l'étape 1 améliorée de l'algorithme MPS . . . 136

4.10 Résultat obtenu après le seuillage adaptatif NSCOT . . . 136

4.11 Sélection des points amorces par la méthode MPS-V2 . . . 137

4.12 Sélection des points amorces avec les trois version MPS . . . 138

4.13 Distribution des valeurs de niveaux de gris des points amorces sélectionnés par les différentes stratégies . . . 139

4.14 Redondance obtenue lors de l'application de la stratégie étoile . . . 140

4.15 Nouvelle stratégie proposée pour calculer les chemins . . . 140

4.16 Illustration des quatre chemins obtenus avec Dijkstra (SSSP) dans la fenêtre $3P \times 2P$ de la figure 3.4.b . . . 141

4.17 Calcul des chemins les plus courts après les trois différentes sélections des points amorces . . . 141

4.18 Répartition des valeurs de coûts des chemins (pdf) associée à la figure 4.17 142

4.19 Segmentation du squelette de fissuration après le seuillage appliqué sur la distribution des valeurs de coûts des chemins illustrée dans la figure 4.18 143

4.20 Illustration de la variation des coefficients de précision, sensibilité et DSC en fonction de $K_c$ pour les deux méthodes MPS-V1 et MPS-V2 . . . 144

4.21 Résultats obtenus après l'application de l'étape 4 de l'algorithme (Raffinage du squelette de fissuration) . . . 144

4.22 Résultats obtenus après l'application de l'étape 5 de l'algorithme (Estimation de l'épaisseur) . . . 145

4.23 Sous-ensemble 1 : 19 images de chaussée de taille $991 \times 462$ pixels acquises par le système Aigle-RN . . . 146

4.24 Sous-ensemble 2 : 14 images de chaussée de taille 311×462 pixels acquises par le système Aigle-RN . . . . . 147

4.25 Exemples illustrant la segmentation finale des six premières images de la figure 4.23 . . . . . 149

4.26 Valeurs moyennes de DSC obtenues lors de l'application des trois versions MPS-V0, MPS-V1 et MPS-V2 sur les 33 images illustrées dans les figures 4.23 et 4.24 . . . . . 150

4.27 Temps de calcul moyens pour traiter les 33 images illustrées dans les figures 4.23 et 4.24 selon les méthodes MPS-V0, MPS-V1 et MPS-V2 . 151

4.28 Résultats de la détection obtenus lors de l'application des trois versions de l'algorithme MPS sur une image $VIAPIX^{®}$ de taille 1216 × 1600 pixels . . . . . 152

**5 : Détection de fissures par sélection de filaments sombres**

5.1 Synoptique de la méthode ADFD . . . . . 156

5.2 Exemple d'une image en niveaux de gris illustrant la présence d'une zone fissurée et une zone ombragée sur la route . . . . . 157

5.3 Étapes principales de la phase du pré-traitement . . . . . 157

5.4 Étapes pour nettoyer l'image originale . . . . . 158

5.5 Étape de la normalisation d'une image en niveaux de gris . . . . . 159

5.6 Exemple de la présence de l'ombre dans une image originale acquise par le système $VIAPIX^{®}$ et contenant une fissure longitudinale (partie entourée en rouge) . . . . . 160

5.7 Étape de la détection de contours . . . . . 161

5.8 Étapes pour sélectionner automatiquement les points amorces . . . . . 162

5.9 Étape du seuillage adaptatif global . . . . . 162

5.10 Application de l'opération du filtrage médian . . . . . 163

5.11 Application de l'opération du filtrage selon la taille des objets . . . . . 164

5.12 Décomposition d'une image en niveaux de gris en plusieurs blocs . . . . 165

5.13 Stratégie pour décomposer une région $ROI$ de taille $P \times P$ en quatre sous-régions de taille $\frac{P}{2} \times \frac{P}{2}$ . . . . . 166

5.14 Organigramme illustrant le processus automatique de sélection d'un point amorce . . . 166
5.15 Algorithme de vérification de la concentration des pixels sombres . . . . 167
5.16 Les trois cas étudiés pour sélectionner un point amorce dans un bloc $P \times P$ 168
5.17 Sélection automatique des points amorces . . . . . 169
5.18 Étapes principales de la phase du traitement . . . . . 169
5.19 Calcul des filaments sombres . . . . . 170
5.20 Sélection des filaments les plus minces . . . . . 171
5.21 Étapes principales de la phase du post-traitement . . . . . 171
5.22 Étapes pour raffiner le squelette de fissuration . . . . . 172
5.23 Fragmentation de l'image . . . . . 173
5.24 Élimination des pixels isolés . . . . . 173
5.25 Addition des pixels sombres voisins . . . . . 174
5.26 Dilatation morphologique . . . . . 175
5.27 Analyse des objets dilatés ayant une petite taille . . . . . 176
5.28 Érosion morphologique . . . . . 177
5.29 Squelettisation morphologique . . . . . 178
5.30 Sélection des points extrémités . . . . . 179
5.31 Ajustement entre les points extrémités d'un objet ramifié . . . . . 179
5.32 Estimation de l'orientation d'un objet dans un squelette de fissuration . 180
5.33 Sélection des points début et fin des trois objets présentés dans la figure 5.32 181
5.34 Exemple d'un squelette de fissuration fragmenté et formé de deux objets proches . . . . . 181
5.35 Calcul des distances minimales entre deux objets proches et non-connectés 183
5.36 Sélection des deux points les plus proches séparant deux objets différents 184
5.37 Nouveau filament sombre obtenu l'étape de la connectivité . . . . . 185
5.38 Exemple d'application de l'étape de la connectivité . . . . . 185

5.39 Squelette de fissuration raffiné . . . . . 186

5.40 Détection complète des fissures . . . . . 186

5.41 Application de la méthode OMPS et la méthode ADFD sur des images Aigle-RN de taille 462 × 991 pixels . . . . . 188

5.42 Application de la méthode OMPS et la méthode ADFD sur des images Aigle-RN de taille 462 × 311 pixels . . . . . 190

5.43 Application de la méthode OMPS et la méthode ADFD sur des images $VIAPIX^{®}$ de taille 1216 × 1600 pixels acquises tout les 2 mètres . . . 192

5.44 Application de la méthode OMPS et la méthode ADFD sur des images $VIAPIX^{®}$ de taille 1216 × 1600 pixels acquises tout les 1 mètres . . . 193

5.45 Application de la méthode ADFD sur des images LCMS "élévation" d'une chaussée endommagée incluant un mosaïquage . . . . . 195

5.46 Application de la méthode ADFD sur des images LCMS "élévation" d'une chaussée peu endommagée . . . . . 196

**6 : Conclusion et perspectives**

6.1 Catégories des marquages routiers à traiter dans le futur . . . . . 202

6.2 Tests préliminaires pour classifier les fissures détectées . . . . . 203

**A : Opérations élémentaires de la morphologie mathématique**

A.1 Exemple illustrant l'effet de la dilatation morphologique sur une image binaire . . . . . 222

A.2 Exemple illustrant l'effet de l'érosion morphologique sur une image binaire 223

A.3 Exemple illustrant l'effet de l'ouverture morphologique sur une image binaire . . . . . 223

A.4 Exemple illustrant l'effet de la fermeture morphologique sur une image binaire . . . . . 224

**B : Données techniques sur le système $VIAPIX^{®}$**

B.1 Photographie d'un ipad et d'un PC portable connecté à l'interface WEB 227

B.2 Mise en situtation du $VIAPIX^{®}$ exploitation . . . . . 228

B.3 Cartographie . . . . . 229

B.4 $VIAPIX^{®}$ exploitation . . . . . 229

**C : Algorithme de Dijkstra appliqué à la segmentation de fissures**

C.1 Point source (S) et destination (D) pour le calcul du plus court chemin 233

C.2 Notations . . . . . 234

C.3 Chemin minimal obtenu avec l'algorithme de Dijkstra . . . . . 237

# Liste des tableaux

**1 : État de l'art**

1.1 Facteur de luminance des marquages routiers en Europe. . . . . 14

1.2 Coordonnées chromatiques des marquages routiers en Europe. . . . . 15

**2 : Extraction des dégradations de la signalisation horizontale**

2.1 Sélection des valeurs médianes et des blocs correspondants à conserver. 58

2.2 Spécificités du passage piéton illustré dans la figure 2.34. . . . . 77

2.3 Résultats de la reconnaissance des passages piétons à l'aide de 4 séquences vidéo différentes. . . . . 78

**3 : Détection de fissures par sélection de chemins minimaux**

3.1 Valeurs de précision obtenues lors de l'application des méthodes (Markovienne, GC, FFA, MPS) sur les images Aigle-RN de la figure 3.24. . . 118

3.2 Valeurs de sensibilité obtenues lors de l'application des méthodes (Markovienne, GC, FFA, MPS) sur les images Aigle-RN de la figure 3.24. . . 119

3.3 Valeurs de DSC obtenues lors de l'application des méthodes (Markovienne, GC, FFA, MPS) sur les images Aigle-RN de la figure 3.24. . . . 119

**4 : Optimisations de la sélection des chemins minimaux**

4.1 Temps de calcul global et valeurs DSC obtenus en traitant les six images de la figure 4.23 avec les trois versions de MPS. Les valeurs en gras indiquent les meilleures valeurs obtenues à la fin du traitement. . . . . 149

**5 : Détection de fissures par sélection de filaments sombres**

5.1 Résultats moyens d'évaluation des deux méthodes OMPS et ADFD sur des images Aigle-RN de taille $462 \times 991$ pixels. . . . . 189

5.2 Résultats moyens d'évaluation des deux méthodes OMPS et ADFD sur des images Aigle-RN de taille $462 \times 311$ pixels. . . . . 191

5.3 Résultats moyens d'évaluation des deux méthodes OMPS et ADFD sur des images $VIAPIX^{®}$ de taille $1216 \times 1600$ pixels acquises tout les 2 mètres. . . . . 192

5.4 Temps de traitement des deux méthodes OMPS et ADFD sur des images $VIAPIX^{®}$ de taille $1216 \times 1600$ pixels acquises tous les 1 mètres. . . 194

# Introduction générale

L'entretien du réseau routier est un élément essentiel pour assurer la durabilité de la structure de la chaussée et maintenir une qualité de service et de sécurité suffisante. L'entretien nécessite une phase d'auscultation, pour faire l'inventaire des défauts et en estimer la gravité. Le relevé de dégradations sert à établir un indice de service, qui, en France, permet d'évaluer le financement annuel à consacrer pour l'entretien du réseau national.

Par le passé, l'auscultation était basée sur des relevés visuels au pas d'un opérateur. Cette méthodologie a été remplacée à partir des années 80 par des systèmes d'auscultation à grand rendement, qui collectent à vitesse de trafic des images de la chaussée, qu'un opérateur dépouille a posteriori à partir d'une interface graphique spécialisée. En parallèle, les défauts à relever traditionnellement par l'opérateur (fissures, nids-de-poule, arrachement, etc.) se sont étendus aux défauts de marquages routiers (signalisation horizontale et verticale).

Dans ce contexte, l'automatisation du relevé de défauts par des techniques de traitement d'images et/ou d'intelligence artificielle est devenue un champ de recherche important. L'enjeu est à la fois de fournir un résultat fiable de détection de défauts, d'accélérer l'auscultation du réseau, d'améliorer la qualité de l'entretien du réseau et de sécuriser le travail des opérateurs. Ce travail se focalise sur la segmentation automatique de deux types de défauts, qui présentent des caractéristiques opposées en taille, forme, couleur et photométrie : les marquages routiers horizontaux (chapitre 2), et les fissures (chapitres 3 à 5).

L'objectif du chapitre 1 est de donner une vision globale de la problématique de l'automatisation de détection des défauts en surface de chaussée. Pour ce faire, nous présentons dans un premier temps les deux types de dégradations de surface visées par ce travail de thèse. Ensuite, nous présentons un aperçu de l'état de l'art des nombreuses méthodes de traitement d'images pour la détection de ces deux objets. Enfin, le chapitre 1 présente succinctement les trois systèmes d'imagerie opérationnels dont nous avons exploités les images : Aigle-RN, LCMS et VIAPIX. Du point de vue traitement d'images, les trois systèmes se distinguent par des conditions différentes de prise de vue (incidence verticale vs. oblique), d'éclairage (contrôlé vs. naturel), d'étendue du champ de vision (limité à la chaussée ou panoramique), de résolution, et enfin, par la nature

de l'information délivrée ($2D$ ou $3D$).

Le chapitre 2 expose le procédé automatique de traitement d'images pour l'extraction, la catégorisation et la qualification du marquage au sol à partir d'une image panoramique acquise par un système monté sur le toit d'un véhicule (système VIAPIX). En se focalisant sur les marquages de couleurs blanches, nous proposons dans un premier temps une méthode de segmentation de toutes les zones claires de l'image. La reconnaissance des marquages repose pleinement sur deux techniques différentes : géométrique et optique. La première technique a été utilisée pour la reconnaissance du passage piéton (nombre de bandes, forme des bandes, etc.) alors que la deuxième technique, basée sur la méthode de corrélation optique, a été proposée pour la reconnaissance des pictogrammes spécifiques à la signalisation des couloirs de bus et de pistes cyclables. Dans un deuxième temps, nous déterminons l'état d'usure des marquages. Finalement, l'évaluation de l'état d'usure des marquages routiers est réalisée en prenant en considération par analyse géométrique, colorimétrique et photométrique des signatures des objets identifiés.

Les chapitres 3 à 5 se focalisent sur la segmentation automatique de fissures. Le chapitre 3 détaille le principe et le fonctionnement de la méthode récente MPS (Minimal Path Selection), qui a été sélectionnée au chapitre 1 à l'issue de l'étude sur l'état de l'art. MPS est basée sur la sélection des chemins minimaux dans l'image ; elle favorise ainsi la connexité entre les pixels de niveaux de gris faibles, qui par hypothèses, ont une grande probabilité d'appartenir à une fissure. Elle comporte une phase de segmentation du squelette de fissuration (en 3 étapes), et une phase de post-traitement (en 2 étapes) qui consiste à raffiner le squelette et estimer l'épaisseur locale des fissures. Le cœur de l'algorithme consiste à initialiser la recherche de fissures par un calcul de chemins le plus court (en utilisant l'algorithme de Dijkstra) entre des pixels sombres sélectionnés dans l'image. Des résultats sur des images Aigle-RN montrent que MPS délivre une segmentation plus précise que cinq autres méthodes de la littérature. Par ailleurs, l'analyse des différents résultats permet de déterminer les améliorations envisageables de cet algorithme.

Le chapitre 4 présente une version optimisée de la méthode MPS (nommée "OMPS"), que nous proposons afin de faire face à tous les limites de la méthode classique, à savoir le temps d'exécution et la détection des fissures fines. OMPS propose ainsi d'améliorer les performances de MPS d'un facteur supplémentaire (notamment le taux de similitude) et de réduire le temps d'exécution dans l'environnement de programmation Matlab. Pour ce faire, OMPS utilise conjointement une méthode de seuillage adaptative et une méthode de fusion de données pour réduire la quantité d'information à traiter dans l'image. Les performances d'OMPS sont comparées à l'algorithme initial à partir du même benchmarking qu'au chapitre 3, composé d'images acquises sous un éclairage contrôlé.

Le chapitre 5 propose d'étendre l'usage des méthodes automatiques de segmentation de fissures au cas des images panoramiques collectées en lumière ambiante. En comparaison de MPS et OMPS, la nouvelle méthode ADFD (Automatic Darkest Filament Detection) utilise des techniques rapides et efficaces. Elle permet à la fois d'obtenir une bonne détection de fissures sur des images $2D$ sous éclairage ambiant, mais également, d'améliorer les résultats de segmentation sur les images initiales en éclairage contrôlé. Elle repose comme précédemment sur la sélection de pixels sombres de l'image et le calcul de chemins minimaux par l'algorithme de Dijkstra. ADFD se distingue d'OMPS cependant, par l'exploitation de statistiques locales de l'image. Une méthode de détection de contours est appliquée en pré-traitement pour identifier les zones d'ombrage dans l'image. Ensuite, ADFD inclut une nouvelle étape d'analyse de connectivité entre les fissures afin d'éviter la suppression des fissures isolées de petites tailles. Enfin, la méthode ADFD peut facilement être adaptée pour analyser et traiter des images $3D$ acquises par le système LCMS (Laser Crack Measurement System).

# Chapitre 1

# État de l'art

## Sommaire


1.1 Introduction . . . . . **7**

1.2 Problématiques . . . . . **7**

1.3 Dégradations de surface de chaussées . . . . . **8**

1.3.1 Généralités . . . . . 8

1.3.2 Types de dégradations routières . . . . . 8

1.3.3 Méthodologie pour le relevé de dégradations . . . . . 10

1.3.3.1 Relevé visuel in situ . . . . . 11

1.3.3.2 Relevé visuel sur écran et systèmes imageurs à grand rendement . . . . . 11

1.3.3.3 Relevé automatique . . . . . 12

1.4 Méthodes de détection des marquages routiers . . . . . **13**

1.4.1 Caractérisation optiques des marquages routiers . . . . . 13

1.4.2 Spécification des différents types de marquages . . . . . 13

1.4.3 Propriétés optiques des marquages routiers . . . . . 14

1.4.4 Approches existantes pour la détection du marquage au sol 16

1.4.4.1 Méthodes de détection de lignes . . . . . 16

1.4.4.2 Méthodes orientées contours . . . . . 19

1.4.4.3 Méthodes orientées régions . . . . . 21

1.4.5 Outils de pré-traitement dédiés à l'extraction du marquage au sol . . . 22
1.4.5.1 Transformée perspective inverse (IPM) . . . 22
1.4.5.2 Morphologie mathématique . . . 25
1.4.5.3 Méthodes de seuillage . . . 26
1.4.6 Extracteurs du marquage au sol basés sur la forme . . . 27
1.4.6.1 Opérateur par variation de luminosité . . . 27
1.4.6.2 Opérateur par recherche des gradients symétriques 28
1.4.6.3 Opérateur par patron . . . 29
**1.5 Méthodes de détection de fissures . . . 30**
1.5.1 Hypothèses . . . 30
1.5.2 Synoptique de la détection des fissures . . . 31
1.5.3 Méthodes de segmentation de fissures existantes . . . 32
1.5.3.1 Approches par seuillage . . . 32
1.5.3.2 Approches par morphologie mathématique . . . 34
1.5.3.3 Approches par apprentissage . . . 35
1.5.3.4 Approche par filtre adaptatif . . . 35
1.5.3.5 Approche par transformation en ondelettes continue 36
1.5.3.6 Approche par modélisation markovienne . . . 36
1.5.3.7 Approche heuristique . . . 37
1.5.3.8 Approches par chemins minimaux . . . 38
1.5.3.9 Discussion . . . 41
**1.6 Systèmes d'imagerie utilisés . . . 42**
1.6.1 Systèmes spécialisés d'imagerie de la route . . . 42
1.6.1.1 Appareil Aigle-RN . . . 42
1.6.1.2 Système LCMS . . . 43
1.6.2 Système d'imagerie $VIAPIX^{®}$ . . . 44
**1.7 Conclusion . . . 45**

## 1.1 Introduction

L'entretien du réseau routier est un élément essentiel pour assurer la durabilité de la structure de chaussée et maintenir une qualité de service suffisante. L'entretien passe par une phase de reconnaissance et d'auscultation, pour évaluer l'état du réseau et programmer les opérations de maintenance nécessaires.

Plus généralement, l'auscultation contribue à dresser l'inventaire du patrimoine routier (d'un pays, d'une région, d'une collectivité ou d'un réseau concédé) et à évaluer son état de service. Les dégradations qui sont relevées en surface de chaussée servent à établir un indice de service. Par exemple en France, le Cerema produit un indice IQRN pour le réseau de routes nationales, sur la base duquel l'Assemblée Nationale décide du financement annuel consacré à l'entretien du réseau.

Au départ, l'auscultation était basée sur des relevés visuels d'un opérateur à vitesse de marche. Cette méthodologie a été remplacée à partir des années 80 par des outils d'auscultation à grand rendement. Ces outils sont dans la plupart des cas basés sur des systèmes de collecte d'images à vitesse de trafic, que l'opérateur dépouille a posteriori à partir d'une interface graphique spécialisée. En parallèle, les éléments à relever traditionnellement par l'opérateur (défauts en surface de chaussée) se sont diversifiés aux marquages routiers horizontal et vertical.

Dans ce contexte, l'automatisation du dépouillement des images est devenue un champ de recherche important, qui fait intervenir plusieurs disciplines, e.g., la robotique automobile [1], l'électronique, le traitement d'images et l'intelligence artificielle. L'enjeu est à la fois d'accélérer l'auscultation et d'améliorer la qualité de l'entretien du réseau.

L'objectif de ce chapitre est de donner une vision globale du problème de l'automatisation de la détection des défauts de la chaussée. Pour ce faire, nous présentons dans un premier temps les différents types de dégradations de chaussées, ainsi que les méthodologies existantes pour le relevé de ces dégradations. Ensuite, nous présentons un aperçu général des nombreuses méthodes de traitement d'images pour la détection de dégradations dans des images de surface de chaussées.

## 1.2 Problématiques

Dans le cadre de cette thèse, nous cherchons à développer un système de détection automatique de défauts de surface à partir des images de chaussées. Cet objectif regroupe deux problématiques :

La première problématique concerne la collecte des données routières, qui seront intégrées dans une base de données. Dans ce cadre, des véhicules d'auscultation à grand rendement équipés de plusieurs capteurs sont utilisés. En effet, ils permettent de saisir

à un instant donné une représentation numérique de l'environnement routier. Parmi les capteurs, les caméras embarquées qui possèdent dorénavant des vitesses d'acquisition élevées et des capacités adaptées de stockage de plusieurs Giga Octets [2].

La deuxième problématique, la principale de cette thèse, concerne l'automatisation du traitement des données. Le système d'acquisition génère une grande quantité de données, qu'il convient d'analyser d'une manière massive, automatique et rapide, pour répondre à la problématique de l'auscultation du réseau routier.

## 1.3 Dégradations de surface de chaussées

### 1.3.1 Généralités

Les chaussées se dégradent principalement sous l'effet du trafic (notamment des poids lourds) et des conditions climatiques [3]. En général, les dégradations n'atteignent rarement les couches les plus profondes de la chaussée mais la majorité d'entre elles apparaissent à la surface de la première couche, à savoir *la couche de roulement*. Elles constituent des indicateurs de l'état de la structure de chaussées. Un bon entretien de la première couche permet de protéger l'ensemble de la structure de la route, cf. figure 1.1.

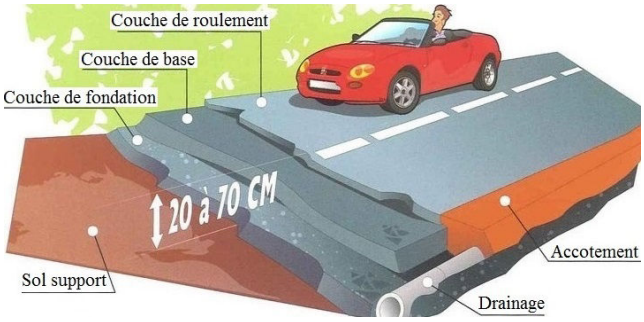



FIGURE *1.1 – Structure et couches de la chaussée.*

### 1.3.2 Types de dégradations routières

Les défauts en surface des chaussées (dénommés également anomalies ou dégradations dans ce document) constituent, dans la plupart des cas, des indicateurs sensibles à l'évolution des caractéristiques structurelles de la chaussée. En France, les différents types de dégradations font l'objet du catalogue associé à la méthode d'essai no. 38-2 intitulée "*Relevé de dégradations de surface des chaussées*" [3]. Selon ce catalogue, nous pouvons distinguer les familles de défauts suivantes :

- **Déformations :**
  - **Affaissement de rive** : Tassement en rive de chaussée formant une cuvette accompagnée d'un bourrelet à l'extérieur de la chaussée.

  - **Flache** : Tassement localisé en pleine, de forme ovale.
  - **Orniérage** : Tassement longitudinal de la chaussée localisé au niveau des bandes de roulement.

- **Fissures :**
  - **Fissure longitudinale** : Cassure de la couche de roulement parallèlement à l'axe de la chaussée.
  - **Fissure transversale** : Cassure de la couche de roulement perpendiculairement à l'axe de la chaussée.
  - **Fissure oblique** : Cassure de dalle rejoignant deux côtés adjacents.
  - **Faïençage** : Maillage de fissures en réseaux sur une zone localisée (très souvent dans les bandes de roulement) formant une série de polygones.
  - **Fissures diverses** : Ils correspondent aux fissures apparaissant d'une façon anarchique à la surface des chaussées. Par exemple, les fissures paraboliques, en étoile ou en Y, etc.

- **Arrachements :**
  - **Nid de poule** : Trou en surface de chaussée laissant apparaitre les matériaux de structure.
  - **Pelade** : Arrachement de la couche de surface par plaques entières.
  - **Plumage** : Arrachement des gravillons d'un enduit superficiel.

- **Remontées :**
  - **Ressuage** : Remontée du liant bitumineux à la surface de la chaussée.

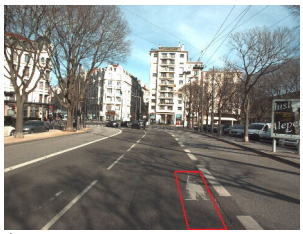
Toujours dans le cadre des dégradations rencontrées, les dommages qui apparaissent sur les signalisations routières sont considérés comme étant des défauts marquants qui attirent l'attention des chercheurs. Parmi d'eux, nous citons par exemple :

- **Détérioration des signalisations routières :**
  - **Signalisation verticale** (panneaux, panonceaux, bornages) : Déformation de la forme ou du continu de la signalisation à cause des accidents.
  - **Signalisation horizontale** (marquage au sol) : Effacement de la peinture sur la chaussée.

Parmi les divers défauts cités, nous nous focalisons dans ce travail de thèse sur la détection des deux types de défauts suivants :

✓ Les anomalies de la **signalisation horizontale** (voir figure 1.2). Notre contribution sur ce sujet est détaillée dans le chapitre 2.

*(a) : Passage piéton*

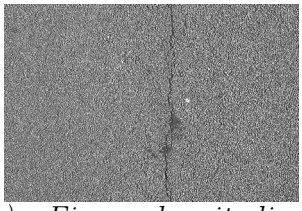


*(b) : Flèche directionnelle*

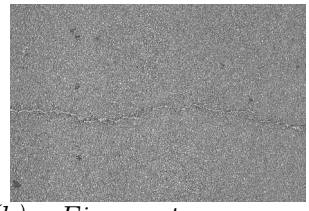
*(c) : Pictogramme vélo*

FIGURE *1.2 – Exemple d'images illustrant l'effacement de plusieurs types du marquage au sol.*

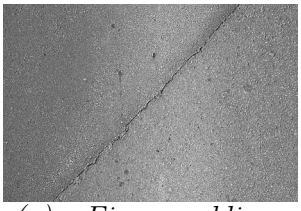
✓ Les anomalies de types **fissures** (voir figure 1.3). Cette partie sera détaillée dans les chapitres 3, 4 et 5. Pour la clarté, nous allons plus se consacrer dans cette thèse sur ce type de dégradations vu qu'il s'agit d'un défaut le plus fréquent et prédominant sur la chaussée.



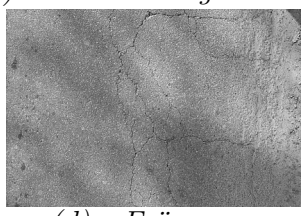
*(a) : Fissure longitudinale*



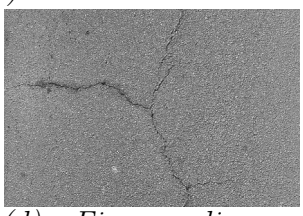
*(b) : Fissure transversale*



*(c) : Fissure oblique*



*(d) : Faïençage*



*(d) : Fissures diverses*

FIGURE *1.3 – Exemple d'images de chaussées illustrant plusieurs types de fissures - Nous distinguons différentes catégories de fissures en fonction de leur orientation et leur concentration.*

### 1.3.3 Méthodologie pour le relevé de dégradations

À la base de toute méthodologie d'auscultation des chaussées, figurent généralement un relevé de dégradations de surface et une interprétation associée [3]. Dans cette section, nous présentons l'évolution des méthodes du relevé des dégradations sur la chaussée en commençant par les méthodes traditionnelles pour arriver aux méthodes les plus avancées.

#### 1.3.3.1 Relevé visuel in situ

Par le passé, des agents spécialisés observaient directement sur le terrain les différentes dégradations de la chaussée. La saisie des dégradations se faisait soit à pied, soit à bord d'un véhicule parcourant la chaussée à une très faible vitesse (entre 5 et 7 km/h).

De nos jours, ce genre d'auscultation est devenu rare ; les dégradations sont notées sur tablette informatique georéférencée. La faible vitesse d'inspection limite le linéaire de réseau que l'on peut ausculter. De plus, cette méthode n'est pas adaptée au trafic routier l'insécurité des agents et des usagers de la route. Cette méthode traditionnelle a été remplacée par la méthode présentée dans le paragraphe suivant.

#### 1.3.3.2 Relevé visuel sur écran et systèmes imageurs à grand rendement

La méthode *in situ* du paragraphe précédent a été remplacée au fil du temps par une méthode en deux étapes. La première étape consiste à collecter des images de chaussées à vitesse du trafic à l'aide de systèmes d'imagerie spécifiques [1]. Dans une seconde étape, les images sont analysées sur écran a posteriori par un opérateur à l'aide d'une interface graphique adaptée.

En France, les premiers appareils d'auscultation à grand rendement (AGR) collectaient les images de chaussée à incidence verticale avec des caméras analogiques, situés à l'arrière du véhicule et sous un éclairage contrôlé, e.g., le système Gerpho [3] développé pour ausculter le réseau de routes nationales. Le film développé était ensuite visualisé par l'opérateur sur un support adapté. Le passage aux systèmes d'imagerie numérique s'est opéré à partir des années 90, avec notamment le système Aigle-RN [4, 5] pour les routes nationales. Le Cerema a utilisé l'outil Visiodec [6] de 1992 à 2018 pour visualiser et dépouiller les images a posteriori. De nos jours, la qualité des systèmes d'imagerie numérique s'est améliorée grâce à l'évolution technologique des capteurs CCD et de la qualité de l'éclairage laser et/ou LED. Parmi les autres systèmes existants du même type ayant opéré en France, nous pouvons citer par exemple le système AMAC (Appareil Multifonction d'Auscultation des Chaussées) [7].

Des systèmes AGR plus polyvalents sont apparus en parallèle à partir des années 2000. Ils utilisent des caméras panoramiques qui prennent à vitesse du trafic une image du paysage routier vers l'avant du véhicule et sous éclairage ambiant. Le paysage routier comprend les bordures de voie, la signalisation horizontale (marquages routiers) et verticale (panneaux de signalisation) ainsi qu'une certaine bande de chaussée. Ces systèmes permettent de dresser l'inventaire du patrimoine routier (marquages), de mesurer la largeur de chaussée et de détecter les dégradations de surface sur une certaine bande de chaussée en avant du véhicule. En comparaison des systèmes AGR spécialisés, les systèmes par caméras panoramiques voient la chaussée sous incidence oblique et la taille des pixels est plus importante. Nous verrons après que cette thèse développe des outils de traitement pour les deux types de systèmes d'imagerie.

#### 1.3.3.3 Relevé automatique

L'objectif principal du relevé automatique est de faire réaliser à une machine les tâches pénibles et répétitives de détection de défauts que réalisent un opérateur sur écran.

L'automatisation nécessite la mise en œuvre et la maîtrise d'une chaîne d'acquisition et de traitement entièrement numérique [8]. Ses performances seront d'autant meilleures que l'image est de bonne qualité (dynamique du signal, taille de pixel adaptée à celle des défauts à détecter).

Plusieurs recherches ont été initiées pour automatiser la détection de défauts de surface sur chaussées [9]. Il convient de distinguer cependant les travaux selon les deux types de systèmes d'auscultation cités au paragraphe précédent : les appareils AGR spécialisés et les appareils polyvalents à base de caméras panoramiques.

Un système pionnier en matière d'auscultation est sans doute l'appareil australien RoadCrack [10], développé par le CSIRO (Commonwealth Scientific and Industrial Research Organisation) et le RTA (Road and Traffic Authority). Il permet de détecter et classifier en temps réel les défauts sur la chaussée à vitesse de trafic (100 km/h). Peu d'informations techniques sont disponibles sur cet appareil ainsi que sur le traitement associé. La partie optique du système a été récemment remise à niveau. Un autre système automatique conçu par la société japonaise Komatsu [11] permet l'extraction de dégradations dans une image $2\,048 \times 2\,048$ pixels. Bien que novateur dans les années 90, ce système n'a été utilisé que pour les chaussées japonaises et ne pouvait fonctionner que la nuit.

Certains fabricants de capteurs fournissent également leur solution logicielle de traitement automatique. En prenant l'exemple de Pavemetrics, dont nous utilisons le capteur par la suite, le résultat de segmentation automatique est superposé (avec un léger décalage latéral paramétrable pour faciliter la visualisation) à l'image de chaussée. La gravité du défaut est traduite par un code de couleurs.

Les systèmes polyvalents par caméras panoramiques ont donné lieu à des traitements automatiques spécifiques à chaque application (reconnaissance de largeur de chaussée, de marquages verticaux. En ce qui concerne la détection de défauts de surface, les traitement automatiques sont confrontés aux difficultés spécifiques liés aux conditions d'éclairage ambiant.

Dans la suite de ce chapitre, nous présentons plus en détail un état de l'art des méthodes de traitement d'images proposées aussi bien dans le domaine de recherche que le domaine industriel pour détecter, analyser et classifier les deux types de défauts de surface visés dans cette thèse : (1) les défauts de marquages routiers (marquage au sol), et (2) les défauts de surface de type fissures. Ces deux types de défauts donnent

lieu à des traitements spécifiques car ils présentent des caractéristiques photométriques et des formes géométriques différentes.

## 1.4 Méthodes de détection des marquages routiers

La détection des dégradations des marquages routiers nécessite au préalable une étape d'identification de ces derniers dans l'image du paysage routier [12, 13]. Une fois les objets reconnus, une phase d'interprétation de leur état peut être réalisée (cf. chapitre 2). Dans la suite de cette section, l'état de l'art se focalise sur les traitements existants pour détecter et extraire dans les images les différents types de marquages routiers horizontaux.

### 1.4.1 Caractérisation optiques des marquages routiers

Les méthodes de détection des marquages routiers exploitent leurs caractéristiques photométriques et géométriques. Ces dernières varient selon les normes en vigueur dans les différents pays. C'est la raison pour laquelle nous présentons au paragraphe 1.4.2 suivant quelques spécifications des signalisations routières existantes à travers le monde, pour pouvoir développer un modèle générique.

### 1.4.2 Spécification des différents types de marquages

Ce paragraphe permet de lister un certain nombre de caractéristiques de marquages routiers à travers le monde même si la liste reste loin d'être exhaustive. La signalisation destinée aux usagers de la route incluent les deux catégories suivantes :

- **La signalisation verticale :** définie par l'ensemble des panneaux, des panonceaux, et des bornages implantés verticalement par rapport au plan de la route. Elle aide l'usager dans ses prises de décisions, l'informe quant au potentiel danger et impose certaines règles de conduite.

- **La signalisation horizontale :** située sur la chaussée, regroupe l'ensemble des marquages au sol (passage piéton, flèches directionnelles, pictogrammes vélos, lignes zigzag, places de stationnement, etc.). Elle a pour fonction de guider l'automobiliste dans l'espace navigable ainsi que d'informer les usagers quant aux règles de conduite, et de stationnement. Comme précisé auparavant, nous nous focalisons au paragraphe 1.4.4 sur la présentation de l'état de l'art des méthodes de traitement d'images pour détecter ce type de marquage.

La convention de Vienne [14] sur la circulation routière, dont la France est un pays signataire, tente d'harmoniser les règles de marquage au sol à travers le monde. Elle spécifie par exemple que les marquages au sol pour les voies de circulation doivent être blancs ou jaunes, et que les les marquages continus permettent de délimiter les deux

côtés du sens de circulation. Même si cette convention est largement adoptée à travers le monde, elle ne peut pas être considérée comme étant une règle générale.

D'autres types de marquages jaunes peuvent exister sur la chaussée pour un usage spécifique : espace bus, espace taxis, délimitation de zones de travaux de maintenance. Le choix du jaune est essentiellement historique, il correspondait à l'époque des premiers marquages à la couleur associée aux alertes.

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

FIGURE *1.4 – Illustration d'autres types de marquages au sol – (a) Place réservée pour les bus - (b) Place réservée pour les taxis - (c) Marquage en jaune pour informer l'usager que des travaux de maintenance sont en cours de réalisation.*

### 1.4.3 Propriétés optiques des marquages routiers

Les méthodes de détection de marquages sont principalement basées sur les propriétés photométriques des peintures utilisées pour la signalisation horizontale. D'après la convention de Vienne [14], le contraste entre la chaussée et les marquages au sol doit être le plus élevé possible. Le critère de luminance (énergie réfléchie par le marquage) est ainsi un critère essentiel sur lequel la majorité des méthodes de traitement d'images se base pour détecter correctement les marquages au sol. Selon les travaux cités dans [15], le critère de chromaticité (couleur) permet d'améliorer la détection. Dans ce travail de thèse, nous utilisons la norme européenne NF EN 1871, pour spécifier les caractéristiques de luminance et de chromaticité des marquages jaunes et blancs.

L'exigence envers le facteur de luminance est définie par classe en fonction des caractéristiques et de l'environnement du marquage. Nous distinguons trois classes de facteurs de luminance pour la couleur blanche et deux pour la couleur jaune (tableau 1.1).

| **Couleur** | **Classe** | **Facteur de luminance** |
|---|---|---|
| **Blanche** | LF5 | $\geq 0.75$ |
| | LF6 | $\geq 0.80$ |
| | LF7 | $\geq 0.85$ |
| **Jaune** | LF1 | $\geq 0.40$ |
| | LF2 | $\geq 0.50$ |

TABLE *1.1 – Facteur de luminance des marquages routiers en Europe.*

La chromaticité des marquages blanc et jaune est spécifiée par des zones de coordonnées chromatiques dans le diagramme de la figure 1.5. Ces zones sont définies par 4 coordonnées $(x, y)$ pour chacun des sommets des deux quadrilatères. Les coordonnées sont exprimées selon le système CIE XYZ (Commission Internationale de l'Éclairage) défini en 1931, pour avoir description des couleurs plus conforme à la vision humaine.

| **Numéro du sommet** | | **1** | **2** | **3** | **4** |
|---|---|---|---|---|---|
| **Marquage blanc** | x | 0.355 | 0.305 | 0.285 | 0.335 |
| | y | 0.355 | 0.305 | 0.325 | 0.375 |
| **Marquage jaune** | x | 0.494 | 0.545 | 0.465 | 0.427 |
| | y | 0.427 | 0.455 | 0.535 | 0.483 |

TABLE *1.2 – Coordonnées chromatiques des marquages routiers en Europe.*

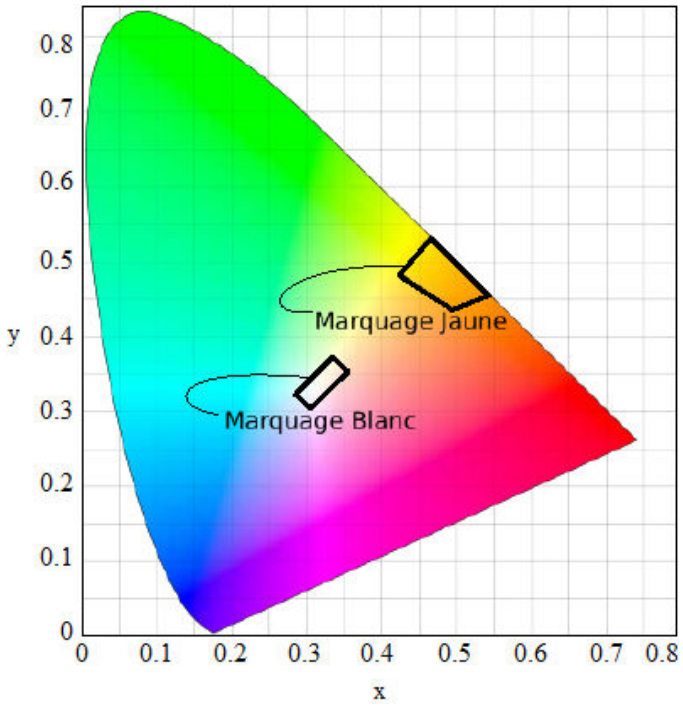



FIGURE *1.5 – Zones de chromaticité pour les marquages jaunes et blancs en Europe.*

Selon la figure 1.5, la zone de chromaticité des marquages blancs englobe la lumière blanche. Ces marquages sont ainsi considérés comme achromatiques dans le cas d'une source lumineuse blanche (ce qui est le cas en période diurne). De plus, la visibilité des marquages de nuit est assurée soit par l'éclairage ambiant, soit par la lumière des phares. Des dispositifs rétro-réfléchissants permettent aussi d'assurer la visibilité des marquages en renvoyant la lumière vers l'automobiliste, améliorant ainsi leur visibilité. Cette rétro-réflexion peut être obtenue par des micro-billes de verre ajoutées aux marquages routiers, couramment utilisées sur les routes hors des agglomérations.

### 1.4.4 Approches existantes pour la détection du marquage au sol

Nous décrivons dans cette partie les méthodes existantes pour la détection de la signalisation horizontale d'une part, et d'autre part, pour la reconnaissance du type de marquage. Une synthèse de la problématique de la détection du marquage au sol (Automatic Lane Finding ou ALF en anglais) est proposée dans [16]. Dans cette synthèse, les techniques de vision doivent prendre en considération des phénomènes perturbateurs comme les conditions d'éclairage variables (lumière contrôlée ou ambiante) ou des variations de luminosité dans l'image (ombres portées).

#### 1.4.4.1 Méthodes de détection de lignes

Ces méthodes de détection sont très nombreuses. En général, elles sont appliquées pour estimer les paramètres de marquages linéaires sur la route. D'après [17–20], les méthodes de détection de lignes les plus connues sont les méthodes basées sur le principe de *thinning*, les méthodes basées sur l'Analyse en Composantes Principales (ACP) et celles basées sur la transformée de Hough.

**a. Thinning**

La technique de *thinning*, souvent appelée *squelettisation*, est une technique de vectorisation largement répandue [21]. Elle fait intervenir des opérations de morphologie mathématique [22]. Selon la figure 1.6, le squelette de la surface noire de l'objet est la représentation la plus synthétique possible qui conserve la même structure topologique que celle de l'image originale. La squelettisation des objets simplifie l'analyse de l'image originale dans le processus de détection de lignes dans l'image.

Les auteurs dans [23] ont proposé un paradigme d'hypothèse prédiction-vérification. Ce paradigme permet de prédire les segments de droites et leur orientation selon la position et l'orientation des pixels du contour. Ensuite, une vérification avec des tests statistiques est effectuée sur la ligne afin de la valider ou de la rejeter. Nous notons que l'objectif principal de cette technique est d'obtenir les squelettes de tous les objets dans une image. Cette technique a montré quelques limitations liées à la génération du squelette qui crée parfois de fausses branches au niveau des jonctions [21].

FIGURE *1.6 – Application de la technique de thinning sur une image binaire – (a) Image binaire représentant une bande rectangulaire décrivant une partie d'une ligne linéaire sur la route - (b) Résultat obtenu après la squelettisation.*

### b. Analyse en Composantes Principales (ACP)

L'analyse en composantes principales consiste à réduire l'espace des données en projetant celles-ci sur les axes représentant aux mieux leur distribution. Les composantes principales de l'espace de données sont déduites des vecteurs propres et des valeurs propres de la matrice de covariance des données. Les vecteurs propres représentent la direction principale de la répartition des points de données tandis que les valeurs propres représentent leur importance. Dernièrement, des études ont été réalisées pour la détection des lignes en utilisant la technique ACP. Dans [19], les auteurs ont proposé d'appliquer le détecteur de contour Canny [24] afin de générer l'image binaire de contour. Sur cette image de contour, les segments de droites horizontaux et verticaux sont extraits selon leurs formes primitives. Ces derniers sont marqués, et l'analyse en composantes principales est réalisée pour chaque segment marqué. L'utilisation des composantes principales permet de détecter les lignes droites ainsi que leurs orientations [25]. Cependant, l'application de la technique ACP dans la littérature montre des résultats insuffisants pour les détecteurs des lignes.

### c. Transformée de Hough

La transformée de Hough est une technique de reconnaissance de formes telles que la détection des lignes [26,27], la détection de cercles et d'ellipses [20]. Inventée en 1959 par Paul Hough, le principe de cette transformée se base sur la représentation polaire des points correspondants aux contours présents dans l'image. En effet, les contours peuvent être extraits en utilisant une technique de seuillage et de détection de contours [24, 28, 29]. L'idée de base de la transformée de Hough est de représenter chaque pixel du contour par les coordonnées polaires $(\theta, \rho)$ de toutes les droites qui le traversent, en sachant que $\theta$ et $\rho$ sont respectivement l'angle et la norme de chacune de ces droites dans l'espace polaire. Autrement dit, $\rho$ représente la longueur du segment perpendiculaire à la droite de l'angle $\theta$ passant par l'origine, comme c'est illustré dans la figure 1.7 à gauche. Pour chaque point, une sinusoïde est obtenue à partir des coordonnées polaires des droites hypothétiques passant par ce point, comme le montre la figure 1.7 à droite. L'ensemble des sinusoïdes obtenues représente l'espace de Hough où un croisement d'une multitude de sinusoïdes traduit l'alignement d'un ensemble de points dans l'image originale.

Cette méthode de détection de lignes donne de très bons résultats dans [26, 27]. Cependant elle reste insuffisante pour notre problématique de détection des lignes spécifiques de marquage linéaire au sol. De plus le principe de de la transformée de Hough n'est pas adapté au cas des virages.

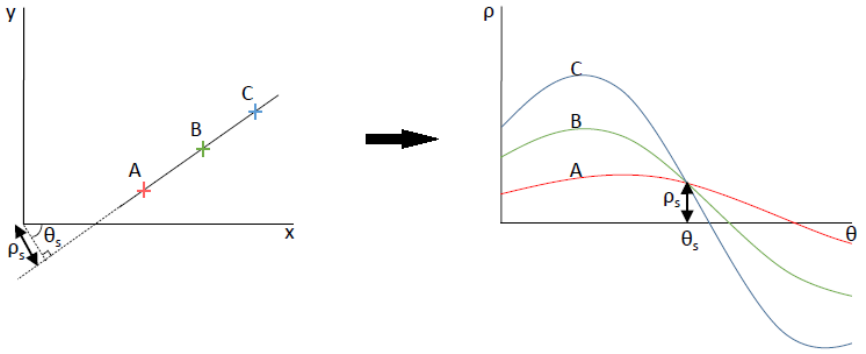



FIGURE *1.7 – Principe de la détection de lignes en utilisant la transformée de Hough – Les trois points A, B et C sont alignés (figure à gauche) si leurs courbes dans l'espace de Hough (figure à droite) se croisent en un point.*

### d. Algorithme RANdom SAmple Consensus (RANSAC)

L'algorithme RANSAC [30] a été proposé pour déterminer la population majoritaire appartenant à une courbe. Il est utilisé dans [31] pour déterminer simultanément les deux lignes de marquage de bordure de route. Cette méthode sépare de manière itérative les données observées en un ensemble de valeurs *inliers* et *outliers* (aberrantes). Il s'agit d'un algorithme non-déterministe au sens où il produit un résultat correct avec une certaine probabilité.

La figure 1.8 illustre l'application de cette algorithme pour la détection d'une ligne dans un ensemble de points. Dans cet exemple, nous supposons que l'ensemble de données de la figure 1.8.a contient à la fois des points inliers (points qui peuvent être approximativement ajustés à une ligne droite), et des points outliers (points éloignés de ce modèle de ligne). La méthode RANSAC génère un modèle qui ne prend en compte que les points pertinents, à condition que la probabilité, lorsque nous sélectionnons ces points pertinents au hasard, soit suffisamment élevée.

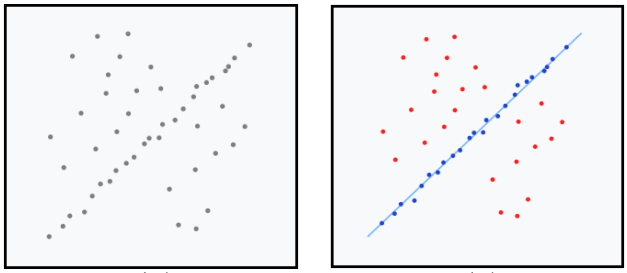



FIGURE *1.8 – Ajustement d'une ligne située dans un plan 2D à une série d'observations en appliquant l'algorithme RANSAC – (a) Série d'observations avec de nombreuses de valeurs aberrantes - (b) Ligne droite ajustée (en bleu).*

D'après [32], cette méthode donne de bons résultats pour extraire des segments indépendants. Cependant, son utilisation est restreinte par la difficulté d'ajuster les paramètres de l'algorithme pour maintenir un niveau de probabilité suffisamment élevé.

#### 1.4.4.2 Méthodes orientées contours

**a. Définition**

La détection de contours est une étape préliminaire à de nombreuses applications de traitement et d'analyse d'images [33]. L'objectif principal de la détection de contours est de transformer une image en niveaux de gris en une image résultante dans laquelle les contours apparaissent par convention en blanc sur fond noir [29]. Les contours constituent des indices riches pour toute interprétation ultérieure de l'image. La détection de contours d'une image permet de réduire la quantité de l'information à traiter tout en préservant les propriétés structurelles importantes dans l'image.

Plusieurs méthodes de détection de marquages existent dans la littérature. Ces approches reposent sur le fait que les contours des marquages correspondent à des discontinuités d'ordre 0 de l'intensité de l'image. Elles sont regroupées en deux catégories : **approche gradient** et **approche laplacien**. La première catégorie détermine les extremums locaux de la dérivée première de l'intensité. La deuxième catégorie recherche les annulations de la dérivée seconde, qui correspondent aux passages par zéro du laplacien. Nous notons que pour la détection des marquages routiers, les approches gradient sont les plus utilisées.

En termes mathématiques, le gradient d'une fonction de deux variables $f(x, y)$, à savoir l'intensité en fonction des coordonnées de l'image, est un vecteur à deux dimensions dont les composantes sont les dérivées selon les directions horizontales et verticales (équation 1.1). En chaque point, le gradient pointe dans la direction du plus fort changement d'intensité, et sa longueur représente le taux de variation dans cette direction.

Le gradient d'une image $f(x, y)$ est donné par le vecteur suivant :

$$\overrightarrow{G} = \overrightarrow{\nabla f} = \left[\frac{\partial f(x,y)}{\partial x} \quad \frac{\partial f(x,y)}{\partial y}\right]^t \tag{1.1}$$

Le module du gradient est présenté par l'équation 1.2 suivante :

$$G = |\overrightarrow{\nabla f}| = \left[(\frac{\partial f(x,y)}{\partial x})^2 + (\frac{\partial f(x,y)}{\partial y})^2\right]^{\frac{1}{2}} \tag{1.2}$$

**b. Opérateurs du gradient**

Ces opérateurs sont à considérer comme des filtres que l'on applique à l'image. Les réponses impulsionnelles de ces filtres peuvent se présenter sous la forme des fonctions

analytiques souvent d'une seule variable ou bien sous la forme de masques bidimensionnels [33]. Dans les deux cas, le filtrage a lieu en deux étapes : un filtrage suivant les lignes de l'image puis suivant les colonnes dans le cas d'une expression mono-dimensionnelle de la réponse impulsionnelle du filtre, une convolution bi-dimensionnelle de l'image avec deux masques pour obtenir les gradients dans des directions orthogonales dans l'autre cas.

Plusieurs opérateurs existent [33] dans la littérature : filtre de Roberts, Prewitt, Kirsch, Robinson, boussole, etc. L'expérience nous a montré le choix d'un filtre peut dépendre du type d'image et de l'application. Par exemple, nous utilisons le filtre de Roberts sur des images composées de contours obliques, le filtre de Prewitt sur des images qui contiennent davantage de contours horizontaux, le filtre de Kirsch pour des contours de directions prédéterminées.

Au chapitre 5, nous avons utilisé le filtre de *Sobel* pour sa polyvalence et sa rapidité. Cet opérateur utilise généralement deux matrices de convolution de taille $3 \times 3$. Par exemple, les gradients horizontaux $D_x$ et verticaux $D_y$ d'une image $I(x, y)$ sont calculés selon les deux équations 1.3 et 1.4.

$$D_x = I * \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \tag{1.3}$$

$$D_y = I * \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \tag{1.4}$$

Le module du gradient est donné par l'équation suivante 1.5.

$$G = \sqrt{D_x^2 + D_y^2} \tag{1.5}$$

### c. Approches existantes

Une approche citée dans [34] est utilisée en développant une technique utilisant les *B-Snake* (principe de contours actifs présenté dans [35,36]). Proposée pour l'estimation des courbes de marquage, cette approche est innovante car les *B-Snake* représentent des courbes paramétrées fondées sur les *spline*. En effet, les contours actifs permettent, quant à eux, de converger sous l'influence de forces externes provoquées dans le cas présent par des contours. La méthode est donc dépendante de l'extraction de contours (en l'occurrence à partir du filtre de Canny [24] souvent utilisé pour générer l'image binaire de contours [19]). En effet, la détection de contours reste cependant une méthode très sensible au bruit pour extraire à elle seule le marquage au sol. Dans [37] par exemple,

les auteurs couplent la détection de contours avec la stéréovision pour pouvoir filtrer les contours n'appartenant pas au plan de la route.

La notion de filtre orienté (*steerable filter*) généralise le calcul du gradient à toutes les directions de l'image. Nous les introduisons au paragraphe 1.4.6.3 en référence aux travaux dans [38, 39] mais sont présentés comme des filtres insuffisants face à d'autres filtres dans [40]. Une technique de mesure proche de celle des filtres orientés a été proposée dans [31]. Elle attribue une saillance importante ou *"ridgeness"* aux pixels dont le profil de niveau de gris correspond à une crête (*ridge*) dans l'image.

Pour l'analyse du marquage au sol, la majorité des techniques orientées contours ne sont pas utilisées seules, mais d'après [2], combinées avec d'autres algorithmes sophistiqués. Leur domaine d'application privilégié semble être le domaine de l'aide à la conduite [41].

#### 1.4.4.3 Méthodes orientées régions

La détection de la signalisation horizontale en utilisant des méthodes orientées régions diffère des approches précédentes par le fait que l'on cherche à segmenter dans l'image des objets étendus à part entière [2].

Dans ce cadre, plusieurs recherches ont été proposées dans la littérature. Les auteurs de [42] par exemple, ont choisi de travailler dans l'espace image $YUV$, où $Y$ est la composante de luminance , et $U$ et $V$ sont les composantes bleue et rouge de la chrominance. Un seuil global est établi à partir d'un histogramme pour séparer la zone de la chaussée et la zone qui contient le marquage. Cette méthode présente des difficultés lorsque les marquages sont des zones localement ombragées.

Dans une autre approche qui sera davantage détaillée dans la section 1.4.6, les auteurs de [43,44] considèrent que les marquages sur la chaussée correspondent à une transition spatiale *noir-blanc-noir* (resp. *noir-jaune-noir*) dans le cas du marquage blanc (resp. jaune). Cette approche est considérée comme l'une des méthodes d'extraction les plus efficaces de la littérature ; comme toute méthode, ses performances sont moindres lorsque le marquage est en partie effacé. Cependant, les auteurs de [40] démontrent que les algorithmes de seuillage local, détaillés à la section 1.4.5.3, peuvent être tout aussi efficaces [45].

### 1.4.5 Outils de pré-traitement dédiés à l'extraction du marquage au sol

#### 1.4.5.1 Transformée perspective inverse (IPM)

Sous l'hypothèse d'une route localement plane, la transformation perspective inverse (IPM : Inverse Perspective Mapping) [46] a pour objectif de créer une image de chaussée vue du nadir (à incidence verticale), qui élimine les effets de distorsion géométriques de la perspective. Ainsi, la transformée IPM permet de faciliter la mise en œuvre des traitements de segmentation automatique pour extraire la signalisation horizontale sur la chaussée. En outre, elle permet de réduire l'instabilité ressentie par les caméras montées sur un véhicule en mouvement [47].

Dans ce contexte, plusieurs travaux ont été menés ces dernières décennies. Par exemple, les auteurs de [48] proposent une technique de reconnaissance des modulations du marquage routier grâce à du filtrage morphologique sur une image perspective inverse. Dans [44, 49], les auteurs proposent d'utiliser l'IPM couplée à une détection de transition orientée région qui considère les lignes du marquage au sol comme un passage "noir-blanc-noir".

La mise en oeuvre de la transformée IPM nécessite une phase préliminaire de calibrage des caméras par rapport à un plan. Grâce aux paramètres de calibrage intrinsèques et extrinsèques de la caméra, nous sommes capable de déterminer l'homographie entre le plan de la route et le plan d'image (voir figure 1.9).

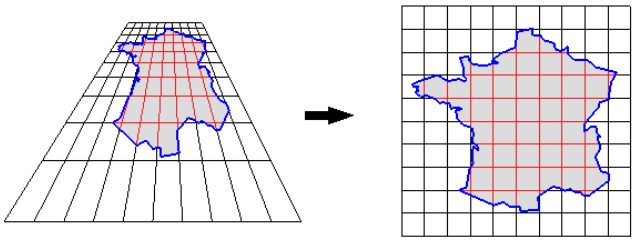


FIGURE *1.9 – Illustration de la transformée IPM sur la vue perspective d'une carte géographique.*

L'ensemble des paramètres intrinsèques modélise la géométrie interne et les caractéristiques optiques du capteur, il contient :

- La distance focale $f$, qui correspond à la distance (en mm) séparant le plan image (rétinien) et le point focal $O_c$.

- Les paramètres de conversion $k_u$ et $k_v$ qui représentent l'ajustement horizontal et vertical permettant de passer d'un repère du plan rétinien (en mm) au repère image (en pixel).

– La position ($u_0$, $v_0$) du point central (centre de l'image) qui est définie comme étant le projeté du centre optique $O_c$ sur le plan image, exprimé en pixel.

L'ensemble des paramètres extrinsèques définit la relation entre le repère caméra localisé sur le centre optique avec le repère du monde localisé dans la scène.

Il est souvent utile de décrire les points du monde réel, la géométrie de la caméra et les points d'image dans des systèmes de coordonnées séparés. Ainsi, la description formelle de la projection implique des transformations entre ces différents systèmes. La figure 1.10 illustre les transformations existantes entre le repère de la scène et le repère de l'image.

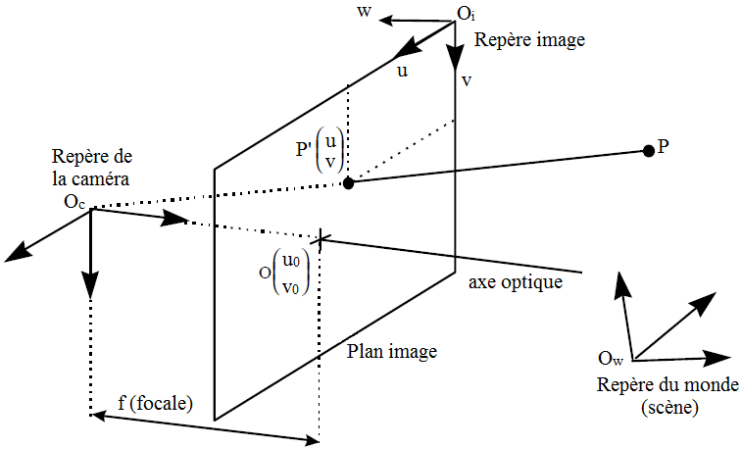



FIGURE *1.10 – Représentation géométrique des repères utilisés dans la transformée perspective inverse.*

Pour mieux comprendre cette technique, nous présentons ci-après un exemple des différentes étapes de la transformée IPM. Nous considérons dans un premier temps un espace à trois dimensions, défini par une matrice $M_{ext}$ des paramètres extrinsèques de la caméra. Les coordonnées du nouveau système sont spécifiés par une translation $T$ et une rotation $R$ par rapport à l'ancien système. D'une part la translation est définie par l'équation 1.6 suivante :

$$T = \begin{bmatrix} 0 \\ 0 \\ h \end{bmatrix} \tag{1.6}$$

où $h$ représente la distance entre le sol et la caméra. D'autre part, $R$ peut être décomposée entre trois matrices obtenues suivant les trois orientations selon les axes X, Y et Z.

$$R_X = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\theta_X) & -sin(\theta_X) \\ 0 & sin(\theta_X) & cos(\theta_X) \end{bmatrix} \tag{1.7}$$

$$R_Y = \begin{bmatrix} cos(\theta_Y) & 0 & sin(\theta_Y) \\ 0 & 1 & 0 \\ -sin(\theta_Y) & 0 & cos(\theta_Y) \end{bmatrix} \tag{1.8}$$

$$R_Z = \begin{bmatrix} cos(\theta_Z) & -sin(\theta_Z) & 0 \\ sin(\theta_Z) & cos(\theta_Z) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{1.9}$$

où $\theta_X$, $\theta_Y$ et $\theta_Z$ représentent respectivement les trois rotations selon les trois axes des $X$, $Y$ et $Z$. En général, ils correspondent aux angles relatifs *Roulis*, *Tangage* et *Lacet* de la caméra et du véhicule.

La matrice regroupant les paramètres extrinsèques est ainsi la multiplication des trois matrices $R_X$, $R_Y$ et $R_Z$, selon l'équation 1.10 suivante :

$$R = R_X.R_Y.R_Z \tag{1.10}$$

Pour prendre en compte la distance sol-caméra, la matrice extrinsèque (matrice permettant de passer des coordonnées dans le monde aux coordonnées dans le repère de la caméra) est donnée par l'équation 1.11 suivante :

$$M_{ext} = \begin{bmatrix} R & T; & 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.11}$$

La caméra est pourvue d'un système optique de focale $f$ et d'un centre optique ($u_0$, $v_0$). Les paramètres dits intrinsèques de la caméra sont notés dans la matrice $M_{int}$, qui est définie par l'équation 1.12 suivante :

$$M_{int} = \begin{bmatrix} \frac{f}{k_u} & 0 & u_0 & fu_0 \\ 0 & \frac{f}{k_v} & v_0 & fv_0 \\ 0 & 0 & 1 & f \end{bmatrix} \tag{1.12}$$

Enfin, nous déduisons la matrice de projection permettant de connaître les coordonnées d'un point dans l'image. Cette matrice est définie par l'équation 1.13 suivante :

$$M = M_{int}.M_{ext} \tag{1.13}$$

Un point crucial à prendre en compte est que la précision de la transformée IPM se dégrade avec la profondeur de champ, i.e., pour les pixels qui s'approchent de l'horizon. En pratique, pour la problématique de l'auscultation de la surface de chaussée, la transformation IPM s'applique dans l'image panoramique sur une profondeur de champ

limitée, c'est-à-dire sur une bande inférieure de l'image au plus proche de la caméra. La figure 1.11 illustre le résultat de la transformation IPM sur une image de paysage routier urbain.

*(a)* *(b)*

FIGURE *1.11 – Exemple d'application de la transformée perspective inverse – (a) Image originale - (b) Image simplifiée.*

Nous notons que cette transformée n'est qu'une phase intermédiaire permettant de filtrer facilement des formes appartenant au plan de la chaussée. En effet, chercher des formes de taille constante prédéfinie devient plus facile si nous disposons d'une vue verticale. Ainsi, des filtres morphologiques peuvent être appliqués pour extraire la signalisation horizontale sur la route. Par exemple, si nous cherchons à extraire les bandes de marquage dans une image en vue d'oiseau, nous pourrions définir un élément structurant rectiligne parallèle à la signalisation.

#### 1.4.5.2 Morphologie mathématique

La morphologie mathématique est une théorie fondamentale de traitement non-linéaire apparue en France dans les années 60 dans les laboratoires de MINES ParisTech [50]. Elle repose sur des ensembles de références appelés *éléments structurants*.

En traitement d'images, les opérateurs de morphologie permettent de fournir des outils pour toute la chaîne de traitement d'images, des pré-traitements (filtrage, rehaussement de contraste, etc.) à la segmentation et à l'interprétation de scènes. Ils permettent de transformer les images et d'extraire les caractéristiques des objets. Nous distinguons quatre opérations élémentaires de morphologie, qui sont illustrés en annexe A.

La littérature indique quelques applications de la morphologie à l'extraction du marquage au sol. La méthode de [51] pour la détection des routes utilise une combinaison d'opérations morphologiques. Dans un premier temps, l'image originale est segmentée et les routes sont séparées grossièrement de leur environnement. Une analyse est réalisée afin de déterminer la distribution de la taille et de la forme des objets dans l'image. Ensuite, une ouverture est effectuée pour déterminer les composantes connexes de l'ensemble des routes prédétectées. Les auteurs cherchent par la suite l'ellipse de taille minimale englobant chacun de ces composantes connexes ainsi que leur axe prin-

cipal. En effet, ceci permet de garder seulement les composantes dont l'axe principal de l'ellipse est supérieur à une certaine valeur du seuil. Afin de détecter les marquages, une simple méthode de seuillage permettant de détecter la couleur blanche, est ainsi appliquée. L'avantage de cette méthode est qu'elle permet de réduire au maximum la quantité des objets qui n'appartiennent pas à la chaussée (ciel, voiture, etc.).

#### 1.4.5.3 Méthodes de seuillage

L'étude proposée dans [40] présente plusieurs méthodes d'extraction de marquages routiers. Selon les auteurs, les méthodes qui présentent des meilleurs résultats utilisent deux types de seuillage local des caractéristiques de l'image : le seuillage local simple et le seuillage local symétrique.

a. **Seuillage local (LT) :** D'après [40], pour qu'un pixel, de coordonnées $(x, y)$ dans une image $I$ en niveau de gris, soit considéré comme un pixel appartenant à un marquage au sol, il faut que la valeur de ce pixel soit supérieure à une certaine valeur $S$ de la moyenne des intensités de l'image $\overline{I}$ sur un voisinage donné $V$.

$$I(x, y) > S + \overline{I}_V(x, y) \tag{1.14}$$

b. **Seuillage local symétrique (SLT) :** Dans cette méthode, les auteurs calculent deux moyennes, une à gauche et l'autre à droite du pixel à traiter. Grâce à ces deux valeurs, une condition de seuillage ***ET*** logique est appliquée. Un pixel est ainsi considéré comme un pixel de marquage au sol si sa valeur de niveau de gris est supérieure à une certaine valeur $S$ de la moyenne du voisinage à gauche $\overline{I}_{V_g}$ ***ET*** de la moyenne du voisinage à droite $\overline{I}_{V_d}$ :

$$I(x, y) > S + \overline{I}_{V_g}(x, y) \quad ET \quad I(x, y) > S + \overline{I}_{V_d}(x, y) \tag{1.15}$$

Ce type de seuillage fait partie de la catégorie des filtres linéaires utilisés pour réduire le niveau de bruit dans une image. Il est optimum pour un bruit gaussien. Dans [2], l'auteur propose de généraliser ce type de seuillage en réécrivant les équations 1.14 et 1.15 sous forme de résidus, c'est-à-dire comme une différence entre l'image originale et son image filtrée sur un élément structurant, i.e., le voisinage $V$, de taille variable, selon :

$$Residus = (I - \overline{I}_V)(x, y) \tag{1.16}$$

La taille du voisinage $V$ est choisie en fonction du numéro de la ligne du pixel dans l'image. Un seuillage global est ensuite appliqué aux images des résidus afin de classifier les pixels en deux classes. Un exemple d'application est illustré à la figure 1.12.

(a) (b) (c)

FIGURE *1.12 – Exemple du calcul du résidus – (a) Image originale - (b) Image filtrée à partir du filtre moyen - (c) Résidus associé.*

Les performances de ces méthodes dépendent de la taille du voisinage qu'il faut choisir dans l'image. Notons que des méthodes de seuillage similaires sont également utilisées pour la segmentation de fissures à la section 1.5.

### 1.4.6 Extracteurs du marquage au sol basés sur la forme

Une primitive de marquage routier est définie comme une information primaire permettant d'estimer la forme d'un marquage routier. Ces primitives sont le plus souvent décrites comme des points sur le plan de la chaussée appartenant au marquage routier. D'après [15], de nombreuses méthodes permettant d'extraire ces marquages, existent dans la littérature.

#### 1.4.6.1 Opérateur par variation de luminosité

Cette catégorie d'opérateurs propose d'exploiter le gradient dans l'image pour identifier les bords des marquages. Dans ce cadre, les méthodes les plus simples utilisent des algorithmes de détection de contours comme présenté dans [52]. Les auteurs ici proposent d'utiliser les images acquises par des caméras embarquées pour extraire les contours et ensuite extraire les marquages à l'aide de la transformée de Hough (cf. section 1.4.4.1.c). Notons qu'il existe plusieurs types d'opérateurs de détection de contours utilisés pour la détection de marquages. Ils se distinguent par leurs masques locaux de convolution influant sur différents paramètres tels que l'orientation du gradient recherché. Par exemple, il y a ceux qui utilisent le filtre de Canny [53] ou des filtres moins complexes comme le filtre de Sobel [54]. Dans ces articles, les auteurs supposent que les résultats de tous ces types d'extracteurs sont toutefois sensiblement les mêmes. L'un des défauts de ces méthodes est qu'elles peuvent confondre les trottoirs, les jonctions entres différentes plaques de bitume et les ombres projetées sur la chaussée avec des primitives de marquages routiers, comme présenté dans la figure 1.13. Pour résoudre en partie ces problèmes, certains travaux proposent d'étudier l'orientation des gradients détectés à travers un filtre de Sobel et de les indexer selon 8 classes de directions [55]. Les points sont ensuite filtrés selon l'intensité et la direction du gradient. D'autres travaux comme dans [56], ont proposé de ne pas filtrer les primitives selon l'orientation mais plutôt de pondérer les gradients en fonction de leur angle avec le bord de voie sur la route. Cette méthode permet de réduire l'effet des gradients mal orientés pendant la détection

des marquages sans avoir besoin d'appliquer une valeur de seuil sur les gradients en fonction de leur orientation. Cette méthode suppose que le véhicule circule dans le sens de l'orientation des voies.

(a) (b)

FIGURE *1.13 – Exemple de la détection de contours – (a) Image originale - (b) Image après la détection de contours.*

#### 1.4.6.2 Opérateur par recherche des gradients symétriques

Cette catégorie d'opérateur permet d'utiliser la forme des marquages en cherchant principalement les transitions noir-blanc-noir dans l'image. Dans [57] par exemple, les auteurs proposent dans un premier temps de rechercher un gradient ayant une intensité supérieure à une certaine valeur de seuil. Ensuite, ils recherchent un deuxième gradient dans le voisinage permettant d'obtenir une paire de gradients $(G_1, G_1)$ de signe opposé, comme illustré à la figure 1.14.

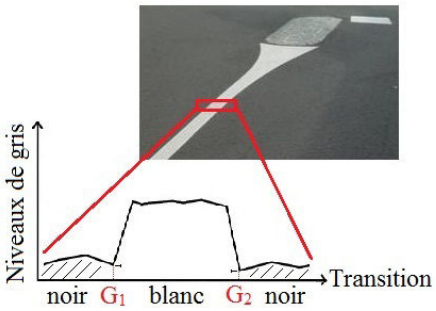



FIGURE *1.14 – Variation des valeurs de niveaux de gris pour un marquage au sol.*

Les auteurs [58] proposent une approche similaire qui consiste à rechercher une paire de gradients dans un voisinage de la largeur du marquage. D'après eux, la mise en correspondance pour chacun des marquages dans le repère véhicule et dans le repère image est donnée à l'aide d'une projection tenant en compte les paramètres de la caméra.

Une autre approche est proposée dans [59]. Elle consiste à caractériser l'ensemble des pics dans un marquage pour ensuite les filtrer en fonction de la position des transitions $(G_1, G_1)$ et de leurs niveaux de gris haut et bas. Les auteurs exploitent la différence de contraste entre la chaussée et le marquage.

Les méthodes basées sur des opérateurs par recherche de gradients symétriques

prennent en compte la réalité physique des marquages routiers. Cependant, elles sont très sensibles à l'état de la chaussée (la fissuration notamment) et à l'usure des marquages.

#### 1.4.6.3 Opérateur par patron

L'objectif de cette catégorie est de pouvoir détecter les marquages blancs appartenant à un fond sombre (route). Dans ce cadre, plusieurs approches ont été proposées. Parmi elles, nous citons les travaux de [38] qui proposent d'appliquer un filtre directionnel (Steerable filter). Ce filtre a pour particularité de pouvoir être exprimé comme une combinaison de plusieurs versions de lui-même suivant plusieurs orientations. L'extraction des primitives est réalisée en seuillant la différence entre la réponse du filtre dans l'orientation de la route et sa réponse minimale, dans le but de filtrer celles qui ne sont pas dans l'orientation de la route.

Dans [60], les auteurs proposent une méthode permettant d'appliquer une corrélation entre un voisinage et une fonction porte où la largeur de la porte est adaptée à la largeur attendue d'un marquage. L'avantage de cette méthode est sa rapidité en temps de calcul. Cependant, elle nécessite en contrepartie de savoir a priori la largeur des marquages que nous cherchons à détecter [40].

Les auteurs de [61,62] proposent la méthode MLT (Median Local Threshold). Cette méthode permet dans un premier temps d'appliquer un filtre médian sur l'image à traiter. Ensuite, elle permet de soustraire l'image ainsi filtrée avec l'image d'origine. D'après les auteurs, cette méthode semble efficace mais elle possède un inconvénient majeur. En effet, si la largeur de la fenêtre de traitement est inférieure à la moitié de la largeur du marquage, l'algorithme va inverser le résultat de détection.

D'après [63], cette méthode semble moins efficace que la méthode SLT (Symmetrical Local Threshold) introduite dans la section 1.4.5.3.b. Ainsi, pour avoir de meilleurs résultats, d'autres chercheurs ont proposé de fusionner plusieurs extracteurs de ce type [64]. D'après les auteurs, cette combinaison reste insuffisante malgré qu'elle ait légèrement amélioré les performances.

Dans cette section, nous avons présenté un ensemble non exhaustif de méthodes d'extraction des marquages routiers. Elles proposent non plus d'extraire les pixels de marquages au sol, mais d'extraire les primitives de plus haut niveau. Ces opérateurs travaillent sur des régions de l'image dans le but d'extraire des primitives contenant plusieurs marquages routiers. Par exemple, les auteurs dans [65] permettent d'extraire un ensemble des segments pour les agréger. Dans [66], les auteurs les classifient en appliquant des méthodes d'apprentissage sur des patchs de taille $9 \times 3$ pixels. D'autres méthodes qui sont considérées cependant minoritaires, permettent de classifier les primitives suivant le contraste et la distance entre les deux classes les plus fortes (chaussée, marquage) [32]. Notons que la majorité des extracteurs que nous avons étudiés ici ex-

ploitent indépendamment chaque marquage dans l'image. L'une des explications les plus pertinentes se base sur un traitement séquentiel défini par la variance des propriétés géométriques des marquages. En effet, un marquage au sol est fortement variant verticalement. L'ensemble des primitives extraits peut être utilisé pour détecter les marquages routiers.

## 1.5 Méthodes de détection de fissures

Ce paragraphe présente un état de l'art de la seconde catégorie de défauts de surface que l'on souhaite détecter sur la chaussée : **les fissures**. Les méthodes de traitement d'images permettent de segmenter une image (en général en niveaux de gris) et d'obtenir une image binaire comportant une zone "fissures" et une zone "texture d'image". D'après [67], la segmentation de fissures sur une image de chaussée est rendue difficile par la forte texture d'image. La texture d'image est liée à plusieurs facteurs qui se combinent : i) le matériau de chaussée est hétérogène car fait de granulats, dont le niveau de gris peut être proche de celui d'une fissure, ii) les interstices entre granulats qui sont comblés par un liant sombre (le bitume), peuvent être de même taille que celle des fissures apparaissant en surface chaussée, iii) enfin, le système d'éclairage peut provoquer des réflexions spéculaires sur certains granulats et renforcer ainsi la texture d'image. En pratique, la texture d'image peut réduire localement fortement le contraste entre la fissure (pixels sombres) et le fond de l'image (plus clair).

### 1.5.1 Hypothèses

Sur la chaussée, une fissure est une structure fine qui décrit une trajectoire contournant les granulats et pouvant se ramifier en plusieurs branches. Elle correspond à un ensemble de pixels sombres et connectés entre eux décrivant une trajectoire chaotique.

Dans les méthodes de traitement d'images, une fissure est définie à partir de ses caractéristiques photométrique (niveau de gris), géométrique (forme) et statistique (occurrence) [68]. Les caractéristiques photométriques (telle que la distribution des niveaux de gris ou histogramme) d'une image de chaussée est généralement unimodale (présentant 1 seul extrema), et ne permet pas de distinguer facilement les pixels fissures des pixels de texture.

Les méthodes de détection de fissures les plus performantes tentent de combiner ces trois caractéristiques dans le traitement. Ainsi, elles supposent que les pixels "fissure" sont plus sombres que la texture d'image, connectés entre eux en formant des éléments de taille variable et faiblement représentés dans l'image.

Dans certains cas cependant, cette description synthétique des fissures ne reflètent pas parfaitement la réalité. Par exemple, les fissures peuvent se présenter dans l'image comme une succession de petites portions de fissures non connectés entre elles, que l'œil

intègre en une seule naturellement.

### 1.5.2 Synoptique de la détection des fissures

Les méthodes de détection de fissures se déroulent généralement en quatre étapes : pré-traitement, segmentation, post-traitement, classification (figure 1.15).

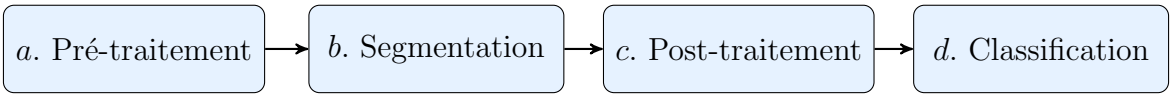



FIGURE *1.15 – Schéma classique pour détecter les fissures sur des images de chaussées.*

**a. Pré-traitement :** Le pré-traitement est une étape essentielle dans le domaine du traitement d'images. Il permet de corriger un certain nombre de défauts de la prise d'image (éclairage non uniforme par exemple), de réduire l'influence de la texture et de réhausser le contraste de la fissure [69]. Parmi les méthodes de pré-traitement, nous citons les méthodes de filtrage comme par exemple le filtrage passe-haut [70], le filtrage morphologique en niveaux de gris pour réduire l'impact de la texture par rapport à la fissure [70], le filtrage médian [71–73], le filtrage gaussien [70, 74]. l'utilisation de la moyenne locale permettant de corriger les défauts d'éclairage [75]. Le traitement combinant des seuillages local et global permet de réduire l'influence du bruit de fond. Nous mentionnons également le travail cité dans [76] qui applique la logique floue pour améliorer le contraste de la fissure dans l'image.

**b. Segmentation :** Cette étape permet de distinguer les pixels appartenant à une fissure des pixels appartenant à la texture. En général, elle consiste à regrouper les pixels pertinents pour construire la forme des défauts que nous cherchons à détecter dans l'image.

**c. Post-traitement :** Dans la majorité des méthodes de détection de fissures, nous obtenons une image résultante qui contient d'une part les pixels dans la fissure désirée (bonne détection) et d'autre part, des pixels erronés (fausse alarme) dans la texture d'image. Ainsi, l'objectif du post-traitement est de réduire les fausses détections dans l'image afin d'obtenir des résultats plus fiables. Parmi les nombreuses méthodes de post-traitement, nous citons par exemple celle qui a été proposée dans [74] et qui applique une segmentation locale autour du pixel étudié et les travaux cités dans [77] qui appliquent les outils de morphologie mathématique pour éliminer les petites régions qui ne correspondent pas à des vraies fissures.

**d. Classification :** Cette étape d'expertise consiste à caractériser le type de la fissure (longitudinale, transversale, faïençage, etc.) et ainsi qualifier la gravité de la fissure détectée (position, taille, etc.). Nous notons que la classification automatique des fissures reste un domaine qui n'a pas été exploré par les chercheurs.

Rappelons que l'objectif de la thèse consiste à déterminer et séparer les pixels de fissures des pixels de texture. C'est la raison pour laquelle, notre travail porte essentiellement sur l'étude des méthodes de segmentation de fissures.

### 1.5.3 Méthodes de segmentation de fissures existantes

Beaucoup de méthodes de détection de défauts existent dans la littérature, mais les applications au domaine des chaussées sont les plus restreintes [8, 75, 78–81]. Les premiers travaux sur l'automatisation de la détection des anomalies routières datent des années 90.

Nous présentons dans ce paragraphe les principales méthodes de traitement d'images pour la détection de dégradations de types fissures. Nous notons que les méthodes attachées aux systèmes d'imagerie commerciales sont en général non documentées.

D'après [68, 69, 82], la plupart des méthodes de segmentation existantes sont composées de deux étapes : la binarisation et la connexion. La binarisation permet d'extraire les pixels de fissuration dans l'image. La connexion consiste à relier les pixels binarisés et ainsi trouver la forme entière de la fissure. Dans ce cadre, nous présentons dans la section suivante les principales approches mises en œuvre lors de l'étape de segmentation.

#### 1.5.3.1 Approches par seuillage

Les méthodes par seuillage sont des méthodes de segmentation conventionnelles dans le domaine du traitement d'images. Elles sont fondées sur la caractéristique de **luminance** des fissures, sont simples à implémenter et peu coûteuses en temps de calcul [69].

**a. Seuillage fixe**

L'ensemble de ces méthodes fait l'hypothèse évoquée au paragraphe 1.5.1, que le niveau de gris des pixels fissure est plus faible que celui des pixels appartenant à la texture. La première idée qui vient à l'esprit pour segmenter les fissures est d'appliquer des techniques de seuillage fixe, basées sur l'analyse de l'histogramme de toute l'image, pour séparer les pixels de l'image en deux classes, e.g., [78, 80, 83, 84]. Par exemple, la technique citée dans [78] est limitée en performance, car l'histogramme des niveaux de gris d'une image de chaussée est unimodal et ne fait pas apparaître clairement les deux classes de pixels. D'autres chercheurs ont proposé de contourner cette difficulté en utilisant une méthode de seuillage local [73, 85–87]. Une autre alternative est d'introduire une méthode de seuillage fixe à deux niveaux [87], en considérant que l'image de chaussée est constituée de trois groupes de pixels, à savoir les pixels noirs (fissure), les pixels sombres (fissure ou fond) et les pixels clairs (fond). La difficulté de toutes ces méthodes concerne le choix de(s) la valeur(s) du seuil, qui dépend fortement de la

texture d'image et du contraste avec la fissure.

**b. Seuillage en 3 classes par logique floue et entropie**

Les travaux de [76,81] reprennent l'idée d'une segmentation des pixels en 3 classes, qui a été introduite au paragraphe précédent [87]. Ils proposent d'améliorer la classification des pixels sombres (qui peuvent appartenir à la texture ou à la fissure) en faisant appel à la théorie de la logique floue.

L'ensemble des niveaux de gris des pixels de l'image à traiter $\Omega$ est caractérisé par une fonction d'appartenance $\mu$ qui associe un élément $a$ de $\Omega$ avec un nombre réel $\mu(a)$ dans l'intervalle $[0,1]$ et ainsi quantifie le degré d'appartenance de l'élément $a$ soit à une zone texture ou à une zone fissure. D'après [67], il existe plusieurs fonctions permettant de calculer le degré d'appartenance. La fonction utilisée pour la détection des fissures est la fonction $S$ de Zadeh. D'après [76,81], cette fonction est définie par l'équation 1.17 comme suit :

$$\mu(a) = S(a,u,v,w) = \begin{cases} 0 & , \quad a \leq u \\ \frac{(a-u)^2}{(v-u)(w-u)} & , \quad u \leq a \leq v \\ 1 - \frac{(a-u)^2}{(v-u)(w-u)} & , \quad v \leq a \leq w \\ 1 & , \quad a \geq w \end{cases} \tag{1.17}$$

où $a$ est la valeur du niveau de gris du pixel, $u$, $v$ et $w$ sont les paramètres qui déterminent la forme de la fonction $S$. La figure 1.16 illustre la courbe représentative de la fonction $S$.

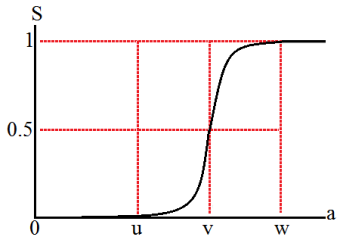



FIGURE *1.16 – Courbe représentative illustrant la fonction S de Zadeh.*

La deuxième partie consiste à seuiller cette nouvelle modélisation de l'image. Le seuillage consiste à déterminer les valeurs des trois paramètres $u$, $v$ et $w$. D'après [82], ces valeurs sont obtenues en maximisant l'entropie définie comme suit :

$$E(a) = \frac{-1}{ln2} \sum_{a \in \Omega} [\mu(a) ln(\mu(a)) + [1 - \mu(a)] ln(1 - \mu(a))] \times h(a) \tag{1.18}$$

Contrairement aux méthodes de seuillage conventionnel citées dans le paragraphe précédent, l'approche par logique floue permet d'obtenir automatiquement le seuil par

un modèle qui s'adapte à la distribution des pixels sur des images de chaussées, sans intervention humaine. Cependant, selon [88], les performances de cette approche restent limitées du fait que l'image est traitée en se basant uniquement sur la luminosité des pixels et d'autre part, parce qu'elle est très sensible à la texture d'image.

#### 1.5.3.2 Approches par morphologie mathématique

Les outils de morphologie mathématique, présentés dans la section 1.4.5.2, sont assez polyvalents pour être appliqué à différentes étapes de la détection de fissures. Les auteurs dans [89] utilisent la morphologie mathématique pour détecter directement les pixels fissure. Dans [77, 90], ils l'utilisent en post-traitement de la segmentation pour connecter les portions de fissures détectées, et pour éliminer les petites fissures isolées.

Selon [89], la morphologie mathématique prend en compte plusieurs caractéristiques de la fissure pour la segmentation. Les auteurs supposent qu'une fissure est une succession des points-selles (saddle points en anglais) [91] sombres possédant un attribut directionnel. Pour ce faire, une transformation *top-hat* est appliquée. Cette transformation permet d'extraire les pics dont l'épaisseur est inférieure à l'élément structurant. Ensuite, un seuillage fixe est réalisé pour sélectionner les pixels les plus sombres dans l'image. Enfin, à partir des images de pixels noirs et de points-selles (pixels communs entre les deux), une recherche de linéarité dans les quatre orientations 0°, 45°, 90°et 135°permet de connecter les segments de fissures. La figure 1.17 illustre un exemple d'une segmentation réalisée en appliquant ces outils morphologiques.

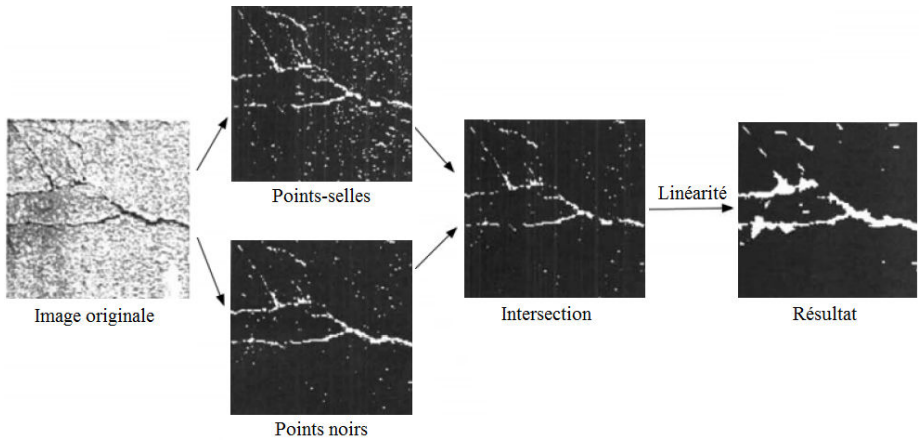



FIGURE *1.17 – Exemple d'une segmentation suivant une approche morphologique.*

Les performances de cette méthode dépendent principalement de la taille et de la forme de l'élément structurant ainsi que du choix du seuil à utiliser pour déterminer les pixels noirs dans l'image.

#### 1.5.3.3 Approches par apprentissage

Dans le contexte de la détection de fissures, les méthodes par apprentissage sont en général des méthodes supervisées. Elles s'appuient sur une base d'images de référence, pour apprendre à un algorithme à classifier correctement les données en un nombre prédéterminé de classes. Il peut s'agir de classifier les images en deux classes (défauts ou sans défauts, e.g., [92]) ou détecter des zones fissures/non fissures dans l'image [93].

Les premières méthodes de détection des fissures par apprentissage s'appuient sur des réseaux des neurones, par exemple [94]. Dans [93, 95], les auteurs utilisent deux réseaux de neurones : un premier réseau joue le rôle d'un classificateur entre les images avec fissures et les images sans fissures, et un second réseau détermine le type de la fissure (longitudinale, transversale, etc.).

Les méthodes par apprentissage se sont diversifiées depuis quelques années dans le domaine de l'auscultation des chaussées. Nous pouvons citer les méthodes SVM [96,97], les arbres de décision du type random forest [98]. Ce type d'algorithmes supervisés nécessite une grande base d'images de référence (par exemple 190 dans [99] et 500 dans [100]) dans l'étape d'apprentissage. Les méthodes à base d'apprentissage profond ou deep learning sont testées sur des bases d'images de plus en plus grande, e.g., [101].

#### 1.5.3.4 Approche par filtre adaptatif

Les auteurs de [102] supposent que les fissures sur la chaussée apparaissent sous forme de segments orientés élémentaires plus sombres que la texture et ayant une faible épaisseur. Pour une direction $\theta$ ($0^\circ$, $45^\circ$, $90^\circ$et $135^\circ$), le signal de la fissure dans l'image est modélisé comme étant soit un signal rectangulaire (équation 1.20), soit une gaussienne (équation 1.21), noyée dans du bruit additif selon le modèle de l'équation (équation 1.19).

$$f^\theta(x) = s^\theta(x) + b^\theta(x) \tag{1.19}$$

avec :

$$s^\theta(x) : \begin{cases} -A & si \quad x \in [-\frac{L}{2}, \frac{L}{2}] \\ 0 & si \quad non \end{cases} \tag{1.20}$$

où $A$ est l'amplitude du signal et $L$ est un paramètre représentant l'épaisseur de la fissure.

$$s^\theta(x) = -Ae^{-\frac{1}{2}(\frac{x}{\sigma})^2} \tag{1.21}$$

où $\sigma$ est l'écart-type de la fonction gaussienne qui détermine la largeur de la fissure.

Selon ce modèle d'observation, la détection de fissure revient à chercher des signaux déterministes (fissures) dans un bruit (texture). Le problème est similaire à la recherche

d'un filtre adapté au signal déterministe attendu. Le filtre adapté permet d'obtenir une première segmentation de l'image, qui est ensuite raffinée par une segmentation markovienne. Ce type de méthode a été proposé par [102], et repris ensuite par d'autres chercheurs du même organisme dans [88, 103].

#### 1.5.3.5 Approche par transformation en ondelettes continue

La transformée en ondelettes continue (TOC) permet de réaliser une analyse multi-échelle des images de chaussées. Elle permet de passer de la représentation d'une image numérique à une représentation espace-échelle [104, 105]. Cette transformation fournit de la redondance d'informations puisque le signal d'origine $2D$ (image) est transformé en une série de signaux $2D$, la série étant indexée par le paramètre d'échelle. Cependant, une des difficultés de cette méthode concerne le choix de l'ondelette-mère [88].

Le choix proposé dans [67,68,103,106] consiste à calculer une ondelette-mère adaptée à chaque texture de chaussée, selon le principe de filtrage adapté. Dans [68, 103], les ondelette-mères aux différentes échelles correspondent aux réponses impulsionnelles $h$ des filtres adaptés qui permettent une meillleure distinction entre le signal déterministe (fissure) et le bruit (texture).

Tout d'abord, une TOC est réalisée sur l'image définie par l'équation 1.19 afin de rehausser le contraste et ainsi améliorer le rapport signal (fissure) sur bruit (fond). Ensuite, des cartes de coefficients d'ondelettes sont constituées sur les différentes échelles. La binarisation est réalisée tout de suite soit par la recherche des minima-locaux dans des cartes de coefficients [107], soit par l'initialisation des champs de Markov [67,68,88].

#### 1.5.3.6 Approche par modélisation markovienne

D'après [67], le résultat de la segmentation obtenue à partir des cartes de coefficients de TOC (section 1.5.3.5) est très sensible au bruit, et produit un certain nombre de fausses alarmes. C'est la raison pour laquelle une étape supplémentaire de raffinement du résultat de la méthode du paragraphe précédent est introduite. Dans ce cadre, la modélisation par champs de Markov est utilisée comme outil de la segmentation. Elle permet de tenir compte de relations de connexité spatiale des pixels de l'image dans un voisinage 3, et par ce biais, de prendre en compte les caractéristiques géométriques de la fissure.

En pratique, la segmentation du paragraphe précédent est mise à jour en prenant en compte non plus les pixels seuls mais une grille $3 \times 3$ de pixels pouvant contenir ou non des fissures, avec quatre configurations possibles suivant les quatres directions $0^{\circ}$, $45^{\circ}$, $90^{\circ}$et $135^{\circ}$(cf. figure 1.18). La mise à jour prend en compte les 8 sites voisins du site étudié.

La méthode a été étendue dans [108] à 16 configurations de sites selon la figure 1.18.

Cette configuration inclut des sites décentrés dans le ROI $3 \times 3$ pixels.

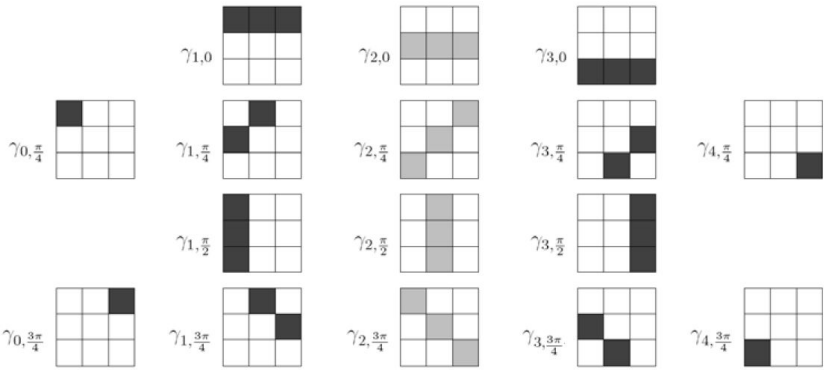



FIGURE *1.18 – Configurations pour modéliser les relations de connexités spatiales entre pixels.*

#### 1.5.3.7 Approche heuristique

Cette méthode a été développée à partir de 2006 par l'IFSTTAR en collaboration avec le Cerema. Elle est basée sur la détection de valeurs significatives du gradient de l'image. Elle a été utilisée comme un outil de support à l'outil Visiodec [109] lors de quelques campagnes annuelles de qualification réseau national français. Une présentation et une évaluation de cette méthode figure dans [110].

Tout d'abord, un pré-traitement est réalisé permettant de rehausser le contraste et ainsi rendre plus fiable le calcul du gradient de l'image. Afin de détecter les zones sombres de l'image, cette méthode ne prend pas en compte les pixels dont le niveau de gris est supérieur à 210. De plus, le traitement se focalise sur la zone centrale de la chaussée et exclue les deux zones latérales où sont situés les marquages routiers. De plus, afin d'atténuer le bruit, cette méthode est basée sur deux étapes principales de nettoyage : filtre médian - morphologie mathématique.

Ensuite, une analyse locale est effectuée pour segmenter l'image dans des régions d'intérêt, Regions-Of-Interest (ROI), de taille $30 \times 30$ pixels. En se basant sur les caractéristiques géométriques, chaque objet détecté est analysé selon sa forme, sa longueur ainsi que sa largeur en prenant des valeurs de seuils spécifiques définies comme suit : 100 pixels pour la longueur et 20 pixels pour la largeur. Dans le but d'améliorer la fiabilité de la segmentation, un processus est effectué successivement sur deux plus grandes ROI dans l'image, soit $50 \times 50$ et $80 \times 80$ pixels. Ainsi, les résultats obtenus sur chacune des ROI sont fusionnés afin d'obtenir l'image segmentée.

Enfin, une étape de post-traitement est appliquée afin d'améliorer la connexité entre les parties détectées. Pour ce faire, une opération de fermeture morphologique est réalisée.

L'inconvénient de cette méthode est qu'elle dégrade la résolution initiale de l'image. En outre, la succession d'étapes et de processus différents rendent difficile l'ajustement des paramètres afin d'avoir une détection performante sur un grand nombre d'images.

#### 1.5.3.8 Approches par chemins minimaux

La notion de chemin minimal a été introduite pour résoudre des problèmes d'optimisation (par exemple, le problème du voyageur de commerce qui cherche à minimiser le temps de trajet de son parcours entre différentes villes). En traitement d'images, cette notion se prête à la segmentation de formes ou la détection de contours d'un objet. Elle a été adaptée à la détection de fissures dans des images à partir de 2003 environ [111–116].

L'approche par chemins minimaux permet de rechercher la connexité entre les pixels d'un même type (fissures) dans un voisinage de taille plus importante que pour les méthodes précédentes, e.g., méthode de Markov. Cette approche a pris plusieurs formes selon la définition de la fonction de coût et le choix de l'algorithme qui résout le problème de minimisation du coût.

**a. Fonctions de coût**

Dans de nombreux domaines, il est nécessaire d'estimer le trajet le plus court entre deux points en respectant certaines contraintes [117]. Prenons l'exemple du voyageur de commerce introduit dans la section 1.5.3.8, une des principales contraintes qu'il faut respecter est **le coût**. En effet, un voyageur cherche toujours à atteindre sa destination avec un minimum de dépense (soit un faible coût).

Ce paragraphe présente les deux fonctions de coût qui sont utilisées dans la littérature pour identifier la fissure sur la chaussée par une méthode de chemin minimal. En considérant que les fissures sur la chaussée sont des filaments sombres dans l'image, la fonction de coût doit atteindre une valeur minimale pour les pixels fissures et une valeur plus élevée pour les pixels appartenant à la texture d'image.

La détection de contours dans une image fait intervenir une fonction de coût assez générale (ou fonction d'énergie), qui tient compte de la forme de la courbe (gradient, rayon de courbure) selon l'équation 1.22 suivante [118] :

$$E(f) = \int_0^m [P_1||f'(x)||^2 + P_2||f''(x)||^2 + P_3(f(x))]\,\mathrm{d}x \tag{1.22}$$

où $f(x)$ est une courbe $2D$ définie dans l'intervalle $[0, m]$ avec $m$ la longueur de la courbe. Les deux premiers termes représentent le terme de force interne qui assure l'élasticité et la flexibilité de la courbe avec $P_1$ et $P_2$ les poids associés à chacun de ces deux termes. $P_3$ représente le terme de force externe (terme d'attache aux données).

Selon les auteurs, cette fonction est moins adaptée pour segmenter les images de chaussées du fait de la forme chaotique des fissures et de l'aspect des images de chaussées à segmenter (résolution spatiale et quantification des niveaux de gris limitées). La fonction de coût est basée uniquement sur les caractéristiques photométriques de l'image [119] ; aucune régularité n'est imposée à la courbe à déterminer. La fonction de coût inclut uniquement le terme d'attache aux données, qui s'exprime comme la somme des valeurs de niveaux de gris, selon :

$$C(ch_{ab}) = \sum_{p=a}^{b} I(x_p) \tag{1.23}$$

où $ch_{ab}$ représente un chemin entre un point source $x_a$ et un point destination $x_b$ dans une image $I$ en niveaux de gris, et $x_p$ est un pixel du chemin. Un exemple de calcul des chemins en minimisant les deux équations 1.22 et 1.23 est présenté dans la figure 1.19.

### b. Algorithmes d'optimisation

Deux algorithmes sont beaucoup cités dans la littérature pour la segmentation d'images en niveaux de gris. L'algorithme du Fast Marching est ainsi dédié à la minimisation de l'équation 1.22. L'algorithme de Dijkstra [120] est l'un des algorithmes le plus connu pour minimiser la fonction définie par l'équation 1.23. Son principe sera détaillé au chapitre 3. Il a donné lieu à de nombreuses variantes, comme par exemple, l'algorithme A* [121] qui minimise plus rapidement la fonction de l'équation 1.23.

De fait, chaque fonction de coût produit une segmentation différente comme l'illustre la figure 1.19. Pour l'application visée par cette thèse et la qualité des images dont nous disposons, la segmentation obtenue par l'algorithme de Fast Marching [122] parait moins précise que celle obtenue avec l'algorithme de Dijkstra.

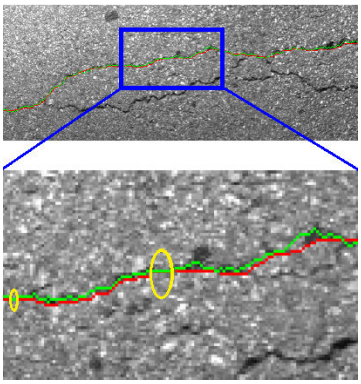


FIGURE *1.19 – Comparaison entre un chemin minimal calculé par l'algorithme de Dijkstra (en vert) et par l'algorithme de Fast Marching (en rouge) – Les différences entre les deux segmentations sont indiquées dans les zones en jaune.*

### c. Adaptation à la segmentation automatique de fissures sur chaussées

Les algorithmes précédents recherchent un chemin minimal entre un pixel source et un pixel destination de l'image. L'adaptation de ce principe à la segmentation automatique de fissures repose sur un choix judicieux et automatique de ces pixels sources dans l'image. Dans ce paragraphe, nous indiquons trois méthodes existantes de détection automatique de fissures sur chaussées basées sur le principe de recherche de chemin minimal.

#### i. Contours géodésique avec détection automatique de points d'intérêt (GC-POI)

Dans [112], l'auteur propose une approche automatique, nommée *Geodesic Contour-based method initialized by Points-Of-Interest*, et constituée de deux étapes. La première étape sélectionne les pixels d'intérêt de manière automatique dans l'image à traiter. Les points d'intérêts sont définis par un niveau faible d'auto-corrélation et un niveau de gris faible (fissure). La deuxième étape consiste à calculer les chemins minimaux entre tous les points d'intérêt sélectionnés, en appliquant une méthode de contours géodésiques de type Fast Marching.

#### ii. Free Form Anisotropy (FFA)

Les auteurs dans [82, 114] calculent les chemins minimaux à partir de chaque pixel de l'image dans un bloc de 30 pixels de côté et dans quatre orientations privilégiées (0°, 45°, 90°et 135°). Ensuite, un facteur d'anisotropie par pixel est calculé à partir des chemins obtenus à l'étape précédente. Ce facteur est établi à partir de la valeur du chemin de coût minimal et celui de coût maximal. Une forte différence entre ces deux valeurs indique la forte probabilité que le pixel d'origine appartienne à une fissure. Ainsi, les pixels fissures sont sélectionnés en appliquant une méthode de seuillage par hystérésis de la carte de tous les coefficients d'anisotropie.

Selon les auteurs, cette approche permet de segmenter les fissures fines et discontinues. Elle semble être l'une des méthodes les plus intéressantes et pertinentes pour rechercher les chemins minimaux. En contrepartie, le principe de la méthode limite la capacité de détection des fissures qui présentent des variations rapides d'orientation. De plus, la méthode surestime l'épaisseur de la fissure et produit ainsi un large taux de faux positifs.

#### iii. Sélection des chemins de coût minimal : Minimal Path Selection (MPS)

Cette section présente brièvement le principe de la méthode MPS (Minimal Path Selection) développée à partir de 2011 par l'IFSTTAR, l'IRIT et l'IRCCyN, pour la détection automatique des fissures dans des images de chaussées [117, 123]. Dans cette

méthode, le calcul des chemins minimaux est réalisé en minimisant la fonction de coût définie par l'expression 1.23. L'algorithme de Dijkstra [120] est utilisé pour le calcul des chemins minimaux entre les points amorces.

Cette approche est composée de deux phases. La première phase permet de segmenter le squelette de fissuration dans l'image, à partir d'un choix judicieux de points amorces et des statistiques de chemins minimaux élémentaires entre points amorces. La deuxième phase permet d'une part de connecter les morceaux de fissures élémentaires, d'autre part, de raffiner le squelette de segmentation et finalement, d'estimer l'épaisseur de fissure.

Comparée aux deux premières méthodes, la méthode MPS produit un résultat de segmentation de meilleure qualité, avec un plus faible taux de faux négatifs (non détection) et de faux positifs (fausse alarme). Elle est capable de détecter les fissures chaotiques sans contraintes d'orientation, en comparaison de FFA. En contrepartie, MPS souffre d'un temps d'exécution plus important, et présente des difficultés à détecter les petits bouts de fissurations discontinues.

#### 1.5.3.9 Discussion

Nous avons présenté au long du paragraphe 1.5.3 l'état de l'art des méthodes de segmentation automatique de fissures dans des images de chaussées. Une analyse comparative des méthodes présentées est disponible dans plusieurs publications [98,108,117,123]. La plupart de ces méthodes présentent des difficultés pour segmenter des fissures chaotiques et produisent dans certaines cas une grande quantité de fausses alarmes (pixels faux positifs) dispersées dans toute l'image. Selon [101], ces limitations sont principalement dues à la forte texture de l'image de chaussée (hétérogénéité de l'image) et à la complexité de la structure de la chaussée.

Dans ce contexte, les travaux de [123] ont notamment montré le grand intérêt des méthodes à base de chemins minimaux. Parmi celles-ci, la méthode MPS [123] a montré de meilleurs résultats de segmentation, avec un plus faible taux de faux négatifs (non détection) et de faux positifs (fausse alarme). MPS réalise une recherche sans contraintes de forme, direction, longueur et épaisseur des fissures.

De fait, la méthode MPS a été sélectionnée dans la suite de ce travail. Nous la présentons ainsi en détails au chapitre 3. Le paragraphe 1.5.3.8.c.iii permet d'envisager pour MPS des perspectives d'améliorations possibles. Aussi, le chapitre 4 présente des optimisations de MPS, afin d'améliorer la qualité de la détection, de diminuer le taux de fausse alarme d'un facteur supplémentaire, et enfin, de réduire le temps de calcul.

## 1.6 Systèmes d'imagerie utilisés

Aux chapitres 3 à 5, nous testons les algorithmes de segmentation automatique sur des images de chaussées de trois origines différentes. Nous présentons dans cette partie les trois appareils d'imagerie à grand rendement dont ces images sont issues. Ces appareils correspondent aux deux types de systèmes introduits au paragraphe 1.3.3.2 : deux systèmes spécialisés qui collectent les images de chaussées à incidence verticale, et un système panoramique plus polyvalent qui collectent des images de chaussée à incidence oblique et enregistre également le paysage routier.

### 1.6.1 Systèmes spécialisés d'imagerie de la route

#### 1.6.1.1 Appareil Aigle-RN

Dans le cadre de ses activités de soutien aux politiques publiques, le Cerema a développé le système Aigle-RN (Appareil d'Inspection Globale de l'État du Réseau National) [4,5] pour assurer la gestion de l'entretien des routes nationales (figure 1.20).

FIGURE *1.20 – Système Aigle-RN.*

Le système Aigle-RN est constitué à partir d'un système d'auscultation Schniering. Les images de chaussée sont prises à incidence verticale, *downward facing images*, c'est-à-dire que l'axe optique de la caméra est perpendiculaire au plan de la chaussée. Comme le montre la figure 1.20, ce système est composé de trois caméras monochromes fixées à l'arrière du véhicule avec un axe optique perpendiculaire au sol pilotées chacune par un PC. Le système reconstitue une image de chaussée de 3.5 $m$ de largeur et 1 $m$ de longueur à partir de chacune des images des 3 caméras. Comme il s'agit d'un appareil industriel, nous n'avons pas accès aux caractéristiques précises ni aux réglages des capteurs.

Ce système permet d'obtenir des images couleur de taille $1\,900 \times 924$ pixels (la taille d'un pixel est d'environ $2 \times 2$ $mm^2$). L'image est codée sur 8 bits et contient donc 256 niveaux de gris.

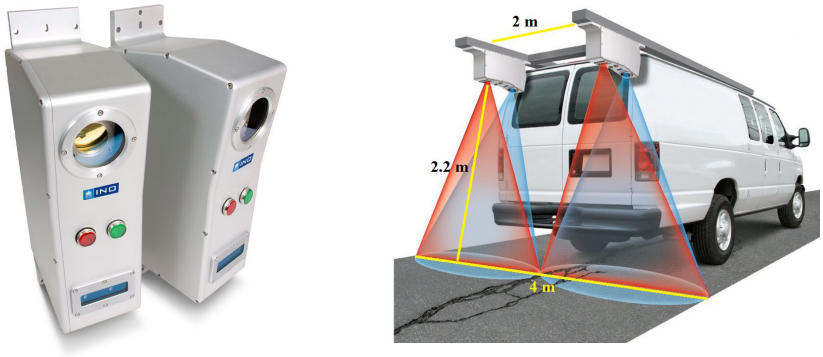
L'éclairage de la chaussée est assuré par une rampe de lampes stroboscopiques, qui permet de faire fonctionner l'appareil de jour comme de nuit. Le système d'éclairage ne délivre pas un éclairage uniforme sur toute la largeur de chaussée. Cet inconvénient contribue à renforcer la texture d'image, qui est déjà naturellement forte du fait de la granularité du matériau et de sa rugosité de surface.

Nous notons que le système Aigle-RN a été récemment remplacé par le système Aigle-RN-3D, conçu à partir du système LCMS, dont le principe est présenté au paragraphe 1.6.1.2 suivant.

#### 1.6.1.2 Système LCMS

Le système LCMS (Laser Crack Measurement System) [124] représenté à la figure 1.21 a été développé depuis dix ans par la société canadienne Pavemetrics, en collaboration avec l'institut National d'Optique (INO) du Canada. Il fait suite au système LRIS, qui délivrait la luminance de la chaussée uniquement.

Le système LCMS équipe de nombreux appareils d'auscultation à travers le monde. En France en particulier, le LCMS équipe les appareils d'auscultation du Cerema (Aigle-3D) et de 4 sociétés (Diagway, Ginger-BTP, Technologies Nouvelles et Vectra).




*(a)* *(b)*

FIGURE *1.21 – Système LCMS.*

Ce système fournit conjointement la luminance et le relief de la surface de la chaussée à vitesse du trafic. Le capteur contient des lasers de forte puissance, dans le proche infrarouge, qui projettent une ligne sur la chaussée. L'image de cette ligne, enregistrée par

des caméras haute vitesse, est déformée par le relief de la surface. Les valeurs d'altitude de la chaussée sont déduites par triangulation (figure 1.22). L'éclairage laser permet une auscultation de jour comme de nuit.

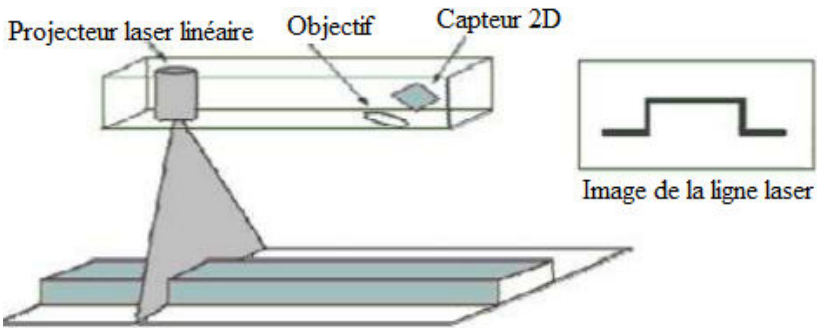



FIGURE *1.22 – Principe de triangulation laser.*

Le système LCMS fournit à la fois des images $2D$ et $3D$ de la route de 4 m de large sur 5 à 10 m de longueur. La résolution latérale de l'image est de 1 mm, la résolution longitudinale varie de 1 mm à 5 mm selon la vitesse, et enfin la résolution verticale est de 0.5 mm. De plus, l'image est codée sur 10 bits.

### 1.6.2 Système d'imagerie $VIAPIX$®

$VIAPIX$® [125] est un équipement opérationnel de relevé de terrain qui produit des données routières géo-référencées. Il permet de géolocaliser des objets du paysage routier (parking, signalisation verticale, signalisation horizontale, etc.) et de déterminer leur dimension (largeur, hauteur, etc.) dans l'image. Cette solution est composée d'un module d'acquisition et d'un puissant logiciel de post-traitement.

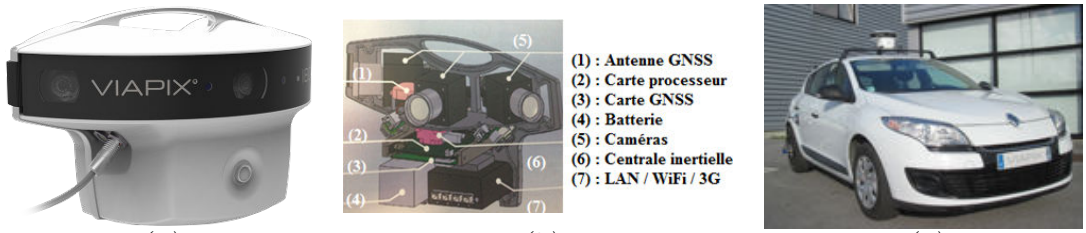
Nous présentons dans ce paragraphe uniquement la partie optique du module d'acquisition, ainsi que les caractéristiques des images que nous avons collectées à incidence oblique. Le module de post-traitement est présenté en annexe B.

Le système d'acquisition de $VIAPIX$® est constitué d'un système de capture d'images panoramiques et d'un système de géolocalisation hybride temps réel. Une fois installé sur le toit du véhicule, ce module capture et enregistre des images à intervalle de distance régulier. L'image panoramique du paysage routier est réalisée par un ensemble de trois caméras couleurs couplées à une carte processeur réalisant les tâches de pré-traitement, de compression et d'enregistrement figure 1.23. Les images couleurs sont codées sur 8 bits, soit 256 valeurs d'amplitude.

L'ensemble des caméras permet de créer une image panoramique à 180°, dont la taille est de 6 Mégapixels. Toutes les données sont géolocalisées dans le référentiel terrestre et PK-métrées dans le référentiel routier. La caméra centrale du dispositif est utilisée pour

acquérir l'image d'une bande de surface de chaussée en avant du véhicule. En pratique, cette bande fait 1 à 2 mètres de long et environ 5 mètres de large. Les dimensions latérale et longitudinale du pixel sur la chaussée sont en moyenne de $4.4 \times 4.4\ mm^2$. Du fait de l'effet de perspective (incidence oblique), la dimension du pixel sur la bande de chaussée varie un peu avec la profondeur de champ.

Pour faciliter les processus ultérieurs qui traitent l'étude de dégradations de la chaussée, la transformée en perspective inverse (IPM) détaillée au 1.4.5.1, permet de générer une image en vue de dessus, comparable à celle obtenue avec les systèmes spécialisés présentés au paragraphe précédent. Après la transformée IPM et le processus d'interpolation sous-jacent, l'image de surface de chaussée présente un échantillonnage spatial régulier.




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

FIGURE *1.23 – VIAPIX® acquisition – (a) Simulation 3D - (b) Architecture - (c) Photographie du module acquisition installé sur le toit d'un véhicule.*

## 1.7 Conclusion

Dans ce chapitre, nous avons présenté la problématique du travail de thèse, qui consiste à détecter deux types de dégradations en surface de chaussée : les dégradations de marquages horizontaux et les fissures. Les techniques de relevés des dégradations en surface de chaussées ont beaucoup évolué au cours du temps. Les techniques de relevé automatiques ont succédé au relevé manuel, qui n'est plus adapté aux exigences actuelles de fiabilité, de sécurité et de rapidité. Elles s'appuient d'une part sur des appareils d'imagerie à grand rendement, qui collectent des images de la chaussée à vitesse de trafic, et d'autre part, sur des algorithmes de traitement d'images dont l'objectif est de segmenter automatiquement les défauts de surface dans les images.

Les méthodes existantes de segmentation automatique des marquages routiers et des fissures ont fait l'objet d'un état de l'art dans ce chapitre. Les méthodes conventionnelles sont limitées par la forte texture des images de chaussée. En comparaison, les méthodes qui exploitent conjointement les caractéristiques photométriques et géométriques des objets à segmenter sont les plus pertinentes. Pour les fissures, la bibliographie a notamment montré l'intérêt des méthodes à base de chemins minimaux, et parmi celles-ci, la méthode MPS [123], qui réalise une segmentation sans contraintes de forme, direction,

longueur et épaisseur des fissures.

Dans la suite de ce document, le chapitre 2 présente notre contribution à la quantification des défauts sur la signalisation horizontale. Les chapitres 3 à 5 sont consacrés à la segmentation automatique de fissures. Ainsi, le chapitre 3 présente le principe de la méthode sélectionnée pour ce faire, MPS, tandis que le chapitre 4 détaille les améliorations proposées pour optimiser ses performances. Enfin, le chapitre 5 décrit le nouvel algorithme ADFD proposé (à base de chemins minimaux) et ses performances par rapport aux algorithmes cités dans les chapitres 3 et 4. Toutes les contributions sont testées sur des images de chaussées issues des 3 systèmes d'acquisition différents, présentés dans ce chapitre.

# Détérioration de la signalisation horizontale

# Chapitre 2

## Extraction des dégradations de la signalisation horizontale

**Sommaire**

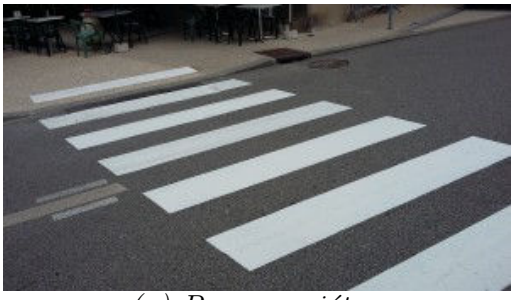

2.1 Introduction . . . . . **49**
2.2 Approche proposée . . . . . **50**
2.2.1 Détection de la signalisation horizontale . . . . . 50
2.2.1.1 Solution proposée . . . . . 52
2.2.1.2 Détection de la route . . . . . 52
2.2.1.3 Segmentation couleur . . . . . 66
2.2.2 Identification des marquages détectés . . . . . 68
2.2.2.1 Méthode de reconnaissance géométrique . . . . . 68
2.2.2.2 Méthode de reconnaissance optique . . . . . 71
2.2.3 Géolocalisation du marquage identifié . . . . . 75
2.2.4 Quantification de l'usure du marquage routier . . . . . 76
2.3 Évaluation des performances de la méthode sur des images $VIAPIX^{®}$ **78**
2.4 Conclusion . . . . . **82**

## 2.1 Introduction

Dans ce chapitre, nous présentons l'approche que nous avons proposée pour la détection automatique des dégradations des marquages en surface de chaussée. L'objectif principal est d'étudier l'état global de la signalisation horizontale sur la chaussée afin de maintenir un niveau de visibilité et de sécurité suffisant.

Pour ce faire, nous identifions, dans un premier temps, le type du marquage routier. Ensuite, nous caractérisons son état d'usure sur la chaussée. Dans ce cadre, nous avons testé notre approche sur des images fournies par le module $VIAPIX^{®}$ [125] présenté dans la section 1.6.2 du chapitre 1.

Notons que dans ce travail, nous nous concentrons sur les trois types de marquages au sol, à savoir **le passage piéton**, **le mot"BUS"**, et **le pictogramme-vélo** (voir figure 2.1).



*(a) Passage piéton*

*(b) Pictogramme "BUS"*

*(c) Pictogramme-vélo*

FIGURE *2.1 – Les trois catégories des marques à traiter dans le cadre de cette thèse.*

## 2.2 Approche proposée

Dans cette section, nous présentons le principe de l'extraction des dégradations de la signalisation horizontale sur la chaussée. Notre algorithme, illustré dans la figure 2.2, comporte quatre phases principales. Il est basé sur l'analyse des caractéristiques photométriques, colorimétriques et géométriques des marquages routiers. La première phase consiste à segmenter l'image d'origine pour détecter tous les objets ayant une similitude avec un marquage routier, la deuxième phase sert à identifier les objets détectés dans l'image, ensuite la troisième phase sert à géolocaliser tous les objets identifiés en calculant leurs positions GPS (Global Positioning System). Enfin, la dernière phase permet d'étudier l'état d'usure de chacun des objets géolocalisés.

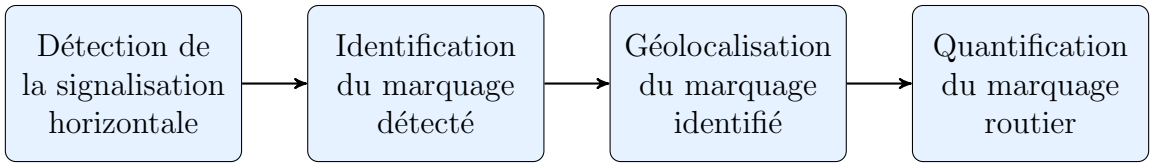



FIGURE *2.2 – Synoptique général de l'approche proposée.*

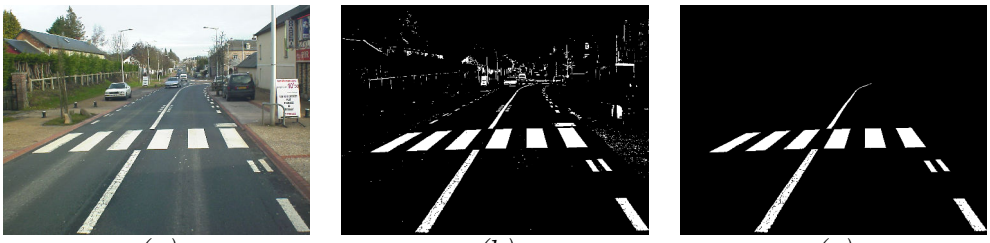
La figure 2.3 illustre un exemple de dégradation de la signalisation horizontale (cas d'un passage piéton). En général, la signalisation horizontale est détériorée soit à cause des mauvaises conditions climatiques, soit en raison du trafic routier.

FIGURE *2.3 – Exemple d'une image $VIAPIX^{®}$ de taille* $1216 \times 1600$ *pixels illustrant un passage piéton détérioré.*

### 2.2.1 Détection de la signalisation horizontale

La première étape de l'approche proposée consiste à détecter le marquage au sol afin de déterminer son niveau de dégradation. Pour y arriver, nous nous sommes basés sur les caractéristiques spécifiques du marquage routier. En effet, la réglementation française exige que tous les marquages au sol soient de couleur blanche avec un fort contraste par rapport à la route. Nous avons effectué ainsi, une segmentation couleur sur les images issues du module $VIAPIX^{®}$ [125] afin de pouvoir détecter tous les objets ayant une couleur blanche sur la route. Nous notons que le seuil adaptatif utilisé

pour la segmentation est défini à partir de la valeur de la moyenne $\mu$ de l'image, de telle sorte que si la valeur d'un pixel $p$ est supérieure ou égale à $\mu$, le pixel est considéré comme étant un pixel appartenant à un objet lumineux sur la route. Un exemple du résultat obtenu avec cette segmentation couleur est illustré dans la figure 2.4.

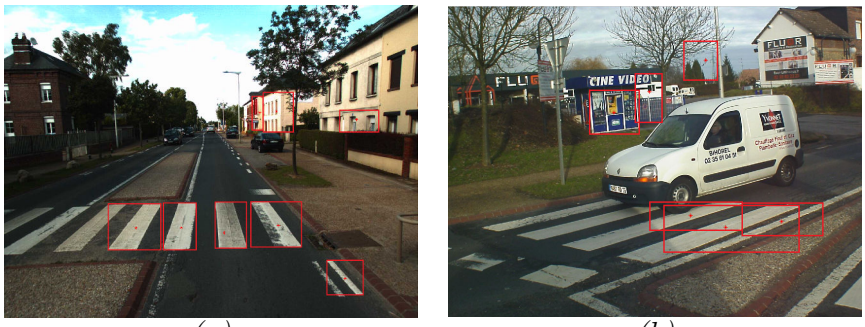


(a) (b) (c)

FIGURE *2.4 – Détection des objets brillants dans une image en appliquant une technique de segmentation colorimétrique – (a) Image originale de taille* $1216 \times 1600$ *pixels acquise par le système* $VIAPIX^{®}$ *[125] - (b) Résultat obtenu après avoir appliqué une méthode de seuillage adaptative permettant la détection des objets lumineux dans l'image - (c) Nettoyage du résultat de la détection en appliquant la morphologie mathématique (voir section 1.4.5.2 du chapitre 1 et annexe A pour plus de détails sur les quatre opérateurs élémentaires de la morphologie).*

D'après la figure 2.4, l'application d'une segmentation couleur permettant de détecter la couleur blanche dans une image donne des résultats acceptables en termes de bonnes détections. Cependant, les différents tests réalisés ont montré les limites de cette méthode. En effet, en appliquant la segmentation couleur sur la totalité de l'image, plusieurs objets qui n'appartiennent pas à la route sont détectés (ciel, voiture, bâtiment, etc.). Ce qui complique, ainsi, l'identification du marquage au sol (figure 2.5).

Pour faire face à ce problème, nous proposons une nouvelle méthode de détection illustrée dans la figure 2.6.




(a) (b)

FIGURE *2.5 – Exemples des fausses alarmes lors de l'application du seuillage adaptative pour la détection de la couleur blanche dans une image.*

Les résultats illustrés dans la figure 2.5 présentent des exemples de fausses alarmes obtenues lors de la détection. En effet, à côté de la détection des marquages routiers, nous remarquons dans la figure 2.5.a une détection d'une partie des bâtiments alors que dans la figure 2.5.b, nous avons une détection de la partie claire du ciel. Nous présentons dans la section suivante la solution proposée pour améliorer la détection.

#### 2.2.1.1 Solution proposée

L'approche proposée consiste, dans un premier temps, à se focaliser sur une zone spécifique dans l'image cible à traiter (image $VIAPIX^{®}$). Dans ce cadre, nous avons proposé à simplifier l'image d'origine d'une manière à se concentrer uniquement sur la chaussée en s'affranchissant de tous les autres objets inutiles (ciel, voitures, etc.). Ensuite, nous détectons la route afin d'extraire uniquement les objets de couleur blanche appartenant à cette route.

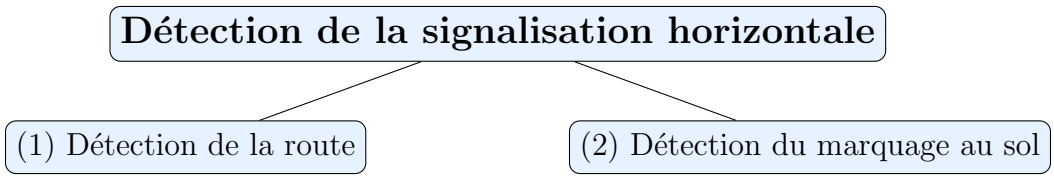



FIGURE *2.6 – Étapes principales pour la détection de la signalisation horizontale.*

#### 2.2.1.2 Détection de la route

En général, la détection des routes est une étape nécessaire souvent utilisée pour la conduite autonome dans le trafic urbain. Dans notre cas, nous l'avons proposée comme une étape intermédiaire visant à simplifier et faciliter l'étape de la détection du marquage au sol. Les étapes que nous avons proposées pour détecter la route sont présentées dans le synoptique de la figure 2.7.

Rappelons que dans ce travail, nous présentons une méthode de détection des zones routières basée sur des images couleur acquises par le système $VIAPIX^{®}$ (voir la section 1.6.2 du chapitre 1 pour plus de détails sur le système, l'installation sur le véhicule, les images, etc.).

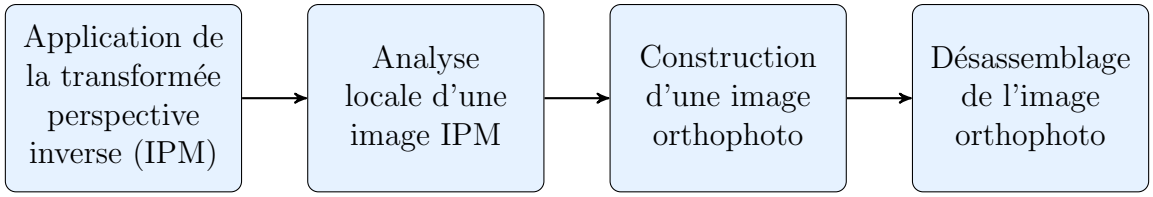
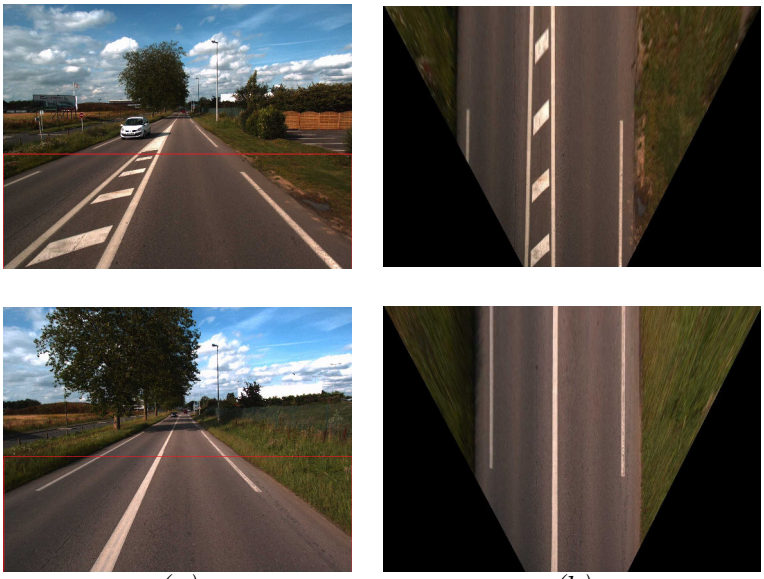



FIGURE *2.7 – Étapes réalisées pour la détection de la route.*

### a. Application de la transformation perspective inverse (IPM)

Comme nous l'avons précisé dans la section 2.2.1.1, notre objectif principal est de simplifier l'image d'origine $VIAPIX^{®}$. Nous cherchons ainsi à éliminer de l'image tous les objets qui ne se rapportent pas à la chaussée (voiture, ciel, etc.). Pour supprimer l'effet de la perspective, nous faisons appel à la technique de la transformée perspective inverse (IPM) qui est détaillée dans la section 1.4.5.1 du chapitre 1. Ci-après quelques exemples d'application de la transformée perspective inverse sur des images $VIAPIX^{®}$.



*(a)* *(b)*

FIGURE *2.8 – Simplification des images $VIAPIX^{®}$ en appliquant la transformée perspective inverse (IPM) – (a) Images originales - (b) Images simplifiées correspondantes aux parties encadrées en rouge dans (a).*

L'image perspective inverse obtenue est utilisée ainsi pour détecter la route et ensuite les marquages routiers.

### b. Analyse locale d'une image : détection locale

Dans ce paragraphe, nous présentons la méthode que nous avons développée pour détecter la route dans l'image simplifiée (figure 2.8.b). Nous notons que dans ce travail, les images $VIAPIX^{®}$ sont acquises à intervalle de distance régulier (soit une image tout les $x$ mètres avec $x = 1, 2, 3, ...$). Les images successives présentent un certain recouvrement, qui nuit à l'efficacité du traitement. Pour éviter la redondance définie par l'analyse de la même zone plusieurs fois et réduire le temps de calcul, la stratégie retenue consiste ainsi à sélectionner dans l'image perspective inverse une zone de $x$ mètres devant la voiture. La zone sélectionnée est considérée par la suite comme étant la zone dans laquelle nous cherchons à délimiter la route.

Nous proposons de réaliser, dans un premier temps, une segmentation colorimétrique, afin de pouvoir détecter les zones les plus sombres de la zone de recherche. Pour ce faire, nous avons proposé de sélectionner directement dans l'image un certains nombre de blocs distincts et de petite taille ($m \times n$ pixels). En effet, l'analyse approfondie des couleurs des blocs choisis va nous permettre de trouver approximativement la valeur moyenne du niveau de gris de la chaussée.

### i. Choix des blocs

Dans cette partie, nous allons présenter les deux cas que nous avons étudiés pour sélectionner les blocs de taille $m \times n$ pixels dans l'image perspective inverse.

Le premier cas, illustré dans la figure 2.9.a, consiste à sélectionner automatiquement des petits blocs dispersés sur toute la zone utile de $x$ mètres devant la voiture (dans notre cas, la valeur de $x$ est égale à 3 mètres). En effet, à 3 mètres devant la caméra du système $VIAPIX^{®}$, nous sommes sûr que la voiture va traverser cette partie de la route. Dans ce cas, l'analyse des couleurs des blocs choisis va certainement donner approximativement une idée sur la couleur de la route que l'on cherche à détecter.

Le deuxième cas, présenté dans la figure 2.9.b, permet de sélectionner aléatoirement les blocs dans l'image sans prendre en compte la zone de recherche limitée par les $x$ mètres devant la voiture. Dans ce cas, l'analyse des couleurs des blocs choisis va nous donner une estimation erronée de la couleur exacte de la chaussée en question. D'après cette figure, les blocs sélectionnés *au hasard* (soit dans le virage, soit à une grande distance de la caméra), ont permis une détection des parties de la route que peut-être la voiture ne va pas parcourir, soit une détection douteuse et donc loin de notre intention.

Ainsi, pour éviter ce problème, nous considérons dans notre travail que le premier cas est le cas le plus approprié qui permet de fournir une meilleure détection.

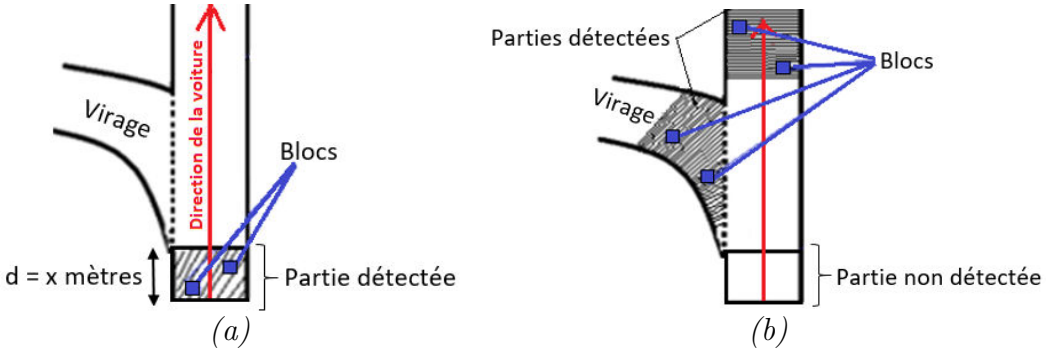



FIGURE *2.9 – Choix des blocs pour détecter la route devant la voiture – (a) Blocs sélectionnés dans la zone de recherche (zone de x mètres devant la voiture) - (b) Blocs sélectionnés aléatoirement dans l'image.*

La figure 2.10 illustre le résultat obtenu lors de l'application de cette méthode de sélection des blocs sur une image réelle $VIAPIX^{®}$ de taille $1216 \times 1600$ pixels.

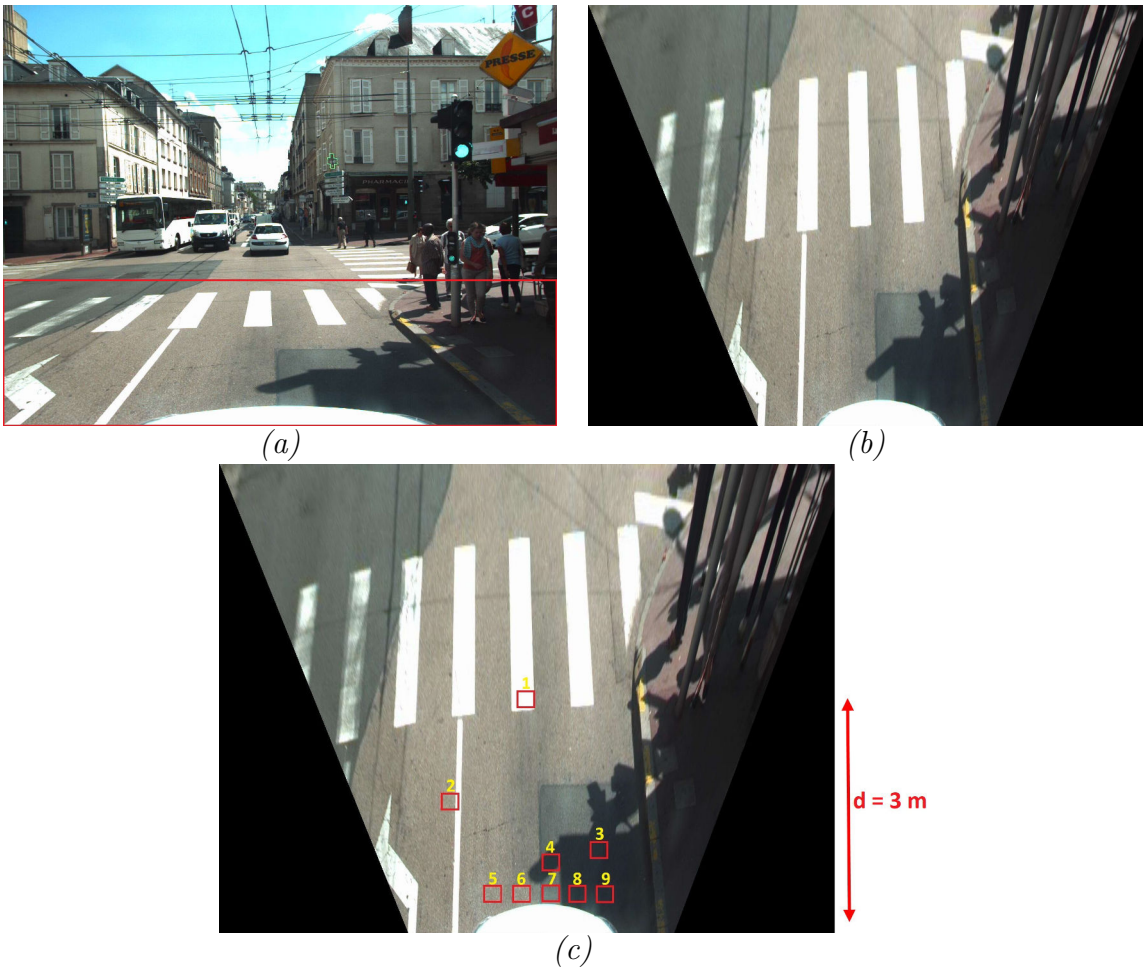



FIGURE *2.10 – Sélection des blocs de taille m×n pixels (m=n=12 pixels) dans une image* $VIAPIX^{®}$ *après avoir fait sa transformée perspective inverse – (a) Image* $VIAPIX^{®}$ *originale de taille* $1216 \times 1600$ *pixels - (b) Image perspective inverse correspondante - (c) Sélection des blocs à une distance* $d \leq 3$ *m devant la voiture.*

### ii. Filtrage des blocs sélectionnés

L'objectif principal de cette étape est de filtrer les blocs sélectionnés à l'étape précédente d'une manière à ne conserver que ceux qui ont une grande probabilité d'appartenir à la route. Autrement dit, nous cherchons à éliminer les blocs qui appartiennent par exemple à un marquage routier, au trottoir, etc. En effet, ce genre des blocs pourrait fausser le résultat de la détection de la route. Pour y arriver, une approche en deux étapes basée sur l'analyse de l'histogramme couleur $RVB$ (rouge, vert, bleu) de chaque bloc, est réalisée.

La première étape consiste à trouver la valeur du pixel le plus clair ($max$) et du pixel le plus sombre ($min$) dans chaque bloc et à calculer la différence $diff$ entre elles selon l'équation 2.1. Ensuite, cette dernière sera comparée à une valeur de seuil $S$ de telle sorte que si la valeur $diff$ est supérieure à $S$, nous considérons que le bloc est à supprimer. Nous notons que la valeur du seuil choisie dans ce travail est égale à 50. En effet, nous supposons que la couleur de la route ne devrait pas dépasser cette valeur et par conséquent tout objet ayant une valeur moyenne supérieure à ce seuil, est considéré comme l'un des objets inutiles qui n'appartiennent pas à la route.

$$diff = mean \begin{bmatrix} max(R) - min(R) \\ max(V) - min(V) \\ max(B) - min(B) \end{bmatrix} \tag{2.1}$$

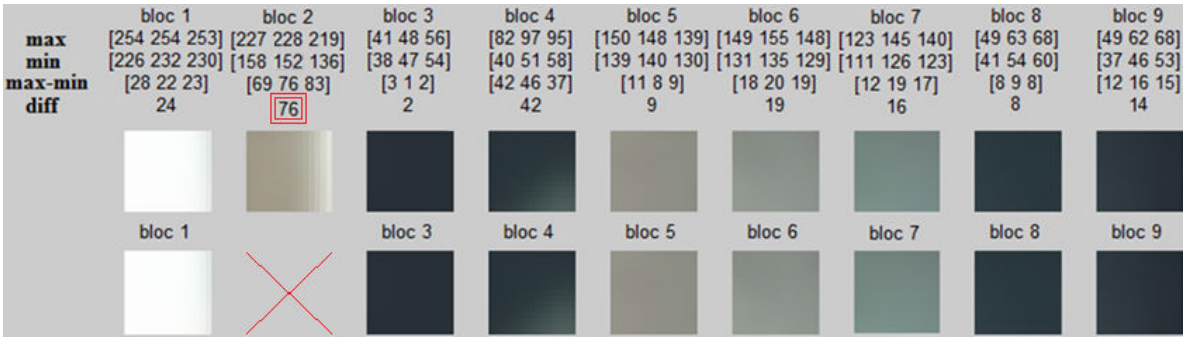



FIGURE *2.11 – Blocs restants après avoir comparé la valeur $diff$ à la valeur du seuil ($S = 50$).*

D'après la figure 2.11, la valeur $diff$ du bloc 2 est supérieure à la valeur du seuil $S$ (soit $76 > 50$), c'est la raison pour laquelle ce bloc a été supprimé. En effet, il appartient à la fois à la route et à une ligne blanche séparant deux voies de circulation (voir bloc 2 sur la figure 2.10.c).

De plus, il est possible qu'un bloc homogène qui ne contient que du blanc, i.e., un bloc qui appartient complètement à un marquage routier (bloc 1 dans notre cas), sera pris en compte comme étant un bloc appartenant à la route du fait que sa valeur $max$ et sa valeur $min$ sont proches et par conséquent la valeur $diff$ correspondante est inférieure au seuil $S$ (bloc 1 : $24 < 50$).

Afin de corriger ce problème, la deuxième étape est ainsi appliquée. Elle consiste tout d'abord à calculer les valeurs moyennes de tous les blocs préservés à l'étape précédente et ensuite à les soumettre à un filtre médian qui a pour objectif principal l'élimination des blocs ayant une grande valeur de moyenne. En effet, un grande valeur de moyenne signifie que la majorité des pixels du bloc en question sont brillants et donc ce bloc pourrait ne pas appartenir à la route.

En général, un filtre médian est un filtre numérique non linéaire, souvent utilisé pour la réduction du bruit. L'idée principale de ce filtre est de remplacer chaque entrée par la valeur médiane de son voisinage. Dans notre cas, il a été appliqué sur un vecteur linéaire contenant les valeurs moyennes des blocs restants.

**Exemple pour comprendre le principe du filtre médian utilisé :**

Supposons que nous avons cinq blocs ayant des valeurs moyennes **100, 20, 30, 15 et 40** (figure 2.12).

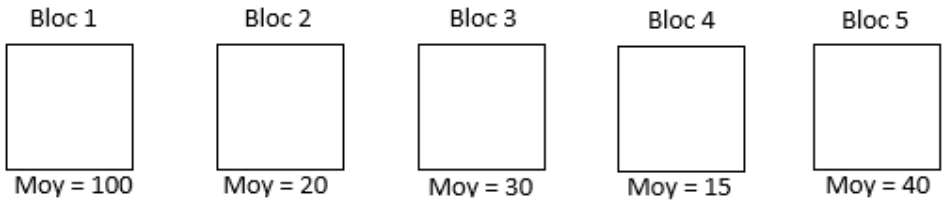



FIGURE *2.12 – Exemple des blocs sur lesquels nous allons appliquer le filtre médian.*

Dans cet exemple, nous cherchons à appliquer le filtre médian sur ces cinq valeurs dans le but de supprimer les blocs ayant une grande moyenne. Pour ce faire, trois étapes sont à réaliser :

- Ajouter des zéros au début et à la fin du vecteur linéaire à traiter ([**0, 100, 20, 30, 15, 40, 0**]). Cette addition va réduire la pénurie au cas où la taille du vecteur linéaire n'est pas cohérent pas avec la taille du filtre à utiliser.

- Sélectionner des séries de valeurs selon la taille du filtre (par exemple si la taille est égale à 3 donc nous prenons des séries de 3 valeurs comme illustré dans la figure 2.13).

- Trier les séries sélectionnées par ordre croissant et ensuite prendre la valeur médiane dans chacune (tableau 2.1).

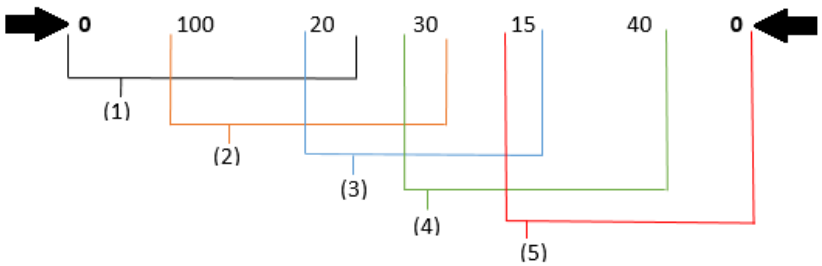



FIGURE *2.13 – Illustration du principe théorique du filtre médian.*

| Série | | Série triée | Valeur médiane | Bloc à conserver |
|---|---|---|---|---|
| (1) | 0 100 20 | 0 20 100 | 20 | **Bloc 2** |
| (2) | 100 20 30 | 20 30 100 | 30 | **Bloc 3** |
| (3) | 20 30 15 | 15 20 30 | 20 | **Bloc 2** |
| (4) | 30 15 40 | 15 30 40 | 30 | **Bloc 3** |
| (5) | 15 40 0 | 0 15 40 | 15 | **Bloc 4** |

TABLE *2.1 – Sélection des valeurs médianes et des blocs correspondants à conserver.*

D'après le tableau 2.1, les blocs 2, 3 et 4 sont les blocs à conserver. Ainsi le filtre médian a permis la suppression des blocs 1 et 5 (blocs ayant les plus grandes valeurs de moyennes).

**Application :**

L'application de ce type de filtrage sur les blocs restants illustrés dans la figure 2.11 donne le résultat présenté dans la figure 2.14.

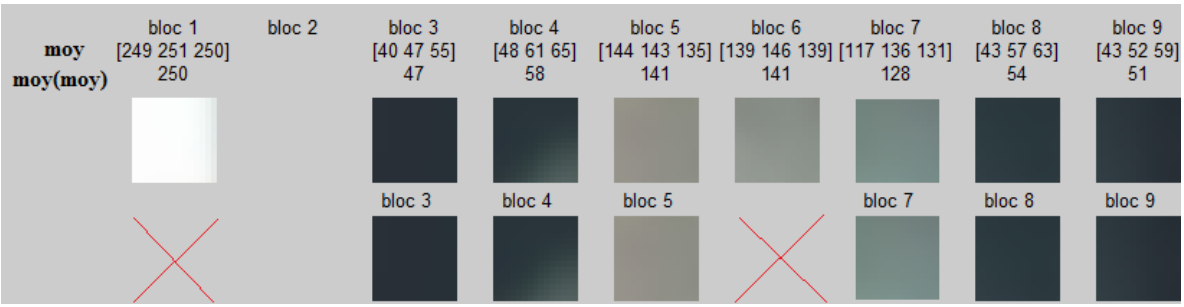



FIGURE *2.14 – Blocs restants après avoir appliqué le filtre médian sur les valeurs moyennes des blocs conservés dans la figure 2.11.*

D'après le résultat de la figure 2.14, cette étape a permis d'éliminer les blocs homogènes (le bloc 1 et le bloc 6) en comparaison des blocs appartenant à la route. Ensuite, pour détecter la route, une étape de segmentation couleur est ainsi appliquée.

### iii. Détection locale

L'objectif principal de cette étape est de segmenter l'image perspective inverse illustrée dans la figure 2.10.b pour extraire la route devant la voiture sur laquelle le système $VIAPIX^{®}$ est installé. Pour y arriver, nous réalisons une segmentation couleur en utilisant un seuillage adaptatif. En effet, ce seuil est calculé localement dans l'image en se basant sur les valeurs des trois composantes (rouge, vert, bleu) des blocs préservés à l'étape précédente (voir figure 2.14). Il est ainsi défini par l'équation 2.2 de telle sorte que si la valeur d'un pixel $P$ est à l'intérieure de cet intervalle, nous supposons que le

pixel $P$ appartient à la route. Dans le cas contraire, le pixel n'est pas pris en compte.

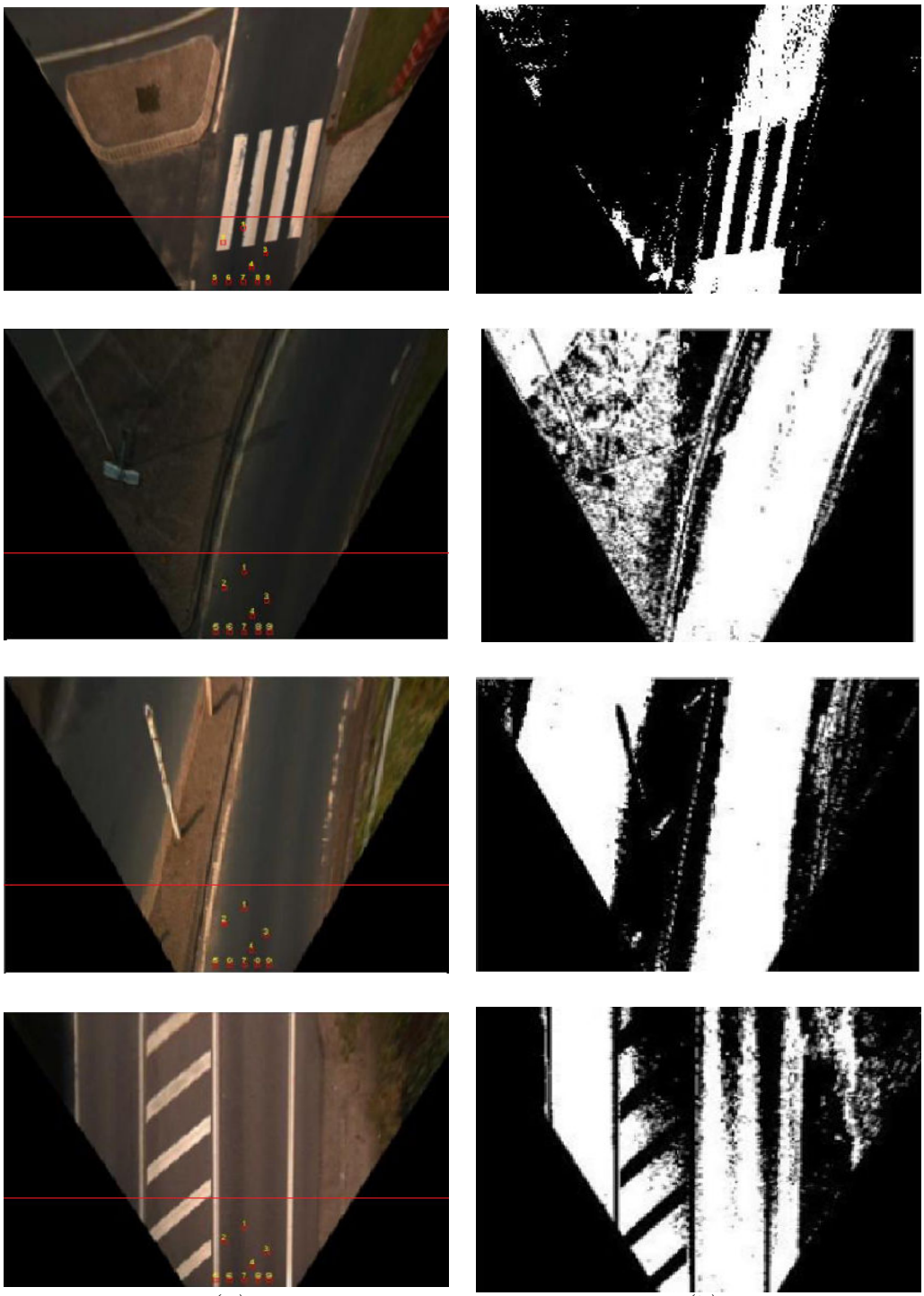
$$S_{route} = \left[min - \sigma, max + \sigma\right] \tag{2.2}$$

avec $min$, $max$ et $\sigma$ correspondent respectivement à la valeur minimale, la valeur maximale et l'écart-type de tous les blocs sélectionnés dans l'image (voir figure 2.14).

La figure 2.15 montre le résultat de la segmentation obtenu après avoir appliqué cette étape du seuillage adaptatif sur l'image illustrée dans la figure 2.10.b

FIGURE *2.15 – Résultat de la détection de la route après le seuillage $S_{route}$ – La partie de la route qui a été détectée est présentée en blanc.*

Après avoir réalisé plusieurs tests, nous avons remarqué que cette méthode de détection basée sur l'analyse locale de l'image, permet dans la plupart des cas de segmenter correctement la route devant la caméra. Cependant, dans certains cas, elle s'est montrée insuffisante du fait qu'elle ne permet de détecter dans l'image que la zone autour des blocs de taille $m \times n$ sélectionnés. Cette imperfection est présentée par la figure 2.16 qui illustre quelques exemples montrant la partie de la route qui a été détectée. Afin de résoudre ce problème et ainsi arriver à détecter complètement la route dans l'image, nous proposons dans la section suivante une méthode d'assemblage d'images permettant de coller les images segmentées obtenues à l'étape précédente pour construire une image "**orthophoto**" contenant la route complète sans aucun manque.



*(a)* *(b)*

FIGURE *2.16 – Illustration de quelques exemples montrant le résultat de la détection de la route sur des images perspectives inverses – (a) Images perspectives inverses ; la ligne horizontale en rouge limite la zone dont laquelle les blocs de taille $m \times n$ ont été sélectionnés - (b) Résultats de la segmentation colorimétrique.*

### c. Construction d'une image orthophoto : détection globale

Cette section présente la méthode de fusion d'images que nous avons proposée afin de pouvoir reconstruire la route. D'après [12, 13], la fusion d'images est une technique pertinente qui consiste à coller un ensemble des images successives pour former une seule image nommée "orthophoto". Son objectif principal est d'améliorer et de simplifier la phase de la détection de la route.

Pour y arriver, nous nous basons sur les images perspectives inverses. En effet, la fusion de ces images nécessite la connaissance de la direction de la voiture et des coordonnées GPS de la caméra lors de l'acquisition de l'image. La figure 2.17 montre une séquence d'images que l'on cherche à fusionner. Tout d'abord, nous notons $C_i$ $(x_i, y_i)$ les coordonnées de la caméra (en mm) pour chaque image $i$. Comme il est montré dans la figure 2.17, ces coordonnées sont obtenues en calculant l'intersection des deux droites $(EF)$ et $(GH)$. À l'issue de cette étape, nous avons les coordonnées de chaque caméra dans son propre repère, soit par exemple $C_1$ les coordonnées de la caméra dans le repère contenant l'image 1, $C_2$ les coordonnées de la caméra dans le repère contenant l'image 2, et ainsi de suite, ...

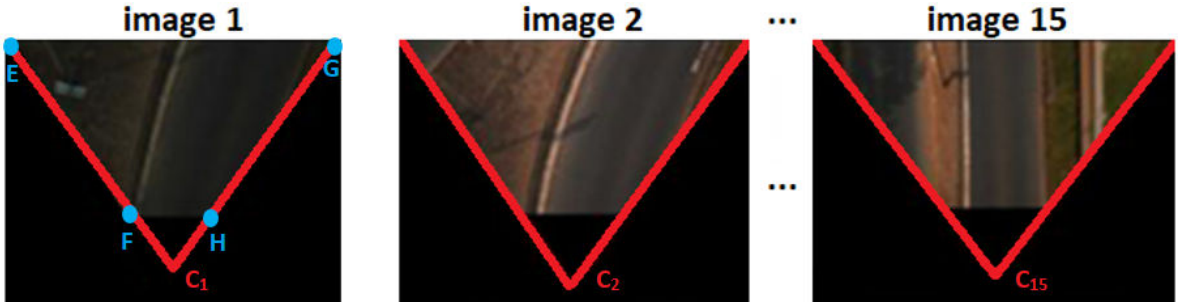



FIGURE *2.17 – Coordonnées de la caméra dans chaque image perspective inverse – Chaque image i est décrite par un point $C_i$ de coordonnées $(x_i, y_i)$ en mm.*

Ensuite, pour pouvoir coller les images, il convient de souligner que dans notre procédure, toutes les coordonnées doivent se référer à un système de coordonnées unique. Autrement dit, il faut que toutes les coordonnées $C_i$ soient calculées dans le même repère de la caméra 1 dont le centre est $C_1$. Du fait que nous travaillons sur des images perspectives inverse, i.e., les coordonnées de tous les pixels sont en mm, il est nécessaire de convertir les coordonnées GPS (latitude et longitude) en coordonnées millimétriques (mm). En utilisant l'orientation de chaque caméra au moment de l'acquisition et sa position par rapport à la position initiale (caméra 1), et en se basant sur la formule de Vincenty [126] donnée par l'équation 2.3, nous pouvons déterminer les coordonnées métriques de chaque caméra.

$$\begin{aligned} X &= R \times (lon_i - lon_1) \times cos(\tfrac{(lat_i + lat_1)}{2}) \\ Y &= R \times (lat_i - lat_1) \end{aligned} \tag{2.3}$$

avec $X$ et $Y$ en mm correspondent aux coordonnées de la caméra dans l'image $i$ ($C_i$) par rapport à la caméra de référence ($C_1$), ($lat_i$, $lon_i$) sont les coordonnées GPS des points $C_i$ en radian et $R = 6371.3$ km est le rayon de la terre considérée ellipsoïde [126].

La figure 2.18 montre l'image orthophoto construite après avoir fusionné les 15 images présentées dans la figure 2.17.

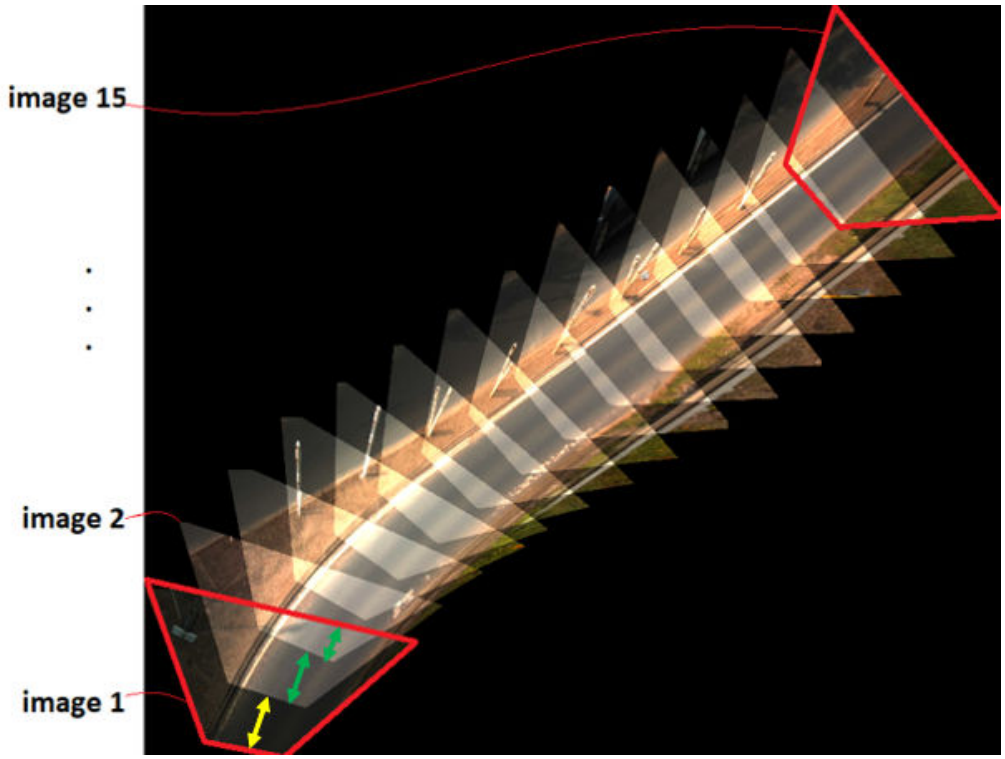
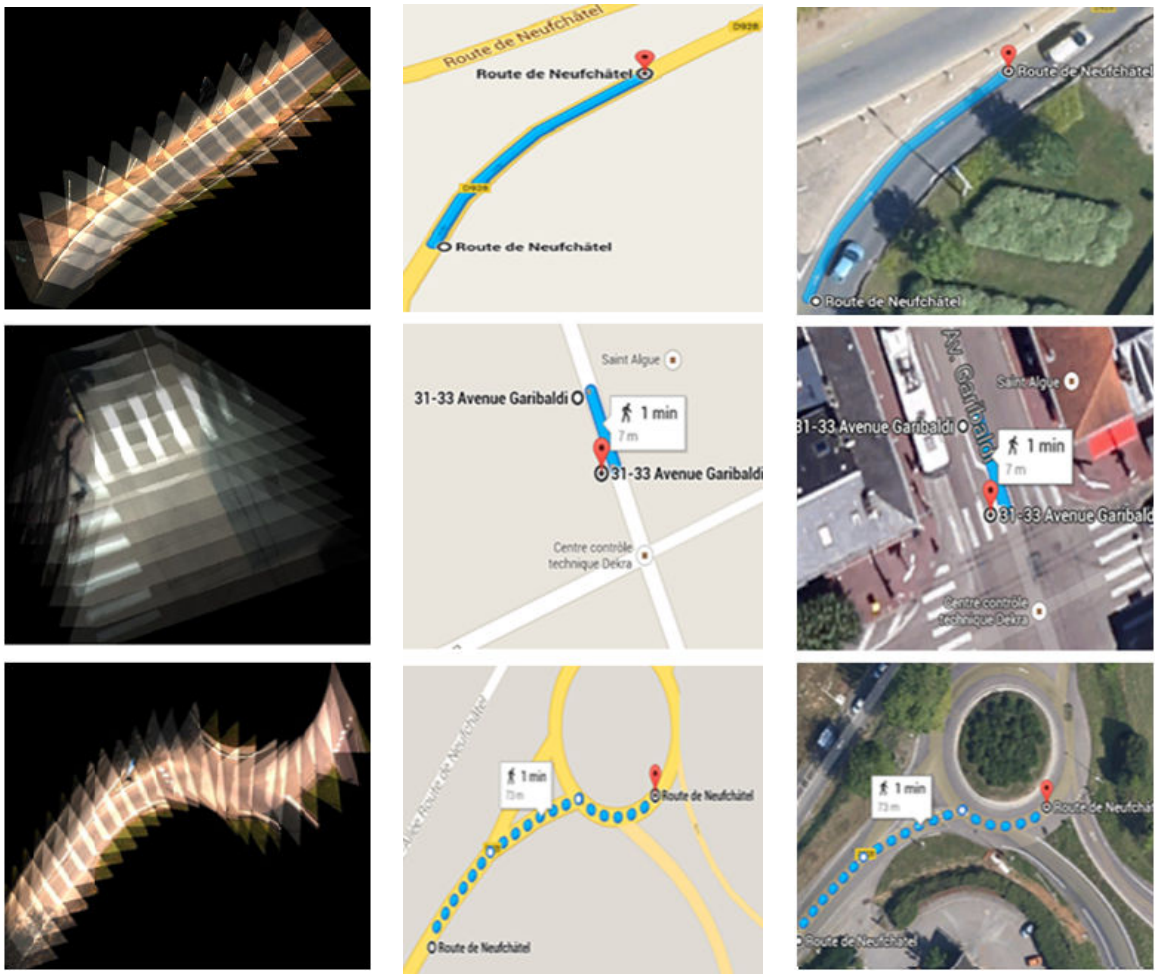



FIGURE *2.18 – L'image orthophoto obtenue après la fusion des* 15 *images perspectives inverses présentées dans la figure 2.17.*

En se basant sur le résultat illustré dans la figure 2.18, il n'est plus nécessaire de détecter toute la route dans une seule image, mais uniquement la partie de la route qui n'appartient pas à la zone d'interférence (la zone jaune représentée sur la figure 2.18), parce que l'autre partie de la route, appartenant à la zone d'interférence (les zones vertes représentées sur la figure 2.18), sera détectée dans les images suivantes.

La figure 2.19.a illustre quelques exemples montrant la construction des images orthophotos après avoir fusionné plusieurs ensembles d'images perspectives inverses. Afin de nous assurer que le résultat de la construction de l'image orthophoto est correct, nous avons visualisé sur Google Maps (en se basant sur les coordonnées GPS), le chemin qui relie l'image de départ et l'image de destination. Selon la figure 2.19.b-c, Google Maps fournit un chemin similaire (quasiment le même chemin construit).

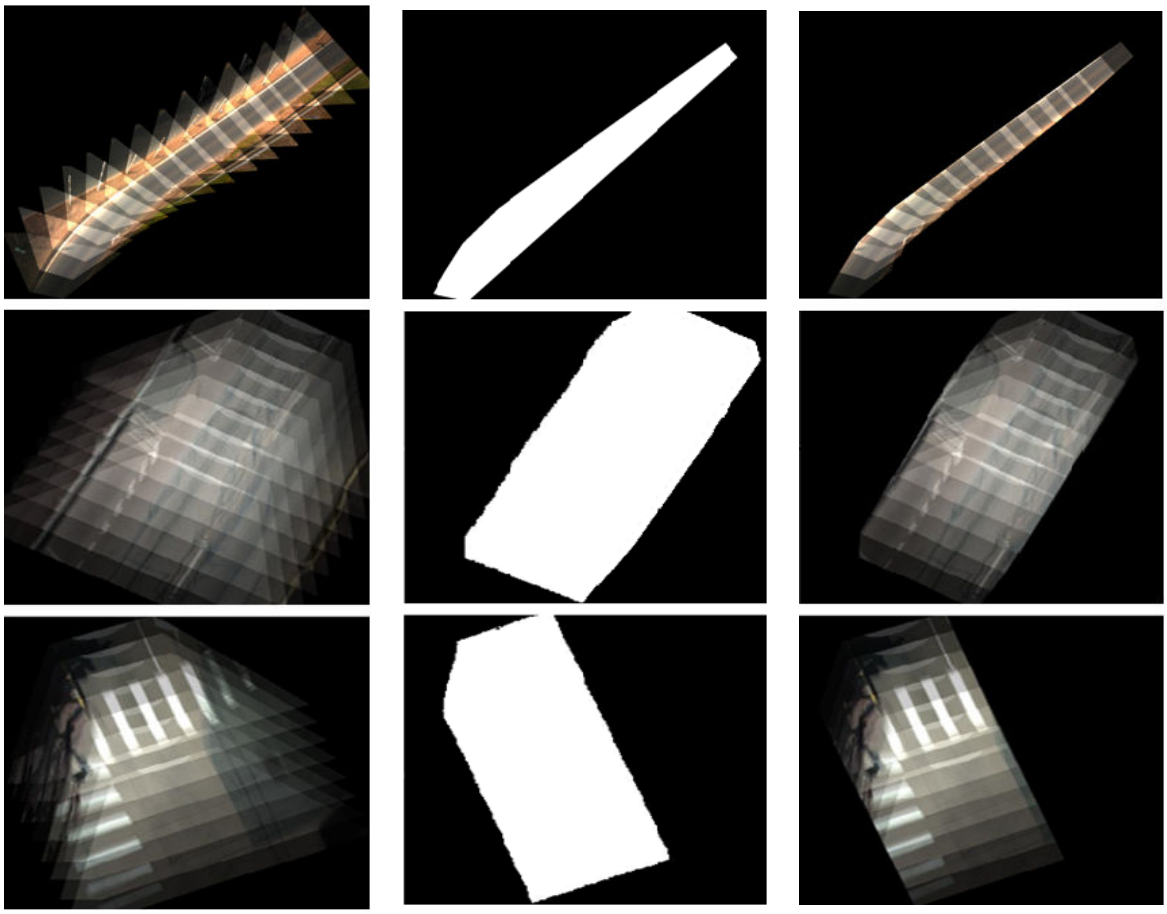



*(a)* *(b)* *(c)*

FIGURE *2.19 – Présentation de quelques exemples illustrant la construction des images orthophotos – (a) Images orthophotos - (b-c) Résultats obtenus en utilisant Google Maps.*

Ensuite, afin de réaliser notre objectif principal qui consiste à détecter la route, nous avons appliqué la méthode de fusion d'images détaillée précédemment sur un ensemble des images perspectives inverses binaires obtenues à l'issue de la phase de la détection locale (voir figure 2.15). Ainsi, le résultat obtenu est une image orthophoto binaire construite à partir des images perspectives inverses segmentées (figure 2.20.b).

Finalement, l'étape suivante consiste à utiliser le masque résultant de la segmentation afin de ne conserver que les informations relatives à la route (figure 2.20.c).

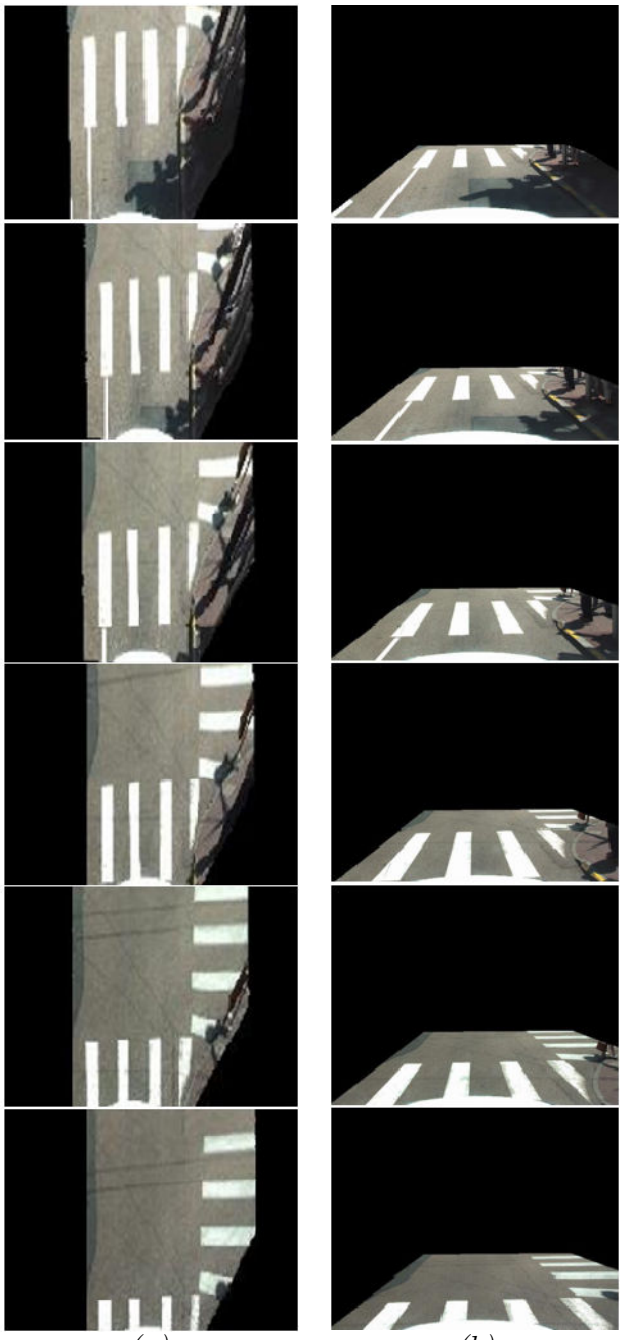


*(a)* *(b)* *(c)*

FIGURE *2.20 – Résultat de la détection globale de la route – (a) Images orthophotos fusionnant les images perspectives inverses - (b) Images orthophotos binaires fusionnant les images perspectives inverses segmentées - (c) Résultats de la détection globale de la route.*

### d. Désassemblage de l'image orthophoto : extraction de la route

Cette étape consiste à désassembler l'image orthophoto résultante (figure 2.20.c) dans le but d'extraire les caractéristiques de la route dans chaque image à part. Pour ce faire, nous avons procédé à séparer l'image orthophoto de manière à avoir en sortie des images perspectives inverses contenant uniquement la partie de la route détectée. De plus, afin de visualiser le résultat final sur l'image $VIAPIX^{®}$ originale, nous n'avons ainsi qu'à réaliser l'inverse de la transformée perspective inverse ($IPM^{-1}$). La figure 2.21 présente le résultat obtenu lors de l'application de cette étape de désassemblage sur l'image orthophoto illustrée dans la figure 2.20.c (troisième exemple).



(a) (b)

FIGURE *2.21 – Détection de la route dans chaque image après avoir désassemblé l'image orthophoto présentée dans la figure 2.20.c (troisième exemple) – (a) Résultats sur les images perspectives inverses - (b) Résultats sur les images* $VIAPIX^{®}$ *originales après avoir appliqué* $IPM^{-1}$ *sur les images de (a).*

L'étape suivante de notre algorithme consiste à détecter tous les objets lumineux appartenant à la route détectée et qui ressemblent à des marquages routiers.

#### 2.2.1.3 Segmentation couleur

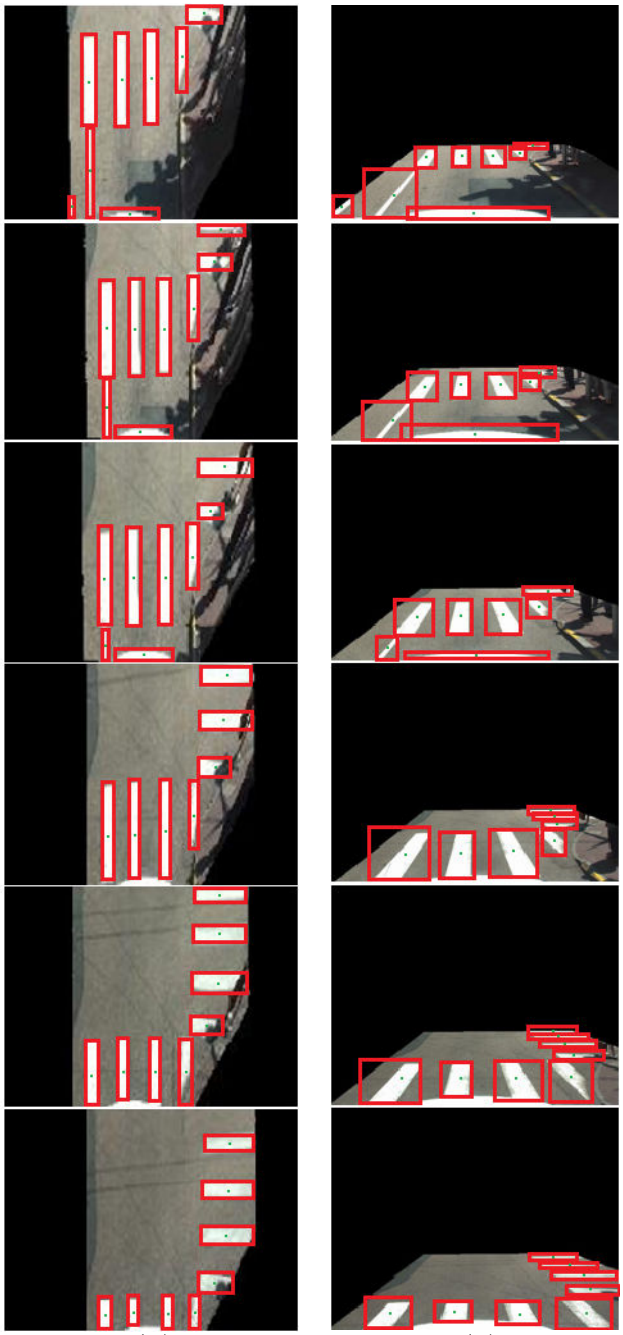
Cette étape consiste à détecter les objets de la signalisation horizontale qui appartiennent à la route obtenue à l'étape précédente (voir figure 2.21). Pour y arriver, nous nous sommes basés sur les résultats obtenus sur des images perspectives inverses. En effet, ces images contiennent uniquement une partie de la route ainsi que le marquage au sol sur cette zone.

Afin de détecter la signalisation horizontale sur la route, nous cherchons à extraire tous les objets ayant une couleur blanche sur la route. Nous proposons, ainsi, de réaliser une étape de segmentation couleur en utilisant un seuillage adaptatif sur l'espace colorimétrique RVB. L'objectif de cette segmentation est l'extraction des objets brillants et lumineux qui pourraient correspondre à des marquages routiers. Dans ce cadre, la valeur du seuil est calculée en fonction des paramètres de luminosité fournis par le système $VIAPIX^{®}$. Elle est donnée par l'équation 2.4 suivante :

$$S_{marquage} = \left[\sigma + avg, max\right] \tag{2.4}$$

avec $\sigma$, $avg$ et $max$ correspondent respectivement à la valeur de l'écart-type, la valeur moyenne et la valeur maximale de chacune des trois composantes rouge, vert et bleu (RVB) de la figure 2.21.a.

Nous attribuons la valeur "0" à chaque pixel dont les valeurs R, G et B sont inférieures au seuil et "1" dans le cas inverse. Le résultat obtenu est, ainsi, une image binaire. Pour la clarté, nous présentons à la figure 2.22 le résultat final de la détection des marquages routiers (les objets considérés comme étant des objets blancs sont encadrés en rouge).



(a) (b)

FIGURE 2.22 – *Détection du marquage routier dans chacune des images de la figure 2.21 – (a) Résultats sur les images perspectives inverses - (b) Résultats sur les images VIAPIX® originales.*

Plusieurs tests sur différentes images contenant des types de marquage autres que les bandes des passages piétons ont été réalisés. Ainsi, quelques résultats obtenus sont illustrés dans la figure 2.23.

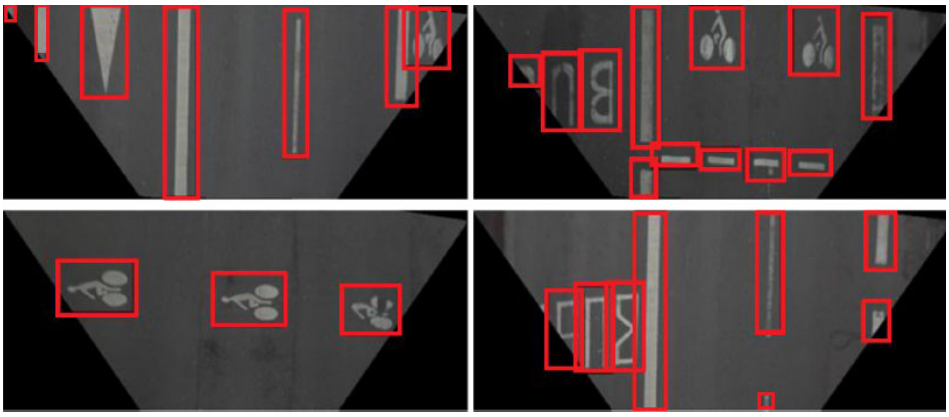


FIGURE *2.23 – Résultats de la détection du marquage au sol en testant d'autres images.*

Les figures 2.22 et 2.23 montrent que l'approche de détection proposée est capable de détecter tous les types de marquages au sol (passages piétons, flèches directionnelles, pictogrammes-vélos, mot "BUS", lignes continues, lignes discontinues, etc.). En effet, l'application d'une segmentation couleur suivie d'une étape de nettoyage réalisée par la morphologie mathématique (voir section 1.4.5.2 du chapitre 1 et annexe A pour plus de détails sur les quatre opérateurs élémentaires de la morphologie), permet avec succès d'extraire tous les objets de couleur blanche appartenant aux zones claires de l'image. La prochaine phase consiste ainsi à identifier le type de chaque marquage détecté.

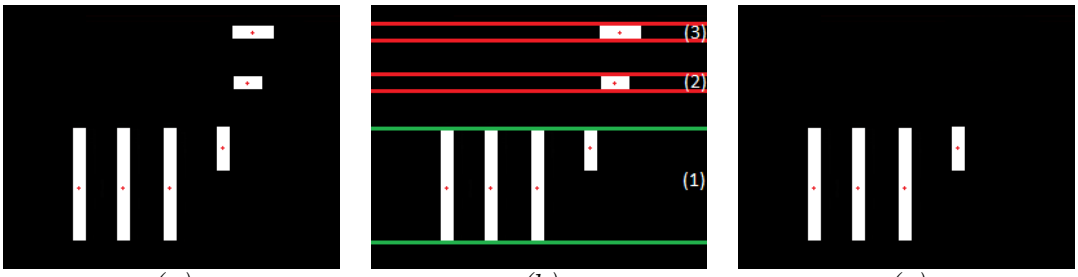
## 2.2.2 Identification des marquages détectés

Dans cette section, nous présentons la méthode proposée pour identifier les objets détectés à l'étape précédente. L'objectif de cette étape est de vérifier si les objets détectés correspondent vraiment à des marquages routiers ou non. Conformément à ce que nous avons déjà mentionné au début de ce chapitre, dans ce travail de thèse, nous n'avons étudié que les trois cas suivants : **le passage piéton - le mot"BUS"**, et **le pictogramme-vélo**. Pour ce faire, deux méthodes géométriques et optiques indépendantes ont été proposées [12, 13].

### 2.2.2.1 Méthode de reconnaissance géométrique

Nous discutons dans cette section la méthode proposée qui consiste à déterminer parmi les objets détectés ceux qui correspondent à des bandes d'un passage piéton sur la route. Plus précisément, notre objectif est d'identifier le marquage au sol de type : **passage piéton**. Par définition, un passage piéton est le type du marquage qui est constitué de plusieurs bandes rectangulaires blanches et parallèles à l'axe de la route ayant une longueur minimale de 2.5 m en ville et de 4 à 6 m en campagne, une largeur de 0.5 m et d'une distance inter-bandes de 0.5 m à 0.8 m. Pour ce faire, une procédure de trois étapes est effectuée. La première étape consiste à trouver le centre de chaque objet détecté (les points rouges présentés sur la figure 2.24.a). Ensuite, la deuxième étape, illustrée dans la figure 2.24.b, permet de calculer pour chaque objet une

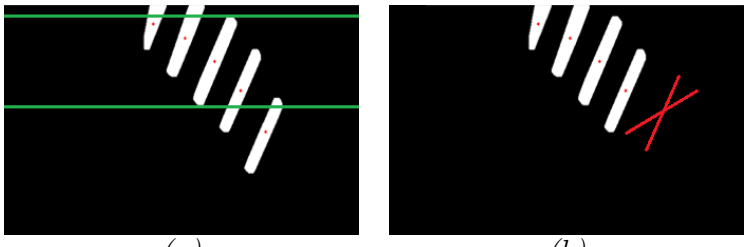
série de 2 lignes horizontales parallèles formant l'intervalle qui contient l'objet détecté. Enfin, la troisième étape est appliquée. Elle consiste à chercher l'intervalle regroupant le maximum de bandes détectées (soit l'intervalle qui contient le plus grand nombre des centres détectés). Dans le cas de la figure 2.24.b par exemple, l'intervalle 1 (en vert) est celui qui contient un maximum des objets (soit 4 bandes juxtaposées). Par conséquent, les objets appartenant à cet intervalle sont par la suite considérés comme étant les vraies bandes du passage piéton dans l'image. Les autres objets sont éliminés (figure 2.24.c).




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

FIGURE *2.24 – Illustration de la méthode des intervalles horizontaux pour reconnaitre les bandes d'un même passage piéton – (a) Centres des objets détectés - (b) Intervalles horizontaux - (c) Bandes reconnues d'un passage piéton horizontal.*

Le résultat de la figure 2.24 présente une bonne reconnaissance d'un passage piéton composé de 4 bandes réelles. En effet, la méthode de calcul des intervalles horizontaux a permis d'éliminer les objets qui sont à l'extérieur de l'intervalle idéal (qui contient un maximum des objets détectés). Les différents tests réalisés ont montré que cette méthode est capable d'identifier avec succès les passages piétons non-orientés devant la voiture. En contrepartie, dans le cas d'un passage piéton oblique, elle reste insuffisante du fait que l'intervalle horizontal permet dans certains cas la suppression des vraies bandes d'un passage piéton. Un contre exemple présentant ce problème est illustré dans la figure 2.25.



*(a)* *(b)*

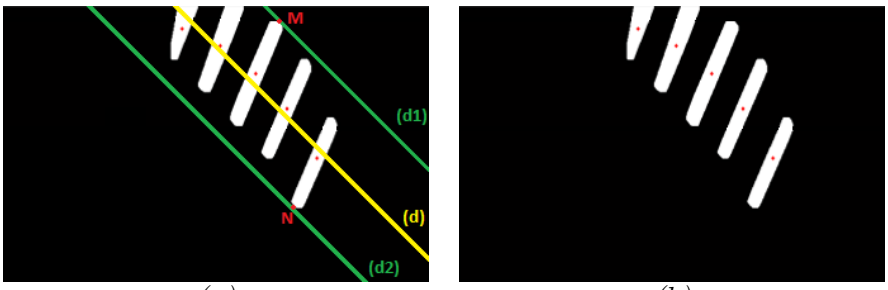
FIGURE *2.25 – Mauvaise identification d'un passage piéton oblique formé de 5 bandes – (a) Intervalle horizontal contenant le maximum des objets détectés -(b) Suppression d'une bande réelle.*

Le résultat de la figure 2.25 montre une mauvaise reconnaissance du passage piéton. En effet, l'intervalle horizontal présenté en vert dans la figure 2.25.a a permis de supprimer la dernière bande du passage piéton. Le problème illustré par cet exemple vient du fait que le passage piéton en question est orienté (oblique) devant la voiture. Et donc la zone qui regroupe toutes les bandes obliques ne ressemble pas à la forme de l'intervalle choisi.

Pour résoudre ce problème et ainsi éviter de supprimer aléatoirement certains objets utiles dans l'image, nous avons proposé de tracer un intervalle oblique et orienté capable de contenir toutes les bandes quelque soit l'orientation du passage piéton à traiter (voir figure 2.26). Pour ce faire, trois étapes sont à réaliser :

- Tracer la droite $(d)$ qui passe par le maximum des centres des objets détectés dans l'image. Dans notre cas, la droite est calculée en appliquant la méthode de Mayer [127] qui permet de trouver un ajustement affine entre une série des points dont le nuage est à peu près aligné.
- Trouver les deux points $M$ et $N$ qui correspondent respectivement aux points les plus éloignés (perpendiculairement) de la droite $(d)$ des deux côtés et qui appartiennent aux objets détectés dans l'image.
- Tracer les deux droites $(d1)$ et $(d2)$ parallèles à $(d)$ et passant respectivement par les points $M$ et $N$.

Ensuite, la zone représentée par les deux droites $(d1)$ et $(d2)$ est ainsi considérée comme étant l'intervalle orienté qui devrait contenir toutes les bandes du passage piéton.




*(a)* *(b)*

FIGURE *2.26 – Bonne identification du passage piéton oblique présenté dans la figure 2.25 – (a) Intervalle oblique - (b) Conservation de toutes les bandes.*

Enfin, pour augmenter le taux de la reconnaissance, nous supposons que si un même objet (bande rectangulaire) est détecté 3 fois dans 3 images successives, il correspondrait à une bande réelle du passage piéton. Un exemple illustrant une bonne reconnaissance d'un passage piéton est illustré dans la figure 2.27.

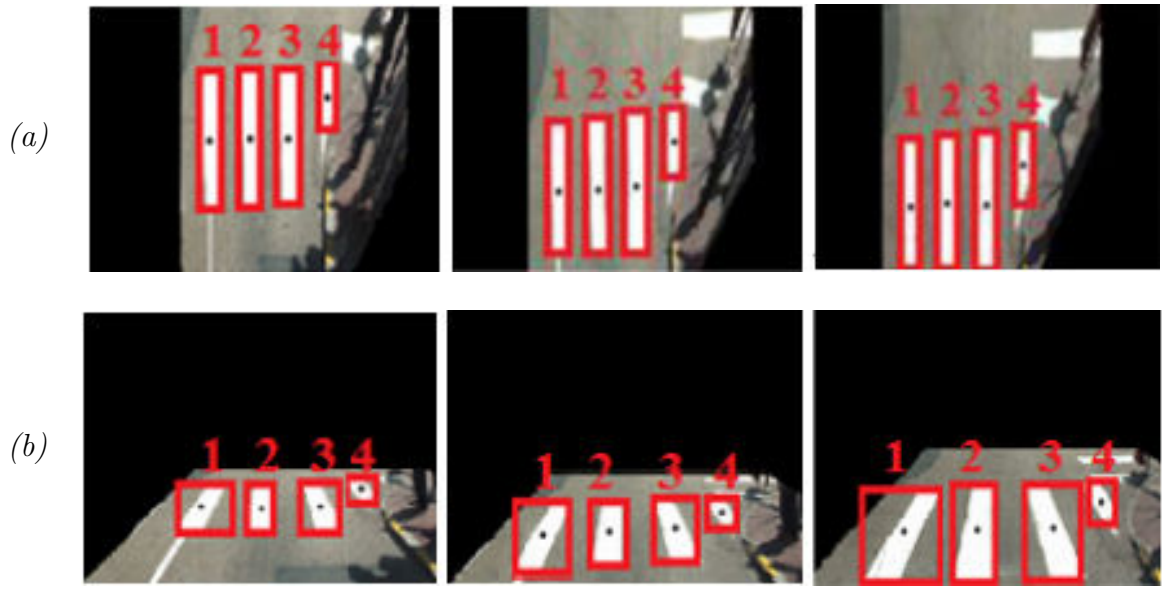



FIGURE *2.27 – Reconnaissance d'un passage piéton sur trois images successives – (a) Résultats sur des images perspectives inverses - (b) Résultats sur les trois images* $VIAPIX^{®}$ *correspondantes.*

La figure 2.27.a présente les marquages routiers identifiés sur trois images successives et constitués de 4 zones lumineuses juxtaposées. Après l'identification des marques sur la route, nous pouvons revenir à l'image d'origine capturée par le module $VIAPIX^{®}$ en effectuant l'inverse de IPM (figure 2.27.b). Le paragraphe suivant présente la méthode proposée pour identifier les deux autres types de marquages détectés dans l'image.

#### 2.2.2.2 Méthode de reconnaissance optique

L'objectif principal de cette étape est d'identifier les deux autres types de marquages de la route (autres que le passage piéton) et d'éliminer toutes les fausses détections. Nous rappelons que la méthode optique proposée dans ce document est appliquée pour l'identification du **mot BUS** et des **pictogrammes-vélos** uniquement.

Dans ce cadre, nous nous basons sur le principe de la corrélation optique. Par définition, la corrélation est une méthode capable de donner un niveau de discrimination très élevé et d'identifier un objet en comparant une image cible à une image de référence [128–131]. Elle est souvent utilisée pour obtenir une décision robuste et discriminante pour les applications de reconnaissance d'objets.

Pour y arriver, une procédure en trois-étapes présentée par la figure 2.28 est effectuée [132, 133].

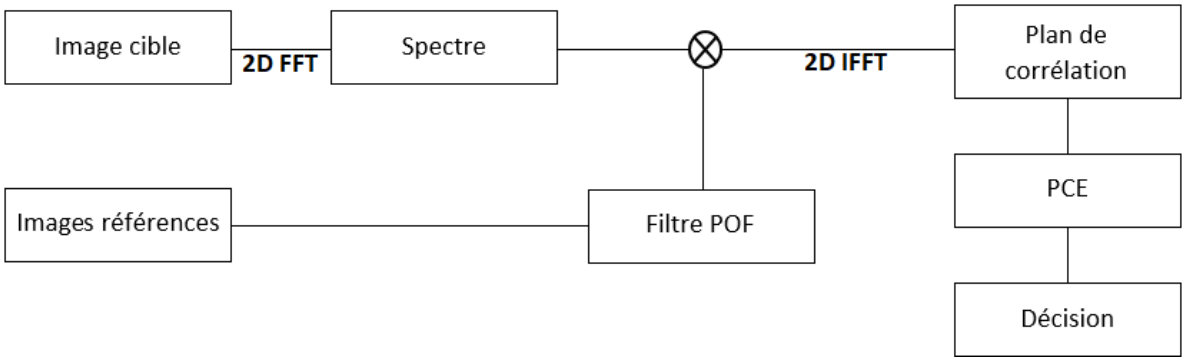



FIGURE *2.28 – Principe de la corrélation optique.*

Premièrement, le spectre de l'image cible est calculé en effectuant une transformation de Fourier rapide FFT. Deuxièmement, le spectre de l'image cible est multiplié par un filtre de corrélation fabriqué à partir d'un ensemble d'images de références stockées dans une base de données. Dans notre cas, le filtre POF (Phase Only-Filter) est celui qui a été utilisé en raison de ses bonnes performances en termes de robustesse et de discrimination [12, 133, 134]. Le calcul du filtre POF est donné par l'équation 2.5.

$$H_{POF}(u,v) = \frac{R^*(u,v)}{|R(u,v)|} \tag{2.5}$$

où u et v sont des variables indépendantes du domaine fréquentiel. $R(u,v)$ et $R^*(u,v)$ représentent respectivement le spectre d'une image référence et son conjugué.

Ensuite, nous appliquons au produit résultant une transformée de Fourier rapide inverse IFFT. Le critère d'évaluation de la reconnaissance est ainsi défini par l'énergie du pic de corrélation (PCE : Peak Correlation Energy). En général, ce critère correspond au ratio entre l'énergie du pic de corrélation et l'énergie totale du plan de corrélation(voir équation 2.6). La valeur calculée est ensuite comparée à un seuil donné $T(< 0.1)$ de telle sorte que si la valeur du PCE est supérieure à $T$, l'algorithme s'arrête et l'objet détecté sera assemblé aux marques associées (bonne reconnaissance). Sinon, nous comparons l'objet à l'image référence suivante. Cette étape est répétée jusqu'à l'exploitation de tous les filtres de la base. Si la valeur du PCE est toujours inférieure au seuil, l'objet sera donc supprimé comme étant une fausse détection.

$$PCE = \frac{\sum_{i,j}^{N} E_{pic}(i,j)}{\sum_{i,j}^{M} E_{plan}(i,j)} \tag{2.6}$$

La figure 2.30 montre des tests expérimentaux pour la reconnaissance des marquages routiers ; le mot BUS et les pictogrammes-vélo, en utilisant la méthode de corrélation

optique décrite précédemment. Dans ces tests, nous nous appuyons sur les images obtenues après la détection de la route et des zones claires de l'image (voir section 2.2.1.3 et figure 2.23). En outre, pour mettre en œuvre cette méthode de corrélation optique, nous avons utilisé plusieurs images de références montrant les lettres (B, U et S) du mot BUS et les pictogrammes-vélos en tenant compte des différentes formes et orientations possibles trouvées sur la route (voir figure 2.29).

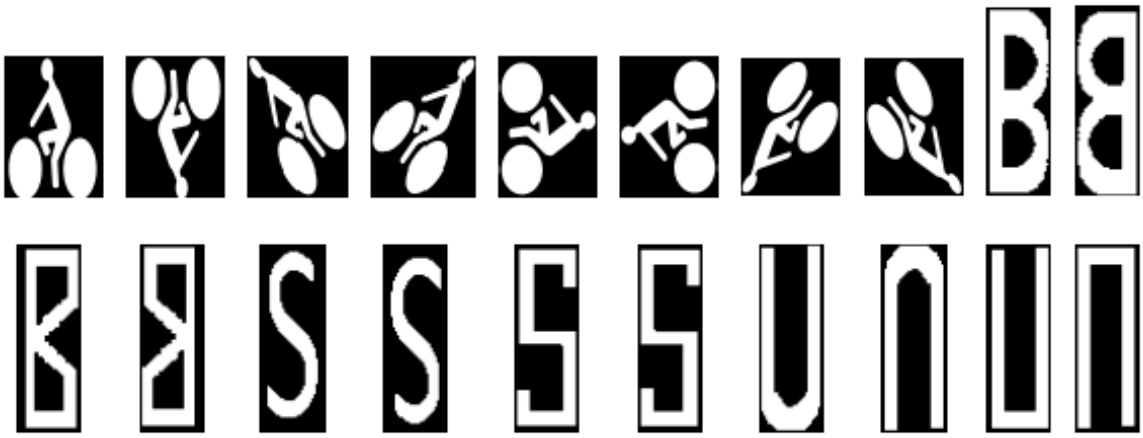


FIGURE *2.29 – Illustration de certaines images références utilisées pour la reconnaissance du mot BUS et du pictogramme-vélo.*

Pour la reconnaissance du mot BUS, trois étapes ont été effectuées. Premièrement, parmi les différents objets détectés (figure 2.23), nous ne conservons que les objets appartenant aux zones droite et gauche de la route. En effet, nous supposons que les zones navigables des BUS ne sont pas les mêmes comme les voitures. Deuxièmement, nous appliquons la méthode de corrélation, basée sur le filtre POF et les différentes images de référence sélectionnées pour chaque objet, et la méthode de seuillage sur la valeur du PCE (figure 2.28). Cela nous permet de décider si l'élément en question correspond ou non à une des trois lettres du mot BUS. Troisièmement pour prendre une décision finale, nous supposons que la reconnaissance de deux lettres est une condition suffisante pour décider si ce type de marquage correspond au mot BUS sur la route.

Pour la reconnaissance du pictogramme-vélo, nous conservons les étapes 1 et 2 mentionnées ci-dessus. Cependant, pour prendre la décision, nous proposons, en utilisant les différentes formes et orientations possibles, de diviser l'image du vélo en trois parties (la tête de la personne, le corps de la personne et les roues du vélo). Ensuite, nous analysons le nombre de pixels détectés dans chaque partie.

Afin de valider notre approche, plusieurs tests ont été réalisés. La figure 2.30 montre les résultats obtenus après l'application de la corrélation aux images présentées à la figure 2.23.

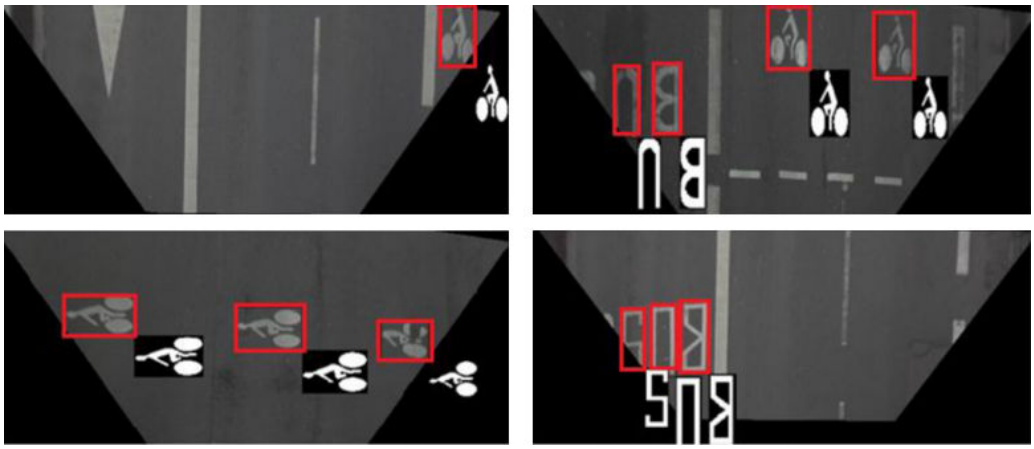


FIGURE *2.30 – Reconnaissance du mot BUS et du pictogramme-vélo.*

D'autres exemples illustrant la bonne reconnaissance du mot BUS et des pictogrammes-vélos sont présentés dans la figure 2.31.

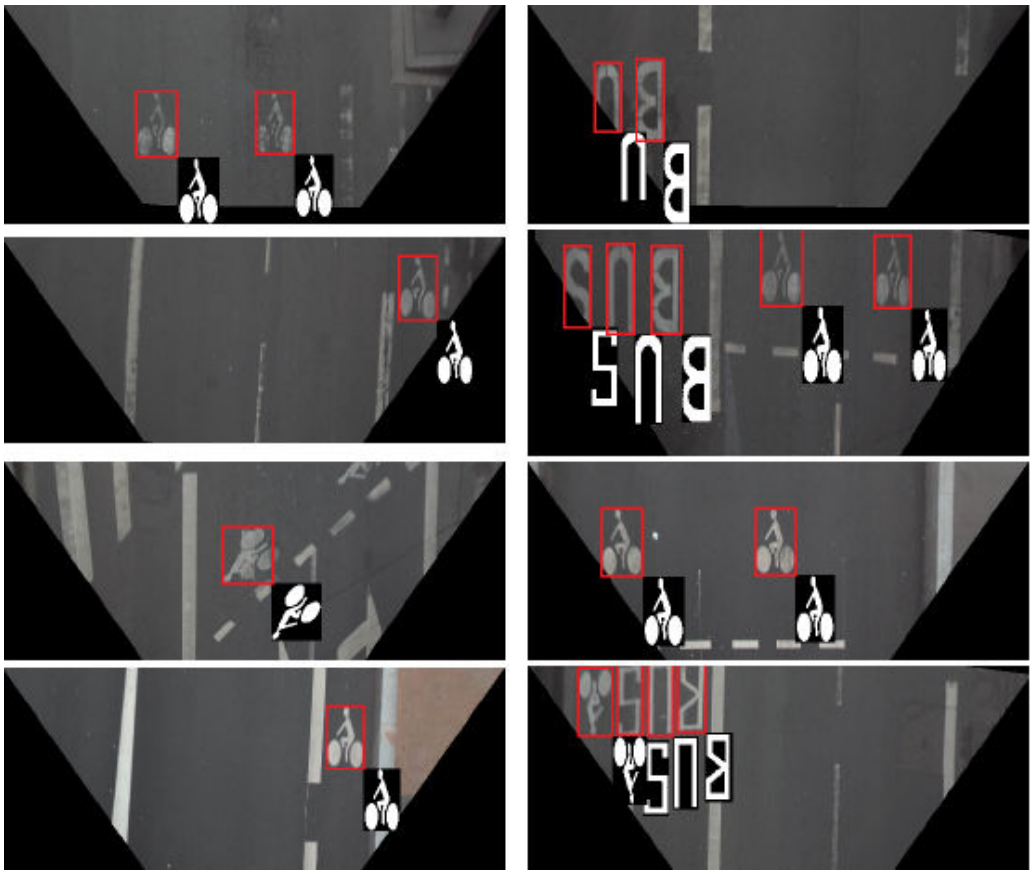


FIGURE *2.31 – Illustration de quelques exemples montrant la bonne identification du mot BUS et du pictogramme-vélo.*

Les résultats présentés dans les figures 2.30 et 2.31 montrent l'efficacité de l'approche automatique proposée.

### 2.2.3 Géolocalisation du marquage identifié

Dans cette section, nous présentons la méthode que nous avons développée pour géolocaliser les marquages routiers identifiés à l'étape précédente [12]. En effet, la méthode proposée est divisée en quatre étapes.

Tout d'abord, nous calculons un point moyen (point A sur la figure 2.32) à partir des centres de $N$ objets reconnus (bandes reconnues du passage piéton par exemple). Le point A obtenu est considéré par la suite comme étant le centre moyen du marquage routier. Deuxièmement, nous calculons la distance en mm entre la caméra et le point $A$ calculé précédemment. En effet, en convertissant les coordonnées (en pixels) de ce point moyen en mm et en se basant sur les coordonnées de la caméra en mm, nous pouvons calculer la distance entre ces 2 points. Cette conversion est réalisée en utilisant les matrices de calibration intrinsèques et extrinsèques présentées dans la section 1.4.5.1 du chapitre 1. Ensuite, nous calculons l'angle d'azimut défini par l'angle entre le nord géographique et la direction de l'objet dans le plan horizontal. Finalement, la méthode de Vincenty [126] est appliquée pour trouver les coordonnées GPS du point moyen $A$.

Un exemple du résultat obtenu en utilisant cette approche de géolocalisation des marquages routiers est illustré dans la figure 2.32. Une fois les marquages routiers sont reconnus (rectangle encadré en rouge), un point fixe $A$ est choisi sur sa surface. Ensuite, les coordonnées GPS de ce point $A$ sont obtenues à partir de la distance et de l'azimut en utilisant la méthode de Vincenty. La figure 2.33 montre le résultat Google Maps correspondant en utilisant les coordonnées GPS calculées $(lat, lon)$.

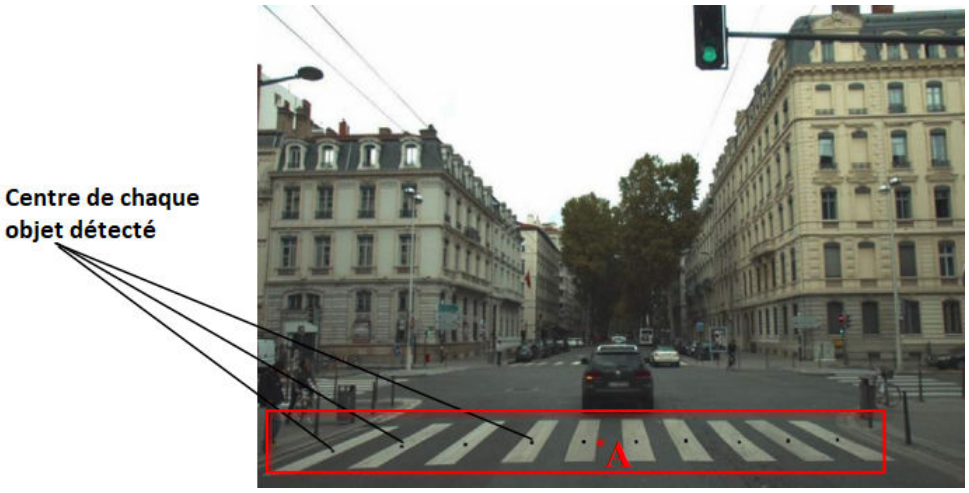



FIGURE *2.32 – Géolocalisation du marquage routier (passage piéton) – "A" est le point moyen calculé à partir de tous les centres des objets reconnus (points noirs) ; Coordonnées GPS : lat=45.7723°et lon=4.8431°.*

La figure 2.32 illustre le résultat obtenu du calcul du centre des différentes bandes du passage piéton. Nous calculons ainsi les coordonnées géographiques du centre. Afin

de vérifier le résultat obtenu, nous affichons ces coordonnées en utilisant la carte fournie par Google Maps.

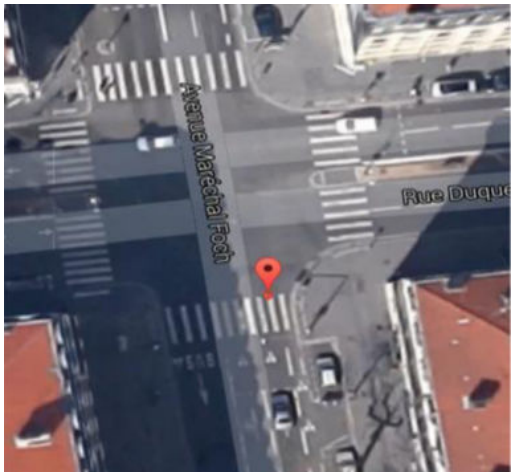



FIGURE *2.33 – Visualisation du passage piéton de la figure 2.32 sur Google Maps.*

### 2.2.4 Quantification de l'usure du marquage routier

Une fois les marquages routiers sont détectés, identifiés ainsi que leurs positions GPS dans différentes images sont calculées, la prochaine étape de notre approche consiste à analyser et à qualifier leur état sur la route.

Nous notons que dans cette section, nous ne présentons que le cas du passage piéton uniquement. En effet, un passage piéton est composé de plusieurs bandes rectangulaires blanches et parallèles à l'axe de la route, ayant en général une longueur variante entre 2.5 m en ville et 6 m en campagne, une largeur de 0.5 m et une distance entre les bandes variante entre 0.5 m et 0.8 m. Ainsi pour évaluer son état sur la chaussée, nous avons proposé de calculer pour chacune de ses bandes reconnues sa longueur, sa largeur, sa surface, son niveau moyen de couleur blanche, etc.

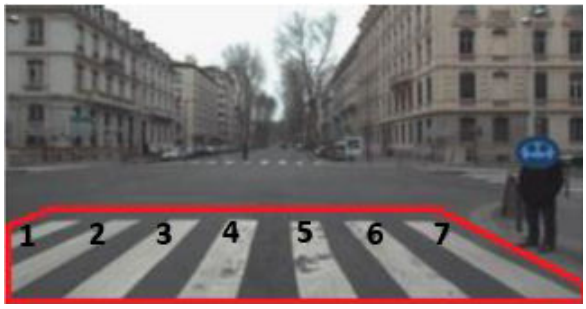



FIGURE *2.34 – Exemple d'un passage piéton identifié constitué de sept bandes juxtaposées.*

| **Bandes** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
|---|---|---|---|---|---|---|---|
| **Largeur (m)** | 0.51 | 0.48 | 0.51 | 0.50 | 0.49 | 0.51 | 0.48 |
| **Longueur (m)** | 2.31 | 3.82 | 4.03 | 3.99 | 3.77 | 3.81 | 3.71 |
| **Distance entre bandes (m)** | 0.56 | 0.54 | 0.50 | 0.56 | 0.49 | 0.53 | × |
| **Niveau moyen de la couleur blanche** | 87.38 | 136.97 | 169.63 | 163.96 | 139.24 | 139.89 | 119.14 |

TABLE *2.2 – Spécificités du passage piéton illustré dans la figure 2.34.*

Le tableau 2.2 illustre les caractéristiques des différentes bandes du passage piéton de la figure 2.34. A ce stade, une première évaluation de l'état du marquage est réalisée par visualisation directe en faisant une comparaison entre les caractéristiques standards (longueur, largeur, etc.) et les valeurs obtenues par notre algorithme. Ainsi, cette comparaison nous a permis dans un premier temps de savoir approximativement l'état du marquage routier (bon état, état moyen, mauvais état). Les résultats obtenus dans ce tableau montrent que le passage piéton de la figure 2.34 est en bon état. En effet, presque toutes ses bandes sont intactes. Par conséquent, aucune opération de maintenance n'est nécessaire dans ce cas.

Enfin, pour avoir une évaluation globale et plus précise, nous avons proposé de calculer le niveau de contraste du marquage par rapport au sol. Autrement dit, une note sur 10 a été calculée et attribuée pour chaque marquage en tenant compte des valeurs affichées dans le tableau 2.2 ainsi que de la valeur moyenne des niveaux de gris de la route détectée $\mu_R$. Le calcul de la note est donné par l'équation 2.7 suivante :

$$note = \ln\left(\frac{X^2 \times \sigma_{(P,R)}}{\sigma_L}\right) \tag{2.7}$$

où $X$ représente le rapport entre la valeur moyenne des niveaux de gris du passage piéton $\mu_P$ et la valeur moyenne des niveaux de gris de la route $\mu_R$, $\sigma_{(P,R)}$ est l'écart-type représentant la variation de la luminosité entre les valeurs moyennes de niveaux de gris du passage piéton et de la route. $\sigma_L$ correspond à l'écart-type des longueurs des bandes. Sur les bases de données d'images, nous avons obtenu expérimentalement des valeurs d'indice comprises dans l'intervalle [0, 10] (voir section 2.3).

Notons que l'évaluation du passage piéton n'est réalisée que sur les bandes reconnues et visibles dans l'image. En effet, les bandes qui ne sont pas visibles complètement dans l'image à traiter pourraient réduire la valeur moyenne des niveaux de gris et ainsi fausser la note que nous cherchons à calculer. En appliquant cette formule sur le passage piéton de la figure 2.34, nous obtenons une note de $\frac{7}{10}$. Cette valeur est obtenue sans prendre en compte les caractéristiques de la bande 1 du passage piéton (la bande 2 est complètement visible dans une image précédente).

## 2.3 Évaluation des performances de la méthode sur des images $VIAPIX^{®}$

Dans cette section, nous présentons les différents tests effectués sur des images acquises par le système $VIAPIX^{®}$ [125]. Comme présenté en annexe B, le module $VIAPIX^{®}$ acquisition nous permet d'obtenir des images à partir d'une caméra orientée vers l'avant et montée sur le toit d'un véhicule. Plusieurs acquisitions dans différentes villes françaises (Brest, Lyon, Limoges, etc.) ont été réalisées.

Quelques résultats peuvent être présentés dans la séquence vidéo suivante : "click". En effet, dans chaque image de cette séquence vidéo, la partie de la route détectée est affichée en bleu. De plus, chaque marquage routier détecté est présenté en vert (respectivement en rouge) lorsque le nombre des objets détectés est supérieur (respectivement égal) à 2. Les coordonnées GPS sont également indiquées sur chaque image.

Afin d'évaluer davantage la bonne performance de notre algorithme, plusieurs tests ont été réalisés sur des images de routes urbaines dans des conditions météorologiques et d'éclairage variables. Les résultats obtenus sont illustrés dans le tableau 2.3.

Les images utilisées ont été capturées à un taux fixé à 25 images/s en utilisant une caméra numérique de résolution $1600 \times 1216$ pixels montée sur un véhicule à une hauteur de 1.7 m dont la vitesse du véhicule varie entre 30 et 50 km/h.

Un exemple de 6500 images acquises dans la ville de Lyon en France (test réalisé sur une section de 6.5 km car les images sont acquises tout les 1 mètres) et contenant 47 marquages routiers a été testé. Une identification réussie de 43 marquages routiers a été obtenue alors que seulement 4 marquages routiers ont été mal reconnus. Pour plus de précision, quelques séries de différents tests effectués en France sont résumés dans le tableau 2.3.

| **Section** | **Condition** | **Nb images** | **Nb marquages** | **Bonne reconnaissance(%)** | **Mauvaise reconnaissance(%)** |
|---|---|---|---|---|---|
| Lyon1 | Temps nuageux | 6500 | 47 | 43 (91.4%) | 4 (8.6%) |
| Lyon2 | Journée ensoleillée | 1000 | 25 | 25 (100%) | 0 (0%) |
| Lyon3 | A midi | 5300 | 32 | 28 (87.5%) | 4 (12.5%) |
| Limoges | Journée ensoleillée | 4200 | 38 | 37 (97.3%) | 1 (2.7%) |

TABLE *2.3 – Résultats de la reconnaissance des passages piétons à l'aide de 4 séquences vidéo différentes.*

La figure 2.35 présente quelques exemples illustrant la bonne reconnaissance du passage piéton dans des images $VIAPIX^{®}$.

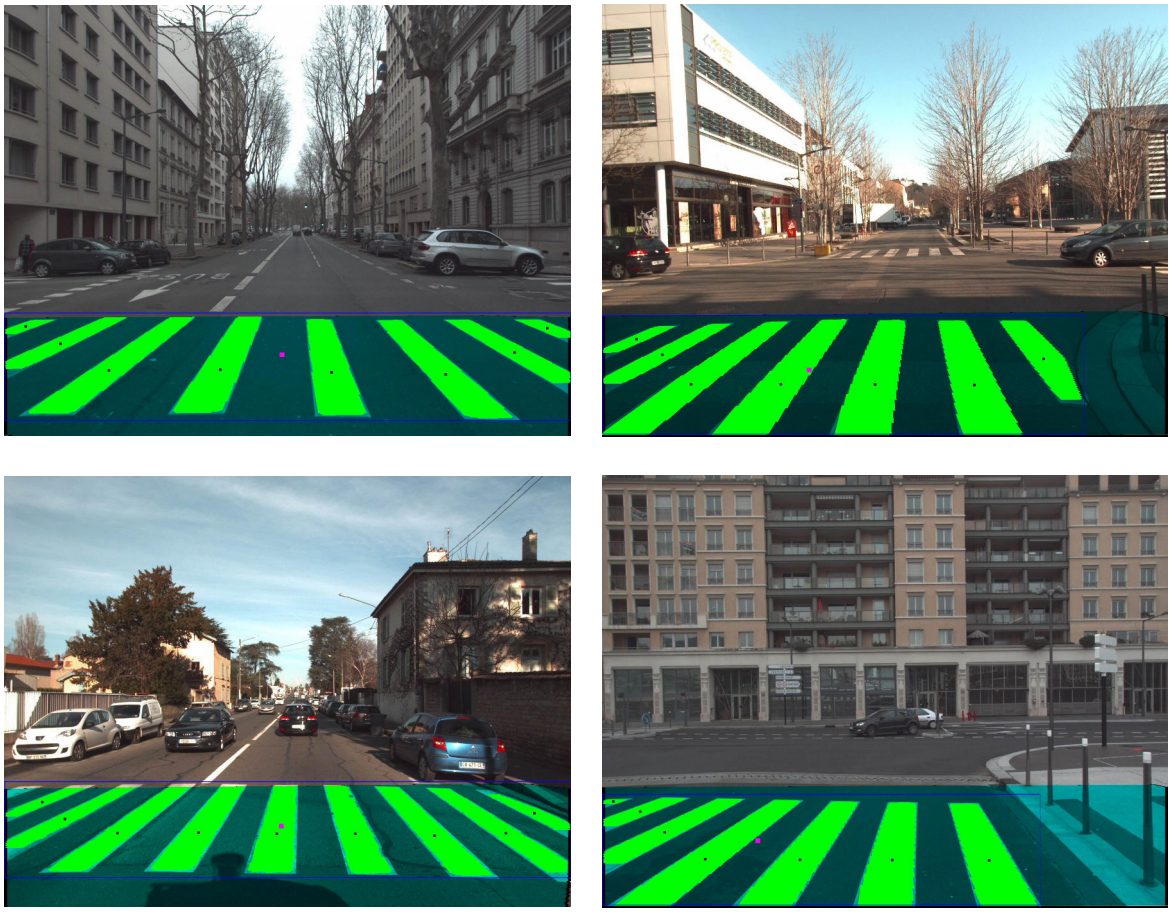


FIGURE *2.35 – Illustration de quelques exemples montrant la bonne identification du passage piéton.*

Une fois que le marquage au sol est identifié, l'étape suivante consiste à calculer ses caractéristiques sur la route (nombre de marques, leur longueur et leur largeur en mètres et en % par rapport à leur valeur maximale respective, etc.) afin d'évaluer leur état sur la chaussée. De plus, en se basant sur l'équation 2.7 obtenue à partir des valeurs calculées, nous vérifions également l'usure de chaque marquage.

Comme il est montré dans la figure 2.34, le marquage numéro 3 (troisième bande du passage piéton) présente la plus petite usure confirmée par visualisation directe. Comme nous pouvons le voir, les marquages numéros 5 et 6 dégradés sont caractérisés par de faibles valeurs du niveau de couleur blanche. Nous notons que le niveau de couleur blanche le plus bas des bandes 1 et 7 est dû à leur détection partielle.

Les différents tests réalisés montrent les bonnes performances et la précision de

notre algorithme pour obtenir des informations précises sur les marquages routiers, leurs conditions de surface et leur géo-référencement. Ci-après nous illustrons quelques exemples montrant l'analyse complète de plusieurs passages piétons. Chacun des passages piétons présenté est évalué par une note sur 10 décrivant son état de dégradation sur la chaussée.

FIGURE *2.36 – Exemple 1 d'un passage piéton quasiment effacé sur la chaussée – lat=45.7833°, lon=4.8487°, note=*$\frac{3}{10}$.

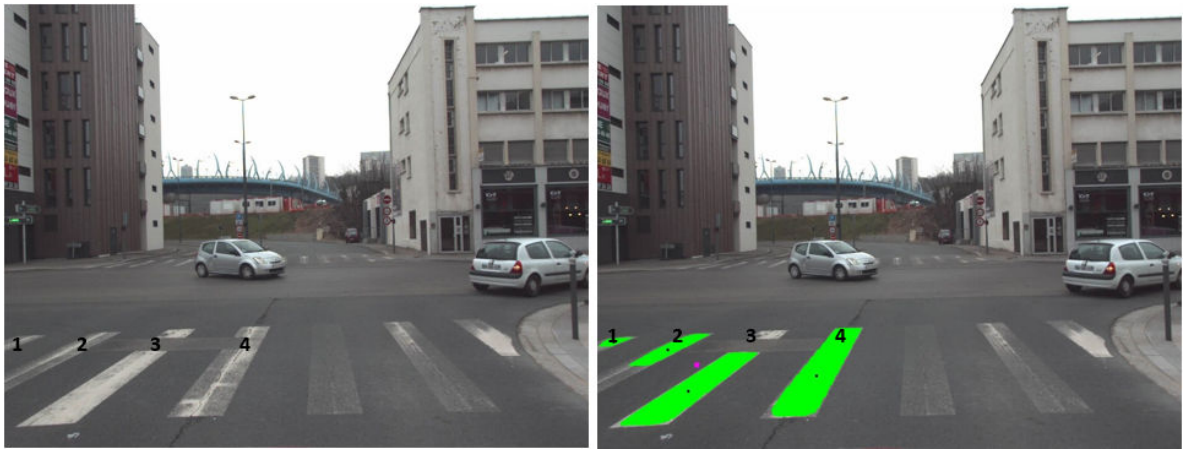



FIGURE *2.37 – Exemple 2 d'un passage piéton en mauvaise état – lat=45.7832°, lon=4.8077°, note=*$\frac{4}{10}$.

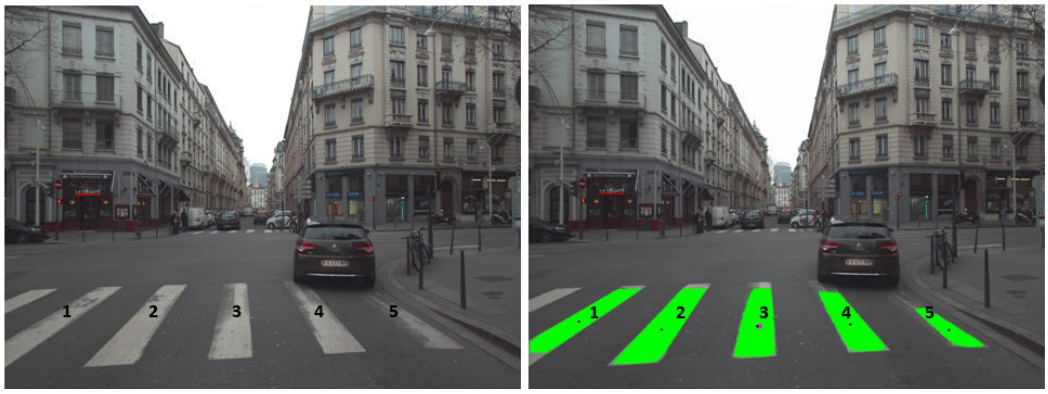



FIGURE *2.38 – Exemple 3 d'un passage piéton ayant quelques bandes détériorées – lat=45.7709°, lon=4.8585°, note=*$\frac{5}{10}$.

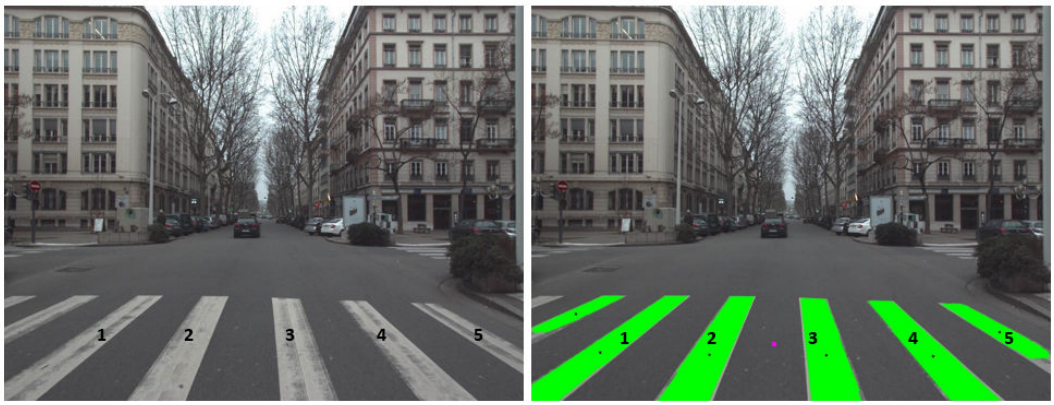



FIGURE *2.39 – Exemple 4 d'un passage piéton en moyen état – lat=45.7693°, lon=4.8490°, note=*$\frac{6}{10}$.

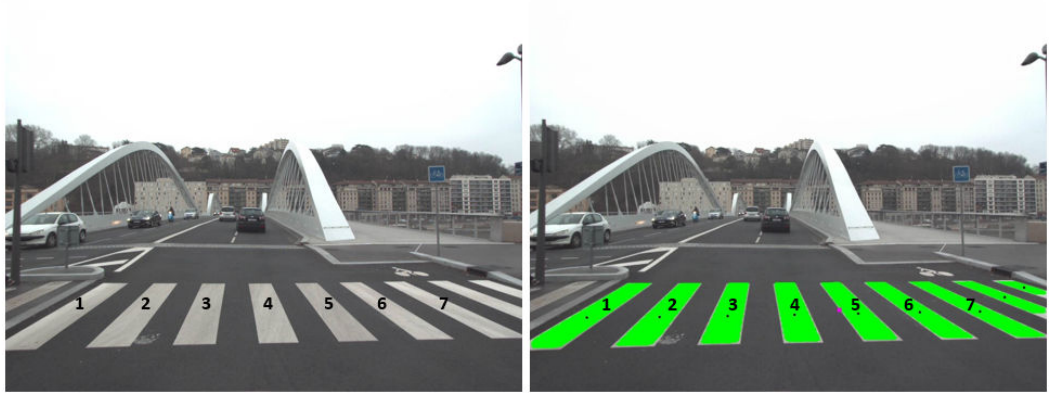



FIGURE *2.40 – Exemple 5 d'un passage piéton en bon état – lat=45.7804°, lon=4.8091°, note=*$\frac{9}{10}$.

## 2.4 Conclusion

L'étude de l'état d'usure de la signalisation horizontale par traitement d'images est un vaste sujet qui nécessite d'aborder différents aspects techniques en amont comme la détection de couleur, la reconnaissance des formes, etc. D'autre part, les marquages au sol sont des objets susceptibles d'être déformés et détériorés par les effets conjugués du trafic et du climat. Dans ce cadre, ce chapitre a présenté la méthode que nous avons développée pour quantifier l'état d'usure du marquage routier. Nous avons abordé le sujet dans sa globalité en commençant par la phase de détection jusqu'à la phase d'évaluation et de quantification.

Tout d'abord, en appliquant la technique de la transformée perspective inverse (IPM), nous avons transformé l'image de paysage routier urbain en une image de chaussée ayant une vue d'oiseau qui élimine les effets de distorsion géométriques de la perspective. Ensuite, basé sur une méthode de reconstruction d'une image orthophoto pour la détection de la route, nous avons réussi à éliminer les objets qui n'appartiennent pas à la chaussée tout en se focalisant sur les zones principales qui pourraient probablement contenir des marquages routiers. Du fait que la route est plus sombre que les marquages au sol, nous notons que pour la détection de la route, nous nous sommes basés sur une méthode de segmentation colorimétrique permettant de détecter les régions sombres dans l'image.

Ensuite, après avoir détecté la route, l'étape de détection des marquages est appliquée. En effet, elle consiste à détecter les objets brillants et lumineux appartenant à la chaussée (objets blancs). Notons que dans le cadre de cette thèse nous n'avons pas traité les marquages de couleurs jaunes.

La méthode décrite pour la reconnaissance des marquages repose sur deux techniques différentes : géométrique et optique. La première technique a été utilisée pour la reconnaissance du passage piéton (nombre de bandes, forme des bandes, etc.) alors que la deuxième technique, basée sur la corrélation optique, a été proposée pour la reconnaissance du mot"BUS" et du pictogramme-vélo sur la route.

La géolocalisation de tous les objets détectés et identifiés est réalisée d'une manière automatique en appliquant la méthode de Vincenty (voir section 2.2.3). Finalement, l'évaluation de l'état des marquages routiers est réalisée pour chaque type de marque identifié en prenant en considération le niveau de gris moyen de la route, le niveau moyen du blanc ainsi que les caractéristiques (longueur, largeur, etc) du marquage en question.

Les différents tests réalisés en utilisant des images $2D$ issues du système $VIAPIX^{®}$ ont montré que la méthode proposée est capable de détecter, reconnaitre, géolocaliser et évaluer les marquages routiers.

# Dégradations de types "Fissures"

# Chapitre 3

## Détection de fissures par sélection de chemins minimaux

### Sommaire


3.1 Introduction . . . . . **86**

3.2 Synoptique de la méthode MPS . . . . . **86**

3.2.1 Étape 1 : Sélection automatique de points amorces . . . . . 87

3.2.2 Étape 2 : Calcul des plus courts chemins entre paires de points amorces . . . . . 89

3.2.2.1 Stratégie de balayage de l'image . . . . . 89

3.2.2.2 Algorithme de calcul du plus court chemin . . . . 90

3.2.2.3 Illustration . . . . . 90

3.2.3 Étape 3 : Sélection automatique des chemins de coût minimal 91

3.2.4 Étape 4 : Raffinage de la segmentation du squelette de fissuration (Post-traitement) . . . . . 93

3.2.4.1 Principe de l'approche basée sur l'analyse de la topologie du squelette . . . . . 93

3.2.4.2 Suppression des objets de petite taille . . . . . 94

3.2.4.3 Suppression des ressemblances et des parties communes entre les chemins adjacents . . . . . 95

3.2.4.4 Mise à jour des points caractéristiques du squelette 96

3.2.4.5 Mise à jour des chemins . . . . . 98

3.2.4.6 Sélection et filtrage des ramifications superflues . 99
3.2.4.7 Sélection et filtrage des boucles superflues . . . . . 101
3.2.4.8 Segmentation finale du squelette de fissuration . . 103
3.2.5 Étape 5 : Estimation de l'épaisseur locale de la fissure . . . 105
3.3 Méthodologie d'évaluation des performances de MPS . . . . . **106**
3.3.1 Pseudo-vérité terrain (PVT) . . . . . 106
3.3.2 Critères d'évaluation . . . . . 107
3.4 Illustration des performances de la méthode MPS . . . . . **110**
3.5 Conclusion . . . . . **119**

## 3.1 Introduction

La méthode MPS (Minimal Path Selection) est une méthode récente de traitement d'images, qui a été développée pour la segmentation automatique de fissures sur des images de chaussée [117,123,135]. Cette technique, basée sur un algorithme de sélection des chemins les plus courts (shortest path algorithm de type Dijkstra), a pour effet de favoriser la connexité entre les pixels "fissures" grâce à un critère de minimisation des valeurs de niveaux de gris. Elle comporte deux phases principales : une phase de segmentation qui permet d'identifier le squelette de la fissure et une phase de post-traitement qui consiste à affiner le résultat de la segmentation et à estimer la largeur de la fissure.

Les paragraphes 3.2.1 à 3.2.5 décrivent les différentes étapes de la méthode MPS pour la détection automatique des fissures en mode non-supervisé (sans aucune intervention humaine).

## 3.2 Synoptique de la méthode MPS

La méthode MPS se compose de deux phases principales, comme le montre la figure 3.1. La première phase (étapes a à c de la figure 3.1) permet la segmentation du squelette de la fissure (de largeur d'un pixel) dans l'image. La deuxième phase (étapes d et e de la figure 3.1) consiste à affiner le squelette de fissure et à estimer sa largeur localement.

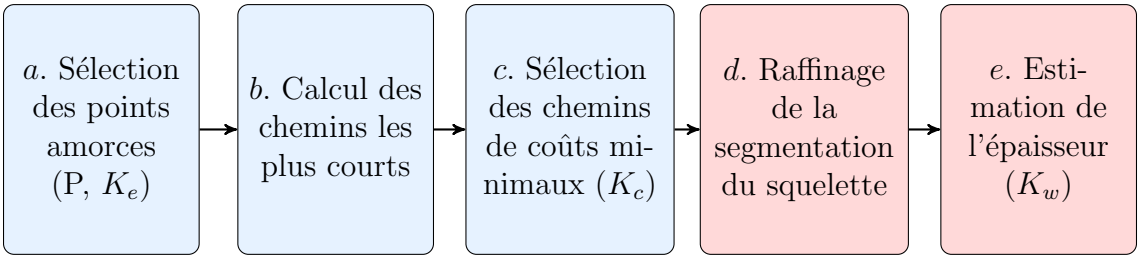



FIGURE *3.1 – Synoptique de la méthode MPS.*

Les performances de MPS dépendent principalement de la qualité de la segmentation du squelette de la fissure, réalisée dans les trois premières étapes (a, b et c). La phase de post-traitement, définie par les étapes d et e, améliore les performances de cette segmentation d'environ 20% à 30%.

Les sections 3.2.1 à 3.2.5 présentent les 5 étapes de l'algorithme MPS et illustrent son fonctionnement sur des sous-images de chaussée.

### 3.2.1 Étape 1 : Sélection automatique de points amorces

Les points amorces servent à initier la segmentation des fissures dans l'image à l'étape 2 et 3. Ils correspondent aux pixels les plus sombres de l'image, car par hypothèse, ces derniers ont une plus grande probabilité d'appartenir à une fissure.

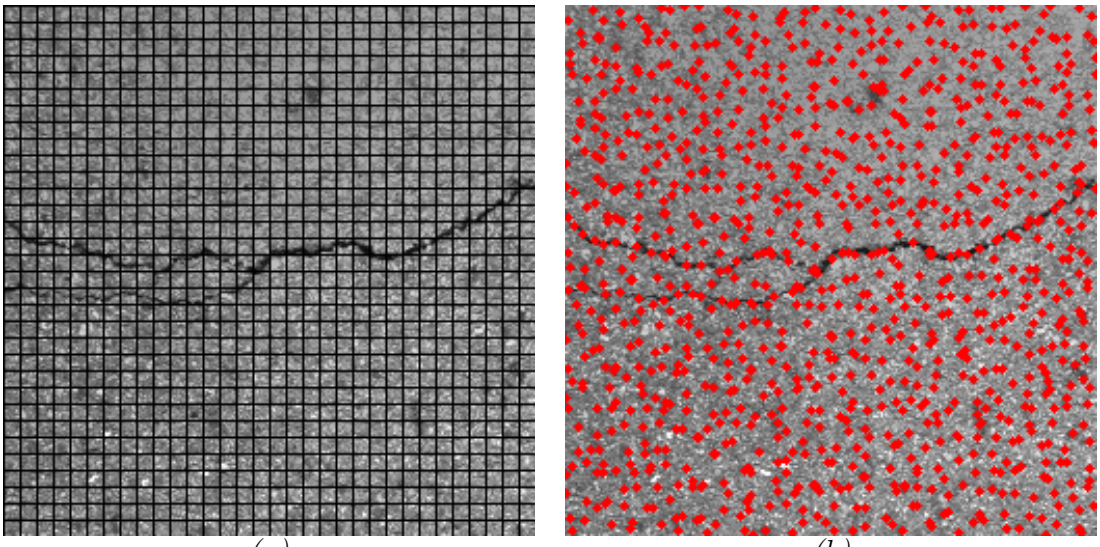
Le fait de traiter toute l'image et d'analyser un grand nombre de pixels nécessite un temps de calcul très important. En effet, la sélection des points amorces situés à l'extérieur de la fissure dans une image, est considérée comme une étape inutile vu qu'elle sera capable de générer des fausses alarmes lors de la détection des fissures. C'est pour cela l'objectif de cette étape est de réduire au maximum le nombre de ces pixels afin de réduire fortement le temps d'exécution et en faciliter le processus de calcul des chemins les plus courts à l'étape suivante.

Dans ce cadre, la sélection des points amorces se déroule en deux étapes.

Nous sélectionnons dans un premier temps les minima locaux (les points marqués en rouge dans la figure 3.2.b) qui correspondent aux pixels les plus sombres dans une ROI (Region Of Interest) de taille $P \times P$ (figure 3.2.a) [123]. Le nombre maximal des points amorces dans une image est défini par l'équation :

$$nb_{A_{max}} = \frac{N}{P^2} \tag{3.1}$$

avec $N$ est le nombre total de pixels de l'image, et $P = 8$ est la taille du bloc (ROI).



*(a)* *(b)*

FIGURE *3.2 – (a) Image originale de taille* $256 \times 256$ *pixels - (b) Sélection des minimaux locaux (ROI de taille* $8 \times 8$ *pixels).*

Ensuite, et parmi ces minima locaux, MPS conserve les pixels dont la valeur de niveaux de gris (intensité/luminosité) est inférieure au seuil $T_e$ suivant, déterminé à partir de l'histogramme de l'image globale (figure 3.3.a) :

$$T_e = \mu - K_e \times \sigma \tag{3.2}$$

où $\mu$ et $\sigma$ représentent les valeurs de la moyenne et de l'écart type de l'image et $K_e$ est une constante ajustée de manière à conserver le taux de vrais positifs (bonne détection) sans augmenter le taux de faux négatifs (pixels fissures non détectés).

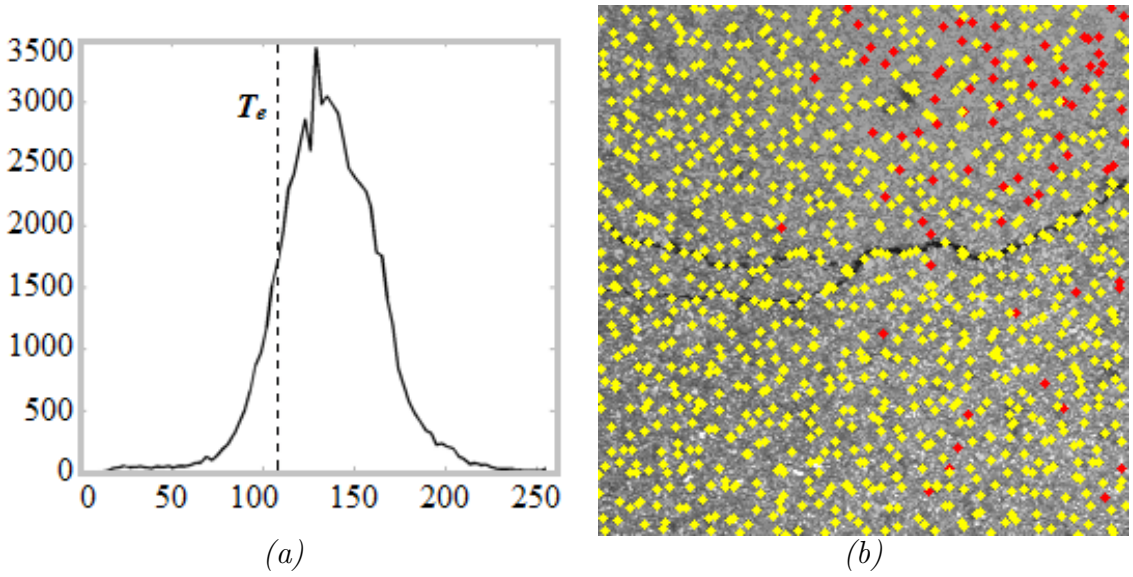



FIGURE *3.3 – (a) Histogramme des niveaux de gris de l'image originale et seuil $T_e$ pour $K_e$=1 - (b) Sélection des points amorces (pixels jaunes).*

La figure 3.3 montre que l'application du seuil (équation 3.2) permet la suppression de quelques pixels dans la texture de l'image (pixels rouge de la figure 3.3.b).

Le résultat obtenu montre que seulement quelques pixels sélectionnés sont à l'intérieur de la fissure, les autres points amorces sont en fait des minima locaux dans la texture d'image ; ils représentent des sources potentielles de fausses alarmes.

Les points amorces détectés peuvent représenter aussi bien des fissures que des zones sombres de l'image. Afin de faire face à ce problème, la notion du chemin est introduite. En effet, l'introduction de cette notion aux deux étapes suivantes permettra de séparer plus efficacement les deux catégories de pixels, i.e., chemin-fissure vs. chemin-texture.

### 3.2.2 Étape 2 : Calcul des plus courts chemins entre paires de points amorces

L'étape 2 de la méthode MPS consiste à calculer les chemins les plus courts entre toutes les paires de points amorces sélectionnés à l'étape 1. Nous détaillons dans ce paragraphe la stratégie choisie pour réaliser ce calcul de manière exhaustive sur toute l'image, et précisons le choix de l'algorithme pour réaliser le calcul du plus court chemin.

#### 3.2.2.1 Stratégie de balayage de l'image

La stratégie "étoile" qui a été choisie dans [123] pour calculer les chemins les plus courts dans l'image à partir d'un point central dans un ROI de taille $3P \times 3P$. Dans cet environnement, un point amorce source central est entouré au maximum par huit points voisins destinations. Le nombre maximal de chemins minimaux que nous devons calculer dans une image est obtenu en négligeant le recouvrement entre les ROIs ; il est ainsi égal à :

$$nb_{Ch_{max}} \sim \frac{8 \times N}{P^2} \tag{3.3}$$

avec $N$ le nombre de pixels dans l'image et $P$ la taille du bloc.

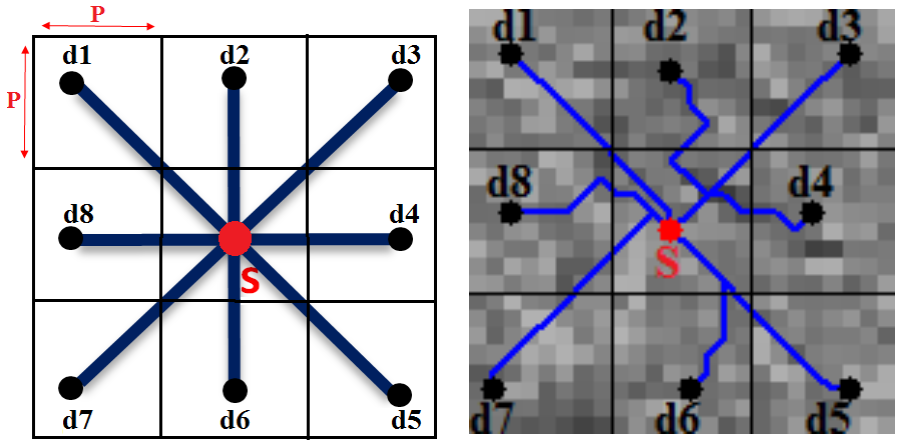



FIGURE *3.4 – Illustration de la stratégie utilisée pour le calcul des chemins les plus courts dans l'image – (a) Stratégie étoile dans une fenêtre* $3P \times 3P$ *appliquée pour calculer les chemins les plus courts entre une source S (en rouge) et ses huit voisins (d1...d8) - (b) Illustration des huit chemins obtenus avec Dijkstra (SSSP) dans une fenêtre* $3P \times 3P$.

En se basant sur la stratégie présentée dans la figure 3.4, le calcul se poursuit dans toute l'image en décalant les ROIs de $P$ pour assurer un recouvrement ainsi une recherche exhaustive de tous les chemins dans l'image.

#### 3.2.2.2 Algorithme de calcul du plus court chemin

Plusieurs algorithmes existent dans la littérature pour calculer le plus court chemin entre deux sommets qui minimise le coût d'une certaine fonction. En supposant que les fissures à détecter correspondent dans l'image de chaussée à des chemins plus sombres que la texture, le principe consiste à rechercher les chemins qui minimisent le cumul des niveaux de gris le long du chemin [114, 118, 120] (voir équation 1.23).

Appliqué à une image, le critère défini par l'équation 1.23 permet de minimiser la distance de Manhattan (ou taxi-distance) sous une norme L1. À l'usage, ce critère permet, d'après [123], de mieux suivre la trajectoire chaotique de la fissure sur la chaussée. La comparaison de différents algorithmes (de normes L1 et L2) est illustrée dans [123, 136–138].

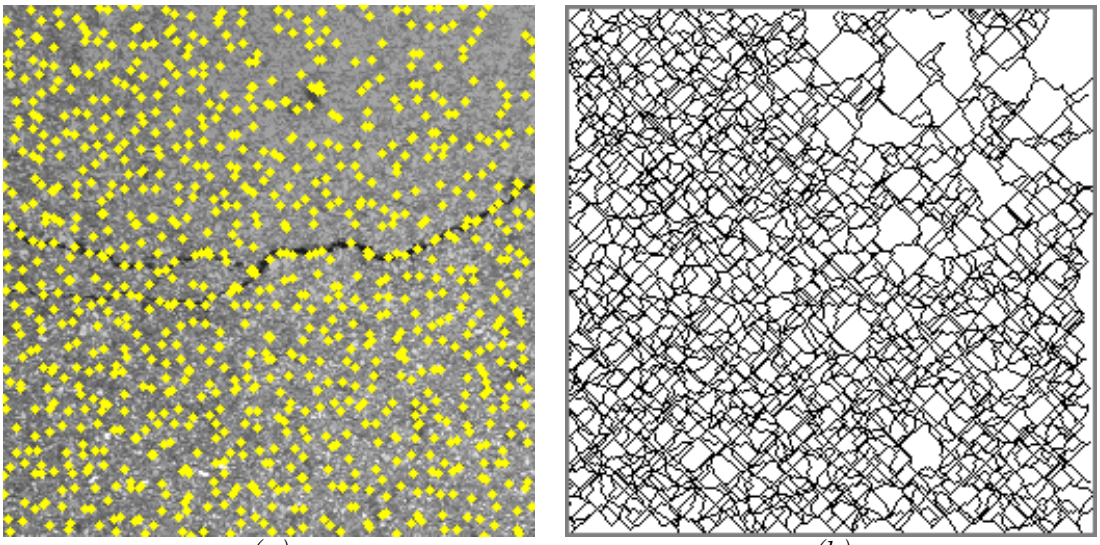
Dans le cadre de la recherche de chemins dans le ROI $3P \times 3P$ introduit au paragraphe précédent, nous distinguons schématiquement deux catégories d'algorithmes : one-to-one (SPSP ou Single Pair Shortest Path) et one-to-all (SSSP ou Single Source Shortest Path). La première catégorie (one-to-one) nécessite d'itérer le calcul de plus court chemin entre le point source central du ROI et les 8 voisins du ROI $3P \times 3P$. La deuxième catégorie (one-to-all) lance la recherche du plus court chemin dans toutes les directions à partir du point central du ROI, jusqu'à atteindre le dernier des 8 voisins.

Parmi les algorithmes de la littérature, l'algorithme de Dijkstra [120] permet d'une part de minimiser la norme L1 (équation 1.23)) et d'autre part, fonctionne pour les deux catégories d'algorithmes one-to-one et one-to-all de recherche du plus court chemin sans nécessiter de modifications du synoptique. Les auteurs dans [123, 136] ont montré que la stratégie one-to-all est la plus rapide (d'un facteur $\sim 4$) pour rechercher les chemins les plus courts dans le contexte de la stratégie présentée dans la section 3.2.2.1 (ROI de taille $3P \times 3P$).

#### 3.2.2.3 Illustration

La figure 3.4.b illustre l'application de cette stratégie dans un cadre $3P \times 3P$ d'une image de chaussée ($P = 8$). Le point d'extrémité central (S) en rouge sert comme de point source pour initialiser la recherche des huit chemins à l'aide de l'algorithme de Dijkstra selon la stratégie one-to-all (SSSP).

La figure 3.5.b illustre le résultat de l'algorithme de Dijkstra appliqué aux différents points amorces obtenus à l'étape précédente (figure 3.5.a).



(a) (b)

FIGURE *3.5 – (a) Points amorces sélectionnés à l'étape 1 - (b) Chemins calculés entre les points amorces par l'algorithme de Dijkstra.*

À la fin de cette étape, nous disposons des chemins élémentaires d'une taille limitée par le cadre $3P \times 3P$ du ROI. Par comparaison avec la pseudo-vérité terrain (PVT) de l'image, définie dans la section 3.3.1 (figure 3.23.b), la plupart des chemins détectés sont situés dans la texture de l'image, tandis que quelques chemins sont situés soit entièrement, soit partiellement à l'intérieur de la fissure. L'étape 3 permet ainsi de sélectionner les chemins-fissures.

### 3.2.3 Étape 3 : Sélection automatique des chemins de coût minimal

Cette étape permet de sélectionner parmi les chemins élémentaires représentés à la figure 3.5.b, les chemins qui sont totalement ou partiellement à l'intérieur des fissures. Chacun de ces chemins est caractérisé par sa longueur en pixels et sa valeur de coût qui correspond à la moyenne des niveaux de gris des pixels qui le forment, selon la formule suivante :

$$C(p_{ij}) = \frac{1}{card(p_{ij})} \sum_{m=i}^{j} I(x_m) \tag{3.4}$$

où $p_{ij}$ est un chemin entre le point source $x_i$ et le point destination $x_j$ et $x_m$ est un pixel du chemin.

La distinction entre chemins-fissures et chemins-texture repose sur les statistiques des valeurs de coûts $C(p_{ij})$. L'histogramme associé (figure 3.6.a) présente une distribution quasi-bimodale, qui, en comparaison avec l'histogramme des niveaux de gris de

l'image (figure 3.3.a), est plus facile à interpréter. Par hypothèse, les chemins-fissures correspondent aux chemins de coût plus faible dans la partie gauche de l'histogramme. L'objectif est alors de garder les chemins d'intensité moyenne les plus faibles.

Pour ce faire, l'étape 3 de l'algorithme consiste à sélectionner les chemins dont le coût $C(p_{ij})$ est inférieur au seuil photométrique $T_c$, défini par l'équation 3.5 comme suit :

$$T_c = \mu_c - K_c \times \sigma_c \tag{3.5}$$

avec $\mu_c$ et $\sigma_c$ sont les valeurs de la moyenne et de l'écart type des coûts des chemins et $K_c$ est une constante ajustée de manière à optimiser le coefficient de similarité (diminution des fausses alarmes).

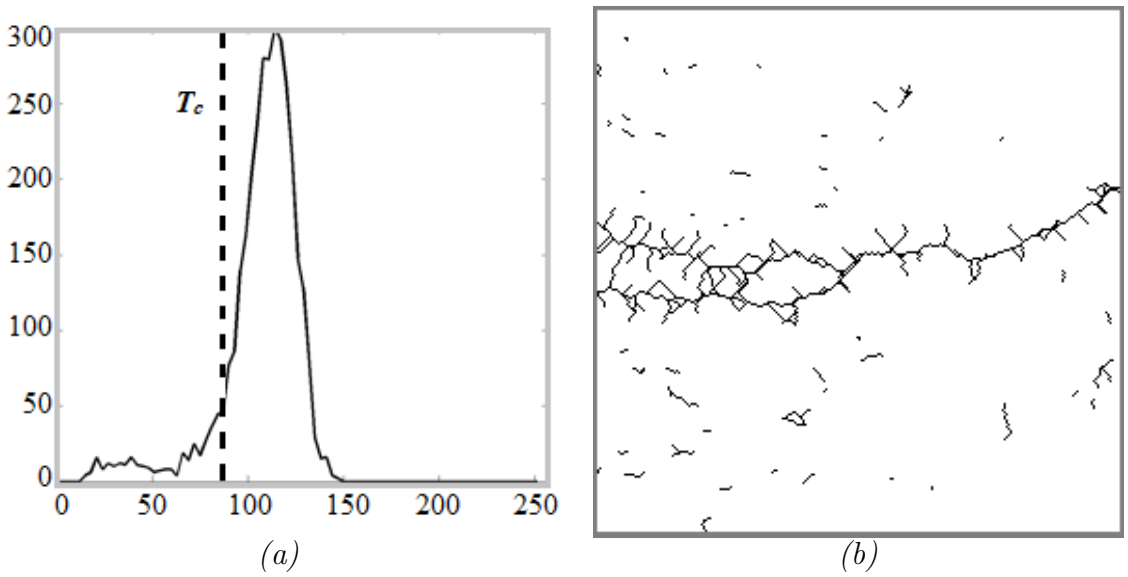



FIGURE *3.6 – (a) Histogramme des coûts des chemins et position du seuil $T_c$ pour $K_c = 1$ - (b) Sélection des chemins de coût minimal.*

La figure 3.6 illustre le résultat obtenu lors de l'application du seuillage sur les valeurs de coûts des chemins. Le seuillage a permis de réduire le nombre de fausse alarme en comparaison avec le résultat illustré dans la figure 3.5.b, et de faire apparaitre nettement à la figure 3.6.b le squelette de fissuration. Ces résultats nous permettent de conclure qu'il existe trois types de chemins sélectionnés à l'issue de l'étape 3 :

- Des chemins qui sont totalement à l'intérieur de la fissure : bonnes détections.
- Des chemins qui sont dans la texture : fausses détections.
- Des chemins qui sont partiellement dans la fissure, et génèrent des artefacts le long du squelette (voir étape 4).

### 3.2.4 Étape 4 : Raffinage de la segmentation du squelette de fissuration (Post-traitement)

#### 3.2.4.1 Principe de l'approche basée sur l'analyse de la topologie du squelette

Le résultat de la segmentation de l'image présente plusieurs types d'artefacts, qui induisent des fausses alarmes. La majorité des artefacts dans la figure 3.6.b correspondent aux ramifications et aux boucles le long du squelette, et sont engendrées par le fait que certains chemins sélectionnés sont partiellement dans la fissure.

Pour améliorer la qualité de la segmentation et raffiner le squelette de fissuration, les auteurs dans [117, 123, 135] proposent plusieurs étapes successives.

Une première étape permet d'éliminer les segments de petites tailles. Ensuite, une seconde étape permettant d'établir la connexité entre les éléments segmentés est appliquée afin de réaliser un (des) squelette(s) continu(s) de taille plus grande (que les ROIs $3P \times 3P$ utilisées aux étapes 2 et 3) et sans redondance. Une analyse topologique permet une mise à jour des points caractéristiques (points d'extrémités, points de ramifications à une ou plusieurs branches, i.e., rond point) qui structurent ce nouveau squelette. Chaque partie du squelette est ainsi analysée de nouveau, à l'aide du critère de l'étape 3 (seuil Tc). Les boucles et les ramifications superflues sont en grande partie éliminées par cette analyse.

Dans [123], la combinaison et l'ordre d'enchainement des post-traitements de l'étape 4 ont permis de définir trois stratégies de raffinement du squelette, dont l'une procède par itération jusqu'à stabilisation de la forme du squelette. La stratégie "R-mixte" a été sélectionnée comme meilleur compromis. Elle consiste ainsi à enchaîner la stratégie "R-continuité" (qui préserve la continuité du squelette) et la stratégie "R-seuillage" (qui élimine toute portion de squelette dont le coût est supérieur au seuil $T_c$).

L'étape 4 permet d'aboutir à un squelette de fissuration simplifié, dont le nombre de pixels faux positifs (fausse alarme) a diminué. Pour plus de précision, un synoptique illustrant les différentes étapes réalisées pour corriger le squelette de fissuration, est présenté dans la figure 3.7.

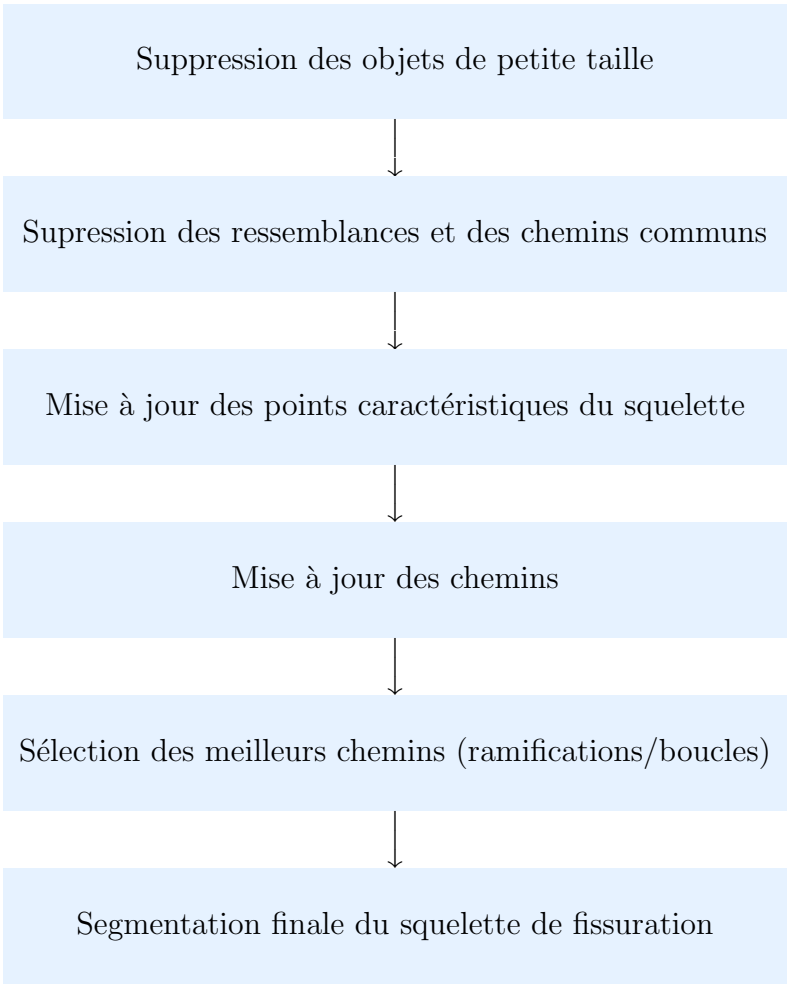
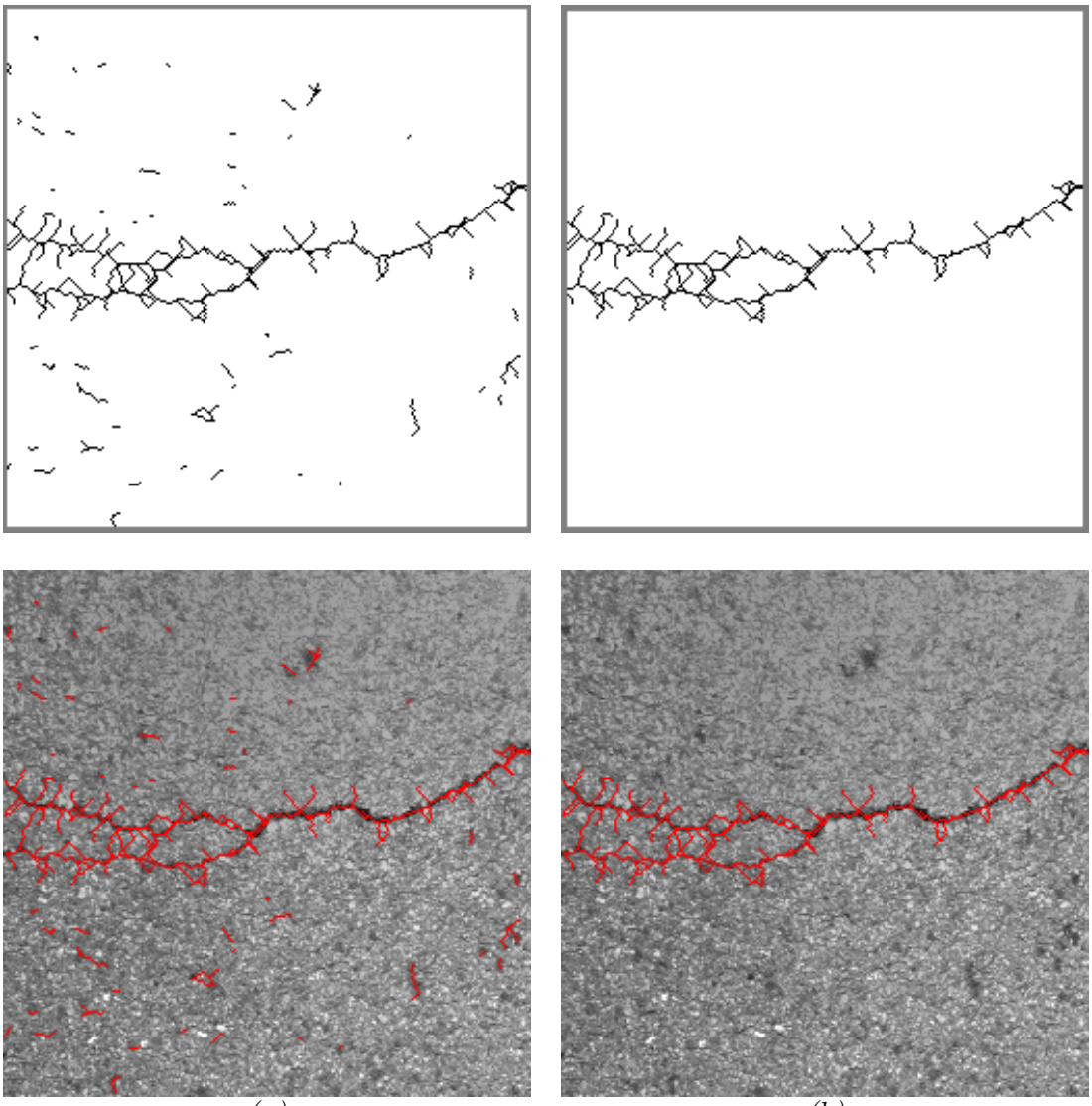



FIGURE *3.7 – Synoptique du post-traitement du squelette de fissuration – Les étapes à suivre pour corriger le squelette de fissuration obtenu à l'issue de l'étape 3 de l'algorithme.*

#### 3.2.4.2 Suppression des objets de petite taille

Cette étape consiste à supprimer les composantes connexes de taille inférieure à $T_s = 60$ pixels, ce qui correspond à des objets d'environ 120 mm de longueur sur les images Aigle-RN.

Ce traitement a pour effet (figure 3.8.b) de supprimer de nombreux segments faux positif (fausse alarme) dans la texture, ainsi que des fissures de petites tailles et des composantes dans des petites zones circulaires sombres.



(a) (b)

FIGURE *3.8 – (a) Squelette de fissuration à l'issue de l'étape 3 de l'algorithme - (b) Squelette de fissuration après l'élimination des chemins de longueur* $< T_s$.

#### 3.2.4.3 Suppression des ressemblances et des parties communes entre les chemins adjacents

Afin d'obtenir un squelette sans redondance, une deuxième étape consiste à supprimer les parties communes entre les chemins voisins. En effet, elle permet d'éviter d'analyser et de comptabiliser les mêmes pixels plusieurs fois. La figure 3.9 présente les quatre types de chemins à éliminer dans cette étape.

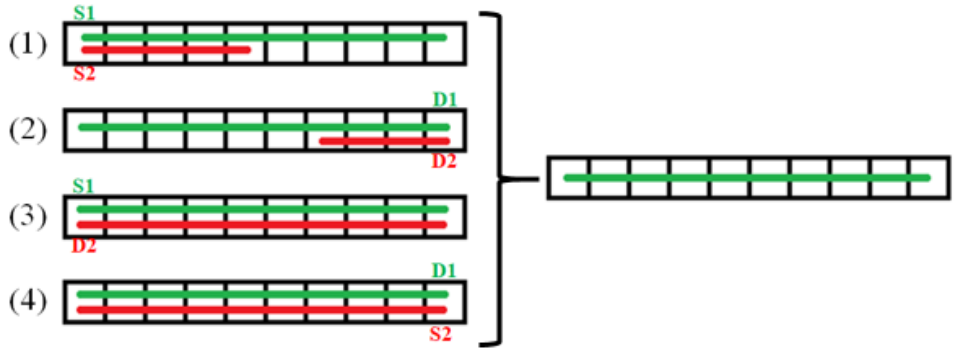



FIGURE *3.9 – Suppression des parties communes entre les chemins adjacents.*

D'après la figure 3.9, cette étape permet de réduire le temps de calcul de l'algorithme du fait qu'elle élimine les informations répétées (chemins en rouge) et ainsi évite de les traiter plusieurs fois dans l'image.

#### 3.2.4.4 Mise à jour des points caractéristiques du squelette

L'étape suivante consiste à faire une mise à jour des points caractéristiques qui structurent le squelette de fissuration, à savoir les points extrémités, les points de jonctions/ramifications à une ou plusieurs branches (ou rondpoints).

À l'exemple de la figure 3.10, pour qu'un point du squelette de fissuration soit considéré comme un pixel extrémité $e_i$ (respectivement rondpoint $r_i$), il faut que le nombre des pixels voisins à ce dernier soit inférieur ou égal à 3 (respectivement supérieur à 3).

| | | |
|---|---|---|
| 0 | 1 | 1 |
| 0 | $e_i$ | 1 |
| 0 | 0 | 0 |

*(a)*

| | | |
|---|---|---|
| 1 | 0 | 1 |
| 0 | $r_i$ | 0 |
| 1 | 0 | 1 |

*(b)*

FIGURE *3.10 – (a) Définition d'un pixel extrémité $e_i$ - (b) Définition d'un pixel rond point $r_i$.*

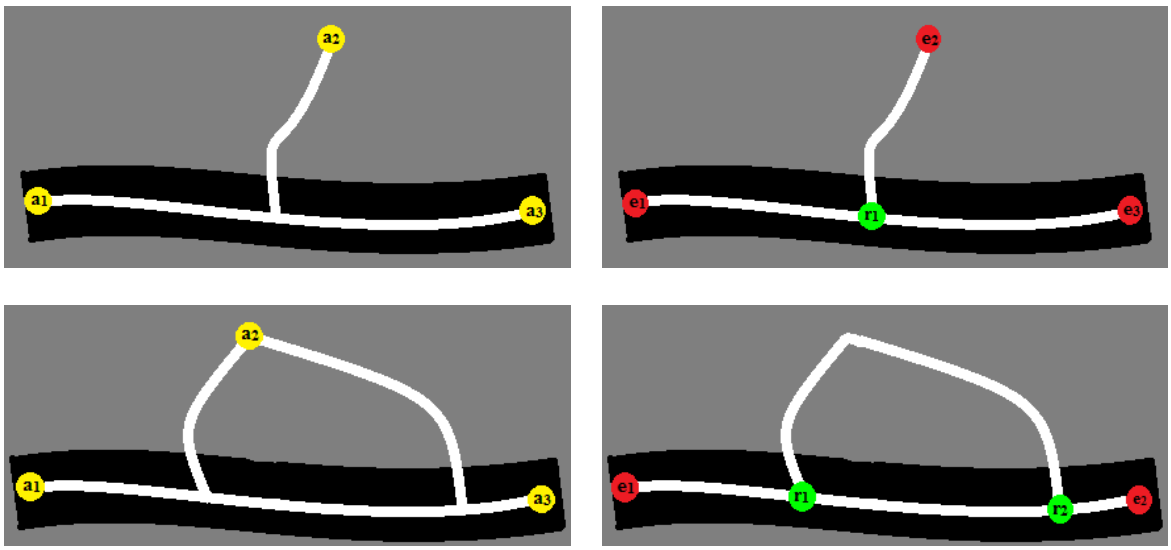



FIGURE *3.11 – Mise à jour des points amorces $a_i$ en jaune (points extrémités $e_i$ en rouge et rondpoints $r_i$ en vert).*

Un traitement particulier concerne les points extrémités isolés $e_i$, qui touchent un point de ramification $r_i$, comme illustré à la figure 3.12. La micro-ramification que cela produit est jugé inutile ; nous préférons ainsi simplifier la forme du squelette du fissuration et éliminer ce point.

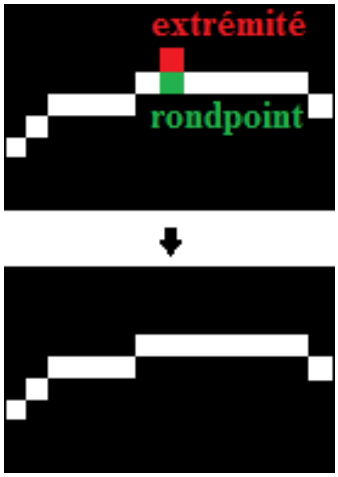



FIGURE *3.12 – Illustration du filtrage d'un point extrémité isolé.*

La figure 3.11 présente un exemple de mise à jour des points amorces dans une portion du squelette. Les anciens points caractéristiques en jaunes sont remplacés par de nouveaux points extrémités ($e_i$ en rouge) et des nouveaux points de jonctions (ou rondpoints $r_i$ en vert). La figure 3.13 illustre l'application de cette méthode de mise à jour sur le squelette de fissuration obtenu à la figure 3.8.b.

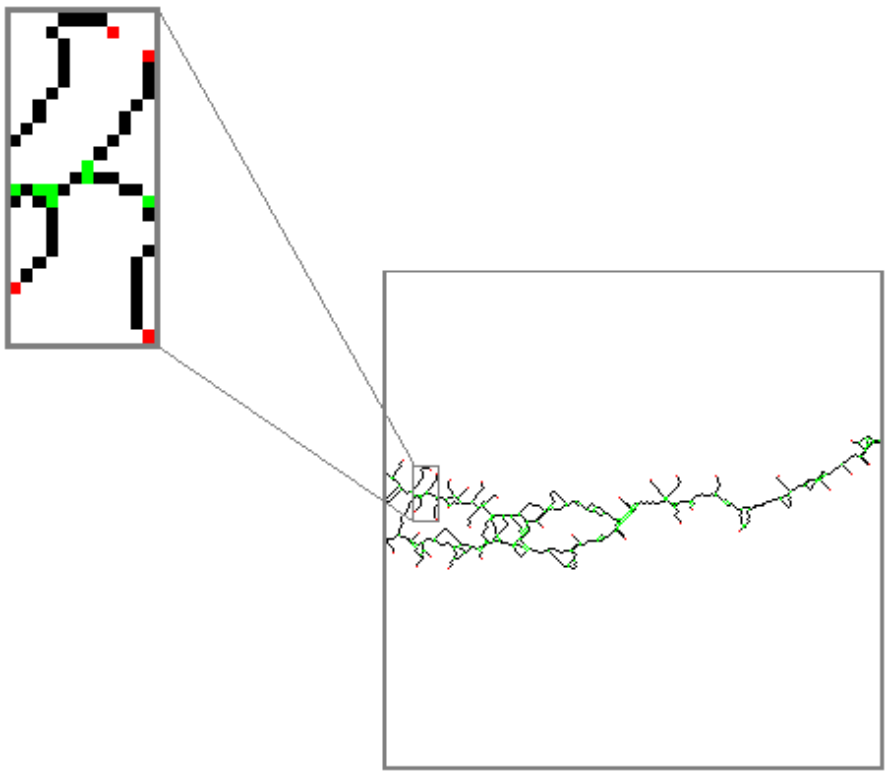


FIGURE *3.13 – Mise à jour des points amorces : points extrémités (en rouge) et points de jonctions (en vert).*

#### 3.2.4.5 Mise à jour des chemins

La mise à jour des points caractéristiques conduit à une nouvelle structure du squelette de fissuration. Il convient ensuite de mettre à jour les chemins du squelette, qui sont définis par hypothèse entre deux points caractéristiques, dont l'un sert de point source du chemin et le second sert de point destination. Nous distinguons à ce niveau les trois types de chemins suivants :

- Les ramifications qui correspondent aux chemins entre une extrémité $e_i$ (en rouge) et un point de ramification $r_i$ (en vert), à l'exemple de la figure 3.14.a.

- Les boucles normales qui sont formées par deux chemins entre deux points de ramifications différents $r_i$ (en vert), à l'exemple de la figure 3.14.b.

- Les boucles fermées qui correspondent à un chemin ayant le même point de ramification $r_i$ (en vert), qui sert à la fois de point source et de point destination sur la figure 3.14.c.

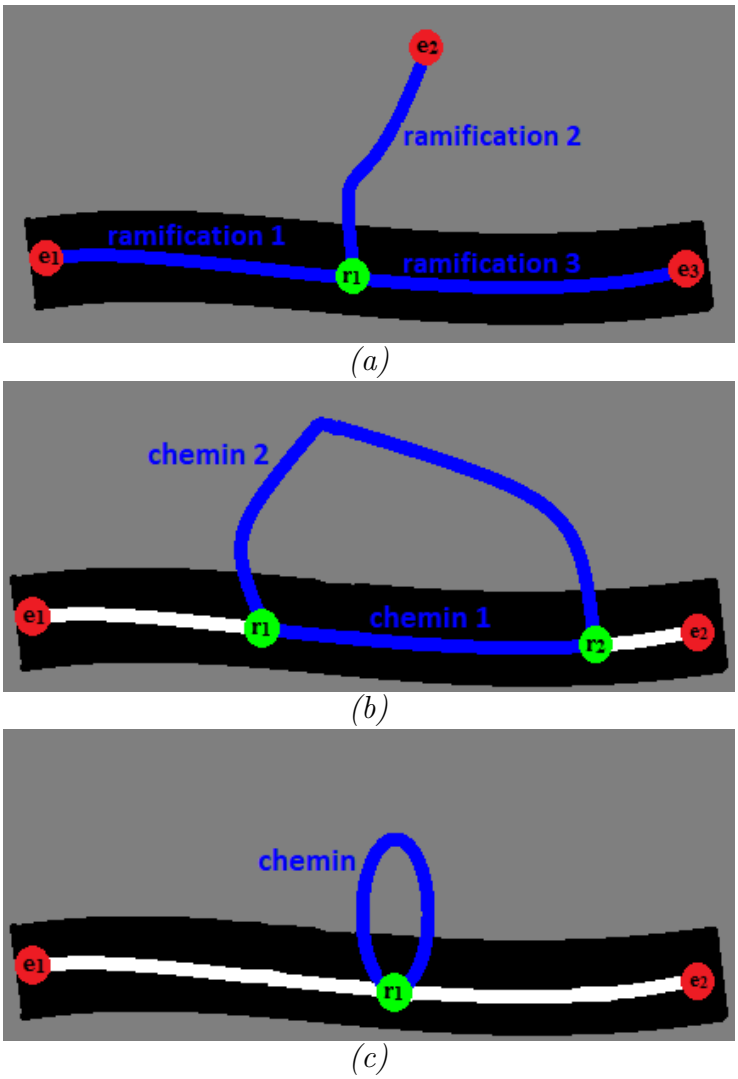



FIGURE *3.14 – (a) Chemins d'une ramification - (b) Chemins d'une boucle normale - (c) Chemins d'une boucle fermée.*

Un exemple de sélection des ramifications (respectivement des boucles) dans le squelette de fissuration est illustré à la figure 3.16 (respectivement figure 3.19).

#### 3.2.4.6 Sélection et filtrage des ramifications superflues

L'objectif de cette étape est de conserver parmi toutes les ramifications sélectionnées dans l'image (voir figure 3.16.a), celles qui représentent une fissure sur la chaussée. En effet, pour y arriver le coût de chaque chemin (équation 3.4) est recalculé et comparé à la valeur du seuil $T_c$, défini à l'étape 3 (paragraphe 3.2.3) de l'algorithme MPS. Un chemin (ramification) est considéré comme superflu (et donc à éliminer du squelette) lorsque le coût associé est supérieur au seuil $T_c$. En pratique, la comparaison avec la PVT montre que dans une ramification superflue, seulement un des deux points amorces

appartient à la fissure [117] ; en conséquence, le chemin minimal est dans la texture, et crée des pixels faux positifs (fausse alarme).

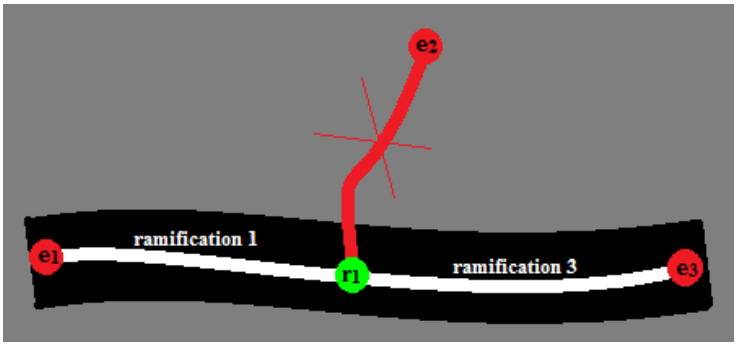
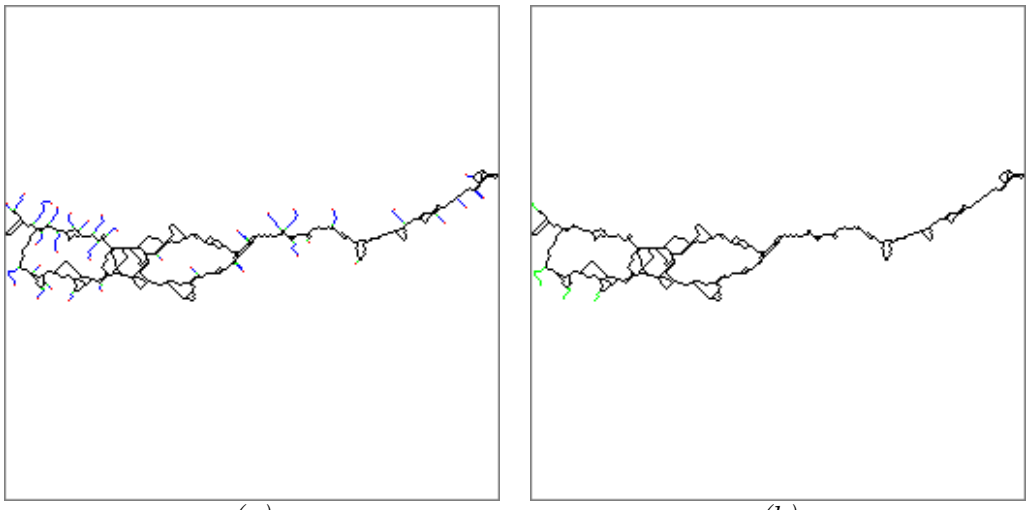



FIGURE *3.15 – Ramifications après le seuillage $T_c$.*

La figure 3.15 illustre le principe de la procédure de filtrage de la ramification en supposant que le coût $C(p_{r_1e_2})$ soit supérieur à $T_c$. La figure 3.16.b illustre le résultat de cette étape sur le squelette initial de la figure 3.16.a. Par comparaison, les ramifications qui ont été conservées (i.e., dont le coût moyen est inférieur à $T_c$) figurent en vert.



(a) (b)

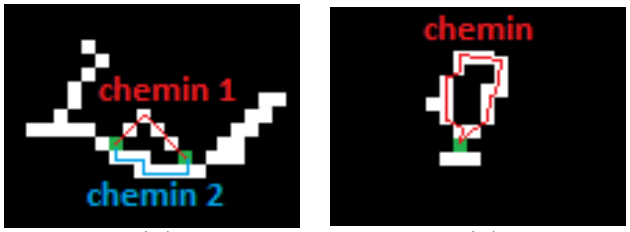
FIGURE *3.16 – (a) Ramifications sélectionnées en bleu (entre un point de jonction en vert et un point extrémité en rouge) dans le squelette de fissuration - (b) Ramifications restantes après le seuillage par $T_c$.*

#### 3.2.4.7 Sélection et filtrage des boucles superflues

Cette étape consiste à chercher les chemins qui relient les rondpoints $r_i$ entre eux. Deux types de boucles ont été sélectionnés dans le squelette de fissuration, à l'exemple de la figure 3.19 :

- Les boucles normales qui correspondent à plusieurs chemins entre deux points $r_i$ (figure 3.17.a).

- Les boucles fermées qui correspondent à un seul chemin ayant un point source $r_i$ qui correspond à son point de destination (figure 3.17.b).

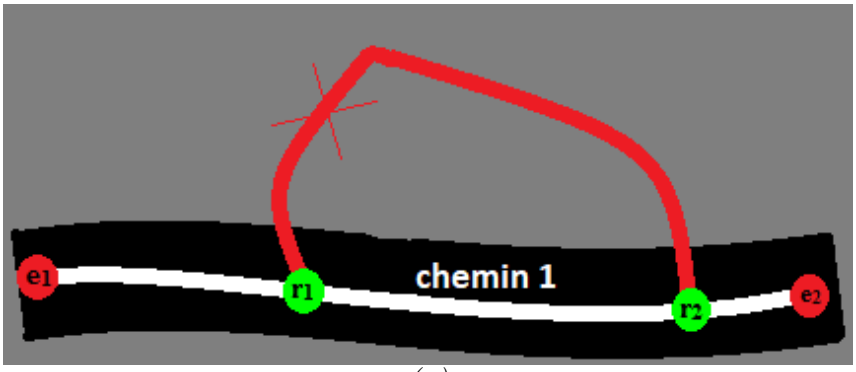



(a) (b)

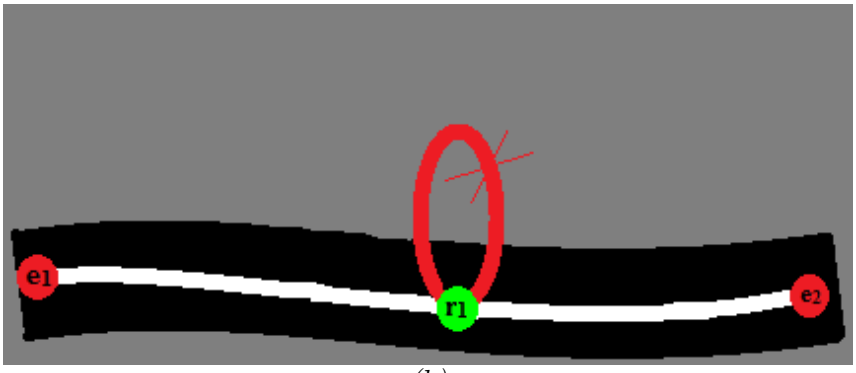
FIGURE *3.17 – (a) Boucle "normale" formée de deux chemins ayant deux $r_i$ - (b) Boucle "fermée" formée d'un seul chemin où se confondent le point source et le point destination.*

Dans le cas d'une boucle "normale", la méthode de seuillage (seuil $T_c$) est appliquée séparément à chacun des chemins constituant la boucle (figure 3.18.a). Trois cas peuvent intervenir : un des chemins est conservé, les deux chemins sont conservés, les chemins sont éliminés créant une déconnexion entre deux parties du squelette. Autrement, la méthode de seuillage s'applique directement au chemin formant une boucle "fermée" (figure 3.18.b).

Une remarque à prendre en compte est que dans le cas où une boucle contient plusieurs rondpoints, nous appliquons une étape de nettoyage des rondpoints avant le seuillage, qui consiste à ne conserver dans la structure de la boucle que le rondpoint qui a le plus de connexions (i.e., pixels voisins).




*(a)*




*(b)*

FIGURE *3.18 – (a) Boucle "normale" après le seuillage $T_c$ - (a) Boucle "fermée" après le seuillage $T_c$.*

Les figures 3.19.a-b illustrent l'identification des deux types de boucles sur le squelette de fissuration initial. Le résultat obtenu après le filtrage des deux types de boucles est illustré aux figures 3.19.c-d. Nous notons la suppression des parties hors fissures (en rouge) dans le squelette de fissuration, et globalement, une certaine simplification du squelette segmenté.

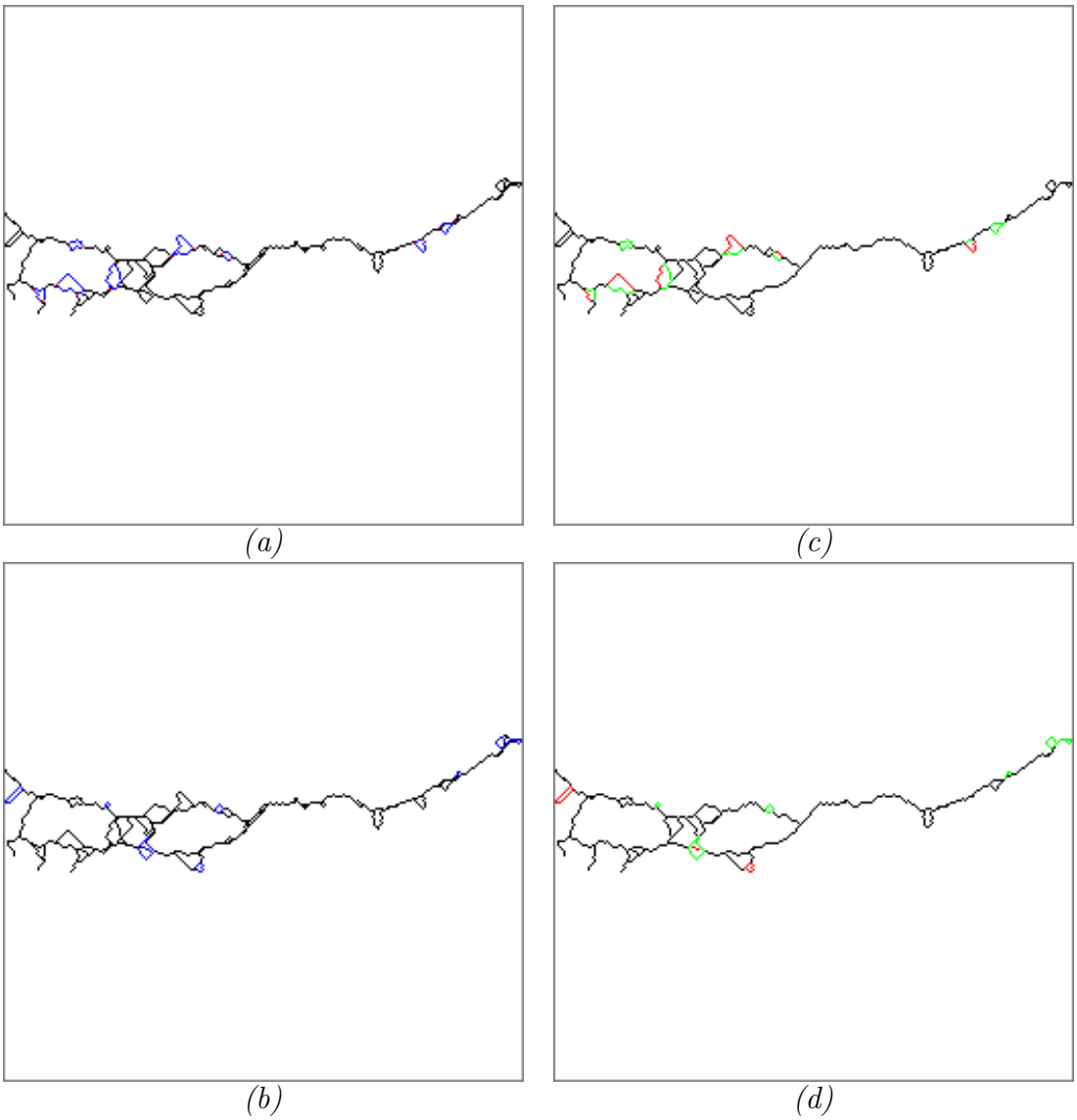


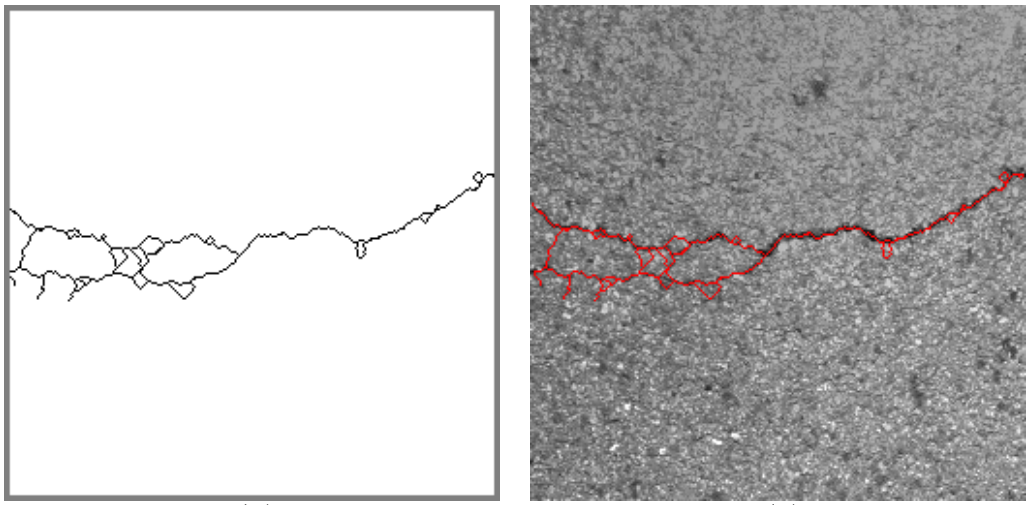

FIGURE *3.19 – Boucles "normales" (a) et "fermées" (b) sélectionnées dans le squelette de fissuration (en bleu) - Boucles "normales" (c) et "fermées" (d) restantes (en vert) et éliminées (en rouge) après le seuillage par* $T_c$.

#### 3.2.4.8 Segmentation finale du squelette de fissuration

L'application séquentielle des différentes étapes du post-traitement ne permet pas en général d'éliminer tous les artefacts le long du squelette. Dans la stratégie "R-continuité", les auteurs [123] ont proposé d'appliquer le post-traitement d'une manière itérative jusqu'à la stabilisation de la forme du squelette. Les figures 3.20 et 3.21 illustrent l'amélioration de la segmentation qui est obtenue grâce à ce processus itératif, en comparaison de la segmentation initiale représentée à la figure 3.8.b. Ce proces-

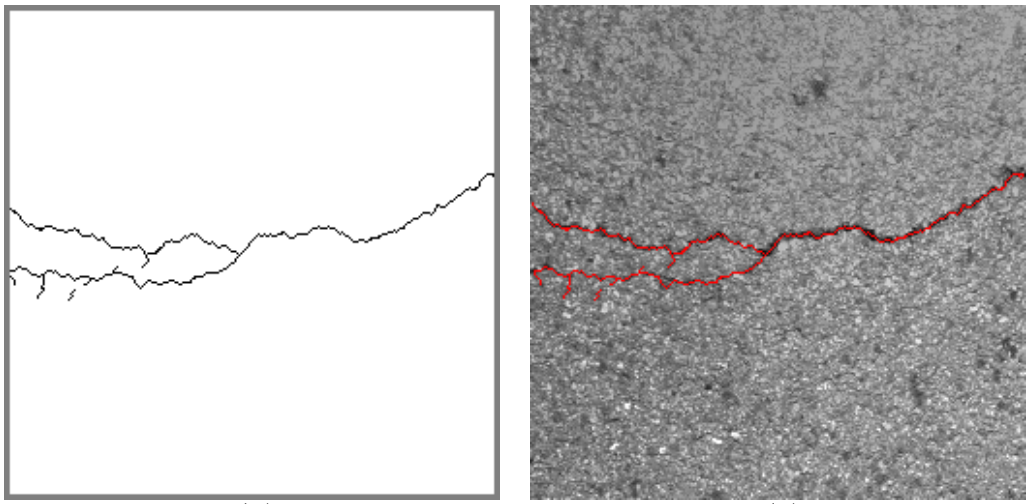
sus itératif favorise la conservation de la connexité entre les chemins calculés dans le squelette initial, tout en réduisant le taux de fausse alarme.

En pratique, la stratégie de raffinage du squelette qui a été retenue dans [123], consiste à enchaîner la stratégie "R-continuité" (qui préserve la continuité du squelette) et la stratégie "R-seuillage" (qui élimine toute portion de squelette dont le coût est supérieur au seuil $T_c$).



*(a)* *(b)*

FIGURE *3.20 – Affinage du squelette de fissuration après une première itération de l'étape 4 (le squelette initial est représenté à la figure 3.8.b).*



*(a)* *(b)*

FIGURE *3.21 – Squelette de fissuration final après quelques itérations de l'étape 4.*

### 3.2.5 Étape 5 : Estimation de l'épaisseur locale de la fissure

L'estimation de l'épaisseur de la fissure est une étape importante dans la détection des fissures. Une forte épaisseur de la fissure implique intuitivement une dégradation plus importante de la chaussée.

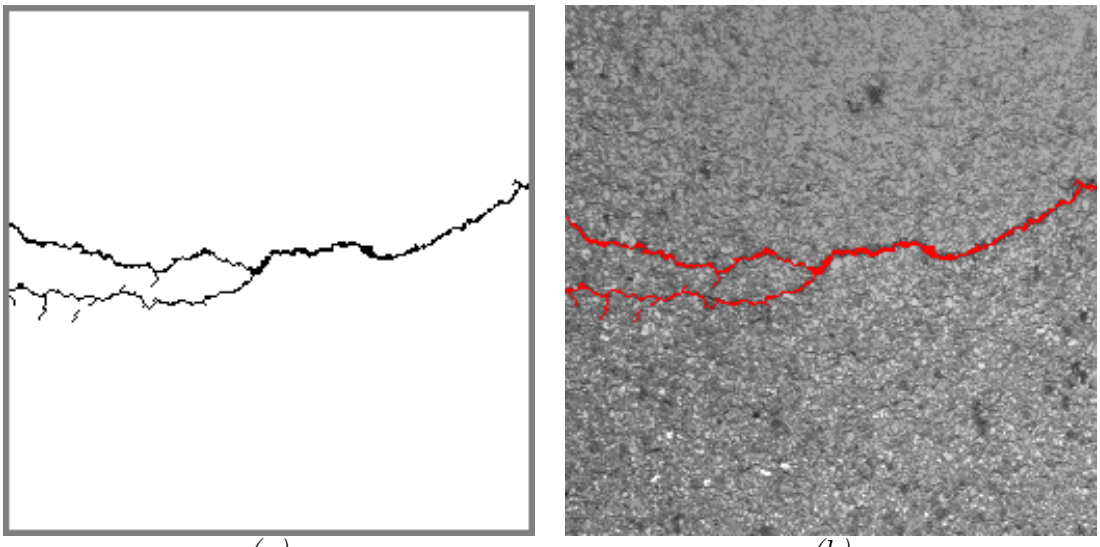
Puisque les fissures observées dans l'image sont de largeur variable, la méthode MPS estime une épaisseur locale le long du squelette de fissuration. Cela permet de réduire le nombre de pixels manquants (faux négatifs), sans augmenter le nombre de pixels faux positifs (fausse alarme). En comparaison, les algorithmes de la littérature délivrent une épaisseur constante, et produisent ainsi davantage de pixels faux positifs (fausse alarme).

Pour calculer cette épaisseur, nous nous basons sur le résultat obtenu à l'étape précédente. Dans ce cadre, nous agrégeons itérativement au squelette de fissuration les pixels voisins (dans un voisinage $3 \times 3$), dont le niveau de gris est inférieur au seuil $T_w$, défini selon l'équation suivante :

$$T_w = \mu - K_w \times \sigma \tag{3.6}$$

où $\mu$ et $\sigma$ sont les valeurs de la moyenne et de l'écart-type des pixels du squelette de fissuration et $K_w$ est une constante qui est ajustée de manière à augmenter le nombre de vrais positifs et limiter le nombre de faux positifs.

Le résultat de l'estimation de l'épaisseur est illustré à la figure 3.22.



*(a)* *(b)*

FIGURE *3.22 – Estimation de l'épaisseur ($K_w$=0.6) – (a) Résultat binaire - (b) Résultat couleur.*

## 3.3 Méthodologie d'évaluation des performances de MPS

Les critères d'évaluation du paragraphe 3.3.2 permettent de quantifier la qualité de la segmentation automatique que fournit l'algorithme MPS. Pour obtenir une méthode d'évaluation très sensible, nous avons choisi de comparer pixel à pixel les images segmentées par MPS à des images de référence, définies au paragraphe 3.3.1 comme pseudo-vérité terrain.

### 3.3.1 Pseudo-vérité terrain (PVT)

La pseudo-vérité terrain (PVT) est une image binaire qui représente la segmentation de fissure "idéale" que l'on pourrait obtenir théoriquement. La génération d'une PVT nécessite un outil de traitement spécifique mais reste une tâche laborieuse. L'étape de collecte de la PVT est néanmoins une étape essentielle pour pouvoir évaluer les performances des algorithmes de segmentation.

Dans [108], la PVT a été établie à partir d'une segmentation manuelle à l'écran, réalisée pixel à pixel, par 4 opérateurs de manière indépendante. Elle a été remplacée dans [123] par une PVT semi–automatique, qui s'est révélée plus précise que la segmentation manuelle précédente (voir figure 3.5 dans [123]).

Dans ce dernier cas, l'opérateur fixe manuellement à l'écran le point source et destination des fissures, et un algorithme de plus court chemin (de type SPSP introduit dans la section 3.2.2.2) détecte le squelette de fissuration. En cas de fissure complexe, l'opérateur réalise cette opération de segmentation pour chaque ramification. Ensuite, nous déterminons l'épaisseur locale de la fissure, en suivant le principe de l'étape 5 de l'algorithme MPS (section 3.2.5). Une comparaison de différents algorithmes pour obtenir la PVT est disponible dans [136, 137].

La base de données que nous avons utilisée dans ce document pour tester les algorithmes proposées, inclut la PVT associée aux images de différents capteurs.

En pratique, dans le processus d'évaluation, les auteurs proposent de tolérer une faible distance (en nombre de pixels) entre la segmentation automatique et la PVT pour le calcul du taux des vrais positifs. Dans [123], un voisinage de 2 pixels est considéré autour de la PVT. Ces pixels sont marqués en jaune dans la figure 3.23.c.

D'autres auteurs prennent une marge de tolérance plus importante, ce qui a pour effet de réduire la sensibilité de l'évaluation. Dans [98] par exemple, les auteurs montrent que le voisinage de 5 pixels augmente la valeur du critère d'évaluation "F1" de tous les algorithmes de segmentation de manière artificielle, tout en réduisant les écarts de performance entre les algorithmes.

### 3.3.2 Critères d'évaluation

Pour pouvoir quantifier la qualité de la segmentation de la méthode MPS, nous comparons le résultat final de la segmentation après l'estimation de l'épaisseur (figure 3.23.a) à l'image pseudo-vérité terrain (figure 3.23.b).

La comparaison entre ces deux images fait apparaître quatre catégories de pixels :

- **TP :** les vrais positifs qui correspondent aux pixels appartenant vraiment à une fissure et étant segmentés automatiquement comme fissure. Ce sont les bonnes détections marquées en vert (figure 3.23.c).

- **FP :** les faux positifs qui correspondent aux pixels détectés comme fissures alors qu'ils ne font pas partie de la pseudo-vérité terrain. Ce sont les fausses alarmes marquées en rouge (figure 3.23.c).

- **FN :** les faux négatifs qui correspondent aux pixels détectés comme non fissure alors qu'ils le sont dans la pseudo-vérité terrain. Ce sont les non détections marquées en bleu (figure 3.23.c).

- **TN :** les vrais négatifs qui correspondent aux pixels non-fissure aussi bien dans le résultat de la segmentation automatique que dans la PVT.

Les quatre catégories de pixels servent dans la littérature à quantifier la qualité de la segmentation. Parmi les critères de la littérature, nous pouvons citer la courbe ROC qui représente les variations du taux de FP en fonction du taux de TP, ainsi qu'un certain nombre d'indices, dont l'intérêt peut varier selon l'application.

En effet, si l'objet à détecter est fortement représenté dans l'image, il est nécessaire d'utiliser des critères généralistes qui prennent en compte les quatre catégories de pixels précédentes ; nous pouvons citer par exemple le critère "simple matching coefficient" [139–142]. Dans notre cas, où l'on cherche à détecter des objets de petite taille et/ou faiblement représentés dans l'image (i.e., les fissures), les critères généralistes précédents manquent de sensibilité, du fait du nombre important de TN. Il faut alors utiliser des critères basés sur les valeurs de TP, FP et FN [143–146], telles que les coefficients de précision et de sensibilité présentés ci-dessous [147] :

- **P :** la précision qui représente la proportion des faux positifs (fausse alarme) dans l'image.

$$P = \frac{TP}{TP + FP} \tag{3.7}$$

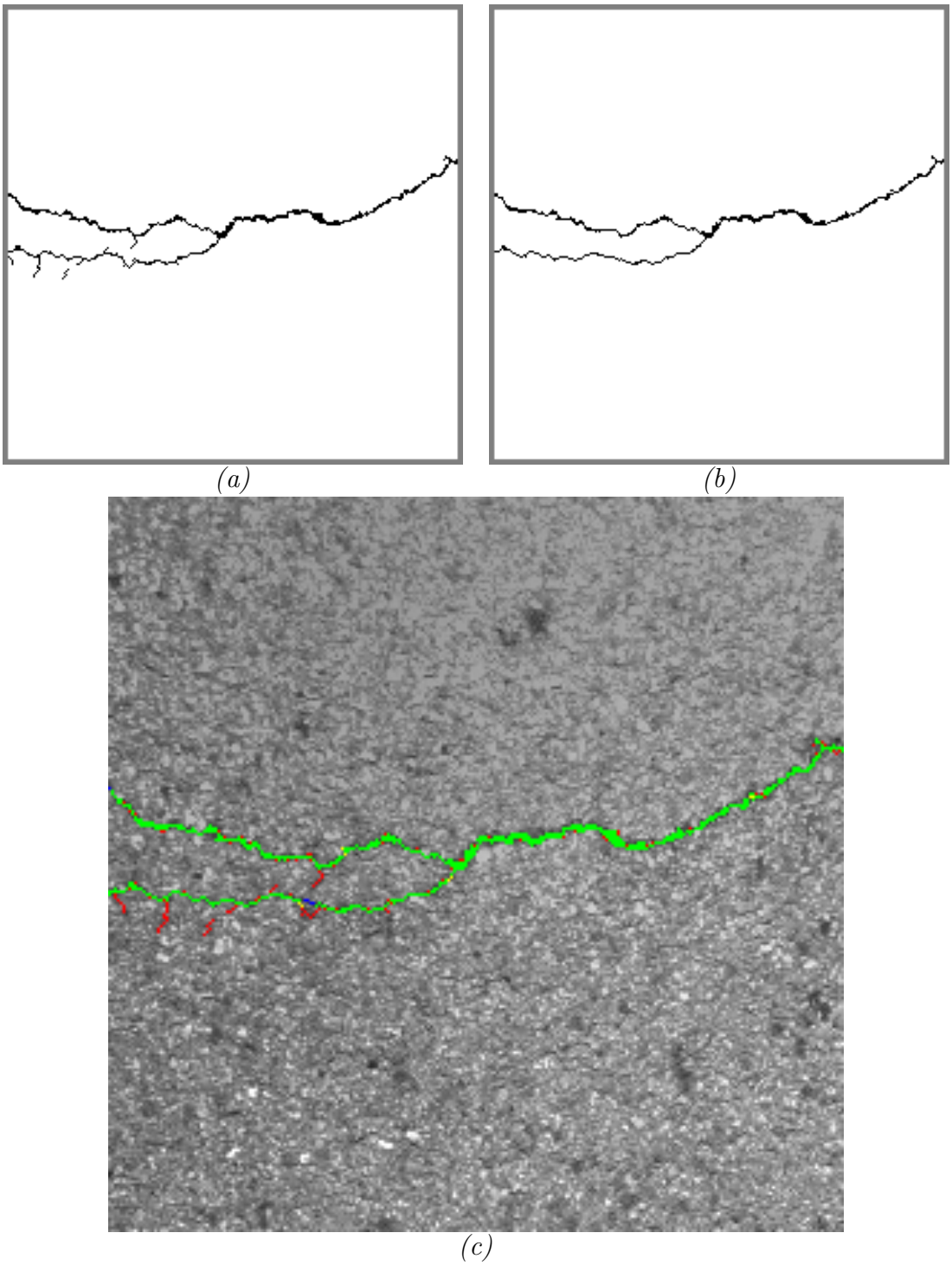
– **S :** la sensibilité qui représente la proportion de faux négatifs (pixels non détectés) dans l'image.

$$S = \frac{TP}{TP + FN} \tag{3.8}$$

Afin de synthétiser les deux critères ci-dessus en un seul, nous utilisons la moyenne harmonique de la précision et de la sensibilité, qui est plus connu sous le nom d'"indice de similarité de DICE-Sorensen (DSC)" ou "F1 score".

$$DSC = \frac{2}{\frac{1}{P} + \frac{1}{S}} = \frac{2TP}{2TP + FP + FN} \tag{3.9}$$

L'indice DSC met en évidence les bonnes détections TP en fonction à la fois des fausses alarmes FP et des non détections FN. Dans le cas d'une détection parfaite, la valeur de DSC est égale à 1 alors que cette valeur tombe à 0 dans le cas d'une décision totalement erronée. Cependant, d'après [148], une valeur supérieure à 0.7 indique une excellente ressemblance.



(a) (b)

(c)

FIGURE *3.23 – Évaluation des performances de la méthode MPS – (a) Résultat de la détection automatique en appliquant la méthode MPS sur l'image en niveaux de gris de taille* $256 \times 256$ *pixels illustrée dans la figure 3.2.a - (b) Image pseudo-vérité terrain (PVT) - (c) Comparaison : TP (vert), FP (rouge) et FN (bleu).*

La figure 3.23 illustre le résultat de la comparaison de l'image segmentée automatiquement par la méthode MPS et l'image pseudo-vérité terrain correspondante (PVT). Pour cet exemple, nous constatons la prédominance de pixels vert (TP), le faible nombre de pixels rouge (FP) qui sont répartis principalement le long du squelette, et le nombre encore plus faible de pixels bleu (FN). Les valeurs de la précision (P), de la sensibilité (S) et du coefficient de similarité (DSC) sont particulièrement élevées :

$$\text{Précision : P} = 85\%\,;$$

$$\text{Sensibilité : S} = 99\%\,;$$

$$\text{Coefficient de similarité : DSC} = 91\%\,;$$

Nous notons que le temps de traitement de cette image de taille $256 \times 256$ pixels est de 92 sec. (*traitement réalisé sous Matlab en utilisant un PC de bureau Intel CoreTM i7-4790 à 3,60 GHz avec 8 Go de RAM*).

## 3.4 Illustration des performances de la méthode MPS

Dans cette section, nous proposons d'illustrer les résultats obtenus en appliquant l'algorithme MPS sur quelques images $2D$ issues du système Aigle-RN [4,5]. Ces images sont extraites de la base de données [149], qui contient également pour comparaison, les résultats de segmentation de trois autres méthodes (à savoir la méthode de Markov, la méthode Geodesic Contour (GC) et la méthode Free-Form Anisotropy (FFA)).

En comparaison, les résultats de la méthode MPS qui figurent dans ce paragraphe ont été obtenus avec notre propre paramétrage automatisé de l'algorithme, et donc, les résultats de segmentation peuvent être légèrement différents des résultats initiaux qui figurent dans [149].

La qualité de segmentation est établie par comparaison (visuelle et à partir des critères introduits dans la section 3.3.2) avec les images de pseudo-vérité terrain, établies de manière semi-automatique et également disponible dans [149].

Une évaluation de MPS plus exhaustive a été proposée dans [117, 123, 150] à partir d'images de chaussées issues de 3 capteurs existants (Aigle-RN, LRIS, Tempest).

Les figures 3.26 à 3.29 illustrent le fonctionnement de ces quatre algorithmes sur dix images pré-traitées de taille $462 \times 991$ pixels de la base de données. D'après ces résultats, la méthode MPS offre la segmentation la plus précise et peut s'adapter aux différentes textures d'image. En effet, les pixels faux positifs sont peu nombreux et la

plupart d'entre eux sont situés le long du squelette de la fissure au lieu d'être dispersés sur toute l'image. En comparaison, le résultat obtenu par la méthode de Markov montre un squelette morcelé et beaucoup de pixels faux positifs dispersés dans toute l'image. La méthode de contour géodésique produit un résultat de segmentation incomplet, incluant une forte proportion de pixels faux négatifs, mais la plus faible proportion de pixels faux positifs en contrepartie. Selon [117], les résultats de ces deux méthodes se dégradent davantage sur des images brutes. Comme MPS, la méthode FFA présente une segmentation proche du squelette de fissuration. En contrepartie, le principe de FFA surestime l'épaisseur de fissure, et produit ainsi une forte proportion de pixels faux positifs le long du squelette.

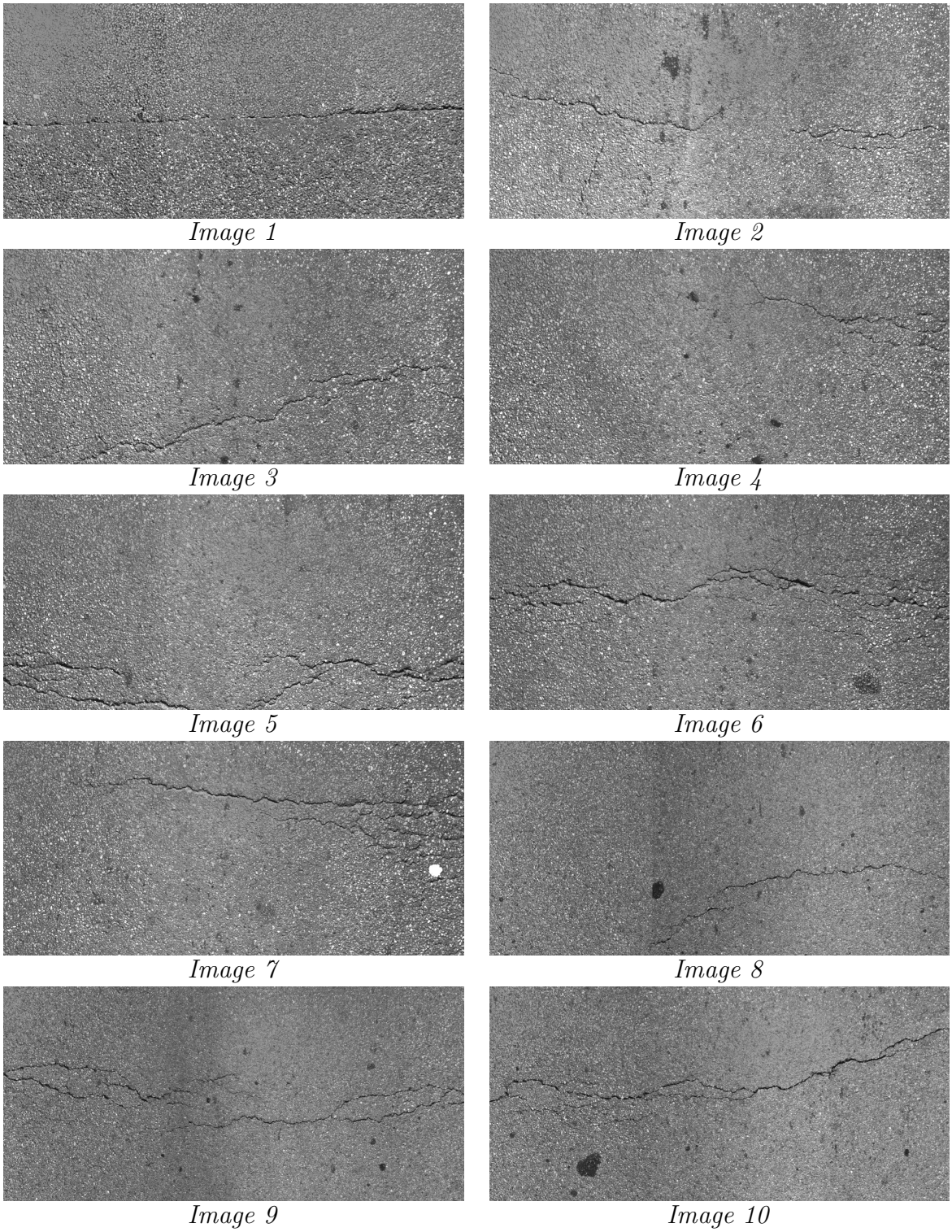



FIGURE *3.24 – Images en niveaux de gris de taille* $462 \times 991$ *pixels acquises par le système Aigle-RN [4, 5].*

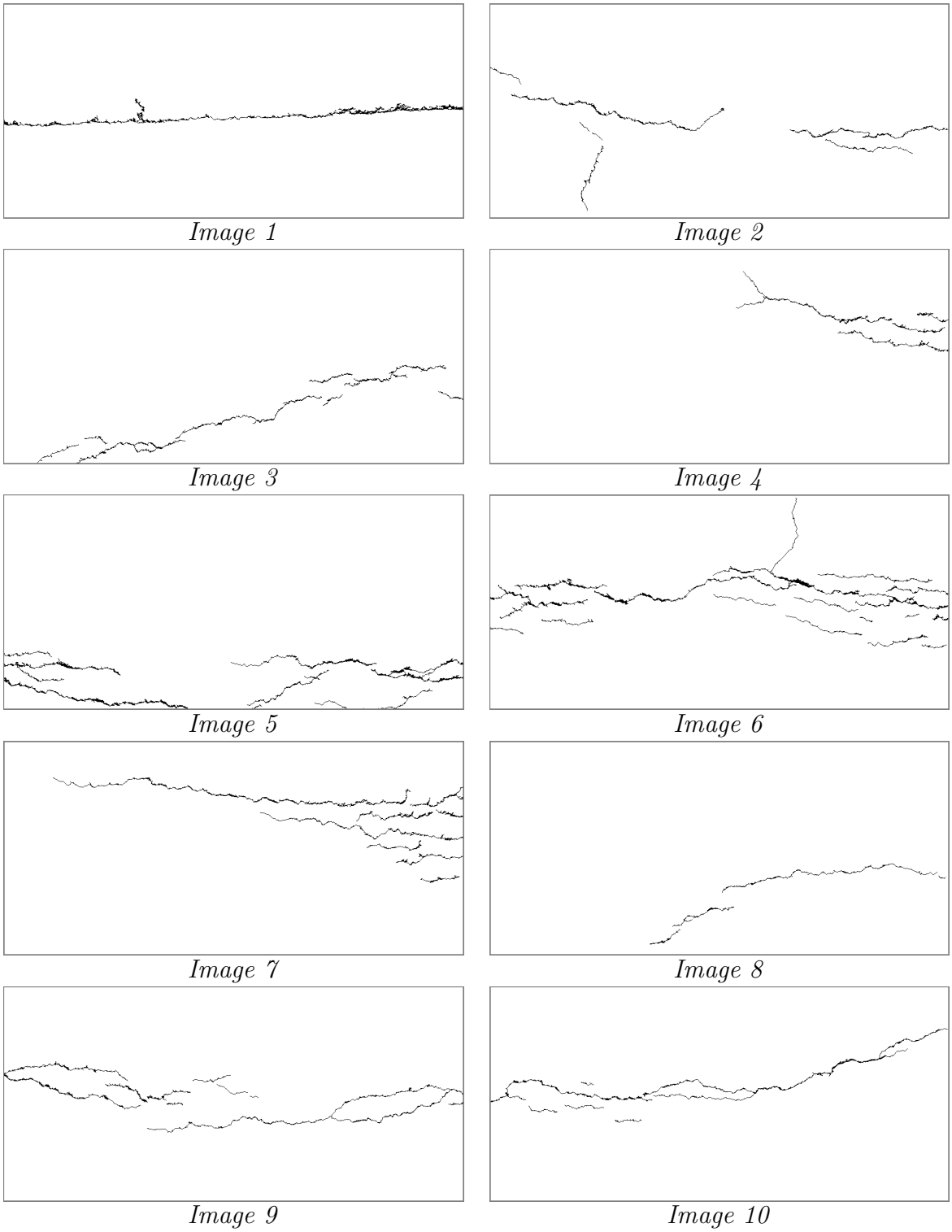



FIGURE *3.25 – Images pseudo-vérité terrain correspondantes aux images de chaussée de la figure 3.24 ; d'après [149].*

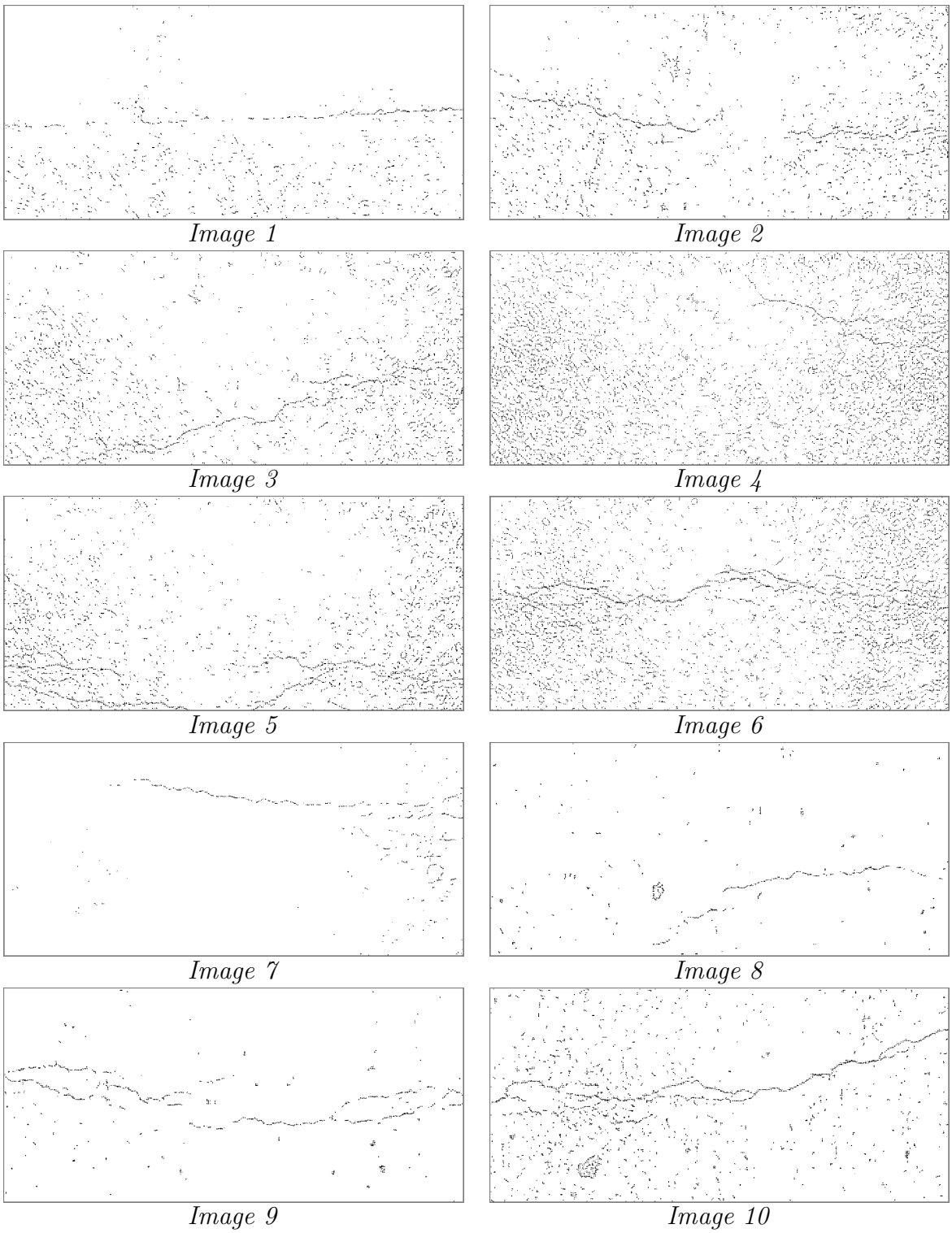



FIGURE *3.26 – Résultats de segmentation en appliquant la méthode markovienne sur les images de la figure 3.24 ; d'après [149].*

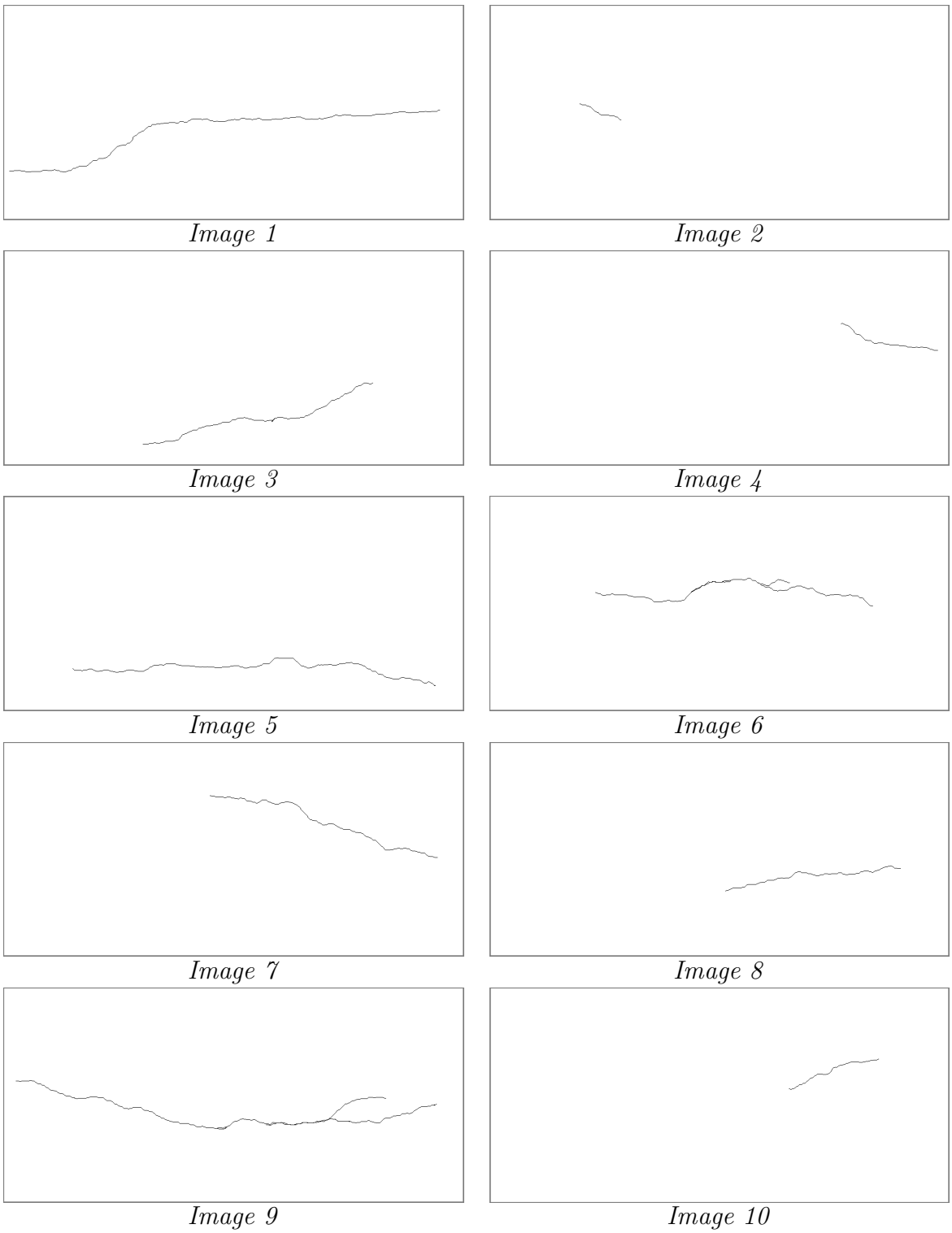



FIGURE *3.27 – Résultats de segmentation en appliquant la méthode Geodesic Contour (GC) sur les images de la figure 3.24 ; d'après [149].*

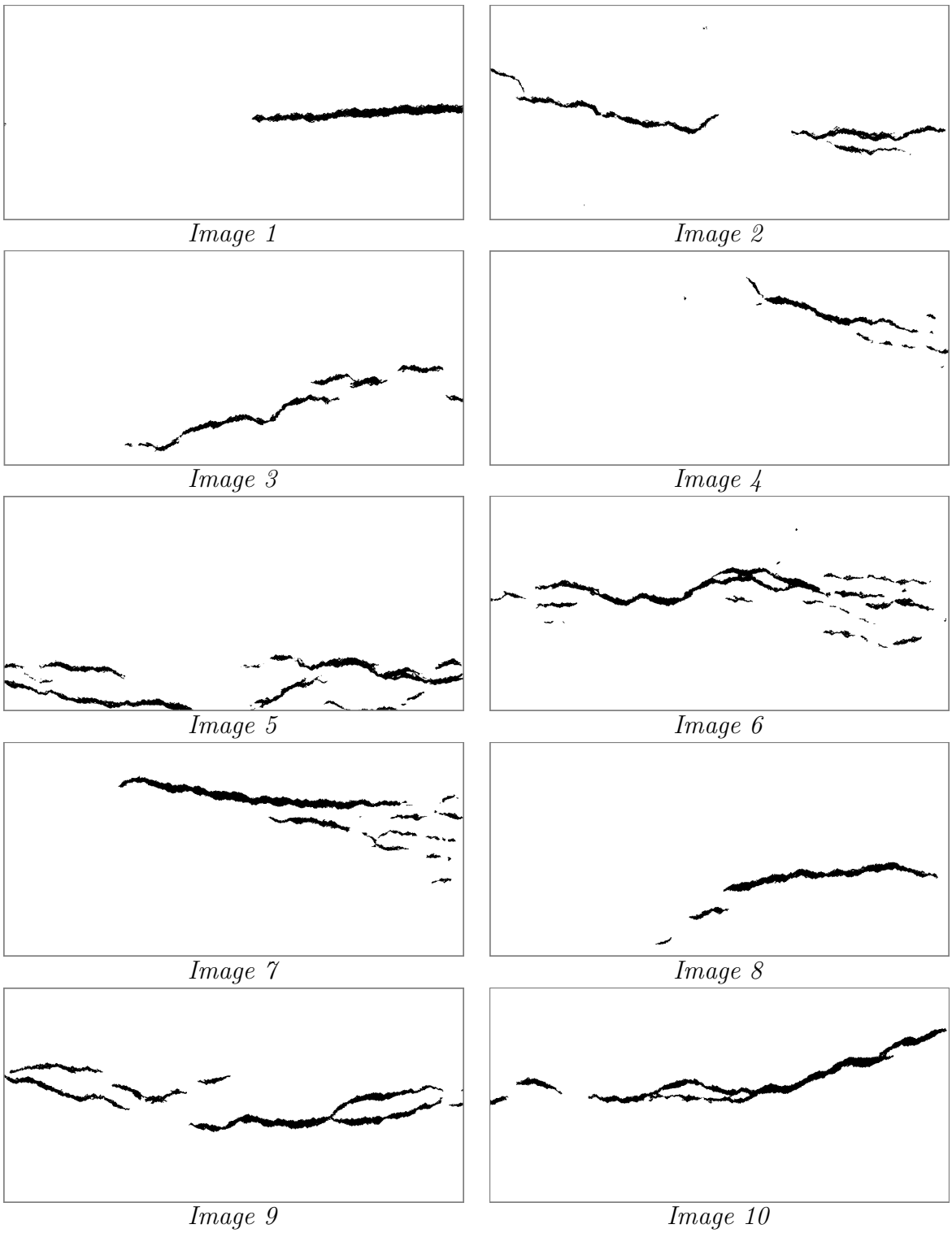



FIGURE *3.28 – Résultats de segmentation en appliquant la méthode Free-Form Anisotropy (FFA) sur les images de la figure 3.24 ; d'après [149].*

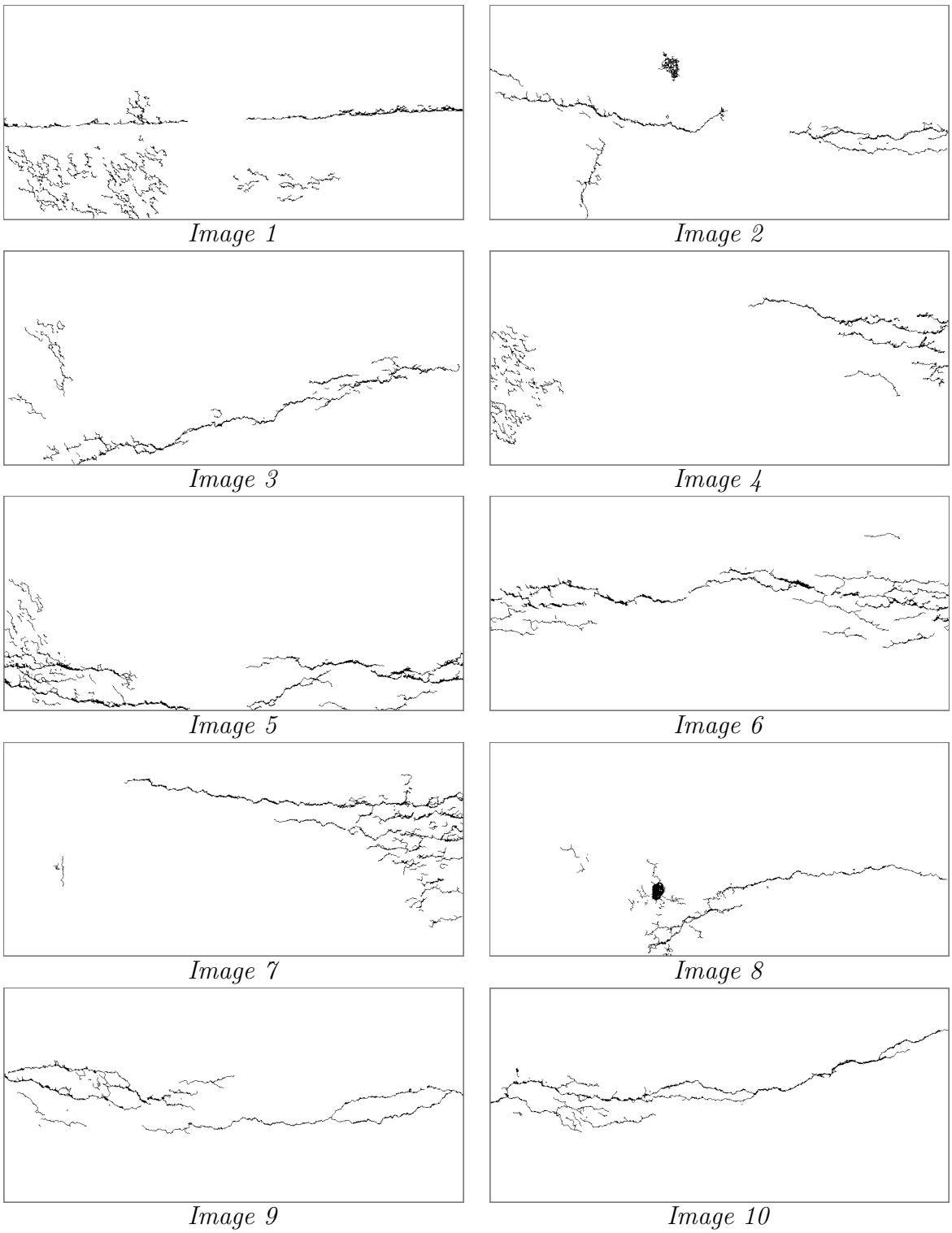



FIGURE *3.29 – Résultats de segmentation en appliquant la méthode Minimal Path Selection (MPS) sur les images de la figure 3.24.*

Les trois tableaux 3.1, 3.2 et 3.3 représentent les valeurs moyennes des critères de précision, de sensibilité et de similarité obtenues par les 4 méthodes de segmentation citées, sur chacune des images de chaussées de la figure 3.24. Les valeurs obtenues montrent la bonne performance de la méthode MPS en comparaison des trois autres méthodes. En effet, la valeur moyenne de la précision pour la méthode MPS (55%) est supérieure à celle obtenue par la méthode de Markov (33%) et la méthode FFA (24%). D'après l'équation 3.7, la méthode MPS présente beaucoup moins de fausses alarmes (pixels faux positifs) par comparaison avec la pseudo-vérité terrain. Le résultat de la méthode GC est particulier car elle produit deux valeurs extrêmes de précision et de sensibilité (le plus fort taux de précision (84%) et le plus faible taux de sensibilité (20%)), qui rend finalement cette méthode moins fiable que les trois autres avec un taux de similitude DSC le plus faible.

De la même manière, les valeurs de la sensibilité obtenues montrent que la méthode MPS reste la meilleure solution en comparaison des trois autres méthodes testées. Plus précisément, nous obtenons une valeur moyenne (93%) largement supérieure à celles obtenues par les trois autres méthodes, correspondante à un taux de non-détection très faible.

Finalement, en tenant compte des deux valeurs de précision et de sensibilité qui mettent en évidence les bonnes détections (VP) en fonction des fausses alarmes (FP) et des manques de détection (FN), la méthode MPS présente de meilleure performance que les trois autres méthodes avec une valeur moyenne de DSC est égale à 68%.

| **Images** | **Markov P (%)** | **GC P (%)** | **FFA P (%)** | **MPS P (%)** |
|---|---|---|---|---|
| Image 1 | 26 | 74 | 28 | 35 |
| Image 2 | 19 | 91 | 25 | 53 |
| Image 3 | 18 | 66 | 25 | 51 |
| Image 4 | 6 | 88 | 27 | 33 |
| Image 5 | 26 | 71 | 30 | 63 |
| Image 6 | 23 | 88 | 31 | 82 |
| Image 7 | 75 | 87 | 22 | 61 |
| Image 8 | 39 | 89 | 14 | 33 |
| Image 9 | 69 | 91 | 20 | 76 |
| Image 10 | 28 | 99 | 18 | 67 |
| **Moyenne** | 33 | **84** | 24 | 55 |

TABLE *3.1 – Valeurs de précision obtenues lors de l'application des méthodes (Markovienne, GC, FFA, MPS) sur les images Aigle-RN de la figure 3.24.*

| Images | Markov S (%) | GC S (%) | FFA S (%) | MPS S (%) |
|---|---|---|---|---|
| Image 1 | 29 | 27 | 55 | 77 |
| Image 2 | 64 | 15 | 83 | 98 |
| Image 3 | 67 | 17 | 66 | 93 |
| Image 4 | 64 | 14 | 67 | 92 |
| Image 5 | 59 | 14 | 84 | 97 |
| Image 6 | 64 | 13 | 68 | 87 |
| Image 7 | 31 | 14 | 68 | 92 |
| Image 8 | 61 | 39 | 84 | 99 |
| Image 9 | 57 | 37 | 83 | 96 |
| Image 10 | 76 | 10 | 79 | 99 |
| **Moyenne** | 57 | 20 | 74 | **93** |

TABLE *3.2 – Valeurs de sensibilité obtenues lors de l'application des méthodes (Markovienne, GC, FFA, MPS) sur les images Aigle-RN de la figure 3.24.*

| Images | Markov DSC (%) | GC DSC (%) | FFA DSC (%) | MPS DSC (%) |
|---|---|---|---|---|
| Image 1 | 27 | 39 | 37 | 48 |
| Image 2 | 29 | 9 | 38 | 69 |
| Image 3 | 28 | 27 | 36 | 66 |
| Image 4 | 11 | 24 | 39 | 49 |
| Image 5 | 36 | 24 | 45 | 77 |
| Image 6 | 33 | 23 | 42 | 84 |
| Image 7 | 44 | 25 | 33 | 73 |
| Image 8 | 47 | 54 | 24 | 49 |
| Image 9 | 63 | 52 | 32 | 85 |
| Image 10 | 40 | 19 | 29 | 80 |
| **Moyenne** | 36 | 30 | 36 | **68** |

TABLE *3.3 – Valeurs de DSC obtenues lors de l'application des méthodes (Markovienne, GC, FFA, MPS) sur les images Aigle-RN de la figure 3.24.*

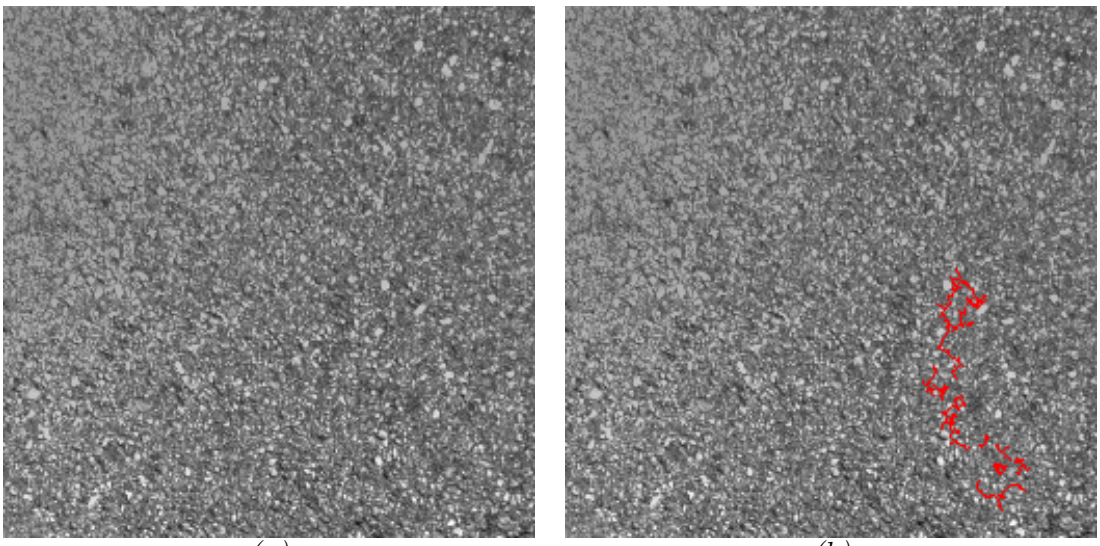
## 3.5 Conclusion

Dans ce chapitre, nous avons présenté le principe de l'algorithme MPS, que nous avons sélectionné au chapitre 1 comme étant un algorithme dédié à la détection automatique de fissures sur chaussée. Nous avons également présenté la méthodologie d'évaluation de la qualité de segmentation (à l'échelle du pixel) de l'algorithme (*nous notons que la méthode MPS a fait l'objet d'un dépôt de logiciel à l'Agence pour la Protection des Programmes* [151]).

L'analyse des résultats obtenus avec MPS (précision, sensibilité, coefficient de similarité et temps de calcul), montre que MPS permet une meilleure segmentation (avec un coefficient de similarité $\sim 70\%$ en moyenne sur la base de données [149]) que d'autres méthodes de la littérature (HA, FFA, etc.). Son principal inconvénient est son temps de calcul trop important (sous un environnement Matlab), qui limite son application au niveau opérationnel.

À l'image des autres méthodes de la littérature, MPS n'est pas capable de détecter les fissures très fines, et produit également des fausses alarmes dans des images de texture seule (figure 3.30). Un paramétrage de l'algorithme (notamment du paramètre $T_c$ à l'étape 3) pourrait permettre de réduire ce défaut, mais au risque d'une très forte augmentation du taux de fausse alarme. Ainsi, l'optimisation des seuils ne permet pas, à elle seule, d'aboutir à une solution de compromis satisfaisante dans certaines situations. Il convient alors d'améliorer et d'optimiser l'algorithme.

L'algorithme MPS a fait l'objet de quelques améliorations dans la littérature [136], dont nous nous sommes inspirés au chapitre 4 pour proposer une nouvelle version de MPS. L'objectif de ce nouvel algorithme est d'améliorer les performances d'un facteur supplémentaire (notamment le taux de similitude) et de réduire le temps d'exécution (dans un environnement Matlab).



*(a)* *(b)*

FIGURE *3.30 – Détection de faux positifs (fausses alarmes) dans une image de texture sans fissures – (a) Image originale de taille* $256 \times 256$ *pixels - (b) Résultat de la détection en appliquant la méthode MPS.*

# Chapitre 4

## Optimisations de la sélection des chemins minimaux

### Sommaire


4.1 Introduction . . . . . **124**

4.2 Améliorations proposées de la méthode MPS . . . . . **124**

4.2.1 Analyse fonctionnelle . . . . . 124

4.2.2 Amélioration de l'étape 1 : Réduction des points amorces . . 125

4.2.2.1 Modélisation de l'image et la fissure . . . . . 125

4.2.2.2 Sélection des points amorces par analyse d'anisotropie locale : MPS-V1 . . . . . 131

4.2.2.3 Sélection des points amorces par fusion des résultats : MPS-V2 . . . . . 135

4.2.3 Amélioration de l'étape 2 : Calcul optimisé des plus courts chemins entre paires de points amorces . . . . . 139

4.2.4 Amélioration de l'étape 3 : Mise à jour du paramétrage . . . 141

4.2.4.1 Répartition des valeurs de coûts des nouveaux chemins calculés . . . . . 142

4.2.4.2 Analyse de la sensibilité . . . . . 143

4.2.5 Post-traitement . . . . . 144

4.3 Étude de cas sur des images de chaussée . . . . . **145**

4.3.1 Base de données testées . . . . . 145

4.3.2 Tests et validation . . . . . 147

4.3.3 Résultats statistiques . . . . . 150
4.4 Conclusion . . . . . **151**

## 4.1 Introduction

Dans le chapitre précédent, nous avons introduit le principe de la première version de la méthode MPS (dénommée MPS-V0 dans ce chapitre) fondée sur la sélection des chemins minimaux, et qui a montré de bonnes performances comparée aux autres méthodes de la littérature. Les différents tests réalisés ont montré que MPS-V0 délivre une segmentation plus fiable et précise que les algorithmes existants dans la littérature. Par ailleurs, l'analyse des résultats obtenus a permis de déterminer les limitations de cet algorithme. Un des principaux désavantages de la méthode MPS-V0 est présenté par le temps de calcul assez élevé. Par exemple, sous une programmation Matlab, nous avons besoin d'environ 23 minutes pour traiter et analyser une image de chaussée de taille 1 MPx.

L'objectif de ce chapitre est de présenter une version optimisée de la méthode MPS, nommée "OMPS" (Optimized Minimal Path Selection), qui tente de réduire les limites de la méthode MPS-V0. Ainsi, OMPS propose d'améliorer les performances de MPS d'un facteur supplémentaire (notamment le taux de similitude) et de réduire le temps de calcul.

Les améliorations proposées concernent les trois premières étapes de MPS-V0 qui permettent la segmentation du squelette de fissuration. Nous notons que l'amélioration des trois premières étapes conduit automatiquement à la simplification du processus des deux dernières étapes.

Enfin, les performances d'OMPS sont comparées à l'algorithme initial à partir des images du même benchmarking qu'au chapitre 3, composé d'images de chaussée acquises à vitesse de trafic sous éclairage contrôlé.

## 4.2 Améliorations proposées de la méthode MPS

Dans cette section, nous présentons en détail les deux améliorations proposées de la méthode MPS d'origine (MPS-V0) dénommées MPS-V1 et MPS-V2 respectivement.

Nous notons que ces deux versions se concentrent sur la segmentation du squelette de fissuration par le biais d'améliorations apportées à la première phase de l'algorithme (voir les étapes 1 à 3 de la figure 3.1). L'évaluation de la performance des deux versions améliorées est présentée à la section 4.3.

### 4.2.1 Analyse fonctionnelle

Le chapitre 3 a souligné la nécessité de réduire le temps de calcul en optimisant la méthodologie de la version originale de MPS (MPS-V0). En effet, les premiers tests réalisés dans [136], ont permis d'identifier le processus de recherche des chemins mini-

maux aux étapes 1 et 2 de l'algorithme comme étant les étapes les plus complexes et longues en termes de temps de calcul.

En outre, nous pouvons remarquer l'apparition de nombreux chemins calculés dans des zones non dégradées (sans défaut), i.e., le fond ou la texture de l'image, comme c'est illustré dans la figure 3.5 du chapitre 3. En effet, ces chemins ralentissent l'algorithme de détection et contribuent ainsi à augmenter le taux de fausses alarmes. D'où l'idée est de trouver une méthode capable de réduire la quantité des chemins calculés à l'étape 2 de l'algorithme. Pour y arriver, nous avons proposé dans un premier temps d'améliorer l'étape 1 de l'algorithme et de la remplacer par une méthode plus fiable capable de fournir une sélection plus précise des points amorces dans l'image.

Plus précisément, afin d'améliorer le résultat de la segmentation (étapes 1, 2 et 3 de la méthode MPS) et ainsi réduire le temps d'exécution élevé par rapport à d'autres méthodes [150], un réexamen du fonctionnement de la méthode a été initié dans [136], ainsi qu'une optimisation du codage sous Matlab. Dans la section suivante, nous allons présenter nos contributions et les méthodes d'optimisation proposées afin d'obtenir un meilleur squelette de fissuration.

### 4.2.2 Amélioration de l'étape 1 : Réduction des points amorces

Les différents tests réalisés ont montré que la présence ou non des fausses alarmes ainsi que le temps de calcul important sont dus au grand nombre des points amorces sélectionnés à l'étape 1 de l'algorithme MPS. En effet, ils dépendent principalement du nombre de pixels à traiter dans l'image. Plus ce nombre est petit, plus la méthode est rapide et robuste.

En général, la méthode de sélection des points amorces (section 3.2.1 du chapitre 3) dépend principalement de la taille de l'image à traiter. Donc plus la taille de l'image est grande, plus nous avons des points amorces, ce qui conduit à un temps de calcul important et donc une méthode très lente. D'où l'idée de chercher une méthode qui consiste à réduire le nombre des points amorces dans l'image pour réduire le temps de calcul. Pour se faire, deux stratégies ont été proposées successivement dans ce chapitre pour affiner la sélection des points amorces [136, 152]. Elles ont pour objectif de retirer les pixels amorces situés à l'extérieur de la fissure et de retenir ceux qui ont une grande probabilité d'appartenir à une fissure (pixels sombres). Les paragraphes suivants présentent en détail les deux méthodes proposées à cette fin.

#### 4.2.2.1 Modélisation de l'image et la fissure

##### a. Introduction

Dans [89, 102], les auteurs considèrent qu'une fissure est supposée avoir la forme de segments suivant une des quatre orientations 0°, 45°, 90° et 135°. En effet, pour des

orientations prédéfinies, seulement les grands défauts sont correctement détectés alors que les défauts minces sont supprimés. Cela conduit à dire que cette hypothèse est insuffisante. Elle peut être satisfaisante pour détecter les fissures assez larges mais elle reste limitée dans le cas d'orientation défavorable (structures fines, forme libre) des fissures.

L'analyse d'anisotropie a été introduite dans [153,154] pour détecter les fissures dans des surfaces texturées. Les auteurs ont constaté que les fissures occupent en général un très petit pourcentage de la surface texturée et qu'elles présentent une orientation dominante dans la plupart des cas [82]. Pour eux, une fissure influence les attributs de la texture le long de son orientation dominante. Cette observation conduit à l'idée de trouver une mesure capable de fournir une petite valeur dans l'orientation de la fissure et des grandes valeurs dans les autres orientations. Pour ce faire, la méthode FFA (Free-Form Anisotropy) peut être utilisée [114,152].

Dans ce cadre, nous supposons dans ce chapitre qu'une fissure est une structure très fine ayant une longueur plus grande que sa largeur et pouvant avoir une orientation libre et une forme arbitraire sur la chaussée. Afin de pouvoir traduire cette définition et ainsi permettre de les coder sous forme algorithmique, nous avons sélectionné la méthode proposée dans [82] qui consiste à modéliser une image de chaussée comme étant un graphe dans le but de trouver des configurations s'approchant de l'hypothèse idéale de fissures.

### b. Généralités sur les graphes

Selon [82], un graphe $G = (N, A)$ est défini comme suit :

- $N$ **:** Ensemble constitué des éléments appelés des nœuds ou des sommets.
- $A \subset (N \times N)$ **:** Ensemble constitué des éléments appelés des arêtes ou des arcs (arête est la notion pour le graphe non-orienté alors que arc est celle pour le graphe orienté).

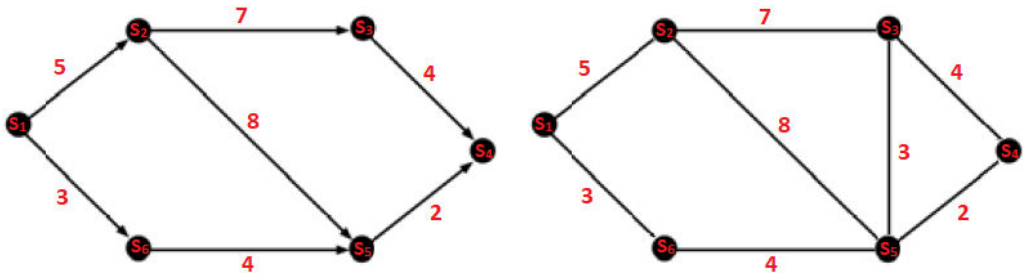
#### Graphe orienté et non-orienté

Dans la théorie des graphes, un graphe $G = (N, A)$ orienté est un couple formé d'un ensemble $N$ des nœuds et d'un ensemble $A$ des arcs dont chacun est considéré comme étant un couple de nœuds. Alors qu'un graphe non-orienté est un couple formé d'un ensemble $N$ des nœuds et d'un ensemble $A$ des arêtes dont chacune est considérée comme étant une paire de nœuds [82].

#### Graphe valué

Un graphe orienté ou non-orienté est dit valué s'il est muni d'une application. Autrement dit, un graphe valué est un couple $G = (N, A)$ sur lequel une fonction de

valuation est définie, qui associe une valeur à chaque élément de $A$. En l'absence d'une telle fonction de valuation, nous parlons d'un graphe non-valué. Un exemple de graphe valué orienté et non-orienté est présenté par la figure 4.1.




*(a)* *(b)*

FIGURE *4.1 – Exemples des graphes valués – (a) Graphe orienté - (b) Graphe non-orienté.*

**Calcul de la valuation**

D'après la figure 4.1, dans un graphe orienté, un chemin d'origine $S_i$ et d'extrémité $S_j$ est défini par une suite finie d'arcs consécutifs permettant de relier $S_i$ à $S_j$. La notion correspondante dans les graphes non-orientés est celle d'une chaine [82]. En effet, pour un graphe valué, la valuation d'un chemin (ou d'une chaine) est la somme des valuations de chacun des arcs (arêtes) qui le composent. Prenons par exemple, le graphe de la figure 4.1.a, la valuation du chemin ($S_1$, $S_2$, $S_3$ et $S_4$) $= 5 + 7 + 4 = 16$.

**Longueur d'un chemin**

En général, la longueur $L$ du chemin calculé est définie par le nombre des arcs utilisés. Autrement dit, elle correspond au nombre de sommets moins un (voir équation 4.1). Par exemple, la longueur du chemin ($S_1$, $S_2$, $S_3$ et $S_4$) calculé est égale à 3.

$$L = Nb_{sommets} - 1 \tag{4.1}$$

Nous notons qu'en se basant sur toutes les notions qui viennent d'être définies, un chemin (ou une chaine) est considéré(e) le plus court (la plus courte) si sa valuation est égale au minimum des valuations des chemins (ou des chaines) reliant les deux extrémités $S_i$ et $S_j$.

### c. Modélisation de l'image

Comme nous avons mentionné au début de cette section que l'approche adoptée consiste à modéliser une image comme étant un graphe. Selon cette approche, l'image à traiter n'est plus définie par une matrice $m \times n$ pixels mais elle correspond à un

graphe valué $G = (N, A)$. Plus précisément, une image $I$ est considérée comme étant un ensemble fini de nœuds noté $S = \{S_1, S_2, ..., S_N\}$. En effet, selon cette modélisation, un sommet (nœud) $N$ correspond à un pixel de l'image $I$, alors que $A$ représente l'ensemble des arêtes (arcs) entre 2 sommets. Ci-après quelques notions de base afin de bien comprendre le principe de cette modélisation.

**Voisinages directionnels**

En générale, la détection des fissures dans une image est basée sur une analyse et une recherche de voisinage tout autour de chaque pixel de l'image. Dans le cas des graphes valués, le voisinage permet de définir l'ensemble $A$ des arêtes (arcs). Dans un graphe non-orienté par exemple, l'ensemble $A$ pourrait être traditionnellement déterminé par un voisinage 4-connexe ou 8-connexe. Du fait que nous prenons en compte la forme libre ainsi que l'orientation de la fissure, donc afin de modéliser et trouver la vraie fissure dans l'image, nous avons utilisé des voisinages que nous qualifions de *voisinages traditionnels* introduits dans [82]. Une illustration de ce type de voisinage est présentée par la figure 4.2.

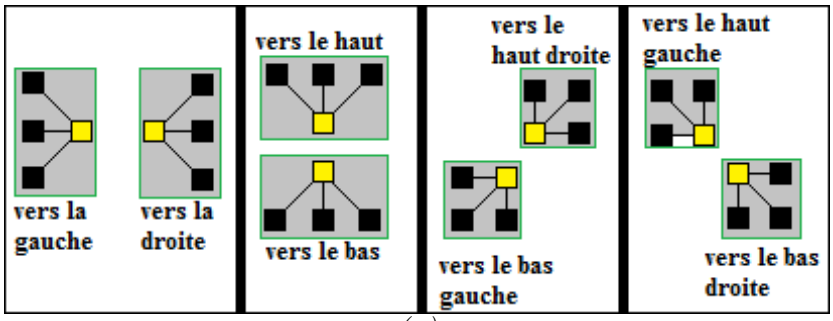



FIGURE *4.2 – Voisinages directionnels – (a) Voisinages pour un graphe valué orienté -(b) Voisinages pour un graphe valué non-orienté.*

Nous notons que dans ce travail, les graphes orientés sont utilisés pour le calcul et la recherche des chemins alors que les graphes non-orientés sont utilisés pour représenter les chemins calculés. Pour la clarté, nous présentons dans le paragraphe suivant, quelques exemples des chemins calculés dans des graphes orientés et non-orientés déterminés par des voisinages directionnels.

**Exemples des chemins calculés dans des graphes orientés et non-orientés**

La figure 4.3 présente deux types de chemins déterminés par des voisinages directionnels :

- Deux chemins ayant pour origine un nœud $l$ présentés sur deux graphes orientés vers la gauche (figure 4.3.a) et vers la droite (figure 4.3.b).

- Un chemin ayant pour origine un nœud $l$ présenté sur un seul graphe transversal non-orienté (voir figure 4.3.c).

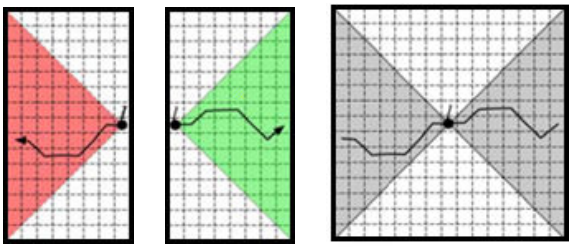



FIGURE *4.3 – Exemples des chemins dans des graphes déterminés par des voisinages directionnels – (a) Chemin de longueur* 7 *pixels, d'origine le pixel l dans un graphe du voisinage orienté vers la gauche - (b) Chemin de longueur* 7 *pixels, d'origine le pixel l dans un graphe du voisinage orienté vers la droite - (c) Chemin de longueur* 14 *pixels, d'origine le pixel l dans un graphe du voisinage transversal non-orienté.*

Nous notons que la recherche d'un chemin passant par un sommet (nœud) $l$ dans un graphe défini par un voisinage directionnel est limitée par la zone correspondante au type de voisinage.

**c. Calcul des chemins minimaux**

Dans ce paragraphe, nous présentons le concept des chemins minimaux qui est très souvent utilisé pour la modélisation des structures très fines, en particulier les fissures sur la chaussée. Pour mieux comprendre le principe, un exemple de modélisation est présenté ci-après.

Prenons le cas de la France, nous modélisons par exemple chaque ville comme étant un nœud et chaque autoroute reliant deux villes différentes comme étant un arc (ou une arête). La valuation d'une arête est en général le coût nécessaire pour dépasser l'autoroute. En effet, le plus court chemin entre la ville de Brest et Paris est ainsi celui qui minimise ce coût sans prendre en compte le nombre des villes parcourues. A ce stade, l'objectif du concept est ainsi de sélectionner le meilleur chemin qui permet de relier Brest et Paris en minimisant la somme ou le coût à payer. Pour plus de détails, les lecteurs sont référés à la section 1.5.3.8 du chapitre 1.

Revenons sur la modélisation par graphe, nous notons que sur un graphe valué non-orienté, un chemin minimal d'origine $l$ et de longueur $2d$ est tout chemin ayant un point central $l$ et une valuation minimale. Alors que sur un graphe valué orienté, un chemin minimal d'origine $l$ et de longueur $d$ correspond à tout chemin d'origine $l$ dont la longueur est égale à $d$ et dont la valuation est minimale. Le choix du paramètre $d$ est présenté dans la section 4.2.2.2.b.

**Illustration d'un chemin minimal**

Il est nécessaire de savoir qu'en utilisant les voisinages directionnels illustrés dans la figure 4.2 ainsi les graphes valués orientés, chaque pixels de l'image possède uniquement 3 pixels voisins au lieu 4 ou 8. En général, la recherche des chemins minimaux sur un graphe non-orienté 4-connexes ou 8-connexes risque de déboucher sur des chemins bouclés (voir figure 3-5 dans [82]). En effet, les voisinages directionnels permettent d'éviter les chemins inappropriés.

Comme le montre la figure 4.3.c, un chemin minimal passant par un point $l$, sur un graphe non-orienté selon une orientation données (transversale, longitudinale, etc.), est obtenu en calculant le chemin minimal d'origine $l$ sur les deux graphes orientés corresponds figure 4.3.a-b. A ce stade, pour chaque pixel $l$, quatre chemins minimaux passant le point $l$ peuvent être calculés dans quatre orientations différentes (0°, 45°, 90° et 135°). Une illustration de ces quatre chemins est présentée dans la figure 4.4.

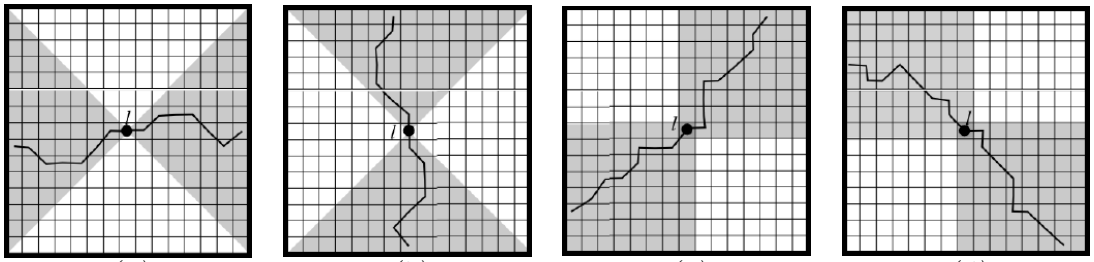



FIGURE *4.4 – Illustration des quatre chemins minimaux dans quatre orientations différentes passant par un point l – (a) Chemin transversal - (b) Chemin longitudinal - (c) Chemin oblique* 45° *- (d) Chemin oblique* 135°.

Nous notons que deux parcours, par exemple transversal et longitudinal, peuvent couvrir l'ensemble des nœuds. Cependant, cela présente des limitations pour les chemins à la frontière des parcours (oblique 45° et 135°). C'est la raison pour laquelle, l'utilisation des 4 parcours permet d'éviter des limitations.

**d. Modélisation de la fissure**

Une modélisation de la fissure que nous cherchons à détecter est donnée comme suit : "*Une fissure sur la chaussée est un chemin minimal de longueur et d'orienta-*

*tion variables avec une largeur de* 1 *pixel*". Nous notons que cette modélisation vient du fait que dans ce chapitre nous introduisons les améliorations proposées pour l'algorithme MPS afin d'obtenir un meilleur squelette de fissuration (épaisseur ou largeur qui ne dépasse pas 1 pixel). En effet, elle correspond à la représentation de la fissure des gestionnaires routiers. La section suivante présente la méthode FFA ainsi que la méthodologie appliquée pour réduire le nombre de points amorces dans l'image.

#### 4.2.2.2 Sélection des points amorces par analyse d'anisotropie locale : MPS-V1

La sélection des points amorces dans [123] s'appuie sur les caractéristiques photométriques. Dans [136,152], nous avons proposé de prendre en compte les caractéristiques géométriques d'une manière à garder la bonne performance de la méthode MPS. Pour ce faire, plusieurs stratégies [136] ont été testées pour éliminer les pixels hors fissures et conserver ceux qui ont une grande probabilité d'appartenir à une fissure. Parmi toutes ces stratégies, nous avons choisi la méthode FFA (Free Form Anisotropy) qui a montré une bonne performance dans la sélection des pixels amorces. Cette méthode d'anisotropie, citée dans [77,82,114], calcule les caractéristiques anisotropes pour chaque pixel dans toutes les directions en se basant qu'une fissure est supposée avoir la forme chaotique suivant une des quatre orientations 0°, 45°, 90° et 135° sur la chaussée.

Nous notons que la méthode FFA a été développée à l'époque pour détecter les dégradations de types fissures sur la chaussée [77,82,114]. Cependant, l'analyse des tests obtenus lors de l'application de cette méthode sur plusieurs types d'images a montré que cette méthode est non suffisante pour avoir une meilleure détection de fissures sur la chaussée. Les avantages et les inconvénients de cette méthodes sont introduits dans la section 1.5.3.8.c.ii du chapitre 1. C'est la raison pour laquelle, nous proposons de l'utiliser dans notre cas comme étant une étape de post-traitement de la sélection de points amorces introduite dans la section 3.2.1 du chapitre 3 (voir figure 4.5).

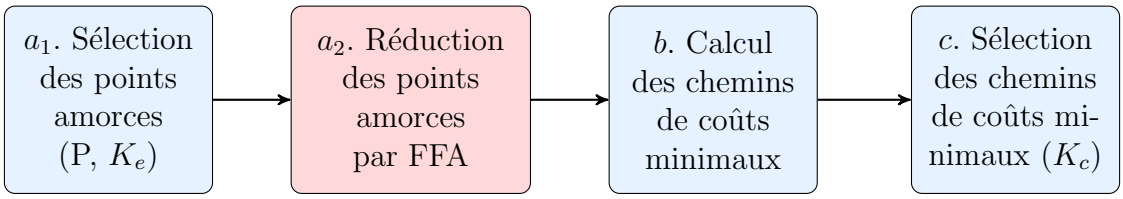



FIGURE *4.5 – Synoptique de la méthode MPS-V1 – Dans cette illustration, nous ne présentons que les étapes de la segmentation du squelette de fissuration.*

**a. Définition**

Cette section présente la première stratégie FFA (Free-Form Anisotropy) proposée pour optimiser l'algorithme MPS [136, 152]. Son objectif principal est ainsi de réduire le taux de fausse alarme de manière à avoir une meilleure détection de fissure avec un

minimum de faux positifs.

En se basant sur l'hypothèse idéale de la fissure ainsi que sur les modélisations définies précédemment, nous pouvons constater que :

- Une fissure présentée suivant une orientation définie ayant un pixel central $l$, correspond au chemin minimal d'origine $l$ trouvé dans ce parcours (orientation).

- Le chemin minimal obtenu correspond au chemin ayant la plus faible valuation parmi les chemins minimaux passant par $l$ trouvés dans les parcours qui ne contiennent pas de fissure.

En considérant les quatre chemins minimaux passant par un point $l$ selon les quatre parcours ($0^{\circ}$, $45^{\circ}$, $90^{\circ}$ et $135^{\circ}$) :

- Nous appelons *parcours min*, le parcours selon lequel le chemin minimum obtenu possède la valuation minimale. Il ainsi dénommé comme étant un *chemin min*.

- Nous appelons *parcours max*, le parcours selon lequel le chemin minimum obtenu possède la valuation maximale. Il ainsi dénommé comme étant un *chemin max*.

Une nouvelle mesure capable de détecter les pixels de fissuration est définie par le calcul d'un indice FFA selon l'équation 4.2 :

$$FFA(l) = 1 - h(\pi_{min}, \pi_{max}) \tag{4.2}$$

avec $h(\pi_{min}, \pi_{max})$ est une fonction qui traduit le degré de cohérence des deux distributions suivantes : $\pi_{min}$ et $\pi_{max}$. $\pi_{min}$ représente la distribution des niveaux de gris des pixels le long du chemin minimal *chemin min* et $\pi_{max}$ représente la distribution des niveaux de gris des pixels le long du chemin minimal *chemin max*.

Nous notons que la fonction de cohérence est définie par l'équation 4.3 suivante :

$$FC = h(\pi_{min}, \pi_{max}) = 1 - \left( \frac{mean_{max} - mean_{min}}{std_{max} - std_{min}} \right) \tag{4.3}$$

avec $mean_{max}$ et $std_{max}$ (respectivement $mean_{min}$ et $std_{min}$) correspondent à la valeur de la moyenne et de l'écart-type des niveaux de gris des pixels le long du chemin le plus clair, i.e., *chemin max*, (respectivement le plus sombre, i.e., *chemin min*).

Enfin, l'indice FFA peut être défini par l'équation 4.4 suivante :

$$FFA(l) = 1 - FC = \frac{mean_{max} - mean_{min}}{std_{max} - std_{min}} \tag{4.4}$$

### b. Réglage des paramètres $d$ et $T_{FFA}$

Il est nécessaire de noter que la distance $d$ qui définie la longueur du chemin minimal à calculer dans l'image influence l'indice FFA ainsi que le temps de calcul. Ainsi, sa valeur est choisie d'une manière à sélectionner le maximum des points amorces à l'intérieur de la fissure.

Ensuite, afin de réaliser notre objectif principal défini par la réduction du nombre des points amorces hors fissures pour réduire le temps de calcul, nous proposons de ne pas retenir que les pixels dont l'anisotropie locale est suffisamment élevée. A ce stade, parmi les points amorces sélectionnés à l'étape 1, nous ne gardons que ceux qui possèdent une valeur FFA supérieure à un seuil $T_{FFA}$ défini par l'équation 4.5.

$$T_{FFA} = T_{ostu} + K_{FFA} \times \sigma \tag{4.5}$$

avec $T_{ostu}$ correspond à la valeur d'Ostu [155] calculée automatiquement à partir de toutes les valeurs FFA obtenues, $\sigma$ est la valeur de l'écart-type de toutes les valeurs FFA calculées et $K_{FFA}$ est une constante ajustée d'une manière à garder un maximum de vrais positifs dans l'image. La figure 4.6 montre l'influence des paramètres $d$ et $T_{FFA}$ sur le taux des vrais positifs.

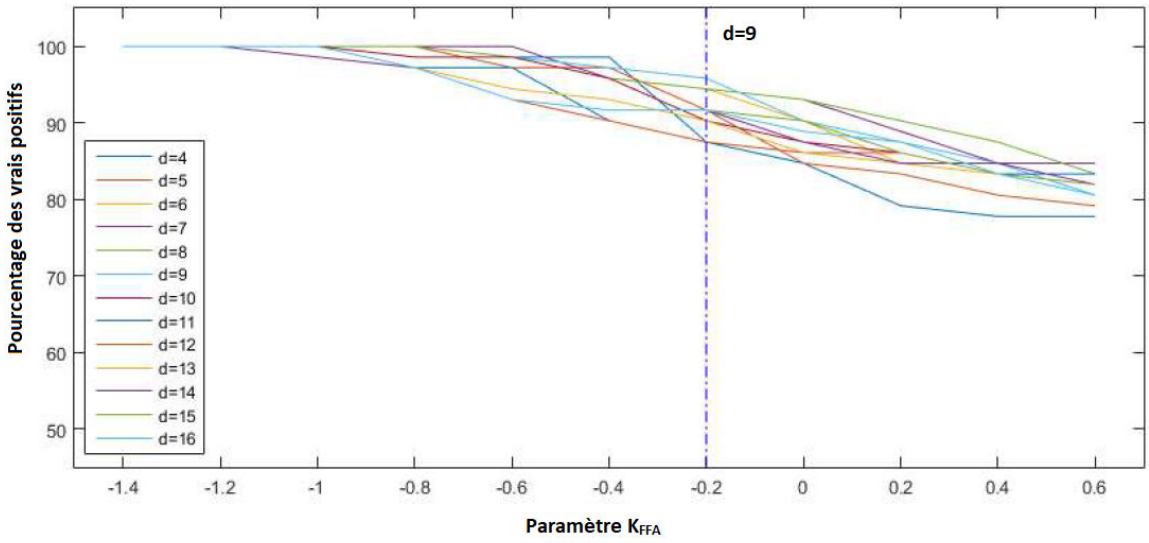
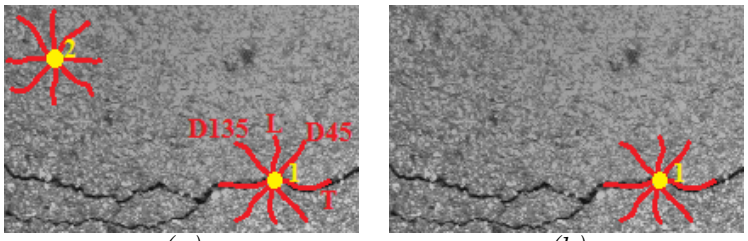



FIGURE *4.6 – Pourcentage des VP des points amorces calculés en variant les deux paramètres $d$ et $K_{FFA}$ [136] – Nous remarquons que parmi les pourcentages au-delà de 97% et pour $d = 9$, il existe une valeur maximale de $K_{FFA} = -0.2$ permettant de réduire au maximum le taux des FP des points amorces.*

### c. Exemples de calcul de la FFA

Un exemple d'application est présenté à la figure 4.7. En effet, nous cherchons à

analyser l'image contenant 2 pixels amorces de manière à supprimer le pixel situé à l'extérieur de la fissure et conserver celui qui appartient à la fissure.

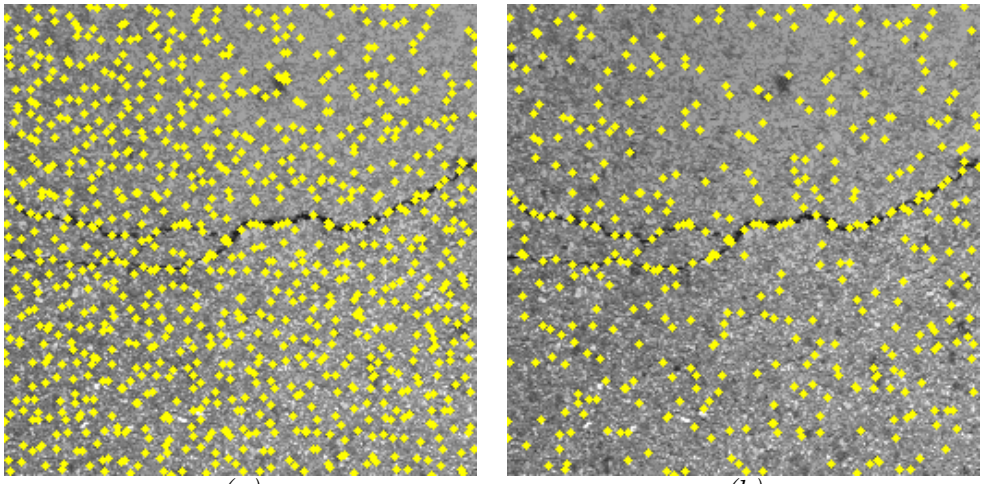



*(a)* *(b)*

FIGURE *4.7 – Exemple de la sélection des points amorces par la méthode FFA –(a) Les quatre chemins les plus courts calculés pour chacun des deux points amorces, étiquetés "1" et "2" dans l'image, selon les quatre orientations suivante :* 0°, 45°, 90° *et* 135° *-(b) Conservation du pixel "1" ayant l'indice FFA le plus élevé.*

Cet exemple montre que l'application de la méthode FFA a permis avec succès d'éliminer le pixel inutile de l'image à traiter. Dans le cas présent, nous n'avons préservé que le pixel 1 qui appartient à une fissure transversale. Le pixel 2 a été supprimé du fait qu'il est considéré comme étant un pixel appartenant à la texture de l'image.

La figure 4.8 illustre le résultat obtenu de la sélection des points amorces en appliquant la méthode FFA sur l'image en niveaux de gris de taille 256 × 256 pixels de la figure 3.2 du chapitre 3.



*(a)* *(b)*

FIGURE *4.8 – Réduction du nombre des points amorces à l'étape 1 de l'algorithme en utilisant la méthode FFA – (a) Résultat obtenu par la version MPS-V0 en calculant les minima-locaux -(b) Résultat obtenu par la version MPS-V1 en appliquant l'analyse d'anisotropie locale.*

Les résultats obtenus montrent que la méthode FFA a éliminé un grand nombre de pixels amorces ayant un indice FFA très faible. D'après [136,152], l'analyse d'anisotropie locale permet une réduction d'environ 60% des points amorces dans l'image.

#### 4.2.2.3 Sélection des points amorces par fusion des résultats : MPS-V2

Afin d'évaluer la performance de la première optimisation (MPS-V1), plusieurs tests sur différentes images ont été réalisés. L'analyse des résultats obtenus (précision, sensibilité, DSC et temps de calcul), nous a permis de remarquer que cette méthode permet une amélioration par rapport à la méthode MPS standard (MPS-V0). Cependant, elle nécessite parfois un temps de calcul élevé surtout en traitant une image de grande taille. En effet, cette dernière influe fortement sur le nombre des points amorces ainsi que sur le nombre de chemins minimaux à calculer par Dijsktra. Plus cette taille est grande, plus le nombre des amorces et des chemins les plus courts est grand. Ce qui provoque à la fin un certain nombre des fausses détections et alors un temps de calcul assez important.

Dans ce cadre, nous proposons dans cette section de réduire davantage le temps de calcul grâce à une sélection supplémentaire de points des amorces dans l'image en niveaux de gris. En effet, cette stratégie donne lieu à la troisième version de MPS, à savoir MPS-V2.

La méthode proposée comporte deux phases principales. Une première phase de pré-traitement de l'image originale et une seconde phase de sélection des points amorces pertinents. La phase du pré-traitement consiste à éliminer un maximum de bruit dans l'image pour en faciliter l'étape de traitement. Dans ce travail, cette phase repose sur une méthode de seuillage adaptatif global permettant de conserver dans l'image les pixels les plus sombres tout en éliminant les pixels clairs présentant une faible probabilité d'appartenir à une fissure sur la chaussée. Ensuite, pour que la sélection des points amorces finaux soit plus robuste et discriminante, la deuxième phase est appliquée. A ce stade, la méthode de seuillage adaptatif global est combinée avec la stratégie présentée dans la section précédente (méthode FFA). Cette combinaison définie comme étant une méthode de fusion de données (ou fusion des résultats des deux méthodes : FFA et NSCOT), est ainsi illustrée dans la figure 4.9. En général, la fusion de données est un lien important de la chaine du traitement de l'information numérique, qui consiste à passer d'un ensemble d'observations à un processus de prise de décision [156].

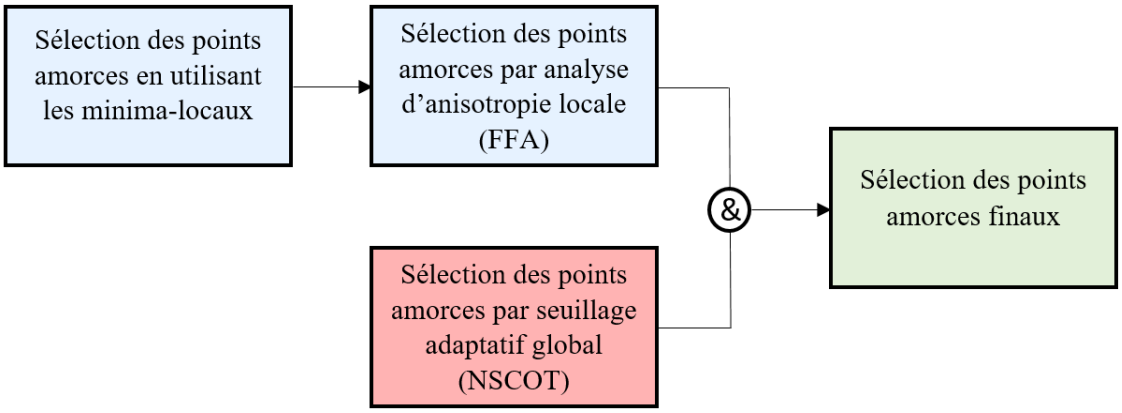



FIGURE *4.9 – Synoptique de l'étape 1 améliorée de l'algorithme MPS.*

Pour ce faire, le seuil adaptatif calculé à partir les minima-locaux et introduit dans la section 3.2.1 (équation 3.2) est ensuite remplacé par un seuil global et plus net nommé par la suite NSCOT (Non-Sensitive Crack Orientation Threshold). Ce dernier est calculé comme étant le maximum des minimums des valeurs de niveaux de gris des colonnes de l'image de la chaussée. Il est ainsi donné par l'équation 4.6.

$$NSCOT = max(min(I)) \tag{4.6}$$

avec $I$ représente l'image en niveaux de gris que nous cherchons à seuiller. Les tests réalisés ont montré que ce seuil est non sensible à l'orientation de la fissure dans l'image, c'est-à-dire quel que soit l'angle de la fissure ; il est capable de détecter les pixels les plus sombres de toute l'image. Le résultat obtenu lors de l'application de cette méthode de seuillage sur l'image en niveaux de gris présentée dans la figure 3.2 est donné par la figure 4.10.

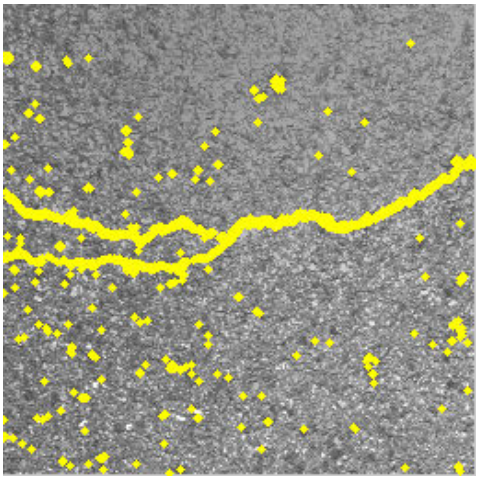


FIGURE *4.10 – Résultat obtenu après le seuillage adaptatif NSCOT.*

D'un côté, nous remarquons que cette méthode de seuillage (NSCOT) ne réduit pas le nombre de points amorces par rapport à la version MPS-V1 détaillée dans la section 4.2.2.2. En effet, une fois appliquée sur l'ensemble de l'image, le NSCOT peut sélectionner plusieurs pixels sombres dans une seule fenêtre $P \times P$ (figure 4.10), ce qui diffère des stratégies du seuil sur les minima-locaux (équation 3.2) et l'anisotropie locale (équation 4.5) qui permettent de sélectionner uniquement un seul pixel dans chaque fenêtre $P \times P$ (figure 4.8).

D'un autre côté, Il est important de noter que les pixels amorces sélectionnés par le NSCOT sont plus conformes à l'hypothèse photométrique illustrée dans la figure 4.13 parce qu'ils correspondent à la partie la plus sombre de la fonction de distribution de probabilité (pdf).

Enfin, afin de réduire le nombre de points amorces sélectionnés par le NSCOT, puis par conséquent le temps de calcul, une méthode de fusion de données est appliquée. Elle consiste à combiner la méthode de seuillage NSCOT avec l'ancienne stratégie (méthode FFA) détaillée dans la section précédente. Les points amorces communs entre ces deux stratégies sont sélectionnés puis ainsi considérés comme étant les points amorces finaux de l'étape 1 de l'algorithme. En effet, cela garantit la conservation d'un seul point amorce dans chaque fenêtre $P \times P$. Les points amorces finaux obtenus à l'issue de cette étape de fusion de données sont présentés dans la figure 4.11.

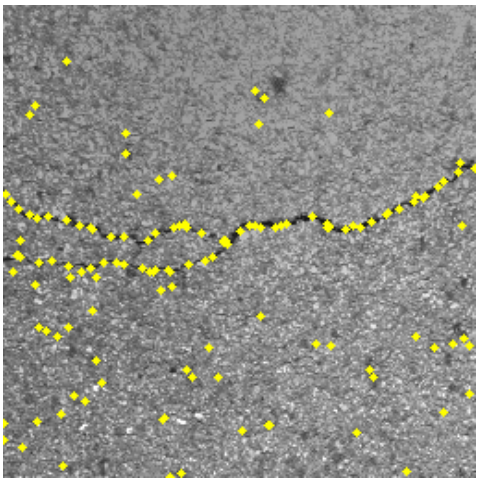
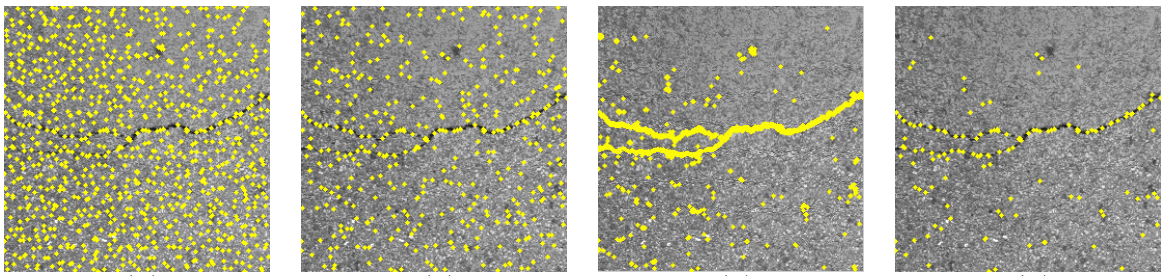


FIGURE *4.11 – Sélection des points amorces par la méthode MPS-V2.*

D'après la figure 4.11, la méthode de fusion de données proposée permet de réduire davantage le nombre des pixels amorces dans l'image. Par une comparaison avec le résultat obtenu avec la méthode FFA (figure 4.8.b), nous remarquons qu'un grand nombre de points amorces situés à l'extérieur de la fissure a été éliminé.

Pour plus de clarté, nous présentons dans la figure 4.12, une comparaison entre les trois méthodes MPS-V0, MPS-V1 et MPS-V2 pour la sélection des points amorces.



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

FIGURE *4.12 – Sélection des points amorces avec les trois version MPS – (a) MPS-V0 par sélection des minima-locaux dans des blocs de taille $P \times P$ - (b) MPS-V1 par analyse d'anisotropie locale dans des blocs de taille $P \times P$ - (c) Après le seuillage NSCOT - (d) MPS-V2 par fusion de données ; points amorces communs entre (b) et (c) dans des blocs de taille $P \times P$.*

La figure 4.12 présente en jaune les pixels amorces sélectionnés dans l'image de la chaussée selon les stratégies proposées. La sélection des minima-locaux (figure 4.12.a) fournit évidemment des points amorces qui sont dispersés sur toute l'image de la chaussée. Le post-traitement de ce dernier, basé sur l'application de la méthode FFA, réduit fortement le nombre de ces points (figure 4.12.b). Comme le montre la figure 4.12.c, le NSCOT peut fournir un nombre plus grand de points sombres que la stratégie des minima-locaux. Enfin, la combinaison avec la méthode FFA met en évidence les points amorces situés à l'intérieur de la fissure et réduit davantage ceux qui appartiennent au fond ou à la texture de l'image (figure 4.12.d).

La figure 4.13 montre l'influence des différentes stratégies proposées à l'étape 1 sur la distribution des niveaux de gris des points amorces. Le pdf (distribution de probabilité) des pixels de la PVT est représenté en noir ; il représente une partie plate dans l'intervalle de niveaux de gris {20–60}, puis une queue au-delà de 60, représentant les pixels de fissuration ayant un contraste plus faible que celui du fond de l'image. Le pdf des minima-locaux (LM) utilisé dans la version MPS-V0 fournit un pic important en dehors de ce dernier intervalle ({20–60}). Après l'application de l'anisotropie locale telle qu'elle est utilisée dans MPS-V1, les deux stratégies (LM + FFA) réduisent le nombre de points amorces brillants, à l'exception que le pdf fournit toujours un pic important en dehors de l'intervalle sélectionné, ce qui signifie qu'il existe encore des points amorces non pertinents dans l'image.

En comparaison, le NSCOT permet une sélection fiable des points amorces les plus sombres dans l'image, comme le montre la bonne correspondance avec le pdf des pixels de la PVT. En contrepartie, il a tendance de surestimer le nombre de points amorces au-delà de la valeur 60. De même, après l'application de l'analyse d'anisotropie locale comme c'est utilisée dans la version MPS-V2, les deux stratégies (NSCOT + FFA) convergent vers le même pdf dans l'intervalle de niveaux de gris {20–60}, ce qui signifie qu'elles sont équivalentes pour sélectionner les pixels amorces les plus sombres. En effet,

au-delà de cet intervalle, la stratégie NSCOT permet de réduire davantage le nombre de points finaux.

Il est nécessaire de noter que l'amplitude du pdf des pixels de la PVT est suffisamment grande par rapport à celle obtenue en appliquant les stratégies NSCOT + FFA. En effet, le pdf du PVT est établi en tenant compte de la totalité de la fissure (y compris l'épaisseur), tandis que le pdf du MPS-V2 correspond uniquement aux pixels amorces sélectionnés à l'intérieur de la fissure (sans prendre en compte l'épaisseur).

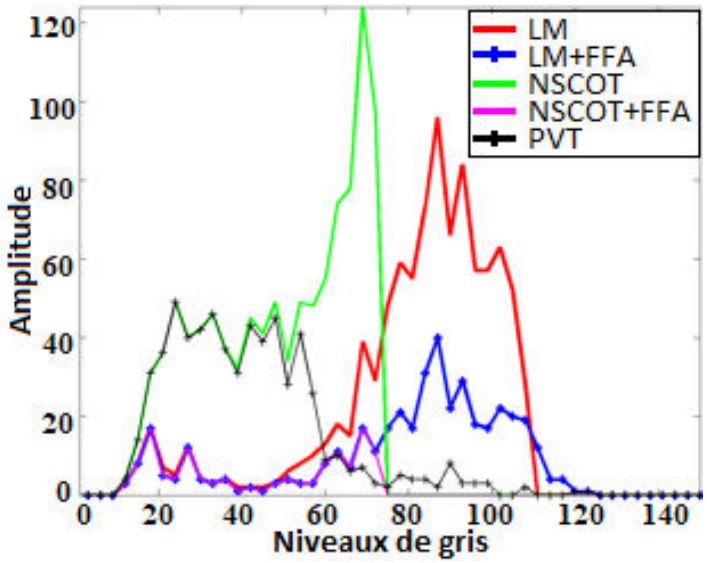



FIGURE *4.13 – Distribution des valeurs de niveaux de gris des points amorces sélectionnés par les différentes stratégies – LM : stratégie originale utilisée dans MPS-V0 (minima-locaux) - LM+FFA : stratégie utilisée dans MPS-V1 après l'analyse de l'anisotropie locale - NSCOT : stratégie utilisée après le seuillage adaptatif global - NSCOT+FFA : stratégie utilisée dans MPS-V2 après la fusion de données - PVT : Pixels de la pseudo-vérité terrain.*

Pour conclure, en comparaison avec la première étape de la méthode MPS-V0, les deux stratégies proposées dans ce chapitre (MPS-V1 et MPS-V2) impliquent un temps de calcul plus long en raison du calcul des plus courts chemins requis par l'analyse FFA. Comme indiqué dans la section suivante, cet écart est fortement compensé à l'étape 2 par le gain de temps dû à la réduction du nombre de chemins les plus courts à calculer.

### 4.2.3 Amélioration de l'étape 2 : Calcul optimisé des plus courts chemins entre paires de points amorces

Dans cette section, nous discutons la solution proposée pour optimiser l'étape 2 de l'algorithme MPS. En effet, il s'agit d'une stratégie plus efficace pour parcourir l'image.

La stratégie initiale (étoile) présentée dans la section 3.2.2.1 du chapitre 3 consiste

à appliquer l'algorithme de Dijkstra [120] sur des blocs de taille $3P \times 3P$ pour relier les points amorces entre eux. En effet, pour chaque bloc sélectionné dans l'image, elle calcule les chemins les plus courts entre le pixel amorce central et ses huit pixels voisins. Ainsi, cette stratégie produit des répétitions lors du calcul des chemins les plus courts, ce qui ralentit par conséquent l'exécution de l'algorithme. Un exemple illustrant cette répétition est présenté par la figure 4.14. En effet, le chemin minimal (en rouge) entre les deux points amorces 1 et 2 est calculé deux fois dans les deux sens.

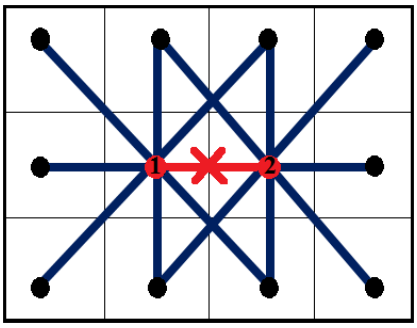



FIGURE *4.14 – Redondance obtenue lors de l'application de la stratégie étoile qui calcule huit chemins pour chaque point amorce central (en rouge) dans un bloc de taille* $3P \times 3P$.

Afin d'éviter cette redondance, une nouvelle stratégie a été proposée [136, 152]. Elle consiste à scanner l'image à traiter en utilisant des blocs de taille $3P \times 2P$. Par conséquent, en supposant que 2 points amorces centraux successifs partagent un seul chemin, il n'y a que quatre chemins à calculer au maximum pour chaque point amorce sans avoir des répétitions, comme le montre la figure 4.15.

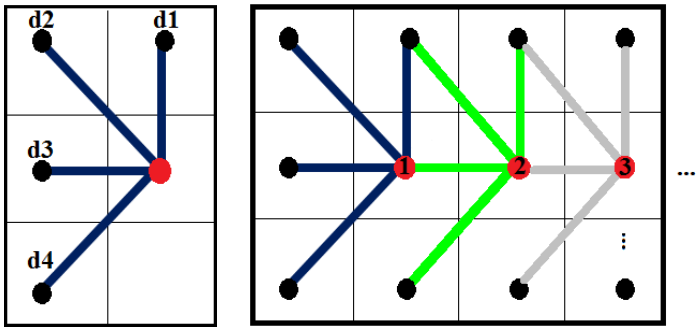



FIGURE *4.15 – Nouvelle stratégie proposée qui ne calcule que quatre chemins pour chaque point amorce central (en rouge) dans un bloc de taille* $3P \times 2P$.

Cette nouvelle stratégie assure une certaine réduction du temps d'exécution tout en maintenant la connexion entre toutes les morceaux de la fissure. Une illustration des quatre chemins obtenus avec l'algorithme de Dijkstra sur le bloc $3P \times 2P$ de la figure 3.4.b, est présentée par la figure 4.16.

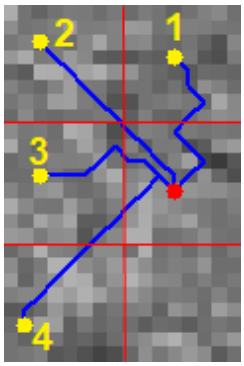
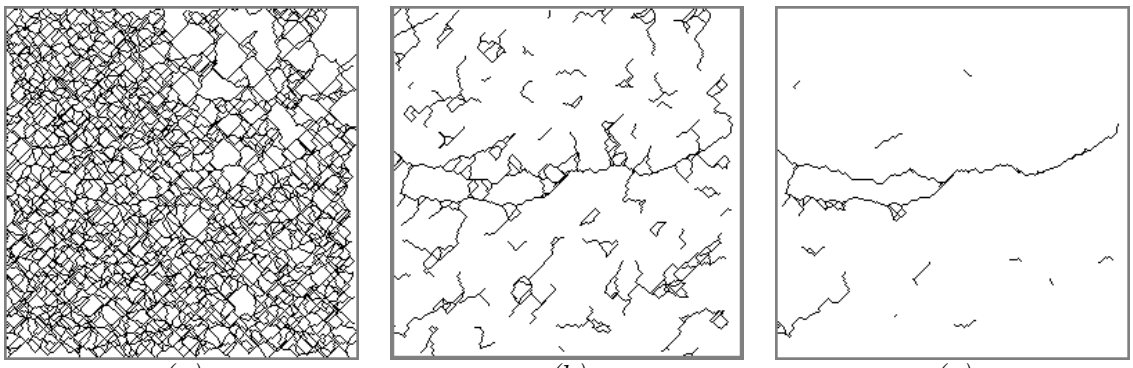



FIGURE *4.16 – Illustration des quatre chemins obtenus avec Dijkstra (SSSP) dans la fenêtre* $3P \times 2P$ *de la figure 3.4.b.*

Ensuite, cette optimisation proposée est appliquée aux derniers points amorces affichés sur les figures 4.12.b-d. Les résultats obtenus sont ainsi illustrés dans la figure 4.17.



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

FIGURE *4.17 – Calcul des chemins les plus courts après les trois différentes sélections des points amorces – (a) MPS-V0 dans la figure 4.12.a - (b) MPS-V1 : méthode FFA dans la figure 4.12.b - (c) MPS-V2 : méthode de fusion de données dans la figure 4.12.d.*

Selon la figure 4.17, le nombre de chemins les plus courts a fortement diminué après les deux optimisations réalisées pour sélectionner les points amorces. Néanmoins, la dernière version fournit la segmentation la plus proche de la pseudo-vérité terrain illustrée dans la figure 3.23.b. De plus, la stratégie 4-connectivités proposée à la place de la stratégie étoile a permis de réduire le temps de calcul d'un facteur 5 pour la version MPS-V1 (méthode FFA) et d'un facteur 20 pour la version MPS-V2 (méthode de fusion de données : NSCOT+FFA).

### 4.2.4 Amélioration de l'étape 3 : Mise à jour du paramétrage

Comme présenté dans la section 3.2.3 du chapitre 3, l'étape 3 de l'algorithme MPS permet de sélectionner les chemins minimaux parmi les chemins les plus courts calculés à l'étape 2. Le chemin minimal est caractérisé par sa valeur de coût donnée par

l'équation 3.4 et supposée être inférieure au seuil $T_c$ défini par l'équation 3.5.

Tout d'abord, cette section met à jour la répartition des coûts des chemins en suivant les améliorations proposées aux étapes 1 et 2 de l'algorithme, ensuite elle effectue une analyse de sensibilité pour faire correspondre le seuil $T_c$ à la nouvelle répartition des coûts.

#### 4.2.4.1 Répartition des valeurs de coûts des nouveaux chemins calculés

La figure 4.18 illustre les histogrammes des valeurs coûts des chemins calculés à l'étape 2 après la sélection des points des amorces par la méthode MPS-V1 et par la méthode MPS-V2. Pour plus de clarté, l'axe vertical est défini en échelle logarithmique afin de rendre plus visible la faible amplitude des pdfs. À des fins de comparaison, le résultat obtenu par la méthode MPS-V0 est affiché dans les deux histogrammes en lignes continues. Le premier pic à gauche repose sur des fissures qui se trouvent entièrement ou partiellement à l'intérieur de la fissure, c'est-à-dire les chemins vrais positifs, tandis que le second pic à droite devrait correspondre à des fissures appartenant au fond ou à la texture de l'image, à savoir des chemins faux positifs. Nous notons que pour les deux versions optimisées (MPS-V1 et MPS-V2), la distribution des valeurs de coûts présente un pdf bimodal ayant toujours le deuxième pic plus faible que celui de la méthode d'origine. Ainsi, le pic correspondant à la texture de l'image est réduit d'un rapport de 6 et 20, respectivement. Toutefois, les deux méthodes optimisées ont supprimé quelques chemins vrais positifs appartenant à la partie gauche de l'histogramme.

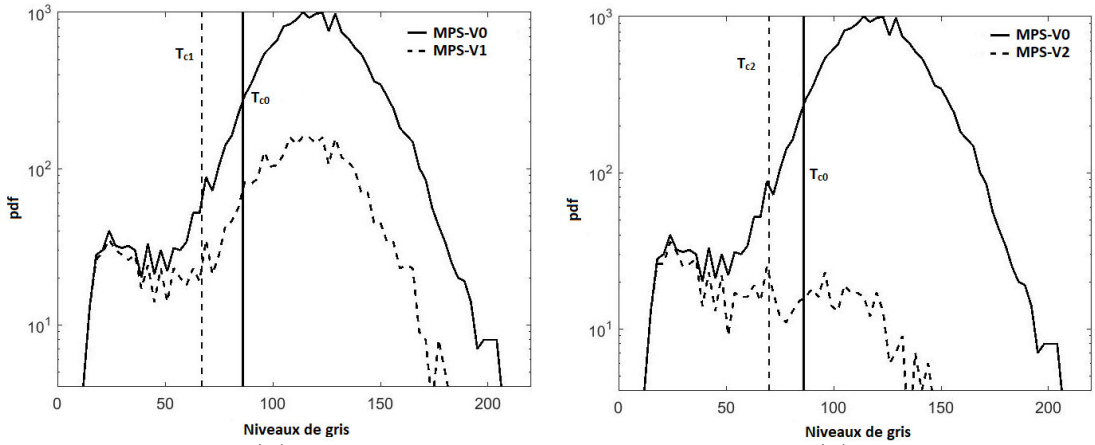
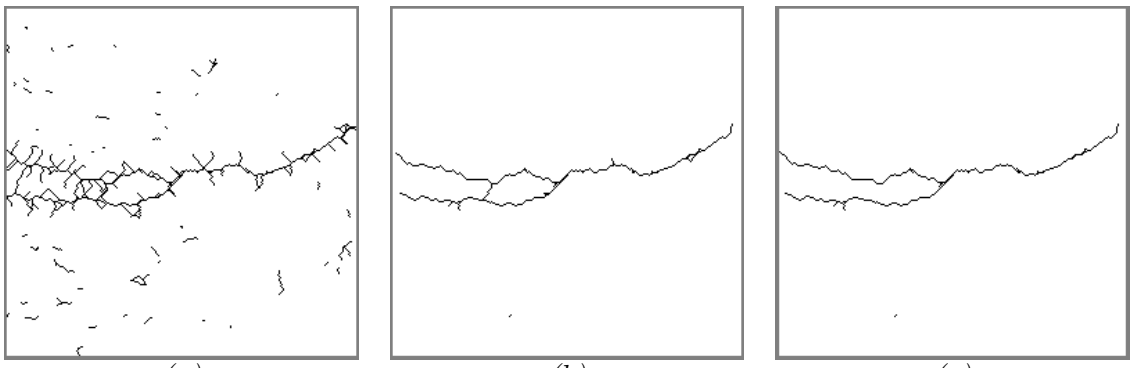



FIGURE *4.18 – Répartition des valeurs de coûts des chemins (pdf) associée à la figure 4.17 – (a) MPS-V0 vs. MPS-V1 - (b) MPS-V0 vs. MPS-V2.*

Le squelette de fissuration final présenté par la figure 4.19 est obtenu en sélectionnant les chemins minimaux dont les valeurs de coûts sont inférieures au seuil $T_c$ défini par l'équation 3.5. Les deux nouveaux seuils, dénommés respectivement $T_{c1}$ et $T_{c2}$, sont

affichés en traits verticaux sur la figure 4.18. Le calcul de ces deux derniers seuils est détaillé dans la section suivante.



(a) (b) (c)

FIGURE *4.19 – Segmentation du squelette de fissuration après le seuillage appliqué sur la distribution des valeurs de coûts des chemins illustrée dans la figure 4.18 par : (a) MPS-V0 avec* $T_{c0} = 86$ *- (b) MPS-V1 avec* $T_{c1} = 67$ *- (c) MPS-V2 avec* $T_{c2} = 70$.

#### 4.2.4.2 Analyse de la sensibilité

La réduction de la quantité des chemins les plus courts modifie fortement la valeur de la moyenne $\mu_c$ et de l'écart-type $\sigma_c$ des valeurs coûts. Par conséquent, le réglage initial du paramètre $K_c$ n'est plus valide après les optimisations effectuées. A ce stade, une nouvelle analyse de sensibilité est effectuée pour faire correspondre les seuils $T_{c1}$ et $T_{c2}$ à la répartition des nouvelles valeurs de coûts. Pour cela, le paramètre $K_c$ introduit dans l'équation 3.5 varie dans une certaine plage afin de scanner un grand intervalle de niveaux de gris, à savoir, {30–100} pour les deux méthodes, garantissant ainsi un optimum global à atteindre.

Pour chaque valeur de $K_c$, le squelette de fissuration obtenu est comparé à la pseudo-vérité terrain et les taux des faux positifs, faux négatifs et vrais positifs sont ainsi déterminés afin de calculer les coefficients de sensibilité (S), de précision (P) et de DSC définis par les équations 3.8, 3.7 et 3.9. Cette analyse de sensibilité est effectuée pour les deux versions proposées MPS-V1 et MPS-V2.

Pour la clarté, nous notons que l'analyse de sensibilité réalisée correspond à l'image en niveaux de gris présentée dans la figure 3.2.a. Selon la figure 4.20, les deux méthodes MPS-V1 et MPS-V2 présentent des variations proches de la forme et de l'amplitude des trois coefficients S, P et DSC. En comparaison avec le résultat obtenu par la méthode MPS-V0, les modifications proposées aux étapes 1 et 2 fournissent un paramétrage plus sensible pour la constante $K_c$. Aisni, le coefficient de similarité DSC atteint une valeur maximale à $K_c = 0.8$ pour MPS-V1 et $K_c = -0.6$ pour MPS-V2.

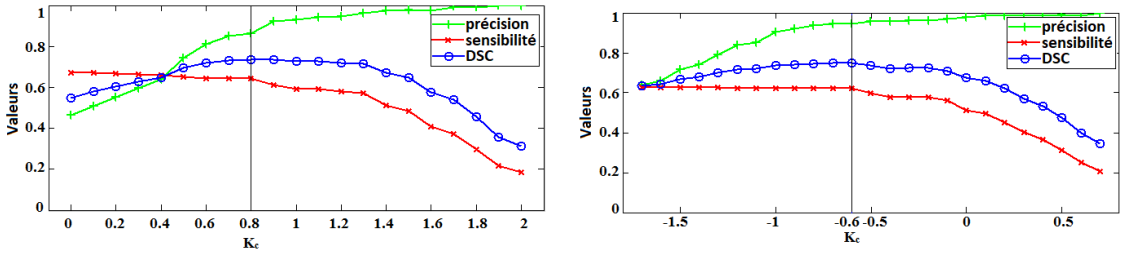



FIGURE *4.20 – Illustration de la variation des coefficients de précision, sensibilité et DSC en fonction de $K_c$ pour les deux méthodes (a) MPS-V1 et (b) MPS-V2.*

### 4.2.5 Post-traitement

La phase de post-traitement introduite dans cette section correspond au même post-traitement utilisé dans la méthode MPS originale (MPS-V0). En effet, ces étapes sont détaillées dans les sections 3.2.4 et 3.2.5 du chapitre 3 à la base du travail cité dans [117].

Comme le montre la figure 4.19, le résultat de la segmentation brute à l'étape 3 a été amélioré par les deux versions proposées (MPS-V1 et MPS-V2), mais certains artefacts doivent encore être retirés du squelette de fissuration. À cette fin, nous appliquons la phase de post-traitement (étapes 4 à 5 de l'algorithme) qui permet d'affiner le résultat de la segmentation obtenue. Il est crucial de noter que l'amélioration des trois premières étapes de l'algorithme a considérablement simplifié le processus de la phase de post-traitement. En effet, la réduction du nombre des points amorces ainsi que le nombre des chemins minimaux dans l'image, a permis de réduire le nombre des instructions à suivre pour raffiner le squelette de fissuration et ainsi estimer l'épaisseur de la fissure. Les figures 4.21 et 4.22 illustrent les résultats obtenus lors de l'application des deux étapes de la phase du post-traitement sur les squelettes de fissuration présentés par la figure 4.19.

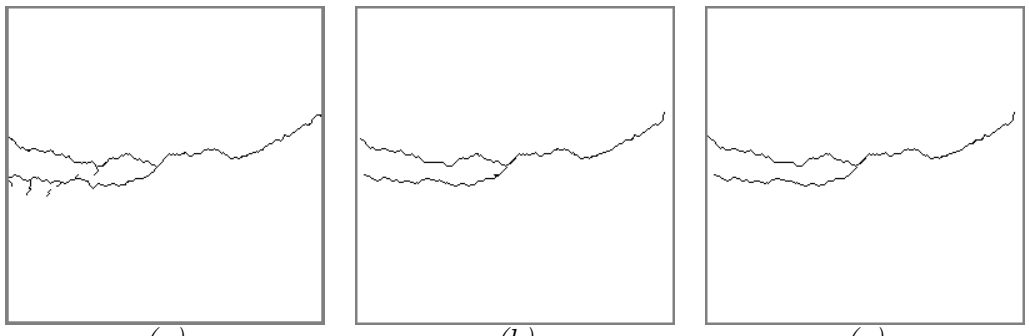
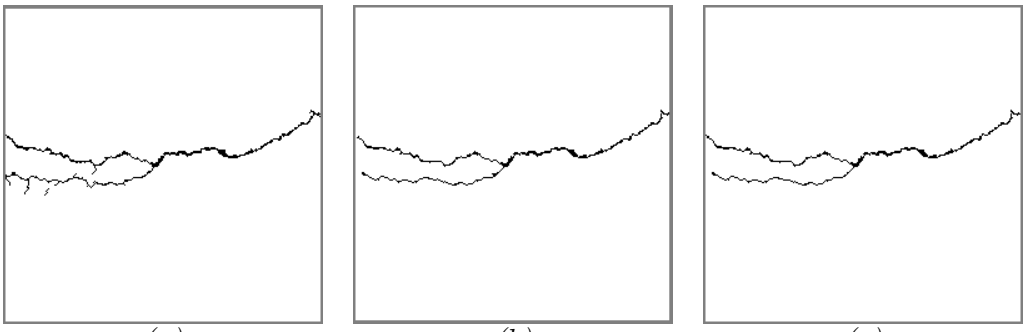



FIGURE *4.21 – Résultats obtenus après l'application de l'étape 4 de l'algorithme (Raffinage du squelette de fissuration) – (a) MPS-V0 - (b) MPS-V1 - (c) MPS-V2.*



(a) (b) (c)

FIGURE *4.22 – Résultats obtenus après l'application de l'étape 5 de l'algorithme (Estimation de l'épaisseur) – (a) MPS-V0 - (b) MPS-V1 - (c) MPS-V2.*

## 4.3 Étude de cas sur des images de chaussée

Dans cette section, les performances des deux versions améliorées de la méthode MPS, à savoir MPS V1 et MPS-V2, sont comparées à la version originale (MPS-V0) sur des images de chaussée acquises par le système Aigle-RN [4, 5] et provenant de la base de données citée dans [149].

### 4.3.1 Base de données testées

Afin d'évaluer les performances des méthodes proposées, nous avons considéré 33 images de chaussée prises à incidence verticale [149]. Elles ont été collectées à la vitesse de la circulation sous éclairage contrôlé par le système d'imagerie Aigle-RN [4,5], afin de surveiller périodiquement l'état de la surface de la chaussée du réseau national français. Comme nous l'avons mentionné dans la section 1.6.1.1 du chapitre 1, ces images ont une taille $1900 \times 924$ pixels et couvrent une surface de chaussée de 3.8 m de largeur et 1 m de longueur (la taille d'un pixel est d'environ $2 \times 2$ $mm^2$).

Les images de la chaussée sont acquises par trois caméras monochromes à réglage indépendant. Afin d'assurer un fonctionnement du jour comme de nuit, un éclairage de la surface de la chaussée en examen a été appliquée en utilisant une technique stroboscopique. Cela se traduit par des conditions d'éclairage non uniformes et des calibrages d'image différents, qu'il a été trouvé difficile à atténuer par des techniques de pré-traitement. La partie droite des images de la chaussée (collectées par la troisième caméra) montre en particulier un fond plus large, ce qui renforce la texture de l'image par rapport à la partie gauche. Dans ce cadre, nous avons décidé de considérer les images de la chaussée comme étant deux sous-ensembles différents pour les tests. Ces deux sous-ensembles ont une taille de $991 \times 462$ pixels (0.46 MPx pour la partie gauche) et $311 \times 462$ (0.15 MPx pour la partie droite), comme illustré sur les figures 4.23 et 4.24, respectivement. De plus, l'évaluation des résultats obtenus a été effectuée en se basant sur les images pseudo-vérité terrain correspondantes citées dans [149].

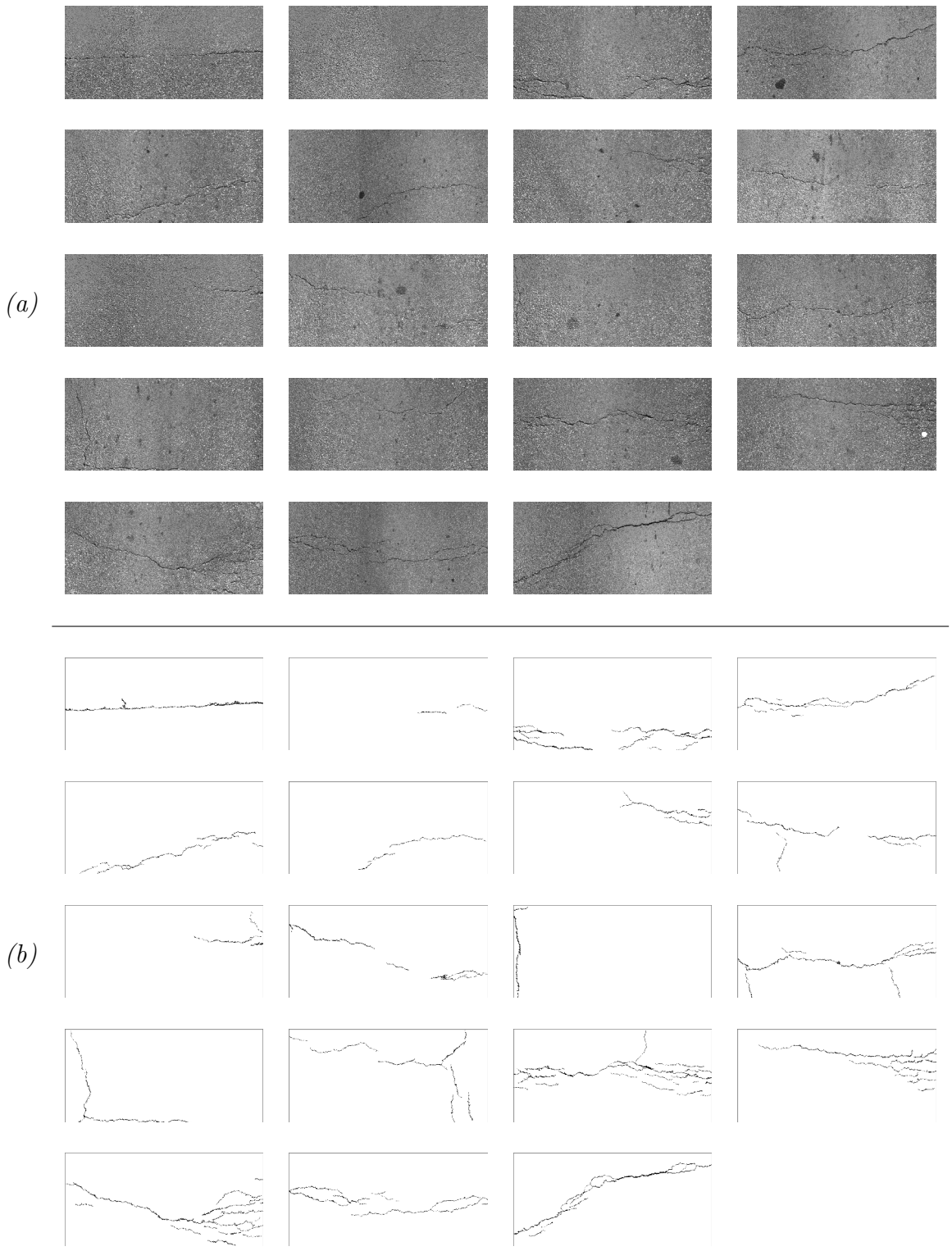



FIGURE *4.23 – Sous-ensemble 1 : 19 images de chaussée de taille* $991 \times 462$ *pixels acquises par le système Aigle-RN – (a) Images originales en niveaux de gris - (b) Images pseudo-vérité terrain correspondantes.*

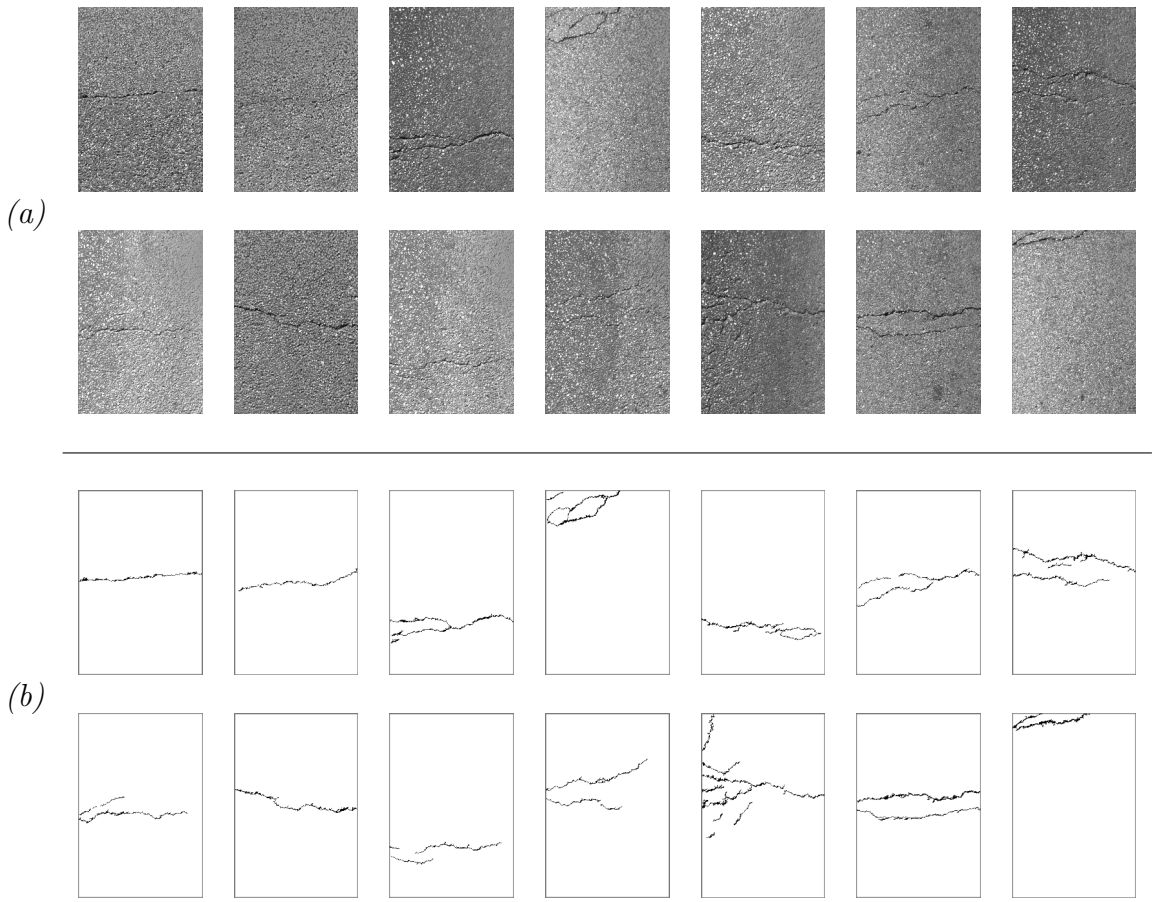



FIGURE *4.24 – Sous-ensemble 2 : 14 images de chaussée de taille* 311 × 462 *pixels acquises par le système Aigle-RN –(a) Images originales en niveaux de gris - (b) Images pseudo-vérité terrain correspondantes.*

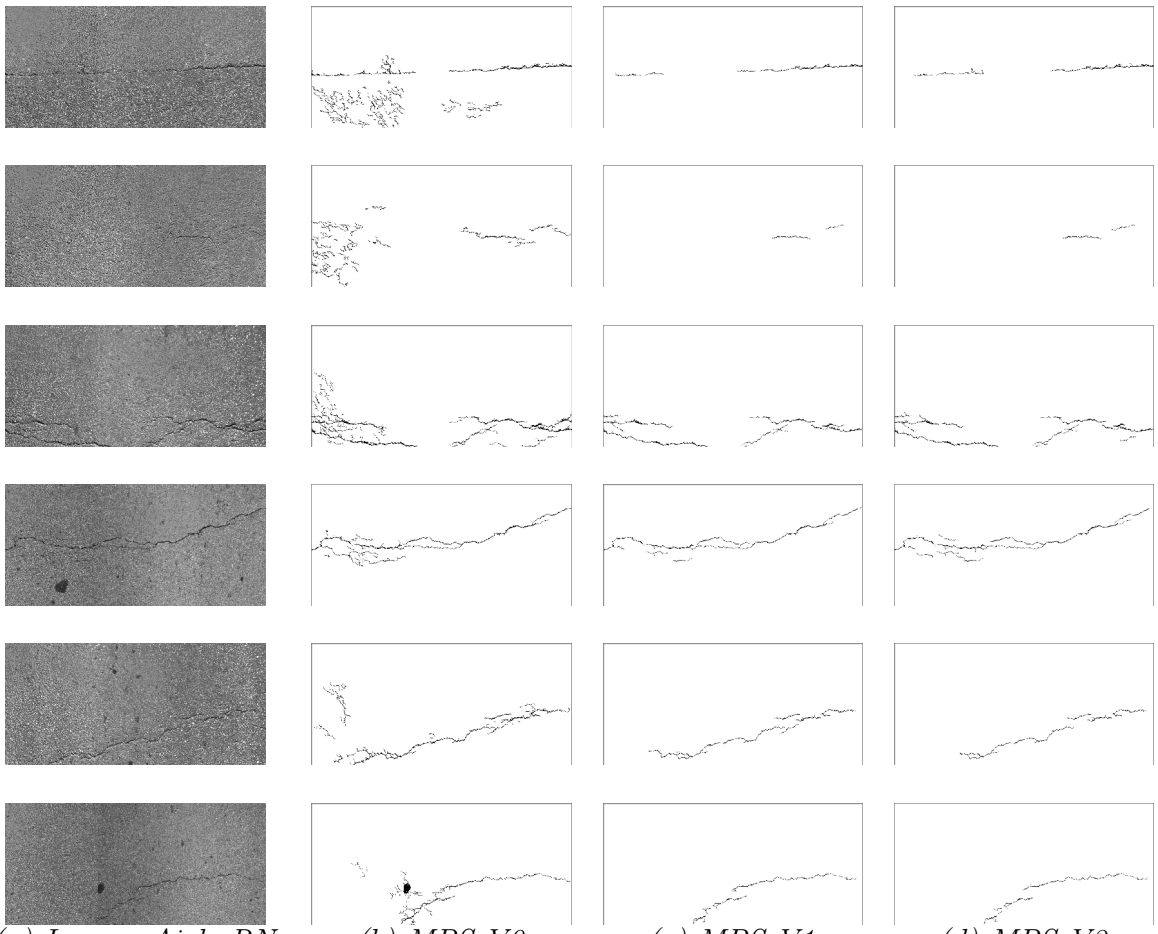
### 4.3.2 Tests et validation

Parmi la série des images de la base de données, les six images présentées dans la figure 4.25 ont été sélectionnées dans cette section comme étant des exemples illustratifs permettant d'évaluer les performances des différentes versions de la méthode MPS. En effet, ces images ont été utilisées dans la littérature pour réaliser une analyse comparative de certaines méthodes de segmentation de fissures [98, 110, 117, 123, 157]. Selon la figure 4.25.a, les quatre premières images ont été utilisées dans [110, 123] pour comparer la version MPS-V0 à d'autres méthodes (citées dans le chapitre 1), à savoir la méthode de Markov, la méthode heuristique, la méthode GC et la méthode FFA. Les trois dernières images ont été récemment utilisées dans [98] pour une comparaison avec la méthode CrackIT [96] ainsi que l'algorithme CrackForest [98]. La quatrième image, qui a été utilisée jusqu'à présent pour illustrer les différentes étapes de la méthode MPS, est la seule image pouvant supporter la comparaison visuelle avec l'ensemble des autres méthodes citées.

Les six images sélectionnées présentent des cas typiques et des difficultés rencontrées lors de la phase de segmentation. En effet, la plupart d'entres elles comportent de longues fissures transversales avec et sans ramifications. Les fissures minces sont montrées dans l'image 2 et au milieu de l'image 1. Les deux premières images ont été considérées dans [110, 123] comme étant les images les plus texturées et donc les plus difficiles à segmenter. De plus, les images 4 et 6 comprennent également des taches sombres susceptibles de perturber la segmentation de la fissure.

La figure 4.25 permet une analyse visuelle des résultats finaux obtenus après la segmentation de chacune des six images décrites précédemment. En effet, le regroupement des trois versions de MPS sur un même graphique montre l'amélioration constante des performances. Le tableau 4.1 illustre le temps de calcul et la valeur DSC définie dans la section 3.3.2 du chapitre 3, en tant qu'une évaluation conventionnelle des performances des deux méthodes proposées MPS-V1 et MPS-V2. De plus, les travaux récents dans [98] permettent de comparer les méthodes que nous proposons à la méthode de Crack-Forest, basée sur un algorithme d'apprentissage automatique supervisé. Le résultat de la segmentation est disponible pour l'image 4 uniquement en comparant la figure 12 de [98] à la figure 4.25 de ce chapitre. Crack-Forest fournit une segmentation dont le rendu visuel est similaire à FFA. Nous notons que l'algorithme Crack-Forest surestime l'épaisseur de la fissure et, par conséquent, il est difficile de détecter les ramifications de la fissure ainsi que les autres détails minces dans l'image. La valeur du DSC moyenne présentée dans [98] pour les méthodes MPS et FFA est surestimée par rapport au résultat présenté dans cette section et dans le résultat initial cité dans [117]. Cette divergence entre les résultats (avec les mêmes méthodes et sur les mêmes données d'entrée) est probablement due à la marge la plus grande considérée dans [98] autour des pixels de la pseudo-vérité terrain, contrairement à ce que nous avons considéré dans notre travail, à savoir, 5 pixels contre 2 pixels, respectivement. C'est la raison pour laquelle, toutes les méthodes de segmentation testées dans [98] atteignent des valeurs de DSC plus élevées. Toutefois, la marge de 2 pixels telle qu'elle est utilisée dans le présent document permet d'un côté d'obtenir des scores DSC plus petits, et d'un autre côté une évaluation plus sensible des performances tout en augmentant les différences de score entre les méthodes.

Le tableau 4.1 montre que les deux versions améliorées fournissent des valeurs de DSC plus grandes par rapport à la version standard (MPS-V0). En moyenne, la méthode MPS-V2 atteint une valeur de DSC de 81%, tandis que la méthode MPS-V1 atteint 80% contre 57% pour la méthode MPS-V0. De plus, le temps de calcul a été divisé en moyenne par un facteur de 17 et 60 pour les MPS-V1 et MPS-V2, respectivement.



*(a) Images Aigle-RN* *(b) MPS-V0* *(c) MPS-V1* *(d) MPS-V2*

FIGURE *4.25 – Exemples illustrant la segmentation finale des six premières images de la figure 4.23.*

| | MPS-V0 | | MPS-V1 | | MPS-V2 | |
|---|---|---|---|---|---|---|
| | **Temps (s)** | **DSC (%)** | **Temps (s)** | **DSC (%)** | **Temps (s)** | **DSC (%)** |
| 1 | 845 | 48 | 41 | 59 | **11** | **62** |
| 2 | 815 | 21 | 36 | 77 | **11** | **80** |
| 3 | 866 | 77 | 52 | 82 | **16** | **83** |
| 4 | 738 | 80 | 46 | 93 | **17** | **94** |
| 5 | 752 | 66 | 48 | **83** | **9** | 77 |
| 6 | 722 | 49 | 51 | 85 | **11** | **90** |
| Moy. | 790 | 57 | 46 | 80 | **13** | **81** |

TABLE *4.1 – Temps de calcul global et valeurs DSC obtenus en traitant les six images de la figure 4.23 avec les trois versions de MPS. Les valeurs en gras indiquent les meilleures valeurs obtenues à la fin du traitement.*

### 4.3.3 Résultats statistiques

Enfin, les trois versions de l'algorithme MPS sont testées sur les deux sous-ensembles d'images de chaussée présentés par les figures 4.23 et 4.24, respectivement.

La valeur moyenne de DSC est ainsi calculée sur chacun des sous-ensembles d'images, comme illustré dans la figure 4.26. Globalement, pour les images de taille 991 × 462 (respectivement, 311 × 462), la valeur moyenne de DSC a augmenté progressivement de 64% (respectivement, 70%) pour la méthode MPS-V0 à 74% (respectivement, 71%) pour la méthode MPS-V1 et à 75% (respectivement, 72%) pour la méthode MPS-V2. Notons que l'amélioration semble être mineure sur le deuxième sous-ensemble d'images, i.e., images de taille 311 × 462, qui souffre du problème de calibration et présente une texture d'image très élevée. Comme prévu, nous remarquons que les deux versions optimisées augmentent le taux des vrais positifs (bonnes détections) et réduisent le taux de faux positifs (fausses alarmes).

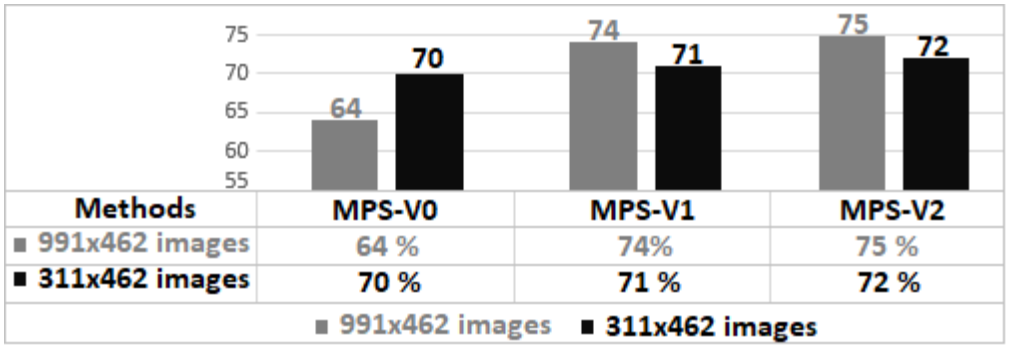
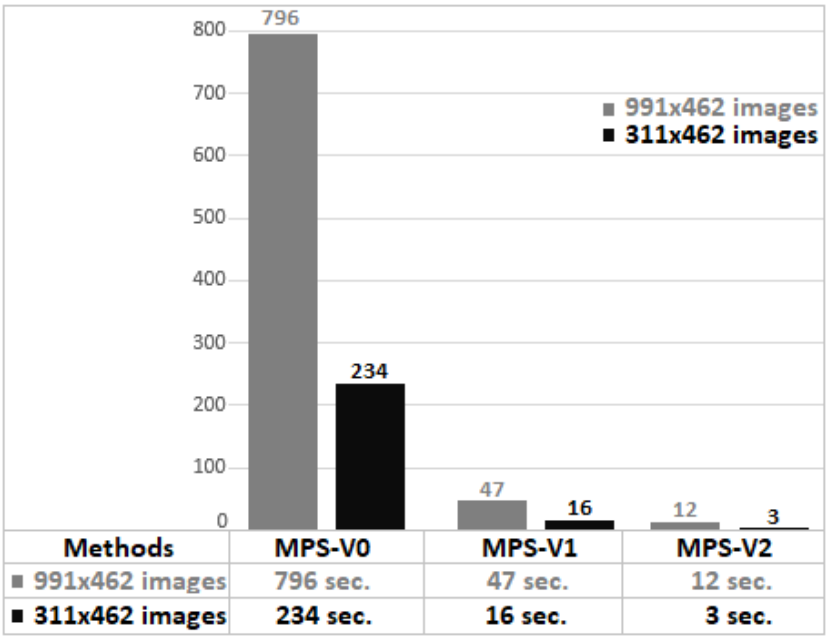



FIGURE *4.26 – Valeurs moyennes de DSC obtenues lors de l'application des trois versions MPS-V0, MPS-V1 et MPS-V2 sur les* 33 *images illustrées dans les figures 4.23 et 4.24.*

Selon la figure 4.27, le temps de calcul a été considérablement réduit lors de l'application des deux dernières versions. En effet, MPS-V0, MPS-V1 et MPS-V2 atteignent une vitesse de calcul moyenne de 28.5 mn/Mpx, 106 s/Mpx et 23.5 s/Mpx, respectivement. Le temps de calcul est ainsi divisé par un facteur de 16 et 70 respectivement pour les deux versions MPS-V1 et MPS-V2.




| Methods | MPS-V0 | MPS-V1 | MPS-V2 |
|---|---|---|---|
| 991x462 images | 796 sec. | 47 sec. | 12 sec. |
| 311x462 images | 234 sec. | 16 sec. | 3 sec. |

FIGURE *4.27 – Temps de calcul moyens pour traiter les* 33 *images illustrées dans les figures 4.23 et 4.24 selon les méthodes MPS-V0, MPS-V1 et MPS-V2.*

Nous notons que tous les traitements ont été effectués sous programmation Matlab et calculés sur un processeur Intel $Core^{TM}$ i7-4790 @ 3.60 GHz avec 8 Go de RAM.

Les résultats montrent que les deux versions améliorées fonctionnent mieux que la version originale avec un temps de calcul plus court et un coefficient de similarité plus grand (DSC). Par conséquent, il est utile d'adopter à l'avenir la version la plus récente et la plus rapide, à savoir MPS-V2. Notons que cette version adoptée est dénommée par la suite de ce document OMPS (Optimized Minimal Path Selection).

## 4.4 Conclusion

Ce chapitre a présenté les deux améliorations de l'algorithme MPS réalisées afin de parvenir à une segmentation robuste et efficace des fissures dans des images de chaussée en niveaux de gris. Nous notons que nos contributions se sont concentrées sur les trois premières étapes de l'approche pour produire un squelette de fissuration plus fiable.

Pour l'étape 1 de l'algorithme, deux techniques de traitement ont été proposées pour réduire davantage le nombre de points amorces dans l'image. La première technique est basée sur une méthode d'analyse d'anisotropie locale (FFA). La deuxième technique est définie comme étant une combinaison de la première technique (FFA) avec une méthode de seuillage adaptatif (NSCOT), permettant une sélection plus précise des points amorces. Nous notons que la seconde technique basée sur la fusion de données a permis de réduire considérablement le taux de faux positifs et de rapprocher le résultat

de la segmentation à la pseudo-vérité terrain (PVT). Pour l'étape 2 de l'algorithme, le calcul des chemins les plus courts a été optimisé grâce à une nouvelle stratégie de balayage permettant de scanner et de parcourir la totalité de l'image de la chaussée suivant des fenêtres de taille $3P \times 2P$. Il est ainsi crucial de noter que la réduction du nombre des points amorces obtenue après avoir optimisé la première étape de l'algorithme, a permis considérablement de réduire le nombre de chemins les plus courts à calculer dans l'image. Ensuite, concernant l'étape 3, le paramétrage a été adapté à la nouvelle répartition des coûts des chemins de manière à conserver un maximum de vrais positifs et un minimum de faux positifs.

L'évaluation des performances des deux méthodes proposées (MPS-V1 et MPS-V2) ainsi que l'étude de cas présentées à la section 4.3 ont montré que les deux versions mises à jour améliorent la valeur du DSC par rapport à la version d'origine (MPS-V0). En effet, parmi les deux versions optimisées, la deuxième MPS-V2 (dénommée OMPS par la suite dans ce document) permet la réduction la plus importante du temps de calcul sans perte de performance.

Afin de valider notre travail, d'autres tests ont été réalisés en appliquant les approches proposées sur des images acquises par d'autres systèmes d'acquisitions, à savoir le système de nouvelle génération incluant la dimension $3D$ du relevé de surface orienté vers le bas (par exemple, LCMS [124]) et le système de visualisation en perspective (par exemple, $VIAPIX^{®}$ [125]). Les résultats obtenus ont montré que ces approches semblent insuffisantes dans certains cas. En effet, elles dépendent principalement des conditions d'acquisition des images. Par exemple, lors de la présence d'une zone ombragée dans l'image, ces méthodes (MPS-V0, MPS-V1 et MPS-V2) prennent en compte toute cette zone sombre et la considère comme étant une zone de fissuration (voir figure 4.28). Afin d'éviter ce problème et ainsi rendre les méthodes de segmentation plus robustes aux conditions d'éclairage, le chapitre 5 présente en détail une nouvelle approche que nous avons proposée (ADFD : Automatic Darkest Filament Detection) pour la détection automatique des fissures sur chaussée.

*(a) MPS-V0* *(b) MPS-V1* *(c) MPS-V2 (ou OMPS)*

FIGURE *4.28 – Résultats de la détection obtenus lors de l'application des trois versions de l'algorithme MPS sur une image* $VIAPIX^{®}$ *de taille* $1216 \times 1600$ *pixels.*

# Chapitre 5

## Détection de fissures par sélection de filaments sombres

### Sommaire


5.1 Introduction . . . . . **155**
5.2 Approche proposée . . . . . **155**
5.3 Synoptique générale . . . . . **156**
5.3.1 Pré-traitement . . . . . 156
5.3.1.1 Nettoyage de l'image originale . . . . . 157
5.3.1.2 Sélection des points amorces . . . . . 161
5.3.2 Traitement . . . . . 169
5.3.2.1 Calcul des filaments sombres . . . . . 170
5.3.2.2 Sélection des filaments de coût minimal . . . . . 170
5.3.3 Post-traitement . . . . . 171
5.3.3.1 Raffinage du squelette de fissuration . . . . . 171
5.3.3.2 Estimation de l'épaisseur de fissure . . . . . 186
5.4 Tests et validation . . . . . **186**
5.4.1 Tests sur des images Aigle-RN . . . . . 187
5.4.2 Tests sur des images $VIAPIX^{®}$ . . . . . 191
5.4.3 Tests sur des images LCMS "élévation" . . . . . 194
5.5 Conclusion . . . . . **197**

## 5.1 Introduction

Nous avons présenté au chapitre précédent la méthode OMPS (Optimized Minimal Path Selection) pour la segmentation automatique de fissures sur chaussées. Les tests réalisés au chapitre 4 ont montré que cette méthode était davantage dédiée au traitement des images acquises à incidence verticale sous éclairage contrôlé, e.g., images Aigle-RN [4] ou LCMS [124]. En contrepartie, la méthode OMPS est sensible aux conditions d'éclairage dans les images panoramiques en lumière naturelle, e.g., images $VIAPIX^{®}$ [125]. Par exemple, OMPS détecte les zones d'ombre sous éclairage ambiant comme des zones de fissuration [158]. L'analyse de cette zone nécessite un calcul supplémentaire qui augmente considérablement le temps de traitement sans réduire le taux de fausses alarmes.

Pour rendre les méthodes de segmentation automatique plus robustes aux conditions d'éclairage, nous proposons dans ce chapitre une nouvelle méthode nommée ADFD (Automatic Darkest Filament Detection). Elle est basée sur une combinaison de techniques de segmentation colorimétrique et géométrique, qui font davantage intervenir les caractéristiques locales de l'image. Cette nouvelle méthode est toujours basée sur un algorithme de chemin minimal pour la segmentation de fissures, mais diffère des travaux existants par trois aspects novateurs.

Premièrement, une phase de détection de contours est réalisée en pré-traitement afin de localiser dans l'image les zones de conditions d'éclairage similaire (ombres, etc.). Deuxièmement, la méthode ADFD inclut une nouvelle étape d'analyse de connexité entre les fissures afin d'éviter la suppression des fissures de petites tailles comme dans [117, 123, 138, 152]. Troisièmement, elle peut être adaptée pour analyser et traiter des images $3D$ acquises par le système LCMS [124]. Finalement, nous montrons que la méthode ADFD permet d'améliorer les résultats de segmentation des images $2D$ panoramiques en lumière naturelle, mais également les résultats obtenus sur les images conventionnelles en éclairage contrôlé. La section suivante présente en détail la méthode ADFD introduite dans [158].

## 5.2 Approche proposée

Dans cette section, nous allons présenter le principe de la nouvelle méthode non-supervisée de segmentation de "fissures" de chaussée, nommée ADFD (*Automatic Darkest Filament Detection*).

D'après les hypothèses citées au paragraphe 1.5.1 du chapitre 1, nous nous appuyons dans cette méthode sur les caractéristiques de "**forme**" et de "**luminosité**" de la fissure.

En se basant sur ces deux caractéristiques photométrique et géométrique de la fissure ainsi que sur la notion du "**chemin minimal**", nous cherchons à sélectionner tout

d'abord les pixels ayant une forte probabilité d'appartenir à une zone sombre dans l'image, ensuite à les relier en calculant les chemins les plus courts par l'algorithme de Dijkstra [120, 152] détaillé dans la section 3.2.2.2 du chapitre 3.

## 5.3 Synoptique générale

Le synoptique de la méthode proposée obéit à un schéma classique illustré à la figure 5.1, à savoir, une phase de pré-traitement, une phase de traitement et une phase de post-traitement.

La première phase de la méthode ADFD consiste à sélectionner les pixels les plus sombres, qui sont par hypothèse susceptibles d'appartenir à la structure de la fissure. Ensuite, la deuxième phase consiste à segmenter l'image en deux classes représentant respectivement les fissures (défauts) et la chaussée (fond ou texture d'image). Enfin, la troisième phase permet d'extraire les défauts à l'échelle globale de l'image. Le but est ainsi d'estimer le niveau de dégradation de la chaussée d'une manière précise, robuste et discriminante.

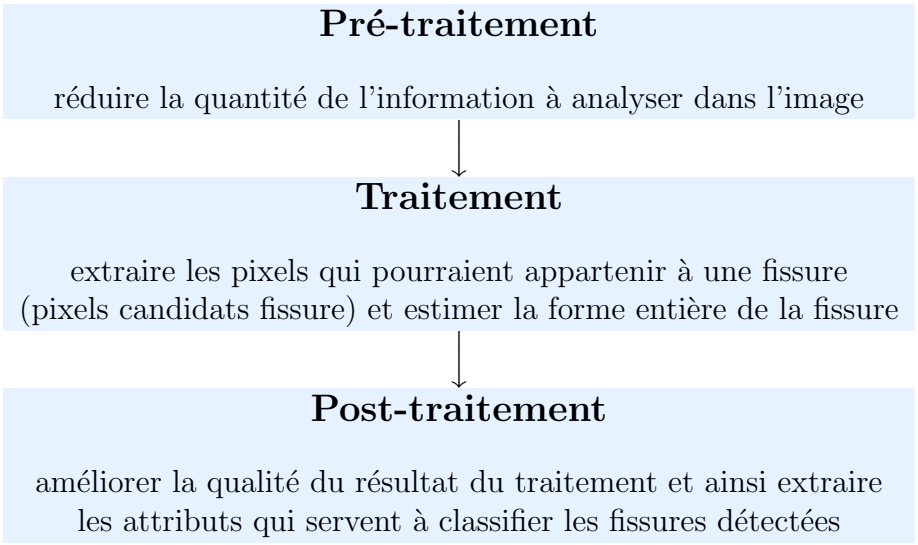
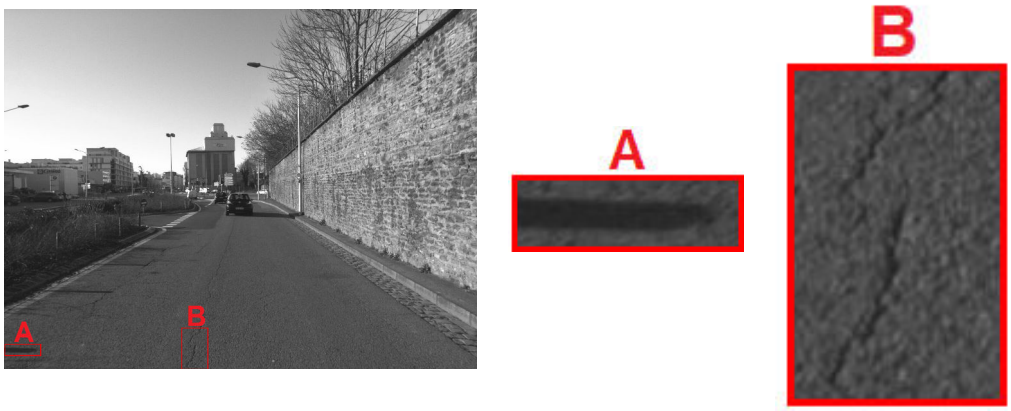



FIGURE *5.1 – Synoptique de la méthode ADFD.*

### 5.3.1 Pré-traitement

L'objectif de cette phase est d'améliorer les caractéristiques visuelles de l'image en niveaux de gris et de rendre le traitement plus efficace. Cette phase sert généralement à rehausser le contraste, et a pour effet de réduire le bruit présent dans l'image. La figure 5.2 illustre la difficulté de la segmentation en éclairage ambiant : l'information qui nous intéresse (la zone de fissuration étiquetée *"B"*) est proche en niveaux de gris de la zone d'ombre portée (étiquetée *"A"*).




(a) (b) (c)

FIGURE 5.2 – *Exemple d'une image en niveaux de gris illustrant la présence d'une zone fissurée et une zone ombragée sur la route – (a) Image $VIAPIX^{®}$ en niveaux de gris contenant de l'ombre "A" et de la fissure "B" - (b) Zoom de la partie "A" encadrée en rouge dans (a) - (c) Zoom de la partie "B" encadrée en rouge dans (a).*

Pour y arriver, nous nous basons sur une approche par imagettes ou par blocs dans l'image [85, 159]. Dans ce cadre, deux étapes principales, illustrées dans la figure 5.3 et détaillées dans les paragraphes suivants, sont appliquées sur chacune des imagettes. La première consiste à réduire le nombre de pixels à traiter dans l'image. La deuxième permet de sélectionner automatiquement les pixels sombres pouvant être des pixels de fissuration.

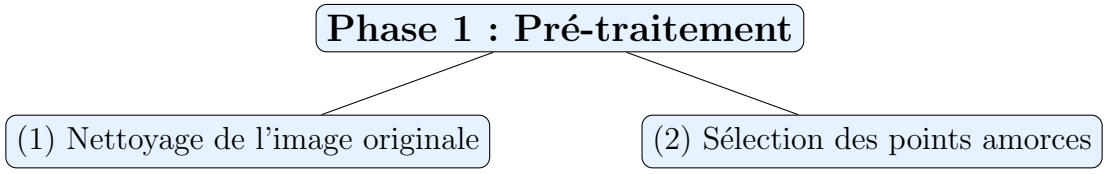



FIGURE 5.3 – *Étapes principales de la phase du pré-traitement.*

Par souci de simplicité, nous n'affichons dans ce chapitre que le résultat final sur l'image entière de départ après avoir rassemblé toutes les imagettes pré-traitées.

#### 5.3.1.1 Nettoyage de l'image originale

L'analyse des images panoramiques de chaussée (images $VIAPIX^{®}$) pour la segmentation de fissures soulève plusieurs difficultés spécifiques. En effet, l'image panoramique subit l'influence de facteurs extérieurs non maîtrisables, tels que l'éclairement, l'humidité de surface ou encore la nature du revêtement de la chaussée. Afin de pallier ces problèmes, des méthodes de "nettoyage" peuvent être appliquées sur l'image.

Les difficultés mentionnées précédemment nous ont amené à appliquer des techniques de **normalisation** et de **détection de contours**, détaillées respectivement dans les sections 5.3.1.1.a et 5.3.1.1.b. Ceci permet de réduire l'information à traiter pour faciliter la tâche de segmentation détaillée dans la phase 2 de l'algorithme.

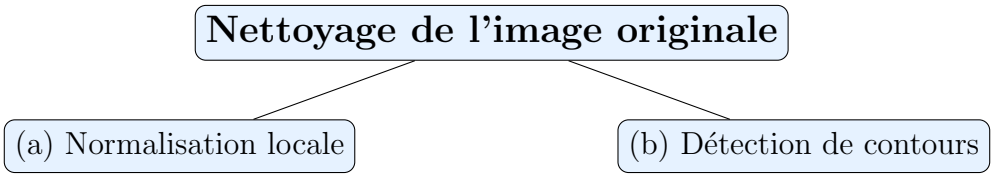



FIGURE 5.4 – *Étapes pour nettoyer l'image originale – (a) et (b) représentent les deux étapes réalisées pour garder l'information pertinente dans l'image à traiter.*

### a. Normalisation locale

La normalisation de l'image consiste à projeter un score (une image en niveaux de gris dans notre cas), sur un espace de valeurs cibles, en changeant par exemple sa moyenne et son écart-type. Elle est généralement indispensable dans des pré-traitements basés sur un seuillage de l'image et pour des images d'éclairage inhomogène [75, 87].

Cette étape est appliquée pour corriger la non-uniformité des niveaux de gris moyens causée par les variations de la texture dans différentes zones de l'image [76, 160, 161]. Elle permet de renforcer le contraste d'une image en appliquant une transformation sur chaque pixel. Parmi les techniques qui ont été proposées dans la littérature, nous citons [162, 163]. La technique *Z-score* [163] est une méthode conventionnelle qui normalise une valeur $p(i,j)$ d'un ensemble de données $p$ par :

$$p_{norm}(i,j) = \frac{(p(i,j) - \mu)}{\sigma} \tag{5.1}$$

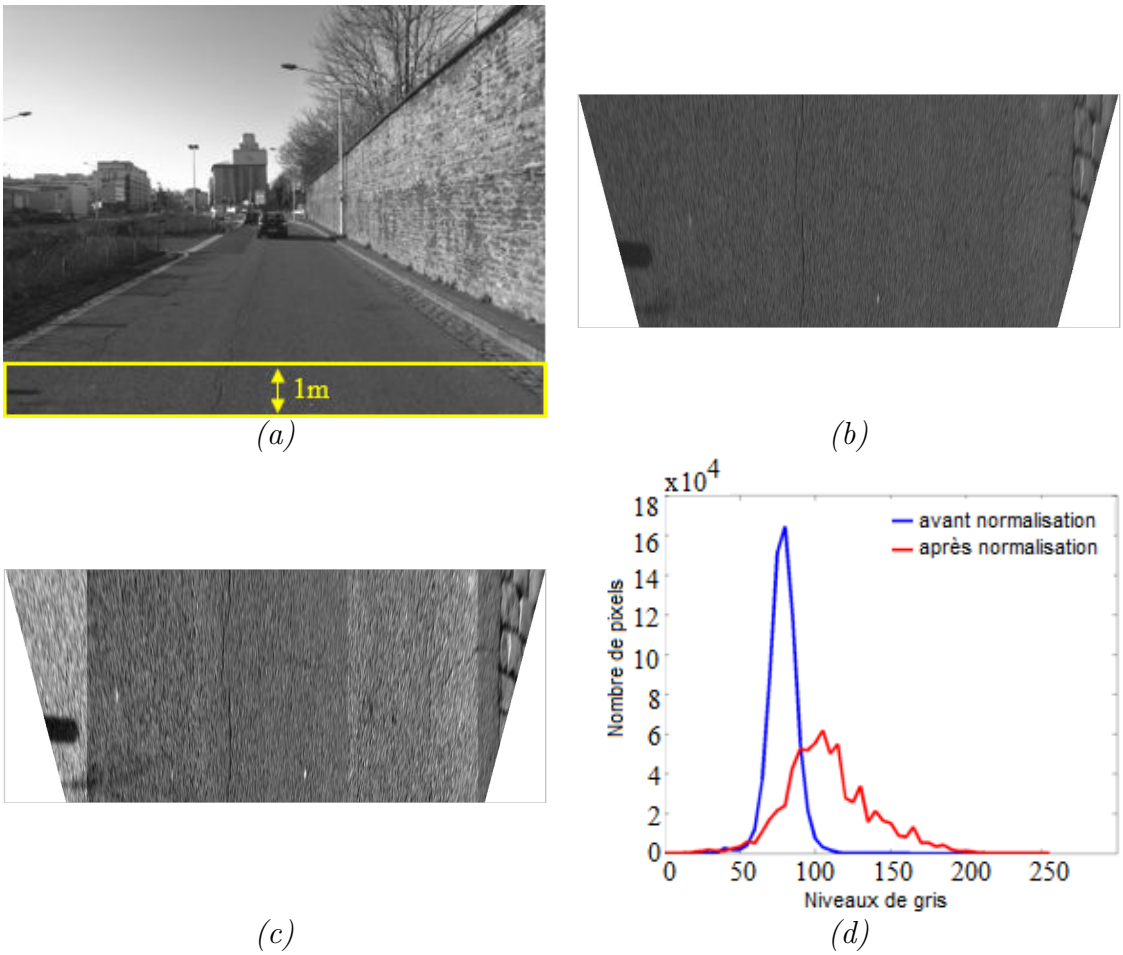
avec $p(i,j)$ est la valeur en niveau de gris d'un pixel, $\mu$ et $\sigma$ représentent respectivement la moyenne et l'écart-type de l'image à normaliser. Les inconvénients majeurs de cette méthode sont :

- Elle ne permet pas d'obtenir des valeurs dans l'intervalle $[0, 1]$.
- Elle change la distribution des données.

Pour pallier ces inconvénients, la technique *Min-Max* [163] peut être utilisée. Elle consiste à normaliser les données dans l'intervalle $[0, 1]$ tout en conservant la distribution originale. La normalisation de la valeur $p(i,j)$ est donnée par :

$$p_{norm}(i,j) = \frac{(p(i,j) - Min_{image})}{(Max_{image} - Min_{image})} \tag{5.2}$$

avec $Min$ et $Max$ représentent respectivement la valeur minimale et maximale du niveau de gris de l'image à normaliser. L'application de cette technique est illustrée à la figure 5.5. Nous constatons que la dynamique de l'image est davantage étalée après la normalisation.




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

FIGURE 5.5 – *Étape de la normalisation d'une image en niveaux de gris – (a) Image panoramique VIAPIX® en niveaux de gris - (b) Transformée perspective inverse (IPM) de l'image (a) appliquée sur une bande de chaussée d'un mètre devant la voiture - (c) Résultat de la normalisation de l'image IPM (b) - (d) Comparaison des histogrammes de chacune des deux images de (b) et (c).*

La figure 5.5.d montre que l'étape de normalisation améliore beaucoup l'image originale. L'image normalisée utilise toute la dynamique des niveaux de gris (entre 0 et 255), et augmente ainsi le contraste entre les fissures et la texture d'image.

Cependant, d'autres facteurs viennent influencer et perturber la détection, à savoir l'ombre et la réflexion de la lumière comme illustré à la figure 5.6.

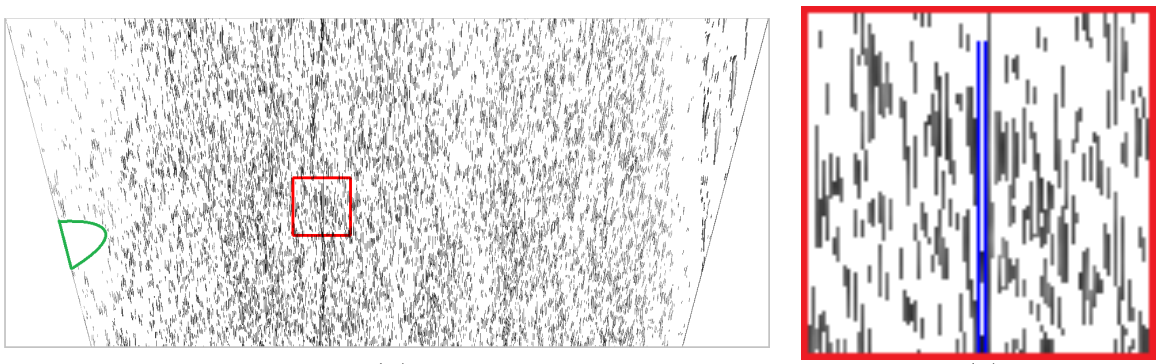
FIGURE *5.6 – Illustration de l'effet perturbateur d'une ombre portée (arbre) pour la détection d'une fissure longitudinale (partie entourée en rouge) dans une image* $VIAPIX^{®}$.

D'après la figure 5.6, les fissures que nous cherchons à détecter et les zones ombragées correspondent aux régions sombres dans l'image. Par conséquent, afin de distinguer les pixels de fissuration des pixels ombragés, nous proposons d'ajouter une deuxième étape consistant à extraire les contours de l'image.

**b. Détection de contours**

En général, la détection de contours permet de réduire d'une manière significative la quantité de données dans l'image tout en conservant les informations pertinentes. Son objectif principal est de délimiter les zones qui correspondent à un changement brutal de l'intensité lumineuse dans l'image à traiter. Il se produit généralement entre deux régions relativement homogènes en niveaux de gris, mais de valeurs différentes. Dans notre cas, les deux régions ayant approximativement les mêmes caractéristiques photométriques (faible luminosité) correspondent respectivement à la zone de dégradations contenant les fissures d'une part et à la zone sombre définissant l'ombre d'une autre part.

Pour délimiter les contours de zones, nous appliquons l'opérateur de *"Sobel"*, qui est apprécié dans la littérature pour sa rapidité et sa fiabilité [28]. La figure 5.7 montre le résultat obtenu lors de l'application de cette méthode sur l'image en niveaux de gris de la figure 5.5.c.



(a) (b)

FIGURE 5.7 – *Étape de la détection de contours – (a) Résultat de la détection de contours sur l'image perspective inverse normalisée illustrée dans la figure 5.5.c- (b) Zoom de la partie encadrée en rouge dans (a).*

En comparaison de l'image normalisée illustrée dans la figure 5.5.c, cette étape permet de supprimer la zone ombragée située à l'extrémité gauche de la chaussée (zone entourée en vert dans la figure 5.7.a). De plus, nous avons réduit le nombre de pixels à traiter, c'est-à-dire, au lieu de traiter tous les pixels de l'image illustrée dans la figure 5.5.c, nous n'avons à analyser que les pixels de contours préservés dans la figure 5.7.

Selon la figure 5.7.b, nous disposons à l'issue de cette étape de détection deux types de contours. Il y a ceux qui appartiennent à la zone de fissuration définie par la fissure longitudinale (pixels coloriés en bleu), et d'autres qui appartiennent à la zone de défaut-libre définie par la texture sur la chaussée (pixels dispersés partout dans l'image).

#### 5.3.1.2 Sélection des points amorces

Dans cette section, nous cherchons à sélectionner parmi les pixels des contours ceux qui appartiennent à une fissure sur la chaussée. En résultat, cette étape permet de ne conserver que les pixels contours les plus sombres.

La méthode proposée est illustrée à la figure 5.8. La première opération permet de séparer et distinguer les régions de défauts des régions de la chaussée dans l'image contenant les pixels de contours en niveaux de gris illustrée dans la figure 5.7. Ensuite, la deuxième opération consiste à réduire le bruit dans l'image résultante tout en préservant l'information recherchée. Et enfin, la troisième opération sert à analyser et vérifier la concentration des pixels dans les régions sombres de l'image tout en se basant sur l'hypothèse qu'un pixel de fissuration correspond au pixel le plus sombre appartenant à une zone bien chargée.

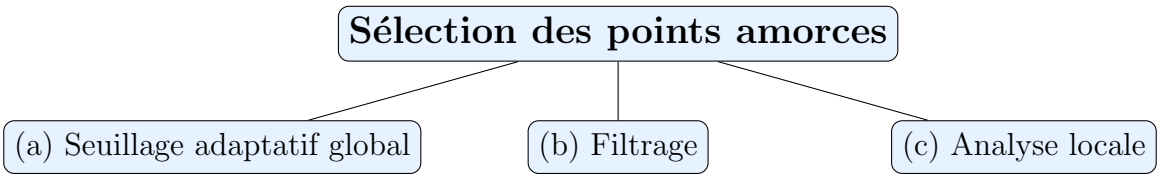
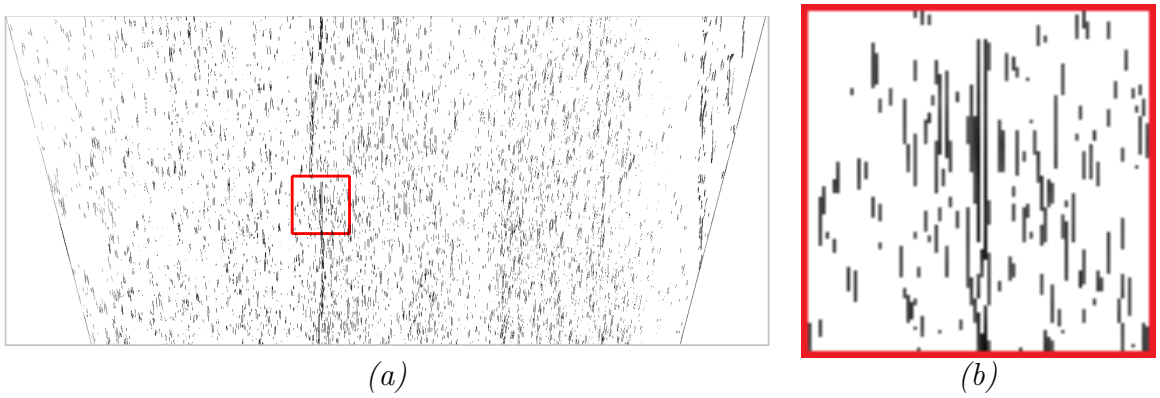



FIGURE 5.8 – *Étapes pour sélectionner automatiquement les points amorces – (a), (b) et (c) représentent les trois étapes réalisées pour sélectionner automatiquement les points sombres dans l'image.*

### a. Seuillage adaptatif global

Cette opération est appliquée aux valeurs en niveaux de gris des pixels de contours sélectionnés à l'issue de l'étape précédente (voir figure 5.7). Elle a pour but de séparer les contours décrivant la vraie fissure sur la chaussée des contours appartenant à la texture.

Cette étape est basée sur la méthode de seuillage adaptatif, nommée *NSCOT* définie par l'équation 4.6 du chapitre 4. Elle utilise le calcul du maximum des minimums des valeurs de niveaux de gris des colonnes de l'image. Ce seuil permet de conserver un maximum des pixels de contours sombres appartenant soit à la fissure, soit à son entourage dans l'image. Une fois appliquée à l'image de contours figure 5.7, nous obtenons le résultat présenté dans la figure 5.9.



*(a)* *(b)*

FIGURE 5.9 – *Étape du seuillage adaptatif global – (a) Résultat obtenu sur l'image de contours illustrée dans la figure 5.7 - (b) Zoom de la partie encadrée en rouge dans (a).*

Selon la figure 5.9, cette opération permet d'améliorer la qualité de l'image résultante. Elle sert à éliminer dans une certaine mesure la quantité de pixels de contours qui sont à l'extérieur de la fissure que nous cherchons à détecter. Cependant, un nombre important de points n'appartenant pas à la fissure n'a pas été éliminé. Ainsi afin de réduire davantage ces pixels, nous proposons une opération de filtrage.

### b. Filtrage

Dans cette section, nous présentons les deux techniques du filtrage appliquées sur l'image obtenue à l'issue de l'opération du seuillage adaptatif. La première est définie par l'application d'un filtrage médian. La deuxième est appliquée sur la taille des objets à considérer comme des vraies fissures sur la chaussée.

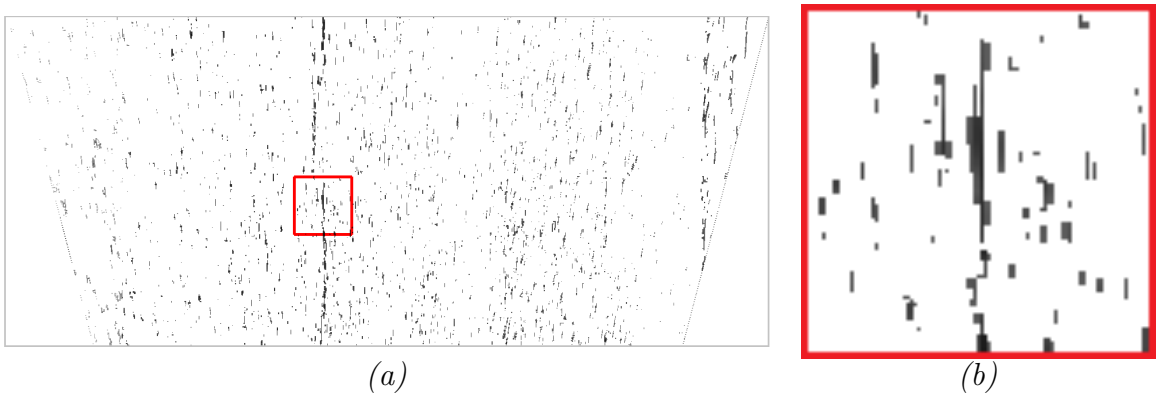
#### i. Filtrage médian

Par définition, le filtre médian est un filtre spatial non-linéaire qui calcule en chaque pixel la valeur médiane des niveaux de gris des pixels de sa fenêtre (le voisinage du pixel en question), ce qui donne le niveau de gris du pixel dans l'image filtrée.

En principe, l'objectif de ce filtre est de transformer une image $I$ en une autre image filtrée $J$ de telle sorte que pour tout pixel $I(p)$, la valeur médiane $J(p)$ correspondante est obtenue en appliquant la formule présentée à l'équation 5.3. Plus précisément, pour tout pixel $p$, soit $W(p)$ la fenêtre associée à $p$ ; nous supposons que sa taille $|W(p)|$ est impaire (c'est ainsi le cas pour une fenêtre centrée en $p$, par exemple une fenêtre $3\times3$ ou $5\times5$, ...). Étant donnée une image $I$ en niveaux de gris, le filtre médian transformera l'image $I$ en une image $J$ en niveaux de gris, telle que pour tout pixel $p$, le niveau de gris $J(p)$ est la médiane des niveaux de gris $I(q)$ des pixels $q$ dans la fenêtre $W(p)$ :

$$J(p) = med(I(q)|qdansW(p)) \tag{5.3}$$

Dans notre cas, ce filtre est appliqué afin de conserver les pixels de contours adéquats pour avoir une meilleure détection de fissures avec un minimum de fausses alarmes. Le résultat obtenu lors de l'application de cette opération de filtrage est affichée dans la figure 5.10.



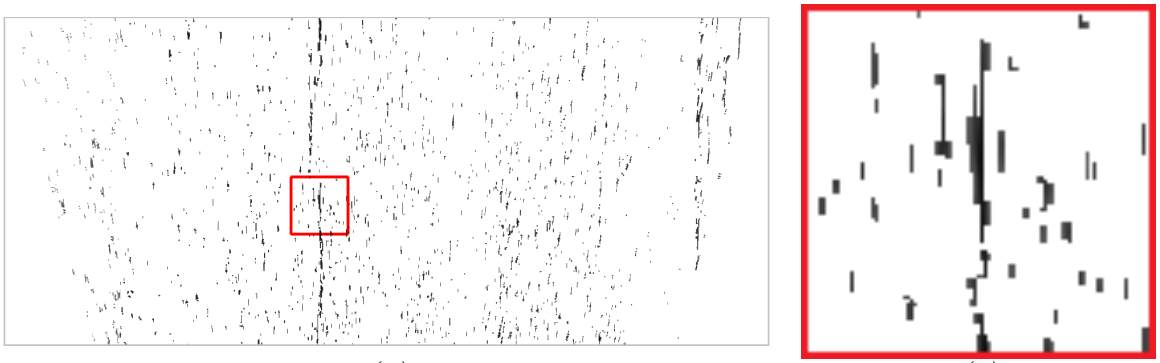
*(a)* *(b)*

FIGURE *5.10 – Application de l'opération du filtrage médian – (a) Résultat obtenu sur l'image illustrée dans la figure 5.9 - (b) Zoom de la partie encadrée en rouge dans (a).*

Cette opération de filtrage permet de bonifier la qualité de l'image résultante, comme nous pouvons le constater en comparant les résultats obtenus aux figures 5.9 et 5.10.

**ii. Filtrage selon la taille**

Une autre technique de filtrage est appliquée sur la taille des objets préservés dans l'image. En effet, elle sert à éliminer toutes les composantes connexes isolées de taille inférieure à un seuil $S = 4$ pixels. À ce stade, tout objet de taille (en pixels) inférieure à $S$ ne sera pas traité par l'algorithme. Nous notons que la suppression des objets qui correspondent à des petites fissures sur la chaussée ne pose pas de problème de discontinuité dans l'image parce que la phase 3 de la méthode ADFD mentionnée dans la figure 5.1 comprend une étape de connectivité qui consiste à relier toutes les fissures discontinues.



*(a)* *(b)*

FIGURE *5.11 – Application de l'opération du filtrage selon la taille des objets – (a) Suppression des composantes connexes de petites tailles - (b) Zoom de la partie encadrée en rouge dans (a).*

À l'issue de ces opérations de filtrage, nous obtenons l'image illustrée dans la figure 5.11. En effet, nous retenons dans cette image résultante un maximum des pixels sombres ayant une forte vraisemblance d'être des pixels de fissurations. Cette image, nommée $I_{base}$ dans la suite de ce chapitre, est ainsi utilisée principalement dans la phase du post-traitement, détaillée dans la section 5.3.3.

**c. Analyse locale**

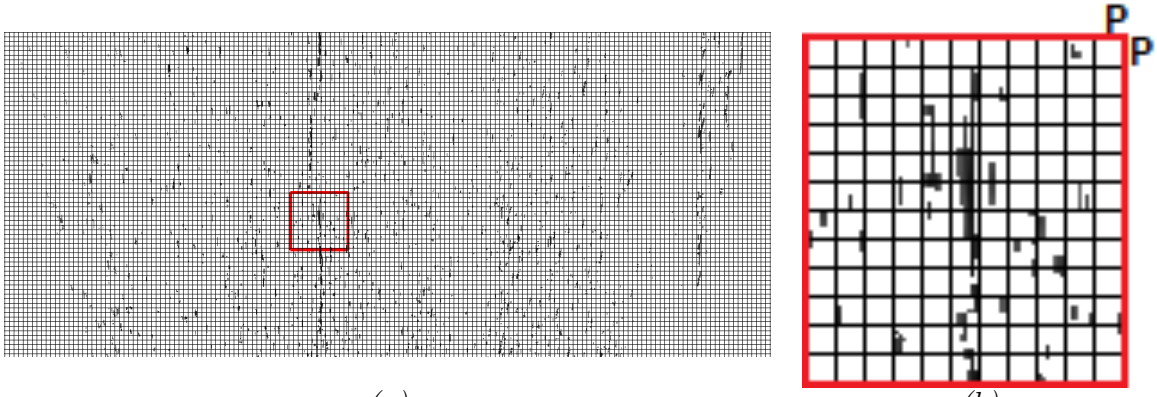
En s'appuyant sur la technique introduite dans [117, 123, 138, 152], l'analyse locale de l'image consiste à ne garder dans l'image de la chaussée que les pixels amorces (pixels très sombres par rapport au fond) appartenant à des zones considérées comme des zones dégradées. Nous proposons dans la méthode ADFD une analyse locale en 3 étapes :

- Décomposer l'image en plusieurs sous-images nommées par la suite des régions d'intérêt : *ROI*.

- Appliquer, sur chacune des régions *ROI*, l'algorithme CADP de la figure 5.15 qui consiste à vérifier si la zone à étudier correspond à une zone de fissuration ou non.
- Sélectionner un seul point amorce dans chaque région *ROI*.

### i. Décomposition en régions d'intérêt

Dans cette étape, l'image de base $I_{base}$ présentée dans la figure 5.11, est partitionnée en blocs $P \times P$, de la même manière que dans la section 3.2.1 du chapitre 3. Nous rappelons que l'ajustement de la valeur du paramètre $P$ est le résultat d'un compromis entre la qualité de la segmentation à l'issue de la phase 2 de l'algorithme et le temps de calcul nécessaire pour traiter l'image en niveaux de gris. Les tests réalisés montrent que plus la distance qui sépare les points amorces est grande, plus le temps de calcul des filaments sombres est élevé. Le résultat de cette étape de décomposition en blocs est illustré à la figure 5.12.




*(a)* *(b)*

FIGURE *5.12 – Décomposition d'une image en niveaux de gris en plusieurs blocs – (a) Décomposition de l'image de base illustrée dans la figure 5.11 en plusieurs régions d'intérêt (ROI) de taille $P \times P$ - (b) Zoom de la partie encadrée en rouge dans (a).*

L'étape suivante consiste à vérifier, selon l'algorithme CADP détaillé dans la section suivante, la présence d'une forte concentration des pixels sombres dans chacune des *ROI*. Si c'est le cas, la *ROI* courante peut correspondre à une zone de fissuration. Dans le cas contraire, nous passons à l'analyse de la *ROI* suivante.

### ii. Algorithme CADP

L'objectif de l'algorithme CADP est de sélectionner automatiquement les pixels amorces qui serviront à la segmentation du squelette de fissuration dans la phase de traitement. Il remplace la méthode *FFA* utilisée dans la méthode OMPS (cf. chapitre 4).

La première étape de cet algorithme consiste à analyser le contenu des différents *ROI* de l'image, après la décomposition en blocs $P \times P$ illustrée à la figure 5.12. Chacune des *ROI* contient les pixels sombres détectés à la fin de l'opération du filtrage détaillée aux paragraphes précédents. La stratégie appliquée dans cette méthode (figure 5.13), permet d'examiner si un bloc $P \times P$ représente une zone de fissuration sur la chaussée ou non à partir d'un décompte de pixels sombres qui s'y trouvent. La méthode CADP est illustrée dans la figure 5.14 et l'algorithme associé est détaillé à la figure 5.15.

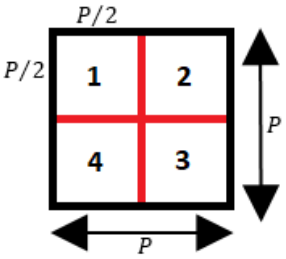



FIGURE *5.13 – Stratégie pour décomposer une région ROI de taille $P \times P$ en quatre sous-régions de taille $\frac{P}{2} \times \frac{P}{2}$.*

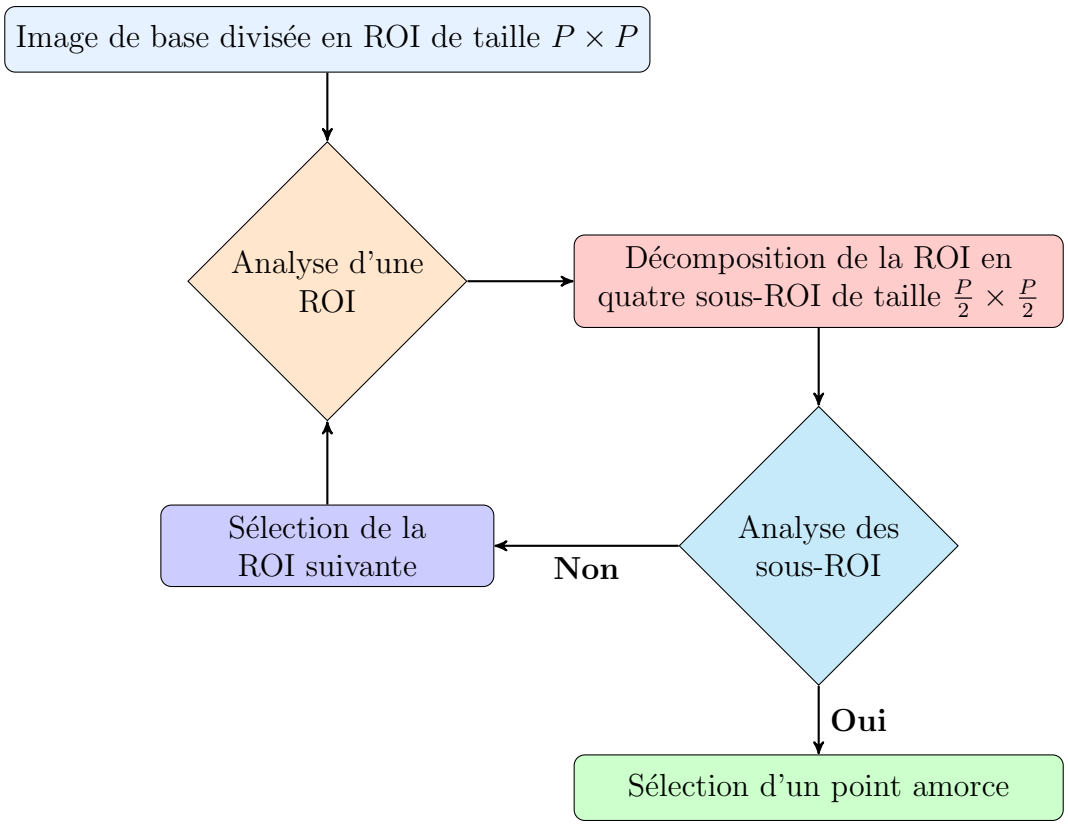



FIGURE *5.14 – Organigramme illustrant le processus automatique de sélection d'un point amorce.*

**Algorithme 1** : CADP

**for** $i = 1 \rightarrow n_v$ **do** ▷ $n_v$=***nombre des*** *ROI* ***verticalement***
  **for** $j = 1 \rightarrow n_h$ **do** ▷ $n_h$=***nombre des*** *ROI* ***horizontalement***
    Sélection d'une $ROI = image[(i-1) \times P + 1 : i \times P, (j-1) \times P + 1 : j \times P]$
    Calcul du nombre de pixels sombres dans la $ROI$ sélectionnée : $n_t$
    **if** $n_t < 2$ pixels **then**
      Pas de concentration de pixels sombres dans cette zone
      Pas de point amorce → Analyse de la $ROI$ suivante
    **else** ▷ $n_t \geq 2$ ***pixels***
      Décomposition de la $ROI$ en quatre $Sous_{ROI}$ (figure 5.13)
      Initialisation d'un compteur : $cpt \leftarrow 0$
      **for** $i = 1 \rightarrow 4$ **do**
        Sélection d'une $Sous_{ROI}$
        Calcul du nombre de pixels sombres dans la $Sous_{ROI}$ sélectionnée : $n$
        **if** $n \geq 2$ pixels **then**
          **if** $n \geq 4$ pixels **then**
            Forte densité de pixels sombres
            $ROI$ à conserver → Sélection d'un point amorce
          **else** ▷ $n = 3$ ***pixels***
            Zone douteuse → Analyse de $Sous_{ROI}$ suivante
            Incrémentation du compteur : $cpt \leftarrow cpt + 1$
          **end if**
        **else** ▷ $n < 2$ ***pixels***
          Cette $Sous_{ROI}$ est insuffisante pour prendre une décision
          Analyse de la $Sous_{ROI}$ suivante
        **end if**
      **end for**
      **if** cpt=3 **then**
        Présence d'un doute sur 3 $Sous_{ROI}$ différentes
        $ROI$ à conserver → Sélection d'un point amorce
      **else** ▷ $cpt < 3$
        Analyse de la $ROI$ suivante
      **end if**
    **end if**
  **end for**
**end for**

FIGURE *5.15 – Algorithme de vérification de la concentration des pixels sombres.*

L'algorithme permet d'analyser localement l'image binaire décomposée en blocs $P \times P$. Il comptabilise le nombre de pixels sombres à l'intérieur de chaque bloc ($ROI$). La densité de pixels sombre est considérée comme faible si ce nombre est inférieur à 2 ;

un bloc de ce type se sera pas sélectionné comme un bloc de fissuration. Dans le cas contraire, le bloc est décomposé en quatre sous-blocs, et une analyse du même type est réalisée, en tenant compte du nombre de pixels conservés à l'intérieur de chaque sous-bloc. À ce stade, trois cas peuvent être étudiés comme le montre la figure 5.16 :

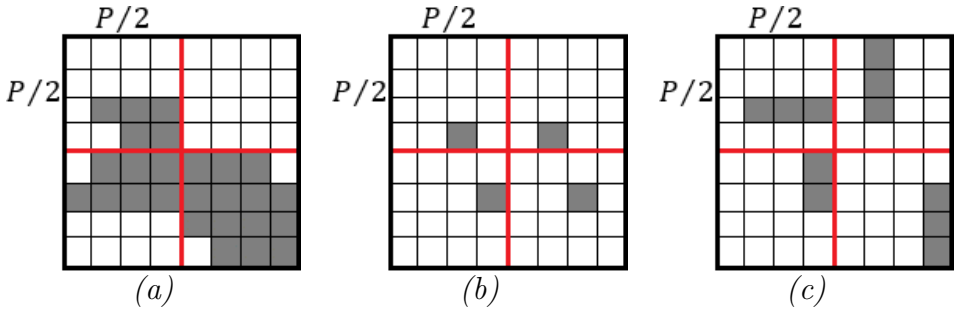



FIGURE *5.16 – Les trois cas étudiés pour sélectionner un point amorce dans un bloc $P \times P$ – (a) Exemple d'un bloc $P \times P$ à conserver - (b) Exemple d'un bloc $P \times P$ à supprimer (c) Exemple d'un bloc $P \times P$ représentant trois incertitudes dans trois sous-blocs différents.*

- Si le nombre de pixels à l'intérieur du sous-bloc est supérieur à 4, cela signifie qu'il y a une forte densité de points sombres dans cette région et ce n'est pas la peine d'analyser les autres sous-blocs (figure 5.16.a). Le bloc $P \times P$ est par conséquent considéré comme étant une zone de fissuration, il faut le conserver.

- Si le nombre de pixels à l'intérieur du sous-bloc est inférieur à 2, nous analysons le sous-bloc suivant. Si l'analyse des quatre sous-blocs ne fournit aucun potentiel de présence d'une zone sombre, c'est à dire nous avons peu ou très peu de pixels sombres dans tous les sous-blocs (figure 5.16.b), le bloc $P \times P$ est dans ce cas-là à supprimer. En effet, une zone dans laquelle il n'y a pas assez de pixels sombres à l'intérieur ne peut jamais contenir une fissure.

- Si le nombre de pixels à l'intérieur du sous-bloc est égal à 3, cela crée une certaine incertitude lors de la prise de la décision. Nous proposons, ainsi, de vérifier si au moins trois autres sous blocs contiennent plus de 3 *pixels* (figure 5.16.c). Dans ce cas, le bloc $P \times P$ est considéré comme étant une zone de fissuration, il faut donc le conserver.

Dans le cas où le bloc est conservé (présence d'une forte concentration), l'étape suivante consiste à sélectionner le point amorce correspondant. En effet, afin d'éviter de traiter tous les pixels sombres à l'intérieur de ce bloc et donc accroitre la fiabilité de l'algorithme, nous proposons de ne pas retenir que le pixel le plus sombre ayant une valeur de niveau de gris la plus petite et inférieure à un seuil défini à partir de la valeur

de la moyenne $\mu_{norm}$ et de l'écart-type $\sigma_{norm}$ de l'image normalisée illustrée dans la figure 5.5.c.

$$S_{Amorces} = \mu_{norm} + \sigma_{norm} \tag{5.4}$$

Cette étape de seuillage permet de réduire le pourcentage des points amorces calculés dans l'image pour pouvoir détecter facilement les chemins de fissuration (phase 2 de l'algorithme détaillée dans la section 5.3.2) d'une manière rapide. Un exemple de la sélection automatique des points amorces dans l'image normalisée, est illustré dans la figure 5.17.

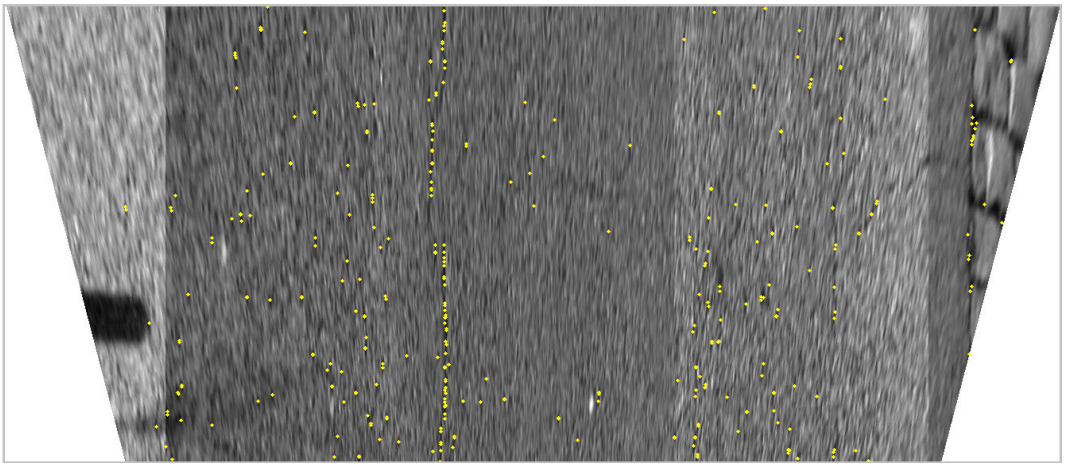


FIGURE *5.17 – Sélection automatique des points amorces.*

### 5.3.2 Traitement

Après la sélection des pixels amorces, la deuxième phase de l'algorithme ADFD consiste à segmenter les fissures sur toute l'image. La segmentation est une opération qui a pour objectif de regrouper les pixels de l'image entre eux suivant des critères prédéfinis de forme et de niveaux de gris. Pour ce faire, nous utilisons le principe de recherche de chemin minimal, qui est détaillé à la section 4.2.3 du chapitre 4, et qui a été appliqué dans la méthode OMPS. Le traitement est constitué des deux étapes de la figure 5.18.

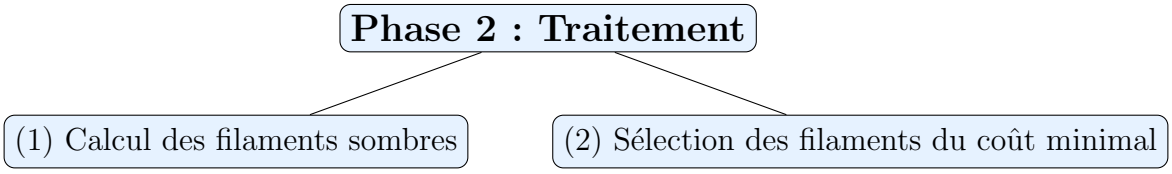



FIGURE *5.18 – Étapes principales de la phase du traitement.*

#### 5.3.2.1 Calcul des filaments sombres

C'est la tâche centrale de l'algorithme. Elle consiste à détecter le squelette de fissuration en supposant qu'une fissure sur la chaussée correspond à un filament (chemin) chaotique et sombre entre un pixel de départ et un pixel d'arrivée. Les points amorces déterminés à la figure 5.17 servent tour à tour de pixel source et destination. Ils permettent d'initialiser la recherche des chemins les plus courts dans l'image.

Pour ce faire, nous adoptons la même stratégie que celle présentée à la figure 4.15, et qui consiste à appliquer l'algorithme de Dijkstra [120] pour calculer le chemin minimal entre deux pixels amorce, sur des $ROI$ de taille $3P \times 2P$ [138, 152]. Ce processus sera répété sur toutes les fenêtres afin de calculer les filaments sombres reliant tous les points amorces entre eux. Le résultat de l'application de cette étape sur l'image contenant les points amorces est donné par la figure 5.19.

FIGURE *5.19 – Calcul des filaments sombres – Résultat obtenu lors de l'application de l'algorithme de Dijkstra sur les points amorces illustrés dans la figure 5.17.*

Chacun des filaments sombres (chemins minimaux) calculés est caractérisé par sa longueur en pixels et sa valeur de coût obtenue par l'équation 3.4. L'étape suivante consiste ainsi à nettoyer ces filaments afin de garder uniquement ceux qui représentent une fissure.

#### 5.3.2.2 Sélection des filaments de coût minimal

L'objectif de cette étape est d'aboutir à un squelette "brut" de fissuration à l'échelle de l'image. Pour ce faire, la méthode du seuillage adaptatif de la section 3.2.3, est appliquée sur les valeurs des coûts moyens (ou intensités moyennes) des filaments sombres calculés à l'étape précédente. Nous conservons les filaments d'intensité moyenne inférieure au seuil défini par l'équation 3.5. L'image résultante, nommée par la suite $I_{fil}$, est présentée à la figure 5.20.

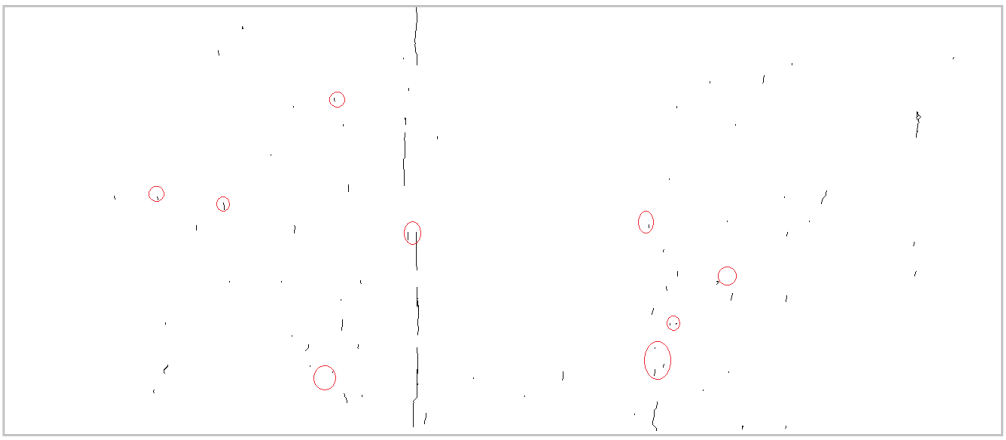


FIGURE *5.20 – Sélection des filaments les plus minces – Résultat obtenu après le seuillage sur les valeurs de coûts.*

Selon la figure 5.20, cette étape a permis d'éliminer certains filaments, qui ne représentaient pas des fissures. La comparaison avec le résultat de la figure 5.19 montre que le seuillage adaptatif appliqué permet de réduire le taux de fausses détections (parties entourées en rouge). Cependant, plusieurs fausses détections (artefacts, objet isolés) n'ont pas été éliminées. Pour cette raison, nous proposons d'appliquer dans le synoptique de la méthode ADFD (figure 5.1) une phase de post-traitement pour raffiner le résultat de segmentation du squelette de fissuration.

### 5.3.3 Post-traitement

L'objectif de cette phase est d'améliorer la qualité de la segmentation du squelette de fissuration, et d'extraire les attributs qui servent à classifier les fissures et estimer le degré de gravité des dégradations de chaussée. Elle est composée des deux étapes décrites à la figure 5.21.

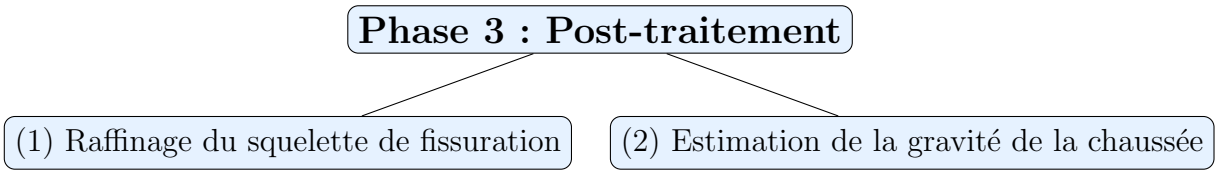



FIGURE *5.21 – Étapes principales de la phase du post-traitement.*

#### 5.3.3.1 Raffinage du squelette de fissuration

L'étape du raffinement du squelette de fissuration s'appuie sur une analyse topologique qui utilise des outils différents de ceux présentés aux chapitres 3 et 4 pour les méthodes MPS et OMPS, respectivement. Elle permet ainsi de rectifier l'image segmentée en se basant sur une technique de squelettisation [21] (souvent utilisée pour

fournir un meilleur squelette bien connecté et largeur qui ne dépasse pas un pixel). Cette étape de raffinement est composée des trois sous-étapes présentées dans la figure 5.22.

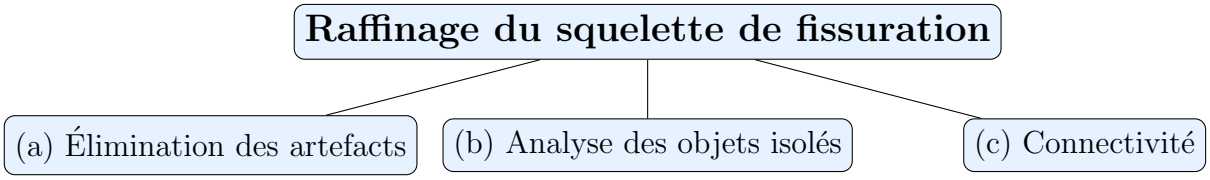



FIGURE *5.22 – Étapes pour raffiner le squelette de fissuration – (a), (b) et (c) représentent les trois étapes réalisées pour obtenir un meilleur squelette de fissuration de largeur égale à un pixel.*

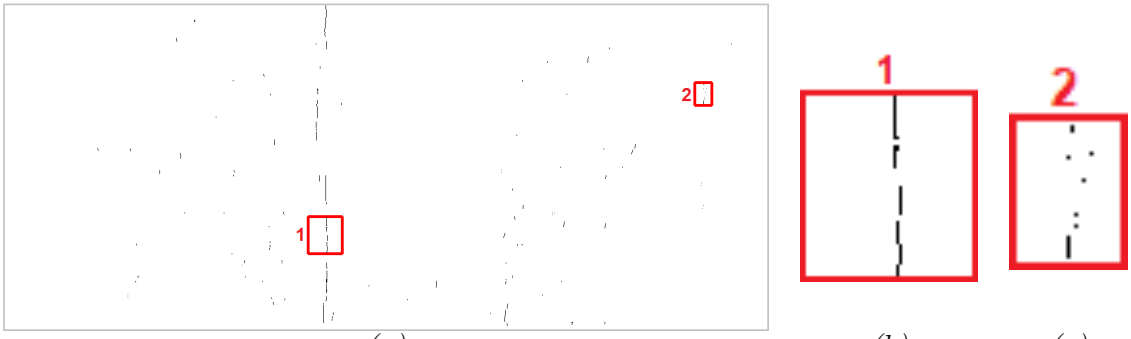
### a. Élimination des artefacts

Afin de réduire le taux de fausse alarme défini par la conservation des objets sombres dispersés partout dans l'image (objets entourés en rouge dans la figure 5.20), une méthode de purification de l'image est ainsi appliquée. Tout d'abord, elle consiste à fragmenter tous les filaments pertinents (à l'intérieur de la fissure) et non pertinents (à l'extérieur de la fissure). Ensuite, elle permet d'éliminer les petites composantes isolées de petite taille. Enfin, elle restitue la continuité entre les filaments fragmentés appartenant à la fissure.

#### i. Fragmentation des filaments

Dans cette étape, nous nous basons sur l'image de base illustrée dans la figure 5.11 qui contient un maximum des pixels sombres détectés. Parmi ces pixels, il y a ceux qui sont connectés entre eux et représentent la fissure et d'autres non connectés qui représentent les fausses alarmes. En effet, la multiplication de cette image avec le résultat obtenu lors de la sélection des filaments de coût minimal (figure 5.20) donne naissance à une nouvelle image contenant des objets fragmentés. Cette dernière, illustrée dans la figure 5.23, est obtenue selon l'équation suivante :

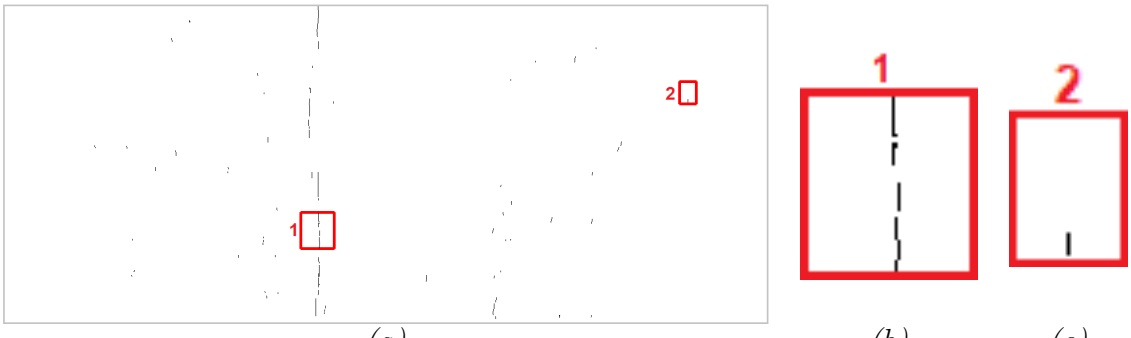
$$I_{frag} = I_{base} \times I_{fil} \tag{5.5}$$




(a) (b) (c)

FIGURE *5.23 – Fragmentation de l'image – (a) Résultat obtenu après la multiplication de l'image de base illustrée dans la figure 5.11 avec l'image contenant les filaments de coût minimal illustrée dans la figure 5.20 - (b-c) Zoom des deux parties "1" et "2" encadrées en rouge dans (a).*

À l'issue de cette étape, nous disposons d'une image fragmentée contenant des petits bouts des filaments. Parmi ces bouts, il y a ceux qui appartiennent à la vraie fissure (figure 5.23.b) et d'autres qui sont des fausses détections (figure 5.23.c). Pour y remédier, nous proposons par la suite une étape de filtrage qui consiste à retirer les objets discontinus qui ne représentent pas des parties de fissuration. En effet, cette étape permet de supprimer les objets de petite taille tout en supposant qu'un très petit objet pourrait ressembler à une texture et non pas à une dégradation sur la chaussée. Pour faire cela, nous appliquons le même filtrage selon la taille (introduit dans la section 5.3.1.2.b.ii) afin d'éliminer les chemins dont la longueur est inférieure au seuil $S$. L'image résultante, nommée par la suite $I_{net}$, est illustrée dans la figure 5.24.




(a) (b) (c)

FIGURE *5.24 – Élimination des pixels isolés – (a) Résultat obtenu après l'étape du filtrage selon la taille - (b-c) Zoom des deux parties "1" et "2" encadrées en rouge dans (a).*

La comparaison du résultat obtenu avec celui de la figure 5.23 montre d'un côté que l'étape du filtrage permet de réduire la quantité des objets. D'un autre coté, l'élimination

de ce type des composantes permet de créer dans certains cas une légère discontinuité entre les fissures fines détectées dans l'image (par exemple la discontinuité obtenue dans la figure 5.24.b). Pour éviter ce problème, une étape d'ajout de points à la fissure est ainsi appliquée.

### ii. Ajout de points à la fissure

L'objectif principal de cette étape est de résoudre le problème de la discontinuité obtenue. En effet, elle consiste à relier les pixels des filaments fragmentés tout en se basant sur une technique de vérification de l'entourage. Pour vérifier le voisinage de chaque pixel conservé, nous nous appuyons sur le voisinage de Moore [164] d'ordre 1 qui considère qu'un pixel est supposé avoir 8 pixels voisins.

En partant de chaque pixel sombre conservé dans l'image $I_{net}$ illustrée dans la figure 5.24, l'algorithme va vérifier, dans l'image $I_{fil}$ illustrée dans la figure 5.20, le voisinage jusqu'à trois pixels plus loin (sur la ligne, la colonne ou les diagonales) pour voir si le pixel est considéré sombre ou non. Si c'est le cas, il va relier ces points. Une fois la vérification des huit pixels voisins est terminée, l'algorithme va passer au pixel sombre suivant et ainsi de suite, ... Cette étape est répétée jusqu'à ce qu'il n'ait plus d'addition de nouveaux de pixels dans l'image résultante.

FIGURE *5.25 – Addition des pixels sombres voisins – (a) Résultat obtenu après avoir vérifié l'entourage de chacun des pixels sombres conservés dans la figure 5.24 - (b-c) Zoom des deux parties "1" et "2" encadrées en rouge dans (a).*

La figure 5.25 montre l'avantage apporté par les techniques décrites précédemment pour raffiner le squelette de fissuration. Nous observons à ce stade que le taux de faux positifs a diminué progressivement sans perte d'information utile. Cependant, certains objets de petite taille ont été gardés dans l'image, nous proposons ainsi une dernière étape supplémentaire de filtrage.

### b. Analyse des petits objets isolés

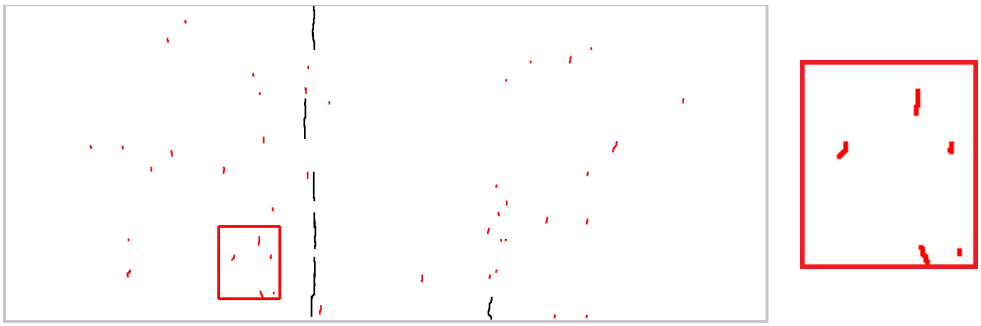
Dans cette section, nous présentons l'approche proposée pour nettoyer davantage

l'image résultante obtenue après l'élimination des artefacts. En effet, la conservation des objets inutiles dans l'image entrave dans la plupart des cas le processus de la prise de décision qui permet de vérifier la présence ou non des fissures dans une zone sur la chaussée. Nous proposons ainsi une étape supplémentaire de filtrage basée sur les opérations de la morphologie mathématique [22]. Cette méthode consiste à générer le squelette de fissuration raffiné tout en gardant une structure topologique identique à celle des fissures existantes dans l'image de la chaussée.

**i. Analyse des objets selon la longueur**

Cette section présente la méthodologie appliquée pour examiner les objets conservés dans l'image selon leur longueur en pixels. Dans un premier temps, nous sélectionnons dans l'image, de la même manière comme la méthode OMPS, tous les objets (bruit et fissures) de petite taille ayant une longueur inférieure au seuil fixe (60 pixels). Ensuite, afin d'éviter de supprimer les petites fissures, nous faisons une analyse morphologique permettant de vérifier si l'objet de petite taille correspond à une vraie fissure ou bien c'est une fausse alarme et il faut donc le supprimer.

D'après la figure 5.25.a, nous remarquons que la méthode d'ajout de pixels sombres selon le voisinage, présentée dans la section 5.3.3.1.a, n'a pas pu relier tous les objets très proches. Afin de résoudre ce problème, nous proposons d'élargir le squelette de fissuration. En effet, cette opération, définie par la dilatation morphologique (voir annexe A), consiste à étendre les objets binaires dans l'image. L'élément structurant utilisé dans cette opération est choisi selon les composantes connexes au sens de 8−connexité. Dans ce cadre, chaque pixel dans l'image non-dilatée est ainsi remplacé par huit pixels selon la forme de l'élément structurant sélectionné. En appliquant ce même principe sur l'image binaire contenant le squelette de fissuration illustré dans la figure 5.25, nous obtenons le résultat présenté dans la figure 5.26.



*(a)* *(b)*

FIGURE *5.26 – Dilatation morphologique – (a) Résultat obtenu lors de l'application de l'opération de dilatation sur l'image présentée dans la figure 5.25 (les objets dilatés de petite taille sont coloriés en rouge) - (b) Zoom de la partie encadrée en rouge dans (a).*

Pour analyser les petits objets dilatés (objets coloriés en rouge dans la figure 5.26) et ainsi éviter de les supprimer aléatoirement (en comparaison avec les méthodes MPS et OMPS par exemple), nous nous basons sur l'image de base $I_{base}$ illustrée dans la figure 5.11 du fait qu'elle contient un maximum des pixels sombres. À ce stade, une étape de détection de contours est appliquée sur l'image dilatée. Elle consiste à définir dans l'image la zone qui entoure chacun des objets dilatés à analyser. Ensuite, l'étape suivante consiste à vérifier si les pixels des objets dilatés correspondent à des vrais pixels de fissuration ou non. C'est pour cela, pour chacun des objets, nous comparons, dans l'image de base $I_{base}$, le nombre de pixels considérés comme des pixels sombres (fissures) à l'intérieur de la zone construite par les contours et le nombre de pixels appartenant à la chaussée (fond). Si dans cette zone, nous avons plus de pixels sombres ($R \geq 1$ dans l'équation 5.6), cela signifie que l'objet dilaté appartient à une zone sombre de fissuration dans l'image, il faut donc le conserver. Si non ($R < 1$ dans l'équation 5.6), il faut le supprimer.

$$R = \frac{nbr_{pixels_{sombres}}}{nbr_{pixels_{fond}}} \tag{5.6}$$

Le résultat obtenu après la vérification de l'appartenance des objets dilatés à une fissure est présenté dans la figure 5.27.

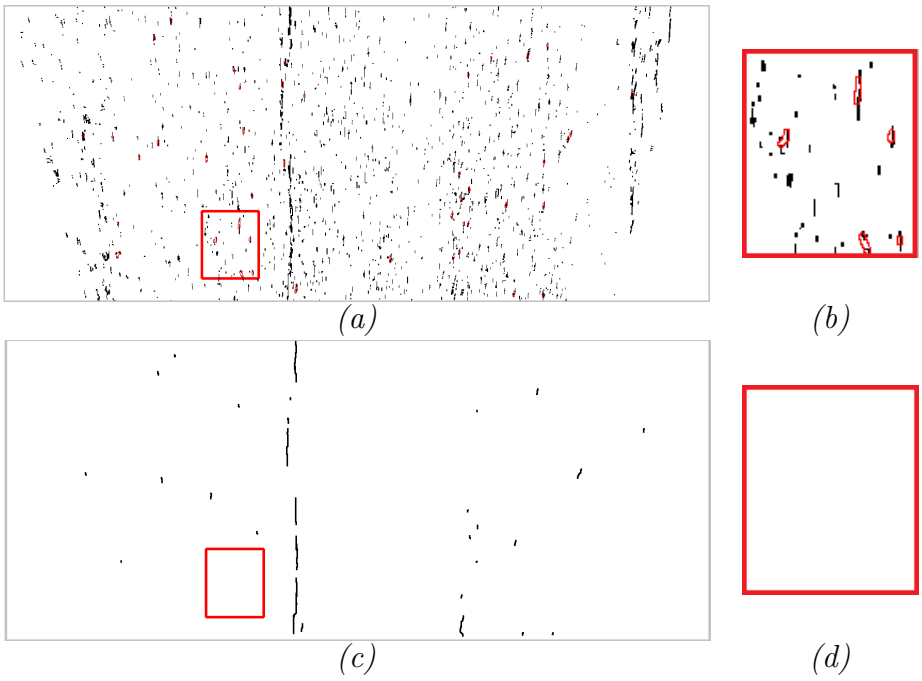



FIGURE *5.27 – Analyse des objets dilatés ayant une petite taille – (a) Les contours des petits objets dilatés sur l'image de base $I_{base}$ illustrée dans la figure 5.11- (c) Résultat obtenu après la vérification du nombre de pixels sombres dans chaque zone - (b-d) Zoom des deux parties encadrées en rouge dans (a) et (c), respectivement.*

Afin d'améliorer davantage le résultat obtenu, une opération d'érosion morphologique est appliquée sur l'image résultante dilatée. En effet, elle sert à rétrécir les objets dilatés conservés dans l'image. Le résultat de cette opération est présenté dans la figure 5.28.

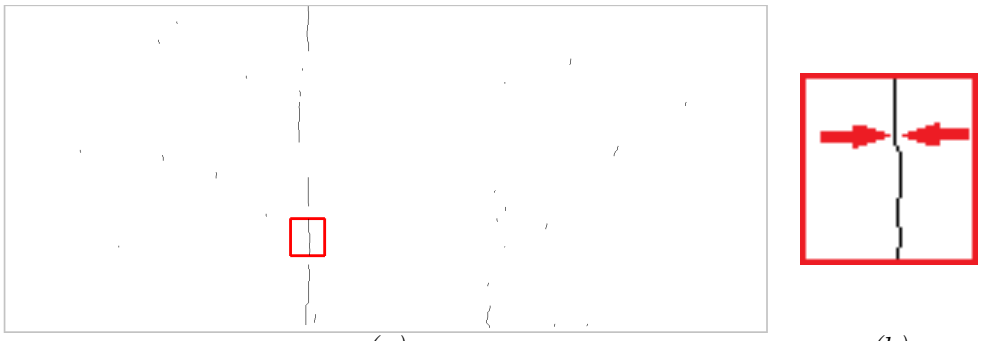
*(a)* *(b)*

FIGURE *5.28 – Érosion morphologique – (a) Résultat obtenu lors de l'application de l'opération d'érosion sur l'image présentée dans la figure 5.27.c - (b) Zoom de la partie encadrée en rouge dans (a).*

D'après le résultat de la figure 5.28, nous remarquons que l'érosion morphologique appliquée a permis d'adoucir le squelette de fissuration. L'étape suivante consiste ainsi à raffiner le résultat obtenu afin d'avoir un squelette de fissuration ayant une épaisseur qui ne dépasse pas un pixel. Un exemple des objets à raffiner est illustré dans la figure 5.28.b. Pour y arriver, une étape de squelettisation [21] est ainsi appliquée.

### ii. Squelettisation

L'objectif principal de cette étape est d'améliorer la qualité du résultat obtenu à l'issue de l'étape de l'érosion. Autrement dit, nous cherchons à raffiner au maximum le squelette de fissuration obtenu pour avoir un squelette de largeur qui ne dépasse par un pixel. Pour ce faire, nous proposons d'appliquer la méthode de squelettisation. En effet, elle consiste à supprimer au fur et à mesure certains pixels pour ne garder que ceux qui sont situés tout au long du vrai squelette en préservant la topologie globale de l'image, les relations de connexité ainsi que la géométrie de l'objet à raffiner.



(a) (b)

FIGURE *5.29 – Squelettisation morphologique – (a) Résultat obtenu lors de l'application de l'opération du squelettisation sur l'image présentée dans la figure 5.28 - (b) Zoom de la partie encadrée en rouge dans (a).*

Selon la figure 5.29, cette étape a permis de raffiner au maximum le squelette de fissuration. En effet, nous disposons d'une image nette dont les objets préservés ont une largeur égale **un pixel** (en comparaison avec le résultat illustré dans la figure 5.28). Néanmoins, par une simple comparaison visuelle, nous remarquons, dans certains cas, la présence de quelques parties non-détectées entre les filaments sombres. Pour éviter cette non détection, nous proposons de réaliser à ce stade une étape de connectivité.

### c. Connectivité

Cette étape est proposée pour résoudre le problème de la discontinuité entre les fissures. Elle permet de relier et connecter toutes les parties de fissuration qui n'ont pas été détectées par les étapes précédentes. Pour ce faire, les trois opérations suivantes doivent être mises en œuvre : **(i) Sélection des points extrémités** - **(ii) Sélection des points de départ et d'arrivé** et **(iii) Liaison inter-fissures**.

#### i. Sélection des points extrémités

L'objectif principal de cette opération est de sélectionner les points extrémités de chaque objet conservé dans le squelette de fissuration : *début* et *fin*. En effet, pour qu'un point soit considéré comme un point d'extrémité, il faut qu'il soit entouré au plus par trois autres pixels connectés entre eux. Cette sélection nous permet ainsi de ne traiter par la suite que les points extrémités présentés dans la figure 5.30.

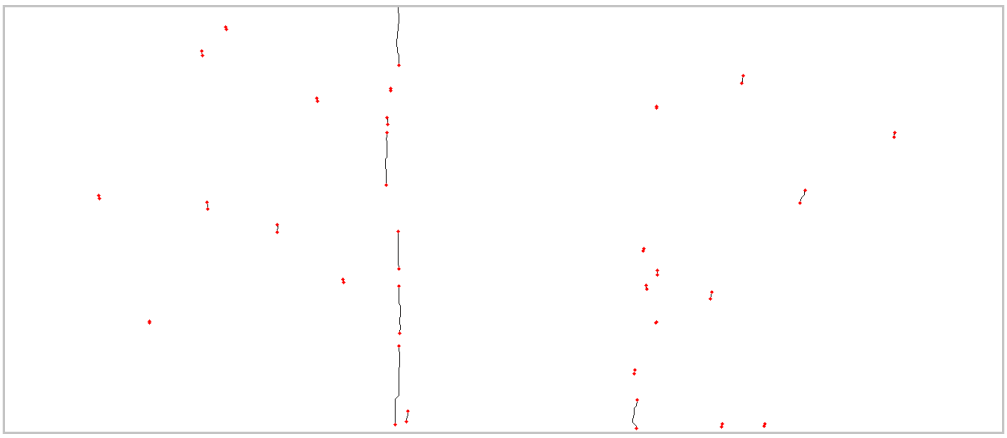


FIGURE 5.30 – *Sélection des points extrémités.*

### ii. Sélection des points de départ et d'arrivée

En général, chacun des filaments minces détectés sur la chaussée possède un point de départ et un point d'arrivée. Pourtant, les différents tests réalisés montrent qu'un seul objet sur la chaussée peut avoir plus que deux points extrémités (voir par exemple l'objet illustré dans la figure 5.31 qui contient plusieurs points extrémités). C'est pour cette raison, nous présentons dans cette section la méthode proposée permettant de sélectionner seulement deux points extrémités par objet. Nous les nommons par la suite : *début* et *fin*. Dans ce cadre, deux étapes principales sont appliquées :

☐ La première étape consiste à tracer la droite $(d)$ de Mayer [127] décrite dans la section 2.2.2.1 du chapitre 2. Voir la figure 5.31.

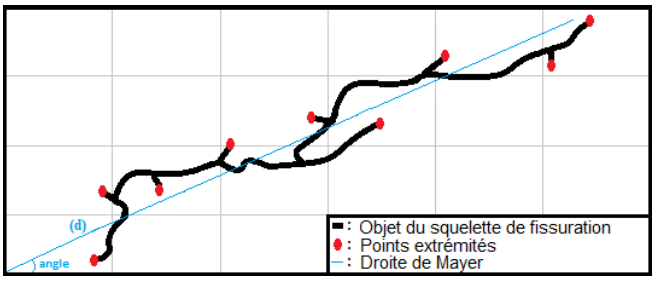



FIGURE 5.31 – *Ajustement entre les points extrémités d'un objet ramifié.*

☐ La deuxième étape permet d'estimer l'orientation de chaque objet définie par l'angle entre l'axe horizontal et la droite $(d)$. À ce stade, trois catégories peuvent être discutées :

- L'objet est considéré comme un objet vertical si son angle s'accorde avec la condition présentée dans l'équation 5.7. Un exemple de ce type d'objet est illustré dans la figure 5.32.a.

$$a_1 \leq angle \leq a_2 \tag{5.7}$$

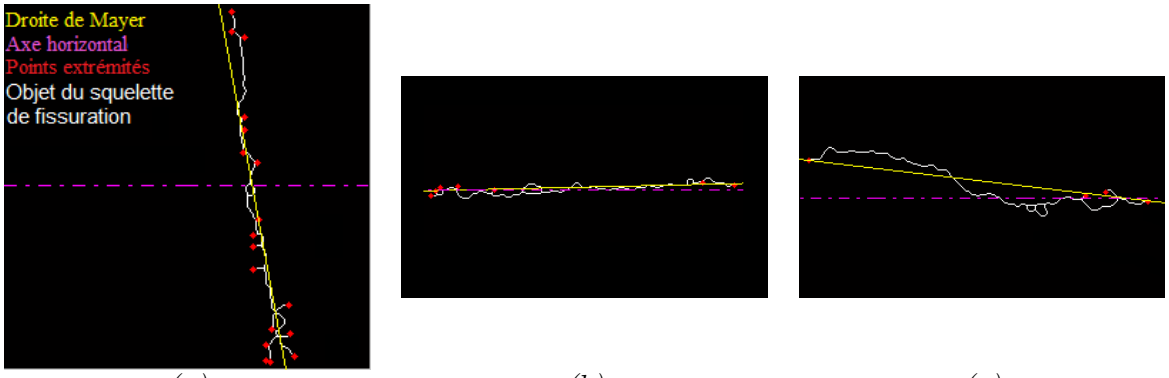
avec $a_1$ et $a_2$ représentent les valeurs en degrés ajustées d'une manière à identifier les objets verticaux dans l'image.

- L'objet est ainsi considéré horizontal dans le cas où son angle s'accorde avec la condition présentée dans l'équation 5.8. Un exemple de ce type d'objet est illustré dans la figure 5.32.b.

$$a_3 \leq angle \leq a_4 \tag{5.8}$$

avec $a_3$ et $a_4$ représentent les valeurs en degrés ajustées d'une manière à identifier les objets horizontaux dans l'image.

- Dans les autres cas, l'objet est considéré oblique. Un exemple de ce type d'objet est présenté dans la figure 5.32.c.



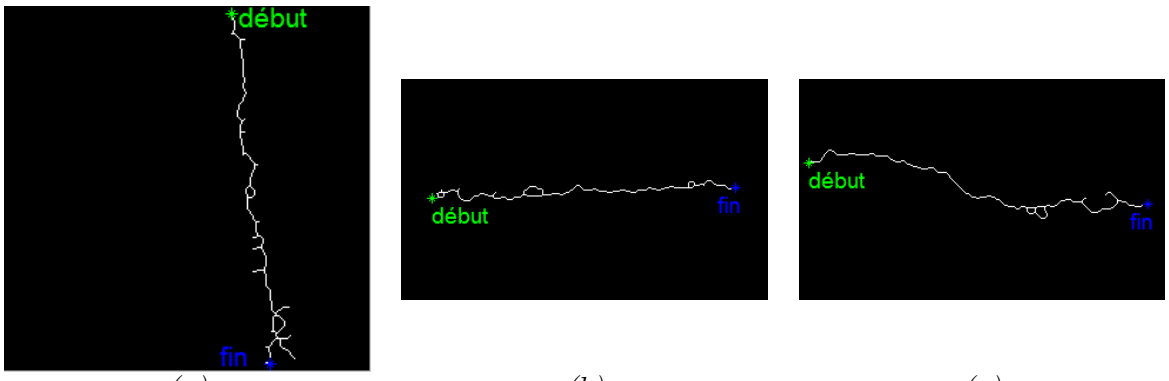

*(a)* *(b)* *(c)*

FIGURE *5.32 – Estimation de l'orientation d'un objet dans un squelette de fissuration – (a) Exemple d'un objet vertical - (b) horizontal - (c) oblique.*

L'étape suivante consiste à sélectionner, parmi les points extrémités appartenant à l'objet, les deux points principaux définis étant le début et la fin de l'objet. En effet, la sélection de ces deux points est liée principalement à l'estimation de l'orientation. Elle est définie comme suit :

- Si l'objet est vertical alors le point *début* correspond au point extrémité d'ordonné minimal alors que le point *fin* est le point extrémité d'ordonné maximal.
- Si l'objet est horizontal ou oblique alors le point *début* correspond au point extrémité d'abscisse minimal alors que le point *fin* est le point extrémité d'abscisse maximal.

Pour la clarté, les points *début* et *fin* des trois objets de la figure 5.32 sont présentés dans la figure 5.33.




(a) (b) (c)

FIGURE 5.33 – *Sélection des points début et fin des trois objets présentés dans la figure 5.32.*

### iii. Liaison inter-fissures

Cette étape permet d'assurer la continuité entre les fissures non-connectées. En effet, elle sert à relier les objets les plus proches dans l'image en se basant sur ses points *début* et *fin*. Dans notre cas, la liaison entre deux objets distincts (petits bouts de fissures non-connectées) se réalise par un calcul d'un nouveau filament sombre reliant les deux objets ensemble. Nous notons que le nombre de nouveaux filaments à calculer dans l'image dépend du nombre des objets. Il est ainsi défini par l'équation suivante :

$$nbr_{filaments} = nbr_{objets} - 1 \tag{5.9}$$

Nous présentons ci-après un exemple illustrant les différentes étapes nécessaires pour connecter un squelette de fissuration fragmenté et formé de deux objets distincts, nommés $objet_1$ et $objet_2$ dans la figure 5.34. Dans cet exemple, chacun de ces deux objets est caractérisé par son début et sa fin nommés respectivement, $debut_1$, $fin_1$, $debut_2$, $fin_2$.

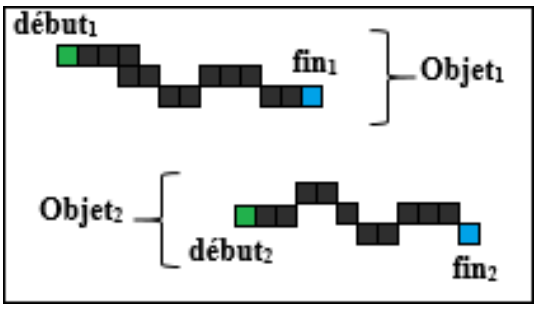



FIGURE 5.34 – *Exemple d'un squelette de fissuration fragmenté et formé de deux objets proches.*

Nous visons à regrouper ces deux objets pour en avoir un seul qui représente le squelette de fissuration. Pour ce faire, une méthode basée sur le calcul de la distance euclidienne est appliquée. Elle permet d'identifier le couple de pixels les plus proches appartenant à ces deux objets dans l'image. En mathématique, la distance euclidienne entre deux points différents $p$ et $p'$ de coordonnées $(x, y)$ et $(x', y')$ respectivement, représente la longueur séparant ces deux points. Elle est définie par l'équation suivante :

$$||p - p'|| = \sqrt{(x - x')^2 + (y - y')^2} \tag{5.10}$$

À ce stade, quatre distances $d_{min1}$, $d_{min2}$, $d_{min3}$ et $d_{min4}$ sont à calculer entre chaque paire des objets. Ces distances sont détaillées ci-après et illustrées dans la figure 5.35.

- La distance $d_{min1}$ (figure 5.35.$a_2$), représente la distance minimale sélectionnée parmi toutes les distances calculées entre le $debut_1$ de l'$objet_1$ et tous les pixels de l'$objet_2$ (figure 5.35.$a_1$).

- La distance $d_{min2}$ (figure 5.35.$b_2$), représente la distance minimale sélectionnée parmi toutes les distances calculées entre le $fin_1$ de l'$objet_1$ et tous les pixels de l'$objet_2$ (figure 5.35.$b_1$).

- La distance $d_{min3}$ (figure 5.35.$c_2$), représente la distance minimale sélectionnée parmi toutes les distances calculées entre le $debut_2$ de l'$objet_2$ et tous les pixels de l'$objet_1$ (figure 5.35.$c_1$).

- La distance $d_{min4}$ (figure 5.35.$d_2$), représente la distance minimale sélectionnée parmi toutes les distances calculées entre le $fin_2$ de l'$objet_2$ et tous les pixels de l'$objet_1$ (figure 5.35.$d_1$).

Nous notons que chacune des quatre distances minimales calculées, est définie comme étant un segment caractérisé par son point de départ et son point d'arrivée.

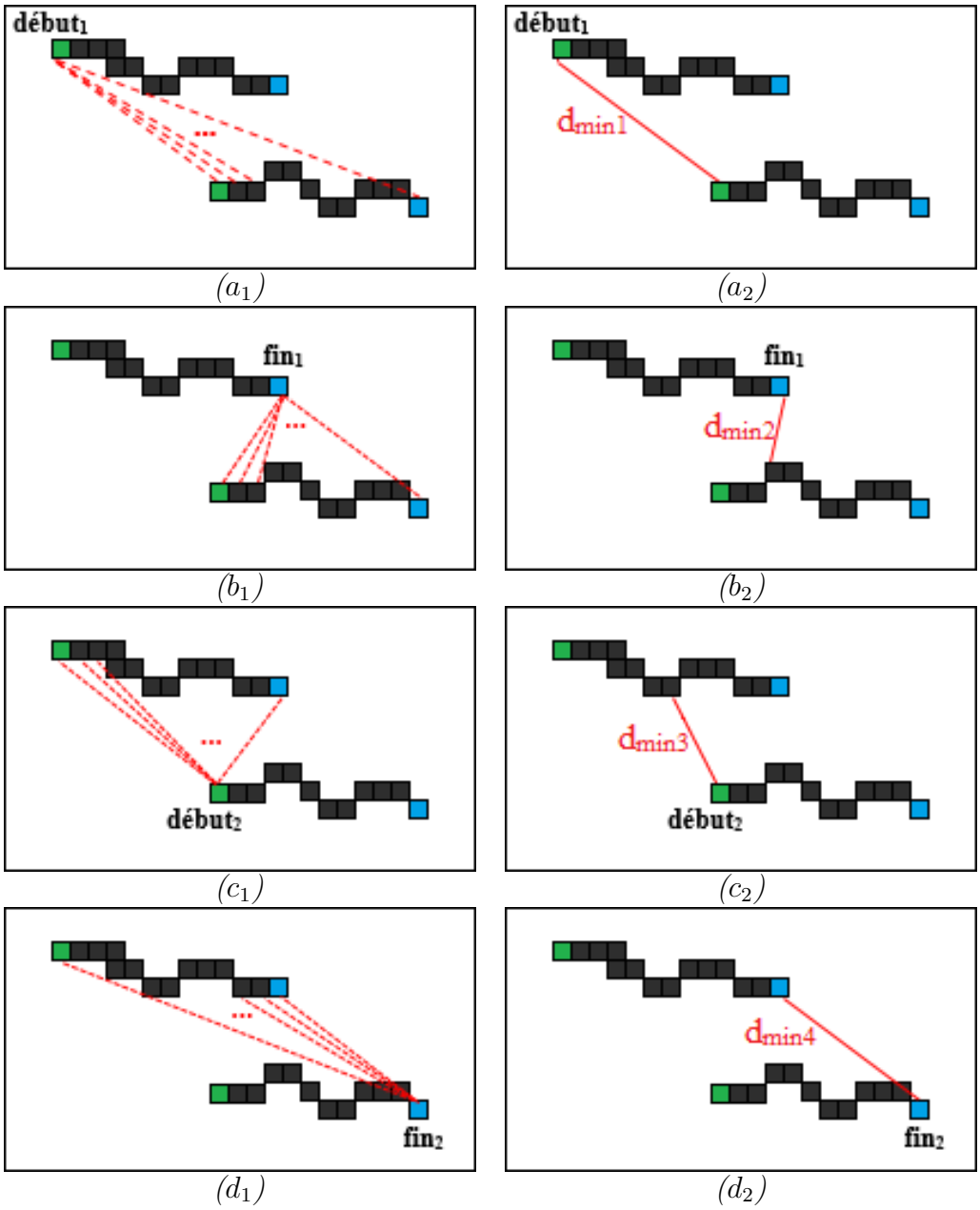



FIGURE *5.35 – Calcul des distances minimales entre deux objets proches et non-connectés – $d_{min1}$, $d_{min2}$, $d_{min3}$ et $d_{min4}$ correspondent aux quatre distances minimales calculées entre les deux objets distincts $objet_1$ et $objet_2$ illustrés dans la figure 5.34.*

Ensuite, nous cherchons à déterminer les deux points les plus proches qui séparent les deux objets $Objet_1$ et $Objet_2$ en calculant la plus courte distance $D_{min}$ parmi les quatre distances $d_{min1}$, $d_{min2}$, $d_{min3}$ et $d_{min4}$.

$$D_{min} = min(d_{min1}, d_{min2}, d_{min3}, d_{min4}) \tag{5.11}$$

Afin d'éviter de lier les objets trop éloignés et par conséquent, générer des mauvaises détections, nous appliquons un seuil $D_{max}$ à la distance minimale $D_{min}$. Dans notre cas, nous fixons $D_{max}$ à 100 pixels.

$$D_{min} \leq D_{max} \tag{5.12}$$

D'après la figure 5.35, nous remarquons que $d_{min2}$ correspondent à la distance minimale séparant $objet_1$ et $objet_2$. Par conséquent, les deux points les plus proches qui séparent ces objets ne sont à la fin que les deux extrémités de la droite illustrée dans la figure 5.35.$b_2$. Ces deux points, nommées par la suite $P_1$ (appartenant à $objet_1$) et $P_2$ (appartenant à $objet_2$), sont présentés dans la figure 5.36.

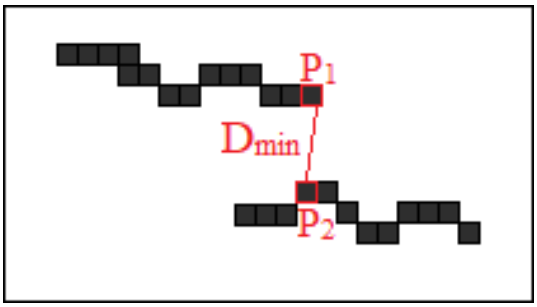



FIGURE *5.36 – Sélection des deux points les plus proches séparant deux objets différents – $P_1$ et $P_2$ correspondent aux deux points les plus proches séparant les deux objets $objet_1$ et $objet_2$ illustrés dans la figure 5.34.*

L'étape suivante permet de relier les deux points $P_1$ et $P_2$. Pour ce faire, nous nous basons sur le principe de la notion du chemin minimal introduit par l'application de l'algorithme de Dijkstra [120].

Parmi les nouveaux filaments calculés, nous ne sélectionnons que ceux qui correspondent à des vrais fissures sur la chaussée. Pour faire cela, deux conditions principales sont à valider :

- La première consiste à vérifier si la valeur de coût du filament sombre obtenu est inférieure au seuil définie par l'équation 3.5.

- La deuxième permet de vérifier si 50% des pixels de ce nouveau filament ont été détectés dans l'image de base $I_{base}$ illustrée dans la figure 5.11.

Une fois ces deux conditions vérifiées, nous obtenons le squelette de fissuration final raffiné et connecté. La figure 5.37 illustre le résultat obtenu après avoir appliqué l'étape de la connectivité sur les deux objets $objet_1$ et $objet_2$ illustrés dans la figure 5.34.

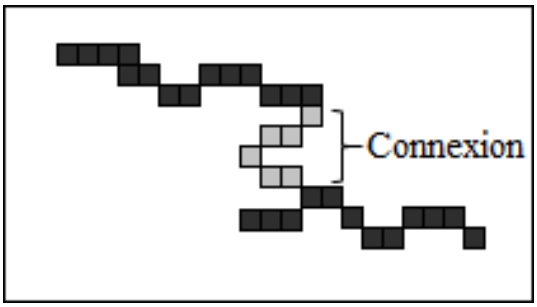



FIGURE *5.37 – Nouveau filament sombre obtenu l'étape de la connectivité – Résultat obtenu après avoir connecté les deux objets objet$_1$ et objet$_2$ illustrés dans la figure 5.34.*

### Exemple d'application

Un autre exemple illustrant l'étape de la connectivité sur un squelette de fissuration formé de deux objets, est présenté dans la figure 5.38.

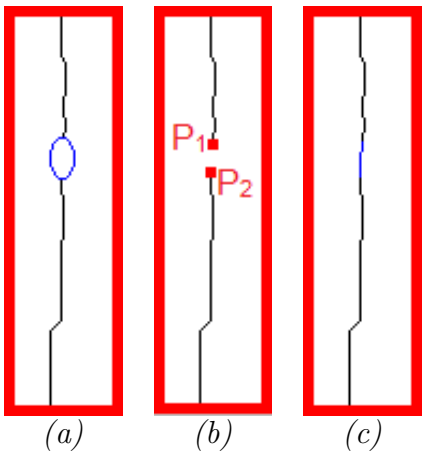



FIGURE *5.38 – Exemple d'application de l'étape de la connectivité – (a) représente la partie non détectée (partie entourée en bleu) - (b) illustre les deux points les plus proches des deux objets - (c) affiche le filament sombre obtenu après l'application de l'algorithme de Dijkstra.*

Enfin, une dernière étape de réduction de fausses alarmes est réalisée. En effet, nous ne gardons dans l'image que les objets connectés ayant une longueur $L_{objet}$ en pixels appartenant à l'intervalle suivant :

$$L_{min} \leq L_{objet} \leq L_{max} \tag{5.13}$$

Une fois toutes ces étapes sont appliquées sur l'image résultante illustrée dans la figure 5.29, nous obtenons ainsi le squelette de fissuration final présenté dans la figure 5.39.

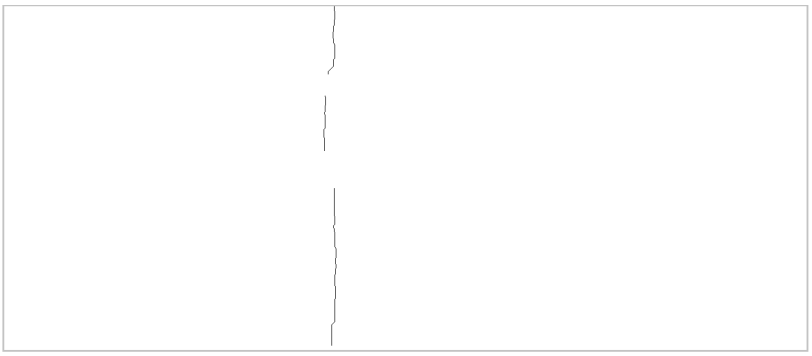


FIGURE *5.39 – Squelette de fissuration raffiné – Résultat obtenu lors de l'application de l'étape du raffinage du squelette de fissuration détaillée dans la section 5.3.3.1.*

#### 5.3.3.2 Estimation de l'épaisseur de fissure

Dans cette section, nous utilisons la même méthode appliquée dans les chapitres 3 et 4 pour calculer l'épaisseur locale de la fissure le long du squelette de fissuration. Cette étape consiste à ajouter dans l'image les pixels sombres ayant statistiquement les mêmes caractéristiques de luminosité que les pixels du squelette de fissuration, obtenu à l'étape précédente. Le résultat final de la détection des fissures est ainsi illustré dans la figure 5.40.

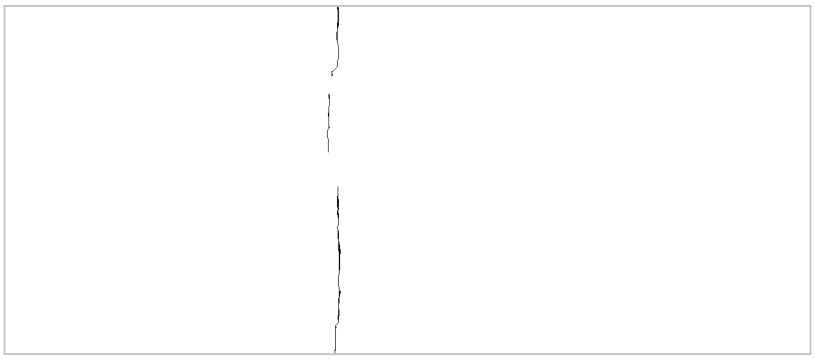


FIGURE *5.40 – Détection complète des fissures – Résultat final obtenu après l'estimation de l'épaisseur des fissures.*

## 5.4 Tests et validation

Afin d'évaluer les performances de la méthode ADFD, nous avons réalisé des tests sur des images issues des trois systèmes d'imagerie suivants : Aigle-RN [4, 5], $VIAPIX^{®}$ [125] et LCMS [124]. Nous comparons les résultats de la méthode ADFD avec ceux obtenus par la méthode OMPS [152], qui montrait les meilleures performances au chapitre 4 en détection automatique de fissures sur chaussée.

### 5.4.1 Tests sur des images Aigle-RN

Dans cette section, nous allons présenter les résultats de détection à partir des images $2D$ réelles de la base de données [149] issues du système Aigle-RN. Nous montrons ci-après quelques images de tailles différentes. En effet, les neuf images illustrées dans la figure 5.41 ont une taille de $462 \times 991$ pixels alors que les sept images de la figure 5.42 ont une taille de $462 \times 311$ pixels. Les deux tableaux 5.1 et 5.2 présentent les résultats obtenus sur ces deux types d'images, en se basant sur la valeur du coefficient de similarité (DSC) introduit dans la section 3.3.2 du chapitre 3.

La plupart des images de taille $462 \times 991$ pixels sélectionnées dans la figure 5.41 comportent des fissures avec quelques cas typiques, à savoir des fissures fines, larges, avec ou sans ramifications. Par exemple, l'image 4 présente une fissure longitudinale affichée sur quasiment toute la hauteur de l'image. Les images 2 et 5 illustrent des taches noires tout autour des fissures qui peuvent perturber la segmentation. Les images 3 et 9 comportent une fissure de largeur variable (une partie mince et l'autre épaisse). Alors que les images 1, 6, 7 et 8 contiennent des fissures transversales ramifiées affichées quasiment surtout toute la largeur de l'image.

Les résultats obtenus montrent les meilleures performances de la méthode ADFD en comparaison de la méthode OMPS. Les valeurs du tableau 5.1 montrent la plus grande efficacité de l'algorithme proposé, avec une valeur de DSC supérieure à celle de la méthode OMPS. En moyenne, la méthode ADFD atteint 76% de valeur de DSC alors que la méthode OMPS ne dépasse pas 68%.

Nous pouvons remarquer également que, dans certains cas, le temps d'exécution de la méthode ADFD est plus important que celui de la méthode OMPS. Ceci est dû au temps d'analyse supplémentaire nécessaire à l'étape de connectivité proposée.

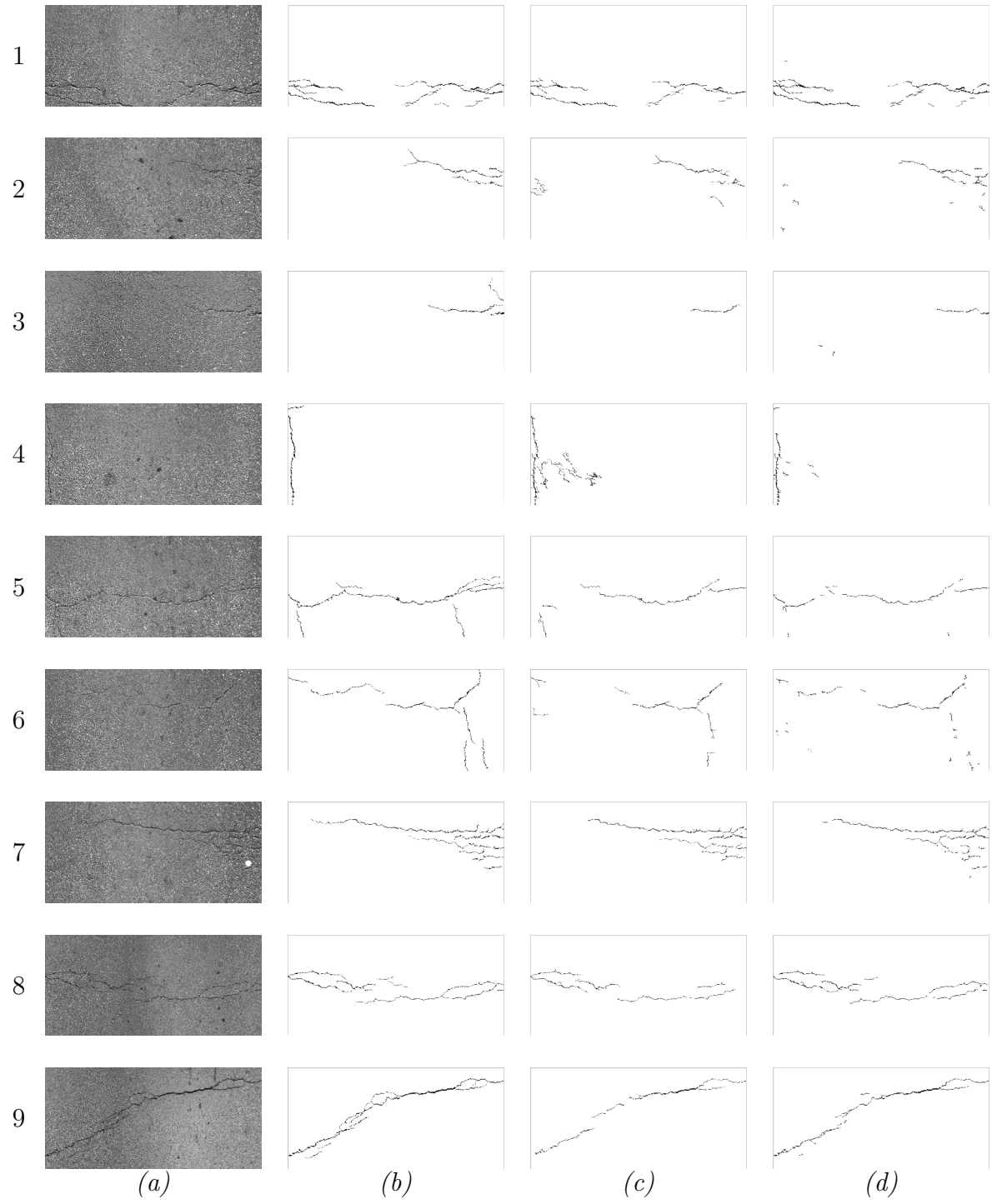



FIGURE *5.41 – Application de la méthode OMPS et la méthode ADFD sur des images Aigle-RN de taille* $462 \times 991$ *pixels – (a) Images originales - (b) Images pseudo vérité terrain - (c) Résultats de la méthode OMPS - (d) Résultats de la méthode ADFD.*

| Images | Méthode OMPS | | Méthode ADFD | |
|---|---|---|---|---|
| | **Temps (s)** | **DSC (%)** | **Temps (s)** | **DSC (%)** |
| 1 | 16 | 83 | 20 | 90 |
| 2 | 13 | 66 | 15 | 75 |
| 3 | 6 | 56 | 2 | 58 |
| 4 | 20 | 50 | 5 | 68 |
| 5 | 7 | 64 | 14 | 68 |
| 6 | 15 | 65 | 11 | 73 |
| 7 | 17 | 82 | 75 | 82 |
| 8 | 9 | 78 | 25 | 89 |
| 9 | 8 | 69 | 11 | 83 |
| **Moyenne** | 12 | 68 | 20 | **76** |

TABLE *5.1 – Résultats moyens d'évaluation des deux méthodes OMPS et ADFD sur des images Aigle-RN de taille* $462 \times 991$ *pixels.*

Nous avons également réalisé des tests sur des images de même origine mais de taille plus faibles $462 \times 311$ pixels. Les résultats de segmentation sont présentés à la figure 5.42 et le tableau 5.2 indiquent les temps de traitement et les valeurs du coefficient de similarité avec la PVT. Les résultats obtenus prouvent les meilleures performance de la méthode ADFD. En moyenne, le DSC de la méthode ADFD atteint une valeur de 77% supérieure à celle de la méthode OMPS (69%).

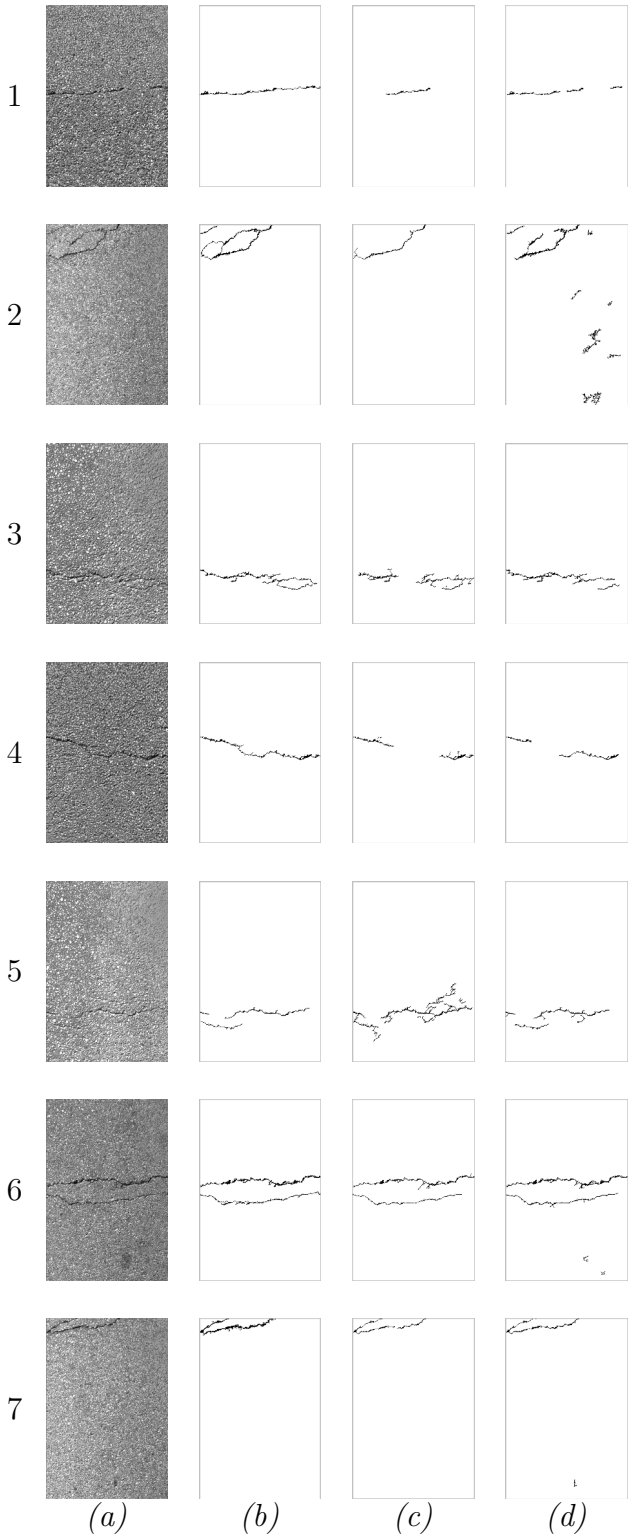



FIGURE *5.42 – Application de la méthode OMPS et la méthode ADFD sur des images Aigle-RN de taille* $462 \times 311$ *pixels – (a) Images originales - (b) Images pseudo vérité terrain - (c) Résultats de la méthode OMPS - (d) Résultats de la méthode ADFD.*

| Images | Méthode OMPS | | Méthode ADFD | |
|---|---|---|---|---|
| | Temps (s) | DSC (%) | Temps (s) | DSC (%) |
| 1 | 3 | 78 | 2 | 78 |
| 2 | 2 | 58 | 9 | 60 |
| 3 | 3 | 75 | 4 | 86 |
| 4 | 2 | 74 | 1 | 75 |
| 5 | 4 | 50 | 3 | 83 |
| 6 | 3 | 82 | 2 | 86 |
| 7 | 2 | 63 | 1 | 69 |
| **Moyenne** | 3 | 69 | 3 | **77** |

TABLE *5.2 – Résultats moyens d'évaluation des deux méthodes OMPS et ADFD sur des images Aigle-RN de taille* $462 \times 311$ *pixels.*

### 5.4.2 Tests sur des images $VIAPIX^{®}$

Afin d'étudier la capacité de la méthode ADFD à détecter les fissures dans des images en lumière naturelle, nous avons testé l'algorithme sur des images acquises par le système $VIAPIX^{®}$ dans des paysages routiers variés.

Les quatre images de la figure 5.43 correspondent à des images acquises successivement dans un environnement péri-urbain de la ville de Brest en France. Ces images sont de taille $1216 \times 1600$ pixels et ont été acquises avec un pas de 2 mètres. Elles contiennent une zone ombragée sur le bas-côté droit de la route. La zone ombragée ne perturbe pas le fonctionnement de la méthode ADFD, qui parvient à détecter la fissure longitudinale dans la partie centrale de la chaussée. Au contraire, la méthode OMPS manque la fissure et détecte des fausses alarmes liées à la partie ombragée. De plus, selon le tableau 5.3, ces fausses détections influent négativement sur le temps de calcul. En moyenne, la méthode ADFD a besoin de 41 $s$ pour traiter une image de taille $1216 \times 1600$ pixels alors que la méthode OMPS a besoin de 338 $s$ pour traiter la même image.

FIGURE 5.43 – *Application de la méthode OMPS et la méthode ADFD sur des images VIAPIX® de taille* $1216 \times 1600$ *pixels acquises tout les 2 mètres – (a) Images originales - (b) Résultats OMPS - (c) Résultats ADFD.*

| **Images** | **Méthode OMPS** | **Méthode ADFD** |
|---|---|---|
| | **Temps (s)** | **Temps (s)** |
| 1 | 414 | 23 |
| 2 | 278 | 62 |
| 3 | 99 | 50 |
| 4 | 560 | 30 |
| **Moyenne** | 338 | 41 |

TABLE 5.3 – *Résultats moyens d'évaluation des deux méthodes OMPS et ADFD sur des images VIAPIX® de taille* $1216 \times 1600$ *pixels acquises tout les 2 mètres.*

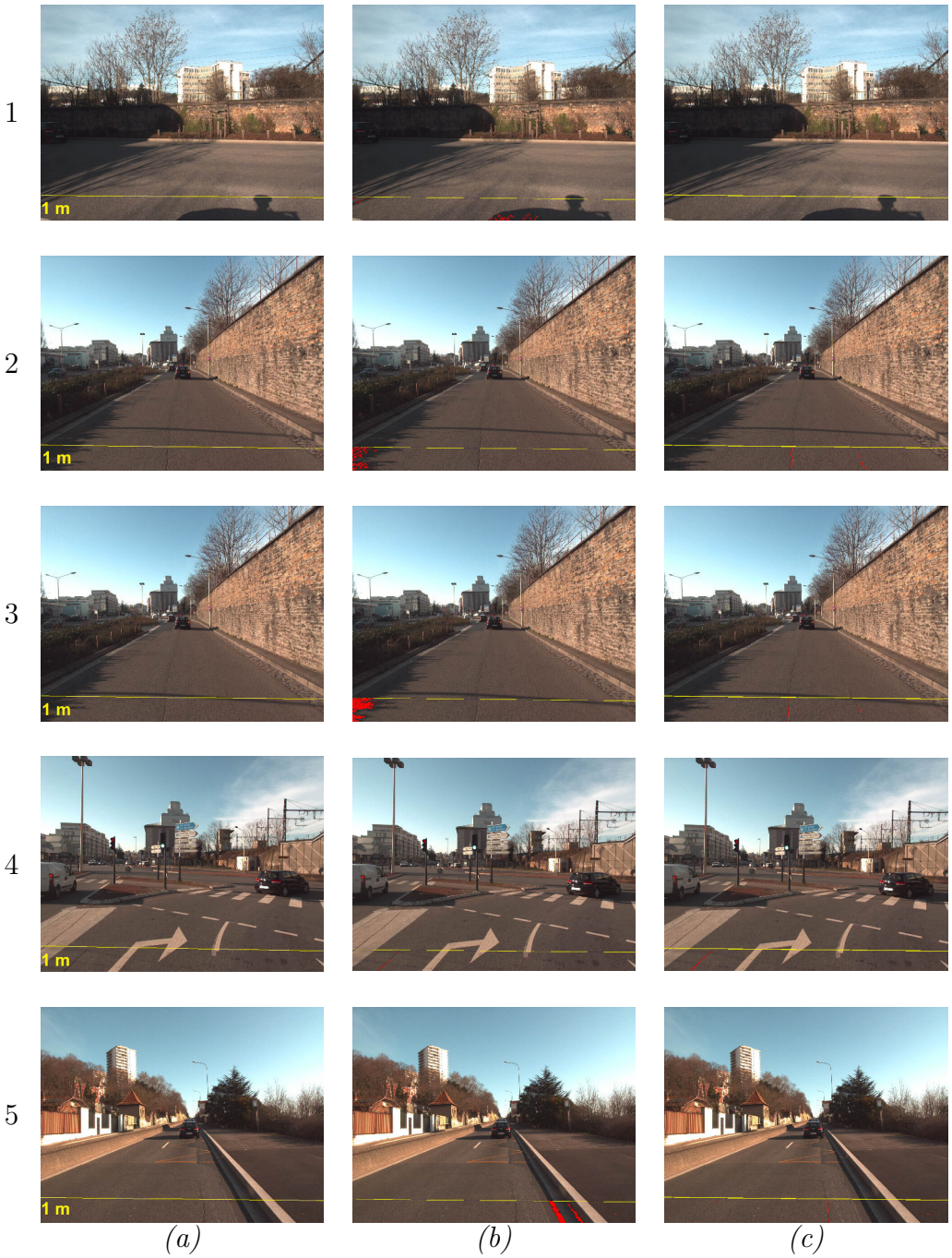



FIGURE *5.44 – Application de la méthode OMPS et la méthode ADFD sur des images VIAPIX® de taille* $1216 \times 1600$ *pixels acquises tout les 1 mètres – (a) Images originales - (b) Résultats OMPS - (c) Résultats ADFD.*

| Images | Méthode OMPS | Méthode ADFD |
|---|---|---|
| | Temps (s) | Temps (s) |
| 1 | 86 | 4 |
| 2 | 127 | 17 |
| 3 | 205 | 18 |
| 4 | 18 | 10 |
| 5 | 102 | 38 |
| **Moyenne** | 108 | 17 |

TABLE *5.4 – Temps de traitement des deux méthodes OMPS et ADFD sur des images* $VIAPIX^{®}$ *de taille* $1216 \times 1600$ *pixels acquises tous les 1 mètres.*

Les images de la figure 5.44 ont été acquises avec un pas de 1 mètre devant la voiture. Elles représentent un paysage routier varié en milieu urbain ou péri-urbain. Par exemple, l'image 1 ne contient pas de fissure mais de l'ombre devant la voiture venant la réflexion du système $VIAPIX^{®}$ sur la route. Les images 2 et 3 contiennent des fissures longitudinales avec une zone ombragée sur le bas-côté gauche de la route. L'image 4 contient une fissure avec un marquage au sol (flèche directionnelle) qui peut gêner la détection. L'image 5 contient une fissure longitudinale située à côté d'une zone sombre provenant de la pente du trottoir. Les résultats obtenus montrent que la méthode ADFD présente de meilleures performances en termes de détection et de temps de calcul. La méthode OMPS est sensible à l'ombre et les zones sombres non-fissurées dans l'image ; elle passe beaucoup de temps à rechercher des objets dans les zones sombres non-dégradées. De fait, selon le tableau 5.4, la méthode ADFD est la méthode de segmentation la plus rapide et la plus fiable.

### 5.4.3 Tests sur des images LCMS "élévation"

Dans cette section, nous présentons les résultats de la segmentation obtenus par la méthode ADFD sur des images LCMS "élévation" pleine résolution de taille $10\,000 \times 4\,118$ pixels (voir la section 1.6.1.2 pour la description du système LCMS). Les images sélectionnées dans ce travail ont été acquises en 2018 sur des routes départementales de la ville de Brest en France. L'accès aux images LCMS pleine résolution a fait l'objet d'une autorisation spécifique des deux sociétés **Diagway** et **ACTRIS**.

Les images de chaussées utilisées présentent différents niveaux de dégradations. Par exemple, les images 1 et 2 de la figure 5.45 représentent une chaussée très endommagée, contenant une mosaïque de fissures de largeur variable. L'image 1 de la figure 5.46 présente une chaussée peu endommagée, avec une fissure longitudinale d'épaisseur variable. L'image 2 contient deux fissures obliques, l'une assez grande et épaisse, une seconde, perpendiculaire à la première et relativement fine et petite.

L'application de la méthode ADFD sur ces images a fourni une détection similaire à celle obtenue par Pavemetrics.

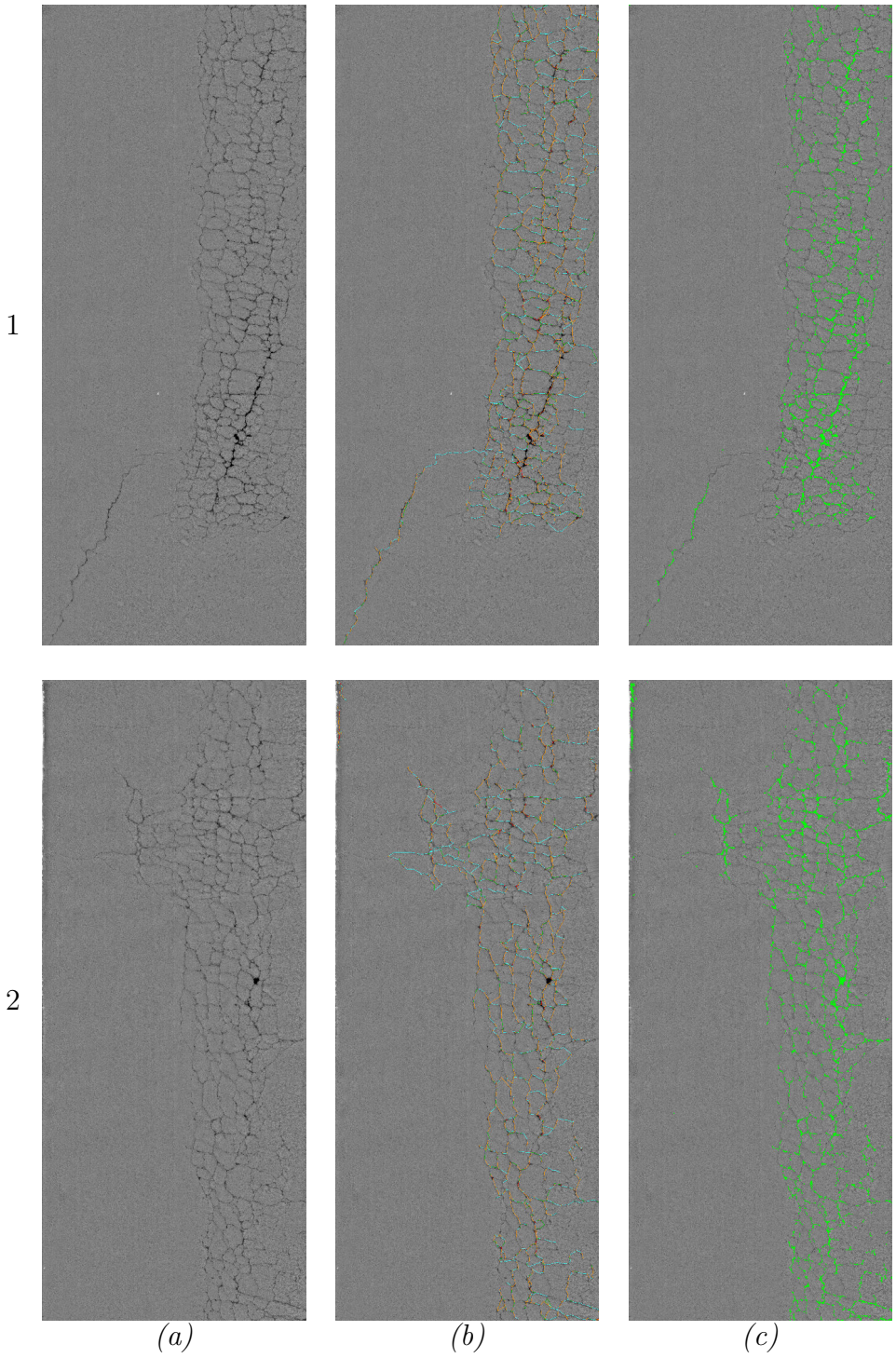



FIGURE 5.45 – *Application de la méthode ADFD sur des images LCMS "élévation" d'une chaussée endommagée incluant un mosaïquage – (a) Images originales de taille* $10\,000 \times 4\,118$ *pixels - (b) Résultats Pavemetrics - (c) Résultats ADFD.*

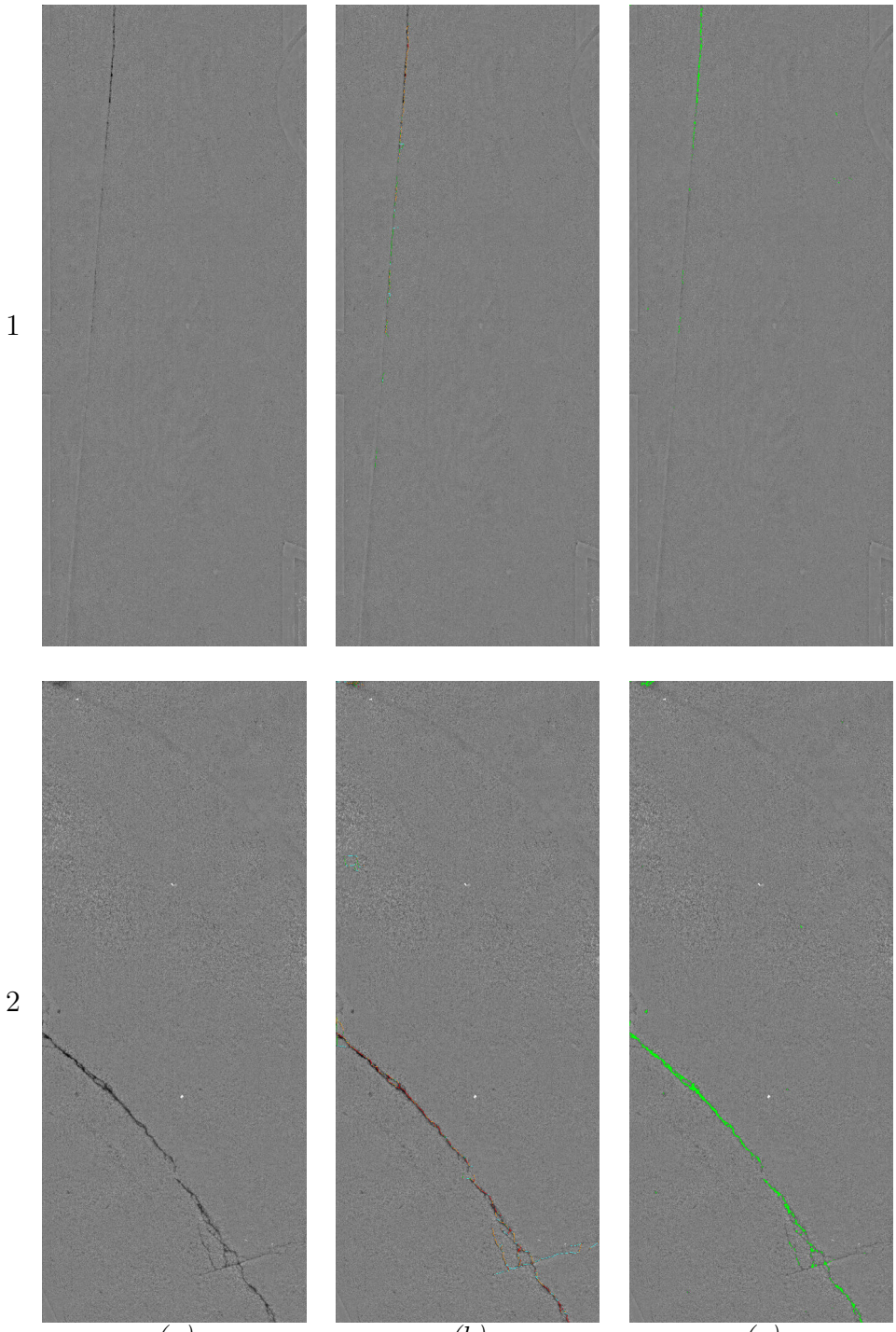



FIGURE *5.46 – Application de la méthode ADFD sur des images LCMS "élévation" d'une chaussée peu endommagée – (a) Images originales de taille* $10\,000 \times 4\,118$ *pixels - (b) Résultats Pavemetrics - (c) Résultats ADFD.*

Le temps de calcul nécessaire à la segmentation varie d'une image à une autre, selon le nombre de fissures dans l'image. En moyenne, la méthode ADFD a besoin d'environ 20 *min* pour traiter une image pleine résolution, soit un temps de calcul d'environ 30 s/MPx .

## 5.5 Conclusion

Dans ce chapitre, nous avons présenté la nouvelle méthode ADFD (Automatic Darkest Filament Detection) pour la détection automatique des fissures sur la chaussée. Elle a pour objectif de rendre le résultat de segmentation plus robuste aux conditions d'éclairage, et ce faisant, de faciliter l'usage des méthodes de segmentation au cas des images panoramiques (conditions lumière naturelle) et des images issues de nouveaux capteurs de type LCMS.

La méthode ADFD diffère des deux précédentes méthodes à base de chemin minimal i.e., MPS et OMPS, par plusieurs aspects innovants. Tout d'abord, une méthode de détection de contours est appliquée en pré-traitement pour délimiter les zones d'ombrage dans l'image. Ensuite, une analyse approfondie de tous les objets détectés dans l'image est réalisée. En particulier, elle inclut une nouvelle étape d'analyse de connectivité entre les segments de fissures, afin d'éviter la suppression des fissures isolées de petites tailles, et de réduire la fragmentation des résultats. La phase de post-traitement du squelette de fissuration est basée dans cette méthde sur un principe d'analyse topologique alternatif. Enfin, la méthode ADFD peut facilement être adaptée pour traiter des images $3D$ acquises par le système LCMS (Laser Crack Measurement System).

Par rapport à la méthode OMPS, présentée au chapitre 4, les résultats montrent les meilleures performances de la méthode ADFD sur les deux types d'images $2D$ (sous éclairage ambiant, i.e., images $VIAPIX^{®}$ et sous éclairage contrôlé, i.e., images Aigle-RN). En particulier, les différents pré-traitements proposés permettent de s'affranchir de l'effet d'ombrage dans l'image. Nous observons une réduction du taux de fausse alarme, sans augmentation significative du taux de faux négatifs. Enfin, la méthode ADFD conserve la précision de la segmentation, qui est une caractéristique des méthodes à base de chemin minimal, comme cela a déjà été illustré aux chapitres 3 et 4.

# Chapitre 6

## Conclusion et perspectives

**Sommaire**

6.1 Conclusion . . . . . **200**
6.2 Perspectives . . . . . **201**

## 6.1 Conclusion

Ce document a présenté le résultat d'un travail de thèse dont l'objectif est de contribuer à l'entretien du réseau routier. Cet objectif nécessite une phase d'auscultation, pour faire l'inventaire des défauts et établir un indice de service, qui permet de prévoir les travaux d'entretien. Les techniques de relevé des dégradations en surface de chaussées ont beaucoup évolué au cours du temps. Le relevé automatique a succédé au relevé manuel, qui n'est plus adapté aux exigences actuelles de fiabilité, de sécurité et de rapidité.

Le relevé automatique s'appuie d'une part sur des appareils d'imagerie à grand rendement, qui collectent des images de la chaussée à vitesse de trafic, et d'autre part, sur des algorithmes de traitement d'images dont l'objectif est de segmenter automatiquement les défauts de surface dans les images.

Le chapitre 1 a présenté la problématique du travail de thèse, qui consiste à détecter deux types de dégradation en surface de chaussée : l'usure de marquages horizontaux et les fissures. L'état de l'art a montré que les performances des méthodes de segmentation conventionnelles sont limitées par la forte texture des images de chaussée. En comparaison, les méthodes qui exploitent conjointement les caractéristiques photométriques et géométriques des objets à segmenter sont les plus pertinentes. Pour les fissures, la bibliographie a notamment montré l'intérêt des méthodes à base de chemins minimaux.

Le chapitre 2 a présenté la méthode que nous avons développée pour quantifier l'usure du marquage routier à partir d'une image du paysage routier. Le travail a nécessité d'aborder différentes disciplines de traitement d'images, colorimétrie, photométrie, et d'utiliser différents outils tels que la transformée en perspective inverse, la reconnaissance des formes, etc. La première étape du traitement consiste à identifier l'espace de la route à partir d'une méthode de segmentation colorimétrique. Dans cet espace, les marquages sont détectés comme des objets de forte luminosité. Le traitement se focalise principalement sur deux types de marquages spécifiques, les passages piétons (nombre de bandes, forme des bandes, etc.) et deux types de pictogrammes (couloir de bus et piste cyclable), qui sont détectés par des critères géométriques et de corrélation optique, respectivement. La géolocalisation de tous les objets détectés est réalisée par la méthode de Vincenty. Finalement, l'évaluation de l'état d'usure des marquages est réalisée par une analyse colorimétrique et photométrique des objets identifiés. Les tests qui ont été réalisés sur des images $2D$ du système $VIAPIX^{®}$, ont montré les bonnes performances de la méthode proposée sur les deux types de marquages cités.

Les chapitres 3 à 5 se focalisent sur la segmentation automatique de fissures. Le chapitre 3 a présenté le principe de l'algorithme MPS, que nous avons sélectionné au chapitre 1 à l'issue de l'état de l'art. La méthode MPS réalise une segmentation sans contraintes de forme, de direction, de longueur, ni d'épaisseur. Elle est basée sur le calcul

de chemins minimaux dans l'image à l'aide de l'algorithme de Dijkstra. Nous avons également présenté la méthodologie d'évaluation des performances de segmentation de l'algorithme (à l'échelle du pixel). MPS permet une segmentation plus précise que d'autres méthodes de la littérature (HA, FFA, etc.). Le coefficient de similarité moyen est de 70% environ sur la base de données Aigle-RN. Son principal inconvénient est son temps de calcul (sous un environnement Matlab), qui limite son application au niveau opérationnel.

Dans le chapitre 4, nous avons présenté une version optimisée de la méthode MPS classique. La méthode proposée, nommée OMPS, utilise conjointement une méthode de seuillage adaptative et une méthode de fusion de données pour réduire la quantité d'information à traiter dans l'image. Les tests réalisés sur la même base d'images qu'au chapitre 3, montrent que OMPS atteint un coefficient de similitude de 10% supérieur en moyenne environ, et de réduire le temps d'exécution d'un facteur 50 à 60 dans un environnement de programmation Matlab.

Enfin, la méthode ADFD (Automatic Darkest Filament Detection) présentée au chapitre 5, a pour objectif de rendre le résultat de segmentation plus robuste aux conditions d'éclairage, et ce faisant, d'étendre l'usage des méthodes de segmentation au cas des images panoramiques (conditions lumière naturelle) et des images issues de nouveaux capteurs de type LCMS. La méthode ADFD diffère des deux précédentes méthodes à base de chemin minimal par plusieurs aspects innovants. En particulier, les différents pré-traitements proposés (une détection de contours notamment) permettent de s'affranchir de l'effet d'ombrage dans l'image. Ensuite, une analyse approfondie de tous les objets détectés dans l'image est réalisée. En particulier, elle inclut une nouvelle étape d'analyse de connectivité entre les segments de fissures, afin d'éviter la suppression des fissures isolées de petites tailles, et de réduire la fragmentation des résultats. Enfin, la méthode ADFD permet également de traiter des images $3D$ acquises par le système LCMS. Les résultats montrent les meilleures performances de la méthode ADFD sur les deux types d'images 2D (sous éclairage ambiant, i.e., images $VIAPIX^{®}$, et sous éclairage contrôlé, i.e., images Aigle-RN). Nous observons une réduction du taux de fausse alarme, sans augmentation significative du taux de faux négatifs. Ainsi, la méthode ADFD conserve la précision de la segmentation, qui est caractéristique des méthodes à base de chemin minimal.

## 6.2 Perspectives

Nous avons développé dans cette thèse un système automatique pour le relevé automatique de dégradations sur chaussées. Les résultats obtenus sont encourageants et semblent confirmer notre choix de technologie. Cependant, de nombreux de tests restent à effectuer afin de valider une méthode pour la détection automatique des différents types de dégradations sur la chaussée. Les deux types de dégradations que nous avons traitées dans ce travail de recherche sont : la détérioration des marquages routiers et les

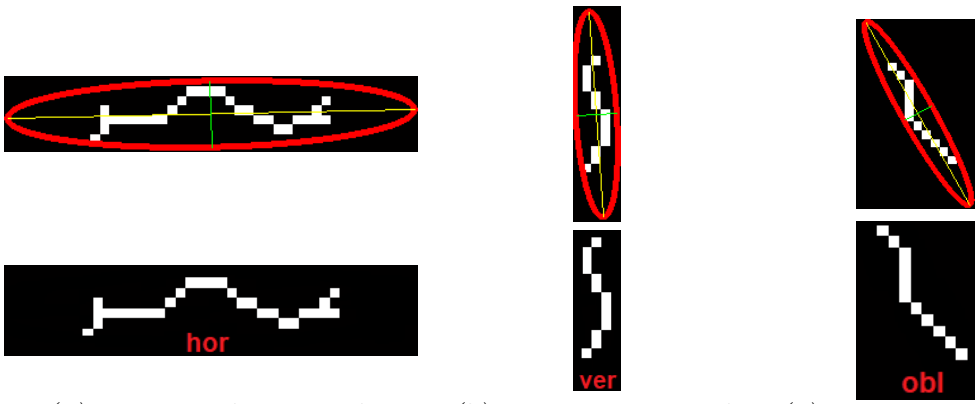
fissures en surfaces de chaussées. Dans ce cadre, des améliorations sont possibles pour les méthodes proposées.

La première partie de cette thèse décrit les méthodes proposées pour la détection de l'état d'usure des marquages au sol. En effet, nous avons traité les cas du passage piéton, du mot BUS et des pictogrammes-vélos. A ce stade, il parait intéressant d'approfondir les tests afin de généraliser notre algorithme pour introduire une méthode globale de détection, de reconnaissance, de classification et de quantification des marquages routiers avec des symboles (par exemple, les flèches directionnelles). Plus précisément, Il serait intéressant d'étendre davantage cette étude pour pouvoir interpréter tous les types et catégories du marquage au sol ; à savoir les lignes zigzag, les lignes continues et discontinues, les marquages de type cédez-le passage, les flèches directionnelles, les places handicapées, etc (voir figure 6.1).

FIGURE *6.1 – Catégories des marquages routiers à traiter dans le futur – De gauche vers la droite : ligne zigzag, lignes continues et discontinues, cédez-le passage, flèche directionnelle, place de stationnement pour les handicapés.*

D'autre part, la deuxième partie de cette thèse est consacrée à la détection des fissures sur la chaussée. Les méthodes adoptées ont montré leur capacité de détecter les fissures dans des images $2D$ et $3D$ dans des conditions d'éclairage très variées (éclairage contrôle, ambiant, etc.). Un des axes de recherche possible pour la suite de ces travaux, consiste à la classification des fissures détectées ; à savoir une fissure transversale, une fissure longitudinale, un faïençage, etc. Notons qu'en France, les différents types de dégradations font l'objet du catalogue associé à la méthode d'essai no. 38-2 intitulée "*Relevé de dégradations de surface des chaussées*" [3]. Dans ce cadre, il serait utile dans le futur de développer une approche automatique pour effectuer cette étape de classification. Une des méthodes que nous avons commencée à tester est basée sur l'analyse en composantes principales introduite dans la section 1.4.4.1.b du chapitre 1. En effet, les premiers tests réalisés permettent d'identifier les fissures non-ramifiées (horizontales, verticales, obliques) [165]. Pour cela, nous proposons dans un premier temps de trouver l'ellipse qui entoure la fissure détectée et ensuite à calculer son orientation dans

l'image pour décider à la fin s'elle correspond à un objet vertical, horizontal ou oblique. Un exemple montrant les résultats préliminaires de l'étape de classification est illustré dans la figure 6.2.




*(a) : Fissure horizontale* *(b) : Fissure verticale* *(c) : Fissure oblique*

FIGURE *6.2 – Tests préliminaires pour classifier les fissures détectées.*

# Bibliographie


[1] K. Wang and O. Smadi. Automated imaging technologies for pavement distress surveys. *Transportation Research E-Circular*, (E-C156), 2011.

[2] J. Ninot. *Reconnaissance et analyse automatique d'images pour la numérisation de l'environnement routier*. PhD thesis, Télécom Bretagne de Brest, 2011.

[3] L. Bertrand and P. Lepert. Catalogue des dégradations de surface de chaussées : méthode d'essai no. 52, Laboratoire Central des Ponts et Chaussées, 1998.

[4] *AIGLE-RN* project. http://media.lcpc.fr/ext/pdf/sem/2008_jtr_aigle.pdf.

[5] *AIGLE-RN* system. http:/www.normandie-centre.cerema.fr/IMG/pdf/15-AigleRNpress_cle243947.pdf.

[6] *VISIODEC* system. http://www.cete-aix.fr/imgarea/Cotita_SIR%2014%20AIGLE%20RN.pdf.

[7] *AMAC* system. http://www.erpug.org/media/files/forelasningar_2013/12_Luc%20Amaruy%20George%20AMAC_more%20than%20a%20profiler.pdf.

[8] K. Wang. Designs and implementations of automated systems for pavement surface distress survey. *Journal of Infrastructure Systems*, 6(1) :24–32, 2000.

[9] B. Schmidt. Automated pavement cracking assessment equipment : State of the art. *Routes-Roads, World Road Association (PIARC)*, 320 :35–44, 2003.

[10] *RoadCrack* system. http://www.csiro.au.

[11] T. Fukuhara, K. Terada, M. Nagao, A. Kasahara, and S. Ichihashi. Automatic pavement-distress-survey system. *Journal of Transportation Engineering*, 116(3) :280–286, 1990.

[12] W. Kaddah, Y. Ouerhani, A. Alfalou, M. Desthieux, C. Brosseau, and C. Gutierrez. Road marking features extraction using the $VIAPIX^{®}$ system. *Optics Communications*, 371 :117–127, 2016.

[13] M. Desthieux, C. Gutierrez, Y. Ouerhani, W. Kaddah, and A. Alfalou. ACTRIS brings its expertise in a new approach to improve knowledge and quality of topological and regulatory data dedicated to road, for OPTICITIES project needs. In *22nd (ITS) World Congress, Bordeaux, France*, 2015.

[14] VCRT, Vienna Convention on Road Traffic, 1968.

[15] M. Revilloud. *Architecture de perception adaptative pour la détection et le suivi multivoies*. PhD thesis, Université Paris-Saclay, 2016.

[16] V. Kastrinaki, M. Zervakis, and K. Kalaitzakis. A survey of video processing techniques for traffic applications. *Image and vision computing*, 21(4) :359–381, 2003.

[17] R.G. Von Gioi, J. Jakubowicz, J.M. Morel, and G. Randall. Lsd : A fast line segment detector with a false detection control. *IEEE transactions on pattern analysis and machine intelligence*, 32(4) :722–732, 2010.

[18] C. Schmid and A. Zisserman. Automatic line matching across views. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 666–671. IEEE, 1997.

[19] Y.S. Lee, H.S Koo, and C.S. Jeong. A straight line detection using principal component analysis. *Pattern Recognition Letters*, 27(14) :1744–1754, 2006.

[20] R.O. Duda and P.E. Hart. Use of the hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1) :11–15, 1972.

[21] M. Sarfraz. *Computer-aided intelligent recognition techniques and applications*. John Wiley & Sons, 2005.

[22] R.M. Haralick, S.R. Sternberg, and X. Zhuang. Image analysis using mathematical morphology. *IEEE transactions on pattern analysis and machine intelligence*, (4) :532–550, 1987.

[23] A.R. Mansouri, A.S. Malowany, and M.D. Levine. Line detection in digital pictures : A hypothesis prediction/verification paradigm. *Computer Vision, Graphics, and Image Processing*, 40(1) :95–114, 1987.

[24] J. Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, 8(6) :679–698, 1986.

[25] D.S. Guru, B.H. Shekar, and P. Nagabhushan. A simple and robust line detection algorithm based on small eigenvalue analysis. *Pattern Recognition Letters*, 25(1) :1–13, 2004.

[26] A. Rosenfeld. Picture processing by computer. *ACM Computing Surveys (CSUR)*, 1(3) :147–176, 1969.

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

[28] N. Kanopoulos, N. Vasanthavada, and R.L. Baker. Design of an image edge detection filter using the sobel operator. *IEEE Journal of solid-state circuits*, 23(2) :358–367, 1988.

[29] La détection de contours dans une image. http://dictionnaire.sensagent.leparisien.fr/D%C3%A9tection%20de%20contours/fr-fr/.

[30] M.A. Fischler and R.C. Bolles. Random sample consensus : A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6) :381–395, 1981.

[31] A. Lopez, C. Canero, J. Serrat, J. Saludes, F. Lumbreras, and T. Graf. Detection of lane markings based on ridgeness and ransac. In *Intelligent Transportation Systems, 2005. Proceedings. 2005 IEEE*, pages 254–259. IEEE, 2005.

[32] C. Lipski, B. Scholz, K. Berger, C. Linz, T. Stich, and M. Magnor. A fast and robust approach to lane marking detection and lane tracking. In *Image Analysis and Interpretation, 2008. SSIAI 2008. IEEE Southwest Symposium on*, pages 57–60. IEEE, 2008.

[33] H. Maitre. Le traitement des images. *Hermes Science Publications*, 43 :47–49, 2003.

[34] Y. Wang, E.K. Teoh, and D. Shen. Lane detection and tracking using b-snake. *Image and Vision computing*, 22(4) :269–280, 2004.

[35] M. Kass, A. Witkin, and D. Terzopoulos. Snakes : Active contour models. *International journal of computer vision*, 1(4) :321–331, 1988.

[36] C.H. Yeh and Y.H. Chen. Development of vision-based lane and vehicle detecting systems via the implementation with a dual-core dsp. In *Intelligent Transportation Systems Conference, 2006. ITSC'06. IEEE*, pages 1179–1184. IEEE, 2006.

[37] B. Soheilian, N. Paparoditis, D. Boldo, and J.P. Rudant. Automatic 3d extraction of rectangular roadmarks with centimeter accuracy from stereo-pairs of a ground-based mobile mapping system. In *In International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. Citeseer, 2007.

[38] J.C. McCall and M.M. Trivedi. Video-based lane estimation and tracking for driver assistance : Survey, System, and Evaluation. *IEEE transactions on intelligent transportation systems*, 7(1) :20–37, 2006.

[39] J.C. McCall and M.M. Trivedi. An integrated, robust approach to lane marking detection and lane tracking. In *Intelligent Vehicles Symposium, 2004 IEEE*, pages 533–537. IEEE, 2004.

[40] T. Veit, J.P. Tarel, P. Nicolle, and P. Charbonnier. Evaluation of road marking feature extraction. In *Intelligent Transportation Systems, 2008. ITSC 2008. 11th International IEEE Conference on*, pages 174–181. IEEE, 2008.

[41] K.C. Kluge, C.M. Kreucher, and S. Lakshmanan. Tracking lane and pavement edges using deformable templates. In *Enhanced and Synthetic Vision 1998*, volume 3364, pages 167–177. International Society for Optics and Photonics, 1998.

[42] M. Lan, M. Rofouei, S. Soatto, and M. Sarrafzadeh. Smartldws : A robust and scalable lane departure warning system for the smartphones. In *Intelligent Transportation Systems, 2009. ITSC'09. 12th International IEEE Conference on*, pages 1–6. IEEE, 2009.

[43] R. Danescu, S. Nedevschi, and T.B. To. A stereovision-based lane detector for marked and non-marked urban roads. In *Intelligent Computer Communication and Processing, 2007 IEEE International Conference on*, pages 81–88. IEEE, 2007.

[44] M. Bertozzi, A. Broggi, G. Conte, and A. Fascioli. Obstacle and lane detection on the argo autonomous vehicle, proceedings of IEEE Intelligent Transportation Systems conference. *Boston, MA*, 1997.

[45] F. Diebolt. *Reconnaissance des marquages routiers par traitement d'images*. PhD thesis, Strasbourg 1, 1996.

[46] H.A. Mallot, H.H. Bülthoff, J.J. Little, and S. Bohrer. Inverse Perspective Mapping simplifies optical flow computation and obstacle detection. *Biological cybernetics*, 64(3) :177–185, 1991.

[47] M. Nieto, L. Salgado, F. Jaureguizar, and J. Cabrera. Stabilization of Inverse Perspective Mapping images based on robust vanishing point estimation. In *Intelligent Vehicles Symposium, 2007 IEEE*, pages 315–320. IEEE, 2007.

[48] J. Rebut, A. Bensrhair, and G. Toulminet. Image segmentation and pattern recognition for road marking analysis. In *Industrial Electronics, 2004 IEEE International Symposium on*, volume 1, pages 727–732. IEEE, 2004.

[49] M. Bertozzi and A. Broggi. Real-time lane and obstacle detection on the gold system. In *Intelligent Vehicles Symposium, 1996., Proceedings of the 1996 IEEE*, pages 213–218. IEEE, 1996.

[50] J. Serra. *Image analysis and mathematical morphology*. Academic Press, Inc., 1983.

[51] C. Zhang, S. Murai, and E.P. Baltsavias. Road network detection by mathematical morphology. In *ISPRS Workshop" 3D Geospatial Data Production :*

*Meeting Application Requirements*. Institute of Geodesy and Photogrammetry, ETH-Hoenggerberg, 1999.

[52] S.C. Yi, Y.C. Chen, and C.H. Chang. A lane detection approach based on intelligent vision. *Computers & Electrical Engineering*, 42 :23–29, 2015.

[53] K. Ishikawa, T. Onishi, Y. Amano, T. Hashizume, J.I. Takiguchi, and S. Yoneyama. Precise road line localization using single camera and 3D road model. In *International Symposium on Robotics and Automation in Construction*. ISARC, 2006.

[54] J.W. Lee. A machine vision system for lane-departure detection. *Computer vision and image understanding*, 86(1) :52–78, 2002.

[55] D.J. Kang and M.H. Jung. Road lane segmentation using dynamic programming for active safety vehicles. *Pattern Recognition Letters*, 24(16) :3177–3185, 2003.

[56] C. Kreucher, S. Lakshmanan, and K. Kluge. A driver warning system based on the lois lane detection algorithm. In *Proceedings of IEEE International Conference on Intelligent Vehicles*, volume 1, pages 17–22. Stuttgart, Germany, 1998.

[57] J. Douret, R. Labayrade, J. Laneurit, and R. Chapuis. A reliable and robust lane detection system based on the parallel use of three algorithms for driving safety assistance. In *MVA*, pages 398–401, 2005.

[58] S.J. Wu, H.H. Chiang, J.W. Perng, C.J. Chen, B.F. Wu, T.T. Lee, et al. The heterogeneous systems integration design and implementation for lane keeping on a vehicle. *IEEE Transactions on Intelligent Transportation Systems*, 9(2) :246–263, 2008.

[59] S.S. Huang, C.J. Chen, P.Y. Hsiao, and L.C. Fu. On-board vision system for lane recognition and front-vehicle detection to enhance driver's awareness. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 3, pages 2456–2461. IEEE, 2004.

[60] K. Kluge and C. Thorpe. The yarf system for vision-based road following. *Mathematical and Computer Modelling*, 22(4-7) :213–233, 1995.

[61] D. Frank. Road markings recognition. In *Image Processing, 1996. Proceedings., International Conference on*, volume 2, pages 669–672. IEEE, 1996.

[62] A.S. Huang, D. Moore, M. Antone, E. Olson, and S. Teller. Finding multiple lanes in urban road networks with vision and lidar. *Autonomous Robots*, 26(2-3) :103–122, 2009.

[63] P. Foucher, Y. Sebsadji, J.P. Tarel, P. Charbonnier, and P. Nicolle. Detection and recognition of urban road markings using images. In *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, pages 1747–1752. IEEE, 2011.

[64] E. Pollard, D. Gruyer, J.P. Tarel, S.S. Ieng, and A. Cord. Lane marking extraction with combination strategy and comparative evaluation on synthetic and camera images. In *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, pages 1741–1746. IEEE, 2011.

[65] C. Nuthong and T. Charoenpong. Lane detection using smoothing spline. In *Image and Signal Processing (CISP), 2010 3rd International Congress on*, volume 2, pages 989–993. IEEE, 2010.

[66] Z. Kim. Robust lane detection and tracking in challenging scenarios. *IEEE Transactions on Intelligent Transportation Systems*, 9(1) :16–26, 2008.

[67] P. Subirats. *Conception et validation de méthodes de traitement d'images appliquées à la détection de fissures sur les images de surface de chaussées*. PhD thesis, Nantes, 2006.

[68] S. Chambon, P. Subirats, and J. Dumoulin. Introduction of a Wavelet Transform based on 2D matched filter in a Markov Random Field for fine structure extraction : Application on road crack detection. In *Image Processing : Machine Vision Applications II*, volume 7251, page 72510A. International Society for Optics and Photonics, 2009.

[69] H. Salam. Détection automatique de fissures dans des images de chaussée par modélisation markovienne. Technical report, Ecole central de Nantes, 2010.

[70] J. Dumoulin, P. Subirats, V. Legeay, D. Meignen, C. Gourraud, J.P. Delmulle, and M. Garnaud. Progressive automation of pavement surface distress detection by imaging techniques. *Actes des journées scientifiques du LCPC*, pages 32–45, 2005.

[71] S. Iyer and S.K. Sinha. A robust approach for automatic detection and segmentation of cracks in underground pipeline images. *Image and Vision Computing*, 23(10) :921–933, 2005.

[72] H. Rababaah, D. Vrajitoru, and J. Wolfer. Asphalt pavement crack classification : A comparison of GA, MLP, and SOM. In *Proceedings of Genetic and Evolutionary Computation Conference, Late-Breaking Paper*, 2005.

[73] S. Jitprasithsiri, H. Lee, R. Sorcic, and R. Johnston. Development of digital image-processing algorithm to compute unified crack index for salt lake city. *Transportation Research Record : Journal of the Transportation Research Board*, (1526) :142–148, 1996.

[74] C. Rasse, V. Leemans, M.F. Destain, and J.C. Verbrugge. Bearing capacity of roads, railways and airfiels, chapter application of image analysis to the identification and rating of road surface distress. *ISBN*, 90(5809) :396, 2002.

[75] K.M. Chua and L. Xu. Simple procedure for identifying pavement distresses from video images. *Journal of transportation engineering*, 120(3) :412–431, 1994.

[76] J. Chou, W.A. O'Neill, and H. Cheng. Pavement distress evaluation using fuzzy logic and moment invariants. *Transportation research record*, (1505), 1995.

[77] T.S. Nguyen, P. Vrignat, M. Avila, S. Begot, and F. Duculty. Détection de défauts sur la surface de chaussées par système de vision. In *Colloque GRETSI*, 2007.

[78] J.A. Acosta, J.L. Figueroa, and R.L. Mullen. Low-cost video image processing system for evaluating pavement surface distress. *Transportation research record*, (1348), 1992.

[79] J.H. Hansen, S.P. DeMerchant, and Pujol J.A. An adaptive segmentation technique for automated crack detection. In *78th Annual Meeting, TRB, Washington, DC*, 1999.

[80] S. Nallamothu and K. Wang. Experimenting with recognition accelerator for pavement distress identification. *Transportation Research Record : Journal of the Transportation Research Board*, (1536) :130–135, 1996.

[81] H.D. Cheng, J.R. Chen, C. Glazier, and Y.G. Hu. Novel approach to pavement cracking detection based on fuzzy set theory. *Journal of Computing in Civil Engineering*, 13(4) :270–280, 1999.

[82] T.S. Nguyen. *Extraction de structures fines sur des images texturées : application à la détection automatique de fissures sur des images de surface de chaussées.* PhD thesis, Université d'Orléans, 2010.

[83] J. Dumoulin, V. Legeay, D. Meignen, et al. Détection de fissures de surface de chaussees par techniques d'imagerie dans le visible. *JSI 2003-JOURNEES DES SCIENCES DE L'INGENIEUR DU RESEAU DES LABORATOIRES DES PONTS ET CHAUSSEES, DOURDAN, FRANCE, 9-11 DECEMBRE 2003*, 2003.

[84] V.K. Kalikiri, N.W. Garrick, and L.E.K. Achenie. Image-processing methods for automated distress evaluation. *Transportation research record*, (1435), 1994.

[85] H.N. Koutsopoulos and A.B. Downey. Primitive-based classification of pavement cracking images. *Journal of Transportation Engineering*, 119(3) :402–418, 1993.

[86] D. Darwin, M.N. Abou-Zeid, and K.W. Ketcham. Automated crack identification for cement paste. *Cement and concrete research*, 25(3) :605–616, 1995.

[87] D. Meignen, M. Bernadet, and H. Briand. One application of neural networks for detection of defects using video data bases : Identification of road distresses. In *Database and Expert Systems Applications, 1997. Proceedings., Eighth International Workshop on*, pages 459–464. IEEE, 1997.

[88] S. Chambon, C. Gourraud, J.M. Moliard, and P. Nicolle. Road crack extraction with adapted filtering and markov model-based segmentation : Introduction and validation. In *International Joint Conference on Computer Vision Theory and Applications, VISAPP*, page sp, 2010.

[89] N. Tanaka and K. Uematsu. A crack detection method in road surface images using morphology. *MVA*, 98 :17–19, 1998.

[90] K.C.P. Wang and W. Gong. Automated pavement distress survey : A review and a new direction. In *Pavement Evaluation Conference*, pages 21–25, 2002.

[91] Saddle points. http://fr.wikipedia.org/wiki/Point_col.

[92] A. Cord and S. Chambon. Automatic road defect detection by textural pattern recognition based on adaboost. *Computer-Aided Civil and Infrastructure Engineering*, 4(27) :244–249, 2011.

[93] J. Bray, B. Verma, X. Li, and W. He. A neural network based technique for automatic classification of road cracks. In *Neural Networks, 2006. IJCNN'06. International Joint Conference on*, pages 907–912. IEEE, 2006.

[94] H. Cheng, J. Wang, Y. Hu, C. Glazier, X. Shi, and X. Chen. Novel approach to pavement cracking detection based on neural network. *Transportation Research Record : Journal of the Transportation Research Board*, (1764) :119–127, 2001.

[95] R. Oullette, M. Browne, and K. Hirasawa. Genetic algorithm optimization of a convolutional neural network for autonomous crack detection. In *Evolutionary Computation, 2004. CEC2004. Congress on*, volume 1, pages 516–521. IEEE, 2004.

[96] Oliveira H. and P.L. Correia. Automatic road crack detection and characterization. *IEEE Transactions on Intelligent Transportation Systems*, 1(14) :155–168, 2013.

[97] M. Gavilán, D. Balcones, O. Marcos, D.F. Llorca, M.A. Sotelo, I. Parra, M. Ocaña, P. Aliseda, P. Yarza, and A. Amírola. Adaptive road crack detection system by pavement classification. *Sensors*, 11(10) :9628–9657, 2011.

[98] Y. Shi, L. Cui, Z. Qi, F. Meng, and Z. Chen. Automatic road crack detection using random structured forests. *IEEE Transactions on Intelligent Transportation Systems*, 17(12) :3434–3445, 2016.

[99] D. Meignen. Méthode d'imagerie : Premiers résultats. Technical report, Laboratoire Central des Ponts et Chaussées, 2003.

[100] H. Cheng, X. Jiang, J. Li, and C. Glazier. Automated real-time pavement distress analysis. *Transportation Research Record : Journal of the Transportation Research Board*, (1655) :55–64, 1999.

[101] L. Zhang, F. Yang, Y.D. Zhang, and Y.J. Zhu. Road crack detection using deep convolutional neural network. In *Image Processing (ICIP), 2016 IEEE International Conference on*, pages 3708–3712. IEEE, 2016.

[102] P. Delagnes and D. Barba. A markov random field for rectilinear structure extraction in pavement distress image analysis. In *Image Processing, 1995. Proceedings., International Conference on*, volume 1, pages 446–449. IEEE, 1995.

[103] N. Coudray, A. Karathanou, and S. Chambon. Multi-resolution approach for fine structure extraction : Application and validation on road images. In *International Joint Conference on Computer Vision Theory and Applications, VISAPP*, page sp, 2010.

[104] M. Do and M. Vetterli. The finite ridgelet transform for image representation. *IEEE Transactions on image Processing*, 12(1) :16–28, 2003.

[105] M.J. Shensa. The discrete wavelet transform : Wedding the a trous and mallat algorithms. *IEEE Transactions on signal processing*, 40(10) :2464–2482, 1992.

[106] O. Fabre. Traitement d'images par ondelettes appliqué à la détection de fissures de surface de chaussées. Technical report, Laboratoire Central des Ponts et Chaussées, 2003.

[107] P. Subirats, J. Dumoulin, V. Legeay, and D. Barba. Automation of pavement surface crack detection using the continuous wavelet transform. In *Image Processing, 2006 IEEE International Conference on*, pages 3037–3040. IEEE, 2006.

[108] S. Chambon and J.M. Moliard. Automatic road pavement assessment with image processing : Review and comparison. *International Journal of Geophysics*, 2011, 2011.

[109] D. Jouin, J. Cordonnier, A. Cord, and J.M. Moliard. Orsi fissures - opération de développement VISIODEC ii. Technical report, CEREMA - Direction Territoriale Normandie-Centre, 2014.

[110] D. Wright and V. Baltazart. Monitoring structural and surface conditions. Technical report, Tomorrow's Road Infrastructure Monitoring and Management, TRIMM, 2014.

[111] C. Sun and S. Pallottino. Circular shortest path in images. *Pattern Recognition*, 36(3) :709–719, 2003.

[112] S. Chambon. Detection of points of interest for geodesic contours : Application on road images for crack detection. In *International Joint Conference on Computer Vision Theory and Applications, VISAPP*, page sp, 2011.

[113] P. Charbonnier and J.M. Moliard. Calculs de chemins minimaux, suivi de fissures et autres applications. *Journées scientifiques du LCPC, Dourdan Google Scholar*, 2003.

[114] T.S. Nguyen, S. Begot, F. Duculty, and M. Avila. Free-Form Anisotropy : A new method for crack detection on pavement surface images. In *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pages 1069–1072. IEEE, 2011.

[115] Q. Zou, Y. Cao, Q. Li, Q. Mao, and S. Wang. Cracktree : Automatic crack detection from pavement images. *Pattern Recognition Letters*, 33(3) :227–238, 2012.

[116] V. Kaul, A. Yezzi, and Y. Tsai. Detecting curves with unknown endpoints and arbitrary topology using minimal paths. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(10) :1952, 2012.

[117] R. Amhaz. *Détection automatique de fissures dans des images de chaussée par sélection de chemins minimaux*. PhD thesis, École centrale de Nantes, 2015.

[118] L.D. Cohen and R. Kimmel. Global minimum for active contour models : A minimal path approach. *International journal of computer vision*, 24(1) :57–78, 1997.

[119] C. Sun and B. Appleton. Multiple paths extraction in images using a constrained expanded trellis. *IEEE transactions on pattern analysis and machine intelligence*, 27(12) :1923–1933, 2005.

[120] E.W. Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1) :269–271, 1959.

[121] P.E. Hart, N.J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2) :100–107, 1968.

[122] J.A. Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences*, 93(4) :1591–1595, 1996.

[123] R. Amhaz, S. Chambon, J. Idier, and V. Baltazart. Automatic crack detection on Two-Dimensional pavement images : An algorithm based on Minimal Path Selection. *IEEE Transactions on Intelligent Transportation Systems*, 17(10) :2718–2729, 2016.

[124] *LCMS* system. http://www.pavemetrics.com/fr/applications-fr/inspection-des-routes/lcms-fr/.

[125] $VIAPIX^{®}$ system. http://www.viapix.eu/.

[126] T. Vincenty. Transformation of co-ordinates between geodetic systems. *Survey Review*, 18(137) :128–133, 1965.

[127] J.T. Mayer. Méthode et droite de Mayer. http://www.bibmath.net/dico/index.php?action=affiche&quoi=./m/mayer.html.

[128] A. AlFalou, M. Elbouz, and H. Hamam. Segmented Phase-Only Filter binarized with a new error diffusion approach. *Journal of Optics A : Pure and Applied Optics*, 7(4) :183, 2005.

[129] Y. Ouerhani, M. Jridi, A. Alfalou, and C. Brosseau. Graphics processor unit implementation of correlation technique using a segmented phase only composite filter. *Optics Communications*, 289 :33–44, 2013.

[130] A. Alfalou and C. Brosseau. Robust and discriminating method for face recognition based on correlation technique and independent component analysis model. *Optics letters*, 36(5) :645–647, 2011.

[131] Y. Ouerhani, M. Elbouz, A. Alfalou, W. Kaddah, and M. Desthieux. Road sign identification and geolocation using JTC and $VIAPIX^{®}$ module. In *Pattern Recognition and Tracking XXIX*, volume 10649, page 106490J. International Society for Optics and Photonics, 2018.

[132] W. Kaddah, M. Elbouz, Y. Ouerhani, A. Alfalou, and M. Desthieux. Ongoing Studies for Automatic Road Anomalies Detection on $2D$ and $3D$ Pavement Images. In *The International Symposium on Optoelectronic Technology and Application (OTA), Beijing, China*, 2018.

[133] A. Alfalou, C. Brosseau, and W. Kaddah. Optimization of decision making for face recognition based on nonlinear correlation plane. *Optics Communications*, 343 :22–27, 2015.

[134] J.L. Horner and P.D. Gianino. Phase-only matched filtering. *Applied optics*, 23(6) :812–816, 1984.

[135] R. Amhaz, S. Chambon, J. Idier, and V. Baltazart. A new minimal path selection algorithm for automatic crack detection on pavement images. In *IEEE International Conference on Image Processing (ICIP 2014)*, pages 788–792, 2014.

[136] L. Yang. Contributions to improve and speed the existing MPS-based algorithm for both the semi-automated and the automated crack segmentation on 2D pavement images. Technical report, Master ARIA, Ecole Centrale de Nantes, 2015.

[137] V. Baltazart, L. Yang, P. Nicolle, and J.M. Moliard. Pseudo-Ground Truth data collection on pavement images. In *Signal Processing Conference (EUSIPCO), 2017 25th European*, pages 2021–2025. IEEE, 2017.

[138] V. Baltazart, P. Nicolle, and L. Yang. Ongoing tests and improvements of the MPS algorithm for the automatic crack detection within grey level pavement images. In *Signal Processing Conference (EUSIPCO), 2017 25th European*, pages 2016–2020. IEEE, 2017.

[139] G. Udny Yule. On the association of attributes in statistics : With illustrations from the material of the childhood society, &c. *Philosophical Transactions of the Royal Society of London Series A*, 194 :257–319, 1900.

[140] R. Sokal. A statistical method for evaluating systematic relationship. *University of Kansas science bulletin*, 28 :1409–1438, 1958.

[141] D.J. Rogers and T.T. Tanimoto. A computer program for classifying plants. *Science*, 132(3434) :1115–1118, 1960.

[142] R.R. Sokal and P.H.A. Sneath. Principles of numerical taxonomy WH freeman and CO. *San Francisco*, 1963.

[143] P. Jaccard. The distribution of the flora in the alpine zone. 1. *New phytologist*, 11(2) :37–50, 1912.

[144] L.R. Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3) :297–302, 1945.

[145] G.G. Simpson. Notes on the measurement of faunal resemblance. *American Journal of Science*, 258(2) :300–311, 1960.

[146] Z. Hubalek. Coefficients of association and similarity, based on binary (presence-absence) data : An evaluation. *Biological Reviews*, 57(4) :669–689, 1982.

[147] M.P. Sampat, Z. Wang, S. Gupta, A.C. Bovik, and M.K. Markey. Complex wavelet structural similarity : A new image similarity index. *IEEE transactions on image processing*, 18(11) :2385–2401, 2009.

[148] A.P. Zijdenbos, B.M. Dawant, R.A. Margolin, and A.C. Palmer. Morphometric analysis of white matter lesions in mr images : Method and validation. *IEEE transactions on medical imaging*, 13(4) :716–724, 1994.

[149] S. Chambon. Image database available. https://www.irit.fr/~Sylvie.Chambon/Crack_Detection_Database.html.

[150] D. Wright, V. Baltazart, N. Elsworth, R. Hamrouche, J. Karup, M.L. Antunes, S. McRobbie, V. Merecos, and T. Saarenketo. D4.3 monitoring structural and surface conditions. *Tomorrow's Road Infrastructure Monitoring and Management (TRIMM), FP7 project*, 285119, 2014.

[151] R. Amhaz, S. Chambon, J. Idier, and V. Baltazart. Minimal Path Selection (MPS) version 00, numéro IDDN.FR .001.420010.000.S.P.2018.000.21000, avril, 2016.

[152] W. Kaddah, M. Elbouz, Y. Ouerhani, V. Baltazart, M. Desthieux, and A. Alfalou. Optimized minimal path selection (OMPS) for automatic and unsupervised crack segmentation within Two-Dimensional pavement images. *The Visual Computer Journal (TVCJ)*, pages 1–17, 2018.

[153] F. Roli. Measure of texture anisotropy for crack detection on textured surfaces. *Electronics Letters*, 32(14) :1274–1275, 1996.

[154] L. Bruzzone, F. Roli, and S.B. Serpico. Crack detection by a measure of texture anisotropy. In *International Conference on Image Analysis and Processing*, pages 743–747. Springer, 1995.

[155] N. Otsu. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1) :62–66, 1979.

[156] V. Schmidlin, G. Favier, J.P. Stromboni, and B. Tomasini. Réseaux de neurones et fusion de données. In *14ème Colloque sur le traitement du signal et des images*. GRETSI, Groupe d'Etudes du Traitement du Signal et des Images, 1993.

[157] V. Baltazart, J.M. Moliard, R. Amhaz, L.M. Cottineau, A. Wright, D. Wright, and M. Jethwa. Automatic crack detection on pavement images for monitoring road surface conditions - some results from the collaborative fp7 trimm project. In *8th RILEM International Conference on Mechanisms of Cracking and Debonding in Pavements*, pages 719–724. Springer, 2016.

[158] W. Kaddah, M. Elbouz, Y. Ouerhani, A. Alfalou, and M. Desthieux. Automatic darkest filament detection (ADFD) : A new algorithm for cracks segmentation on Two-Dimensional $VIAPIX^{®}$ pavement images. *The Visual Computer Journal (TVCJ)*, 2019.

[159] F. Liu, G. Xu, Y. Yang, X. Niu, and Y. Pan. Novel approach to pavement cracking automatic detection based on segment extending. In *Knowledge Acquisition and Modeling, 2008. KAM'08. International Symposium on*, pages 610–614. IEEE, 2008.

[160] R.A. Fundakowski, R.K. Graber, R.C. Fitch, E.L. Skok, and E.O. Lukanen. Video image processing for evaluating pavement surface distress. *Final Report, National Cooperative Highway Research Program*, pages 1–27, 1991.

[161] H. Oliveira and P.L. Correia. Identifying and retrieving distress images from road pavement surveys. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pages 57–60. IEEE, 2008.

[162] N. Morizet. *Reconnaissance biométrique par fusion multimodale du visage et de l'iris*. PhD thesis, Télécom ParisTech, 2009.

[163] A. Ross and A.K. Jain. Multimodal biometrics : An overview. In *Signal Processing Conference, 2004 12th European*, pages 1221–1224. IEEE, 2004.

[164] E.W. Weisstein. Moore neighborhood. from mathworld-a wolfram web resource. *URL : http ://mathworld. worlfram. com/MooreNeighborhood. html*, 2005.

[165] W. Kaddah, M. Elbouz, Y. Ouerhani, A. Alfalou, and M. Desthieux. Automatic Pavement Crack Classification on Two-Dimensional $VIAPIX^{®}$ Images. International Society for Optics and Photonics, 2019.

[166] ACTRIS technologie & innovation. http://www.actris.com/.

[167] Q. Li, Q. Zou, D. Zhang, and Q. Mao. Fosa : F* seed-growing approach for crack-line detection from pavement images. *Image and Vision Computing*, 29(12) :861–872, 2011.

[168] L. Yang, V. Baltazart, R. Amhaz, and P. Jiang. A new a-star algorithm adapted to the semi-automatic detection of cracks within grey level pavement images. In *Eighth International Conference on Digital Image Processing (ICDIP 2016)*, volume 10033, page 100333Z. International Society for Optics and Photonics, 2016.

[169] A. Felner. Position paper : Dijkstra's algorithm versus uniform cost search or a case against Dijkstra's algorithm. In *Fourth annual symposium on combinatorial search*, 2011.

# Annexes

# Annexe A

## Opérations élémentaires de la morphologie mathématique

**Sommaire**


A.1 Opérateurs de la morphologie mathématique . . . . . **222**
A.1.1 Dilatation morphologique . . . . . 222
A.1.2 Érosion morphologique . . . . . 222
A.1.3 Ouverture morphologique . . . . . 223
A.1.4 Fermeture morphologique . . . . . 224

## A.1 Opérateurs de la morphologie mathématique

Quatre opérateurs morphologiques sont habituellement appliqués sur des images binaires ou en niveaux de gris : **dilatation** - **érosion** - **ouverture** - **fermeture**. Plus clairement, pour un élément structurant $B$ et pour une fonction $I$ de $\mathbb{R}^n \to \mathbb{R}$ :

### A.1.1 Dilatation morphologique

La dilatation morphologique est un opérateur permettant d'élargir la figure ou l'image à traiter. La hauteur et la largeur de la figure dilatée seront les sommes respectivement des hauteurs et des largeurs de la figure originale et l'élément structurant. Elle est définie par l'équation A.1 suivante :

$$\forall \overrightarrow{x} \in \mathbb{R}^n, \delta_B(I)(\overrightarrow{x}) = \sup_{\overrightarrow{y} \in K} [I(\overrightarrow{y}) + B(\overrightarrow{y} - \overrightarrow{x})] \tag{A.1}$$

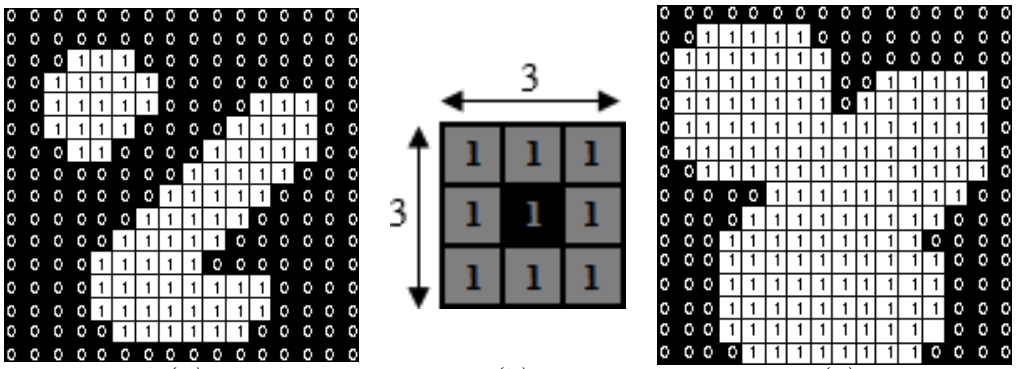



FIGURE *A.1 – Exemple illustrant l'effet de la dilatation morphologique sur une image binaire – (a) Une image binaire avant la dilatation - (b) Élément structurant en forme de carré de taille* $3 \times 3$ *- (c) Résultat de la dilatation en utilisant l'élément structurant illustré dans (b).*

### A.1.2 Érosion morphologique

L'érosion morphologique est l'opérateur inverse de la dilatation. Elle consiste à rétrécir la figure ou l'image à traiter. La hauteur et la largeur de la figure érodée seront les différences respectivement des hauteurs et des largeurs de la figure originale et l'élément structurant. Elle est définie par l'équation A.2 suivante :

$$\forall \overrightarrow{x} \in \mathbb{R}^n, \varepsilon_B(I)(\overrightarrow{x}) = \inf_{\overrightarrow{y} \in K} [I(\overrightarrow{y}) - B(\overrightarrow{y} - \overrightarrow{x})] \tag{A.2}$$

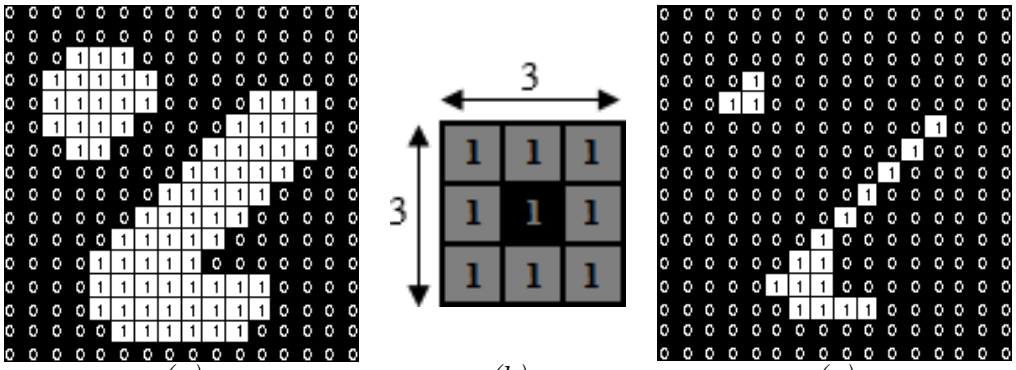



FIGURE *A.2 – Exemple illustrant l'effet de l'érosion morphologique sur une image binaire – (a) Une image binaire avant l'érosion - (b) Élément structurant en forme de carré de taille* $3 \times 3$ *- (c) Résultat de l'érosion en utilisant l'élément structurant illustré dans (b).*

Nous notons que la composition d'une dilatation avec une érosion (ou l'inverse) par le même élément structurant ne produit pas en général l'identité (ne permet pas de revenir à l'image départ) mais deux autres opérateurs s'interviennent dans ce cas.

### A.1.3 Ouverture morphologique

L'ouverture morphologique est obtenue en faisant une érosion suivie d'une dilatation de la figure. Elle a pour propriété d'éliminer toutes les parties des objets qui ne peuvent pas contenir l'élément structurant. Elle est ainsi définie par l'équation A.3 suivante :

$$\forall \overrightarrow{x} \in \mathbb{R}^n, O_B(I)(\overrightarrow{x}) = \delta_B(\varepsilon_B(I)(\overrightarrow{x})) \tag{A.3}$$

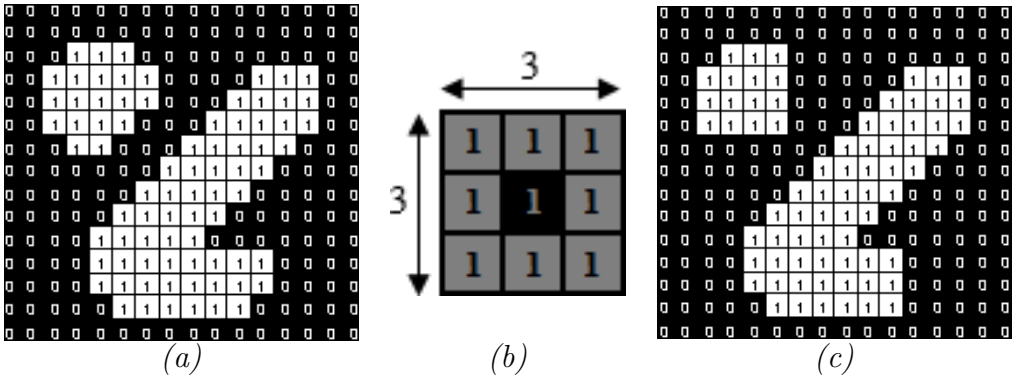



FIGURE *A.3 – Exemple illustrant l'effet de l'ouverture morphologique sur une image binaire – (a) Une image binaire avant l'ouverture - (b) Élément structurant en forme de carré de taille* $3 \times 3$ *- (c) Résultat de l'ouverture en utilisant l'élément structurant illustré dans (b).*

### A.1.4 Fermeture morphologique

La fermeture morphologique est l'inverse de l'ouverture. Elle est obtenue en faisant une dilatation suivie d'une érosion de la figure. Elle a pour propriété de combler tous les espaces entre les objets qui ne peuvent pas contenir l'élément structurant. Elle est ainsi définie par l'équation A.4 suivante :

$$\forall \overrightarrow{x} \in \mathbb{R}^n, C_B(I)(\overrightarrow{x}) = \varepsilon_B(\delta_B(I)(\overrightarrow{x})) \tag{A.4}$$

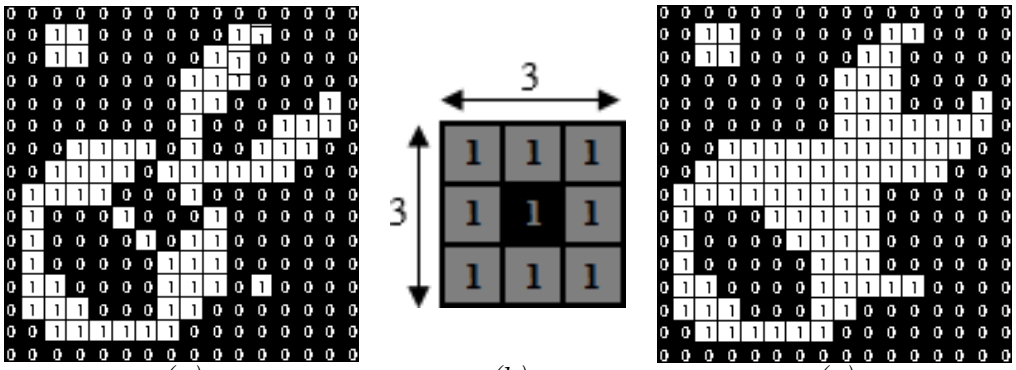



FIGURE *A.4 – Exemple illustrant l'effet de la fermeture morphologique sur une image binaire – (a) Une image binaire avant la fermeture - (b) Élément structurant en forme carré de taille* $3 \times 3$ *- (c) Résultat de la fermeture en utilisant l'élément structurant illustré dans (b).*

# Annexe B

## Données techniques sur le système $VIAPIX^{®}$

**Sommaire**


B.1 Système $VIAPIX^{®}$ . . . . . **227**
B.1.1 $VIAPIX^{®}$ acquisition . . . . . 227
B.1.2 $VIAPIX^{®}$ exploitation . . . . . 228

## B.1 Système $VIAPIX^{®}$

Dans cette annexe, nous présentons la solution $VIAPIX^{®}$ [125] entièrement imaginée et conçue par la société ACTRIS [166] pour la production de données géo-référencées.

### B.1.1 $VIAPIX^{®}$ acquisition

*(a)* *(b)*

FIGURE *B.1 – Photographie d'un ipad et d'un PC portable connecté à l'interface WEB.*

- **Odomètre** : Fixé par aimantation aux vis des roues du véhicule, il permet de positionner précisément chacune des données acquises par le système, dans le référentiel routier.
- **Carte processeur** : Elle réalise les tâches de compression et d'enregistrement.
- **Centrale inertielle** : Équipé d'une centrale inertielle spécialement développée pour l'environnement automobile, le module $VIAPIX^{®}$ acquisition conserve, en cas de masquage, la précision fournie par le système satellite durant plusieurs minutes. Elle tient compte de l'ensemble des capteurs, accéléromètres gyroscopes, encodeur et satellites pour calculer la position la plus précise. D'autre part, elle fournit en temps réel l'attitude du véhicule permettant la mesure de la déclivité et du dévers afin de vérifier la géométrie de la chaussée.
- **Alimentation électrique** : Le module $VIAPIX^{®}$ acquisition est un système basse consommation et est alimenté par une simple prise allume cigare. Conçu pour recevoir une tension d'entrée comprise entre 11 et 24 Volt, il ne consomme que 24 Watts en acquisition. Il intègre une alimentation sans interruption qui permet de fonctionner lors d'une coupure électrique.
- **Stockage embarqué** : Les systèmes de stockage conventionnels de type mémoire USB, disque dur ne permettent pas d'allier la rapidité, la capacité de stockage

et la fiabilité. Ceci est la raison pour laquelle, le système $VIAPIX^{®}$ acquisition est équipé de 200 Go de mémoire FLASH. Directement soudé sur la carte électronique, ce stockage FLASH est 4 fois plus rapide qu'un disque dur portable, et comme il n'intègre aucun élément amovible, le stockage est insensible aux variations et aux chocs. De plus, une carte mémoire SD est extractible de ce module afin de rapatrier les données sur le poste d'exploitation. Le téléchargement peut également s'effectuer par Ethernet ou WiFi.

- **Batterie** : La batterie permet l'alimentation sans interruption. Lorsque le système n'est brusquement plus alimenté durant un enregistrement, il continue à fonctionner pendant quelques minutes.

### B.1.2 $VIAPIX^{®}$ exploitation

Ce module permet traiter les données acquises le module d'acquisition. Plus précisément, il consiste à :

- Consulter les images de l'environnement routier en vue simple ou panoramique.
- Réaliser des mesures directement dans l'image.
- Inventorier le patrimoine routier.
- Exporter les traces ainsi que les objets inventoriés dans différents formats.

*(a)* *(b)*

FIGURE *B.2 – Mise en situtation du $VIAPIX^{®}$ exploitation.*

- **Construction d'un référentiel réseau** : L'utilisateur pointe directement dans l'image le PK, $VIAPIX^{®}$ exploitation crée automatiquement le référentiel de

la route quelque soit le sens de la mesure (croissant ou décroissant). Une fois le référentiel généré, toutes les images ainsi que toutes les données terrains (pente de la route, évènements ponctuels, etc.) sont référencées en : PK + abscisse, coordonnées terrestres (longitude, latitude et altitude).

FIGURE *B.3 – Cartographie.*

- **Visualisation des images relevées** : Ce module permet de consulter les images des routes en vue simple ou panoramique. La navigation immersive permet à l'utilisateur de se déplacer dans le réseau d'une façon intuitive. Travaillant simultanément avec une représentation cartographique, l'utilisateur dispose de la position géographique, des conditions de prise de vue et de l'altitude du véhicule de chaque image.

FIGURE *B.4 – VIAPIX® exploitation.*

- **Relevé des objets** : Le module d'exploitation calcule la position géographique (référence terrestre et routier) de chaque élément pointé dans l'image. L'objet résultant peut être : continu, discontinu, ponctuel. En parallèle, l'utilisateur dispose d'une représentation cartographique l'informant sur la progression de son relevé.

# Annexe C

## Algorithme de Dijkstra appliqué à la segmentation de fissures

### Sommaire


C.1 Introduction : les problématiques du plus court chemin dans le cadre de la thèse . . . . . **232**

C.2 Utilisation de l'algorithme de Dijkstra dans le cadre de la thèse . . . **232**

C.3 Illustration du fonctionnement de l'algorithme de Dijkstra . . . . . **233**

C.3.1 Cas de la problématique SPSP . . . . . 233

C.3.2 Cas de la problématique SSSP . . . . . 238

## C.1 Introduction : les problématiques du plus court chemin dans le cadre de la thèse

On considère qu'une image peut être représentée par un graphe, dont le niveau de gris de chaque pixel représente le poids (non négatif) de chaque branche du graphe. La recherche de fissures (de niveau de gris faible) revient donc à rechercher dans ce graphe le chemin le plus court (cf. chapitre 1).

Dans le cadre de la thèse, nous distinguons deux problématiques de plus court chemin :

- La problématique "one-to-one" (Single Pair Shortest Path ou SPSP) consiste à déterminer le plus court chemin entre un pixel source et un pixel destination ; elle intervient pour établir l'image de pseudo-vérité terrain (PVT) de manière semi-automatique (section 3.3.1 du chapitre 3).
- La problématique "one-to-all ou one-to-many" (Single Source Shortest Path ou SSSP) consiste à déterminer les plus courts chemins entre un pixel source et plusieurs pixels destination en une seule passe ; elle est utilisée dans l'étape 2 de l'algorithme MPS pour calculer les 8 chemins encadrant un pixel source dans une ROI de taille $3P \times 3P$ (cf. stratégie "étoile" de la section 3.2.2.2 du chapitre 3).

Une version naïve serait d'itérer un algorithme SPSP sur le nombre de pixels destination. En contrepartie, le principe de l'algorithme SSSP permet d'éviter de parcourir le même voisinage plusieurs fois et ainsi de gagner du temps dans la recherche de chemins minimaux ayant un même point source.

## C.2 Utilisation de l'algorithme de Dijkstra dans le cadre de la thèse

La version originale de Dijkstra [120] permet de résoudre la problématique SPSP, en minimisant la distance de Manhattan (ou *taxi-cab geometry*) définie par l'équation 1.23. De nombreux algorithmes concurrents existent dans la littérature pour identifier les fissures dans une image de chaussée (par exemple, les algorithmes F* [167] et A* [168]).

L'algorithme de Dijkstra a été rapidement étendu à la problématique complémentaire, i.e. SSSP, sans modification du synoptique. Dijkstra est connu comme l'algorithme SSSP asymptotiquement le plus rapide de la littérature. Il effectue une recherche en aveugle des plus courts chemins dans l'image de manière optimale et exhaustive.

De fait, Dijkstra est devenu au fil du temps un algorithme incontournable, qualifié de *ubiquitous* dans la littérature, et utilisé pour les deux problématiques citées. Les problématiques SPSP et SSSP concernent de nombreuses applications dans le domaine

de l'informatique, des communications, de la logistique (réduction du temps de trajet dans un réseau de transport) [169]. Dijkstra est communément enseigné en informatique.

Par souci de simplicité et de performance, nous avons utilisé l'algorithme de Dijkstra dans la thèse pour les deux problèmatiques SPSP et SSSP. De plus, nous utiliserons une version directionnelle de Dijkstra au chapitre 4 pour optimiser l'étape 1 de la méthode MPS (méthode FFA [114]).

## C.3 Illustration du fonctionnement de l'algorithme de Dijkstra

### C.3.1 Cas de la problématique SPSP

Dans l'exemple de la figure C.1, nous détaillons le fonctionnement de l'algorithme de Dijkstra pour calculer le chemin minimal entre le pixel source en rouge et le pixel destination en vert.

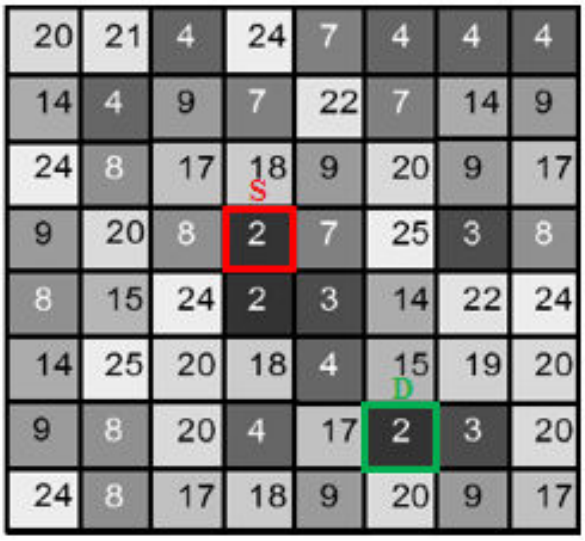



FIGURE *C.1 – Point source (S) et destination (D) pour le calcul du plus court chemin – Dans cet exemple, nous cherchons à trouver le meilleur chemin (ou le plus court chemin) qui relie le point S avec le point D.*

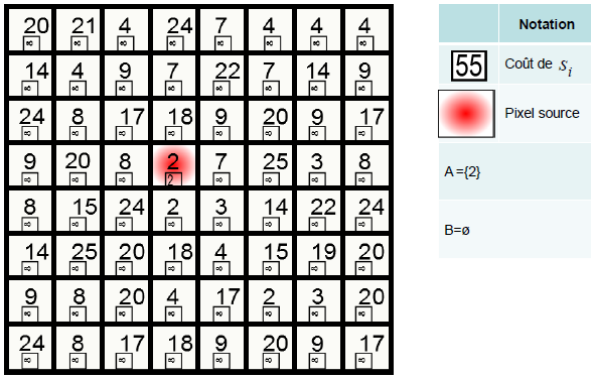
Nous présentons dans la figure C.2 ci-dessous les différentes notations et explications nécessaires pour la compréhension de la méthode de calcul du chemin minimal.

| Représentation | Notation | Explication |
|---|---|---|
| $\infty$ | Coût de $S_i$ | Somme des niveaux de gris de la source jusqu'au pixel $S_i$ |
| | Pixel source | Pixel central |
| | Pixel dans B | B : ensemble des pixels dont le coût est déjà calculé/connu |
| | Pixel $b_{min}$ | Pixel de l'ensemble B qui a le coût minimal |
| | Pixel dans A | A : ensemble des pixels qui ont le coût minimal |
| | Indicateur | Indicateur du pixel suivant |
| $R_P$ | Référent | Pixel qui a permis d'obtenir la valeur de coût minimal |

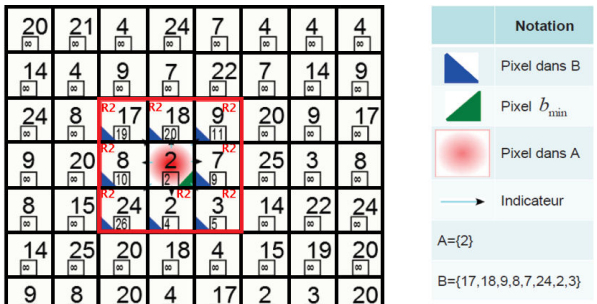
FIGURE *C.2 – Notations.*

Pour la clarté, nous notons que la notation $R_i = \{P\}$ dans le texte ci-après (explication de toutes les étapes de l'algorithme de Dijkstra) correspond à $R_P$ dans les illustrations. Par exemple, la notation $R_1 = \{2\}$ qui correspond à $R_2$, signifie que le pixel d'indice égal à 1 correspond au pixel ayant une valeur de niveau de gris égale à 2 dans l'image à traiter.

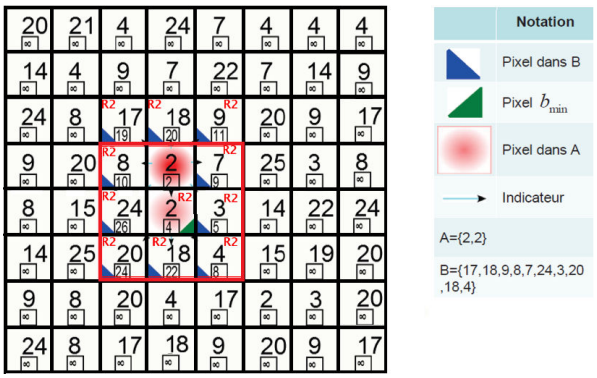
**Étape 1** : Elle consiste à initialiser les valeurs de coûts dans l'image. L'ensemble A est initialisé à la valeur du pixel source en rouge. Les autres pixels sont initialisés à l'infini. L'ensemble B est vide, ainsi que l'ensemble R.



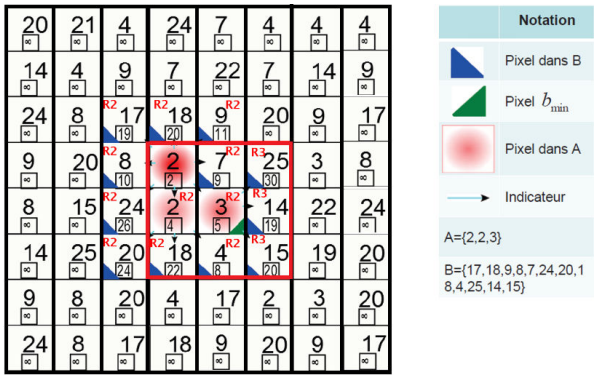

**Étape 2** : Pour chacun des pixels voisins du pixel source (il y a 8 voisins à la première itération), le coût des pixels dans l'ensemble B est calculé, comme la somme des valeurs du pixel courant avec le dernier pixel de l'ensemble A. Si le coût est inférieur au coût initial, l'ensemble B des coûts est réactualisé et le pixel référent est sauvegardé. A la première itération, le coût initial du pixel adjacent 17 de l'exemple est remplacé par la nouvelle valeur calculée (19 = 17+2). Le pixel référent qui a permis d'obtenir cette valeur de coût est le pixel 2 (pixel central).



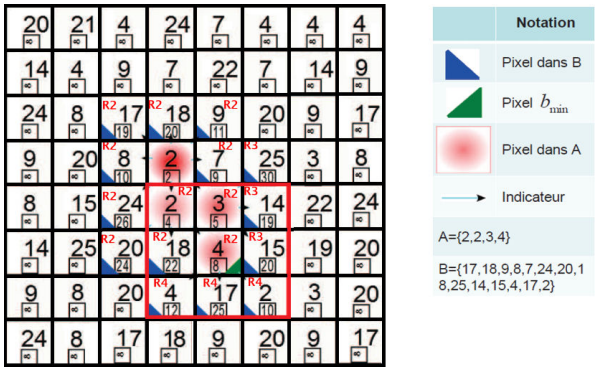

**Étape 3** : Nous sélectionnons $b_{min}$, le pixel de l'ensemble B, qui a la valeur de coût minimale. Dans l'exemple, il s'agit du pixel {2} au-dessous du pixel central. Ce pixel est ajouté à l'ensemble A, et est pris comme nouveau pixel central pour poursuivre la recherche du chemin le plus court. Le pixel référent est $R_1 = \{2\}$.



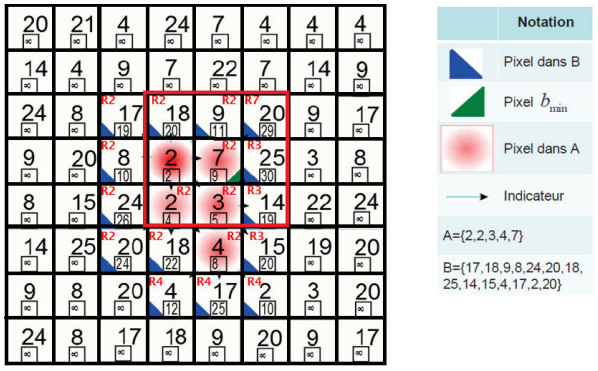
**Étape 4** : Les étapes 2 et 3 sont répétées à partir du nouveau pixel central {2}. Nous ajoutons ainsi les 3 pixels voisins du pixel central à l'ensemble B (les pixels {20, 18, 4}) et nous calculons la valeur de coût du chemin associé. Dans l'exemple, le pixel {3} qui correspond à la valeur de coût minimale (égale à 5), sera ajouté à l'ensemble A, et pris comme nouveau pixel central. Le pixel référent est $R_2 = \{2\}$ et R = $\{R_1, R_2\}$.



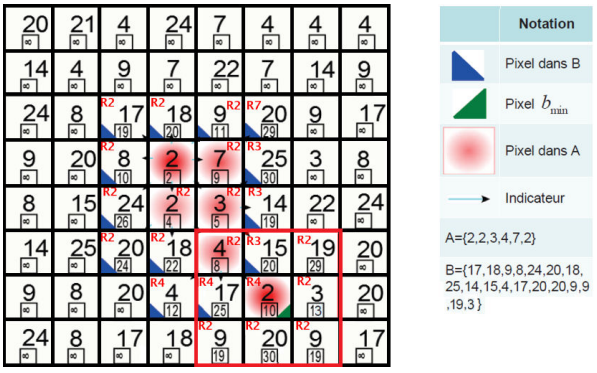

**Étape 5** : L'étape 4 est répétée à partir du nouveau pixel central {3}. Dans l'exemple, le pixel {4} ayant une valeur de coût égale à 8, sera ajouté à l'ensemble A. Le pixel référent est $R_3 = \{2\}$ et R = $\{R_1, R_2, R_3\}$.




**Étape 6** : L'étape 5 est répétée à partir du nouveau pixel central {4}. Dans l'exemple, le pixel {7} ayant une valeur de coût égale à 9, sera ajouté à l'ensemble A. Le pixel référent est $R_4 = \{2\}$ et R = $\{R_1, R_2, R_3, R_4\}$.



**Étape 7** : L'étape 6 est répétée à partir du nouveau pixel central {7}. Dans l'exemple, le pixel {2} ayant une valeur de coût égale à 10, sera ajouté à l'ensemble A. Le pixel référent est $R_5 = \{4\}$ et R = $\{R_1, R_2, R_3, R_4, R_5\}$. Une fois le pixel de destination atteint, i.e., le pixel 2 dans l'exemple, l'algorithme génère le chemin minimal trouvé à partir du pixel destination. Pour ce faire, nous utilisons les pixels de l'ensemble A et les pixels référents de l'ensemble R. Le pixel référent au pixel destination est $R_5 = \{4\}$. Ce pixel {4} a une valeur de coût minimale égale à 8 obtenue grâce au pixel référent $R_3 = \{2\}$. Cette étape est réalisée d'une manière itérative jusqu'à atteindre le pixel source. Nous avons donc à la fin de cette étape (Figure C.3) la liste des pixels formant le plus court chemin du pixel central en rouge au pixel destination en vert (Figure C.1).



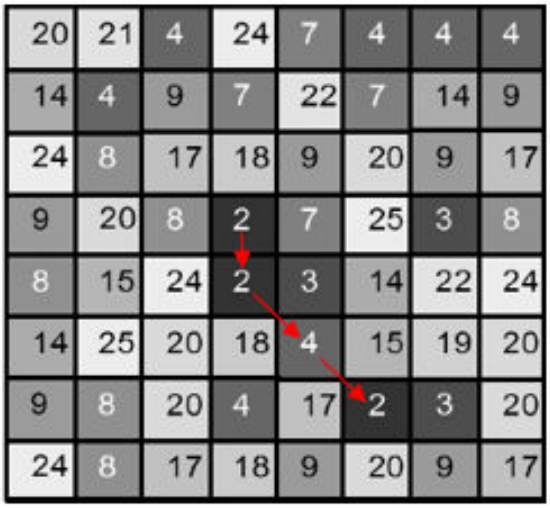



FIGURE *C.3 – Chemin minimal obtenu avec l'algorithme de Dijkstra.*

### C.3.2 Cas de la problématique SSSP

Pour illustrer le cas SSSP, il faut imaginer d'autres points destination disposés dans la ROI de taille $8 \times 8$ de l'exemple précédent.

En effet, l'algorithme peut se poursuivre de proche en proche à partir du dernier pixel central sans modification du synoptique. Dijkstra effectue une recherche sans nécessiter de revenir à l'étape initiale (pixel source central de l'étape 1). A chaque fois qu'un des pixels destination est atteint, l'algorithme détermine la liste des pixels du chemin correspondant et poursuit vers un autre pixel destination.

# Publications de l'auteur

## Articles de journaux

1. **W. Kaddah**, M. Elbouz, Y. Ouerhani, A. Alfalou, and M. Desthieux. Automatic darkest filament detection (ADFD) : A new algorithm for crack extraction on Two-Dimensional $VIAPIX^{®}$ pavement images. *The Visual Computer Journal (TVCJ)*, 2019 (***en soumission***).

2. **W. Kaddah**, M. Elbouz, Y. Ouerhani, V. Baltazart, M. Desthieux, and A. Alfalou. Optimized minimal path selection (OMPS) method for automatic and unsupervised crack segmentation within Two-Dimensional pavement images. *The Visual Computer Journal (TVCJ)*, 1-17, 2018.

3. **W. Kaddah**, Y. Ouerhani, A. Alfalou, M. Desthieux, C. Brosseau, and C. Gutierrez. Road marking features extraction using the $VIAPIX^{®}$ system. *Optics Communications*, 371, 117-127, 2016.

4. A. Alfalou, C. Brosseau, and **W. Kaddah**. Optimization of decision making for face recognition based on nonlinear correlation plane. *Optics Communications*, 343, 22-27, 2015.

## Articles de conférences internationales

1. **W. Kaddah**, M. Elbouz, Y. Ouerhani, A. Alfalou, and M. Desthieux. Automatic Pavement Crack Classification on Two-Dimensional $VIAPIX^{®}$ Images. International Society for Optics and Photonics, SPIE, 2019.

2. **W. Kaddah**, M. Elbouz, Y. Ouerhani, A. Alfalou, and M. Desthieux. Ongoing Studies for Automatic Road Anomalies Detection on $2D$ and $3D$ Pavement Images. The *International Symposium on Optoelectronic Technology and Application (OTA)*, Beijing, China, 2018.

3. Y. Ouerhani, M. Elbouz, A. Alfalou, **W. Kaddah**, and M. Desthieux. Road sign identification and geolocation using JTC and $VIAPIX^{®}$ module. In *Pattern Recognition and Tracking XXIX*, 10649, 106490J, International Society for Optics and Photonics, 2018.

4. M. Desthieux, C. Gutierrez, Y. Ouerhani, **W. Kaddah**, and A. Alfalou. ACTRIS brings its expertise in a new approach to improve knowledge and quality of topological and regularity data dedicated to road, for OPTICITIES project needs. In *proceeding 22nd ITS world Congress*, Bordeaux, France, 2015.

UNIVERSITE BRETAGNE LOIRE / MATHSTIC

UBO
Université de Bretagne Occidentale

**Titre :** « Apport de nouveaux outils de traitement d'images et de programmation pour le relevé automatique de dégradations sur chaussées ».

**Mots clés :** Traitement d'images, détection automatique, dégradations, marquage au sol, fissure, chaussée.

**Résumé :**
Le réseau routier subit des dégradations sous l'effet du trafic et des conditions climatiques. Le relevé dans les images de différents types de défauts de surface permet d'évaluer l'état du réseau et de programmer des opérations de maintenance nécessaires. Le but de cette thèse est ainsi de développer des méthodes non-supervisées dédiées à l'analyse des images 2D et 3D. Nous nous focalisons sur la détection de dégradations du marquage routier et la détection des fissures sur la chaussée. Dans le cadre de la signalisation horizontale, notre objectif est de réaliser un algorithme capable de détecter, reconnaitre, géo-localiser et quantifier l'état du marquage routier à l'aide d'un système d'imagerie panoramique. Le traitement d'images effectué utilise une méthode de segmentation couleur pour faciliter la phase d'extraction des zones de marquages routiers. Ensuite, une technique de perspective inverse est appliquée pour faciliter l'identification des objets détectés. L'état du marquage est établi à partir des variations des caractéristiques géométriques (longueur, largeur, etc.) et colorimétriques (niveau de couleur blanche) des objets identifiés dans l'image. Dans le cadre de la détection des fissures, notre aspiration consiste à extraire automatiquement les fissures en surface de chaussée, en supposant que celles-ci sont des structures fines et sombres dans l'image. Parmi les nombreuses méthodes existantes, nos approches retenues suivent un schéma classique composé de trois phases principales, à savoir une phase de pré-traitement pour réduire la quantité d'information à traiter, une phase de traitement pour extraire les points ayant une forte vraisemblance d'appartenir à une fissure et une phase de post-traitement pour estimer la gravité du matériel. Les performances de nos algorithmes sont évaluées sur des images réelles 2D et 3D issues de 3 capteurs différents (VIAPIX®, LCMS et Aigle-RN).

**Title:** « Contribution of new image processing and programming tools for automatic pavement degradations detection ».

**Keywords:** Image processing, automatic detection, degradations, road marking, crack, pavement.

**Abstract:**
The road network is subject to degradations due to traffic and weather conditions. The detection of surface defects within pavement images is used to evaluate the road network and to schedule the necessary maintenance operations. The goal of this thesis is to develop unsupervised processing techniques for the analysis of 2D and 3D pavement images, which originate from imaging systems operating in the field of road engineering. We focus on the detection of road marking damage and the detection of cracks on the pavement. In the context of road marking, our objective is to realize an algorithm for detecting, recognizing, geo-locating and monitoring the wearing conditions of road marking using a panoramic imaging system. The performed image processing uses a color segmentation method to facilitate the extraction phase of the road marking zones. Then, an inverse perspective technique is applied to ease the identification of detected objects. The wearing conditions of road marking is established from the variations in the geometric (length, width, etc.) and colorimetric (white color level) characteristics of the objects identified in the image. In the context of road crack detection, our aspiration is the automatic segmentation of cracks within pavement images, assuming that they represent fine and dark features in the image. Among the many existing methods, our chosen approaches follow a classical scheme composed of three main phases, namely, a pre-processing phase to reduce the amount of information to be processed in the image, a processing phase to extract the points having a high likelihood of belonging to a crack on the road and a post-processing phase to estimate the severity and the damage level of the pavement. The performances of our proposed algorithms are evaluated on 2D and 3D real images, coming from 3 types of existing imaging devices for road engineering (VIAPIX®, LCMS and Aigle-RN).