



HAL
open science

Détection de contours dans les images CFA

Arezki Aberkane

► **To cite this version:**

Arezki Aberkane. Détection de contours dans les images CFA. Vision par ordinateur et reconnaissance de formes [cs.CV]. Université Lille 1 Sciences et technologies, 2017. Français. NNT : . tel-01682133

HAL Id: tel-01682133

<https://hal.science/tel-01682133v1>

Submitted on 24 Jan 2018

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.

Numéro attribué par la bibliothèque

| | | | | | | | | | |
|--|--|--|--|--|--|--|--|--|--|
| | | | | | | | | | |
|--|--|--|--|--|--|--|--|--|--|

THÈSE

présentée à

L'UNIVERSITÉ des SCIENCES et TECHNOLOGIES de LILLE

pour l'obtention du titre de

DOCTEUR

en Automatique, Génie informatique, Traitement du Signal et des Images

par

Arezki ABERKANE

| |
|--------------------------------------------------|
| DÉTECTION DE CONTOURS DANS LES IMAGES CFA |
|--------------------------------------------------|

Soutenue le – décembre 2017 devant le jury composé de :

| | | |
|---------------|------------------------------------------------|--------------------|
| K. HAMMOUCHE | Professeur à l'Université de Tizi Ouzou | Président |
| H. LAURENT | Maître de conférence HDR à l'INSA Val de Loire | Rapportrice |
| J-C. BURIE | Professeur à l'Université de la Rochelle | Rapporteur |
| P. MONTESINOS | Professeur à l'École des Mines d'Alès | Examinateur |
| L. MACAIRE | Professeur à l'Université de Lille1 | Directeur de Thèse |
| O. LOSSON | Maître de conférences à l'Université de Lille1 | Co-Encadrant |

Table des matières

| | |
|--------------------------------------------------|-----------|
| Notations | 7 |
| Introduction | 11 |
| 1 Contexte de la thèse | 11 |
| 2 Organisation de la thèse | 12 |
| Chapitre 1. Image numérique couleur | 15 |
| Introduction | 15 |
| 1 Représentation de la couleur | 15 |
| 2 Acquisition d'une image couleur | 16 |
| 2.1 Caméra tri-capteur | 17 |
| 2.2 Caméra mono-capteur | 18 |
| 3 Images CFA | 19 |
| 3.1 Différentes configurations du CFA | 19 |
| 3.2 Motifs locaux dans le CFA de Bayer | 21 |
| 3.3 Description de l'image CFA | 22 |
| 3.4 Débruitage de l'image CFA | 23 |
| 4 Formalisation du dématricage | 24 |
| Conclusion. | 26 |
| Chapitre 2. Dématricage | 27 |
| 1 Introduction | 27 |

| | | |
|---------------------------------------------------------------|-------------------------------------------------------------------------|-----------|
| 2 | Méthodes de dématricage | 28 |
| 2.1 | Méthode par interpolation bilinéaire | 28 |
| 2.2 | Méthodes utilisant la corrélation spectrale | 29 |
| 2.2.1 | Interpolation des rapports et différences de composantes | 30 |
| 2.2.2 | Interpolation des résidus | 31 |
| 2.3 | Méthodes basées sur une analyse spatiale | 32 |
| 2.3.1 | Interpolation directionnelle par sélection de direction . . | 33 |
| 2.3.2 | Interpolation directionnelle pondérée | 34 |
| 2.3.3 | Correction de la direction d'interpolation | 37 |
| 2.4 | Méthodes fréquentielles | 39 |
| 2.5 | Méthodes de dématricage et de débruitage | 40 |
| 3 | Analyse de l'image CFA | 41 |
| 3.1 | Caractérisation de textures | 41 |
| 3.2 | Compression/décompression | 42 |
| 3.3 | Détection des contours | 47 |
| 4 | Conclusion | 48 |
| Chapitre 3. Calcul du gradient à partir de l'image CFA | | 51 |
| 1 | Introduction | 51 |
| 2 | Gradient d'une image couleur dématricée | 52 |
| 2.1 | Gradient morphologique couleur | 52 |
| 2.2 | Gradient de Di Zenzo | 53 |
| 2.3 | Filtres de Deriche | 54 |
| 3 | Approches vectorielles pour le calcul des dérivées partielles | 56 |
| 3.1 | Estimation simple de la dérivée (SDE) | 56 |
| 3.2 | Estimation des dérivées par filtre de Deriche (DDE) | 60 |
| 3.3 | Estimation des composantes lissées par filtre de Deriche (SEDD) . | 61 |
| 3.4 | Filtres de Deriche décalés | 62 |
| 3.5 | Dérivation par filtre de Deriche décalé (SDD) | 64 |

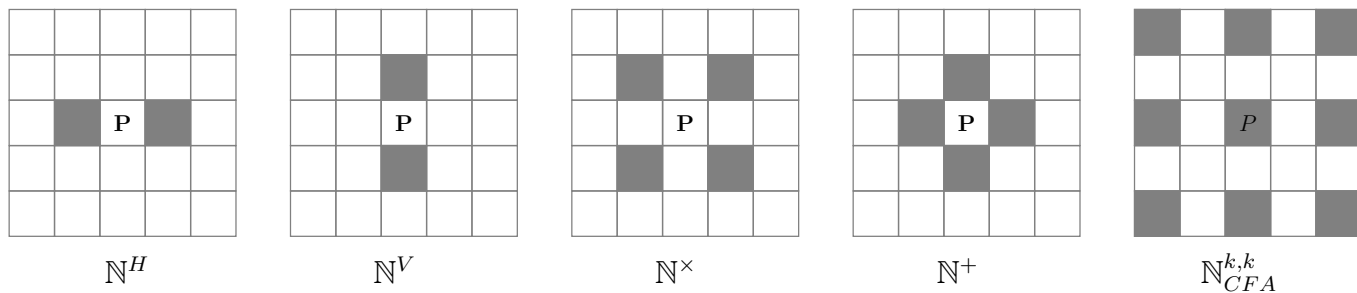
| | | |
|---------------------------------------------------------------------------------|----------------------------------------------------------------------------------------|------------|
| 3.6 | Lissage et dérivation par filtres de Deriche décalés (SDS) | 67 |
| 4 | Approches scalaires pour le calcul des dérivées partielles | 71 |
| 4.1 | Estimation de la luminance par sélection de fréquences (FS) | 72 |
| 4.2 | Estimation de la luminance par filtre gaussien (GL) | 73 |
| 4.3 | Estimation de la luminance par filtres de Deriche (DL) | 74 |
| 5 | Conclusion | 74 |
| Chapitre 4. Évaluation objective de la qualité de détection des contours | | 77 |
| 1 | Introduction | 77 |
| 2 | Données et procédure expérimentales | 78 |
| 2.1 | Base de données expérimentales | 78 |
| 2.1.1 | Carte des contours vérité terrain | 78 |
| 2.1.2 | Images couleur de référence | 78 |
| 2.2 | Procédure expérimentale | 81 |
| 2.3 | Critère d'évaluation de la qualité de détection des contours | 84 |
| 3 | Résultats expérimentaux | 85 |
| 3.1 | Détermination du paramètre α du filtre de Deriche | 85 |
| 3.2 | Résultats sur des images synthétiques | 85 |
| 3.3 | Résultats sur des images synthétiques iso-luminantes | 90 |
| 4 | Complexité algorithmique | 91 |
| 5 | Conclusion | 98 |
| Conclusion | | 101 |
| 1 | Synthèse des chapitres | 101 |
| 1.1 | Chapitre 1 : Image numérique couleur | 101 |
| 1.2 | Chapitre 2 : Dématricage | 101 |
| 1.3 | Chapitre 3 : Calcul du gradient à partir de l'image CFA | 102 |
| 1.4 | Chapitre 4 : Évaluation objective de la qualité de détection des contours | 103 |

| | | |
|----------------------------------------------------------------------|-----------------------------------------------------------------|------------|
| 2 | Perspectives | 103 |
| 2.1 | Dématriçage | 104 |
| 2.2 | Filtres optimaux sur l'image CFA | 104 |
| 2.3 | Évaluation des performances de détection des contours | 104 |
| 2.4 | Détection de points d'intérêt | 105 |
| Annexe A. Calcul des coefficients du filtre décalé de Deriche | | 107 |
| 1 | Quelques calculs préliminaires | 107 |
| 2 | Filtre de lissage décalé | 107 |
| 2.1 | Coefficient de normalisation | 107 |
| 2.2 | Équations de récurrence | 108 |
| 2.2.1 | Partie causale | 108 |
| 2.2.2 | Partie anticausale | 110 |
| 2.2.3 | Expression finale du filtre | 111 |
| 3 | Filtre de dérivation décalé | 111 |
| 3.1 | Coefficient de normalisation | 111 |
| 3.2 | Équations de récurrence | 112 |
| 3.2.1 | Partie causale | 112 |
| 3.2.2 | Partie anticausale | 113 |
| 3.2.3 | Expression finale du filtre | 113 |
| Bibliographie | | 114 |
| Table des figures | | 125 |
| Table des tableaux | | 127 |

Notations

Notations relatives aux données de l'image.

| | |
|----------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| I | Image en niveaux de gris. |
| \mathbf{I} | Image couleur. |
| M | Largeur d'une image. |
| N | Hauteur d'une image. |
| x | Abscisse d'un pixel dans une image ($0 \leq x \leq M - 1$). |
| y | Ordonnée d'un pixel dans une image ($0 \leq y \leq N - 1$). |
| $P(x,y)$ | Pixel de coordonnées (x,y) dans une image. |
| I^k | Image de la composante k dans l'image \mathbf{I} , où $k \in \{R,G,B\}$ pour l'espace RGB . |
| $I(P)$ ou $I(x,y)$ | Valeur du pixel $P(x,y)$ dans l'image I . On définit ainsi $I^R(P)$ ou $I^R(x,y)$, $I^G(P)$ ou $I^G(x,y)$, et $I^B(P)$ ou $I^B(x,y)$. |
| \hat{I}^L | Image de luminance estimée. |
| I^T | Carte des contours vérité terrain (image binaire). |
| I^E | Carte des contours détectés (image binaire). |
| $\tilde{\mathbf{I}}$ | Image couleur bruitée. |
| \tilde{I}^{CFA} | Image CFA bruitée. |
| \check{I}^{CFA} | Image CFA débruitée. |
| I^{CFA} | Image CFA. |
| S | Ensemble des pixels de I^{CFA} . |
| S^R | Ensemble des pixels de I^{CFA} où la composante R est disponible. |
| S^G | Ensemble des pixels de I^{CFA} où la composante G est disponible. |
| S^B | Ensemble des pixels de I^{CFA} où la composante B est disponible. |
| $S^{G,R}$ | Sous-ensemble des pixels de S^G dont les voisins horizontaux appartiennent à S^R . |
| $S^{G,B}$ | Sous-ensemble des pixels de S^G dont les voisins horizontaux appartiennent à S^B . |
| \ddot{I}^k | Image creuse formée des valeurs des pixels de S^k et 0 ailleurs. |



\mathbb{N}^x , \mathbb{N}^+ , \mathbb{N}^H ,
 \mathbb{N}^V , $\mathbb{N}_{CFA}^{k,k}$

Ensembles de pixels voisins du pixel considéré (illustrés dans la figure ci-dessus).

Notations relatives au processus de dématricage.

| | |
|------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| \mathcal{D}_{mos} | Procédure de dématricage. |
| $\hat{\mathbf{I}}$ | Image numérique couleur estimée par dématricage à partir de I^{CFA} . |
| $\hat{I}^k(x,y)$ ou $\hat{I}^k(P)$ | Valeur estimée de la composante $k \in \{R,G,B\}$ au pixel $P(x,y)$. |
| Δ^H , Δ^V | Norme (approchée) du gradient dans les directions horizontale et verticale. |
| $\hat{I}^{k,H}$, $\hat{I}^{k,V}$ | Valeur de la composante $k \in \{R,G,B\}$ estimée par interpolation horizontale ou verticale. |
| $\tilde{\delta}^{k-G,H}$ | Différence horizontale entre la valeur de la composante G disponible I^G (ou estimée $\hat{I}^{G,H}$) et la composante $k \in \{R,B\}$ estimée $\hat{I}^{k,H}$ (ou disponible I^k). $\tilde{\delta}^{k-G,V}$ est définie de manière analogue pour la direction verticale. |
| d^H , d^V | Norme des gradients horizontal et vertical calculées à partir de $\tilde{\delta}^{k-G,H}$ et $\tilde{\delta}^{k-G,V}$. |
| $w(x,y)$ | Pondération d'interpolation associée au pixel $P(x,y)$. |

Notations relatives à l'estimation des dérivées partielles.

| | |
|-----|------------------------------------------------------------------------------------------------------------|
| d | Distance spatiale séparant le pixel considéré P de son plus proche voisin horizontal (ou vertical) Q . |
|-----|------------------------------------------------------------------------------------------------------------|

Les notations des composantes lissées obtenues avec les différentes méthodes citées dans ce manuscrit sont illustrées dans le tableau ci-dessous avec :

(DD) Deriche Derivative, (SD) Simple Derivative, (H) Horizontal, (V) Vertical et (HV) Horizontal et Vertical.

| | | Composantes lissées | | | | | |
|------------|---------|-----------------------------------|-------------------------------------|-------------------------------------|--------------------------------------|------------------------|----------------------|
| | | I^{CFA} | | | | I^k | \hat{I}^L |
| | | DD | DD décalée | | | DD | DD |
| | | | H | V | HV | | |
| Lissage 1D | $d = 1$ | $\bar{I}_{\alpha,1}^{CFA}$ | – | – | – | $\bar{I}_{\alpha,1}^k$ | \bar{I}_{α}^L |
| | $d = 2$ | $\bar{I}_{2\alpha,2}^{CFA}$ | $\bar{I}_{2\alpha,2}^{CFA,H}$ | $\bar{I}_{2\alpha,2}^{CFA,V}$ | $\bar{I}_{2\alpha,2}^{CFA,HV}$ | – | – |
| Lissage 2D | $d = 1$ | \hat{I}_{α}^{DL} | – | – | – | – | – |
| | $d = 2$ | $\bar{\bar{I}}_{2\alpha,2}^{CFA}$ | $\bar{\bar{I}}_{2\alpha,2}^{CFA,H}$ | $\bar{\bar{I}}_{2\alpha,2}^{CFA,V}$ | $\bar{\bar{I}}_{2\alpha,2}^{CFA,HV}$ | – | – |

Les notations des dérivées partielles selon x et y calculées avec les différentes méthodes sont montrées dans le tableau ci-dessous :

| | | Dérivées partielles | | | | | | | | | | | |
|-----------|---------|---------------------|------------------------|--------------------------|--------------------------|---------------------------|---------|--------------------|------------------------|------------------------|-------------------------|---------------|--------------------------|
| | | I^{CFA} | | | | | I^k | | | | | \hat{I}^L | |
| | | SD | DD | DD décalée | | | SD | DD | DD décalée | | | SD | DD |
| | | | | H | V | HV | | | H | V | HV | | |
| Selon x | $d = 1$ | I_x^{CFA} | – | – | – | – | I_x^k | $I_{x,\alpha,1}^k$ | $I_{x,\alpha,1}^{k,H}$ | $I_{x,\alpha,1}^{k,V}$ | $I_{x,\alpha,1}^{k,HV}$ | \hat{I}_x^L | $\hat{I}_{x,\alpha,1}^L$ |
| | $d = 2$ | – | $I_{x,\alpha,2}^{CFA}$ | $I_{x,\alpha,2}^{CFA,H}$ | $I_{x,\alpha,2}^{CFA,V}$ | $I_{x,\alpha,2}^{CFA,HV}$ | – | – | – | – | – | – | – |
| Selon y | $d = 1$ | I_y^{CFA} | – | – | – | – | I_y^k | $I_{y,\alpha,1}^k$ | $I_{y,\alpha,1}^{k,H}$ | $I_{y,\alpha,1}^{k,V}$ | $I_{y,\alpha,1}^{k,HV}$ | \hat{I}_y^L | $\hat{I}_{y,\alpha,1}^L$ |
| | $d = 2$ | – | $I_{y,\alpha,2}^{CFA}$ | $I_{y,\alpha,2}^{CFA,H}$ | $I_{y,\alpha,2}^{CFA,V}$ | $I_{y,\alpha,2}^{CFA,HV}$ | – | – | – | – | – | – | – |

 $\hat{\mathbf{I}}_x$ Estimation de la dérivée simple selon x de l'image dématriciée $\hat{\mathbf{I}}$. $\hat{\mathbf{I}}_{x,\alpha,1}$ Estimation de la dérivée de Deriche selon x de l'image dématriciée $\hat{\mathbf{I}}$. $\mathbf{I}_{x,\alpha,2}$ Dérivée de Deriche selon x calculée par dérivation décalée à partir de I^{CFA} .

| | |
|---------------------------------|-------------------------------------------------------------------------------------------------------------------------------|
| \hat{I}_x^k | Estimation de la dérivée simple de I^k selon x par interpolation de I_x^{CFA} . |
| $\dot{\mathbf{I}}_x$ | Estimation de la dérivée simple selon x à partir de I^{CFA} . |
| $\hat{I}_{x,\alpha,2}^k$ | Estimation de la dérivée de Deriche de I^k selon x par interpolation de $I_{x,\alpha,2}^{CFA}$. |
| $\dot{\mathbf{I}}_{x,\alpha,2}$ | Estimation de la dérivée de Deriche selon x à partir de I^{CFA} . |
| $\dot{\mathbf{I}}_{x,\alpha,1}$ | Calcul de la dérivée de Deriche selon x par estimation des composantes lissées à partir de I^{CFA} . |
| $\mathbf{I}_{x,\alpha,1}$ | Dérivée de Deriche selon x calculée à partir de l'image I^{CFA} lissée par lissage décalé. |
| \hat{I}_x^L | Estimation de la dérivée simple selon x de l'image de luminance \hat{I}^L à partir de I^{CFA} . |
| $\hat{I}_{x,\alpha,1}^L$ | Estimation de la dérivée de Deriche selon x de l'image de luminance \hat{I}^L à partir de I^{CFA} . |
| $\hat{I}_{x,\alpha}^{DL}$ | Dérivée simple selon x de l'image de luminance \hat{I}^{DL} estimée avec le filtre de Deriche à partir de I^{CFA} . |
| $\hat{I}_{x,\alpha,1}^{DL}$ | Dérivée de Deriche selon x de l'image de luminance \hat{I}^{DL} estimée avec le filtre de Deriche à partir de I^{CFA} . |

Introduction

1 Contexte de la thèse

Il existe deux grandes familles de caméras numériques couleur : celles comportant trois capteurs dites *tri-capteurs* et les caméras qui ne contiennent qu'un seul capteur appelées caméras *mono-capteur*. Dans les caméras tri-capteurs, les stimuli de couleur sont séparés sur les trois capteurs photosensibles par un jeu de prisme et les trois images de composante R , G et B sont acquises simultanément sur les trois capteurs. Après recombinaison des trois images de composante, on obtient une image couleur. Bien que cette technologie fournisse des images d'excellente qualité, elle n'est quasiment pas utilisée dans les appareils compacts, car elle est encombrante et coûteuse. Afin de palier ces contraintes, les caméras *mono-capteur* ont été mises au point. Dans ce type de caméras, l'unique capteur est recouvert d'une mosaïque de filtres chromatiques (*Color Filter Array*, ou CFA), de telle sorte que l'image formée (dite « brute » ou CFA) ne comporte qu'une seule composante couleur par pixel. Sachant que chaque pixel d'une image couleur est caractérisé par trois composantes couleur, il manque donc deux composantes couleur en chaque pixel de l'image CFA. Une procédure appelée dématricage est appliquée sur l'image CFA afin d'obtenir ainsi une image couleur estimée.

Nos travaux de thèse s'inscrivent dans la continuité de travaux de recherche déjà réalisés au laboratoire CRISAL. Une première thèse de doctorat a été menée par Yanqin Yang [72] sur l'évaluation des méthodes de dématricage et l'étude des différents artefacts générés par ces méthodes. De cette étude, il est ressorti que les artefacts sont préjudiciables aux procédures d'analyse bas niveau de l'image couleur estimée telles que la détection des contours. Partant de cette conclusion, d'autres travaux ont étudié la possibilité d'utiliser directement l'image CFA pour caractériser des textures couleur. Après avoir mis en évidence la dégradation de l'information de texture par le dématricage, Olivier Losson et Ludovic Macaire ont proposé plusieurs descripteurs de texture adaptés aux images CFA dans un but de classification. Des tests sur des bases de référence

présentant différentes conditions (variation de l'illuminant, notamment) ont fourni des performances prometteuses, tant en termes de taux de bonne classification (comparables à ceux obtenus sur les images estimées) qu'en temps de calcul. Mes travaux de thèse s'inscrivent dans le prolongement de ces travaux. Il s'agit d'explorer dans quelle mesure l'image CFA peut être directement exploitée pour en détecter des pixels caractéristiques à savoir les pixels contours. Cela nécessite de transposer à l'image CFA les méthodes conçues pour les images couleur en les adaptant à la structure particulière de cette image.

2 Organisation de la thèse

Ce manuscrit est organisé en quatre chapitres. Le premier est consacré à l'acquisition des images numériques couleur avec les deux familles de caméras, tri-capteurs ou mono-capteur. Nous commençons par quelques considérations générales sur la représentation de la couleur ainsi que sur le système visuel humain dont s'inspirent ces deux techniques d'acquisition d'images numériques couleur. Ensuite, nous présentons ces deux techniques en se focalisant davantage sur le dispositif mono-capteur. La troisième partie est consacrée à la description de l'image CFA. Nous présentons les différentes configurations des filtres équipant les caméras mono-capteur en s'attardant particulièrement sur le filtre de Bayer utilisé dans l'ensemble de nos travaux. Un aperçu des méthodes de débruitage de l'image CFA clôt cette partie. Nous terminons ce premier chapitre en formalisant le processus de dématricage qui permet d'estimer une image couleur pleinement définie à partir de l'image CFA.

Le deuxième chapitre est divisé en deux grandes parties. La première est consacrée à un rapide tour d'horizon des méthodes de dématricage, à commencer par la plus ancienne et la plus simple utilisant une interpolation bilinéaire. Puis sont présentées les approches basées sur les corrélations spectrale et spatiale et celles basées sur l'analyse fréquentielle. Nous décrivons notamment en détails les quelques méthodes les plus performantes de l'état de l'art. La deuxième partie est consacrée à l'état de l'art sur les travaux exploitant directement l'image CFA. Nous commençons par présenter les travaux qui s'intéressent à l'exploitation de l'image CFA afin de caractériser et classifier des textures couleur ou à sa compression. Ensuite, nous détaillons l'unique méthode que propose la littérature pour la détection des contours directement à partir de l'image CFA.

Dans le troisième chapitre, nous commençons par rappeler comment peut être calculé un gradient couleur, puis nous présentons le filtre de Deriche qui permet de calculer

les dérivées premières partielles d'une composante couleur ou d'une image en niveaux de gris. Ensuite, nous détaillons les méthodes proposées pour le calcul des dérivées partielles à partir de l'image CFA. Ces méthodes se répartissent en deux approches, *vectorielle* et *scalaire*. La première approche exploite l'aspect vectoriel de l'information couleur en calculant les dérivées partielles des trois composantes couleur en chaque pixel. Dans cette approche, nous proposons trois façons de procéder : la première estime les composantes de dérivée partielle manquantes en chaque pixel, la deuxième estime les composantes couleur lissées manquantes et la dernière calcule directement les trois composantes de dérivée partielle en chaque pixel de l'image CFA en utilisant les filtres de Deriche décalés. L'approche scalaire exploite l'image mono-canal de luminance. Nous proposons une méthode qui utilise le filtre de lissage bidimensionnel de Deriche afin d'estimer, à partir de l'image CFA, une image de luminance adaptée à la détection des contours.

Dans le quatrième chapitre, nous nous intéressons à l'évaluation objective des différentes méthodes que nous proposons pour la détection des contours dans l'image CFA et nous comparons les performances de ces méthodes avec celles des approches classiques. Comme nous ne disposons pas de base d'images couleur avec une carte des contours vérité terrain, nous construisons une base d'images synthétiques en prenant en compte tous les paramètres qui influent sur la détection des contours dans une image couleur. Ensuite, nous présentons la procédure expérimentale suivie et le critère utilisé afin d'évaluer les performances des différentes méthodes. Finalement, nous fournissons et commentons les résultats de détection des contours et discutons la complexité algorithmique de chaque méthode testée.

Chapitre 1

Image numérique couleur

Introduction

Ce premier chapitre est consacré à l'acquisition des images numériques couleur. Deux techniques permettent d'acquérir ces images : les caméras dotées de trois capteurs sensibles aux trois couleurs primaires (rouge, vert et bleu) et les caméras équipées d'un seul capteur recouvert d'une matrice de filtres spectralement sélectifs. Avant de détailler ces deux techniques, nous introduisons quelques notions sur la représentation de la couleur ainsi que sur le système visuel humain dont s'inspirent ces deux techniques d'acquisition d'images numériques couleur. Dans la deuxième partie de ce chapitre, nous développons les deux technologies de caméras et détaillons le dispositif d'acquisition mono-capteur. La troisième partie est consacrée à la définition de l'image dite « brute » obtenue avec les caméras mono-capteur. Ensuite, nous montrons les différents filtres permettant d'acquérir ces images brutes et les différentes approches proposées dans la littérature afin de les débruiter. La dernière partie est consacrée à la formalisation du processus du dématricage qui permet de transformer l'image brute délivrée par les caméras mono-capteur en une image couleur pleinement définie.

1 Représentation de la couleur

Il existe différents systèmes de représentation de la couleur qui peuvent être classés par famille : les systèmes trichromes, basés sur des couleurs primaires comme l'espace RGB , les systèmes basés sur la distinction entre luminance et chrominance comme le système (Y, C_b, C_r) ou encore les systèmes perceptuels comme le modèle $CIECAM02$ (pour plus de détails voir la thèse de Vandembroucke [69]).

Dans le système tri-chromatique (R, G, B) , la couleur est définie par trois composantes

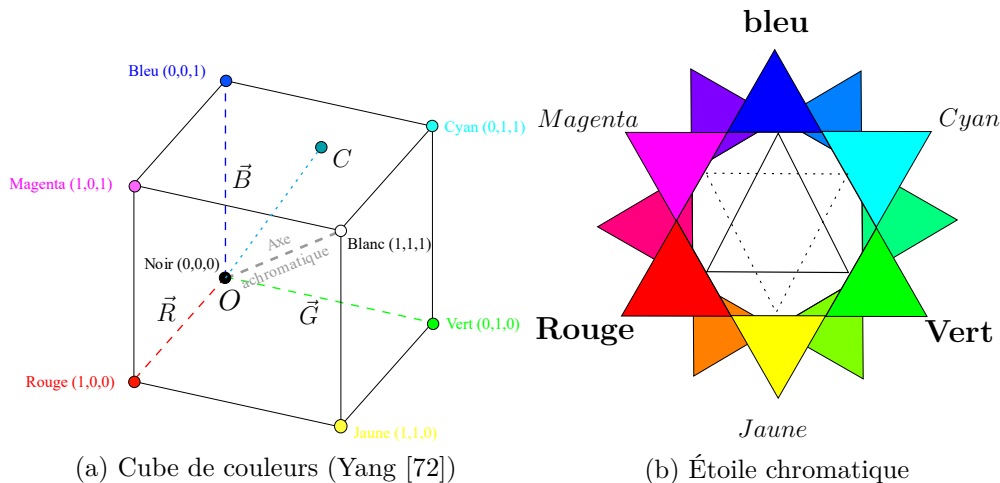


Figure 1.1 : Couleurs primaires, secondaires et intermédiaires.

couleur primaires proposées par la CIE (Commission Internationale de l'Éclairage) : rouge [R], verte [G] et bleue [B]. En se basant sur ces trois couleurs primaires, on peut obtenir une couleur secondaire, intermédiaire ou encore complémentaire selon les proportions des couleurs primaires utilisées. Une couleur est dite secondaire si elle est obtenue à proportion égale de deux couleurs primaires seulement, et intermédiaire si elle est obtenue par mélange en proportion inégale de couleurs primaires (*cf.* figure 1.1). Un couple de couleurs est dit complémentaire si leur mélange annule la perception de la couleur et produit un gris. Dans la figure 1.1a, la droite passant par les points Noir et Blanc et qui est appelée axe achromatique représente ce mélange de couleurs complémentaires. Dans l'étoile chromatique (*cf.* figure 1.1b), deux couleurs diamétralement opposées sont complémentaires.

Le système RGB est un espace vectoriel à trois dimensions avec \vec{R} , \vec{G} et \vec{B} comme vecteurs unitaires. Ce système peut être représenté sous forme d'un cube (*cf.* figure 1.1a). Chaque stimulus de couleur C peut être décrit par un vecteur \vec{OC} de coordonnées R , G et B :

$$\vec{OC} = R \cdot \vec{R} + G \cdot \vec{G} + B \cdot \vec{B} \quad (1.1)$$

L'origine O correspond au point couleur Noir (0,0,0), tandis que le point couleur Blanc (1,1,1) est défini par le mélange unitaire des trois primaires.

2 Acquisition d'une image couleur

Les caméras matricielles convertissent les stimuli de couleur de la scène observée en une image numérique couleur à l'aide de capteurs photosensibles. Deux principaux types

de technologies sont disponibles pour réaliser ces capteurs : la technologie CCD (*Charge Coupled Device*) et la technologie CMOS (*Complementary Metal Oxide Semiconductor*). La majorité des caméras grand public sont munies de capteurs à base de la technologie CMOS car ce type de capteurs a une faible consommation énergétique et permet une lecture très rapide des charges. Plus simples à fabriquer, les capteurs CCD sont en revanche utilisés dans les appareils privilégiant la qualité de l'image. Cette règle n'est pas stricte et tend à évoluer avec les progrès de chacune de ces technologies. Dans nos travaux, nous ne nous intéressons pas à la technologie de fabrication de ces capteurs mais à leur nombre dans une caméra. En effet, il existe deux grandes familles de caméras numériques couleur : celles comportant trois capteurs dites *tri-capteur* et les caméras qui ne contiennent qu'un seul capteur appelées *mono-capteur*.

2.1 Caméra tri-capteur

La couleur de chaque pixel d'une image numérique couleur est caractérisée par trois niveaux correspondant aux primaires rouge, verte et bleue. L'acquisition d'une telle image en technologie tri-capteur comporte donc trois capteurs (un par primaire).

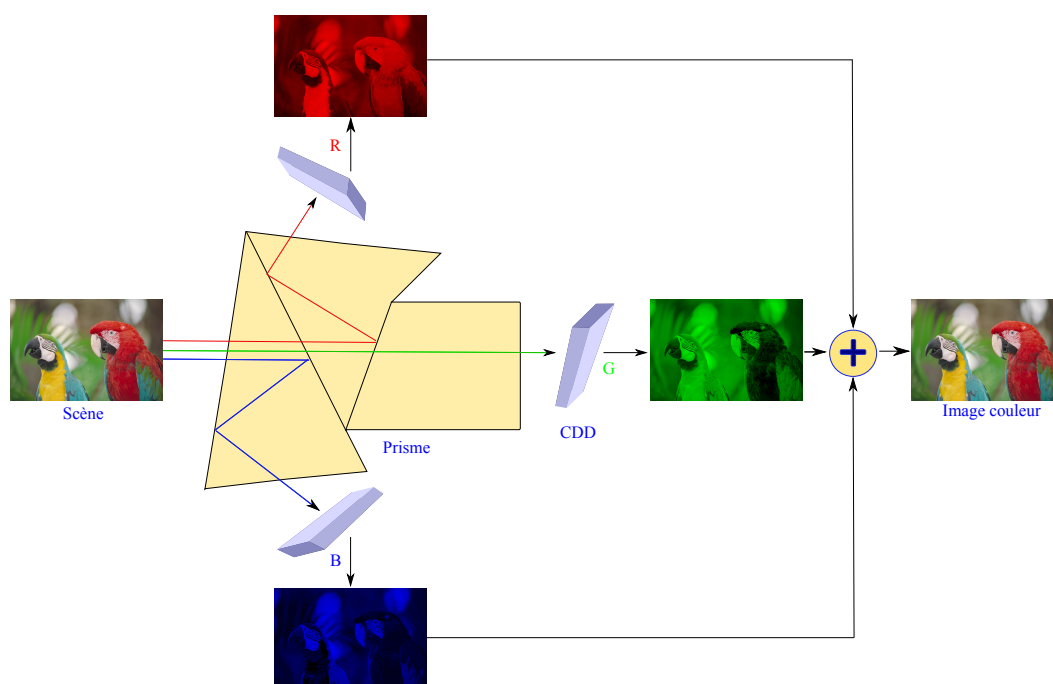


Figure 1.2 : Schéma simplifié de la technologie tri-capteur.

La figure 1.2 illustre le fonctionnement d'une caméra tri-capteur. Les stimuli de couleur de la scène sont séparés pour être projetés sur les trois capteurs photosensibles

par un jeu de prismes dichroïques dans lequel les longueurs d'ondes faibles (proches du bleu) et élevées (proche du rouge) subissent une réflexion interne totale. Les trois images de composante R , G et B sont acquises simultanément sur les trois capteurs. Après une recombinaison des trois images de composante, on obtient une image couleur.

Bien que cette technologie fournisse des images d'excellente qualité, elle n'est quasiment pas utilisée dans les appareils compacts car elle est encombrante et coûteuse, excluant le grand public de l'usage de ce type de caméras.

2.2 Caméra mono-capteur

Pour pallier les contraintes citées précédemment liées à l'utilisation des capteurs tri-capteur, des technologies utilisant un seul capteur ont été mises au point. Afin de rendre chaque photosite sensible à une bande de longueurs d'onde, une matrice de filtres couleur (*Color Filter Array*, CFA) est disposée sur la surface du capteur. La figure 1.3 illustre le fonctionnement d'une caméra mono-capteur dotée du CFA de Bayer.

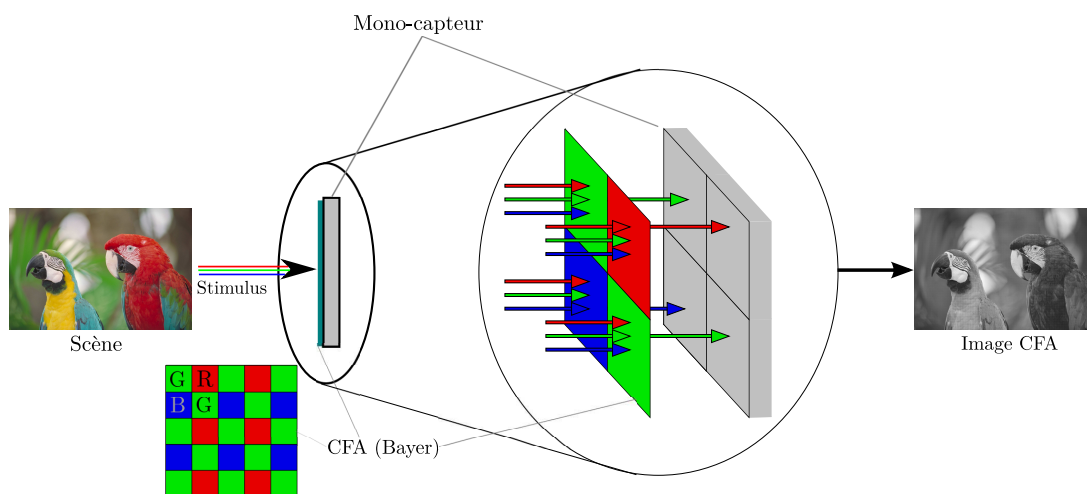


Figure 1.3 : : Schéma simplifié de la technologie mono-capteur.

Le CFA permet de multiplexer spatialement l'information chromatique, de façon à échantillonner une seule des trois composantes couleur Rouge (R), Vert (G) ou Bleu (B) en chaque photosite du capteur. En conséquence, une seule valeur de composante couleur est disponible en chaque pixel de l'image ainsi acquise. L'ensemble de ces pixels constitue l'image dite brute, ou encore image CFA.

3 Images CFA

3.1 Différentes configurations du CFA

Il existe dans la littérature plusieurs configurations de CFA mais celle de Bayer représentée sur la figure 1.4a est largement la plus répandue dans les caméras commercialisées et dans la littérature; c'est donc celle que nous utilisons dans nos travaux et que nous détaillons ci-après.

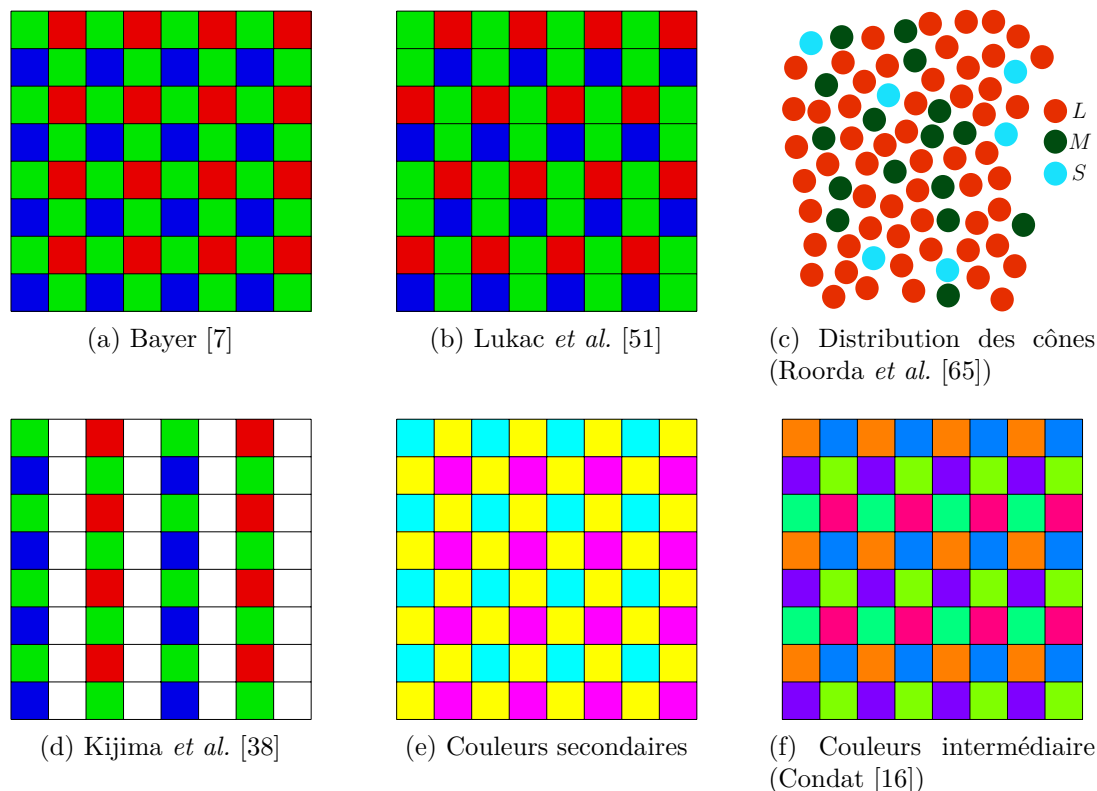


Figure 1.4 : : Différentes configurations du CFA.

Pour proposer ce filtre basé sur les couleurs primaires (R, G, B), Bayer a assimilé la composante verte à la luminance. À partir de cette hypothèse, Bayer a choisi de peupler son CFA de deux fois plus de filtres sensibles aux longueurs d'onde proches du vert. Un capteur doté du CFA de Bayer se compose donc schématiquement de 50% de photosites sensibles au vert, disposés en quinconce, de 25% de photosites sensibles au rouge et autant au bleu, disposés selon des motifs réguliers.

L'hypothèse énoncée par Bayer est basée sur le principe de fonctionnement de la rétine humaine. En effet, on trouve dans la rétine deux variétés de cellules photosensibles. Les bâtonnets sont spécialisés dans la vision nocturne car ils ne captent que des lumières de faible intensité et ne sont donc pas sensibles à la couleur. Les cônes, sensibles à la

couleur, sont répartis en trois types selon de leur sensibilité spectrale :

1. Les cônes S , qui sont sensibles à des longueurs d'onde courtes (*short*).
2. Les cônes M , qui sont sensibles à des longueurs d'onde moyennes (*middle*).
3. Les cônes L , qui sont sensibles à des longueurs d'onde longues (*long*).

Dartnall *et al.* [22] ont démontré que les maxima d'absorption de la lumière correspondent à $\lambda = 419 \text{ nm}$ pour les cônes S , $\lambda = 530,8 \text{ nm}$ pour les cônes M et $\lambda = 558,4 \text{ nm}$ pour les cônes L (*cf.* figure 1.5). Cela explique la représentation de la couleur par trois couleurs primaires (R,G,B).

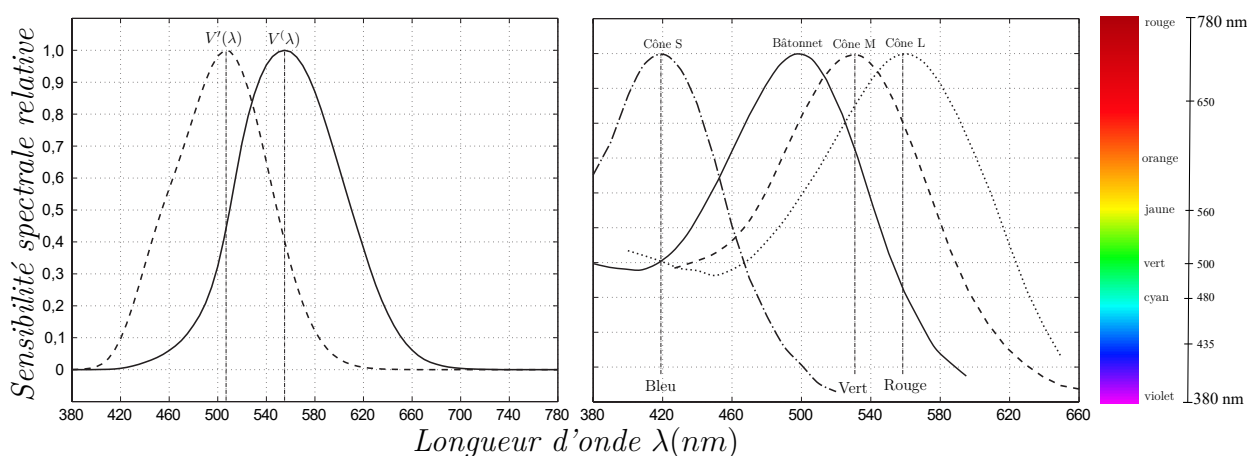


Figure 1.5 : : Fonction d'efficacité lumineuse normalisée en vision photopique $V(\lambda)$ et scotopique $V'(\lambda)$ (à gauche) et sensibilité spectrale relative des cônes L , M , et S et des bâtonnets (à droite) (sensibilités spectrales relatives de l'observateur standard) [68].

Ces cellules photosensibles ne sont pas réparties uniformément sur la rétine. Par ailleurs, les cônes S sont de loin les moins nombreux (moins de 10% de la population totale des cônes) (Curcio *et al.* [20]). Les cônes L sont les plus nombreux et le rapport L/M entre le nombre de cônes L et M change d'un individu à l'autre (*cf.* figure 1.4c) (Roorda *et al.* [65]). Les tests effectués par Trémeau *et al.* [68] démontrent que la fonction d'efficacité lumineuse $V(\lambda)$, qui représente la sensibilité lumineuse relative de l'œil à différentes longueurs d'onde du domaine du visible (*cf.* figure 1.5), est maximale pour la longueur d'onde $\lambda = 555 \text{ nm}$ qui correspond à la couleur verte. Les auteurs concluent ainsi que la rétine humaine est beaucoup plus sensible à la luminance qu'à la chrominance.

Lukac et Plataniotis [50] se sont inspirés du CFA de Bayer afin de proposer une autre mosaïque de filtres. Celle-ci est aussi composée des trois couleurs primaires (R,G,B) avec deux fois plus de filtres verts, mais avec un agencement différent de celui de Bayer (*cf.* figure 1.4b). La conclusion tirée par les auteurs est que la configuration du CFA influe

sur les résultats de dématricage¹(défini dans la section 4) sans pour autant déterminer la meilleure configuration (Lukac et Plataniotis [50]).

D'autres auteurs ont proposé des CFA à base de couleurs non primaires, notamment à base de couleurs secondaires cyan, magenta et jaune (*cf.* figure 1.4e) ou de couleurs intermédiaires comme celui de Condat [16]. Ce dernier CFA est composé d'un couple de couleurs complémentaires sur chaque ligne (*cf.* figure 1.4f). L'avantage de ces filtres est leur grande sensibilité à la lumière, ce qui permet de réduire le temps d'exposition (et donc le flou de bougé dû aux mouvements de l'appareil et/ou des objets de la scène). De plus, cette forte sensibilité spectrale permet d'utiliser une sensibilité ISO élevée, ce qui est très intéressant dans des environnement bruités (le lecteur pourra se référer à Condat [16], Wang [70]).

Le CFA proposé par Kijima *et al.* [38] (*cf.* figure 1.4d) a une sensibilité plus élevée que celle de Bayer à des stimuli de faible énergie. Cela est dû à la présence de photosites sensibles à toutes les longueurs d'onde du stimulus. Ce CFA rend le capteur beaucoup plus sensible à la luminance (plus de détails dans la thèse de Yang [72]).

Comme mentionné précédemment, le CFA de Bayer reste le plus utilisé et sert de base à une très grande majorité des travaux sur les images CFA.

3.2 Motifs locaux dans le CFA de Bayer

Quel que soit l'arrangement des premiers filtres du CFA de Bayer (*cf.* figure 1.4a), on y trouve 4 motifs locaux différents (*cf.* figure 1.6) : $\{GRG\}$, $\{GBG\}$, $\{RGR\}$ et $\{BGB\}$. Par exemple, le motif $\{GRG\}$ désigne un filtre central sensible au rouge et exprime que ses deux filtres voisins horizontaux sont sensibles au vert. Les motifs $\{GRG\}$ et $\{RGR\}$ sont respectivement similaires à $\{GBG\}$ et $\{BGB\}$ sauf que les composantes R et B sont permutées.

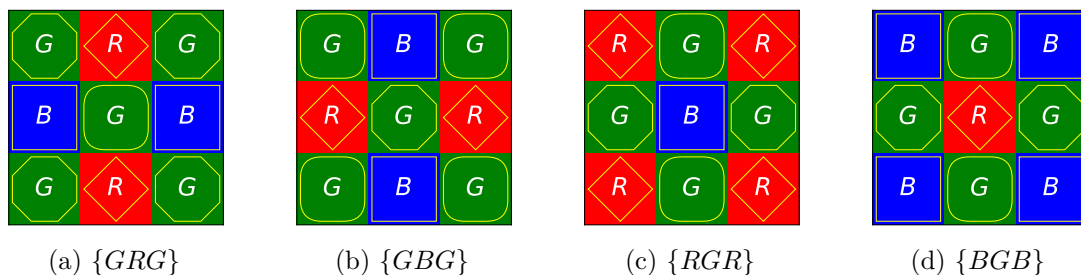


Figure 1.6 : Motifs locaux dans une image CFA. Les notations R , G et B indiquent que le niveau de la composante couleur respective est disponible en ce pixel.

1. Dématricage : estimation des deux composantes couleur manquantes en chaque pixel.

3.3 Description de l'image CFA

Nous appelons I^{CFA} l'image délivrée par une caméra mono-capteur recouverte d'un CFA de Bayer (*cf.* figure 1.4a). En conséquence, une seule valeur de composante couleur est disponible en chaque pixel de cette image. L'ensemble de tous les pixels de l'image, noté S , peut être divisé en $S = S^R \cup S^G \cup S^B$, où S^k désigne l'ensemble des pixels de l'image CFA dont la composante couleur $k \in \{R, G, B\}$ est disponible (*cf.* figures 1.7b–1.7d). Chacun des ensembles S^R et S^B contient un quart des pixels de l'image I^{CFA} et l'ensemble S^G contient la moitié des pixels de I^{CFA} . Ce dernier ensemble peut être divisé en $S^G = S^{G,R} \cup S^{G,B}$, où $S^{G,k}$, $k \in \{R, B\}$, est le sous-ensemble de pixels de S^G dont les voisins horizontaux appartiennent à S^k (*cf.* figures 1.7e et 1.7f).

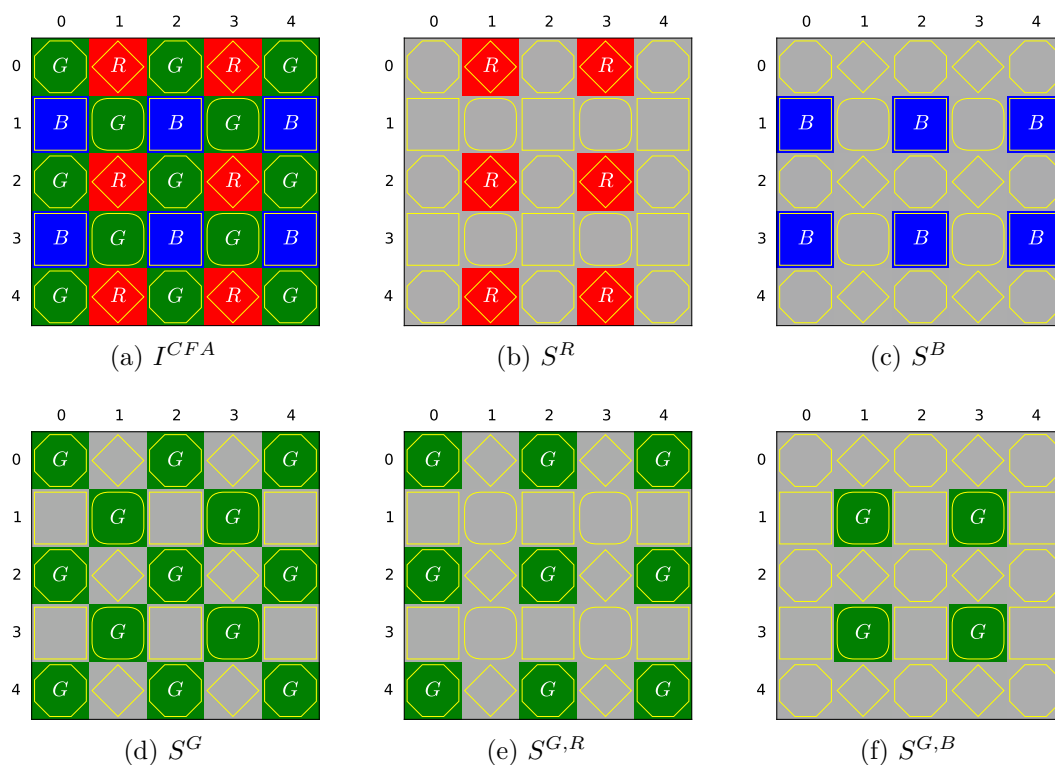


Figure 1.7 : Image CFA de Bayer (a) et sous-ensembles de pixels par composante couleur (b)–(f).

À partir de chaque ensemble S^k , $k \in \{R, G, B\}$, nous définissons une image « creuse » \check{I}^k qui contient les valeurs des pixels disponibles dans S^k et 0 ailleurs (*cf.* figure 1.7b–1.7d). Ces images creuses peuvent être définies par :

$$\check{I}^k = I^{CFA} \odot m^k, \quad (1.2)$$

où \odot désigne le produit élément par élément et m^k est la fonction d'échantillonnage :

$m^k(P) = 1$ si $P \in S^k$ et $m^k(P) = 0$ sinon.

3.4 Débruitage de l'image CFA

Comme toute image délivrée par un capteur CMOS ou CCD, l'image CFA est altérée par la présence de bruit. Zhang *et al.* [75] et Akiyama *et al.* [3] ont proposé un modèle de bruit spécifique aux images CFA. Elles sont entachées d'un bruit additif de type blanc gaussien, dont l'écart-type dépend de l'énergie (valeur moyenne) présente au sein des pixels de chaque sous-ensemble S^k , $k \in \{R, G, B\}$.

Les algorithmes de débruitage classiques conçus pour les images couleur ou pour les images en niveaux de gris ne peuvent pas être appliqués directement sur l'image CFA en raison de son treillis particulier sous-jacent à la mosaïque de filtres.

Une première approche de débruitage décompose l'image CFA en quatre sous-images obtenues en sous-échantillonnant I^{CFA} selon les sous-ensembles de pixels S^k , $k \in \{R, G, B\}$. Chaque sous-image est traitée séparément et les résultats sont réorganisés afin de reformer l'image CFA débruitée. Park *et al.* [59] proposent de convertir ces quatre sous-images dans l'espace YC_bC_r et d'appliquer ensuite les algorithmes classiques conçus pour les images en niveaux de gris. Ceci ne donne pas un résultat optimal car la corrélation spectrale n'est pas prise en compte (Ahirwar et Richariya [2], Lukac *et al.* [51]). Une autre approche est de former une image couleur de taille divisée par deux à partir de l'image CFA. Chaque bloc de 2×2 pixels de l'image CFA correspond à un pixel de l'image couleur construite. La composante verte est obtenue en moyennant les deux niveaux de cette composante disponibles dans le bloc CFA. Comme pour la première approche, débruiter cette image avec les algorithmes conçus initialement pour les images couleur n'est pas satisfaisant car l'information de haute fréquence présente dans la composante verte de l'image CFA n'est pas suffisamment bien conservée (Park *et al.* [59]).

Afin de prendre en compte la corrélation spectrale, Lee *et al.* [43] proposent d'utiliser les moyennes non locales (*Non-local Means*) qui réalisent une moyenne de la totalité des valeurs des blocs contenus dans l'image CFA, pondérées en fonction de leur similarité avec le bloc cible.

Zhang *et al.* [75] proposent une méthode de débruitage basée sur l'analyse en composantes principales (ACP) dédiée à l'image CFA. Les auteurs décomposent d'abord l'image CFA bruitée en une image de basses fréquences et une deuxième de hautes fréquences. Comme le bruit se trouve principalement dans les zones de hautes fréquences, les auteurs appliquent l'ACP seulement sur cette dernière image afin d'éviter l'apparition

d'artefacts dans les zones homogènes (zones de basses fréquences).

Danielyan *et al.* [21] utilisent le filtre Block Matching 3D (BM3D) dont ils modifient légèrement l'algorithme original développé pour les images en niveaux de gris afin que tous les blocs d'un groupe possèdent le même motif CFA.

Akiyama *et al.* [3] proposent une méthode qui utilise à la fois le filtre BM3D et l'ACP. Les auteurs décomposent l'image CFA en 4 sous-images comme dans la première approche citée au début de cette section, transforment chaque sous-image via l'ACP comme Zhang *et al.* [75], puis appliquent le filtre BM3D sur chacune des sous-images résultantes qui sont finalement réorganisées afin de reformer l'image CFA débruitée.

4 Formalisation du dématricage

Afin de formaliser le dématricage, comparons les processus d'acquisition d'une image couleur par une caméra tri-capteur et par une caméra mono-capteur (*cf.* figure 1.8). Soit \mathbf{I} une image couleur obtenue avec une caméra tri-capteur et I^k , $k \in \{R, G, B\}$, chacun des trois canaux couleur de \mathbf{I} . Dans le canal I^k , un pixel est caractérisé par le niveau de la composante couleur k , ce qui implique que chaque pixel de coordonnées (x, y) dans l'image \mathbf{I} est défini par $\mathbf{I}(x, y) = (I^R(x, y), I^G(x, y), I^B(x, y))$, où $I^R(x, y)$, $I^G(x, y)$ et $I^B(x, y)$ correspondent respectivement aux valeurs de composantes couleur rouge, verte et bleue.

Dans une caméra mono-capteur équipée d'un CFA de Bayer, à chaque pixel P de coordonnées spatiales (x, y) de l'image I^{CFA} est associée une seule composante couleur $I^R(P)$, $I^B(P)$ ou $I^G(P)$:

$$I^{CFA}(P) = \begin{cases} I^R(P) & \text{si } P \in S^R, \\ I^G(P) & \text{si } P \in S^G, \\ I^B(P) & \text{si } P \in S^B. \end{cases} \quad (1.3)$$

La procédure de dématricage notée D_{mos} , sert à passer de l'image I^{CFA} à l'image couleur estimée $\hat{\mathbf{I}}$, en estimant les valeurs de deux composantes couleur manquantes en chaque pixel $P \in I^{CFA}$ (*cf.* figure 1.9) :

$$I^{CFA}(x, y) \xrightarrow{D_{mos}} \hat{\mathbf{I}}(x, y) \quad (1.4)$$

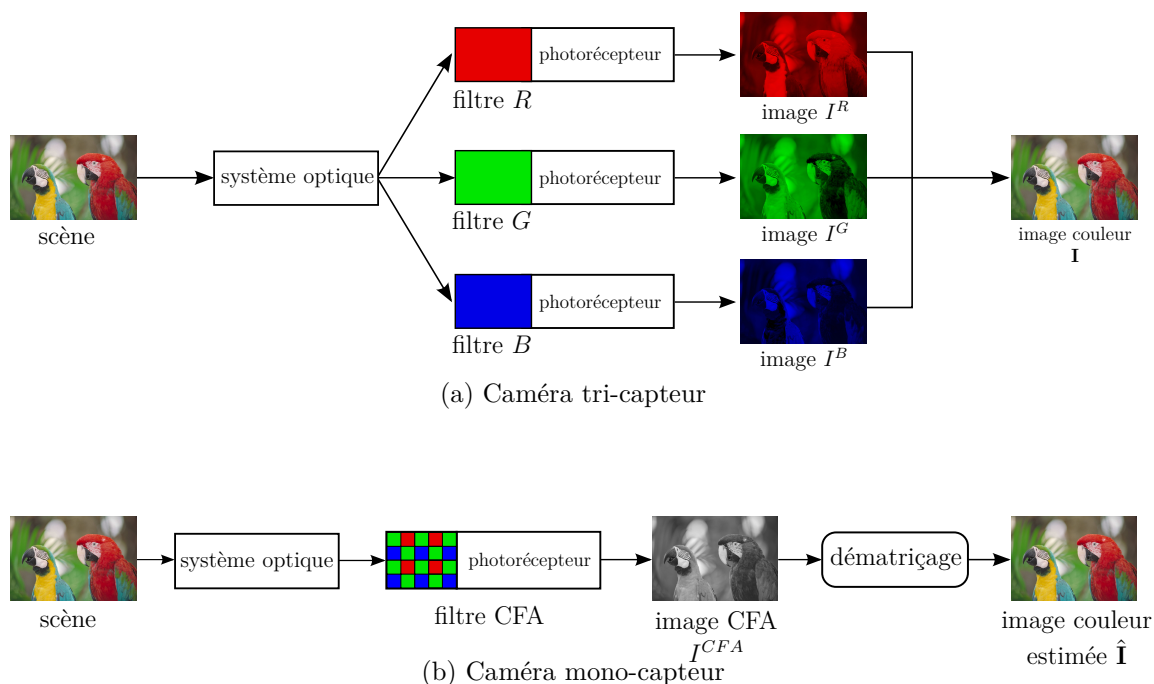


Figure 1.8 : : Acquisition d'une image couleur selon le type de caméra (schémas simplifiés).

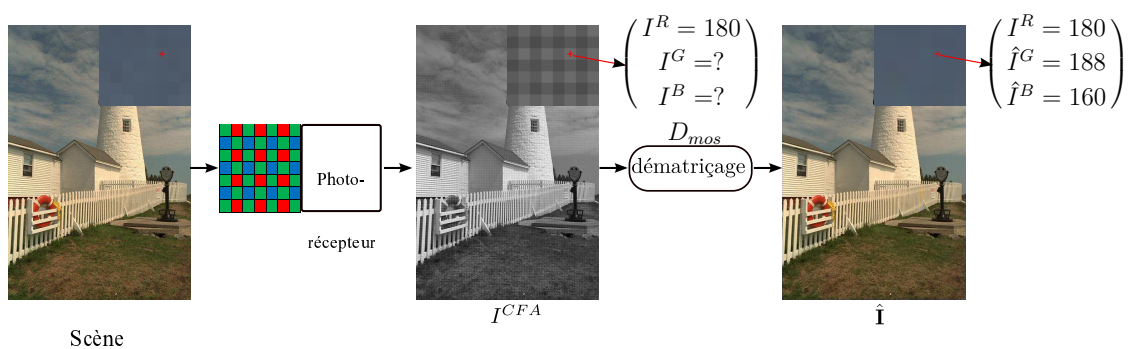


Figure 1.9 : : Principe du dématricage.

En chaque pixel $P(x,y) \in I^{CFA}$, deux composantes couleur sont manquantes et doivent être estimées afin de reconstruire l'image couleur \hat{I} . Pour cela le dématricage retient la composante couleur disponible dans l'image I^{CFA} et estime les deux compo-

santes couleur manquantes :

$$\hat{\mathbf{I}}(P) = \begin{cases} (I^{CFA}(P), \hat{I}^G(P), \hat{I}^B(P)) & \text{si } P \in S^R, \\ (\hat{I}^R(P), I^{CFA}(P), \hat{I}^B(P)) & \text{si } P \in S^G, \\ (\hat{I}^R(P), \hat{I}^G(P), I^{CFA}(P)) & \text{si } P \in S^B. \end{cases} \quad (1.5)$$

Chaque triplet de niveaux de composantes couleur dans l'équation (1.5) représente une couleur estimée. Sur les trois composantes de $\hat{\mathbf{I}}(P)$, celle notée par $I^{CFA}(P)$ est disponible au pixel $P \in I^{CFA}$, et les deux autres parmi $\hat{I}^R(P)$, $\hat{I}^G(P)$ et $\hat{I}^B(P)$ sont estimées par dématricage car elles sont indisponibles. Les procédures de dématricage seront décrites au sein du chapitre suivant.

Conclusion

Ce chapitre, dédié à l'acquisition d'une image numérique couleur, permet d'appréhender le reste de ce manuscrit. Ainsi, nous avons présenté les caméras tri-capteur et les caméras mono-capteur. Dans cette dernière technologie, l'unique capteur est recouvert d'une mosaïque de filtres chromatiques spectralement sélectifs afin de rendre les photosites sensibles à la couleur. L'image ainsi formée (dite « brute » ou CFA) ne comporte donc qu'une seule composante couleur par pixel. Dans la dernière partie de ce chapitre, nous avons vu et formalisé le processus de dématricage qui est nécessaire à la reconstruction de l'image couleur pleinement définie. Dans le chapitre suivant, nous allons examiner les différentes approches de dématricage proposées dans la littérature.

Dans nos travaux, nous ne nous intéressons pas à l'amélioration de la procédure de dématricage mais à l'analyse de l'image CFA pour des traitements bas niveau. Un état de l'art sur les différentes approches exploitant directement l'image CFA sera également fait dans le chapitre suivant. Ces travaux précurseurs nous ont incité à explorer dans quelle mesure d'autres méthodes d'analyse, conçues pour les images couleur estimées, peuvent être transposées à l'image CFA.

Chapitre 2

Dématriçage

1 Introduction

Le dématriçage est une étape clé dans le processus d'obtention d'une image couleur avec une caméra mono-capteur. Ce traitement a une très grande importance sur la qualité de cette image car il estime les niveaux de deux composantes couleur sur trois en chaque pixel de l'image CFA. Depuis la généralisation de l'utilisation des appareils numériques mono-capteur, de nombreux algorithmes de dématriçage ont été mis au point, les plus récents étant souvent des améliorations des principes de base.

Dans le présent chapitre, nous regroupons ces algorithmes de dématriçage par famille à commencer par la plus ancienne et la plus simple qui procède par interpolation bilinéaire. La deuxième famille est basée sur la corrélation spectrale entre les composantes couleur. Elle s'appuie sur les travaux de Gunturk *et al.* [29] qui ont montré que les trois composantes couleur sont fortement corrélées.

Ensuite, nous détaillons les différentes approches qui exploitent la corrélation spatiale dans le plan image. La corrélation spatiale est exploitée par la différence de composantes ou l'estimation de résidus. En effet, au sein d'une zone de l'image CFA représentant une surface homogène en couleur, les niveaux de chaque composante couleur sont localement corrélés. L'idée principale est d'estimer la direction de la transition entre ces zones homogènes.

La plupart des algorithmes de dématriçage récents couplent la corrélation spatiale avec la corrélation spectrale. Alleysson *et al.* [4, 5] ont été les premiers à utiliser le domaine fréquentiel pour réaliser le dématriçage. Le principe de leur méthode est de représenter une image CFA comme une combinaison d'une composante de luminance en basses fréquences spatiales et de deux composantes de chrominance modulées en hautes fréquences spatiales, puis d'estimer l'image dématriçée en sélectionnant les fréquences

de manière adéquate.

Enfin, nous présentons succinctement les approches de débruitage-dématriçage. En effet, comme l'image CFA peut être bruitée, certains auteurs utilisent conjointement un dématriçage et un débruitage.

Les méthodes de dématriçage peuvent générer des artéfacts d'estimation dans les zones présentant des informations de hautes fréquences, telles que les textures et les contours. Ces artéfacts peuvent alors altérer les performances des méthodes d'analyse automatique des images dématriçées qui s'appuient sur ces informations de haute fréquence. Nous allons donc décrire les approches qui évitent le dématriçage et analysent directement l'image CFA pour exploiter les informations de hautes fréquences. Nous nous focalisons sur trois types de problèmes à savoir la caractérisation des textures (*cf.* section 3.1), la compression (*cf.* section 3.2) et la détection des contours (*cf.* section 3.3).

2 Méthodes de dématriçage

2.1 Méthode par interpolation bilinéaire

En raison de la structure particulière de la mosaïque du CFA de Bayer, les structures de voisinage utilisées dans une image CFA sont complètement différentes de celles utilisées classiquement dans une image pleinement définie. Nous jugeons utile de définir les différents voisinages utilisés dans une image CFA. En effet, un voisinage dans une image CFA est pris de telle sorte à considérer toujours la même composante couleur. Quel que soit le motif local de l'image CFA, on trouve 4 voisinages d'un pixel P dans une image CFA de Bayer (*cf.* figure 2.1) : $\mathbb{N}^H(P)$, $\mathbb{N}^V(P)$, $\mathbb{N}^+(P)$ et $\mathbb{N}^\times(P)$. L'exposant H (ou V) indique que les deux pixels voisins horizontaux (ou verticaux) présentent la même composante couleur ; on parle d'un voisinage horizontal (ou vertical) (*cf.* figures 2.1a et 2.1b). L'exposant $+$ (ou \times) indique que les quatre pixels voisins horizontaux et verticaux (ou diagonaux) présentent la même composante couleur (*cf.* figures 2.1c et 2.1d).

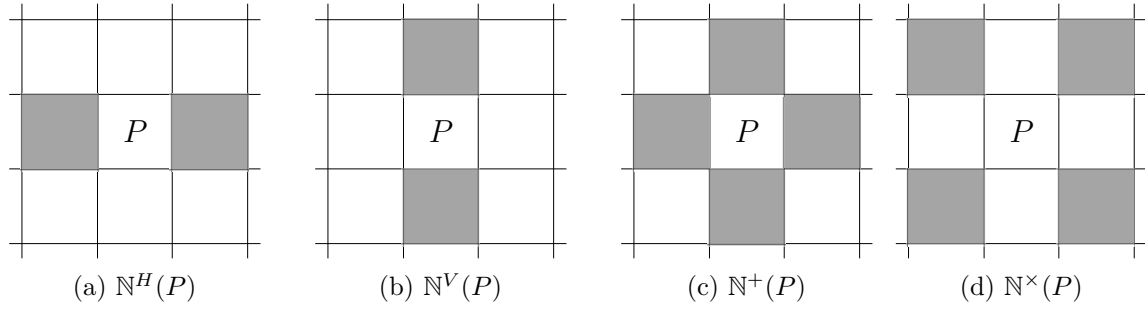


Figure 2.1 : : Voisinage d'un pixel P dans une image CFA.

Comme dans la grande majorité des algorithmes de dématricage, la méthode par interpolation bilinéaire retient la composante couleur disponible en chaque pixel P dans I^{CFA} et estime les deux composantes couleur manquantes (*cf.* équation (2.1)). Une interpolation linéaire dans chaque direction dans l'image I^{CFA} est réalisée afin d'estimer les deux composantes couleur manquantes en chaque pixel $P \in S$:

$$\hat{\mathbf{I}}(P) = \begin{cases} \left(I^{CFA}(P), \frac{1}{4} \sum_{Q \in N^+(P)} I^{CFA}(Q), \frac{1}{4} \sum_{Q \in N^x(P)} I^{CFA}(Q) \right) & \text{si } P \in S^R, \\ \left(\frac{1}{2} \sum_{Q \in N^H(P)} I^{CFA}(Q), I^{CFA}(P), \frac{1}{2} \sum_{Q \in N^V(P)} I^{CFA}(Q) \right) & \text{si } P \in S^G, \\ \left(\frac{1}{4} \sum_{Q \in N^x(P)} I^{CFA}(Q), \frac{1}{4} \sum_{Q \in N^+(P)} I^{CFA}(Q), I^{CFA}(P) \right) & \text{si } P \in S^B, \end{cases} \quad (2.1)$$

Cette méthode de dématricage est très peu coûteuse en temps de calcul mais génère essentiellement deux types d'artéfacts : l'effet de fermeture éclair et les fausses couleurs (plus de détails dans la thèse de Yang [72]).

2.2 Méthodes utilisant la corrélation spectrale

De nombreuses méthodes de dématricage exploitent la corrélation entre les trois composantes couleur en chaque pixel d'une image couleur naturelle. Gunturk *et al.* [29] ont en effet démontré cette corrélation spectrale en décomposant chaque plan de composante couleur en quatre sous-bandes à l'aide d'un filtre bidimensionnel passe-bande obtenu par combinaison de deux filtres passe-haut $h_0 = [1 \ -2 \ 1]/4$ et passe-bas $h_1 = [1 \ 2 \ 1]/4$. Les 4 sous-bandes sont obtenus de la manière suivante : (LL) filtrage passe-bas en ligne et en colonne, (HH) filtrage passe-haut en ligne et en colonne, (LH) filtrage passe-bas en ligne et passe-haut en colonne et (HL) filtrage passe-haut en ligne et passe-bas en colonne.

Ensuite, les auteurs calculent le coefficient de corrélation pour chaque couple de

composantes couleur $(k,l) \in \{R,G,B\}^2$, $k \neq l$, et ce, pour chaque sous-bande d'une image de taille $M \times N$ pixels :

$$C^{k,l} = \frac{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} (I^k(x,y) - \mu^k) (I^l(x,y) - \mu^l)}{\sqrt{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} (I^k(x,y) - \mu^k)^2} \sqrt{\sum_{x=0}^{N-1} \sum_{y=0}^{M-1} (I^l(x,y) - \mu^l)^2}}, \quad (2.2)$$

où $I^k(x,y)$ et $I^l(x,y)$ représentent le niveau du pixel de coordonnées (x,y) dans la même sous-bande, μ^k et μ^l sont les moyennes des niveaux des canaux couleur I^k et I^l sur ces mêmes sous-bandes. Ces coefficients de corrélation sont très élevés dans toutes les sous-bandes, particulièrement celles contenant au moins une partie de hautes fréquences ((HL), (LH) et (HH)). Cette corrélation spectrale intra-pixel est utilisée par plusieurs auteurs afin d'estimer les composantes couleur manquantes en chaque pixel $P \in I^{CFA}$.

2.2.1 Interpolation des rapports et différences de composantes

Cok [15] a été le premier à exploiter la corrélation spectrale entre composantes couleur pour le dématriçage. Il s'est basé sur l'hypothèse que la teinte approchée par les rapports des composantes $\frac{I^R}{I^G}$ et $\frac{I^B}{I^G}$ varie très peu localement. Premièrement, l'auteur estime les valeurs manquantes de la composante G par interpolation bilinéaire (*cf.* équation (2.1)) pour obtenir le canal estimé \hat{I}^G pleinement défini. Ensuite, les valeurs manquantes des composantes R et B sont estimées comme suit :

$$\hat{\mathbf{I}}(P) = \begin{cases} \left(I^{CFA}(P), \hat{I}^G(P), \frac{1}{4}\hat{I}^G(P) \sum_{Q \in \mathbb{N}^{\times}(P)} \frac{I^B(Q)}{I^G(Q)} \right) & \text{si } P \in S^R, \\ \left(\frac{1}{2}I^G(P) \sum_{Q \in \mathbb{N}^H(P)} \frac{I^R(Q)}{I^G(Q)}, I^{CFA}(P), \frac{1}{2}I^G(P) \sum_{Q \in \mathbb{N}^V(P)} \frac{I^B(Q)}{I^G(Q)} \right) & \text{si } P \in S^G, \\ \left(\frac{1}{4}\hat{I}^G(P) \sum_{Q \in \mathbb{N}^{\times}(P)} \frac{I^R(Q)}{I^G(Q)}, \hat{I}^G(P), I^{CFA}(P) \right) & \text{si } P \in S^B. \end{cases} \quad (2.3)$$

Une autre façon d'exploiter la corrélation spectrale est de considérer que les différences (et non plus les rapports) des composantes sont localement constantes. Après avoir estimé \hat{I}^G par interpolation bilinéaire, Cok, Pei et Tam [60] proposent d'interpoler bilinéairement les différences de composantes $(I^k - I^G)$, $k \in \{R,B\}$, afin d'estimer les niveaux des composantes R et B manquantes (*cf.* figure 2.2b) :

$$\hat{\mathbf{I}}(P) = \begin{cases} \left(I^{CFA}(P), \hat{I}^G(P), \hat{I}^G(P) + \frac{1}{4} \sum_{Q \in \mathbb{N}^{\times}(P)} (I^B(Q) - \hat{I}^G(Q)) \right) & \text{si } P \in S^R, \\ \left(\hat{I}^R(P), I^{CFA}(P), \hat{I}^B(P) \right) & \text{si } P \in S^G, \\ \left(\hat{I}^G(P) + \frac{1}{4} \sum_{Q \in \mathbb{N}^{\times}(P)} (I^R(Q) - \hat{I}^G(Q)), \hat{I}^G(P), I^{CFA}(P) \right) & \text{si } P \in S^B, \end{cases} \quad (2.4)$$

avec :

$$\hat{I}^k(P) = I^G(P) + \frac{1}{2} \sum_{Q \in \mathbb{N}^H(P)} (I^k(Q) - I^G(Q)) \quad , k \in \{R, B\}.$$

Dans ses travaux de thèse, Yang [72] a évalué ces deux hypothèses utilisant la corrélation spectrale et en a conclu que la différence des composantes donne en général de meilleurs résultats. Dans la littérature, c'est la stratégie la plus souvent utilisée pour exploiter la corrélation spectrale dans les méthodes de dématricage.

2.2.2 Interpolation des résidus

Kiku *et al.* [39, 40] proposent autre façon d'exploiter la corrélation spectrale. Au lieu d'interpoler la différence des composantes, les auteurs [39] proposent d'estimer provisoirement les valeurs de la composante R et B manquantes en utilisant la composante G préalablement estimée avec l'approche de Pekkuksen et Altunbasak [61] (*cf.* section 2.3.2) comme image guide à l'instar de Monno *et al.* [55]. Ensuite, Kiku *et al.* [39] calculent les résidus qui sont les différences entre les valeurs de la composante R (ou B) disponibles et les valeurs de la même composante estimées provisoirement. Une interpolation bilinéaire est réalisée sur cette image des résidus afin d'estimer les valeurs manquantes. Les auteurs utilisent l'interpolation des résidus car le domaine des résidus contient moins de hautes fréquences que le domaine des différences de composantes, ce qui y rend l'interpolation bilinéaire plus performante (Ye et Ma [73]). Finalement, l'image de la composante R (ou B) estimée provisoirement est ajoutée à celle des résidus interpolée afin d'obtenir l'image de composante estimée \hat{I}^R (ou \hat{I}^B) (*cf.* figure 2.2a). Kiku *et al.* [40] améliorent cette méthode en réalisant aussi l'interpolation des résidus sur la composante G . La figure 2.2 montre la différence entre l'interpolation des résidus (*Residual Interpolation*, RI) et l'interpolation des différences de composantes couleur.

Quelle que soit la méthode d'interpolation choisie pour les composantes R et B , la qualité de l'image couleur dématricée $\hat{\mathbf{I}}$ dépend grandement de celle de l'estimation préalable du canal \hat{I}^G . Coupler l'utilisation d'un filtre guidé avec l'interpolation résiduelle

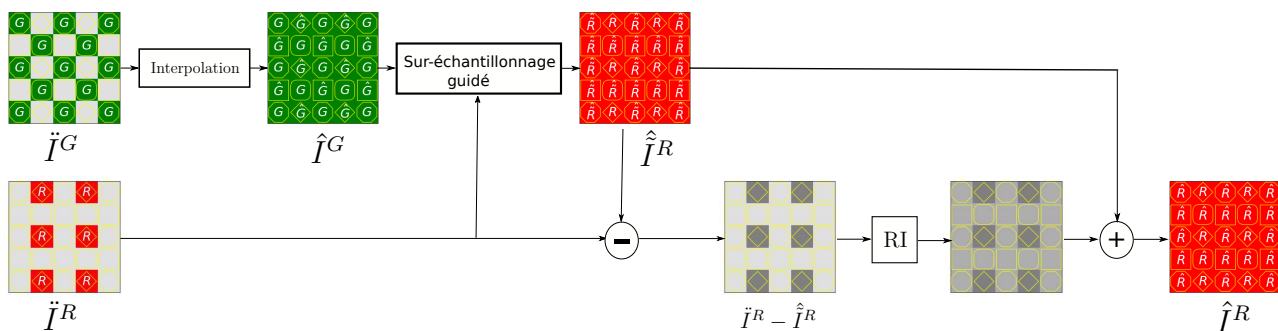
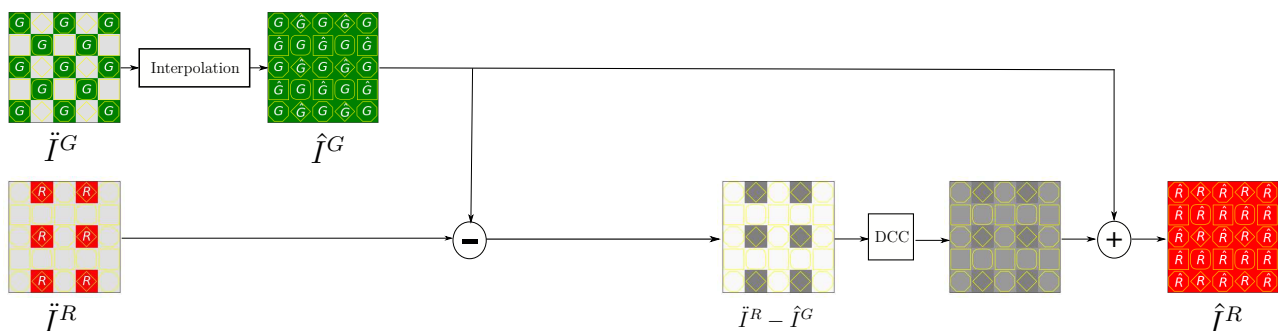
(a) Calcul de \hat{I}^R par interpolation des résidus (RI) (Kiku *et al.* [39]).(b) Calcul de \hat{I}^R par interpolation des différences de composantes (DCC).

Figure 2.2 : Comparaison de l'estimation de la composante R dans la méthode d'interpolation des résidus proposée par Kiku *et al.* [39] et la méthode d'interpolation des différences de composantes couleur.

comme proposé par Kiku *et al.* [39, 40] donne une image couleur dématricée $\hat{\mathbf{I}}$ de très bonne qualité.

2.3 Méthodes basées sur une analyse spatiale

On peut supposer qu'au sein d'une zone de l'image CFA représentant une surface homogène, les niveaux de chaque composante couleur sont localement corrélés. Plusieurs auteurs ont exploité cette corrélation spatiale afin d'estimer les valeurs manquantes de la composante G . Systématiquement, \hat{I}^G est estimé avant \hat{I}^R et \hat{I}^B car l'ensemble des pixels S^G comporte deux fois plus de pixels que S^R et S^B . L'idée est d'estimer la composante G en tenant compte des seuls pixels voisins appartenant à la même zone homogène. Quand le voisinage d'un pixel est situé sur une zone de transition, on estime ses valeurs de composantes manquantes le long de la transition et non pas à travers elle. Afin d'exploiter la corrélation spatiale, il faut estimer la direction de la transition entre zones homogènes. Nous allons ici détailler plusieurs méthodes d'estimation de cette direction de transition proposées dans la littérature.

2.3.1 Interpolation directionnelle par sélection de direction

Pour estimer les niveaux manquants de la composante G , Hibbard [36] a proposé de calculer la norme d'un « gradient » dans un voisinage 3×3 afin d'identifier la direction de transition entre zones homogènes. Les « gradients » horizontal et vertical sont calculés en chaque pixel $P(x,y) \in S^R \cup S^B$:

$$\Delta^H(x,y) = \left| I^{CFA}(x-1,y) - I^{CFA}(x+1,y) \right| \quad (2.5)$$

$$\Delta^V(x,y) = \left| I^{CFA}(x,y-1) - I^{CFA}(x,y+1) \right|. \quad (2.6)$$

La direction d'interpolation qui présente le « gradient » le plus faible est privilégiée :

— Si $\Delta^H(x,y) > \Delta^V(x,y)$, l'interpolation est verticale :

$$\hat{I}^G(x,y) = \frac{1}{2} \left(I^{CFA}(x,y-1) + I^{CFA}(x,y+1) \right). \quad (2.7)$$

— Si $\Delta^H(x,y) < \Delta^V(x,y)$, l'interpolation est horizontale :

$$\hat{I}^G(x,y) = \frac{1}{2} \left(I^{CFA}(x-1,y) + I^{CFA}(x+1,y) \right). \quad (2.8)$$

— Si $\Delta^H(x,y) = \Delta^V(x,y)$, l'interpolation de la composante verte est bilinéaire (*cf.* équation (2.1)).

Laroche et Prescott [42] proposent une amélioration de la méthode de Hibbard [36] qui estime plus précisément le gradient en tenant compte des niveaux disponibles dans un voisinage 5×5 de $P(x,y) \in S^R \cup S^B$:

$$\Delta^H(x,y) = \left| 2I^{CFA}(x,y) - I^{CFA}(x-2,y) - I^{CFA}(x+2,y) \right| \quad (2.9)$$

$$\Delta^V(x,y) = \left| 2I^{CFA}(x,y) - I^{CFA}(x,y-2) - I^{CFA}(x,y+2) \right|. \quad (2.10)$$

Hamilton et Adams [30] proposent de combiner les deux approches citées précédemment afin de mieux estimer la direction d'interpolation pour estimer le canal \hat{I}^G . Les auteurs calculent pour cela la différence absolue de la composante G de deux voisins immédiats, ainsi que celle du R (ou B) dont les échantillons sont distants de deux pixels de $P(x,y) \in S^R \cup S^B$:

$$\Delta^H(x,y) = \left| I^{CFA}(x-1,y) - I^{CFA}(x+1,y) \right| + \left| 2I^{CFA}(x,y) - I^{CFA}(x-2,y) - I^{CFA}(x+2,y) \right|, \quad (2.11)$$

$$\Delta^V(x,y) = \left| I^{CFA}(x,y-1) - I^{CFA}(x,y+1) \right| + \left| 2I^{CFA}(x,y) - I^{CFA}(x,y-2) - I^{CFA}(x,y+2) \right|. \quad (2.12)$$

La direction d'interpolation qui présente le « gradient » le plus faible est privilégiée :

— Si $\Delta^H(x,y) > \Delta^V(x,y)$, l'interpolation est verticale :

$$\begin{aligned} \hat{I}^G(x,y) = & \frac{1}{2} \left(I^{CFA}(x,y-1) + I^{CFA}(x,y+1) \right) \\ & + \frac{1}{4} \left(2I^{CFA}(x,y) - I^{CFA}(x,y-2) - I^{CFA}(x,y+2) \right). \end{aligned} \quad (2.13)$$

— Si $\Delta^H(x,y) < \Delta^V(x,y)$, l'interpolation est horizontale :

$$\begin{aligned} \hat{I}^G(x,y) = & \frac{1}{2} \left(I^{CFA}(x-1,y) + I^{CFA}(x+1,y) \right) \\ & + \frac{1}{4} \left(2I^{CFA}(x,y) - I^{CFA}(x-2,y) - I^{CFA}(x+2,y) \right). \end{aligned} \quad (2.14)$$

— Si $\Delta^H(x,y) = \Delta^V(x,y)$, l'interpolation est bilinéaire :

$$\begin{aligned} \hat{I}^G(x,y) = & \frac{1}{4} \sum_{Q \in \mathbb{N}^+(P)} I^{CFA}(Q) + \frac{1}{8} \left(4I^{CFA}(x,y) - I^{CFA}(x,y-2) \right. \\ & \left. - I^{CFA}(x-2,y) - I^{CFA}(x+2,y) - I^{CFA}(x,y+2) \right). \end{aligned} \quad (2.15)$$

L'estimation du canal vert proposée par Hamilton et Adams [30] est bien meilleure que celles proposées par Hibbard [36] et Laroche et Prescott [42] car elle estime les gradients de manière plus précise en exploitant conjointement les corrélations spatiale et spectrale.

Toutes ces méthodes estiment les canaux R et B par interpolation des différences de composantes (*cf.* section 2.2.1) en se basant sur le canal G pré-estimé, puis estiment les canaux R et B par interpolation.

2.3.2 Interpolation directionnelle pondérée

Pekkucuksen et Altunbasak [62] proposent une méthode de dématriçage qualifiée de « multi-échelle » en référence à la manière de calculer les gradients (*Multiscale Gradients*). L'idée est d'utiliser à la fois l'hypothèse de constance locale des différences de composantes et les estimations directionnelles horizontale et verticale.

Les auteurs tiennent compte du fait que l'image CFA se compose de deux types de lignes et de colonnes différentes : celles comportant des pixels appartenant à $S^R \cup S^G$ et celles présentant des pixels de $S^B \cup S^G$.

En chaque pixel d'une ligne ou colonne présentant une alternance de R et G (on procède de la même façon pour une alternance de B et G) les étapes suivantes sont

exécutées :

1. Les auteurs utilisent la méthode proposée par Hamilton et Adams [30] pour calculer deux estimations, l'une horizontale (*cf.* équation (2.14)) et l'autre verticale (*cf.* équation (2.13)) en chaque pixel. En $P \in S^{G,R}$, cela fournit les estimations $\hat{I}^{R,H}$ et $\hat{I}^{R,V}$ et en $P \in S^R$, cela fournit les estimations $\hat{I}^{G,H}$ et $\hat{I}^{G,V}$. À l'aide de ces estimations, les auteurs calculent les différences directionnelles suivantes :

$$\tilde{\delta}^{R-G,H}(P) = \begin{cases} I^G(P) - \hat{I}^{R,H}(P) & \text{si } P \in S^G, \\ \hat{I}^{G,H}(P) - I^R(P) & \text{si } P \in S^R, \end{cases} \quad (2.16)$$

$$\tilde{\delta}^{R-G,V}(P) = \begin{cases} I^G(P) - \hat{I}^{R,V}(P) & \text{si } P \in S^G, \\ \hat{I}^{G,V}(P) - I^R(P) & \text{si } P \in S^R. \end{cases} \quad (2.17)$$

On obtient ainsi deux cartes de différences de composantes, l'une horizontale contenant $\tilde{\delta}^{R-G,H}$ et $\tilde{\delta}^{B-G,H}$ et l'autre verticale contenant $\tilde{\delta}^{R-G,V}$ et $\tilde{\delta}^{B-G,V}$. La figure 2.3 illustre ces différences verticale et horizontale sous forme de cartes.

| | | | | | | | | | | | |
|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| $\tilde{\delta}^{R-G,H}$ | $\tilde{\delta}^{R-G,H}$ | $\tilde{\delta}^{R-G,H}$ | $\tilde{\delta}^{R-G,H}$ | $\tilde{\delta}^{R-G,H}$ | $\tilde{\delta}^{R-G,H}$ | $\tilde{\delta}^{B-G,V}$ | $\tilde{\delta}^{R-G,V}$ | $\tilde{\delta}^{B-G,V}$ | $\tilde{\delta}^{R-G,V}$ | $\tilde{\delta}^{B-G,V}$ | $\tilde{\delta}^{R-G,V}$ |
| $\tilde{\delta}^{B-G,H}$ | $\tilde{\delta}^{B-G,H}$ | $\tilde{\delta}^{B-G,H}$ | $\tilde{\delta}^{B-G,H}$ | $\tilde{\delta}^{B-G,H}$ | $\tilde{\delta}^{B-G,H}$ | $\tilde{\delta}^{B-G,V}$ | $\tilde{\delta}^{R-G,V}$ | $\tilde{\delta}^{B-G,V}$ | $\tilde{\delta}^{R-G,V}$ | $\tilde{\delta}^{B-G,V}$ | $\tilde{\delta}^{R-G,V}$ |
| $\tilde{\delta}^{R-G,H}$ | $\tilde{\delta}^{R-G,H}$ | $\tilde{\delta}^{R-G,H}$ | $\tilde{\delta}^{R-G,H}$ | $\tilde{\delta}^{R-G,H}$ | $\tilde{\delta}^{R-G,H}$ | $\tilde{\delta}^{B-G,V}$ | $\tilde{\delta}^{R-G,V}$ | $\tilde{\delta}^{B-G,V}$ | $\tilde{\delta}^{R-G,V}$ | $\tilde{\delta}^{B-G,V}$ | $\tilde{\delta}^{R-G,V}$ |
| $\tilde{\delta}^{B-G,H}$ | $\tilde{\delta}^{B-G,H}$ | $\tilde{\delta}^{B-G,H}$ | $\tilde{\delta}^{B-G,H}$ | $\tilde{\delta}^{B-G,H}$ | $\tilde{\delta}^{B-G,H}$ | $\tilde{\delta}^{B-G,V}$ | $\tilde{\delta}^{R-G,V}$ | $\tilde{\delta}^{B-G,V}$ | $\tilde{\delta}^{R-G,V}$ | $\tilde{\delta}^{B-G,V}$ | $\tilde{\delta}^{R-G,V}$ |
| $\tilde{\delta}^{R-G,H}$ | $\tilde{\delta}^{R-G,H}$ | $\tilde{\delta}^{R-G,H}$ | $\tilde{\delta}^{R-G,H}$ | $\tilde{\delta}^{R-G,H}$ | $\tilde{\delta}^{R-G,H}$ | $\tilde{\delta}^{B-G,V}$ | $\tilde{\delta}^{R-G,V}$ | $\tilde{\delta}^{B-G,V}$ | $\tilde{\delta}^{R-G,V}$ | $\tilde{\delta}^{B-G,V}$ | $\tilde{\delta}^{R-G,V}$ |
| $\tilde{\delta}^{B-G,H}$ | $\tilde{\delta}^{B-G,H}$ | $\tilde{\delta}^{B-G,H}$ | $\tilde{\delta}^{B-G,H}$ | $\tilde{\delta}^{B-G,H}$ | $\tilde{\delta}^{B-G,H}$ | $\tilde{\delta}^{B-G,V}$ | $\tilde{\delta}^{R-G,V}$ | $\tilde{\delta}^{B-G,V}$ | $\tilde{\delta}^{R-G,V}$ | $\tilde{\delta}^{B-G,V}$ | $\tilde{\delta}^{R-G,V}$ |
| $\tilde{\delta}^{R-G,H}$ | $\tilde{\delta}^{R-G,H}$ | $\tilde{\delta}^{R-G,H}$ | $\tilde{\delta}^{R-G,H}$ | $\tilde{\delta}^{R-G,H}$ | $\tilde{\delta}^{R-G,H}$ | $\tilde{\delta}^{B-G,V}$ | $\tilde{\delta}^{R-G,V}$ | $\tilde{\delta}^{B-G,V}$ | $\tilde{\delta}^{R-G,V}$ | $\tilde{\delta}^{B-G,V}$ | $\tilde{\delta}^{R-G,V}$ |

Figure 2.3 : : Cartes des différences de composantes horizontale et verticale [62].

2. À partir des gradients directionnels multi-échelle $d^H(x,y)$ et $d^V(x,y)$ (eux-mêmes calculés à partir des cartes des différences de composantes horizontale et verticale (*cf.* figure 2.3)), les auteurs calculent des poids directionnels (w^N, w^S, w^E, w^W) dans une fenêtre locale horizontale de 5×3 pixels et verticale de 3×5 pixels, ainsi que les poids des directions horizontale et verticale (w^H, w^V) dans une fenêtre de

5 × 5 pixels :

$$\begin{aligned}
w^N(x,y) &= 1 / \left(\sum_{i=x-1}^{x+1} \sum_{j=y-4}^y d^V(i,j) \right)^2, & w^S(x,y) &= 1 / \left(\sum_{i=x-1}^{x+1} \sum_{j=y}^{y+4} d^V(i,j) \right)^2 \\
w^E(x,y) &= 1 / \left(\sum_{i=x-4}^x \sum_{j=y-1}^{y+1} d^H(i,j) \right)^2, & w^W(x,y) &= 1 / \left(\sum_{i=1}^{x+4} \sum_{j=y-1}^{y+1} d^H(i,j) \right)^2 \\
w^H(x,y) &= 1 / \left(\sum_{i=x-2}^{x+2} \sum_{j=y-2}^{y+2} d^H(i,j) \right)^2, & w^V(x,y) &= 1 / \left(\sum_{i=x-2}^{x+2} \sum_{j=y-2}^{y+2} d^V(i,j) \right)^2
\end{aligned} \tag{2.18}$$

Ces poids sont d'autant plus faibles que la somme des gradients directionnels des différences de composantes est élevée.

3. Après avoir calculé $\hat{\delta}^{R-G}$ par interpolation bilinéaire pondérée de $\tilde{\delta}^{R-G,H}$ et $\tilde{\delta}^{R-G,V}$:

$$\hat{\delta}^{R-G}(x,y) = [w^V \otimes f \tilde{\delta}^{R-G,V}(x,y-1 : y+1) + w^H \cdot \tilde{\delta}^{R-G,H}(x-1 : x+1,y) \otimes f'] / w^c, \tag{2.19}$$

où $f = \frac{1}{4}[1 \ 2 \ 1]$ et \otimes désigne la somme des produits des matrices élément par élément, les auteurs calculent une interpolation pondérée des différences directionnelles :

$$\begin{aligned}
\tilde{\delta}^{R-G}(x,y) &= (1-w) \hat{\delta}^{R-G}(x,y) + [w^N \hat{\delta}^{R-G}(x,y-2) + w^S \hat{\delta}^{R-G}(x,y+2) \\
&\quad + w^E \hat{\delta}^{R-G}(x-2,y) + w^W \hat{\delta}^{R-G}(x+2,y)] w / w^T,
\end{aligned} \tag{2.20}$$

où $w^c = (w^H + w^V)$, $w^T = (w^N + w^S + w^E + w^W)$ et $w = 0.7$.

4. Le canal vert est alors estimé par :

$$\hat{I}^G(P) = \begin{cases} I^R(P) + \tilde{\delta}^{R-G}(P) & \text{si } P \in S^R, \\ I^B(P) + \tilde{\delta}^{B-G}(P) & \text{si } P \in S^B. \end{cases} \tag{2.21}$$

Afin d'estimer les composantes R et B manquantes, Pekkucuksen et Altunbasak [62] calculent d'abord $\hat{I}^k(P)$ en $P \in S^l$, $(k,l) \in \{R,B\}^2$, $k \neq l$, en utilisant le filtre Prb de taille 7×7 proposé par Paliy *et al.* [57] :

$$Prb = \frac{1}{32} \begin{bmatrix} 0 & 0 & -1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 10 & 0 & 10 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 10 & 0 & 10 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & -1 & 0 & 0 \end{bmatrix} \quad (2.22)$$

$$\begin{aligned} \hat{I}^R(x,y) &= \hat{I}^G(x,y) - \tilde{\delta}^{B-G}(x-3 : x+3, y-3 : y+3) \otimes Prb \\ \hat{I}^B(x,y) &= \hat{I}^G(x,y) - \tilde{\delta}^{B-G}(x-3 : x+3, y-3 : y+3) \otimes Prb \end{aligned} \quad (2.23)$$

Enfin, les composantes manquantes R et B aux positions G sont estimées en réutilisant les poids directionnels (*cf.* équation (2.18)) calculés pour l'estimation du canal vert.

2.3.3 Correction de la direction d'interpolation

Dans ses travaux de thèse, Phelippeau [63] propose de corriger les erreurs d'estimation en partant de l'idée que la direction d'interpolation en un pixel P doit être identique à celle de la majorité de ses voisins. Un tel choix local de direction majoritaire (*Local Majority Direction Choice*, LMDC [64]) repose sur l'idée que si un pixel est isolé dans son choix de direction d'interpolation, il est fort probable que cette direction soit erronée et doive être remplacée par la direction majoritaire des ses huit voisins les plus proches. La figure 2.4 montre cette procédure de correction.

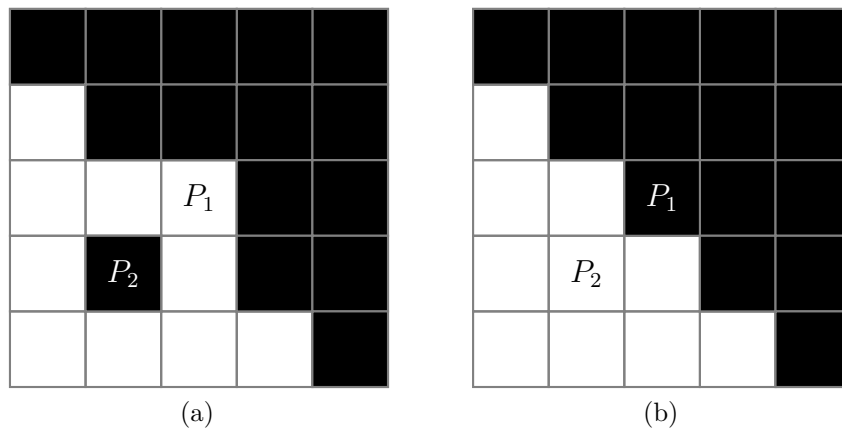


Figure 2.4 : : Correction des erreurs de choix des directions d'interpolations par la méthode de LMDC [63].

Dans la même logique, Chen *et al.* [14] utilisent un système de vote afin de choisir la direction d'interpolation parmi plusieurs possibles. Afin d'estimer la direction d'inter-

polation de la valeur de composante G manquante en $P \in S^R$ par exemple, Chen *et al.* y calculent la direction avec la méthode de Hamilton et Adams [30] ainsi qu'aux quatre pixels voisins $Q \in N^\times(P)$. Une fois ces cinq directions d'interpolation déterminées, la direction appropriée est choisie par un système de vote que l'on peut résumer comme suit :

- Si $\sum_{Q \in N^\times(P) \cup P} flag(Q) = 4$ ou 5 , la direction d'interpolation en P est verticale,
- Si $\sum_{Q \in N^\times(P) \cup P} flag(Q) = 0$ ou 1 , la direction d'interpolation en P est horizontale,
- Sinon, l'interpolation se fait dans toutes les directions,

où $flag(Q) = 1$ si la direction d'interpolation en Q est verticale, et $flag(Q) = 0$ sinon. Le point fort de cette méthode réside dans ce système de vote qui permet de ne pas choisir de direction d'interpolation si on est situé sur une région homogène. Dans ce cas, l'interpolation est faite dans toutes les directions, ce qui réduit les artéfacts dus à un mauvais choix de direction.

Afin d'estimer les valeurs G manquantes, Zhou *et al.* [77] calculent aussi les différences de composantes couleur horizontale et verticale selon les équations (2.16) et (2.17). Une fois obtenues les cartes de différences de composantes (*cf.* figure 2.3), les auteurs calculent les gradients des différences de composantes horizontale et verticale en chaque pixel. Afin de choisir la direction d'interpolation, ils calculent la somme des gradients verticaux et la somme des gradients horizontaux dans un voisinage de 5×5 pixels, puis comparent le rapport de ces sommes à un seuil T :

- Si $(1 + \sum d^H(x,y))/(1 + \sum d^V(x,y)) > T$, la direction d'interpolation choisie est verticale, $\hat{I}^G(x,y) = \hat{I}^{G,H}(x,y)$ donné par l'équation (2.13).
- Si $(1 + \sum d^V(x,y))/(1 + \sum d^H(x,y)) > T$, la direction d'interpolation choisie est horizontale, $\hat{I}^G(x,y) = \hat{I}^{G,H}(x,y)$ donné par l'équation (2.14).
- Sinon $\hat{I}^G(x,y) = \frac{w^1 \hat{I}^{G,H}(x,y) + w^2 \hat{I}^{G,V}(x,y)}{w^1 + w^2}$, où w^1 et w^2 sont les inverses des gradients horizontal et vertical calculés par Hamilton et Adams [30] (*cf.* équations (2.11) et (2.12)).

Le choix du seuil T est très important dans cette méthode. C'est pourquoi Zhou *et al.* déterminent T après avoir comparé le rapport signal-bruit pic-à-pic (*Peak Signal-Noise Ratio*, PSNR) obtenu avec différentes valeur de ce seuil. Les résultats fournis par les auteurs dans [77] montrent l'efficacité de cette méthode.

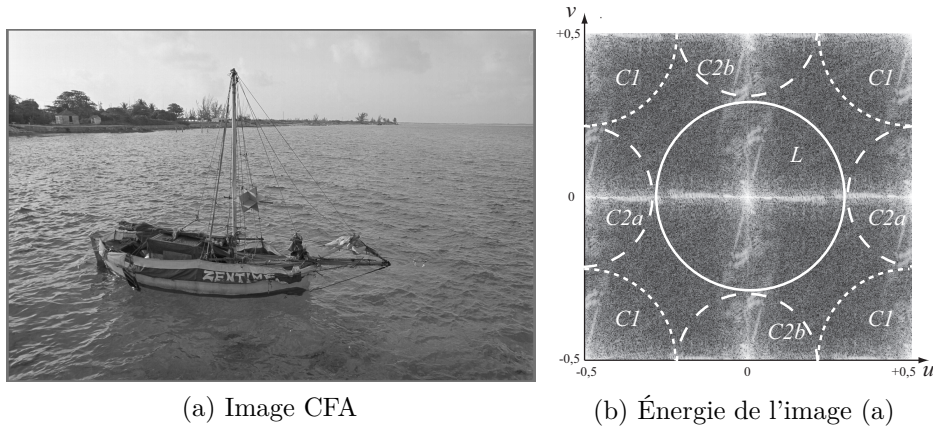


Figure 2.5 : Répartition de l'énergie d'une image CFA dans le plan de Fourier.

2.4 Méthodes fréquentielles

Contrairement à l'hypothèse de Bayer, les méthodes fréquentielles n'assimilent pas la composante G à la luminance ni R et B à la chrominance. Alleysson *et al.* [5] ont démontré qu'une image I^{CFA} peut être représentée comme la combinaison d'une composante de luminance L modulée en basses fréquences spatiales et de deux composantes de chrominance C_1 et C_2 en hautes fréquences spatiales. Alleysson *et al.* [5] et Dubois [26] ont montré que la transformée de Fourier F^{CFA} de l'image I^{CFA} peut être décomposée en celles des composante de luminance et de chrominance :

$$F^{CFA}(u,v) = F^L(u,v) + F^{C_1}(u - 0.5, v - 0.5) + F^{C_2}(u - 0.5, v) + F^{C_2}(u, v - 0.5). \quad (2.24)$$

La figure 2.5 montre la répartition correspondante de l'énergie de I^{CFA} dans le plan fréquentiel. Sur la figure 2.5b, on remarque que l'énergie de l'image CFA est répartie en neuf zones différentes. L'énergie du premier terme F^L de l'équation (2.24) est quasi-exclusivement située au centre du plan fréquentiel, celle de F^{C_1} est localisée dans les quatre coins du plan et celle de F^{C_2} se trouve sur les axes u et v des fréquences horizontales et verticales. Le dématricage est réalisé en estimant la luminance L et les chrominances C_1 et C_2 par sélection des fréquences correspondantes dans le plan de Fourier. La clé des méthodes basées sur la sélection de fréquences réside donc dans la conception de filtres permettant de séparer efficacement ces composantes L , C_1 et C_2 .

Dubois et ses collaborateurs ont effectué plusieurs travaux sur le dématricage par sélection de fréquences (FS). Remarquant que localement dans l'image, le chevauchement spectral entre la luminance et la chrominance se produit principalement soit dans la direction horizontale soit dans la direction verticale, Dubois [26] propose de pondérer

davantage la sous-composante (C_{2a} ou C_{2b}) la moins sujette au chevauchement. Leung *et al.* [45] améliorent la méthode précédente en utilisant une approche par moindres carrés pour une meilleure conception du filtre de sélection.

Lian *et al.* [46] proposent une méthode de dématriçage qui combine les analyses fréquentielle et spatiale. Ils estiment d'abord la luminance aux pixels $P \in S^G$ en appliquant un filtre passe-bas afin d'éliminer les hautes fréquences. Ensuite, ils estiment la luminance aux pixels $P \in S^R \cup S^B$ en utilisant les corrélations spectrale et spatiale. Finalement, une fois le plan de luminance obtenu, Lian *et al.* estiment les deux composantes couleur manquantes par interpolation bilinéaire (*cf.* équation (2.1)). L'utilisation du domaine fréquentiel aux seules positions G du CFA permet aux auteurs d'obtenir une très bonne estimation de la luminance.

2.5 Méthodes de dématriçage et de débruitage

La qualité de l'image estimée dépend de la procédure de dématriçage mais aussi du bruit généré par le processus d'acquisition. Le post-traitement de l'image estimée par un schéma de débruitage est insatisfaisant car cela génère de nombreux artefacts de couleur causés par ce bruit [75]. La plupart des méthodes de dématriçage ont en effet été conçues sous l'hypothèse de données sans bruit et introduisent une corrélation spatiale dans les caractéristiques du bruit, ce qui le rend très difficile à éliminer. Pour éviter ce phénomène, il convient de mener conjointement un dématriçage et un débruitage.

Étant donné que le dématriçage et le débruitage s'appuient tous deux sur une estimation en utilisant des échantillons voisins, certains schémas proposent d'effectuer conjointement les deux tâches. Condat [17] et Condat et Mosaddegh [18] proposent une méthode conjointe de dématriçage-débruitage avec pour idée de débruiter les canaux de chrominance avant d'estimer la luminance. Les auteurs débruiter les canaux de chrominance estimés par filtrage linéaire puis débruiter l'image de luminance par une méthode classique dédiée aux images en niveaux de gris. Hirakawa et Parks [37] proposent de coupler le dématriçage avec une méthode de débruitage basée sur l'approche des moindres carrés. Zhang *et al.* [76] estiment les différences de composantes couleur avec l'approche LMMSE (*Linear Minimum Mean Square error Estimation*) qui exploite à la fois la corrélation spectrale et spatiale afin de minimiser simultanément les erreurs d'interpolation et le bruit. Le canal qui dépend du bruit est alors supprimé à l'aide d'une approche par ondelettes. Bien que ces approches améliorent la qualité de l'image finale et assurent des estimations relativement rapides, elles ne permettent pas une conception

indépendante des algorithmes de débruitage et de dématricage.

3 Analyse de l'image CFA

Les méthodes de dématricage sont généralement conçues pour produire des images « visuellement satisfaisantes » sans se préoccuper de l'impact des artéfacts résultants sur d'ultérieures analyses bas niveau de ces images. De nombreuses méthodes de dématricage utilisent des filtres passe-bas afin d'atténuer les éventuels artéfacts présents dans les zones de hautes fréquences. Cependant, ces informations caractérisant les textures et/ou les contours dans une image sont essentielles pour tout un ensemble d'analyses bas niveau de celle-ci. La perte de ces informations peut donc altérer la qualité des résultats produits par une procédure de traitements de bas niveau appliquée aux images couleur estimées.

Yang [72] a étudié les artéfacts générés par différentes méthodes de dématricage et a proposé des mesures spécifiques afin de déterminer celles qui estiment des images couleur les mieux adaptées à la détection des contours. L'auteur a conclu que ces artéfacts sont en effet préjudiciables aux procédures d'analyses bas niveau de l'image couleur estimée. Breier *et al.* [10] confirment que le dématricage influe sur les performances de détection des contours en menant des expériences sur une base d'images industrielles.

Losson *et al.* [48, 47, 49] ont étudié la possibilité d'utiliser directement l'image CFA pour caractériser et classifier des textures couleur. Ils ont proposé d'adapter aux images CFA les motifs locaux binaires [47] et les matrices de co-occurrences chromatiques [48, 49] initialement conçus pour les images couleur.

Nous décrivons l'adaptation des matrices de co-occurrences chromatiques aux images CFA dans la première section de cette partie. L'image CFA est aussi exploitée par les algorithmes de compression car elle permet de réduire la taille des images compressées ainsi que le temps de compression ; ces approches seront détaillées dans la seconde section. Enfin, le seul travail s'étant intéressé à l'analyse bas niveau de l'image CFA pour la détection des contours est présenté dans la dernière section.

(*Local Binary Patterns*, LBPs)

3.1 Caractérisation de textures

Les matrices de co-occurrences chromatiques (*Chromatic Co-occurrence Matrices*, CCMs) sont des attributs de textures du second ordre qui s'inspirent de ceux proposés par Haralick *et al.* [32] pour les images en niveaux de gris et ont été étendus aux images couleur par Palm [58].

Soit \mathbf{I} une image couleur dont chaque composante est quantifiée sur q niveaux. Notons $M_{d,\theta}^{k,l}[\mathbf{I}]$ la CCM qui mesure l'interaction spatiale entre les composantes couleur $(k,l) \in \{R,G,B\}^2$ des pixels de \mathbf{I} selon une distance et une direction données d et θ . La cellule $M_{d,\theta}^{k,l}[\mathbf{I}](i,j)$ de cette matrice, $0 \leq i \leq (q-1)$, $0 \leq j \leq (q-1)$, contient le nombre de fois où le pixel Q dont le niveau de la composante l est égal à j se situe dans un voisinage $\mathbb{N}(P)$ d'un pixel P dont le niveau de la composante k est égal à i . Pour un voisinage donné $\mathbb{N}(P)$, six CCMs sont calculées pour caractériser l'image couleur \mathbf{I} :

- trois matrices intra-composantes $M^{R,R}[\mathbf{I}]$, $M^{G,G}[\mathbf{I}]$ et $M^{B,B}[\mathbf{I}]$.
- trois matrices inter-composantes $M^{R,G}[\mathbf{I}]$, $M^{R,B}[\mathbf{I}]$ et $M^{G,B}[\mathbf{I}]$.

Initialement conçues pour la classification de textures couleur, les CCMs ont été adaptées par Losson *et al.* [49] pour la caractérisation de textures représentées par des images CFA. Pour une image CFA, les auteurs calculent six CCMs appelées CFA CCMs. La cellule $M_{CFA}^{k,l}[I^{CFA}](i,j)$ de chacune de ces matrices contient le nombre de fois où le pixel Q de l'image CFA dont le niveau $I^{CFA}(Q)$ est égal à j se situe dans un voisinage $\mathbb{N}_{CFA}(P)$ d'un pixel P dont le niveau $I^{CFA}(P)$ est égal à i . Pour calculer les CFA CCMs, il faut définir le voisinage \mathbb{N}_{CFA} considéré en chaque pixel de I^{CFA} qui, à la différence des images couleur, dépend du couple de composantes couleur (k,l) considéré. Les auteurs définissent le voisinage considéré $\mathbb{N}_{CFA}^{k,l}(P)$ ainsi :

- Pour les matrices intra-composantes, le voisinage $\mathbb{N}_{CFA}^{k,k}(P)$, $k \in \{R,G,B\}$, est défini exactement comme celui $\mathbb{N}(P)$ utilisé pour la classification des textures dans des images couleur mais en prenant une distance $d = 2$ (*cf.* figures 2.6a, 2.6b et 2.6c).
- Pour les matrices inter-composantes, le voisinage $\mathbb{N}_{CFA}^{k,l}(P)$, $(k,l) \in \{R,G,B\}^2$, $k \neq l$, est défini par les quatre pixels voisins horizontaux ($\mathbb{N}^{R,G}$), diagonaux ($\mathbb{N}^{R,B}$), ou verticaux ($\mathbb{N}^{B,G}$) (*cf.* figures 2.6d, 2.6e et 2.6f).

Les auteurs ont démontré que le coût calculatoire de classification est trois fois inférieur en utilisant une image CFA par rapport à l'utilisation d'une image couleur et cela pour une performance de classification meilleure. Ces résultats confortent l'hypothèse que le dématriçage génère des artéfacts qui dégradent la caractérisation d'une image estimée $\hat{\mathbf{I}}$.

3.2 Compression/décompression

Plusieurs auteurs dont Xie *et al.* [71], Koh et Mitra [41] ou encore Lee et Ortega [44], ont étudié la faisabilité de compresser directement l'image CFA. L'application directe

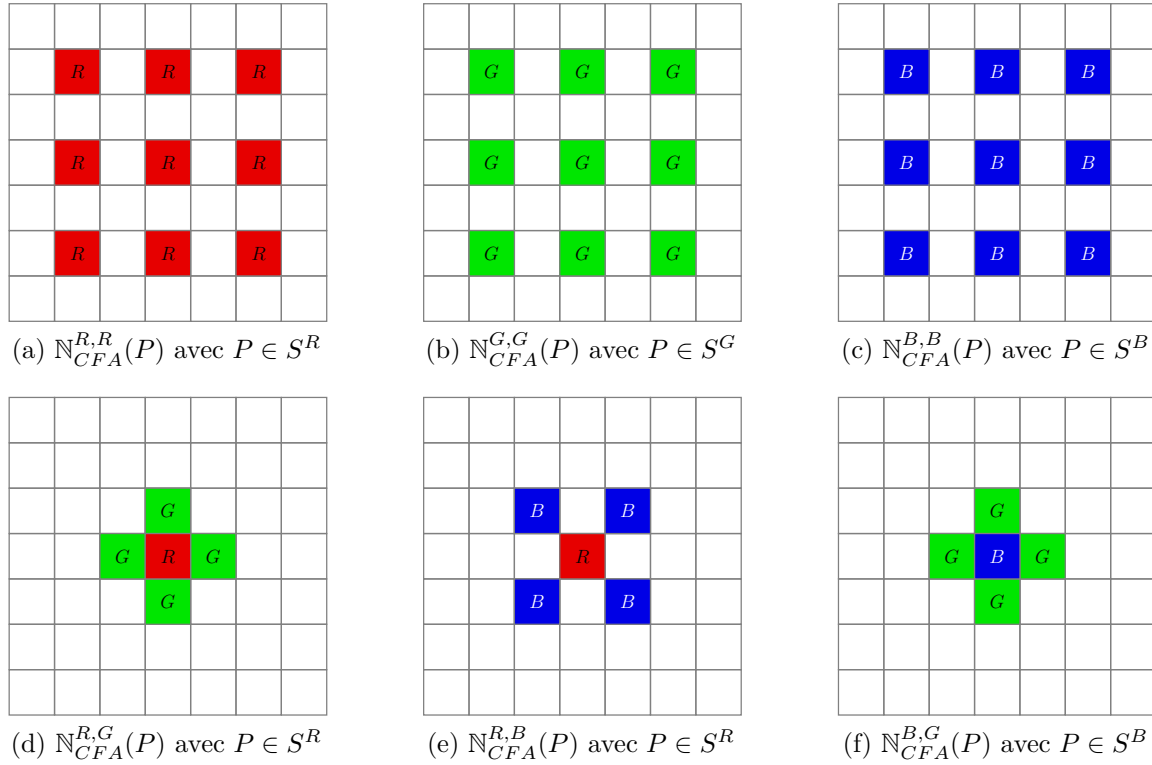


Figure 2.6 : : Voisinages utilisés pour la classification des textures dans une image CFA [49].

des algorithmes de compression sur l'image CFA réduit considérablement la taille des images compressées ainsi que le temps de compression. La figure 2.7 illustre la méthode conventionnelle de compression ou décompression dans laquelle le dématricage est effectué avant la compression (*cf.* colonne de gauche) et les méthodes qui exploitent l'image CFA dans lesquelles la compression est réalisée avant le dématricage (*cf.* colonne de droite).

Les algorithmes de compression initialement développés pour les images couleur ou en niveaux de gris, comme celui défini par la norme JPEG (*Joint Photographic Experts Group*), ne peuvent pas s'appliquer directement à l'image CFA en raison de son treillis particulier. Il faut pour cela transformer l'image CFA afin d'obtenir un format approprié. Le cas de l'algorithme JPEG est envisagé ci-dessous.

Comme dans l'algorithme original, l'image CFA est convertie de l'espace RGB vers l'espace YC_bC_r (*cf.* figure 2.8) pour compresser davantage les composantes de chrominance que celle de luminance à laquelle l'œil humain est plus sensible. Cependant, cette conversion ne peut pas être réalisée en chaque pixel de l'image CFA car une seule composante couleur y est disponible. Lee et Ortega [44] proposent de prendre en compte des blocs disjoints de 2×2 pixels contenant deux niveaux disponibles de la composante G

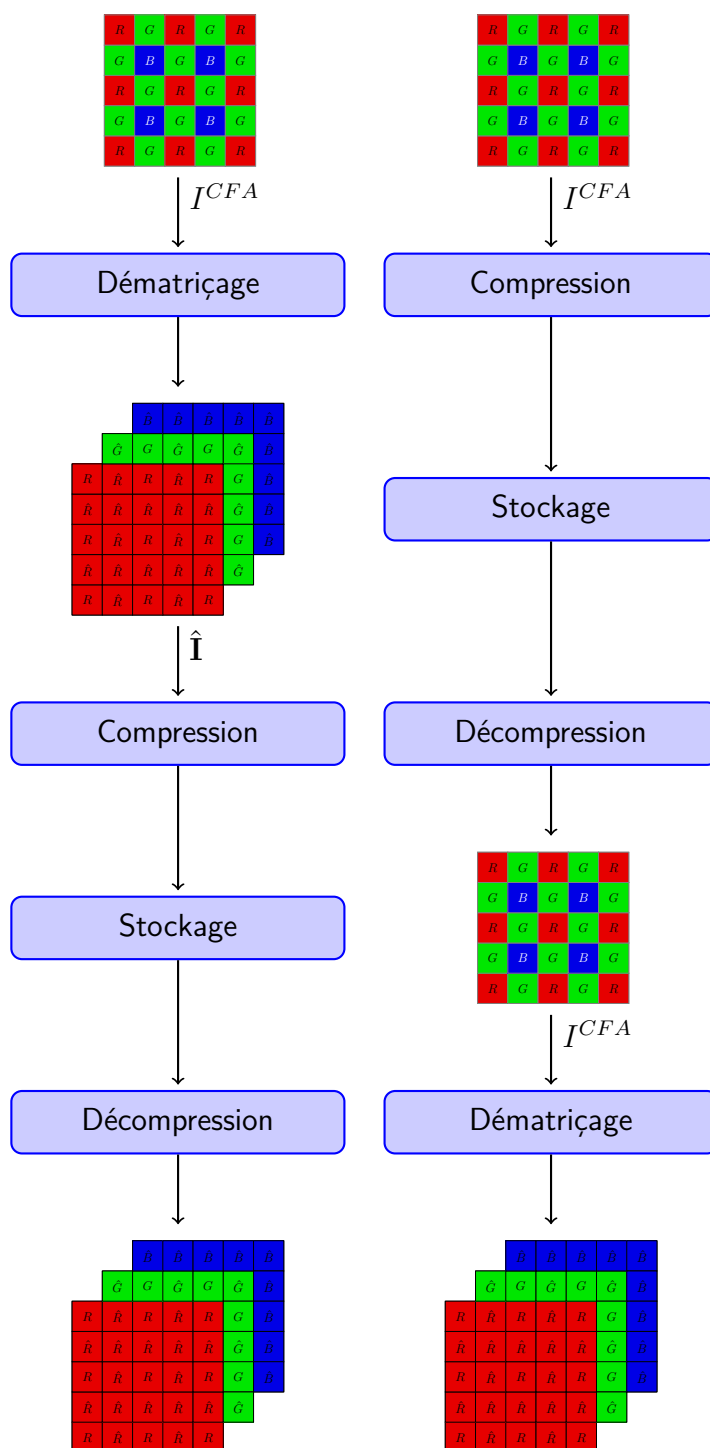


Figure 2.7 : Compression/décompression de l'image $\hat{\mathbf{I}}$ (après dématriçage) et I^{CFA} (avant dématriçage).

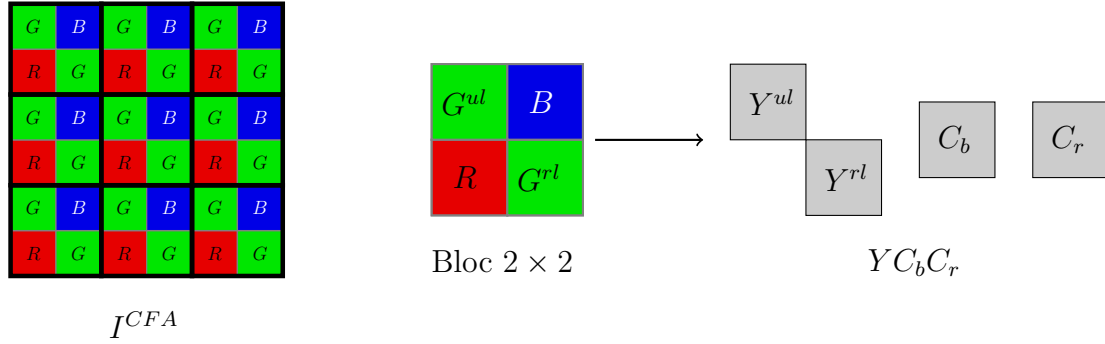


Figure 2.8 : : Calcul des composantes YC_bC_r pour un bloc 2×2 de l'image CFA.

(G^{ul} (*upper left*) et G^{rl} (*right lower*)), un niveau de R et de B .

Les équations utilisées pour calculer les deux composantes de luminance Y^{ul} et Y^{rl} et les deux composantes de chrominance C_b et C_r dans chaque bloc de 2×2 pixels de l'image I^{CFA} sont données par [44] :

$$\begin{pmatrix} Y^{ul} \\ Y^{rl} \\ C_b \\ C_r \end{pmatrix} = \begin{pmatrix} a_1 & 0 & a_2 & a_3 \\ 0 & a_1 & a_2 & a_3 \\ a_4 & a_4 & a_5 & a_6 \\ a_7 & a_7 & a_8 & a_5 \end{pmatrix} \begin{pmatrix} G^{ul} \\ G^{rl} \\ B \\ R \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 128 \\ 128 \end{pmatrix}, \quad (2.25)$$

où $a_1 = 128.55$, $a_2 = 24.97$, $a_3 = 65.48$, $a_4 = -37.1$, $a_5 = 112$, $a_6 = -37.8$, $a_7 = -37.1$ et $a_8 = -18.21$ sont les coefficients standards de conversion de l'espace RGB vers l'espace YC_bC_r .

Les composantes de chrominance C_b et C_r résultantes se présentent sous la forme de sous-images rectangulaires de taille $M/2 \times N/2$. Elles sont similaires à celles obtenues par le sous-échantillonnage $4 : 2 : 0$ dans le cas des images couleur et sont donc adaptées à la compression JPEG. En revanche, la composante de luminance Y possède le même treillis en quinconce que S^G et nécessite donc une réorganisation afin d'appliquer l'algorithme JPEG. Plusieurs méthodes de réorganisation de Y sont proposées dans la littérature.

Koh et Mitra [41] proposent deux méthodes. La première consiste à filtrer la composante Y en utilisant un filtre passe-bas afin d'atténuer les hautes fréquences verticales et horizontales. Ensuite, les auteurs divisent la composante de luminance lissée \bar{Y} en deux sous-images de taille $M/2 \times N/2$, l'une contenant les composantes \bar{Y}^{ul} (positions paires) et l'autre les composantes \bar{Y}^{rl} (positions impaires) (*cf.* figure 2.9). Ces deux sous-images sont ensuite prêtes à être compressées. La décompression se fait exactement dans le sens inverse de l'étape de compression. La deuxième méthode utilise un filtre passe-bas h_c qui permet à la fois d'atténuer les hautes fréquences et d'obtenir une sous-image rectan-

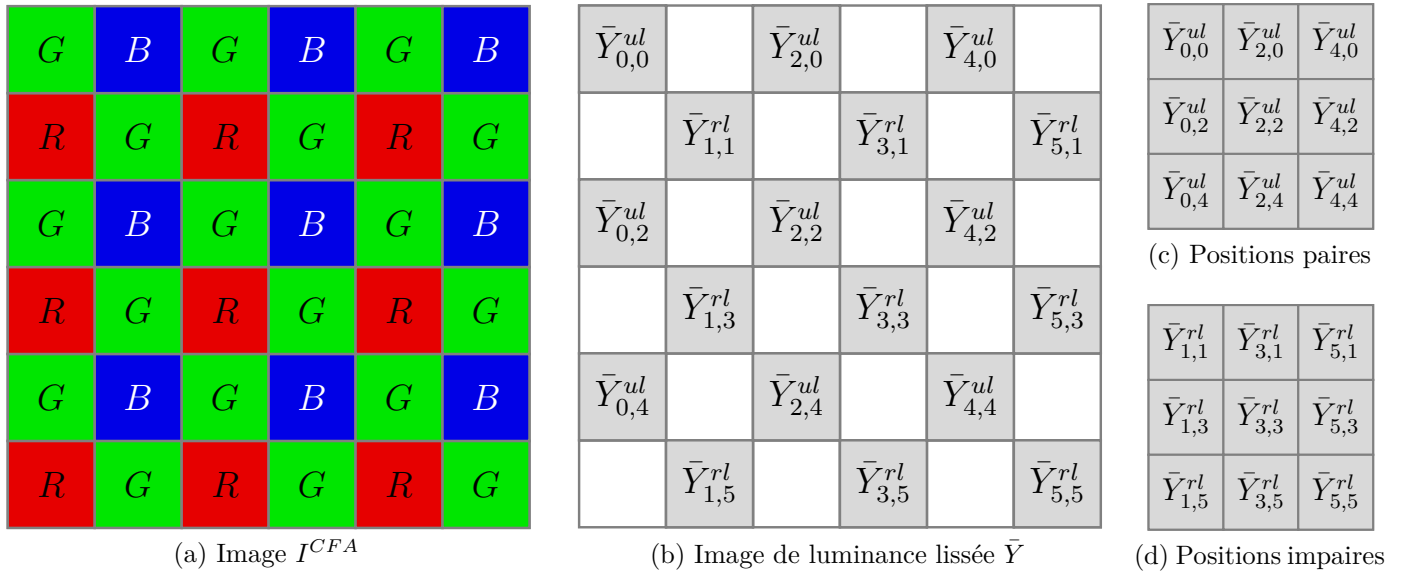


Figure 2.9 : : Séparation des deux sous-composantes Y pour la compression JPEG.

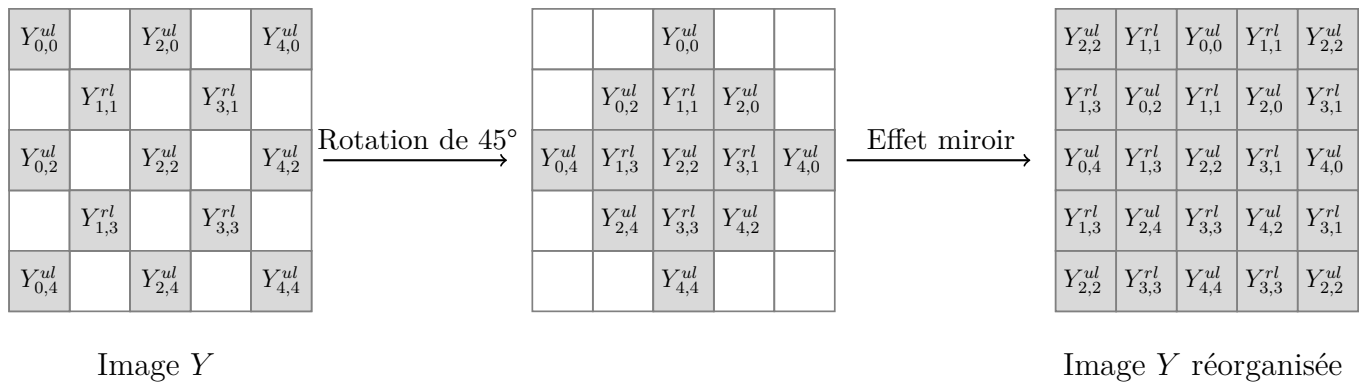


Figure 2.10 : : Rotation de la composante de luminance Y de 45° (Lee et Ortega [44]).

gulaire prête à être compressée par l'algorithme JPEG. Le filtre h_d permet de repasser de la forme rectangulaire au treillis en quinconce lors de l'étape de décompression :

$$h_c(x,y) = \frac{1}{4} \begin{pmatrix} 0 & 0 & 1 \\ 0 & 2 & 4 \\ 0 & 0 & 1 \end{pmatrix} \text{ et } h_d(x,y) = \frac{1}{2} \begin{pmatrix} 0 & -1 & 0 \\ 2 & 4 & 0 \\ 0 & -1 & 0 \end{pmatrix}.$$

Une autre méthode de réorganisation de la composante Y est proposée par Lee et Ortega [44]. Les auteurs appliquent une rotation de 45° dans le sens trigonométrique inverse à l'image Y . L'effet miroir est utilisé afin de traiter les pixels de bord (représentés en blanc sur la figure 2.10). La décompression se fait dans le sens inverse de l'étape de compression.

Contrairement aux méthodes précédentes, Xie *et al.* [71] proposent d'estimer d'abord

les valeurs manquantes de la composante G avec une interpolation bilinéaire (*cf.* figure 2.1), puis d'appliquer un sous-échantillonnage de facteur 2 verticalement pour obtenir une sous-image de G de taille $M \times N/2$. Une conversion de l'espace RGB vers YC_bC_r est ensuite réalisée en utilisant l'équation (2.25). Les sous-images de luminance et de chrominance sont ensuite compressées avec l'algorithme de compression sans perte $JPEG-LIS$. La différence de cette méthode avec celles proposées par Koh et Mitra [41] réside dans le fait que l'étape de réorganisation est faite dans l'espace RGB et non dans YC_bC_r , ce qui permet d'obtenir une meilleure estimation de la luminance [71].

Koh et Mitra [41], Lee et Ortega [44] et Xie *et al.* [71] ont comparé les performances de leurs algorithmes de compression à celles des méthodes standards et en ont conclu que le fait de traiter directement l'image CFA améliore la fidélité des images compressées tout en réduisant nettement le temps de calcul.

3.3 Détection des contours

À notre connaissance, très peu de chercheurs se sont intéressés à la détection des contours à partir de l'image CFA. Chen *et al.* [13] se sont penchés sur ce problème non pas afin d'améliorer la qualité de détection mais pour le gain potentiel en temps de calcul. En effet, il est intéressant d'éviter l'étape de dématricage qui peut être coûteuse.

Avant l'étape de détection des contours dans les images couleur ou en niveaux de gris, un filtre de lissage est classiquement appliqué afin de réduire l'influence du bruit dans l'image. De la même façon, Chen *et al.* [13] ont adapté le filtre gaussien à l'image CFA pour la partie lissage, puis utilisé un filtre laplacien pour détecter les contours.

Le filtre de lissage gaussien proposé (*cf.* figure 2.11) utilise un voisinage creux de façon à tenir compte des niveaux de la même composante couleur k en un pixel $P \in S^k$, $k \in \{R, G, B\}$, et à utiliser la même structure de voisinage que le filtre laplacien utilisé pour l'étape de dérivation (*cf.* figure 2.12). Le filtre laplacien de Chen *et al.* est inspiré des filtres utilisés par Malvar *et al.* [52] pour estimer les composantes couleur manquantes dans l'image CFA.

Chen *et al.* [13] n'ont pas évalué objectivement leur méthode mais se sont contentés de résultats visuels dont la figure 2.13 montre un exemple de résultat. Les résultats obtenus ne sont pas satisfaisants en terme de qualité de détection car le filtre laplacien utilisé est très sensible au bruit (*cf.* figure 2.13). Ces auteurs n'ont pas non plus étudié l'influence du treillis en quinconce caractéristique de l'image CFA sur les résultats de détection des contours. Ils ont néanmoins démontré qu'il est possible d'utiliser directement l'image

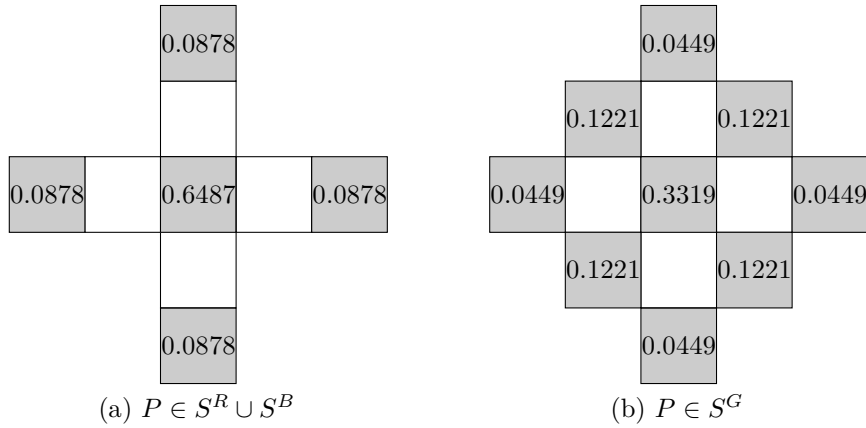


Figure 2.11 : : Voisinages utilisés par Chen *et al.* [13] pour le lissage gaussien de l'image CFA (avec coefficients du filtre pour $\sigma = 1$).

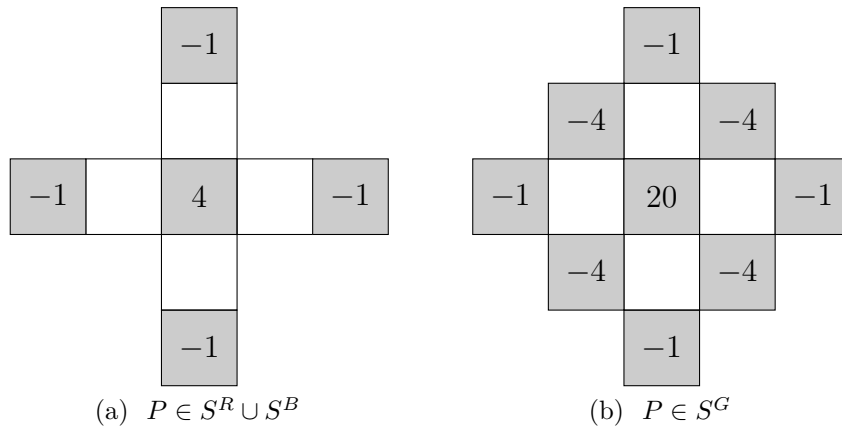


Figure 2.12 : : Voisinages utilisés par le filtre laplacien de Chen *et al.* [13] appliqué à l'image CFA lissée.

CFA pour la détection des contours avec un gain non négligeable en temps de calcul, ce qui permet d'envisager une telle analyse dans les application de traitement en temps réel.

4 Conclusion

Dans ce chapitre, nous avons vu les différentes méthodes de dématriçage, de la plus simple qui procède par interpolation bilinéaire aux plus sophistiquées comme les méthodes fréquentielles. Les méthodes les plus récentes, comme celles proposées par Pekukucksen et Altunbasak [62] ou Kiku *et al.* [39], fournissent des images estimées de très bonne qualité au prix d'un temps de calcul passablement élevé. Elles sont donc pratiquement impossibles à embarquer dans une caméra pour un traitement en temps réel et sont plutôt réservées à un traitement hors ligne sur ordinateur. Toutes ces méthodes ont

(a) Image CFA lissée \bar{I}^{CFA} ($\sigma = 1$)(b) Réponse du filtre laplacien sur \bar{I}^{CFA} (couleurs inversées)

Figure 2.13 : : Détection des contours sur une image CFA comme proposé par Chen *et al.* [13].

été développées pour produire des images les plus « visuellement satisfaisantes » possibles pour l'œil humain. Afin d'atténuer les artéfacts générés par le dématricage dans les zones hautes fréquences, plusieurs auteurs [46, 26] proposent d'appliquer à l'image CFA un filtre passe-bas. Cependant, les textures et les contours présents dans les images se trouvent dans les informations de hautes fréquences et l'application d'un filtre passe-bas altère la qualité des résultats produits par des procédures d'analyse de bas niveau.

Nous avons donc fait l'état de l'art des travaux qui analysent l'image CFA. Plusieurs travaux ont porté sur la possibilité d'utiliser directement l'image CFA pour caractériser et classifier des textures couleur. D'autres ont étudié la compression spécifique de l'image CFA. Seuls Chen *et al.* [13] ont proposé d'utiliser directement l'image CFA pour détecter les contours, au moyen d'un filtre de lissage gaussien et d'un filtre dérivateur laplacien. Que ce soit pour la compresser ou y détecter les contours, l'exploitation directe de l'image CFA nécessite de transformer celle-ci ou d'utiliser un voisinage spécifiquement adapté.

Les résultats obtenus sont très encourageants et nous permettent d'explorer dans quelle mesure d'autres méthodes de détection des contours, conçues pour les images couleurs estimées, peuvent être transposées à l'image CFA.

Dans le chapitre suivant, nous nous inspirons des stratégies mentionnées dans ces travaux pour proposer plusieurs approches de détection des contours à partir d'une

image CFA. En se basant sur un gradient plutôt qu'un laplacien pour éviter une sur-sensibilité au bruit, nous adaptions notamment les filtres de lissage et de dérivation de Deriche à l'image CFA.

Chapitre 3

Calcul du gradient à partir de l'image CFA

1 Introduction

Ce chapitre est dédié aux contributions personnelles principales de cette thèse.

Afin de détecter les contours dans une image couleur, Di Zenzo [25] utilise les dérivées partielles selon x et y de chaque composante couleur afin d'estimer le vecteur gradient en chaque pixel de cette image. Les contours de l'image couleur sont ensuite détectés en analysant la norme et la direction du vecteur gradient. Pour calculer les dérivées partielles, notre choix s'est porté principalement sur le filtre de Deriche car c'est l'un des plus performants et des plus utilisés dans la littérature.

Dans ce chapitre, nous commençons par faire un rappel des méthodes qui calculent un gradient couleur. Ensuite, nous détaillons le filtre proposé par Deriche [23, 24] pour le calcul des dérivées premières partielles d'une image couleur ou en niveaux de gris. Dans la dernière partie de ce chapitre, nous détaillons les méthodes que nous proposons pour le calcul des dérivées partielles à partir de l'image CFA. Nous proposons deux approches afin d'exploiter ce type d'images.

L'approche vectorielle exploite la couleur en calculant les dérivées partielles des trois composantes couleur en chaque pixel. Dans cette approche, nous proposons trois façons de procéder. La première estime les dérivées partielles manquantes en chaque pixel, la deuxième estime les composantes couleur lissées manquantes et la dernière calcule directement les trois dérivées partielles en chaque pixel de l'image CFA en utilisant les filtres de Deriche en précision inter-pixel proposés par Montesinos et Dattenny [56] et Fiorio [28].

L'approche scalaire estime une image de luminance adaptée à la détection des contours.

Nous proposons pour cela une méthode adaptée à l'image CFA qui utilise le filtre de lissage bidimensionnel de Deriche.

2 Gradient d'une image couleur dématriciée

Pour détecter les contours dans une image couleur, soulignons d'abord qu'il existe des approches basées sur les dérivées secondes de l'image. Cumani [19] a ainsi étendu le laplacien de Marr et Hildreth [53] aux images multi-composantes. En raison de la grande sensibilité du laplacien au bruit [54], nous lui préférons les approches basées sur le gradient. Dans cette section, nous présentons les différentes méthodes de la littérature permettant d'estimer un gradient couleur. Nous commençons par le gradient morphologique introduit par Chanussot et Lambert [12] puis présentons le gradient vectoriel de Di Zenzo [25].

2.1 Gradient morphologique couleur

Le gradient morphologique proposé par Beucher [8] s'applique aux images en niveaux de gris. Il consiste à effectuer une dilatation et une érosion de l'image initiale, puis à soustraire ces deux images [33]. Les relations d'ordre sur lesquelles s'appuient l'érosion et la dilatation sont perdues dans l'espace couleur RGB .

Pour définir le gradient morphologique couleur, Chanussot et Lambert [12] ont proposé d'utiliser l'ordre par entrelacement de bits. Cela consiste à considérer la décomposition en séquence de bits de chaque valeur de composante \hat{I}^k en un pixel P :

$$\hat{I}^k(P) = \sum_{j=1}^8 2^{8-j} \cdot \hat{I}_j^k(P), \quad (3.1)$$

où $\hat{I}_j^k(P) \in \{0,1\}$ est le $j^{\text{ème}}$ bit de la valeur de la composante $k \in \{R,G,B\}$. Ces trois composantes sont recombinaées dans une seule composante scalaire de 3×8 bits en plaçant alternativement le premier bit de \hat{I}^R , puis le premier bit de \hat{I}^G et le premier bit de \hat{I}^B , puis le deuxième bit de chaque composante, et ainsi de suite jusqu'au huitième bit. Le gradient morphologique est alors calculé sur l'image résultante considérée comme une image en niveaux de gris sur laquelle est appliquée une érosion et une dilatation. Le gradient morphologique couleur dans l'espace RGB est calculé par décodage de l'image scalaire de 24 bits en faisant l'opération inverse de la méthode de codage.

2.2 Gradient de Di Zenzo

Di Zenzo [25] utilise les dérivées partielles en x et en y de chaque composante couleur afin d'estimer le gradient en chaque pixel d'une image couleur. Comme dans le cas des images en niveaux de gris, les contours de l'image couleur sont détectés en seuillant les maxima locaux de la norme du gradient dans la direction du gradient.

Afin d'estimer le gradient en chaque pixel d'une image couleur dématricée $\hat{\mathbf{I}} = (\hat{I}^R, \hat{I}^G, \hat{I}^B)$, Di Zenzo utilise les dérivées partielles selon x et y de chaque composante couleur. Nous regroupons ces dérivées partielles dans deux vecteurs $\hat{\mathbf{I}}_x = (\hat{I}_x^R, \hat{I}_x^G, \hat{I}_x^B)$ et $\hat{\mathbf{I}}_y = (\hat{I}_y^R, \hat{I}_y^G, \hat{I}_y^B)$.

Les dérivées partielles d'ordre 1 peuvent être calculées à l'aide d'une simple approximation différentielle. En supposant que \hat{I}^k est continue et dérivable pour tout $k \in \{R, G, B\}$, ces dérivées partielles peuvent être calculées en chaque pixel $P(x, y)$ par :

$$\begin{aligned}\hat{I}_x^k(x, y) &= \frac{1}{2} \left(\hat{I}^k(x+1, y) - \hat{I}^k(x-1, y) \right), \\ \hat{I}_y^k(x, y) &= \frac{1}{2} \left(\hat{I}^k(x, y+1) - \hat{I}^k(x, y-1) \right).\end{aligned}\tag{3.2}$$

Ces dérivées partielles premières sont utilisées pour calculer la direction du gradient θ^* ainsi que sa norme $\|\nabla \hat{\mathbf{I}}\|$ en chaque pixel P (omis ici pour alléger les équations), en trouvant la valeur de θ qui maximise la première forme fondamentale :

$$d\hat{\mathbf{I}}^2(\theta) = a \cos^2 \theta + 2b \cos \theta \sin \theta + c \sin^2 \theta,\tag{3.3}$$

où $a = \hat{\mathbf{I}}_x \hat{\mathbf{I}}_x = \sum_{k=R,G,B} (\hat{I}_x^k)^2$, $b = \hat{\mathbf{I}}_x \hat{\mathbf{I}}_y = \sum_{k=R,G,B} \hat{I}_x^k \hat{I}_y^k$, $c = \hat{\mathbf{I}}_y \hat{\mathbf{I}}_y = \sum_{k=R,G,B} (\hat{I}_y^k)^2$. La recherche du maximum s'effectue selon la méthode du compas qui calcule la direction comme suit : $\theta^* = \underset{\theta \in [-\pi, \pi]}{\operatorname{argmax}} d\hat{\mathbf{I}}^2(\theta)$, $\|\nabla \hat{\mathbf{I}}\|^2 = |d\hat{\mathbf{I}}^2(\theta^*)|$.

Afin d'éviter la procédure de dématricage, nous proposons de calculer le gradient couleur directement à partir de l'image CFA en s'appuyant l'approche de Di Zenzo conçue pour les images multi-composantes pleinement définies. Notre contribution consiste à calculer les dérivées partielles directement à partir de l'image CFA. Elles sont notées ici $\hat{\mathbf{I}}_x$ et $\hat{\mathbf{I}}_y$ pour les méthodes qui estiment les dérivées partielles ou les composantes lissées et \mathbf{I}_x et \mathbf{I}_y pour les méthodes qui ne procèdent à aucune estimation.

2.3 Filtres de Deriche

Afin d'atténuer l'influence de bruit, des filtres optimaux sont conçus à partir des trois critères d'optimalité de la détection des contours de Canny [11] :

1. Bonne détection : ce critère revient à maximiser le rapport signal sur bruit (bruit blanc additif de moyenne nulle). Le filtre doit donner une réponse maximale au voisinage d'un contour et minimiser les fausses réponses.
2. Bonne localisation : les points contours doivent être localisés avec précision. Cela revient à minimiser la distance entre le vrai contour et le contour détecté.
3. Unicité de la réponse : un contour unique doit provoquer une réponse unique. Ce critère est inclus dans le premier car détecter deux contours au lieu d'un implique qu'une des réponses du filtre est erronée.

Deriche [23] propose un filtre de dérivation qui répond aux mêmes critères d'optimalité que celui de Canny [11] mais qui possède une réponse impulsionnelle infinie en utilisant un filtre récursif de second ordre. Ce filtre est exprimé par :

$$\mathcal{D}_\alpha(x) = -m'x e^{-\alpha|x|}. \quad (3.4)$$

Le coefficient de normalisation m' du filtre de dérivation de Deriche est donné par :

$$m' = -\frac{(1 - e^{-\alpha})^2}{e^{-\alpha}}. \quad (3.5)$$

Le filtre de lissage de Deriche est obtenu par intégration du filtre de dérivation :

$$\mathcal{L}_\alpha(x) = m(\alpha|x| + 1) e^{-\alpha|x|}. \quad (3.6)$$

Afin de normaliser ce filtre de lissage, m est donné par :

$$m = \frac{(1 - e^{-\alpha})^2}{1 + 2\alpha e^{-\alpha} - e^{-2\alpha}}. \quad (3.7)$$

Les largeurs des filtres de Deriche sont définies par le paramètre α . Fixer celui-ci nécessite de trouver un compromis entre la détection et la localisation des contours. Les petites valeurs de α (généralement, $\alpha < 1$) donnent la priorité à la détection : le filtre est plus robuste au bruit mais les contours sont mal localisés. Par contre, les valeurs élevées de α ($\alpha > 1$) fournissent une bonne localisation mais rendent le filtre sensible au bruit.

Deriche [24] propose une implémentation récursive de ses filtres. Pour une image cou-

leur dématricée $\hat{\mathbf{I}}$, l'algorithme de Deriche s'écrit de façon récursive par l'intermédiaire de deux filtres qui parcourent chaque composante couleur \hat{I}^k , $k \in \{R, G, B\}$, de haut en bas et de bas en haut. Le résultat est dérivé de gauche à droite et de droite à gauche afin d'estimer la dérivée partielle \hat{I}_x^k . La dérivée partielle \hat{I}_y^k est calculée en parcourant chaque composante couleur \hat{I}^k dans les directions perpendiculaires à celles utilisées pour le calcul de \hat{I}_x^k .

Avant de présenter cette implémentation récursive, nous introduisons une distance spatiale notée d qui sépare le pixel considéré P de son plus proche voisin horizontal (ou vertical) Q caractérisé par la même composante couleur. Dans le cas d'une image pleinement définie, telle que chaque canal couleur \hat{I}^k d'une image couleur dématricée, nous fixons $d = 1$ et $Q \in \mathbb{N}^+(P)$ (*cf.* figure 2.1). Pour l'image CFA, il est nécessaire de fixer $d = 2$, de telle sorte que $Q \in \mathbb{N}_{CFA}^{k,k}(P)$, $k \in \{R, G, B\}$ (*cf.* figures 2.6a, à 2.6c).

Considérons une image en niveaux de gris I qui peut aussi être une image de composante \hat{I}^k ou l'image I^{CFA} . Soit $I_{x,\alpha,d}$ la dérivée partielle première de I selon x étant donné le paramètre α et la distance spatiale d . Pour calculer celle-ci, un lissage est d'abord appliqué dans la direction verticale (selon y) de haut en bas (*cf.* équation (3.8)) et de bas en haut (*cf.* équation (3.9)) :

$$\bar{I}_{\alpha,d}^-(x,y) = a_1 I(x,y) + a_2 I(x,y-d) + b_1 \bar{I}_{\alpha,d}^-(x,y-d) + b_2 \bar{I}_{\alpha,d}^-(x,y-2d), \quad (3.8)$$

$$\bar{I}_{\alpha,d}^+(x,y) = a_3 I(x,y+d) + a_4 I(x,y+2d) + b_1 \bar{I}_{\alpha,d}^+(x,y+d) + b_2 \bar{I}_{\alpha,d}^+(x,y+2d), \quad (3.9)$$

où $a_1 = m \doteq \frac{(1-e^{-\alpha})^2}{1+2\alpha e^{-\alpha} - e^{-2\alpha}}$, $a_2 = m(\alpha-1)e^{-\alpha}$, $a_3 = m(\alpha+1)e^{-\alpha}$, $a_4 = -m e^{-2\alpha}$, $b_1 = 2e^{-\alpha}$, et $b_2 = -e^{-2\alpha}$ sont des coefficients de pondération qui dépendent du seul paramètre α . L'image lissée verticalement est alors donnée par : $\bar{I}_{\alpha,d} = \bar{I}_{\alpha,d}^- + \bar{I}_{\alpha,d}^+$. Signalons que les niveaux manquants des pixels de bord de I , $\bar{I}_{\alpha,d}^-$ et $\bar{I}_{\alpha,d}^+$ sont fixés à 0 dans les équations (3.8) et (3.9).

Pour obtenir la dérivée partielle $I_{x,\alpha,d}$, $\bar{I}_{\alpha,d}$ est ensuite dérivée horizontalement en la parcourant de gauche à droite (*cf.* équation (3.10)) et de droite à gauche (*cf.* équation (3.11)) :

$$I_{x,\alpha,d}^-(x,y) = a_5 \bar{I}_{\alpha,d}(x,y) + a_6 \bar{I}_{\alpha,d}(x-d,y) + b_1 I_{x,\alpha,d}^-(x-d,y) + b_2 I_{x,\alpha,d}^-(x-2d,y), \quad (3.10)$$

$$I_{x,\alpha,d}^+(x,y) = a_7 \bar{I}_{\alpha,d}(x+d,y) + a_8 \bar{I}_{\alpha,d}(x+2d,y) + b_1 I_{x,\alpha,d}^+(x+d,y) + b_2 I_{x,\alpha,d}^+(x+2d,y), \quad (3.11)$$

où $a_5 = 0$, $a_6 = 1$, $a_7 = -1$, $a_8 = 0$. Cette seconde étape fournit la dérivée partielle de

Deriche horizontale de $I : I_{x,\alpha,d} = c \left(I_{x,\alpha,d}^- + I_{x,\alpha,d}^+ \right)$, avec $c = -(1 - e^{-\alpha})^2$.

Pour calculer le gradient de Di Zenzo à partir d'une image dématriciée $\hat{\mathbf{I}}$, les équations (3.8)–(3.11) sont appliquées sur les trois composantes couleur $\{\hat{I}^k\}_{k=R,G,B}$ avec $d = 1$. Elles nous permettent de calculer la dérivée partielle de Deriche (vectorielle) $\hat{\mathbf{I}}_{x,\alpha,1}$ d'une image couleur dématriciée $\hat{\mathbf{I}}$. Cette méthode classique est décrite dans la partie haute de la figure 3.1.

La dérivée partielle de Deriche selon y , $\hat{\mathbf{I}}_{y,\alpha,1}$, est calculée de la même manière que $\hat{\mathbf{I}}_{x,\alpha,1}$ mais en lissant chaque composante couleur $\{\hat{I}^k\}_{k=R,G,B}$ selon x avant de la dériver selon y .

3 Approches vectorielles pour le calcul des dérivées partielles

La première approche proposée afin de calculer les dérivées partielles directement à partir d'une image CFA est vectorielle. Elle consiste à calculer les trois composantes de chaque dérivée partielle en chaque pixel. Pour cela, nous proposons cinq méthodes originales.

Les deux premières méthodes d'estimations des dérivées utilisent une simple différentiation (*Simple Derivative Estimation*, SDE) et les filtres de Deriche (*Deriche Derivative Estimation*, DDE) pour calculer d'abord une seule composante de dérivée partielle en chaque pixel. Puis elles estiment les deux composantes de dérivée partielle manquantes. La troisième méthode (*Smoothed component Estimation and Deriche Derivative*, SEDD) utilise les filtres de Deriche pour estimer d'abord les composantes lissées manquantes puis pour calculer les trois dérivées partielles en chaque pixel. Les quatrième et cinquième méthodes utilisent les filtres de Deriche décalés pour calculer les dérivées partielles (*Shifted Deriche Derivative*, SDD) ou les composantes lissées (*Shifted Deriche Smoothed components*, SDS) et ne reposent sur aucune estimation.

3.1 Estimation simple de la dérivée (SDE)

Nous allons décrire comment il est possible d'estimer les dérivées partielles par une simple différentiation de l'image I^{CFA} . Premièrement, nous calculons les dérivées partielles I_x^{CFA} et I_y^{CFA} de I^{CFA} en chaque pixel $P(x,y)$ en utilisant les mêmes approxima-

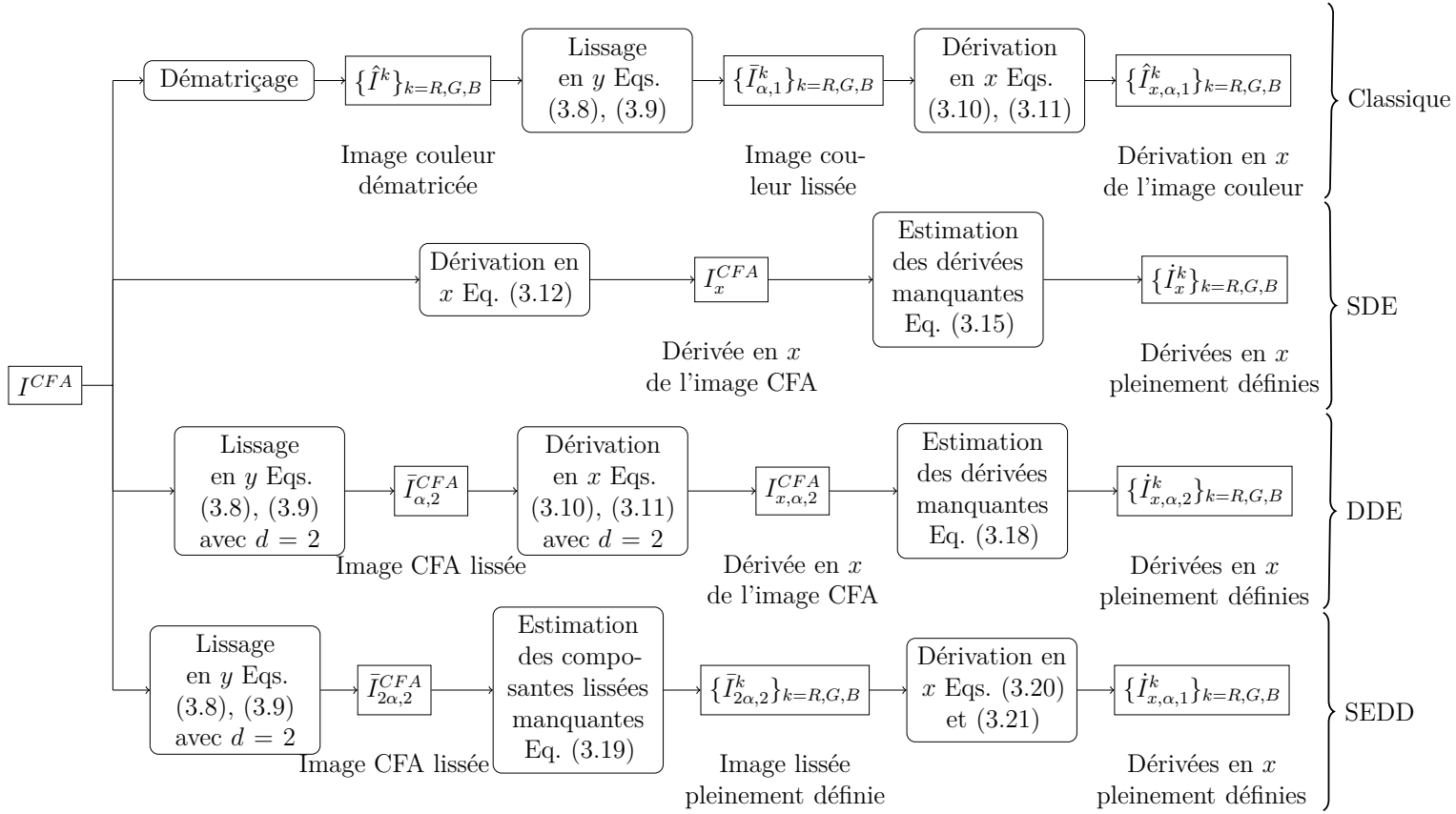


Figure 3.1 : Calcul de la dérivée partielle selon x par différentes méthodes vectorielles.

tions que dans l'équation (3.2) :

$$\begin{aligned}
 I_x^{CFA}(x,y) &= \frac{1}{2} \left(I^{CFA}(x+1,y) - I^{CFA}(x-1,y) \right), \\
 I_y^{CFA}(x,y) &= \frac{1}{2} \left(I^{CFA}(x,y+1) - I^{CFA}(x,y-1) \right).
 \end{aligned}
 \tag{3.12}$$

Ces dérivées sont obtenues directement à partir de I^{CFA} considérée ici comme une simple image en niveaux de gris. Elles peuvent également être considérées comme le résultat d'un multiplexage spatial des composantes des mêmes dérivées de l'image de référence \mathbf{I} spectralement sous-échantillonnée selon le CFA de Bayer. Puisque les deux pixels voisins horizontaux (resp., verticaux) de chaque pixel P appartiennent au même sous-ensemble S^k , leur différence de niveaux correspond à la dérivée horizontale (resp., verticale) I_x^k (resp., I_y^k) de la composante couleur k (cf. figure 3.2). Par exemple, les pixels voisins horizontaux du pixel $P(2,3) \in S^B$ dans la figure 3.2a appartiennent à S^G et leur différence de niveaux est égale à $2I_x^G$ comme le montre la figure 3.2b. Les pixels voisins verticaux appartiennent également S^G (cf. figure 3.2d) et leur différence de niveaux

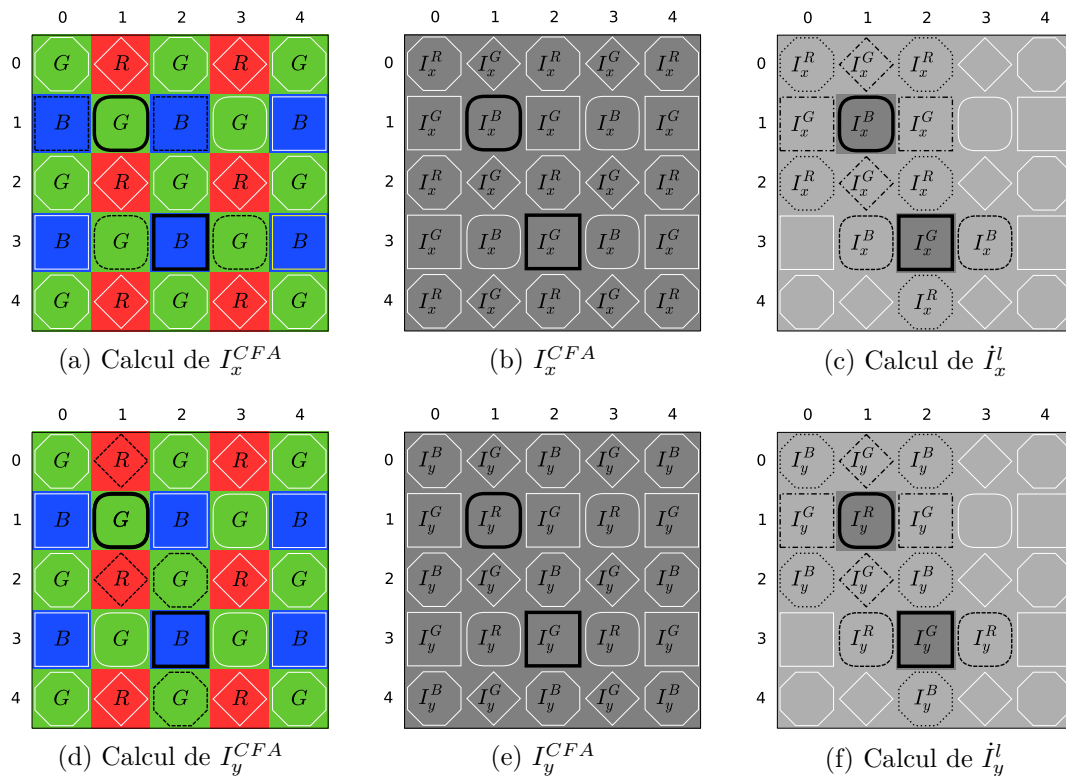


Figure 3.2 : : Dérivées partielles premières de l'image I^{CFA} (cf. équations (3.13)–(3.16)), avec deux pixels comme exemple (encadrés en gras). Les pixels voisins (cadres tiretés) dans les figures (a) et (d) sont utilisés pour calculer I_x^{CFA} et I_y^{CFA} . Les pixels voisins encadrés en lignes discontinues dans les figures (c) et (f) sont utilisés pour calculer les dérivées partielles manquantes.

est égale à $2I_y^G$ (cf. figure 3.2e). Les pixels de S^G doivent être considérés séparément selon qu'ils appartiennent à $S^{G,R}$ ou $S^{G,B}$. Par exemple, les voisins horizontaux (resp., verticaux) du pixel $P(1,1) \in S^{G,B}$ appartiennent à S^B (resp., S^R) et leur différence de niveaux est égale à $2I_x^B$ (resp., $2I_y^R$). Plus généralement, I_x^{CFA} (resp., I_y^{CFA}) correspond à une composante de \mathbf{I}_x (resp., \mathbf{I}_y) en chaque pixel P selon :

$$I_x^{CFA}(P) = \begin{cases} I_x^G(P) & \text{si } P \in S^R \cup S^B, \\ I_x^R(P) & \text{si } P \in S^{G,R}, \\ I_x^B(P) & \text{si } P \in S^{G,B}, \end{cases} \quad \text{et} \quad I_y^{CFA}(P) = \begin{cases} I_y^G(P) & \text{si } P \in S^R \cup S^B, \\ I_y^B(P) & \text{si } P \in S^{G,R}, \\ I_y^R(P) & \text{si } P \in S^{G,B}. \end{cases} \quad (3.13)$$

Afin d'obtenir les dérivées partielles pleinement définies $\dot{\mathbf{I}}_x$ et $\dot{\mathbf{I}}_y$ à partir de I^{CFA} , il nous faut estimer les deux dérivées partielles manquantes en chaque pixel. Notons ces composantes manquantes $\dot{I}_x^l(P)$ et $\dot{I}_y^l(P)$, $l \neq k$ quand $P \in S^k$. Nous pouvons alors

écrire :

$$\dot{\mathbf{I}}_x(P) = \begin{cases} \left(\dot{I}_x^R(P), I_x^{CFA}(P), \dot{I}_x^B(P) \right) & \text{si } P \in S^R \cup S^B, \\ \left(I_x^{CFA}(P), \dot{I}_x^G(P), \dot{I}_x^B(P) \right) & \text{si } P \in S^{G,R}, \\ \left(\dot{I}_x^R(P), \dot{I}_x^G(P), I_x^{CFA}(P) \right) & \text{si } P \in S^{G,B}, \end{cases} \quad (3.14)$$

et $\dot{\mathbf{I}}_y$ de la même façon. Nous calculons alors une composante de dérivée manquante \dot{I}_x^l (ou \dot{I}_y^l) en chaque pixel P par interpolation des valeurs des dérivées I_x^l (ou I_y^l) disponibles dans le voisinage de P . Ce voisinage dépend du sous-ensemble S^k auquel P appartient. Les quatre configurations de voisinage utilisés par cette interpolation sont définies dans la figure 2.1, et les dérivées partielles sont données par :

$$\dot{\mathbf{I}}_x(P) = \begin{cases} \left(\frac{1}{2} \sum_{Q \in \mathbb{N}^H(P)} I_x^R(Q), I_x^{CFA}(P), \frac{1}{2} \sum_{Q \in \mathbb{N}^V(P)} I_x^B(Q) \right) & \text{si } P \in S^R, \\ \left(I_x^{CFA}(P), \frac{1}{4} \sum_{P \in \mathbb{N}^+(P)} I_x^G(P), \frac{1}{4} \sum_{P \in \mathbb{N}^\times(P)} I_x^B(P) \right) & \text{si } P \in S^{G,R}, \\ \left(\frac{1}{4} \sum_{Q \in \mathbb{N}^\times(P)} I_x^R(Q), \frac{1}{4} \sum_{Q \in \mathbb{N}^+(P)} I_x^G(Q), I_x^{CFA}(P) \right) & \text{si } P \in S^{G,B}, \\ \left(\frac{1}{2} \cdot \sum_{Q \in \mathbb{N}^V(P)} I_x^R(Q), I_x^{CFA}(P), \frac{1}{2} \sum_{Q \in \mathbb{N}^H(P)} I_x^B(Q) \right) & \text{si } P \in S^B, \end{cases} \quad (3.15)$$

et :

$$\dot{\mathbf{I}}_y(P) = \begin{cases} \left(\frac{1}{2} \sum_{Q \in \mathbb{N}^V(P)} I_y^R(Q), I_y^{CFA}(P), \frac{1}{2} \sum_{Q \in \mathbb{N}^H(P)} I_y^B(Q) \right) & \text{si } P \in S^R, \\ \left(\frac{1}{4} \sum_{Q \in \mathbb{N}^\times(P)} I_y^R(Q), \frac{1}{4} \sum_{Q \in \mathbb{N}^+(P)} I_y^G(Q), I_y^{CFA}(P) \right) & \text{si } P \in S^{G,R}, \\ \left(I_y^{CFA}(P), \frac{1}{4} \sum_{Q \in \mathbb{N}^+(P)} I_y^G(Q), \frac{1}{4} \sum_{Q \in \mathbb{N}^\times(P)} I_y^B(Q) \right) & \text{si } P \in S^{G,B}, \\ \left(\frac{1}{2} \sum_{Q \in \mathbb{N}^H(P)} I_y^R(Q), I_y^{CFA}(P), \frac{1}{2} \sum_{Q \in \mathbb{N}^V(P)} I_y^B(Q) \right) & \text{si } P \in S^B. \end{cases} \quad (3.16)$$

Par exemple (*cf.* figure 3.2c), les quatre pixels voisins diagonaux de $P(1,1) \in S^{G,B}$ sont utilisés pour le calcul de $\dot{I}_x^R(P)$ et ses quatre plus proches voisins sont utilisés pour le calcul de $\dot{I}_x^G(P)$. Les dérivées partielles selon x qui manquent en $P(2,3) \in S^B$ sont interpolées à partir des deux pixels voisins (*i.e.*, voisinage vertical pour $\dot{I}_x^R(P)$ et voisinage horizontal pour $\dot{I}_x^B(P)$).

Cette méthode (SDE) évite le dématricage mais estime les 2 dérivées partielles manquantes en chaque pixel. Elle est schématisée sur la seconde ligne de la figure 3.1.

3.2 Estimation des dérivées par filtre de Deriche (DDE)

Comme la méthode SDE ne lisse pas l'image CFA préalablement à sa dérivation, les résultats sont très sensibles au bruit. Afin d'atténuer cette sensibilité au bruit, nous proposons d'adapter à l'image CFA le filtre optimal de Deriche présenté dans la section 2.3. Notre méthode (DDE) revient à appliquer les équations de lissage (3.8) et (3.9) et de dérivation (3.10) et (3.11) sur le même sous-ensemble S^k des pixels où est disponible la composante couleur k dans l'image CFA en fixant $d = 2$ (*cf.* figures 2.6a, 2.6b et 2.6c). Cela fournit la dérivée partielle de Deriche $I_{x,\alpha,2}^{CFA}$ de l'image CFA, qui comporte une seule composante de dérivée partielle par pixel (celle de la composante couleur disponible en ce pixel dans l'image CFA).

Comme dans la méthode SDE (*cf.* équation (3.14)), nous avons ensuite besoin d'estimer en chaque pixel $P \in S^k$ les deux composantes manquantes de dérivée partielle $\dot{I}_{x,\alpha,2}^l(P)$, $l \neq k$, afin d'obtenir la dérivée partielle pleinement définie $\dot{\mathbf{I}}_{x,\alpha,2}$:

$$\dot{\mathbf{I}}_{x,\alpha,2}(P) = \begin{cases} \left(I_{x,\alpha,2}^{CFA}(P), \dot{I}_{x,\alpha,2}^G(P), \dot{I}_{x,\alpha,2}^B(P) \right) & \text{si } P \in S^R, \\ \left(\dot{I}_{x,\alpha,2}^R(P), I_{x,\alpha,2}^{CFA}(P), \dot{I}_{x,\alpha,2}^B(P) \right) & \text{si } P \in S^G, \\ \left(\dot{I}_{x,\alpha,2}^R(P), \dot{I}_{x,\alpha,2}^G(P), I_{x,\alpha,2}^{CFA}(P) \right) & \text{si } P \in S^B. \end{cases} \quad (3.17)$$

Nous estimons ces composantes de dérivée partielle manquantes en chaque pixel P par interpolation bilinéaire :

$$\dot{\mathbf{I}}_{x,\alpha,2}(P) = \begin{cases} \left(I_{x,\alpha,2}^{CFA}(P), \frac{1}{4} \sum_{Q \in \mathbb{N}^+(P)} I_{x,\alpha,2}^{CFA}(Q), \frac{1}{4} \sum_{Q \in \mathbb{N}^\times(P)} I_{x,\alpha,2}^{CFA}(Q) \right) & \text{si } P \in S^R \\ \left(\frac{1}{2} \sum_{Q \in \mathbb{N}^H(P)} I_{x,\alpha,2}^{CFA}(Q), I_{x,\alpha,2}^{CFA}(P), \frac{1}{2} \sum_{Q \in \mathbb{N}^V(P)} I_{x,\alpha,2}^{CFA}(Q) \right) & \text{si } P \in S^{G,R}, \\ \left(\frac{1}{2} \sum_{Q \in \mathbb{N}^V(P)} I_{x,\alpha,2}^{CFA}(Q), I_{x,\alpha,2}^{CFA}(P), \frac{1}{2} \sum_{Q \in \mathbb{N}^H(P)} I_{x,\alpha,2}^{CFA}(Q) \right) & \text{si } P \in S^{G,B}, \\ \left(\frac{1}{4} \sum_{Q \in \mathbb{N}^\times(P)} I_{x,\alpha,2}^{CFA}(Q), \frac{1}{4} \sum_{Q \in \mathbb{N}^+(P)} I_{x,\alpha,2}^{CFA}(Q), I_{x,\alpha,2}^{CFA}(P) \right) & \text{si } P \in S^B. \end{cases} \quad (3.18)$$

Les dérivées partielles de Deriche verticales $I_{y,\alpha,2}^{CFA}$ et $\dot{\mathbf{I}}_{y,\alpha,2}$ sont calculées de la même manière que $I_{x,\alpha,2}^{CFA}$ et $\dot{\mathbf{I}}_{x,\alpha,2}$, mais en utilisant les directions perpendiculaires. La troisième ligne de la figure 3.1 montre les différentes étapes de la méthode DDE.

3.3 Estimation des composantes lissées par filtre de Deriche (SEDD)

Contrairement à la méthode DDE, la méthode SEDD estime les composantes lissées manquantes plutôt que les composantes manquantes des dérivées partielles. Cette méthode se décompose en trois étapes. Dans la première étape, l'image CFA est lissée selon la direction verticale (resp., horizontale) avec le filtre de lissage de Deriche adapté à ce type d'image. Ensuite, l'image couleur lissée est calculée en estimant les deux composantes couleur lissées manquantes en chaque pixel. Finalement, le filtre dérivateur de Deriche est appliqué horizontalement (resp., verticalement) en chaque composante couleur lissée.

Cette méthode utilise donc deux filtres de Deriche qui sont définis par le même paramètre α , à savoir le filtre de lissage (vertical) $\mathcal{L}_\alpha(y)$ (cf. équation (3.6)) et le filtre dérivateur (horizontal) $\mathcal{D}_\alpha(x)$ (cf. équation (3.4)) mais avec une distance d entre pixels voisins différente. Or, appliquer $\mathcal{L}_{2\alpha}$ (ou $\mathcal{D}_{2\alpha}$) à l'image I^{CFA} de taille $M \times N$ (cf. équations (3.8) et (3.9)) avec $d = 2$ est équivalent à appliquer \mathcal{L}_α (ou \mathcal{D}_α) en utilisant les mêmes équations avec $d = 1$ sur les images de taille $M/2 \times N/2$ résultant du sous-échantillonnage de I^{CFA} selon S^R , $S^{G,R}$, $S^{G,B}$ et S^B .

La méthode proposée lisse d'abord I^{CFA} selon y en appliquant les équations (3.8) et (3.9) avec $d = 2$, ce qui fournit $\bar{I}_{2\alpha,2}^{CFA} = \bar{I}_{2\alpha,2}^{CFA-} + \bar{I}_{2\alpha,2}^{CFA+}$. Ensuite elle estime les composantes lissées manquantes dans $\bar{I}_{2\alpha,2}^{CFA}$, puis dérive finalement le résultat selon x en utilisant les équations (3.10) et (3.11) avec $d = 1$.

Concrètement, les 3 composantes lissées sont calculées par interpolation des niveaux manquants en chaque pixel P par :

$$\dot{\mathbf{I}}_{2\alpha,2}(P) = \begin{cases} \left(\bar{I}_{2\alpha,2}^{CFA}(P), \frac{1}{2} \sum_{Q \in \mathcal{N}^V(P)} \bar{I}_{2\alpha,2}^{CFA}(Q), \frac{1}{4} \sum_{Q \in \mathcal{N}^\times(P)} \bar{I}_{2\alpha,2}^{CFA}(Q) \right) & \text{si } P \in S^R \\ \left(\frac{1}{2} \sum_{Q \in \mathcal{N}^H(P)} \bar{I}_{2\alpha,2}^{CFA}(Q), \bar{I}_{2\alpha,2}^{CFA}(P), \frac{1}{2} \sum_{Q \in \mathcal{N}^V(P)} \bar{I}_{2\alpha,2}^{CFA}(Q) \right) & \text{si } P \in S^{G,R}, \\ \left(\frac{1}{2} \sum_{Q \in \mathcal{N}^V(P)} \bar{I}_{2\alpha,2}^{CFA}(Q), \bar{I}_{2\alpha,2}^{CFA}(P), \frac{1}{2} \sum_{Q \in \mathcal{N}^H(P)} \bar{I}_{2\alpha,2}^{CFA}(Q) \right) & \text{si } P \in S^{G,B}, \\ \left(\frac{1}{4} \sum_{Q \in \mathcal{N}^\times(P)} \bar{I}_{2\alpha,2}^{CFA}(Q), \frac{1}{2} \sum_{Q \in \mathcal{N}^V(P)} \bar{I}_{2\alpha,2}^{CFA}(Q), \bar{I}_{2\alpha,2}^{CFA}(P) \right) & \text{si } P \in S^B, \end{cases} \quad (3.19)$$

et l'implémentation récursive de l'étape finale de dérivation est donnée par :

$$\mathbf{I}_{x,\alpha,1}^-(x,y) = a_5 \dot{\mathbf{I}}_{2\alpha,2}(x,y) + a_6 \dot{\mathbf{I}}_{2\alpha,2}(x-1,y) + b_1 \mathbf{I}_{x,\alpha,1}^-(x-1,y) + b_2 \mathbf{I}_{x,\alpha,1}^-(x-2,y), \quad (3.20)$$

$$\mathbf{I}_{x,\alpha,1}^+(x,y) = a_7 \dot{\mathbf{I}}_{2\alpha,2}(x+1,y) + a_8 \dot{\mathbf{I}}_{2\alpha,2}(x+2,y) + b_1 \mathbf{I}_{x,\alpha,1}^+(x+1,y) + b_2 \mathbf{I}_{x,\alpha,1}^+(x+2,y). \quad (3.21)$$

Ceci fournit la dérivée partielle de Deriche horizontale $\dot{\mathbf{I}}_{x,\alpha,1} = c (\mathbf{I}_{x,\alpha,1}^- + \mathbf{I}_{x,\alpha,1}^+)$, et $\dot{\mathbf{I}}_{y,\alpha,1}$ est obtenue en utilisant les équations (3.20) et (3.21) avec les directions perpendiculaires.

Cette méthode (SEDD) estime donc d'abord une image lissée pleinement définie afin de pouvoir appliquer ensuite le filtre de Deriche avec $d = 1$. Elle est schématisée par la dernière ligne de la figure 3.1.

3.4 Filtres de Deriche décalés

Dans cette section, nous proposons d'éviter l'estimation des composantes lissées ou des composantes de dérivée partielle. En effet, nous calculons directement des dérivées partielles pleinement définies à partir de l'image CFA. Pour ce faire, nous proposons de calculer les trois composantes des dérivées partielles en chaque pixel en utilisant les filtres décalés de Deriche définis par Montesinos et Dattenny [56] et Fiorio [28]. Ces filtres décalés ou filtres inter-pixels sont les filtres de lissage et de dérivation de Deriche décalés spatialement d'un demi-pixel [28].

Le filtre de lissage de Deriche décalé selon y est donné par :

$$\mathcal{L}_{\alpha,\frac{1}{2}}(y) = m_{\frac{1}{2}} \left(\alpha \left| y - \frac{1}{2} \right| + 1 \right) e^{-\alpha |y - \frac{1}{2}|}. \quad (3.22)$$

Le coefficient de normalisation $m_{\frac{1}{2}}$ est calculé pour que l'énergie du filtre soit unitaire (*cf.* annexe A.2.1) :

$$m_{\frac{1}{2}} = \frac{1}{2 e^{\frac{\alpha}{2}}} \left[\frac{(1 - e^{-\alpha})^2}{e^{-\alpha} (1 + \frac{\alpha}{2}) - e^{-2\alpha} (1 - \frac{\alpha}{2})} \right]. \quad (3.23)$$

Le filtre décalé de dérivation de Deriche selon x est donné par :

$$\mathcal{D}_{\alpha,\frac{1}{2}}(x) = -m'_{\frac{1}{2}} \left(x - \frac{1}{2} \right) \alpha^2 e^{-\alpha |x - \frac{1}{2}|}, \quad (3.24)$$

et son coefficient de normalisation par (*cf.* annexe A.3.1) :

$$m'_{\frac{1}{2}} = \frac{2(1 - e^{-\alpha})^2}{\alpha^2 e^{\frac{\alpha}{2}} (e^{-\alpha} + e^{-2\alpha})}. \quad (3.25)$$

Comme les filtres de Deriche sont séparables et peuvent être implémentés de manière récursive, les filtres décalés le sont aussi. Ces filtres inter-pixel peuvent être implémentés de manière récursive à l'aide des équations de récurrence (3.26) à (3.29).

Les équations de récurrence du filtre décalé de lissage selon y se déduisent de l'équation (3.22) :

$$\bar{I}_{\alpha,d}^{V-}(x,y) = C_0 I(x,y-d) + aC_0 I(x,y-2d) + b\bar{I}_{\alpha,d}^{V-}(x,y-d) + c\bar{I}_{\alpha,d}^{V-}(x,y-2d), \quad (3.26)$$

$$\bar{I}_{\alpha,d}^{V+}(x,y) = C_0 I(x,y) + aC_0 I(x,y+d) + b\bar{I}_{\alpha,d}^{V+}(x,y+d) + c\bar{I}_{\alpha,d}^{V+}(x,y+2d), \quad (3.27)$$

où $C_0 = m_{\frac{1}{2}}(1 + \frac{\alpha}{2})e^{-\frac{\alpha}{2}}$, $a = \frac{\alpha-2}{\alpha+2}e^{-\alpha}$, $b = 2e^{-\alpha}$ et $c = -e^{-2\alpha}$ sont des coefficients de pondération qui dépendent du seul paramètre α (*cf.* annexe A). L'image résultant de ce lissage décalé verticalement est alors donnée par : $\bar{I}_{\alpha,d}^V = \bar{I}_{\alpha,d}^{V-} + \bar{I}_{\alpha,d}^{V+}$. Notons que les niveaux manquants des pixels de bord de I , $\bar{I}_{\alpha,d}^{V-}$, et $\bar{I}_{\alpha,d}^{V+}$ sont fixés à 0 dans les équations (3.26) et (3.27).

La dérivée partielle selon x décalée horizontalement et verticalement, $I_{x,\alpha,d}^{HV}$, est obtenue en dérivant $\bar{I}_{\alpha,d}^V$ horizontalement. Ceci est réalisé de manière récursive grâce au parcours de cette image de gauche à droite et de droite à gauche :

$$I_{x,\alpha,d}^{HV-}(x,y) = C_1 \bar{I}_{\alpha,d}(x-d,y) + d' \cdot C_1 \bar{I}_{\alpha,d}(x-2d,y) + bI_{x,\alpha,d}^{HV-}(x-d,y) + cI_{x,\alpha,d}^{HV-}(x-2d,y), \quad (3.28)$$

$$I_{x,\alpha,d}^{HV+}(x,y) = -C_1 \bar{I}_{\alpha,d}(x,y) - d' \cdot C_1 \bar{I}_{\alpha,d}(x+d,y) + bI_{x,\alpha,d}^{HV+}(x+d,y) + cI_{x,\alpha,d}^{HV+}(x+2d,y), \quad (3.29)$$

où $C_1 = -m'_{\frac{1}{2}} \frac{\alpha^2}{2} e^{-\frac{\alpha}{2}}$, $d' = e^{-\alpha}$, $b = 2e^{-\alpha}$ et $c = -e^{-2\alpha}$ (*cf.* annexe A). La dérivée partielle selon x décalée horizontalement et verticalement est alors donnée par $I_{x,\alpha,d}^{HV} = I_{x,\alpha,d}^{HV-} + I_{x,\alpha,d}^{HV+}$. La dérivée partielle selon y décalée horizontalement et verticalement, $I_{y,\alpha,d}^{HV}$, est calculée de la même manière mais dans les directions perpendiculaires.

La figure 3.3 illustre le décalage provoqué par ces filtres. La réponse du filtre de lissage décalé verticalement (*cf.* équations (3.26) et (3.27) avec $d = 1$) en un pixel P est localisée à la frontière verticale entre P et son voisin de dessous que Fiorio [28] a appelée « lignel ». En dérivant selon x l'image résultante lissée avec le filtre de Deriche décalé horizontalement (*cf.* équations (3.29) et (3.28) avec $d = 1$), les dérivées partielles sont localisées sur la frontière diagonale entre les deux pixels $P(x,y)$ et $Q(x+1,y+1)$ appelée « pointel » (*cf.* figure 3.3).

Nous proposons d'appliquer sur l'image CFA les filtres décalés définis par les équations (3.26) à (3.28) avec $d = 2$. La figure 3.4 illustre que le lissage provoque un décalage

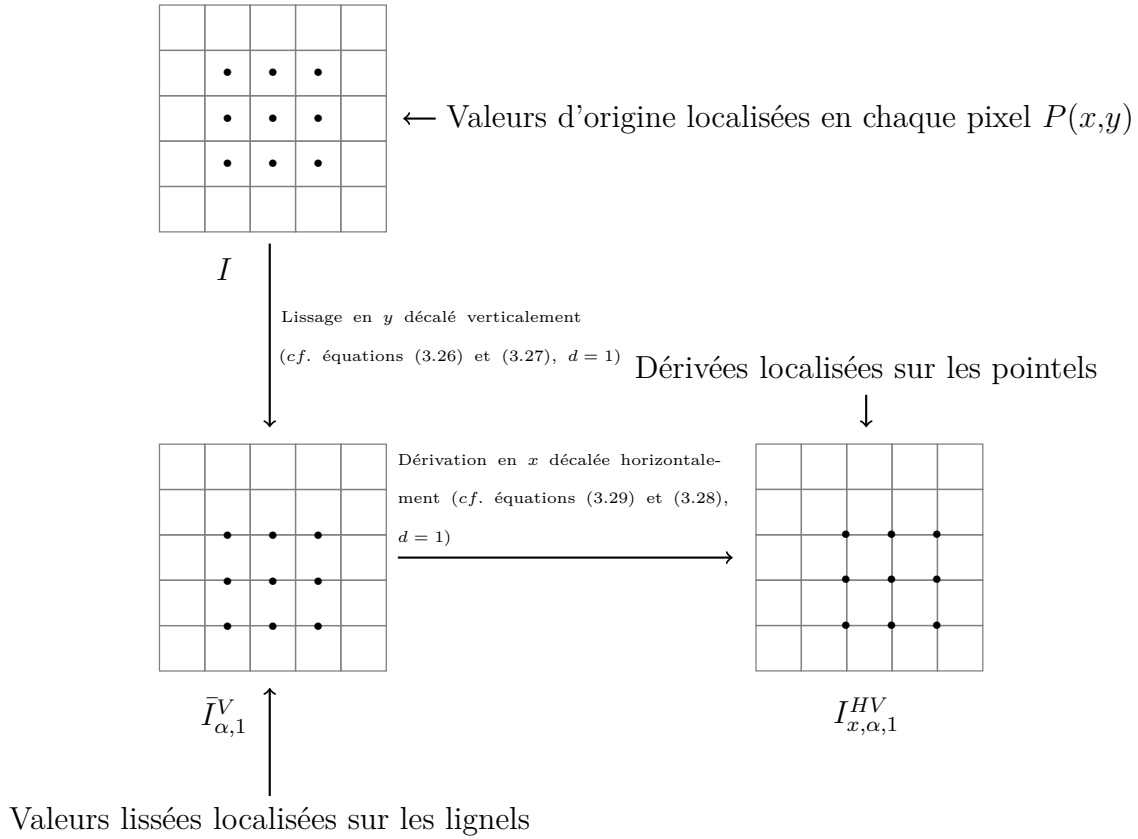


Figure 3.3 : : Décalage provoqué par un filtre de lissage décalé verticalement et un filtre de dérivation décalé horizontalement sur une image I avec $d = 1$.

vertical d'un pixel, la réponse du filtre de lissage décalé en $P(x,y) \in I^{CFA}$ est localisée en $P'(x,y + 1)$. La dérivation provoque un décalage horizontal, la dérivée partielle estimée en P' étant alors localisée en $Q(x + 1,y + 1)$.

Nous proposons donc de calculer les trois composantes des dérivées partielles directement à partir de l'image I^{CFA} en utilisant les filtres de Deriche décalés.

3.5 Dérivation par filtre de Deriche décalé (SDD)

Comme dans la méthode DDE (cf. section 3.2), nous commençons par calculer la dérivée partielle de la composante couleur disponible en chaque pixel en appliquant les filtres de lissage et de dérivation de Deriche sur I^{CFA} avec $d = 2$ (cf. équations (3.8) à (3.11)). Mais dans SDD, nous calculons les deux composantes de dérivée partielle manquantes en chaque pixel grâce aux filtres décalés. Cette méthode est schématisée dans la figure 3.5b.

Pour obtenir par exemple la dérivée partielle pleinement définie $\mathbf{I}_{x,\alpha,2}$ à partir de I^{CFA} , nous avons besoin de calculer les dérivées partielles de I^{CFA} décalées horizontalement

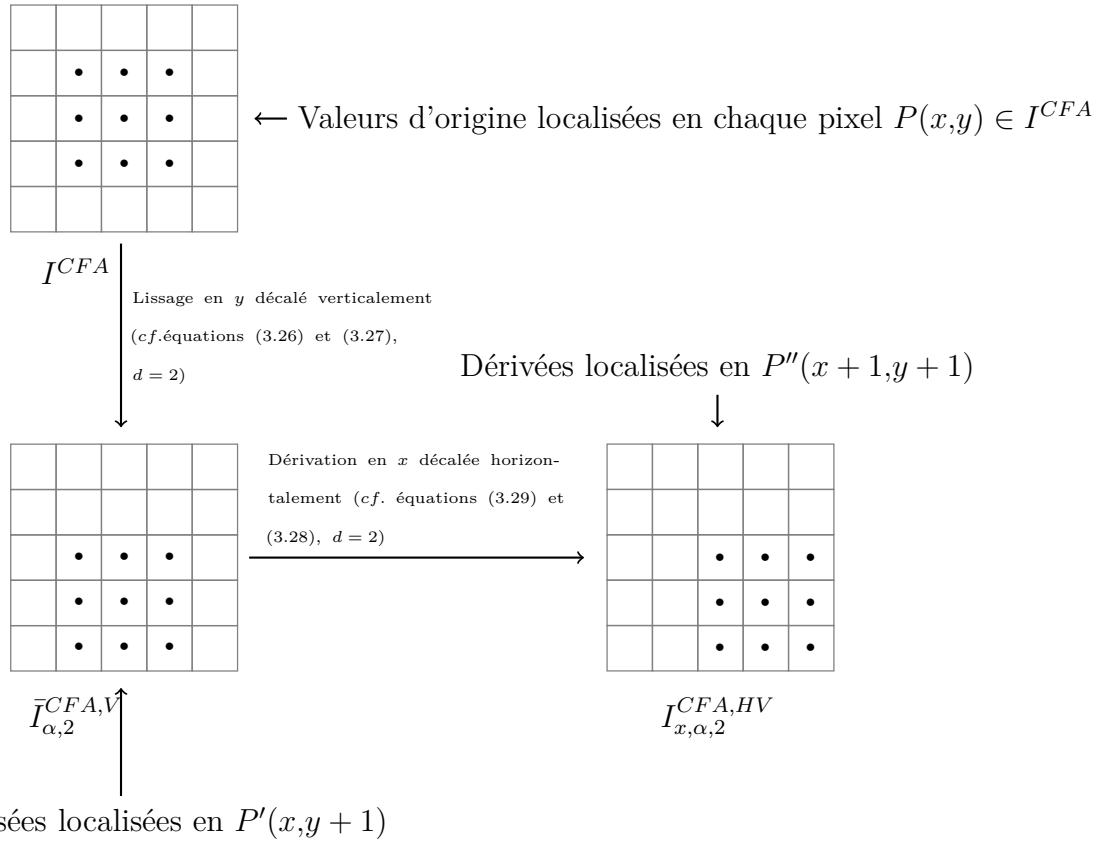
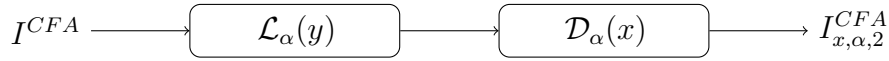


Figure 3.4 : : Décalage provoqué par un lissage décalé verticalement et une dérivation décalée horizontalement dans une image I^{CFA} avec $d = 2$.

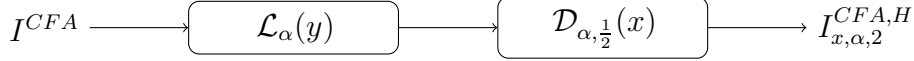
(et/ou verticalement) en utilisant les différents schémas illustrés sur la figure 3.5a. Ainsi, la dérivée partielle de I^{CFA} selon x décalée horizontalement notée $I_{x,\alpha,2}^{CFA,H}$ est obtenue en lissant I^{CFA} selon y avec le filtre de lissage de Deriche (cf. équations (3.8) et (3.9) avec $d = 2$). L'image résultante est dérivée selon x en précision inter-pixel avec le filtre de Deriche décalé horizontalement (cf. équations (3.28) et (3.29) avec $d = 2$).

En se référant aux différentes dérivées illustrées sur la figure 3.5b, nous pouvons écrire :

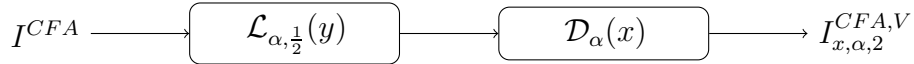
$$\mathbf{I}_{x,\alpha,2}(P) = \begin{cases} \left(I_{x,\alpha,2}^{CFA}(P), I_{x,\alpha,2}^{CFA,H}(P), I_{x,\alpha,2}^{CFA,HV}(P) \right) & \text{si } P \in S^R, \\ \left(I_{x,\alpha,2}^{CFA,H}(P), I_{x,\alpha,2}^{CFA}(P), I_{x,\alpha,2}^{CFA,V}(P) \right) & \text{si } P \in S^{G,R}, \\ \left(I_{x,\alpha,2}^{CFA,V}(P), I_{x,\alpha,2}^{CFA}(P), I_{x,\alpha,2}^{CFA,H}(P) \right) & \text{si } P \in S^{G,B}. \\ \left(I_{x,\alpha,2}^{CFA,HV}, I_{x,\alpha,2}^{CFA,H}(P), I_{x,\alpha,2}^{CFA}(P) \right) & \text{si } P \in S^B. \end{cases} \quad (3.30)$$



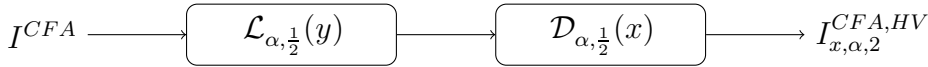
Lissage en y et dérivation en x (filtres de Deriche classiques)



Lissage en y et dérivation décalée en x

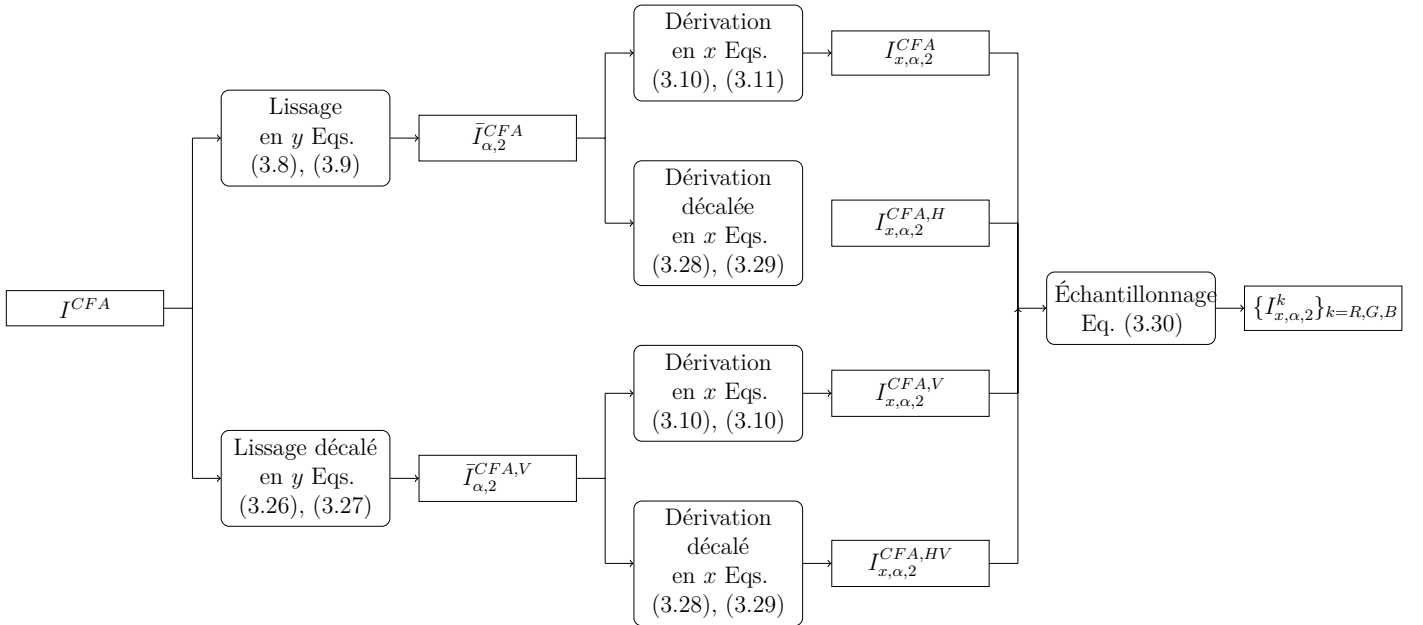


Lissage décalé en y et dérivation en x



Lissage décalé en x et dérivation décalée en y

(a) Calcul des dérivées partielles de I^{CFA} en fonction des filtres de Deriche utilisés (classiques ou décalés).



(b) Calcul des trois composantes de la dérivée partielle selon x à partir de I^{CFA} .

Figure 3.5 : Calcul des trois composantes de la dérivée partielle selon x à partir de I^{CFA} par la méthode SDD.

Par un raisonnement analogue, la dérivée selon y s'exprime par :

$$\mathbf{I}_{y,\alpha,2}(P) = \begin{cases} (I_{y,\alpha,2}^{CFA}(P), I_{y,\alpha,2}^{CFA,V}(P), I_{y,\alpha,2}^{CFA,HV}(P)) & \text{si } P \in S^R, \\ (I_{y,\alpha,2}^{CFA,H}(P), I_{y,\alpha,2}^{CFA}(P), I_{y,\alpha,2}^{CFA,V}(P)) & \text{si } P \in S^{G,R}, \\ (I_{y,\alpha,2}^{CFA,V}(P), I_{y,\alpha,2}^{CFA}(P), I_{y,\alpha,2}^{CFA,H}(P)) & \text{si } P \in S^{G,B}, \\ (I_{y,\alpha,2}^{CFA,HV}, I_{y,\alpha,2}^{CFA,V}(P), I_{y,\alpha,2}^{CFA}(P)) & \text{si } P \in S^B. \end{cases} \quad (3.31)$$

Notons que dans le cas où deux dérivées partielles calculées avec deux décalage différents (horizontal ou vertical) sont disponibles, la direction du décalage choisie doit correspondre à la direction de dérivation.

Prenons par exemple le pixel $P(1,1) \in S^{G,B}$ dans la figure 3.6a. La composante G de la dérivée partielle selon x en P où cette composante est disponible est calculée avec les filtres de Deriche classiques (*cf.* équations (3.9) à (3.11)) comme représenté sur la figure 3.6b. La composante R de la dérivée partielle selon x en P est obtenue en lissant l'image CFA avec le filtre de Deriche décalé verticalement (*cf.* équations (3.26) et (3.27)) et l'image lissée est dérivée avec le filtres de Deriche classique (*cf.* équations (3.10) et (3.11)) pour produire le résultat représenté sur la figure 3.6d. Finalement, la composante B de la dérivée partielle selon x en P est obtenue en lissant l'image CFA avec le filtre de Deriche classique (*cf.* équations (3.8) et (3.9)) et l'image lissée est dérivée avec les filtres de Deriche décalés horizontalement (*cf.* équations (3.28) et (3.29)) pour produire le résultat représenté sur la figure 3.6c.

Pour le pixel $P(2,3) \in S^B$ dans la figure 3.6a, la composante B de la dérivée partielle selon x où cette composante est disponible est calculée avec les filtres de Deriche classiques (*cf.* figure 3.6b). La composante R de la dérivée partielle selon x en P est obtenue en lissant l'image CFA avec le filtre de Deriche décalé verticalement (*cf.* équations (3.26) et (3.27)) et l'image lissée est dérivée avec le filtre de Deriche décalé horizontalement (*cf.* équations (3.28) et (3.29)) pour produire le résultat représenté sur la figure 3.6e. Finalement, la composante G de la dérivée partielle selon x en P est obtenue en utilisant le filtre de lissage classique de Deriche suivi du filtre de dérivation décalé horizontalement pour produire le résultat représenté sur la figure 3.6c.

3.6 Lissage et dérivation par filtres de Deriche décalés (SDS)

Suivant l'idée de la méthode SEDD, nous proposons dans cette méthode de calculer les trois composantes lissées en chaque pixel de l'image CFA afin d'obtenir une image lissée pleinement définie. Cette image est ensuite dérivée avec le filtre de Deriche classique (*cf.* équations (3.10) et (3.11) avec $d = 1$). Nous nous basons sur un lissage bidimensionnel afin de calculer les trois composantes couleur lissées en chaque pixel. Ce choix est dicté par le calcul des composantes lissées en chaque pixel $P \in S^R \cup S^B$. En effet, afin de calculer la composante B lissée en $P \in S^R$ (ou inversement), nous avons besoin d'un lissage selon x décalé horizontalement et le résultat est lissé avec le filtre décalé selon y . Cette méthode est schématisée dans la figure 3.7a. La figure 3.7b illustre le calcul des

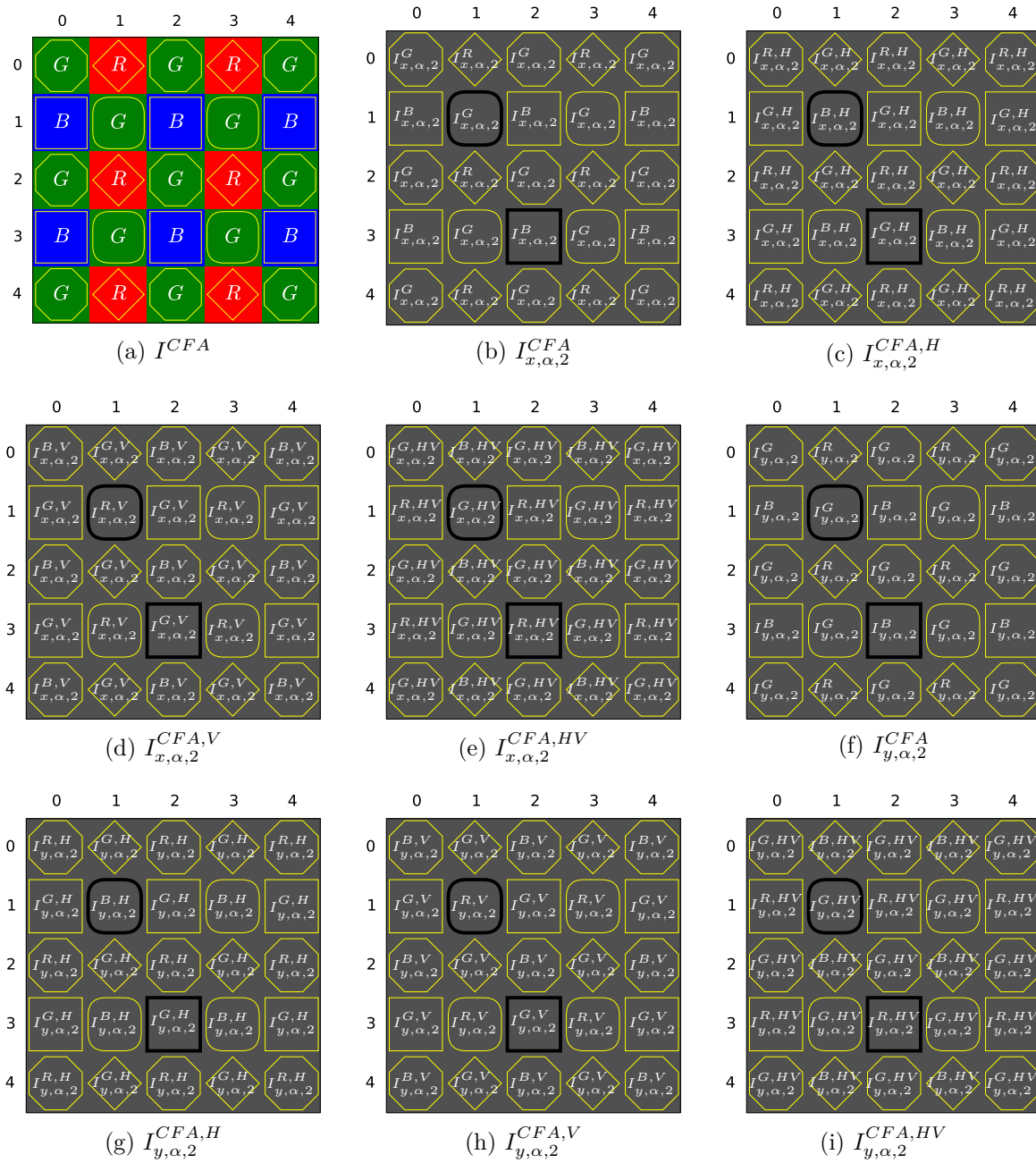


Figure 3.6 : Calcul des dérivées partielles de l'image I^{CFA} en utilisant les filtres de Deriche classiques et les filtres de Deriche décalés, avec deux pixels comme exemple (lignes continues en gras).

trois composantes de la dérivée partielle selon x à partir de I^{CFA} avec la méthode SDS.

Par exemple, afin de calculer les composantes lissées au pixel $P(1,1) \in S^G$ dans la figure 3.8a, la composante lissée de la composante G disponible est calculée avec les filtres de lissage de Deriche classiques selon y (*cf.* équations (3.8) e (3.9)) et le résultat est lissé selon x avec les même équations (*cf.* figure 3.8b). La composante R (resp., B) lissée en $P(1,1)$ est obtenue en lissant I^{CFA} selon x (resp., selon y) avec le filtre de Deriche classique et le résultat est lissé selon y (resp., selon x) avec le filtre décalé verticalement (resp., horizontalement) (*cf.* figures 3.8c et 3.8d). Au pixel $P(2,3) \in S^B$, la composante R lissée est obtenue en réalisant successivement un lissage selon x décalé horizontalement et un autre selon y décalé verticalement (*cf.* figure 3.8e). La composante G lissée est obtenue par un lissage selon x décalé verticalement et le résultat est lissé horizontalement avec le filtre de Deriche classique (*cf.* figure 3.8e). La composante B lissée est directement obtenue avec les filtres de Deriche classiques (*cf.* figure 3.7).

Plus précisément, trois composantes lissées sont calculées en chaque pixel en utilisant les différents schémas illustrés sur la figure 3.7a :

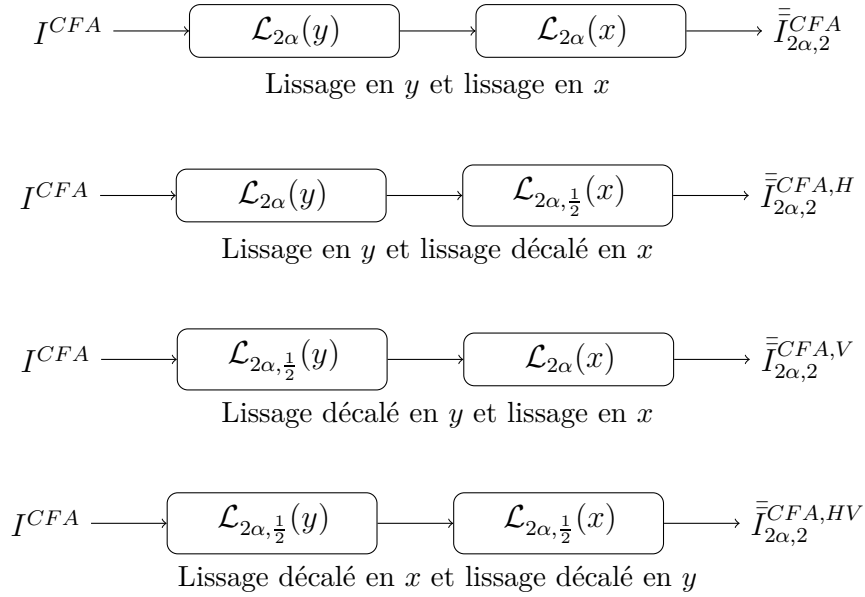
$$\bar{\bar{\mathbf{I}}}_{2\alpha,2}(P) = \begin{cases} \left(\bar{\bar{I}}_{2\alpha,2}^{CFA}(P), \bar{\bar{I}}_{2\alpha,2}^{CFA,V}(P), \bar{\bar{I}}_{2\alpha,2}^{CFA,HV}(P) \right) & \text{si } P \in S^R, \\ \left(\bar{\bar{I}}_{2\alpha,2}^{CFA,H}(P), \bar{\bar{I}}_{2\alpha,2}^{CFA}(P), \bar{\bar{I}}_{2\alpha,2}^{CFA,V}(P) \right) & \text{si } P \in S^{G,R}, \\ \left(\bar{\bar{I}}_{2\alpha,2}^{CFA,V}(P), \bar{\bar{I}}_{2\alpha,2}^{CFA}(P), \bar{\bar{I}}_{2\alpha,2}^{CFA,H}(P) \right) & \text{si } P \in S^{G,B}. \\ \left(\bar{\bar{I}}_{2\alpha,2}^{CFA,HV}, \bar{\bar{I}}_{2\alpha,2}^{CFA,V}(P), \bar{\bar{I}}_{2\alpha,2}^{CFA}(P) \right) & \text{si } P \in S^B. \end{cases} \quad (3.32)$$

Le résultat est dérivé selon x en utilisant les équations (3.10) et (3.11) avec $d = 1$ et l'implémentation récursive de l'étape finale de dérivation est donnée par :

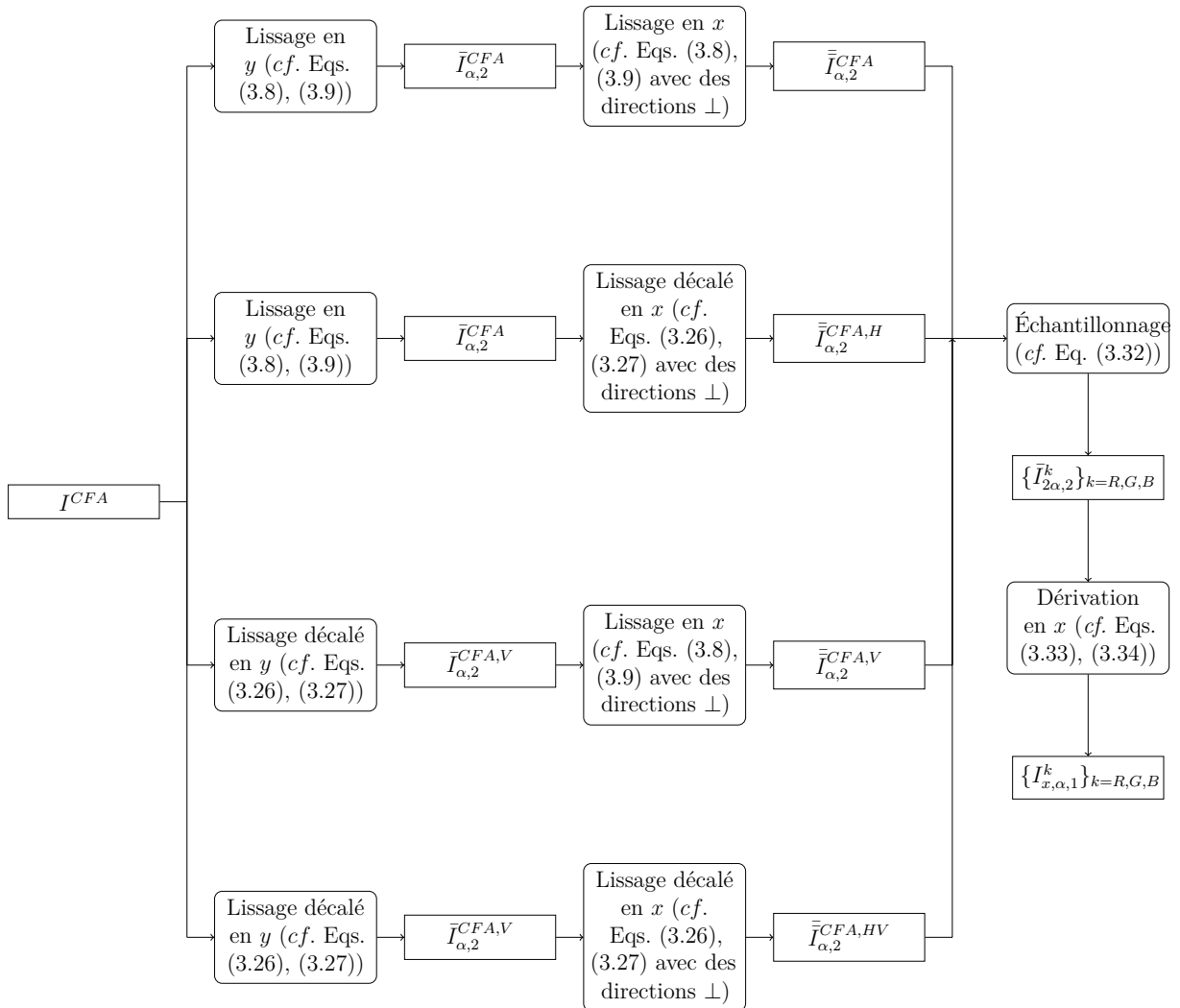
$$\mathbf{I}_{x,\alpha,1}^-(x,y) = a_5 \bar{\bar{\mathbf{I}}}_{2\alpha,2}(x,y) + a_6 \bar{\bar{\mathbf{I}}}_{2\alpha,2}(x-1,y) + b_1 \mathbf{I}_{x,\alpha,1}^-(x-1,y) + b_2 \mathbf{I}_{x,\alpha,1}^-(x-2,y), \quad (3.33)$$

$$\mathbf{I}_{x,\alpha,1}^+(x,y) = a_7 \bar{\bar{\mathbf{I}}}_{2\alpha,2}(x+1,y) + a_8 \bar{\bar{\mathbf{I}}}_{2\alpha,2}(x+2,y) + b_1 \mathbf{I}_{x,\alpha,1}^+(x+1,y) + b_2 \mathbf{I}_{x,\alpha,1}^+(x+2,y). \quad (3.34)$$

Ceci fournit la dérivée partielle de Deriche horizontale $\mathbf{I}_{x,\alpha,1} = c \left(\mathbf{I}_{x,\alpha,1}^- + \mathbf{I}_{x,\alpha,1}^+ \right)$, et $\mathbf{I}_{y,\alpha,1}$ est obtenue en utilisant les équations (3.33) et (3.34) avec les directions perpendiculaires.



(a) Calcul des composantes lissées en fonction des filtres de Deriche utilisés (classiques ou décalés).



(b) Calcul de la dérivée partielle selon x à partir de I^{CFA} .

Figure 3.7 : Calcul des composantes lissées et dérivation selon x par la méthode SDS.

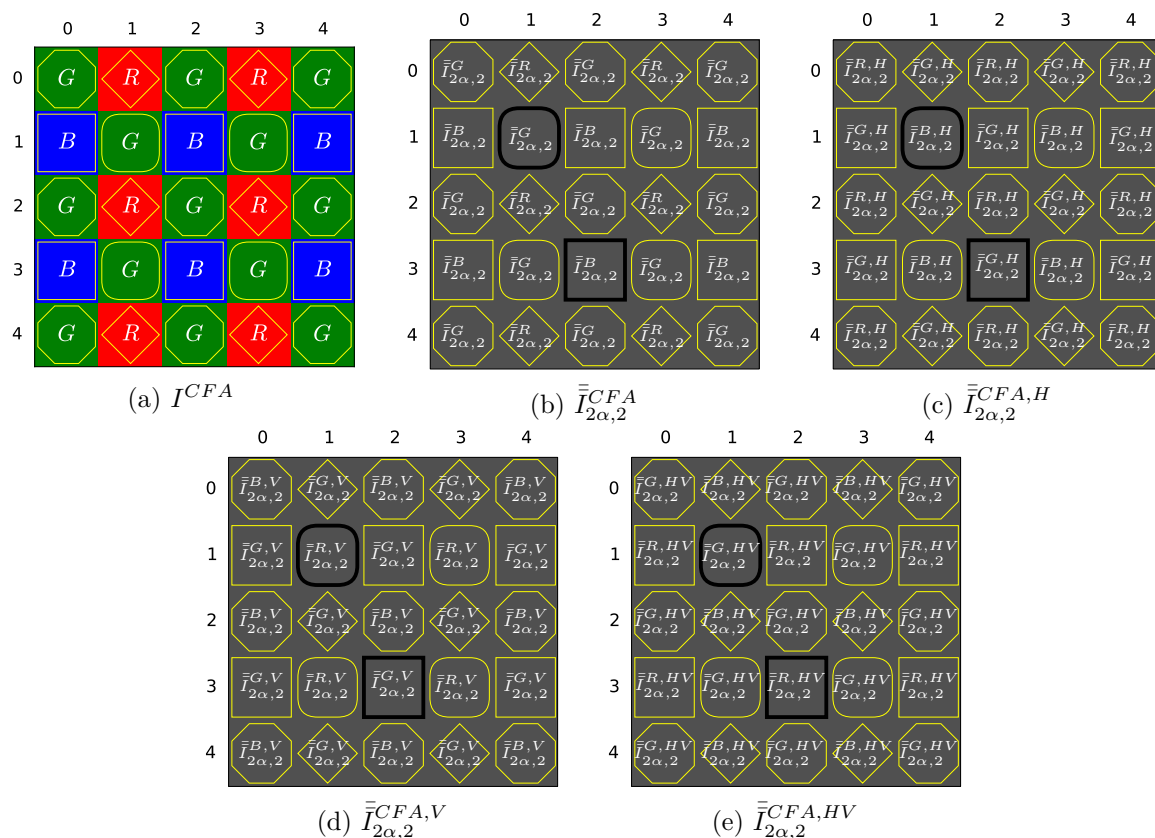


Figure 3.8 : : Calcul des composantes lissées de l'image I^{CFA} en utilisant les filtres de Deriche classique et les filtres de Deriche décalés, avec deux pixels comme exemple (lignes continues en gras).

4 Approches scalaires pour le calcul des dérivées partielles

Cette partie est consacrée aux approches dites scalaires pour calculer les dérivées partielles dans une image CFA. Notre idée s'inspire de la démarche de Herwig et Pauli [34] qui appliquent un filtre gaussien sur l'image CFA afin d'estimer la luminance (pour ensuite l'utiliser dans une procédure de dématricage). La luminance estimée peut ne pas être visuellement satisfaisante mais adaptée pour la détection des contours. La figure 3.9 montre la différence entre le schéma proposé afin de calculer les dérivées partielles à partir de I^{CFA} et le schéma classique où la luminance \hat{I}^L est estimée par dématricage puis dérivée. Cette approche scalaire fournit des dérivées partielles mono-canal \hat{I}_x^L et \hat{I}_y^L qui peuvent être utilisées pour la détection des contours par le détecteur des contours de Canny (ou, de manière équivalente, par l'approche multi-canaux de Di Zenzo utilisée avec trois canaux de luminance identiques).

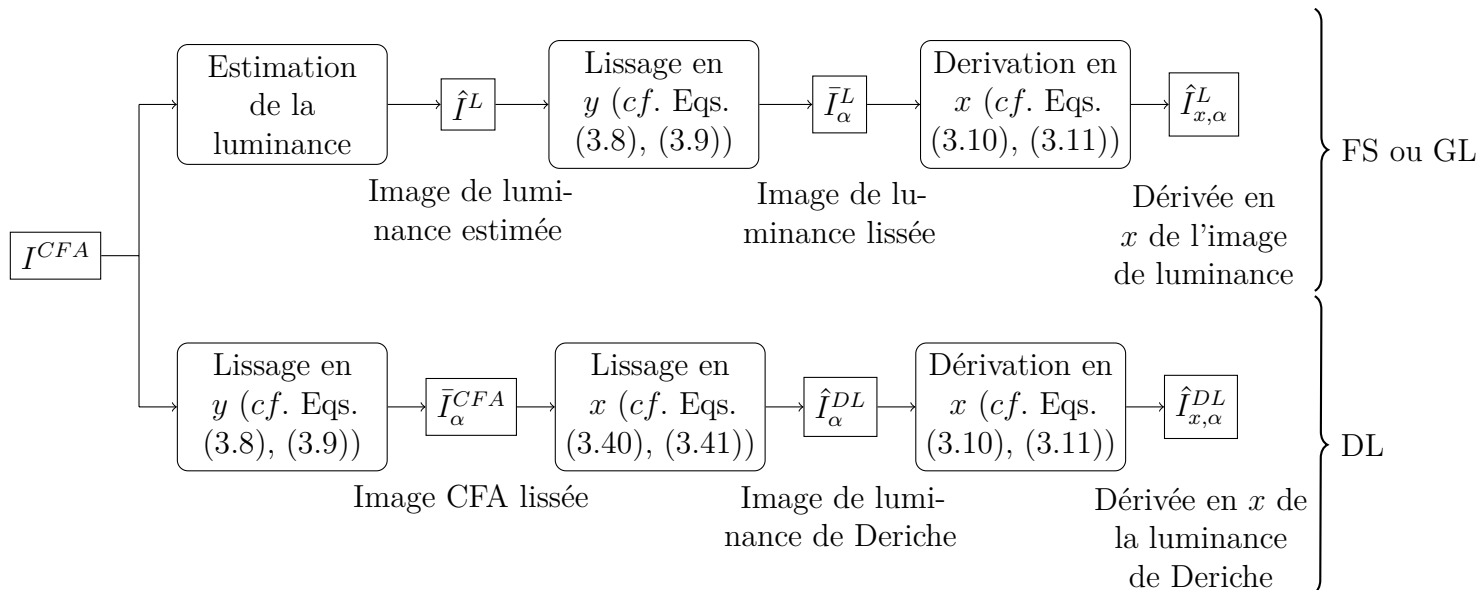


Figure 3.9 : : Calcul de la dérivée partielle selon x par les méthodes scalaires.

4.1 Estimation de la luminance par sélection de fréquences (FS)

Dubois [26] s'est basé sur les travaux d'Alleysson *et al.* [5] pour proposer une méthode de dématricage par sélection de fréquences (FS). Les auteurs représentent une image I^{CFA} comme la combinaison d'une composante de luminance I^L et deux composantes de chrominance I^{C_1} et I^{C_2} (cf. section 2.4 du chapitre 2). En supposant qu'il existe un signal f^{CFA} sous-jacent à l'image I^{CFA} avec laquelle il coïncide en chaque pixel $P \in S$, ce signal s'exprime par :

$$f^{CFA}(P) = \sum_{k \in \{R,G,B\}} f^k(P) m^k(P), \quad (3.35)$$

où m^k est la fonction d'échantillonnage : $m^k(P) = 1$ si $P(x,y) \in S^k$ et $m^k(P) = 0$ sinon. Pour le CFA de Bayer considéré, on a :

$$\begin{cases} m^R(x,y) = \frac{1}{4} (1 - (-1)^x)(1 + (-1)^y) \\ m^G(x,y) = \frac{1}{2} (1 + (-1)^{x+y}) \\ m^B(x,y) = \frac{1}{4} (1 + (-1)^x)(1 - (-1)^y) \end{cases} \quad (3.36)$$

En posant :

$$\begin{bmatrix} f^L \\ f^{C1} \\ f^{C2} \end{bmatrix} = \begin{bmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ -\frac{1}{4} & \frac{1}{2} & -\frac{1}{4} \\ -\frac{1}{4} & 0 & \frac{1}{4} \end{bmatrix} \begin{bmatrix} f^R \\ f^G \\ f^B \end{bmatrix}, \quad (3.37)$$

l'expression de f^{CFA} devient :

$$f^{CFA}(x,y) = f^L(x,y) + f^{C1}(x,y)(-1)^{x+y} + f^{C2}(x,y)((-1)^x - (-1)^y). \quad (3.38)$$

Sur l'équation (3.38), nous constatons que seuls les signaux de chrominance f^{C1} et f^{C2} sont sous-échantillonnés, alors que le signal de luminance f^L est présent en chaque pixel. L'image de luminance estimée \hat{I}^L coïncide donc en chaque pixel P avec le signal f^L , soit $\hat{I}^L(P) = f^L(P)$.

La dérivée partielle \hat{I}_x^L de \hat{I}^L selon x est obtenue directement en lissant \hat{I}^L verticalement grâce aux équations (3.8) et (3.9) avec $d = 1$, puis en dérivant le résultat à l'aide des équations (3.10) et (3.11) toujours avec $d = 1$. \hat{I}_y^L est obtenue de la même manière que \hat{I}_x^L mais avec des directions perpendiculaires. Cette procédure est décrite dans la première ligne de la figure 3.9.

4.2 Estimation de la luminance par filtre gaussien (GL)

Afin d'estimer une image de luminance \hat{I}^L directement à partir de I^{CFA} , Herwig et Pauli [35] convoluent l'image I^{CFA} avec un filtre G_σ de taille 3×3 approximant la fonction gaussienne bidimensionnelle d'écart-type $\sigma = 0.95$:

$$\hat{I}^L = I^{CFA} * G_\sigma, \text{ avec } G_\sigma = \begin{bmatrix} 0.0625 & 0.125 & 0.0625 \\ 0.125 & 0.25 & 0.125 \\ 0.0625 & 0.125 & 0.0625 \end{bmatrix}. \quad (3.39)$$

Cette approximation par filtre gaussien normalisé que l'on note GL (*Gaussian Luminance*) coïncide avec celle estimée par Alleysson *et al.* [5].

Comme pour FS, la dérivée partielle \hat{I}_x^L de \hat{I}^L selon x est obtenue en appliquant les équations (3.8) et (3.9) avec $d = 1$ à \hat{I}^L pour produire une image de luminance lissée verticalement qui est ensuite dérivée horizontalement à l'aide des équations (3.10)

et (3.11) avec $d = 1$. \hat{I}_y^L est obtenue de la même manière mais avec des directions perpendiculaires.

4.3 Estimation de la luminance par filtres de Deriche (DL)

Nous proposons enfin d'utiliser le filtre de lissage bidimensionnel de Deriche sur I^{CFA} afin d'estimer une image de luminance \hat{I}_α^{DL} . Cette méthode est nommée DL (*Deriche Luminance*).

Comme les filtres bidimensionnels de Deriche sont séparables, \hat{I}_α^{DL} peut être estimée par application séquentielle de deux filtres mono-dimensionnels. Premièrement, I^{CFA} est lissée selon y (cf. équations (3.8) et (3.9) avec $d = 1$). Le résultat $\bar{I}_{\alpha,1}^{CFA} = I_{\alpha,1}^{CFA-} + I_{\alpha,1}^{CFA+}$ est ensuite lissé selon x :

$$\bar{I}_{\alpha,1}^{DL-}(x,y) = a_1 \bar{I}_{\alpha,1}^{CFA}(x,y) + a_2 \bar{I}_{\alpha,1}^{CFA}(x-1,y) + b_1 \bar{I}_{\alpha,1}^{DL-}(x-1,y) + b_2 \bar{I}_{\alpha,1}^{DL-}(x-2,y), \quad (3.40)$$

$$\bar{I}_{\alpha,1}^{DL+}(x,y) = a_3 \bar{I}_{\alpha,1}^{CFA}(x+1,y) + a_4 \bar{I}_{\alpha,1}^{CFA}(x+2,y) + b_1 \bar{I}_{\alpha,1}^{DL+}(x+1,y) + b_2 \bar{I}_{\alpha,1}^{DL+}(x+2,y). \quad (3.41)$$

La luminance ainsi estimée $\hat{I}_\alpha^{DL} = \bar{I}_{\alpha,1}^{DL-} + \bar{I}_{\alpha,1}^{DL+}$ est finalement dérivée selon x ou y . Les équations (3.10) et (3.11) appliquées à \hat{I}_α^{DL} avec $d = 1$ fournissent la dérivée partielle horizontale $\hat{I}_{x,\alpha,1}^{DL}$, et $\hat{I}_{y,\alpha,1}^{DL}$ est obtenue avec les directions perpendiculaires.

Cette approche basée sur le filtrage de I^{CFA} fournit une image de luminance lissée (et non visuellement satisfaisante) qui est ensuite exploitée par un filtre de dérivation. Notre idée est d'utiliser un seul paramètre α pour conjointement estimer la luminance et dériver avec le filtre de Deriche. Cette méthode est décrite dans la seconde ligne de la figure 3.9.

5 Conclusion

Dans ce chapitre, nous avons présenté plusieurs méthodes originales afin de calculer les dérivées partielles directement à partir d'une image CFA. Deux approches majeures sont proposées à savoir les approches vectorielle et scalaire.

L'approche vectorielle consiste à calculer les trois composantes des dérivées partielles en chaque pixel. Cette approche est déclinée selon cinq méthodes originales. Les deux premières calculent les dérivées partielles disponibles en chaque pixel soit en utilisant une simple différenciation (SDE), soit en utilisant les filtres récursifs proposés par Deriche [23] avec la distance spatiale d fixée à 2 (DDE). Cela produit une image des dérivées

partielles avec une seule composante par pixel. Dans un second temps, nous estimons les deux composantes manquantes. La méthode SEDD estime les composantes lissées manquantes dans l'image issue de l'application des filtres de lissage de Deriche sur l'image CFA avec $d = 2$. L'image lissée pleinement définie résultante est alors dérivée avec les filtres récursifs de Deriche avec $d = 1$. Enfin, à l'aide des filtres décalés, la méthode SDD calcule les trois composantes des dérivées partielles en chaque pixel directement à partir de l'image CFA, et la méthode SDS calcule d'abord une image lissée pleinement définie à partir de l'image CFA puis les trois composantes des dérivées partielles en chaque pixel sans aucune estimation.

Nous avons également proposé une méthode relevant de l'approche scalaire. Elle consiste à estimer une image de luminance adaptée à la détection des contours en utilisant les filtres de lissage bidimensionnels. L'originalité de cette méthode DL réside dans la détermination d'un seul paramètre α pour les deux étapes d'estimation de la luminance et de dérivation afin de conserver l'optimalité du filtre de Deriche.

Les dérivées partielles calculées sont alors exploitées pour la détection des contours, soit par la méthode de Di Zenzo pour l'approche vectorielle, soit par la méthode de Canny pour l'approche scalaire. La qualité des contours détectés par chaque méthode est évaluée objectivement dans le chapitre suivant.

Chapitre 4

Évaluation objective de la qualité de détection des contours

1 Introduction

Ce chapitre est dédié à l'évaluation objective des différentes méthodes proposées pour la détection des contours dans une image CFA.

Comme la littérature ne présente pas de base d'images couleur disposant d'une vérité terrain satisfaisante pour la détection des contours, nous avons construit notre propre base d'images synthétiques. Nous commençons donc par décrire les différentes étapes nécessaires pour la construction de cette base, depuis la carte des contours constituant la vérité terrain jusqu'aux images couleur de référence en prenant en compte tous les paramètres qui influent sur la détection des contours dans des images couleur.

Ensuite, nous présentons la procédure expérimentale suivie et le critère utilisé afin d'évaluer les performances des différentes méthodes. Nous rappelons d'abord brièvement comment chacune procède selon qu'elle appartient à l'approche vectorielle ou à l'approche scalaire. Nous présentons ensuite le critère retenu pour mesurer la qualité des contours détectés.

La troisième partie est consacrée aux résultats de détection des contours obtenus par les différentes méthodes sur notre base d'images synthétiques. Afin de montrer l'importance des approches vectorielles, nous avons également évalué les différentes méthodes sur une base d'images dites iso-luminantes. Finalement, nous discutons et comparons les complexités algorithmiques des différentes méthodes. Chaque méthode est évaluée selon deux critères : performance de détection et complexité algorithmique.

2 Données et procédure expérimentales

2.1 Base de données expérimentales

Afin d'évaluer quantitativement les performances de détection des contours obtenues par les différentes méthodes, nous avons besoin d'une carte des contours vérité terrain notée I^T (pour *True edge*). Comme la littérature ne fournit pas une base standard d'images couleur disposant d'une carte des contours vérité terrain, les auteurs évaluent leurs détecteurs des contours couleur soit en fixant comme vérité terrain la réponse d'un détecteur particulier (Di Zenzo, Canny) [35], soit en utilisant la base de Berkeley qui fournit une vérité terrain obtenue par sélection des contours détourés manuellement [27]. De notre point de vue, une carte des contours vérité terrain ne peut pas être extraite à partir d'images naturelles. De ce fait, nous utilisons exclusivement notre propre base d'images synthétiques afin d'évaluer la qualité de détection des contours fournies par les différentes méthodes proposées.

2.1.1 Carte des contours vérité terrain

Pour évaluer la robustesse des différentes approches de détection par rapport à l'orientation des contours, nous construisons une carte des contours vérité terrain synthétique de 800×800 pixels qui contient des cercles concentriques (*cf.* figure 4.1a). Le diamètre du plus petit cercle est fixé à 139 pixels de sorte que le pas d'orientation de ses contours est beaucoup plus petit que celui ($\pi/16$) utilisé pour déterminer la direction θ^* du gradient couleur par l'approche du compas de Di Zenzo (*cf.* section 2.2 du chapitre 3). Afin d'évaluer la qualité de détection et de localisation des contours, nous utilisons 27 cercles concentriques dont les diamètres augmentent de sorte que la distance entre deux cercles successifs varie de 3 à 15 pixels par pas de 1. Les cercles les plus intérieurs avec de petites distances inter-cercles nous permettent d'évaluer la qualité de la détection des contours et les plus extérieurs avec de grandes distances inter-cercles sont dédiés à l'évaluation de la qualité de localisation des contours.

2.1.2 Images couleur de référence

À partir de notre carte des contours vérité terrain (*cf.* figure 4.1a), nous construisons une image synthétique couleur de base \mathbf{I}_0 en utilisant deux cercles consécutifs comme contours d'un anneau couleur. Le cercle le plus intérieur est caractérisé par la couleur \mathbf{c} , à partir de laquelle les couleurs des anneaux successifs alternent (*cf.* figure 4.1b). Les deux premiers anneaux (de même largeur 2 pixels) sont caractérisés par les couleurs $\mathbf{c} + \delta\mathbf{c}$ et

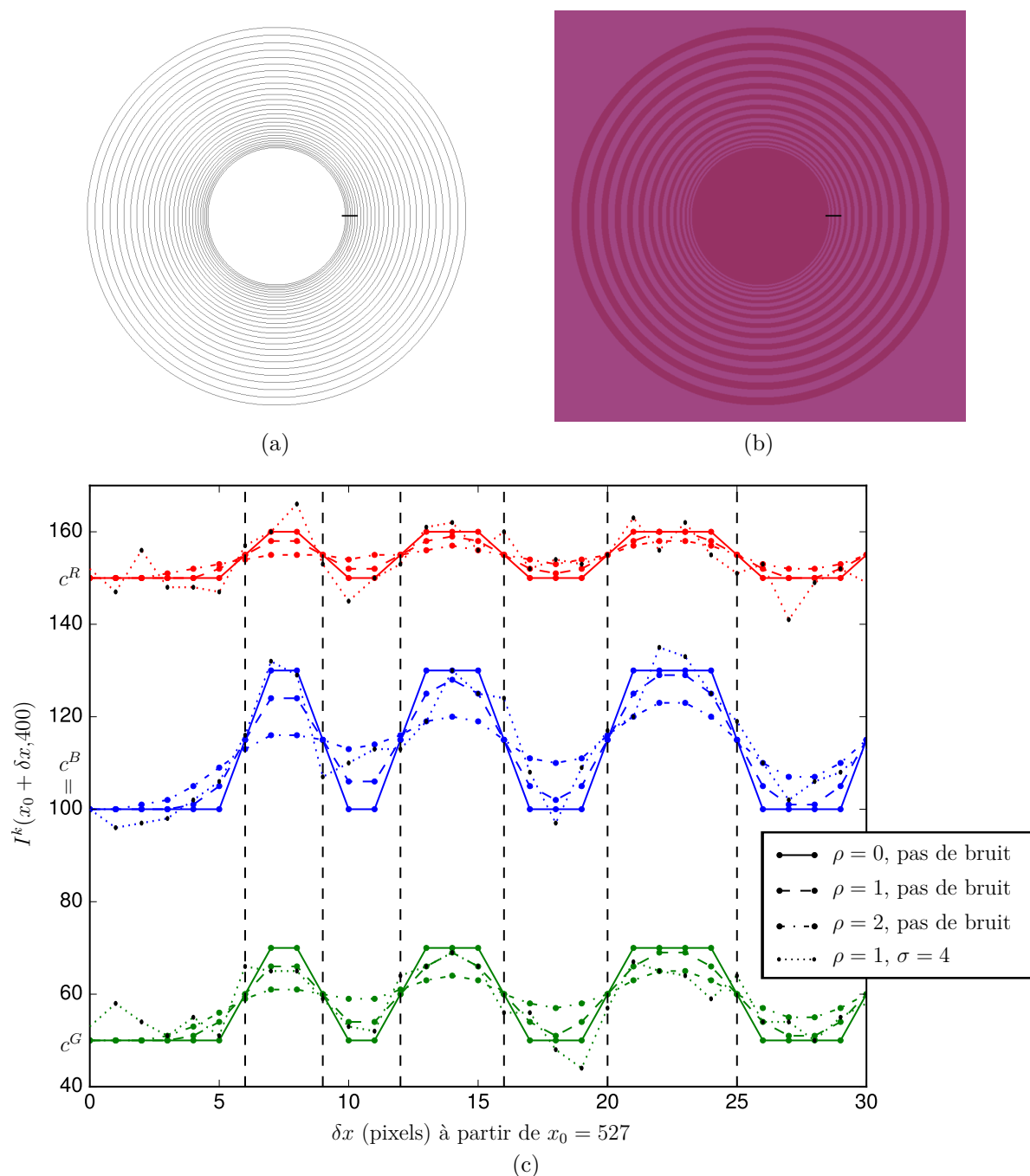


Figure 4.1 : Image de test synthétique. (a) Carte des contours vérité terrain. (b) Image couleur de référence \mathbf{I}_0 pour $\mathbf{c} = (150, 50, 100)$, $\delta\mathbf{c} = (10, 20, 30)$. (c) Profils des composantes couleur et localisation des contours (lignes verticales en pointillés) sur les 31 pixels horizontaux illustrés par des lignes noires sur (a) et (b) pour différents niveaux de flou et de bruit. Les profils en lignes continues ($\rho = 0$, pas de bruit) correspondent à l'image (b).

\mathbf{c} , les deux anneaux suivants (de même largeur 3 pixels) sont eux aussi caractérisés par les couleurs $\mathbf{c} + \delta\mathbf{c}$ et \mathbf{c} , ainsi de suite. La couleur des pixels contours qui séparent deux anneaux consécutifs est $\mathbf{c} + \frac{\delta\mathbf{c}}{2}$.

En plus de la hauteur $\delta\mathbf{c}$ de la transition couleur, la largeur de transition et le niveau de bruit peuvent aussi influencer les résultats de détection des contours. Nous considérons donc deux paramètres supplémentaires, à savoir le rayon ρ (en pixels) du flou gaussien appliqué à $\{I_0^k\}_{k=R,G,B}$ pour ajuster la largeur de transition, et le niveau σ du bruit appliqué aux trois canaux afin d'obtenir une image bruitée $\tilde{\mathbf{I}}$. Nous utilisons le modèle de bruit additif dépendant du canal [75, 3]. Pour chaque canal flouté, nous ajoutons un bruit gaussien de moyenne nulle et dont l'écart-type $\sigma^k = \sigma \frac{E^k}{\frac{1}{3} \sum_{k=R,G,B} E^k}$ est proportionnel au niveau global de bruit σ et à l'énergie (c'est-à-dire la valeur moyenne) E^k de chaque canal I_0^k . La figure 4.1c illustre les profils des composantes couleur des anneaux les plus intérieurs pour différentes valeurs des paramètres ρ et σ . Elle montre que la détection des contours est plus difficile lorsque la largeur de transition et le niveau de bruit augmentent.

Pour évaluer en profondeur la robustesse de la détection des contours, nous construisons un ensemble d'images synthétiques de test en considérant deux couleurs différentes $\mathbf{c} \in \{(100, 150, 50), (150, 50, 100)\}$ pour le disque intérieur et plusieurs valeurs pour chacun des autres paramètres. Ces deux couleurs \mathbf{c} sont motivées par le fait que le bruit est proportionnel à l'énergie de chaque canal. La hauteur de transition est donnée par $\delta c^k \in \{10, 20, 30\}$, $k \in \{R, G, B\}$. Nous fixons $\delta c^B = \delta c^R$ car les composantes R et B jouent un rôle similaire dans le CFA de Bayer. Le rayon du flou gaussien ρ varie de 0 (absence de flou) à 2 (transitions très progressives) et le niveau de bruit σ varie de 1 à 8. Notre base d'images synthétiques est alors composée de 432 images.

Gunturk *et al.* [29] ont démontré qu'il existe une corrélation $C^{k,l}$, $(k,l) \in \{R,G,B\}^2$, $k \neq l$, très élevée entre les niveaux des différentes composantes couleur dans une image naturelle (*cf.* section 2.2 du chapitre 2). Les auteurs calculent les coefficients de corrélation entre les composantes couleur R,G,B sur les images de la base Kodak [6]. Zhang *et al.* [74] montrent que cette corrélation dépend grandement de la base d'images considérée et donnent des résultats ($C^{R,G} = 0.675$, $C^{G,B} = 0.72$ et $C^{R,B} = 0.558$) qui sont des moyennes sur plusieurs bases d'images naturelles. Nous faisons la même étude sur notre base d'images synthétique. Le tableau 4.1 présente les coefficients de corrélation moyens entre composantes couleur en fonction du niveau de bruit dans les images. Nous constatons que la corrélation est très élevée quand les images ne sont pas très bruitées,

Tableau 4.1 : Corrélation spectrale inter-composantes dans nos images synthétiques en fonction de l'écart-type du bruit ajouté.

| Bruit | $C^{R,G}$ | $C^{G,B}$ | $C^{R,B}$ |
|--------------|-----------|-----------|-----------|
| $\sigma = 0$ | 0.9957 | 0.9975 | 0.9990 |
| $\sigma = 1$ | 0.9684 | 0.9780 | 0.9746 |
| $\sigma = 2$ | 0.8965 | 0.9279 | 0.9160 |
| $\sigma = 3$ | 0.8068 | 0.8615 | 0.8418 |
| $\sigma = 4$ | 0.7165 | 0.7903 | 0.7662 |
| $\sigma = 5$ | 0.6330 | 0.7200 | 0.6942 |
| $\sigma = 6$ | 0.5584 | 0.6538 | 0.6281 |
| $\sigma = 7$ | 0.4935 | 0.5927 | 0.5684 |
| $\sigma = 8$ | 0.4374 | 0.5375 | 0.5153 |
| Moyenne | 0.7229 | 0.7844 | 0.7671 |

puis diminue pour des niveaux de bruit plus élevés. Nos corrélations spectrales moyennes ($C^{R,G} = 0.7229$, $C^{G,B} = 0.7844$ et $C^{R,B} = 0.7671$) sont assez similaires à celles calculées par Zhang *et al.* [74] dans des images naturelles. Ceci justifie l'utilisation des méthodes de dématricage/estimation de luminance reposant sur des hypothèses propres aux images naturelles sur notre base.

2.2 Procédure expérimentale

La figure 4.2 montre la procédure utilisée afin d'évaluer les performances de détection des contours avec les différentes méthodes. À partir de la carte des contours vérité terrain I^T , les paramètres \mathbf{c} , $\delta\mathbf{c}$, ρ , et σ sont utilisés pour générer une image couleur synthétique bruitée $\tilde{\mathbf{I}}$ (*cf.* section 2.1.2). Cette image considérée comme l'image de référence ne peut pas être fournie par les caméras couleur mono-capteur. Ces caméras acquièrent une image CFA bruitée \tilde{I}^{CFA} que nous simulons à partir de $\tilde{\mathbf{I}}$ par échantillonnage selon

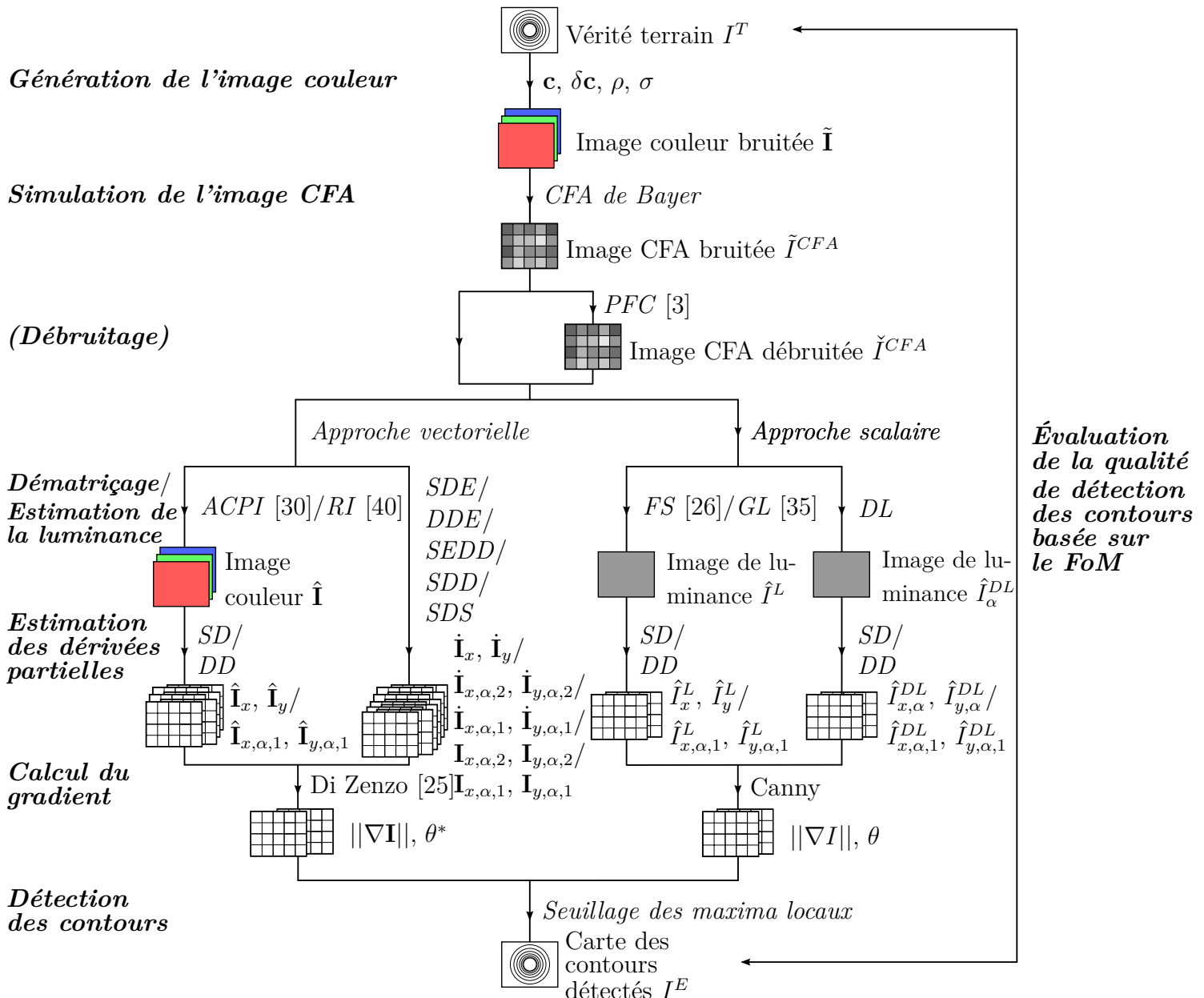


Figure 4.2 : Détection des contours et évaluation de leur qualité selon les différentes méthodes.

le CFA de Bayer.

La première étape est le débruitage de l'image CFA, ce qui fournit l'image CFA débruitée \check{I}^{CFA} . Nous utilisons pour cela l'une des méthodes de débruitage les plus performantes, celle d'Akiyama *et al.* [3] appelée PFC (*Pseudo Four-Channel image denoising*) (*cf.* section 3.4 du chapitre 1).

Notre objectif est de comparer les résultats obtenus par plusieurs méthodes de détection des contours appliquées à \check{I}^{CFA} ou \check{I}^{CFA} . Deux approches nous permettent d'obtenir les dérivées partielles requises pour calculer la norme $\|\nabla\mathbf{I}\|$ et la direction θ^* du gradient :

- L'approche vectorielle calcule les dérivées partielles pour chaque composante couleur.
 - La démarche classique consiste à estimer d'abord une image couleur $\hat{\mathbf{I}}$ en utilisant un algorithme de dématricage. Puis il s'agit de calculer soit ses dérivées simples (*Simple Derivatives*, SD), soit les dérivées de Deriche (*Deriche Derivatives*, DD) en appliquant les équations (3.2) ou (3.8) à (3.11) avec $d = 1$ sur $\{\hat{I}^k\}_{k=R,G,B}$. Nous considérons ici l'algorithme de dématricage simple mais reconnu comme efficace proposé par Hamilton et Adams [30] (*Adaptive Color Plan Interpolation*, ACPI) et l'un des meilleurs algorithmes de dématricage dans la littérature basé sur l'interpolation résiduelle (RI) [40].
 - Nous proposons plutôt de calculer les composantes des dérivées partielles directement à partir de l'image CFA par :
 - l'estimation des SD qui délivre $\dot{\mathbf{I}}_x, \dot{\mathbf{I}}_y$ (SDE) (*cf.* section 3.1 du chapitre 3).
 - l'estimation des DD qui délivre $\dot{\mathbf{I}}_{x,\alpha,2}, \dot{\mathbf{I}}_{y,\alpha,2}$ (DDE) (*cf.* section 3.2 du chapitre 3).
 - le calcul des DD à partir des composantes lissées préalablement estimées (SEDD) (*cf.* section 3.3 du chapitre 3) qui délivre $\dot{\mathbf{I}}_{x,\alpha,1}, \dot{\mathbf{I}}_{y,\alpha,1}$.
 - le calcul des DD décalées (SDD) (*cf.* section 3.5 du chapitre 3) qui délivre $\mathbf{I}_{x,\alpha,2}, \mathbf{I}_{y,\alpha,2}$.
 - le calcul des DD à partir des composantes lissées par filtres de Deriche décalés (SDS) (*cf.* section 3.6 du chapitre 3) qui délivre $\mathbf{I}_{x,\alpha,1}, \mathbf{I}_{y,\alpha,1}$.
- L'approche scalaire estime d'abord une image de luminance, puis calcule soit les SD soit les DD en appliquant les équations (3.2) ou (3.8) à (3.11) avec $d = 1$.
 - L'image de luminance \hat{I}^L peut être obtenue par sélection de fréquences (FS) [26] (*cf.* section 4.1 du chapitre 3) ou filtrage gaussien de l'image CFA (*Gaussian-based Luminance*, GL) [35] (*cf.* section 4.2 du chapitre 3).

- Nous proposons également (*cf.* section 4.3 du chapitre 3) d'appliquer les filtres de lissage de Deriche directement sur l'image CFA, qui fournit une image de luminance \hat{I}_α^{DL} (*Deriche-based Luminance*, DL).

Toutes ces méthodes permettent de calculer la norme $\|\nabla\mathbf{I}\|$ et la direction θ^* du gradient couleur de Di Zenzo (*cf.* section 2.2 du chapitre 3), à partir desquels la carte des contours est déterminée par un seuillage des maxima locaux de $\|\nabla\mathbf{I}\|$ dans la direction θ^* en appliquant un seuil simple T_s .

2.3 Critère d'évaluation de la qualité de détection des contours

Afin d'évaluer la qualité de détection des contours issue des différentes méthodes, nous comparons la carte des contours détectés notée I^E avec la vérité terrain I^T grâce au critère du facteur de mérite (*Figure of Merit*, FoM) proposé par Abdou et Pratt [1]. Ce critère d'évaluation de la qualité de détection des contours est largement utilisé dans la littérature et s'exprime comme suit :

$$FoM(I^T, I^E) = \frac{1}{\max(\text{Card}(I^T), \text{Card}(I^E))} \sum_{P \in I^E} \frac{1}{1 + \frac{1}{9} \text{dis}^2(P)}, \quad (4.1)$$

où $\text{dis}(P)$ est la distance spatiale euclidienne entre un pixel de contour détecté P et le plus proche pixel contour vérité terrain dans I^T , et $\text{Card}(\bullet)$ est le nombre de pixels contours dans une carte des contours. La carte des contours détectés I^E est considérée de bonne qualité si son FoM est proche de 1 et de mauvaise qualité quand il tend vers 0. Comme ce critère est plus sensible à la localisation qu'à la détection, nous calculons sa valeur moyenne lorsque I^T et I^E sont permutés, de telle sorte que $FoM = \frac{1}{2} (FoM(I^T, I^E) + FoM(I^E, I^T))$ est sensible à la sous-détection et à la sur-détection des contours.

Afin de choisir le meilleur seuil T_s pour chaque méthode de détection des contours, nous seuillons l'image des maxima locaux avec toutes les valeurs possibles de T_s . Nous calculons ensuite le FoM entre la carte des contours vérité terrain et celle des contours détectés obtenue avec chaque seuil T_s . Le seuil sélectionné correspond au meilleur FoM. Nous sélectionnons ainsi le seuil qui correspond au meilleur FoM pour chaque méthode. À partir du critère FoM, nous pouvons évaluer la qualité de détection des contours par les différentes méthodes. Nous nous appuyons sur le FoM à la fois pour ajuster le paramètre α du filtre de Deriche et pour comparer les résultats.

3 Résultats expérimentaux

3.1 Détermination du paramètre α du filtre de Deriche

Comme mentionné dans la section 2.3 du chapitre 3, les coefficients de pondération utilisés dans les filtres de Deriche dépendent du seul paramètre α . Les petites valeurs de α (généralement, $\alpha < 1$) donnent la priorité à la détection. Par contre, les valeurs élevées de α ($\alpha > 1$) fournissent une bonne localisation. Pour faire face à un tel compromis, nous choisissons la meilleure valeur de α (en terme de FoM) pour chaque méthode de détection des contours. Nous faisons varier ce paramètre α de 0 à 3, calculons le FoM moyen pour chaque valeur de α , et retenons finalement celle qui donne le meilleur FoM.

La figure 4.3 montre l'évolution du FoM moyen (sur les 432 images) en fonction du paramètre α de Deriche pour chaque méthode de détection des contours. Par exemple, d'après la figure 4.3c consacrée à la méthode DDE, nous pouvons constater que le meilleur FoM moyen est obtenu avec $\alpha = 1.2$ pour les images bruitées ($FoM_{max} = 0.9274$, *cf.* courbe rouge discontinue) et avec $\alpha = 2.0$ pour les images débruitées ($FoM_{max} = 0.9705$, *cf.* courbe verte continue). La valeur de α retenue pour chaque méthode est donnée dans la première ligne des tableaux 4.2a et 4.2b.

Comme attendu, les images bruitées nécessitent plus de lissage, donc des valeurs de α plus petites que les images débruitées. Notons également que les valeurs de α retenues pour les méthodes utilisant des plans de composantes pleinement définis (ACPI, RI, SEDD, FS) sont similaires et représentent environ la moitié de celles retenues pour les méthodes basées sur le treillis CFA (DDE et SDD). En effet, ces méthodes utilisent une distance horizontale et verticale double entre les pixels voisins ($d = 2$ vs. $d = 1$) dans l'implémentation récursive des filtres de Deriche.

3.2 Résultats sur des images synthétiques

Les tableaux 4.2a et 4.2b montrent les résultats de détection des contours obtenue par les différentes méthodes testées à partir des images CFA bruitées et débruitées. Chaque cellule du tableau correspond au FoM moyen calculé sur les 144 images disponibles pour chaque valeur du rayon ρ du flou gaussien. Ces images sont obtenues en considérant 2 valeurs de \mathbf{c} , 9 valeurs de $\delta\mathbf{c}$ et 8 valeurs de σ (*cf.* section 2.2). Pour chaque méthode de dérivation (SD ou DD) et chaque valeur de ρ , le meilleur résultat est inscrit en gras.

L'augmentation du paramètre ρ rend les transitions moins nettes, donc la détection des contours plus difficile et le critère de qualité FoM plus faible. Une comparaison cellule

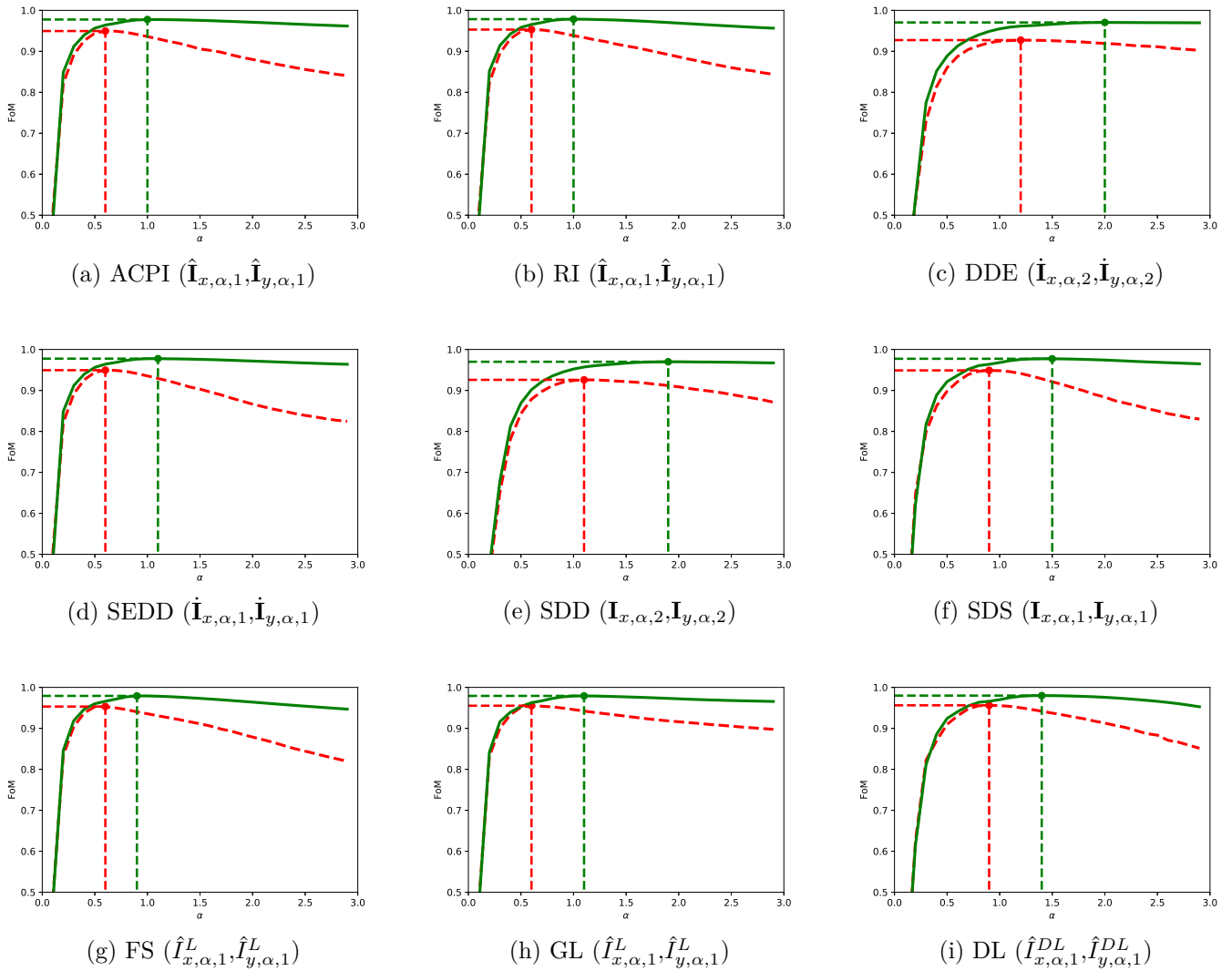


Figure 4.3 : Évolution du FoM en fonction du paramètre α de Deriche pour chaque méthode de détection des contours à partir des images CFA bruitées \tilde{I}^{CFA} (ligne discontinue) et débruitées \check{I}^{CFA} (ligne continue). Les points sur les courbes montrent le FoM maximum et le paramètre α retenu dans chaque cas.

Tableau 4.2 : Performance des différentes méthodes de détection des contours sur notre base d'images synthétiques.(a) Image CFA bruitée \tilde{I}^{CFA}

| | Dérivation simple (SD) | | | | | | Dérivation de Deriche (DD) | | | | | | | | |
|------------|------------------------|---------|--------|---------------------|---------|---------------|----------------------------|---------|--------|--------|--------|--------|---------------------|---------|---------------|
| | Approches vectorielles | | | Approches scalaires | | | Approches vectorielles | | | | | | Approches scalaires | | |
| | ACPI [30] | RI [40] | SDE | FS [26] | GL [35] | DL | ACPI [30] | RI [40] | DDE | SEDD | SDD | SDS | FS [26] | GL [35] | DL |
| α | — | — | — | — | — | 0.9 | 0.6 | 0.6 | 1.2 | 0.6 | 1.1 | 0.9 | 0.6 | 0.7 | 0.9 |
| $\rho = 0$ | 0.8800 | 0.8829 | 0.8758 | 0.8336 | 0.9554 | 0.9718 | 0.9680 | 0.9710 | 0.9456 | 0.9681 | 0.9500 | 0.9700 | 0.9707 | 0.9721 | 0.9736 |
| $\rho = 1$ | 0.7815 | 0.7766 | 0.7802 | 0.7308 | 0.9041 | 0.9465 | 0.9581 | 0.9606 | 0.9365 | 0.9578 | 0.9333 | 0.9572 | 0.9599 | 0.9461 | 0.9636 |
| $\rho = 2$ | 0.6601 | 0.6610 | 0.6639 | 0.6326 | 0.7872 | 0.8770 | 0.9226 | 0.9275 | 0.9002 | 0.9224 | 0.8944 | 0.9202 | 0.9291 | 0.8886 | 0.9314 |
| Moyenne | 0.7739 | 0.7735 | 0.7733 | 0.7323 | 0.8822 | 0.9318 | 0.9495 | 0.9531 | 0.9274 | 0.9494 | 0.9249 | 0.9489 | 0.9532 | 0.9356 | 0.9562 |
| Écart-type | 0.1582 | 0.1499 | 0.1619 | 0.1480 | 0.1181 | 0.0793 | 0.0369 | 0.0340 | 0.0511 | 0.0370 | 0.023 | 0.0214 | 0.0332 | 0.0540 | 0.0313 |

(b) Image CFA débruitée \check{I}^{CFA}

| | Dérivation simple (SD) | | | | | | Dérivation de Deriche (DD) | | | | | | | | |
|------------|------------------------|---------|--------|---------------------|---------|---------------|----------------------------|---------|--------|--------|--------|--------|---------------------|---------|---------------|
| | Approches vectorielles | | | Approches scalaires | | | Approches vectorielles | | | | | | Approches scalaires | | |
| | ACPI [30] | RI [40] | SDE | FS [26] | GL [35] | DL | ACPI [30] | RI [40] | DDE | SEDD | SDD | SDS | FS [26] | GL [35] | DL |
| α | — | — | — | — | — | 1.1 | 1.0 | 1.0 | 2.0 | 1.1 | 1.9 | 1.5 | 0.9 | 0.8 | 1.4 |
| $\rho = 0$ | 0.9837 | 0.9857 | 0.9830 | 0.9702 | 0.9875 | 0.9900 | 0.9897 | 0.9906 | 0.9794 | 0.9902 | 0.9849 | 0.9909 | 0.9900 | 0.9895 | 0.9910 |
| $\rho = 1$ | 0.9606 | 0.9569 | 0.9641 | 0.9383 | 0.9767 | 0.9822 | 0.9828 | 0.9835 | 0.9751 | 0.9829 | 0.9751 | 0.9831 | 0.9836 | 0.9808 | 0.9848 |
| $\rho = 2$ | 0.8876 | 0.8636 | 0.9045 | 0.8464 | 0.9248 | 0.9590 | 0.9653 | 0.9614 | 0.9569 | 0.9596 | 0.9496 | 0.9579 | 0.9640 | 0.9565 | 0.9640 |
| Moyenne | 0.9440 | 0.9354 | 0.9505 | 0.9183 | 0.9630 | 0.9771 | 0.9776 | 0.9785 | 0.9705 | 0.9776 | 0.9700 | 0.9773 | 0.9792 | 0.9756 | 0.9799 |
| Écart-type | 0.0737 | 0.0742 | 0.0680 | 0.0836 | 0.0492 | 0.0238 | 0.0211 | 0.0205 | 0.0202 | 0.0225 | 0.015 | 0.0141 | 0.0172 | 0.0215 | 0.0180 |

à cellule des tableaux 4.2a et 4.2b confirme par ailleurs que le débruitage de l'image CFA par la méthode proposée par Akiyama *et al.* [3] est une étape pertinente car elle permet d'améliorer les résultats de détection des contours et réduit leur dispersion quelle que soit la méthode utilisée. Le gain de performances par rapport aux méthodes sans débruitage est d'environ 25% avec SD et 5% avec DD car cette dernière méthode incorpore un lissage de l'image.

Bien que les résultats obtenus avec SD soient nettement inférieurs à ceux obtenus avec DD pour toutes les méthodes testées, la méthode utilisant une dérivation simple reste intéressante car elle est très peu gourmande en temps de calcul (*cf.* section 4). La méthode DL fournit les meilleurs résultats dans tous les cas testés. Elle surpasse les autres méthodes de jusqu'à 20% lorsqu'elle est couplée à SD sur les images bruitées, grâce à l'estimation de la luminance par lissage bidimensionnel de l'image CFA (comme GL), ce qui réduit énormément son niveau de bruit. Cette conclusion est moins marquée avec DD car toutes les autres méthodes (en particulier celles basées sur le dématricage) profitent des filtres optimaux de Deriche classiques. Les bonnes performances de DL reposent principalement sur la valeur du paramètre α retenue qui est utilisée à la fois pour l'estimation de la luminance et le calcul des dérivées.

Dans le tableau 4.2b, nous constatons que toutes les méthodes donnent des performances assez similaires dans le cas des images débruitées. En effet, dans l'avant-dernière ligne du tableau 4.2b, les FoMs moyens des différentes méthodes sont tous compris entre 0.97 et 0.98 lorsque DD est utilisée. Par conséquent, les résultats montrés dans la suite de ce chapitre sont basés sur des images bruitées car c'est dans ce cas qu'il existe des différences de performances assez marquées entre les méthodes de détection des contours.

La figure 4.4 montre des extraits de résultats de détection des contours obtenus par les différentes méthodes dans le cas le plus compliqué, c'est-à-dire pour une largeur de transition élevée ($\rho = 2$) et une hauteur de transition faible ($\delta\mathbf{c} = \{10,20,10\}$). Comme les performances des différentes méthodes sont très sensibles au paramètre α , toutes les méthodes ont été testées avec la même valeur $\alpha = 0.6$ sauf pour SDD et DDE dans lesquelles nous fixons $\alpha = 1.2$ pour une comparaison équitable. Les résultats visuels confirment les résultats de notre évaluation objective donnée par la moyenne des FoMs. Nous constatons que la méthode DL proposée est la plus efficace ($FoM = 0.8909$) sur la détection des contours intérieurs qui correspondent à des transitions très proches. Notons aussi que la méthode proposée SDS est moins sujette à la sur-détection et présente le meilleur résultat de détection des contours parmi les approches vectorielles.

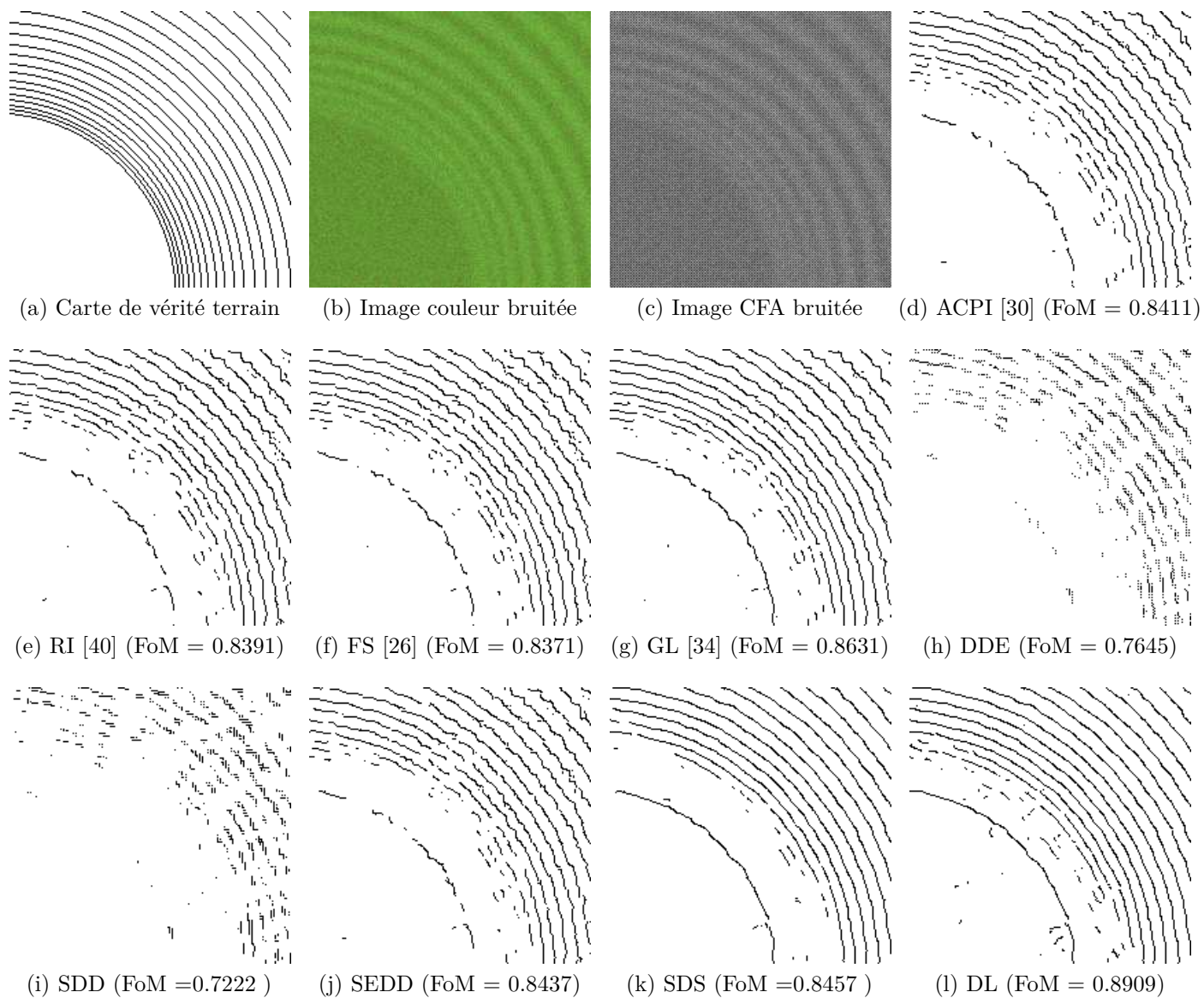


Figure 4.4 : Résultats de détection des contours à partir de l'image CFA bruitée \tilde{I}^{CFA} (c) échantillonnée à partir de l'image couleur bruitée $\tilde{\mathbf{I}}$ (b) générée avec $\mathbf{c} = (100, 150, 50)$, $\delta\mathbf{c} = (10, 20, 10)$, $\rho = 2$, et $\sigma = 8$ (PSNR=22.72 dB).

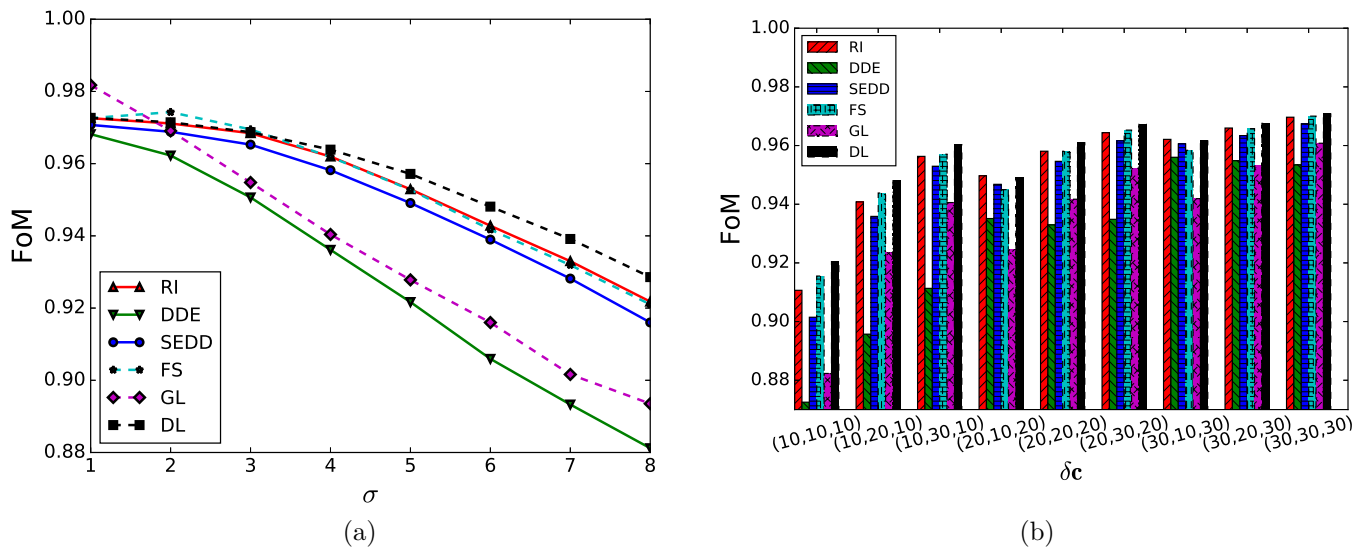


Figure 4.5 : Évolution des FoM en fonction de (a) niveau du bruit gaussien(σ) et (b) hauteur de transition (δc) pour les différentes méthodes utilisant la dérivation de Deriche (DD) à partir de l'image CFA bruitée.

Afin d'étudier l'influence du bruit sur la détection des contours, nous utilisons notre base d'images synthétiques et nous calculons le FoM moyen pour chaque valeur de l'écart-type σ du bruit gaussien ajouté. La figure 4.5a affiche l'évolution du FoM en fonction de σ en moyennant les résultats sur toutes les valeurs des autres paramètres¹. Ces résultats sont obtenus en combinant DD avec les différentes méthodes appliquées aux images CFA bruitées. Soulignons que DDE et SDD ne donnent jamais de résultats satisfaisants quel que soit le niveau de bruit, la qualité des contours détectés avec ces deux méthodes étant toujours inférieure à celle des autres méthodes (*cf.* tableau 4.2a). Cela est dû à la distance spatiale d fixée à 2 fixée pour deux méthodes. Signalons aussi que GL n'est efficace qu'à de faibles niveaux de bruit, et que d'autres méthodes - et principalement DL - sont assez robustes au bruit, ce qui explique les performances présentées dans le tableau 4.2. Une autre étude sur la sensibilité du FoM par rapport à la hauteur de transition δc (*cf.* figure 4.5b) confirme les conclusions précédentes.

3.3 Résultats sur des images synthétiques iso-luminantes

Nous souhaitons maintenant évaluer les performances des différentes méthodes dans le cas particulier d'images présentant des contours iso-luminants. Trois de nos méthodes

1. ACPI et SDS se comportent comme SEDD et SDD comme DDE donc leurs résultats ne sont pas affichés pour préserver la clarté de la figure.

de détection des contours sont en effet basées sur l'estimation de l'image de luminance. Comme les contours iso-luminants ne sont pas rares dans des images de scènes naturelles [31], cette étape est nécessaire en plus de l'étude de corrélation (*cf.* section 2.1.2) pour rendre nos résultats extensibles aux images de scènes naturelles.

En supposant que l'image de luminance est donnée par $I^L = 0.2126I^R + 0.7152I^G + 0.0722I^B$, nous construisons des images synthétiques avec des contours iso-luminants en fixant $\delta c^R = \delta c^B$ et $\delta c^G = -0.3982\delta c^R$, de telle sorte $\delta c^L = 0$. Puis nous répétons l'étude sur le paramètre α (*cf.* section 3.1) en conservant pour chaque méthode la valeur de α qui fournit le FoM moyen le plus élevé.

La figure 4.6 montre l'évolution de la moyenne du FoM sur les 144 images, obtenues en considérant 2 valeurs de \mathbf{c} , 9 valeurs de $\delta\mathbf{c}$ et 8 valeurs de σ , en fonction du paramètre α de Deriche pour chaque méthode de détection des contours.

Le tableau 4.3 montre les performances des différentes méthodes sur les images iso-luminantes. Par comparaison avec le tableau 4.2, nous constatons que les performances des trois méthodes basées sur l'estimation de la luminance (approches scalaires) sont significativement diminuées. En effet, les méthodes scalaires s'appuyant sur l'estimation de la luminance sont pénalisées dans ce cas particulier des contours iso-luminants.

La figure 4.7 montre un exemple de résultat de détection des contours iso-luminants obtenu par les différentes méthodes dans le cas le plus compliqué : largeur de transition élevée ($\rho = 2$) et hauteur de transition faible ($\delta\mathbf{c} = \{15, -6, 15\}$). Toutes les méthodes sont testées avec $\alpha = 0.6$ sauf DDE et SDD qui utilisent une valeur double de α . Les résultats visuels confirment encore une fois les résultats de notre évaluation objective donnés par la moyenne des FoM (*cf.* tableau 4.3). La méthode DL qui était jusqu'à maintenant la plus performante ne fonctionne pas pour la détection des contours iso-luminants ($FoM = 0.599$) car elle estime une luminance à partir de l'image CFA. Le constat est le même pour les méthodes FS ($FoM = 0.559$) et GL ($FoM = 0.634$). En revanche nos deux méthodes vectorielles SEDD et SDS donnent de très bons résultats avec un avantage pour SDS qui fournit le même FoM que RI ($FoM = 0.838$).

4 Complexité algorithmique

Dans cette section, nous évaluons la complexité algorithmique de chaque méthode utilisée pour la détection des contours. Pour cela, nous évaluons le nombre d'opérations élémentaires utilisées dans l'algorithme de chaque méthode. Signalons que seules les opérations arithmétiques sont prises en compte dans le calcul. Hormis les étapes de

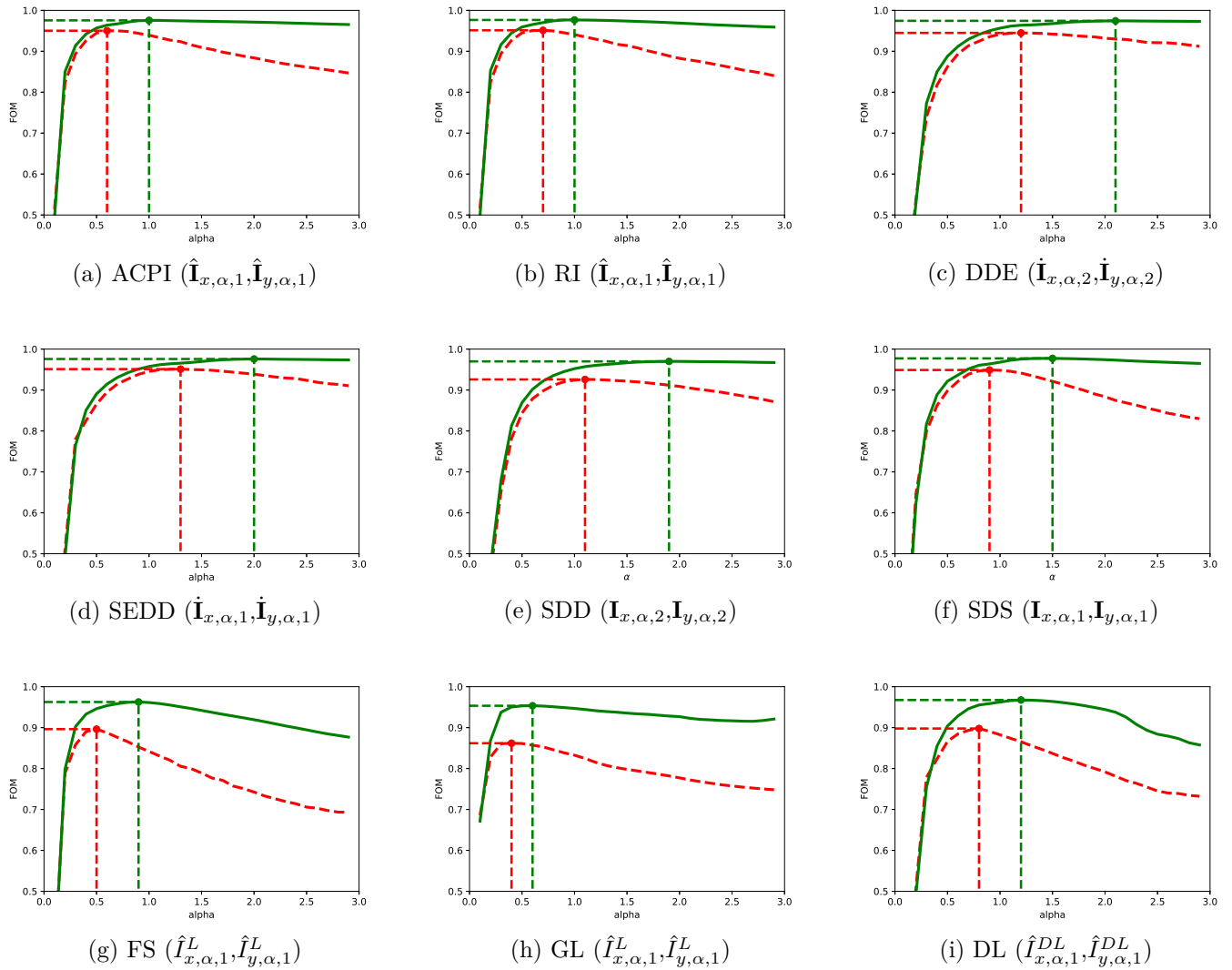


Figure 4.6 : Évolution du FoM en fonction du paramètre α de Deriche pour chaque méthode de détection des contours à partir des images CFA bruitées \tilde{I}^{CFA} (ligne discontinue) et pour les images CFA débruitées \hat{I}^{CFA} (ligne continue). Les points sur les courbes montrent le FoM maximum et le paramètre α retenu dans chaque cas.

Tableau 4.3 : Performance des différentes méthodes de détection des contours sur des images synthétiques iso-luminantes.(a) Image CFA bruitée \check{I}^{CFA}

| | Dérivation simple (SD) | | | | | | Dérivation de Deriche (DD) | | | | | | | | |
|-------------|------------------------|---------|---------------|---------------------|---------|---------------|----------------------------|---------------|--------|---------------|--------|--------------|---------------------|---------|--------|
| | Approches vectorielles | | | Approches scalaires | | | Approches vectorielles | | | | | | Approches scalaires | | |
| | ACPI [30] | RI [40] | SDE | FS [26] | GL [35] | DL | ACPI [30] | RI [40] | DDE | SEDD | SDD | SDS | FS [26] | GL [35] | DL |
| α | — | — | — | — | — | 0.6 | 1.0 | 1.0 | 2.1 | 2.0 | 1.9 | 0.9 | 0.5 | 0.4 | 0.8 |
| $\rho = 0$ | 0.8729 | 0.8431 | 0.8865 | 0.7340 | 0.8035 | 0.8839 | 0.9676 | 0.9706 | 0.9607 | 0.9697 | 0.959 | 0.973 | 0.9218 | 0.8982 | 0.9260 |
| $\rho = 1$ | 0.8061 | 0.7733 | 0.8047 | 0.6439 | 0.7256 | 0.8574 | 0.9599 | 0.9581 | 0.9536 | 0.9584 | 0.952 | 0.9527 | 0.9051 | 0.8719 | 0.9081 |
| $\rho = 2$ | 0.6679 | 0.6652 | 0.6961 | 0.5554 | 0.6112 | 0.7720 | 0.9232 | 0.9249 | 0.9184 | 0.9253 | 0.9177 | 0.9234 | 0.8619 | 0.8155 | 0.8592 |
| Moyenne | 0.7823 | 0.7606 | 0.7957 | 0.6445 | 0.7134 | 0.8378 | 0.9502 | 0.9512 | 0.9442 | 0.9511 | 0.9429 | 0.9497 | 0.8963 | 0.8618 | 0.8978 |
| Écart -type | 0.1740 | 0.1774 | 0.1682 | 0.1312 | 0.1814 | 0.1148 | 0.0383 | 0.0444 | 0.0422 | 0.0402 | 0.0412 | 0.0422 | 0.0776 | 0.0912 | 0.0853 |

(b) Image CFA débruitée \check{I}^{CFA}

| | Dérivation simple (SD) | | | | | | Dérivation de Deriche (DD) | | | | | | | | |
|-------------|------------------------|---------|---------------|---------------------|---------|---------------|----------------------------|---------------|--------|--------|--------|--------------|---------------------|---------|--------|
| | Approches vectorielles | | | Approches scalaires | | | Approches vectorielles | | | | | | Approches scalaires | | |
| | ACPI [30] | RI [40] | SDE | FS [26] | GL [35] | DL | ACPI [30] | RI [40] | DDE | SEDD | SDD | SDS | FS [26] | GL [35] | DL |
| α | — | — | — | — | — | 1.1 | 1.0 | 1.0 | 2.0 | 1.1 | 2.1 | 1.0 | 0.9 | 0.6 | 1.2 |
| $\rho = 0$ | 0.9797 | 0.9753 | 0.9801 | 0.8713 | 0.9104 | 0.9788 | 0.9871 | 0.9881 | 0.9843 | 0.9877 | 0.981 | 0.989 | 0.9794 | 0.9732 | 0.9819 |
| $\rho = 1$ | 0.9696 | 0.9555 | 0.9720 | 0.8742 | 0.8988 | 0.9696 | 0.9815 | 0.9818 | 0.9803 | 0.9817 | 0.979 | 0.982 | 0.9711 | 0.9609 | 0.9732 |
| $\rho = 2$ | 0.8876 | 0.8636 | 0.9045 | 0.8464 | 0.9248 | 0.9590 | 0.9579 | 0.9597 | 0.9583 | 0.9582 | 0.952 | 0.955 | 0.9372 | 0.9260 | 0.9471 |
| Moyenne | 0.9440 | 0.9354 | 0.9505 | 0.9183 | 0.9630 | 0.9771 | 0.9755 | 0.9765 | 0.9743 | 0.9758 | 0.97 | 0.975 | 0.9626 | 0.9534 | 0.9674 |
| Écart -type | 0.0737 | 0.0742 | 0.0680 | 0.0836 | 0.0492 | 0.0238 | 0.0245 | 0.0234 | 0.0245 | 0.0250 | 0.0251 | 0.0213 | 0.0381 | 0.0363 | 0.0318 |

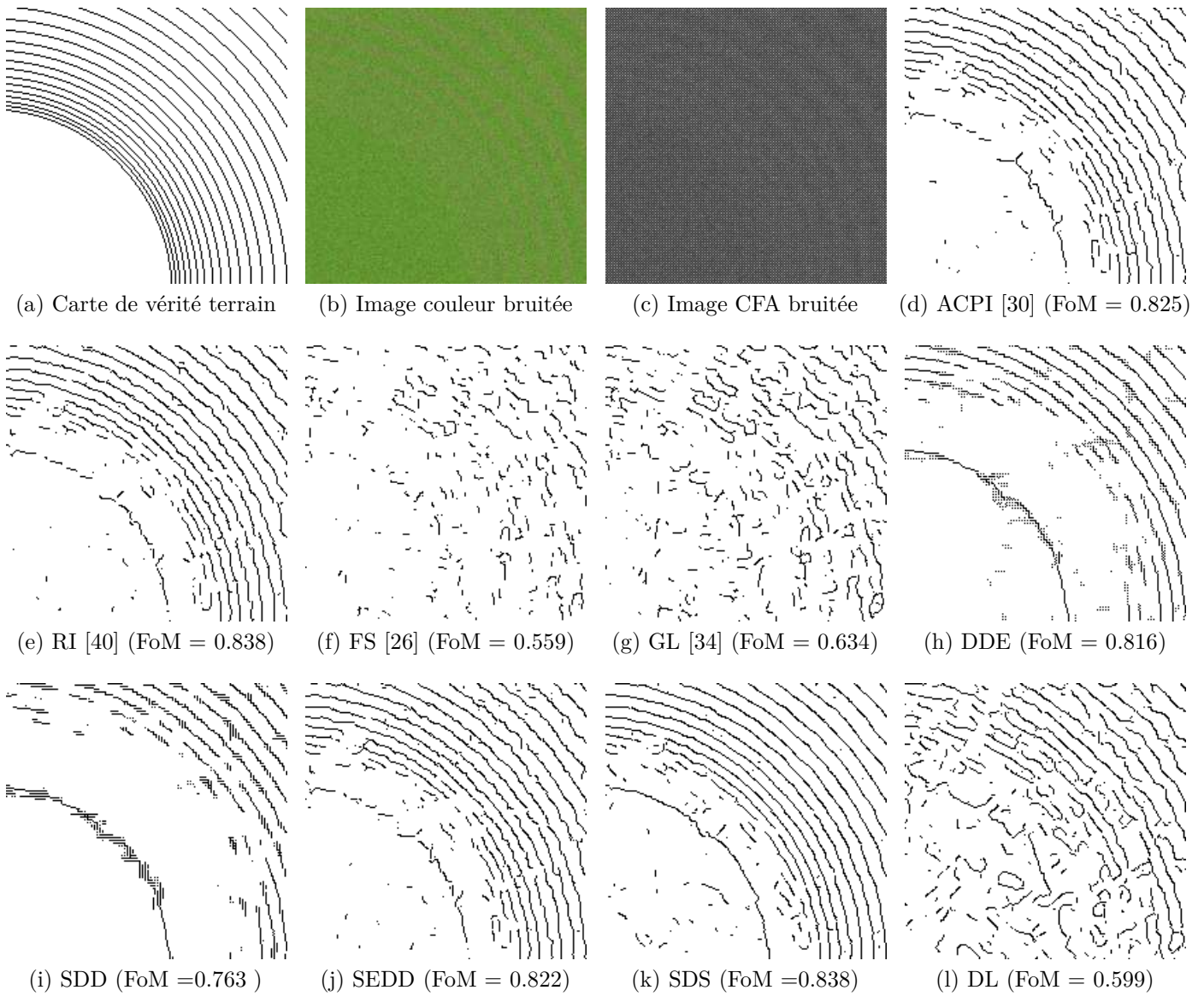


Figure 4.7 : Résultats de détection des contours à partir de l'image CFA \tilde{I}^{CFA} (c) échantillonnée à partir de l'image couleur bruitée $\tilde{\mathbf{I}}$ (b) générée avec $\mathbf{c} = (100, 150, 50)$, $\delta\mathbf{c} = (15, -6, 15)$, $\rho = 2$, et $\sigma = 8$ (PSNR=22.87 dB).

Tableau 4.4 : Nombre d'opérations élémentaires par pixel nécessaire pour estimer les dérivées partielles.

| | Dérivation simple (SD) | | | | | | Dérivation de Deriche (DD) | | | | | | | | |
|---------------------------------------|------------------------|--------------|--------------|--------------------|--------------|--------------|----------------------------|---------------|---------------|---------------|---------------|--------------------|-------------------|---------------|---------------|
| | Approche vectorielle | | | Approches scalaire | | | Approches vectorielle | | | | | | Approche scalaire | | |
| | ACPI [30] | RI [40] | SDE | FS [26] | GL [35] | DL | ACPI [30] | RI [40] | DDE | SEDD | SDD | SDS | FS [26] | GL [35] | DL |
| Dématriçage / Estimation de luminance | 30 | 1212 | 0 | 1099 | 10 | 30 | 30 | 1212 | 0 | 0 | 0 | 0 | 1099 | 10 | 30 |
| Dérivée partielle Estimation | 2×6 | 2×6 | 2×8 | 2×2 | 2×2 | 2×2 | 2×93 | 2×93 | 2×37 | 2×68 | 2×93 | $90 + 2 \times 48$ | 2×31 | 2×31 | 2×16 |
| Total | 42 | 1224 | 16 | 1103 | 14 | 34 | 216 | 1398 | 74 | 136 | 186 | 186 | 1161 | 72 | 62 |
| Temps de calcul (ms) | 162 | -- | 63 | -- | 59 | 129 | 754 | -- | 287 | 672 | 734 | 729 | -- | 275 | 255 |

dématriçage ou d'estimation de la luminance, nous considérons que toutes les méthodes ne diffèrent que dans le calcul des dérivées partielles (*cf.* figure 4.2)². Le tableau 4.4 illustre le nombre d'opérations élémentaires par pixel requises par chaque méthode.

Considérons d'abord les méthodes de dématriçage et d'estimation de luminance. L'algorithme ACPI [30] appliqué à une image CFA de taille $M \times N$ nécessite 23 opérations pour estimer $\hat{I}^G(P)$ en chaque pixel $P \in S^R \cup S^B$ et 37 opérations pour estimer $\hat{I}^R(P)$ ou $\hat{I}^B(P)$ en $P \in S^{G,R} \cup S^{G,B}$. Comme chaque sous-ensemble $S^R, S^B, S^{G,B}$ et $S^{R,B}$ est de cardinal $\frac{MN}{4}$, nous avons donc besoin de $2 \times 23 \frac{MN}{4} + 2 \times 37 \frac{MN}{4}$ opérations afin d'estimer $\hat{\mathbf{I}}$.

La méthode de dématriçage basée sur l'interpolation résiduelle [40] est de loin la plus complexe. Nous évaluons sa complexité algorithmique à 1104 opérations afin d'estimer chaque niveau manquant de la composante G^3 et 440 opérations pour la composante R ou B^4 .

Leung *et al.* [45] ont étudié en détail la complexité algorithmique de leur méthode de dématriçage basée sur FS [26]. En se basant sur leur travail, nous évaluons le nombre de multiplications nécessaires pour estimer \hat{I}^L en un pixel donné par : $Mul = 2(P_1 + P_2 + 1) + M_1M_2 + 2N_1N_2$, où les paramètres sont les ordres des filtres. Ceci omet toute multiplication par un opérande ainsi que les additions, mais nous pouvons évaluer le

2. En pratique, nous utilisons l'approche de Di Zenzo sur $\hat{\mathbf{I}}^L \doteq (\hat{I}^L, \hat{I}^L, \hat{I}^L)$, qui est équivalente à l'approche de Canny sur \hat{I}^L . Dans les deux cas nous utilisons un seuillage simple.

3. 155 opérations pour l'interpolation résiduelle du laplacien minimisé (*Minimized-Laplacian Residual Interpolation*, MLRI) de chacune des six paires composante couleur/direction, 2 opérations pour calculer chacune des deux différences couleur directionnelles, 84 opérations pour chacune des deux combinaisons de différences couleur, et 2 opérations pour ajouter les valeurs de pixels observées et les différences couleurs combinées. Notons que MLRI utilise une fenêtre 3×3 pour le filtrage guidé (GF).

4. 11 opérations pour l'estimation initiale du canal vert basé sur ACPI [30], puis 434 opérations pour le MLRI de chacun des deux canaux rouge et bleu en utilisant une fenêtre 5×5 pour le filtrage guidé.

nombre d'additions nécessaires par : $Add = 2((P_1 - 1) + (P_2 - 1) + 1) + (M_1M_2 - 1) + 2(N_1N_2 - 1)$. Les différents paramètres des filtres utilisés sont donnés dans l'algorithme que les auteurs ont mis à la disposition de la communauté scientifique : $P_1 = P_2 = 11$ et $M_1 = M_2 = N_1 = N_2 = 13$. L'estimation de \hat{I}^L en chaque pixel par la méthode FS nécessite donc $Mul + Add = 1099$ opérations.

GL [35] nécessite seulement 10 opérations. En effet, le filtre de convolution gaussien peut être implémenté comme deux filtres séparables mono-dimensionnels de taille 1×3 et 3×1 . L'estimation de \hat{I}_α^{DL} en chaque pixel nécessite 14 additions et 16 multiplications (*cf.* équations (3.8), (3.9), (3.40) et (3.41)) en supposant les coefficients $\{a_i\}_{i=1}^4$ et $\{b_i\}_{i=1}^2$ préalablement calculés.

Considérons maintenant l'étape d'estimation des dérivées partielles. Comme une dérivation simple (SD) nécessite 2 opérations en chaque pixel P (*cf.* équation (3.2)), le calcul de \hat{I}_x^k ($k \in \{R, G, B\}$), I_x^{CFA} , ou \hat{I}_x^L nécessite $2MN$ opérations. Dans I_x^{CFA} , deux composantes de la dérivée partielle selon x sont manquantes et doivent être estimées (méthode SDE), ce qui nécessite 2 additions (si $P \in S^R \cup S^B$) ou 6 additions (si $P \in S^G$) et 2 multiplications (*cf.* équation (3.15)). Nous avons donc besoin de $2MN + 24\frac{MN}{4}$ opérations pour calculer les trois composantes de $\dot{\mathbf{I}}_x$ (ou de $\dot{\mathbf{I}}_y$) à partir de I^{CFA} .

La dérivation de Deriche (DD) nécessite 7 additions et 8 multiplications pour l'étape de lissage et 7 additions et 9 multiplications pour l'étape de dérivation en chaque pixel (*cf.* équations (3.8) à (3.11)). Le calcul de $\hat{I}_{x,\alpha,1}^k$ ($k \in \{R, G, B\}$), $I_{x,\alpha,2}^{CFA}$ ou $\hat{I}_{x,\alpha,1}^L$ nécessite donc $31MN$ opérations, et la dérivation de \hat{I}_α^{DL} pour calculer $\hat{I}_{x,\alpha,1}^{DL}$ nécessite seulement $16MN$ opérations.

Dans $I_{x,\alpha,2}^{CFA}$ obtenue avec DDE, deux composantes de dérivée partielle selon x manquent en chaque pixel et doivent donc être estimées. Cette estimation nécessite 6 additions (si $P \in S^R \cup S^B$) ou 2 additions (si $P \in S^G$) ainsi que 2 multiplications (*cf.* équation (3.18)). En résumé, Nous avons donc besoin de $31MN + 24\frac{MN}{4}$ opérations pour calculer $\dot{\mathbf{I}}_{x,\alpha,2}$ (ou $\dot{\mathbf{I}}_{y,\alpha,2}$).

Le calcul des composantes couleur lissées dans SEDD nécessite $15MN$ opérations pour calculer $\bar{I}_{2\alpha,2}^{CFA}$ (*cf.* équations (3.8) et (3.9)), puis $20\frac{MN}{4}$ opérations afin d'estimer les composantes couleur lissées manquantes (*cf.* équation (3.19)) et finalement $3 \times 16MN$ opérations pour l'étape de dérivation (*cf.* équations (3.20) et (3.21)). Nous avons donc besoin de $68MN$ opérations pour calculer $\dot{\mathbf{I}}_{x,\alpha,1}$ (ou $\dot{\mathbf{I}}_{y,\alpha,1}$).

Dans la méthode SDD, le calcul des trois composantes de dérivée partielle selon x directement à partir de l'images CFA nécessite exactement le même nombre d'opérations

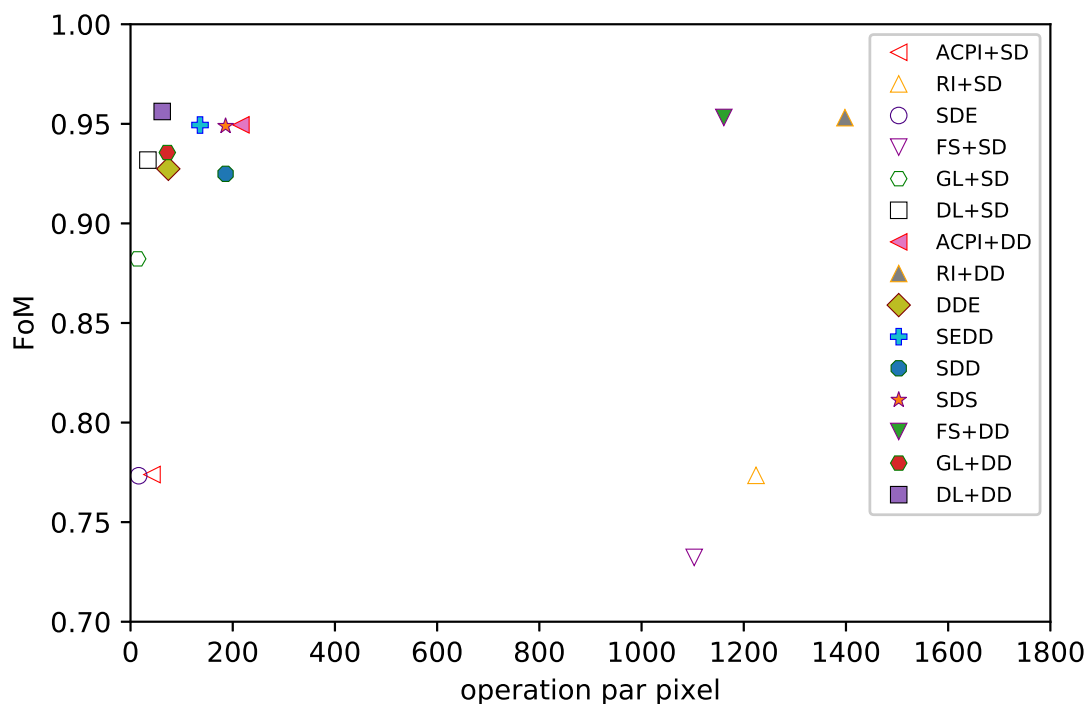


Figure 4.8 : Performance des différentes méthodes de détection des contours par rapport à leurs complexité algorithmique.

que celui requis par DD dans une image pleinement définie. Nous avons donc besoin de $3 \times 31MN$ opérations pour estimer $\mathbf{I}_{x,\alpha,2}$ avec SDD. Dans la méthode SDS, les trois composantes couleur lissées sont calculées une seule fois et utilisées par les deux filtres de dérivation (selon x et y). Cette étape nécessite $3 \times 30MN$ opérations en chaque pixel. Le nombre d'opérations nécessaires afin de dériver ensuite cette image lissée pleinement définie est de 3×16 .

La figure 4.8 résume les performances des différentes méthodes de détection des contours sur les images synthétiques bruitées (*cf.* tableau. 4.2a) en terme de complexité algorithmique. Nous constatons que dans le cas où nous recherchons une méthode de détection des contours très peu gourmande en temps de calcul, il est intéressant d'utiliser SD combinée à DL. Pour une méthode avec un temps de calcul faible mais avec des performances bien meilleures, SD combinée à DL représente un très bon rapport performance de détection/temps de calcul. La méthode DL combinée à DD représente le meilleur compromis entre la qualité de détection et la complexité algorithmique. En effet, seulement 62 opérations sont nécessaires afin d'obtenir la meilleure performance de détection des contours par rapport à toutes les autres méthodes. En analysant l'avant-

dernière ligne du tableau 4.4, nous constatons que pour avoir des performances similaires à celle de DL couplé avec DD (sur une image CFA), nous avons besoin de 22 fois plus d'opérations élémentaires en utilisant RI (sur une image couleur dématricée) ou de 18 fois plus d'opérations en utilisant FS (sur une image de luminance).

La dernière ligne du tableau 4.4 montre le temps de calcul requis par chaque méthode. Ces temps de calcul sont mesurés sur une machine quadri-processeurs de fréquence 2.4 GHz avec 8 Go de RAM. Toutes les méthodes sont implémentées en Java sous ImageJ [66] sauf FS et RI pour lesquelles nous disposons d'une implémentation sous Matlab. Comme de plus ce sont les plus complexes (nombre d'opérations par pixel très élevé) les temps de calcul qu'elles requièrent sont omis. Nous constatons que les temps de calcul sont conformes aux nombres d'opérations. La méthode DL qui présente les meilleures performances de détection des contours est capable de traiter quatre images par seconde ce qui permet d'envisager une implémentation en temps réel.

5 Conclusion

Dans ce chapitre, nous avons proposé une évaluation objective des différentes méthodes proposées pour la détection des contours à partir de l'image CFA. L'absence d'une base d'images couleur disposant d'une carte vérité terrain dans la littérature nous a poussé à construire notre propre base d'images couleur synthétiques. Cette base est construite à partir d'une carte des contours vérité terrain et de manière à pouvoir évaluer la qualité de détection et de localisation des contours. Nous avons donc fait varier la hauteur de la transition couleur, la largeur de transition et le niveau de bruit qui peuvent influencer les résultats de détection des contours.

Nous avons comparé les performances de détection des contours obtenues par les méthodes vectorielles, en mettant en regard celles proposées pour l'image CFA et celles analysant les images couleur dématricées avec deux méthodes : l'une performante mais gourmande en temps de calcul [40] et l'autre moins performante mais peu coûteuse en temps de calcul [30]. Nous avons suivi la même procédure dans l'approche scalaire où nous avons comparé nos résultats à ceux d'une méthode d'estimation de luminance (visuellement satisfaisante) exploitée pour le dématricage [26] et une estimation grossière de la luminance exploitée pour la détection des contours [34].

Les résultats obtenus montrent que pour des images débruitées, toutes les méthodes fournissent des performances similaires avec un temps de calcul avantageux pour nos méthodes. Dans le cas des images bruitées, notre méthode basée sur l'approche scalaire

(DL) fournit les meilleures performances de détection des contours. Les deux méthodes sophistiquées de dématricage et d'estimation de la luminance fournissent des contours de très bonne qualité mais requièrent des temps de calcul conséquents. L'application du filtre gaussien sur l'image CFA afin d'estimer la luminance est intéressante si l'on privilégie le gain en temps de calcul sur les performances de détection. Soulignons que la qualité de détection des contours avec les méthodes DDE et SDD est toujours inférieure à celles des autres méthodes en raison du voisinage utilisé dans ces deux méthodes. Les méthodes ACPI, SDS et SEDD fournissent des résultats similaires avec un avantage pour SDS et SEDD en terme du temps de calcul. La méthode DL combinée à la dérivation de Deriche représente le meilleur compromis entre la qualité de détection et la complexité algorithmique.

Conclusion

Dans ce manuscrit divisé en quatre chapitres, nous avons présenté nos contributions afin de calculer les dérivées partielles à partir d'une image CFA. Nous faisons ici une synthèse de ces chapitres et terminons par les perspectives issues de ce travail.

1 Synthèse des chapitres

1.1 Chapitre 1 : Image numérique couleur

Dans ce chapitre, nous avons commencé par présenter la perception humaine de la couleur et certains éléments de représentation de la couleur nécessaires à la compréhension de ce manuscrit. Nous avons présenté les deux familles de caméras permettant d'acquérir des images numériques couleur. Contrairement aux caméras tri-capteur, les caméras mono-capteur sont dotées d'une matrice de filtres couleur (*Color Filter Array*) disposée sur la surface du capteur afin de rendre chaque photosite sensible à une bande de longueurs d'onde. Il existe plusieurs configurations de CFA mais celle de Bayer est largement la plus répandue dans la littérature. C'est donc celle que nous avons utilisée dans nos travaux. L'image CFA ainsi formée se compose de 50% de pixels caractérisés par la composante couleur verte et disposés en quinconce, de 25% de pixels caractérisés par la composante rouge et autant par la composante bleue, disposés selon un motif régulier. Cette image ne comporte donc en chaque pixel qu'une seule des trois composantes nécessaires à la représentation d'une couleur. Dans la dernière partie de ce chapitre, nous avons vu de manière générique comment la procédure de dématricage permet de passer de l'image CFA à l'image couleur pleinement définie en estimant les deux composantes couleur manquantes en chaque pixel.

1.2 Chapitre 2 : Dématricage

Nous avons présenté les méthodes de dématricage les plus simples, comme celle qui procède par interpolation bilinéaire, pour décrire ensuite les plus sophistiquées comme

les méthodes fréquentielles ou celles basées sur les résidus. La plupart des algorithmes de dématricage couplent des hypothèses qui portent sur la corrélation spatiale avec celles relatives à la corrélation spectrale. Quelles que soient les hypothèses sur lesquelles elles s'appuient, les méthodes de dématricage ont toutes été développées pour produire des images les plus « visuellement satisfaisantes » possibles. Pour ce faire, plusieurs méthodes [46, 26] appliquent un filtre passe-bas à l'image CFA afin d'atténuer les artéfacts générés par le dématricage dans les zones de hautes fréquences. Cependant, cette étape a tendance à altérer ces informations de hautes fréquences qui peuvent correspondre à des contours.

Par conséquent, quelques travaux proposent d'éviter la procédure de dématricage et d'analyser directement l'image CFA. Ils portent principalement sur la caractérisation des textures ainsi que sur la compression de l'image CFA. Par ailleurs, à notre connaissance, seuls Chen *et al.* [13] ont étudié la possibilité d'utiliser l'image CFA pour la détection des contours. Nous avons vu que la proposition porte sur un filtre de lissage gaussien et un filtre dérivateur laplacien propres aux spécificités de l'image CFA. Même si cette proposition n'a pas été évaluée objectivement, elle ouvre d'intéressantes pistes pour l'utilisation de l'image CFA dans des procédures d'analyse de bas niveau.

1.3 Chapitre 3 : Calcul du gradient à partir de l'image CFA

Dans ce chapitre, nous avons proposé une démarche originale qui consiste principalement à transposer les filtres de Deriche à l'image CFA. En effet, après avoir décrit le gradient couleur de Di Zenzo et détaillé le filtre de Deriche, nous avons proposé deux approches principales afin de calculer les dérivées partielles directement à partir d'une image CFA. L'approche vectorielle consiste à calculer les trois composantes des dérivées partielles en chaque pixel et l'approche scalaire estime une luminance adaptée à la détection des contours.

L'approche vectorielle a été déclinée en cinq méthodes : par estimation des dérivées simples (SDE), estimation des dérivées de Deriche (DDE), estimation des composantes couleur lissées (SEDD), dérivation par le filtre de Deriche décalé (SDD) et enfin lissage par le filtre de Deriche décalé (SDS). Les deux premières méthodes estiment les deux composantes de dérivée partielle manquantes en chaque pixel. La méthode SEDD estime les deux composantes lissées manquantes en chaque pixel pour calculer ensuite les dérivées partielles à partir de l'image lissée pleinement définie. Enfin les deux méthodes SDD et SDS fournissent les dérivées partielles pleinement définies, soit en calculant

directement leurs trois composantes en chaque pixel (SDD), soit en dérivant les trois composantes couleur lissées préalablement calculées (SDS).

Dans l'approche scalaire, nous avons proposé une méthode (DL) qui estime une image de luminance adaptée à la détection des contours en utilisant un filtre de lissage bidimensionnel de Deriche. La démarche que nous proposons nécessite d'ajuster un seul paramètre α pour les deux étapes d'estimation de la luminance et de dérivation, avec pour objectif de conserver les propriétés d'optimalité du filtre de Deriche.

1.4 Chapitre 4 : Évaluation objective de la qualité de détection des contours

Dans ce chapitre, nous avons mené une évaluation objective des différentes méthodes de détection des contours à partir de l'image CFA. La littérature ne proposant aucune base d'images couleur avec carte de contours vérité terrain, nous avons construit notre propre base d'images couleur synthétiques. Pour pouvoir évaluer la qualité de détection et de localisation des contours, nous partons d'une carte des contours vérité terrain et faisons varier la hauteur et la largeur de la transition couleur ainsi que le niveau de bruit, autant de paramètres qui peuvent influencer la qualité de détection des contours.

Nous avons comparé les performances de nos méthodes de détection des contours à celles obtenues par deux méthodes de dématricage. Les résultats obtenus montrent que pour des images débruitées, toutes les méthodes fournissent des performances similaires avec un temps de calcul avantageux pour nos méthodes. Dans le cas des images bruitées, notre méthode basée sur l'approche scalaire (DL) fournit les meilleures performances de détection des contours, meilleures mêmes que les deux méthodes sophistiquées mais gourmandes d'estimation de l'image couleur et de luminance. Appliquer le filtre de Deriche sur une image de luminance préalablement estimée par un lissage de l'image CFA est la piste la plus intéressante quand il s'agit de privilégier le temps de calcul par rapport aux performances de détection. En effet, notre méthode DL combinée à la dérivation de Deriche représente le meilleur compromis entre la qualité de détection des contours et la complexité algorithmique.

2 Perspectives

Notre travail ouvre de nombreuses perspectives relatives à l'exploitation des dérivées partielles calculées à partir de l'image CFA ou à l'amélioration de l'estimation de ces dérivées.

2.1 Dématriçage

Les méthodes de dématriçage exploitant la corrélation spatiale estiment un gradient pour soit déterminer la direction d'interpolation, soit pondérer les contributions des voisins dans l'interpolation. Les dérivées partielles premières, calculées à partir de l'image CFA grâce aux méthodes proposées, pourraient être utilisées pour calculer en chaque pixel la direction et la norme du gradient couleur. Il serait alors intéressant d'évaluer l'amélioration apportée par ces estimations du gradient en termes de qualité de dématriçage et de complexité algorithmique.

2.2 Filtres optimaux sur l'image CFA

Les performances des méthodes de détection des contours qui s'appuient sur le filtre de Deriche sont très sensibles à l'ajustement du paramètre α . Pour faire face au compromis entre la bonne détection et la bonne localisation des contours, nous avons choisi la valeur de α qui permet à chaque méthode d'atteindre le FoM le plus élevé. Cet ajustement dépend fortement de la vérité terrain requise pour calculer le FoM. Pour s'affranchir de la vérité terrain, il faudrait développer une démarche qui permet de sélectionner la valeur α en fonction des propriétés statistiques des images à analyser.

Nos méthodes s'appuient sur les filtres de Deriche car ce sont les plus utilisés dans la littérature. Cependant, nos développements ne sont pas spécifiques aux filtres utilisés et pourraient en exploiter d'autres. Il serait intéressant d'adapter nos méthodes au filtre proposé par Shen et Castan [67] sensible aux transitions de niveaux de type échelon ou à celui proposé par Bourennane *et al.* [9] qui est une amélioration du filtre de Deriche pour la détection des transitions de niveaux de type rampe. Une évaluation des performances étendues à ces filtres permettrait de faire ressortir les méthodes les mieux adaptées à la détection de tous les types de contours à partir de l'image CFA.

2.3 Évaluation des performances de détection des contours

Afin d'évaluer la qualité de détection des contours, nous avons proposé une base d'images couleur synthétiques qui sera mise à la disposition de la communauté scientifique. Même si les propriétés de corrélation au sein des images générées respectent celles calculées pour des images de scènes naturelles, tous les types de contours ne sont pas représentés. Une des perspectives de notre travail est d'évaluer nos méthodes sur une base d'images couleur de scènes naturelles disposant d'une vérité terrain probante. Pour contourner cette épineuse question relative à la vérité terrain, il serait également intéres-

sant d'évaluer la fidélité des gradients estimés grâce à une mesure objective, comme le rapport signal-sur-bruit pic-à-pic (PSNR), permettant de comparer le gradient estimé et le gradient couleur de référence. Enfin, nous avons retenu le FoM pour évaluer la qualité des contours détectés. Ce critère est cependant limité, car il mesure de manière imparfaite la connexité des pixels contours détectés. Or, il est essentiel que la carte des contours présente des lignes fermées afin que les méthodes de haut niveau puissent interpréter le contenu de l'image analysée.

2.4 Détection de points d'intérêt

Les contributions de cette thèse s'inscrivent dans la détection des contours à partir de l'image CFA. La méthode scalaire proposée (DL) présente la meilleure performance de détection combinée à un temps de calcul relativement faible. Cette démarche mériterait d'être étendue à la détection des points d'intérêt directement à partir de l'image CFA. Il faudrait alors étendre nos méthodes à l'estimation des dérivées secondes nécessaires à l'extraction des points d'intérêt. L'implémentation de telles méthodes au sein de circuits embarqués dans des caméras couleur équipées d'un seul capteur permettrait alors d'accomplir une détection des points d'intérêt en temps réel.

Annexe A

Calcul des coefficients du filtre décalé de Deriche

Les équations données ici sont extraites de la thèse de Fiorio [28].

1 Quelques calculs préliminaires

Pour $X \in [0,1[$,

$$\sum_{n=0}^{\infty} X^n = \frac{1}{1-X}. \quad (\text{A.1})$$

Donc :

$$\sum_{n=1}^{\infty} X^n = \frac{1}{1-X} - 1 = \frac{X}{1-X}, \quad (\text{A.2})$$

$$\sum_{n=0}^{\infty} nX^n = \frac{X}{(1-X)^2}, \quad (\text{A.3})$$

$$\sum_{n=0}^{\infty} n^2 X^n = \frac{X(X+1)}{(1-X)^3}. \quad (\text{A.4})$$

2 Filtre de lissage décalé

2.1 Coefficient de normalisation

La fonction de lissage décalé de Deriche est donnée par :

$$\mathcal{L}_{\alpha, \frac{1}{2}}(n) = m_{\frac{1}{2}} \left(\alpha \left| n - \frac{1}{2} \right| + 1 \right) e^{-\alpha |n - \frac{1}{2}|}.$$

Cette fonction de lissage est normalisée de la manière suivante :

$$\sum_{n=-\infty}^{\infty} \mathcal{L}_{\alpha, \frac{1}{2}}(n) = 1.$$

Donc :

$$\mathcal{L}_{\alpha, \frac{1}{2}}(n) = \begin{cases} m_{\frac{1}{2}}(\alpha(n - \frac{1}{2}) + 1) e^{-\alpha(n - \frac{1}{2})} & \text{if } n > 0, \\ m_{\frac{1}{2}}(1 - \alpha(n - \frac{1}{2})) e^{\alpha(n - \frac{1}{2})} & \text{if } n \leq 0, \end{cases}$$

et, par conséquent :

$$m_{\frac{1}{2}} \left(\sum_{n=-\infty}^0 (1 - \alpha(n - \frac{1}{2})) e^{\alpha(n - \frac{1}{2})} + \sum_{n=1}^{\infty} (\alpha(n - \frac{1}{2}) + 1) e^{-\alpha(n - \frac{1}{2})} \right) = 1$$

$$m_{\frac{1}{2}} \sum_{n=1}^{\infty} (\alpha(n - \frac{1}{2}) + 1) e^{-\alpha(n - \frac{1}{2})} = \frac{1}{2}$$

$$m_{\frac{1}{2}} e^{\frac{\alpha}{2}} \left((1 - \frac{\alpha}{2}) \sum_{n=1}^{\infty} e^{-\alpha n} + \alpha \sum_{n=1}^{\infty} n e^{-\alpha n} \right) = \frac{1}{2}$$

À partir des équations (A.2) et (A.3), il vient :

$$m_{\frac{1}{2}} e^{\frac{\alpha}{2}} \left((1 - \frac{\alpha}{2}) \frac{e^{-\alpha}}{(1 - e^{-\alpha})} + \alpha \frac{e^{-\alpha}}{(1 - e^{-\alpha})^2} \right) = \frac{1}{2}.$$

Finalement, le coefficient de normalisation de la fonction de lissage décalée de Deriche est :

$$m_{\frac{1}{2}} = \frac{1}{2 e^{\frac{\alpha}{2}}} \left(\frac{(1 - e^{-\alpha})^2}{e^{-\alpha}(1 + \frac{\alpha}{2}) - e^{-2\alpha}(1 - \frac{\alpha}{2})} \right).$$

2.2 Équations de récurrence

2.2.1 Partie causale

Soit $\mathcal{L}_{\alpha, \frac{1}{2}}^{-}(n)$ la partie causale de la fonction de lissage décalé :

$$\mathcal{L}_{\alpha, \frac{1}{2}}^{-}(n) = \begin{cases} m_{\frac{1}{2}}(\alpha(n - \frac{1}{2}) + 1) e^{-\alpha(n - \frac{1}{2})} & \text{if } n > 0, \\ 0 & \text{if } n \leq 0. \end{cases}$$

Soit $F_L^{-}(Z)$ la transformée en Z de cette fonction :

$$F_L^{-}(Z) = \sum_{n=1}^{\infty} Z^{-n} \mathcal{L}_{\alpha, \frac{1}{2}}^{-}(n),$$

qui s'écrit encore :

$$F_L^-(Z) = m_{\frac{1}{2}} \sum_{n=1}^{\infty} \left(\alpha \left(n - \frac{1}{2} \right) + 1 \right) e^{-\alpha \left(n - \frac{1}{2} \right)} Z^{-n}$$

$$F_L^-(Z) = m_{\frac{1}{2}} e^{\frac{\alpha}{2}} \left(\sum_{n=1}^{\infty} e^{-\alpha n} Z^{-n} + \alpha \sum_{n=1}^{\infty} n e^{-\alpha n} Z^{-n} - \frac{1}{2} \alpha \sum_{n=1}^{\infty} e^{-\alpha n} Z^{-n} \right).$$

Posons $X = e^{-\alpha} Z^{-1}$; il vient :

$$F_L^-(Z) = m_{\frac{1}{2}} e^{\frac{\alpha}{2}} \left(\sum_{n=1}^{\infty} X^n + \alpha \sum_{n=1}^{\infty} n X^n - \frac{1}{2} \alpha \sum_{n=1}^{\infty} X^n \right)$$

$$= m_{\frac{1}{2}} e^{\frac{\alpha}{2}} \left(\frac{X}{(1-X)} + \alpha \frac{X}{(1-X)^2} - \frac{1}{2} \alpha \frac{X}{(1-X)} \right)$$

$$= m_{\frac{1}{2}} e^{\frac{\alpha}{2}} \left(1 - \frac{\alpha}{2} \right) \frac{X}{(1-X)} + m_{\frac{1}{2}} \alpha e^{\frac{\alpha}{2}} \frac{X}{(1-X)^2},$$

avec :

$$F_L^-(Z) = \frac{Y^-(Z)}{X(Z)}, \quad \beta = \left(1 - \frac{\alpha}{2} \right), \quad \gamma = \beta e^{\frac{\alpha}{2}} \quad \text{et} \quad \delta = \alpha e^{\frac{\alpha}{2}}.$$

Donc :

$$\frac{Y^-(Z)}{X(Z)} = m_{\frac{1}{2}} \gamma \frac{e^{-\alpha} Z^{-1}}{(1 - e^{-\alpha} Z^{-1})} + m_{\frac{1}{2}} \delta \frac{e^{-\alpha} Z^{-1}}{(1 - e^{-\alpha} Z^{-1})^2},$$

$$= m_{\frac{1}{2}} \frac{e^{-\alpha} (\gamma + \delta) Z^{-1} - \gamma e^{-2\alpha} Z^{-2}}{1 - 2 e^{-\alpha} Z^{-1} + e^{-2\alpha} Z^{-2}}.$$

D'où il découle que :

$$Y^+(Z) = m_{\frac{1}{2}} e^{-\alpha} (\gamma + \delta) X(Z) Z^{-1} - m_{\frac{1}{2}} \gamma e^{-2\alpha} X(Z) Z^{-2} + 2 e^{-\alpha} Y^-(Z) Z^{-1} - e^{-2\alpha} Y^-(Z) Z^{-2}.$$

Posons :

$$C_0 = m_{\frac{1}{2}} e^{-\alpha} (\gamma + \delta) = m_{\frac{1}{2}} \left(1 + \frac{\alpha}{2} \right) e^{-\frac{\alpha}{2}}, \quad a = \frac{\alpha - 2}{\alpha + 2} e^{-\alpha}, \quad b = 2 e^{-\alpha} \quad \text{et} \quad c = -e^{-2\alpha}.$$

Nous avons alors dans l'espace réel :

$$\boxed{Y^-(n) = C_0 X(n-1) + a C_0 X(n-2) + b Y^-(n-1) + c Y^-(n-2).}$$

2.2.2 Partie anticausale

Soit $\mathcal{L}_{\alpha, \frac{1}{2}}^+(n)$ la partie anticausale de la fonction de lissage décalé :

$$\mathcal{L}_{\alpha, \frac{1}{2}}^+(n) = \begin{cases} 0 & \text{if } n > 0, \\ m_{\frac{1}{2}}(1 - \alpha(n - \frac{1}{2})) e^{\alpha(n - \frac{1}{2})} & \text{if } n \leq 0. \end{cases}$$

Soit $F_L^+(Z)$ la transformée en Z de cette fonction :

$$F_L^+(Z) = \sum_{n=-\infty}^0 Z^{-n} \mathcal{L}_{\alpha, \frac{1}{2}}^+(n) .$$

On peut écrire :

$$\begin{aligned} F_L^+(Z) &= m_{\frac{1}{2}} \sum_{n=-\infty}^0 (1 - \alpha(n - \frac{1}{2})) e^{\alpha(n - \frac{1}{2})} Z^{-n} \\ &= m_{\frac{1}{2}} \sum_{n=-\infty}^0 e^{\alpha n} e^{\frac{-\alpha}{2}} Z^{-n} - m_{\frac{1}{2}} \sum_{n=-\infty}^0 \alpha n e^{\alpha n} e^{\frac{-\alpha}{2}} Z^{-n} + \frac{1}{2} m_{\frac{1}{2}} \sum_{n=-\infty}^0 \alpha e^{\alpha n} e^{\frac{-\alpha}{2}} Z^{-n} \\ &= m_{\frac{1}{2}} e^{\frac{-\alpha}{2}} \sum_{n=0}^{\infty} (e^{-\alpha} Z)^n + m_{\frac{1}{2}} \alpha e^{\frac{-\alpha}{2}} \sum_{n=0}^{\infty} n (e^{-\alpha} Z)^n + \frac{1}{2} m_{\frac{1}{2}} \alpha e^{\frac{-\alpha}{2}} \sum_{n=0}^{\infty} (e^{-\alpha} Z)^n . \end{aligned}$$

En utilisant les équations (A.1) et (A.3), il vient :

$$F_L^+(Z) = m_{\frac{1}{2}} e^{\frac{-\alpha}{2}} \frac{(1 + \frac{\alpha}{2}) + (\frac{\alpha}{2} - 1)Z e^{-\alpha}}{1 - 2Z e^{-\alpha} + Z^2 e^{-2\alpha}} .$$

Donc :

$$Y^+(Z) = m_{\frac{1}{2}} e^{\frac{-\alpha}{2}} (1 + \frac{\alpha}{2}) X(Z) + m_{\frac{1}{2}} e^{\frac{-\alpha}{2}} (\frac{\alpha}{2} - 1) e^{-\alpha} X(Z) Z + 2 e^{-\alpha} Y^+(Z) Z - e^{-2\alpha} Y^+(Z) Z^2 .$$

Dans l'espace réel :

$$Y^+(n) = m_{\frac{1}{2}} e^{\frac{-\alpha}{2}} (1 + \frac{\alpha}{2}) X(n) + m_{\frac{1}{2}} e^{\frac{-\alpha}{2}} (\frac{\alpha}{2} - 1) e^{-\alpha} X(n+1) + 2 e^{-\alpha} Y^+(n+1) - e^{-2\alpha} Y^+(n+2) .$$

Donc :

$$Y^+(n) = C_0 X(n) + a C_0 X(n+1) + b Y^+(n+1) + c Y^+(n+2) .$$

2.2.3 Expression finale du filtre

L'expression finale du filtre de lissage décalé de Deriche est donnée par :

$$Y^-(n) = C_0X(n-1) + aC_0X(n-2) + bY^-(n-1) + cY^-(n-2),$$

$$Y^+(n) = C_0X(n) + aC_0X(n+1) + bY^+(n+1) + cY^+(n+2),$$

$$Y(n) = Y^-(n) + Y^+(n).$$

3 Filtre de dérivation décalé

3.1 Coefficient de normalisation

La fonction de dérivation décalée de Deriche est donnée par :

$$\mathcal{D}_{\alpha, \frac{1}{2}}(n) = -m'_{\frac{1}{2}}\alpha^2\left(n - \frac{1}{2}\right)e^{-\alpha|n-\frac{1}{2}|}.$$

Cette fonction est normalisée comme suit :

$$\sum_{n=1}^{\infty} \mathcal{D}_{\alpha, \frac{1}{2}}(n) = -1.$$

Donc :

$$-m'_{\frac{1}{2}}\alpha^2 \sum_{n=1}^{\infty} \left(n - \frac{1}{2}\right)e^{-\alpha(n-\frac{1}{2})} = -1$$

$$m'_{\frac{1}{2}}\alpha^2 e^{\frac{\alpha}{2}} \left(\sum_{n=1}^{\infty} n e^{-\alpha n} - \frac{1}{2} \sum_{n=1}^{\infty} e^{-\alpha n} \right) = 1.$$

À partir des équations (A.2) et (A.3), on peut écrire :

$$\frac{1}{m'_{\frac{1}{2}}\alpha^2 e^{\frac{\alpha}{2}}} = \frac{e^{-\alpha}}{(1-e^{-\alpha})^2} - \frac{1}{2} \left(\frac{e^{-\alpha}}{(1-e^{-\alpha})} \right).$$

Et finalement, $m'_{\frac{1}{2}}$ est donnée par :

$$m'_{\frac{1}{2}} = \frac{2(1-e^{-\alpha})^2}{\alpha^2 e^{\frac{\alpha}{2}}(e^{-\alpha} + e^{-2\alpha})}.$$

3.2 Équations de récurrence

3.2.1 Partie causale

Soit $\mathcal{D}_{\alpha, \frac{1}{2}}^-(n)$ la partie causale de la fonction de dérivation décalée :

$$\mathcal{D}_{\alpha, \frac{1}{2}}^-(n) = \begin{cases} -m'_{\frac{1}{2}} \alpha^2 (n - \frac{1}{2}) e^{-\alpha(n - \frac{1}{2})} & \text{if } n > 0, \\ 0 & \text{if } n \leq 0. \end{cases}$$

Soit $F_D^-(Z)$ la transformée en Z de cette fonction :

$$F_D^-(Z) = \sum_{n=1}^{\infty} Z^{-n} \mathcal{D}_{\alpha, \frac{1}{2}}^-(n).$$

On peut écrire :

$$\begin{aligned} F_D^-(Z) &= \sum_{n=1}^{\infty} Z^{-n} \left(-m'_{\frac{1}{2}} \alpha^2 (n - \frac{1}{2}) e^{-\alpha(n - \frac{1}{2})} \right) \\ &= -m'_{\frac{1}{2}} \alpha^2 e^{\frac{\alpha}{2}} \left(\sum_{n=1}^{\infty} Z^{-n} n e^{-\alpha n} - \frac{1}{2} \sum_{n=1}^{\infty} Z^{-n} e^{-\alpha n} \right) \\ &= -m'_{\frac{1}{2}} \frac{\alpha^2}{2} e^{\frac{\alpha}{2}} \frac{Z^{-1} e^{-\alpha} + Z^{-2} e^{-2\alpha}}{1 - 2Z^{-1} e^{-\alpha} + Z^{-2} e^{-2\alpha}}. \end{aligned}$$

Donc :

$$Y^-(Z) = -m'_{\frac{1}{2}} \frac{\alpha^2}{2} e^{\frac{\alpha}{2}} e^{-\alpha} X(Z) Z^{-1} - m'_{\frac{1}{2}} \frac{\alpha^2}{2} e^{\frac{\alpha}{2}} e^{-2\alpha} X(Z) Z^{-2} + 2 e^{-\alpha} Y^-(Z) Z^{-1} - e^{-2\alpha} Y^-(Z) Z^{-2}.$$

Dans l'espace réel :

$$Y^-(n) = -m'_{\frac{1}{2}} \frac{\alpha^2}{2} e^{\frac{-\alpha}{2}} X(n-1) - m'_{\frac{1}{2}} \frac{\alpha^2}{2} e^{\frac{\alpha}{2}} e^{-2\alpha} X(n-2) + 2 e^{-\alpha} Y^-(n-1) - e^{-2\alpha} Y^-(n-2),$$

avec :

$$C_1 = -m'_{\frac{1}{2}} \frac{\alpha^2}{2} e^{\frac{-\alpha}{2}}, \quad d' = e^{-\alpha}.$$

Finalement, on a :

$$\boxed{Y^-(n) = C_1 X(n-1) + d' C_1 X(n-2) + b Y^-(n-1) + c Y^-(n-2)}.$$

3.2.2 Partie anticausale

Soit $\mathcal{D}_{\alpha, \frac{1}{2}}^+(n)$ la partie anticausale de la fonction de dérivation décalée :

$$\mathcal{D}_{\alpha, \frac{1}{2}}^+(n) = \begin{cases} 0 & \text{if } n > 0, \\ -m'_{\frac{1}{2}} \alpha^2 (n - \frac{1}{2}) e^{\alpha(n - \frac{1}{2})} & \text{if } n \leq 0. \end{cases}$$

Soit $F_D^+(Z)$ la transformée en Z de cette fonction :

$$F_D^+(Z) = \sum_{n=-\infty}^0 Z^{-n} \mathcal{D}_{\alpha, \frac{1}{2}}^+(n)$$

On peut écrire :

$$\begin{aligned} F_D^+(Z) &= -m'_{\frac{1}{2}} \alpha^2 \left(\sum_{n=-\infty}^0 Z^{-n} n e^{\alpha n} e^{\frac{-\alpha}{2}} - \frac{1}{2} \sum_{n=-\infty}^0 Z^{-n} e^{\alpha n} e^{\frac{-\alpha}{2}} \right) \\ &= m'_{\frac{1}{2}} \frac{\alpha^2}{2} e^{\frac{-\alpha}{2}} \left(2 \sum_{n=0}^{\infty} n Z^n e^{-\alpha n} + \sum_{n=0}^{\infty} Z^n e^{-\alpha n} \right) \\ &= -C_1 \frac{Z e^{-\alpha} + 1}{1 - 2Z e^{-\alpha} + Z^2 e^{-2\alpha}}. \end{aligned}$$

Alors :

$$Y^+(Z) = -C_1 X(Z) - C_1 e^{-\alpha} X(Z)Z + 2e^{-\alpha} Y^+(Z)Z - e^{-2\alpha} Y^+(Z)Z^2,$$

et, par conséquent :

$$\boxed{Y^+(n) = -C_1 X(n) - d' C_1 X(n+1) + b Y^+(n+1) + c Y^+(n+2).}$$

3.2.3 Expression finale du filtre

L'expression finale du filtre de dérivation décalée de Deriche est donnée par :

$$Y^-(n) = C_1 X(n-1) + d' C_1 X(n-2) + b Y^-(n-1) + c Y^-(n-2),$$

$$Y^+(n) = -C_1 X(n) - d' C_1 X(n+1) + b Y^+(n+1) + c Y^+(n+2),$$

$$Y(n) = Y^-(n) + Y^+(n).$$

Bibliographie

- [1] Ikram E. ABDOU et William K. PRATT : Quantitative design and evaluation of enhancement/thresholding edge detectors. *Proceedings of the IEEE*, 67(5):753–763, mai 1979.
- [2] Sunil AHIRWAR et Vineet RICHARIYA : Survey on denoising method of color filter array (CFA) images for single-sensor digital cameras. *Journal of Global Research in Computer Sciences*, 3(7), juillet 2012.
- [3] Hiroki AKIYAMA, Masayuki TANAKA et Masatoshi OKUTOMI : Pseudo four-channel image denoising for noisy CFA raw data. *In Proceedings of the IEEE International Conference on Image Processing (ICIP'15)*, pages 4778–4782, Québec City, Canada, septembre 2015.
- [4] David ALLEYSSON, Sabine SÜSTRUNK et Jeanny HÉRAULT : Color demosaicing by estimating luminance and opponent chromatic signals in the Fourier domain. *In Procs. 10th Color Imaging Conference (CIC'2002) : Color Science and Engineering Systems, Technologies, Applications*, volume 10, pages 331–336, Scottsdale, Arizona, USA, novembre 2002.
- [5] David ALLEYSSON, Sabine SÜSTRUNK et Jeanny HÉRAULT : Linear demosaicing inspired by the human visual system. *IEEE Transactions on Image Processing*, 14(4):439–449, avril 2005.
- [6] Eastman Kodak AND VARIOUS PHOTOGRAPHERS : Kodak Photo CD PCD0992, Access Software & Photo Sampler, Final version 2.0. [CD-ROM, Part No. 15-1132-01], 1991.
- [7] Bryce E. BAYER : Color imaging array. US patent 3,971,065, to Eastman Kodak Co., Patent and Trademark Office, Washington D.C., juillet 1976.

-
- [8] Serge BEUCHER : *Segmentation d'images et morphologie mathématique*. Thèse de doctorat, École Nationale Supérieure des Mines de Paris, juin 1990.
- [9] El-Bay BOURENNANE, Pierre GOUTON, Michel PAINDAVOINE et Frédéric TRUCHE-
TET : Generalization of canny-deriche filter for detection of noisy exponential edge.
Signal Processing, Elsevier, 82(10):1317–1328, 2002.
- [10] Matthias BREIER, Constantin HAAS, Wei LI et Dorit MERHOF : Color filter ar-
rays revisited—evaluation of bayer pattern interpolation for industrial applications.
In Procs. 14th International Conference on Industrial Informatics (INDIN 2016),
pages 52–57, Poitiers, France, juillet 2016.
- [11] John CANNY : A computational approach to edge detection. *IEEE Transactions
on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, novembre 1986.
- [12] Jocelyn CHANUSSOT et Patrick LAMBERT : Total ordering based on space filling
curves for multivalued morphology. *In Procs. 4th International Symposium on Ma-
thematical Morphology and its Applications (ISMM'98)*, pages 51–58, Amsterdam,
The Netherlands, juin 1998.
- [13] Chia-Hsiung CHEN, Sao-Jie CHEN et Pei-Yung HSIAO : Edge detection on the
Bayer pattern. *In Procs. IEEE Asia Pacific Conference on Circuits and Systems
(APCCAS'06)*, pages 1132–1135, Singapore, décembre 2006.
- [14] Xiangdong CHEN, Gwanggil JEON et Jechang JEONG : Voting-based directional
interpolation method and its application to still color image demosaicking. *IEEE
Transactions on Circuits and Systems for Video Technology*, 24(2):255–262, février
2014.
- [15] David R. COK : Signal processing method and apparatus for sampled image si-
gnals. US patent 4,630,307, to Eastman Kodak Co., Patent and Trademark Office,
Washington D.C., décembre 1986.
- [16] Laurent CONDAT : A new random color filter array with good spectral properties. *In
Proceedings of the IEEE International Conference on Image Processing (ICIP'09)*,
pages 1613–1616, Cairo, Egypt, novembre 2009.
- [17] Laurent CONDAT : A simple, fast and efficient approach to denoisaicking : Joint
demosaicking and denoising. *In Procs. 17th IEEE International Conference on
Image Processing (ICIP'10)*, pages 905–908, Hong Kong, China, septembre 2010.

- [18] Laurent CONDAT et Saleh MOSADDEGH : Joint demosaicking and denoising by total variation minimization. *In Procs. 19th IEEE International Conference on Image Processing (ICIP'12)*, pages 2781–2784, Orlando, Florida, USA, septembre 2012.
- [19] Aldo CUMANI : Edge detection in multispectral images. *CVGIP : Graphical Models and Image Processing*, 53(1):40–51, janvier 1991.
- [20] Christine A. CURCIO, Kimberly A. ALLEN, Kenneth R. SLOAN, Connie L. LEREA, James B. HURLEY, Ingrid B. KLOCK et Ann H. MILAM : Distribution and morphology of human cone photoreceptors stained with anti-blue opsin. *Journal of Comparative Neurology*, 4(312):610–624, octobre 1991.
- [21] Aram DANIELYAN, Markku VEHVILAINEN, Alessandro FOI, Vladimir KATKOVNIK et Karen EGIAZARIAN : Cross-color BM3D filtering of noisy raw data. *In Procs. 2009 International Workshop on Local and Non-Local Approximation in Image Processing (LNLA 2009)*, pages 125–129, Tuusula, Finland, août 2009.
- [22] Herbert J. A. DARTNALL, James K. BOWMAKER et John D. MOLLON : Human visual pigments : Microspectrophotometric results from the eyes of seven persons. *Proceedings of the Royal Society of London*, 220(1218):115–130, novembre 1983.
- [23] Rachid DERICHE : Using Canny's criteria to derive a recursively implemented optimal edge detector. *International Journal of Computer Vision*, 1(2):167–187, juin 1987.
- [24] Rachid DERICHE : Fast algorithms for low-level vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1):78–87, janvier 1990.
- [25] Silvano DI ZENZO : A note on the gradient of a multi-image. *Computer Vision, Graphics, and Image Processing*, 33(1):116–125, janvier 1986.
- [26] Éric DUBOIS : Frequency-domain methods for demosaicking of Bayer-sampled color images. *IEEE Signal Processing Letters*, 12(12):847–850, décembre 2005.
- [27] N. L. FERNÁNDEZ-GARCÍA, A. CARMONA-POYATO, R. MEDINA-CARNICER et F. J. MADRID-CUEVAS : Automatic generation of consensus ground truth for the comparison of edge detection techniques. *Image and Vision Computing*, 26(4):496–511, 2008.

- [28] Christophe FIORIO : *Approche interpixel en analyse d'images : une topologie et des algorithmes de segmentation*. Thèse de doctorat, Université Montpellier 2, novembre 1995.
- [29] Bahadır K. GUNTURK, Yucel ALTUNBASAK et Russell M. MERSEREAU : Color plane interpolation using alternating projections. *IEEE Transactions on Image Processing*, 11(9):997–1013, septembre 2002.
- [30] John F. HAMILTON et James E. ADAMS : Adaptive color plan interpolation in single sensor color electronic camera. US patent 5,629,734, to Eastman Kodak Co., Patent and Trademark Office, Washington D.C., mai 1997.
- [31] Thorsten HANSEN et Karl R. GEGENFURTNER : Independence of color and luminance edges in natural scenes. *Visual Neuroscience*, 26:35–49, 2009.
- [32] Robert M. HARALICK, K. SHANMUGAN et Its'hak DINSTEIN : Textural features for image classification. *IEEE Transactions on Systems, Man and Cybernetics*, 3(6):610–621, 1973.
- [33] Robert M. HARALICK, Stanley R. STERNBERG et Xinhua ZHUANG : Image analysis using mathematical morphology. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(4):532–550, juillet 1987.
- [34] Johannes HERWIG et Josef PAULI : Spatial Gaussian filtering of Bayer images with applications to color segmentation. In *FarbBV2009 : Tagungsband; 15. Workshop Farbbildverarbeitung*, pages 19–28, Berlin, Germany, octobre 2009.
- [35] Johannes HERWIG et Josef PAULI : Regularized color demosaicing via luminance approximation. In *Procs. 5th European Conference on Colour in Graphics, Image and Vision*, pages 62–69, Joensuu, Finland, juin 2010.
- [36] Robert H. HIBBARD : Apparatus and method for adaptively interpolating a full color image utilizing luminance gradients. US patent 5,382,976, to Eastman Kodak Co., Patent and Trademark Office, Washington D.C., janvier 1995.
- [37] Keigo HIRAKAWA et Thomas W. PARKS : Joint demosaicing and denoising. *IEEE Transactions on Image Processing*, 15(8):2146–2157, août 2006.

- [38] Takayuki KIJIMA, Hideo NAKAMURA, John T. COMPTON et Jr. JOHN F. HAMILTON : Image sensor with improved light sensitivity. US Patent 7,688,368 (B2), to Eastman Kodak Co., Patent and Trademark Office, Washington D.C., mars 2010.
- [39] Daisuke KIKU, Yusuke MONNO, Sunao KIKUCHI, Masayuki TANAKA et Masatoshi OKUTOMI : Residual interpolation for color image demosaicking. *In Procs. IEEE International Conference on Image Processing (ICIP'13)*, pages 2304–2308, Melbourne, Australia, septembre 2013.
- [40] Daisuke KIKU, Yusuke MONNO, Masayuki TANAKA et Masatoshi OKUTOMI : Beyond color difference : Residual interpolation for color image demosaicking. *IEEE Transactions on Image Processing*, 25(3):1288–1300, mars 2016.
- [41] Chin Chye KOH et Sanjit K MITRA : Compression of Bayer color filter array data. *In Procs. IEEE International Conference on Image Processing (ICIP'03)*, volume 2, pages II–255, Barcelona, Spain, septembre 2003. IEEE.
- [42] Claude A. LAROCHE et Mark A. PRESCOTT : Apparatus and method for adaptively interpolating a full color image utilizing chrominance gradients. US patent 5,373,322, to Eastman Kodak Company, Patent and Trademark Office, Washington D.C., juin 1993.
- [43] Min Seok LEE, Sang Wook PARK et Moon Gi KANG : Denoising algorithm for CFA image sensors considering inter-channel correlation. *Sensors*, 17(6):1236, mai 2017.
- [44] Sang-Yong LEE et Antonio ORTEGA : A novel approach of image compression in digital cameras with a Bayer color filter array. *In Procs. 8th International Conference on Image Processing, (ICIP'01)*, volume 3, pages 482–485, Thessaloniki, Greece, octobre 2001.
- [45] Brian LEUNG, Gwanggil JEON et Éric DUBOIS : Least-squares luma–chroma demultiplexing algorithm for Bayer demosaicking. *IEEE Transactions on Image Processing*, 20(7):1885–1894, juillet 2011.
- [46] Nai-Xiang LIAN, Lanlan CHANG, Yap-Peng TAN et Vitali ZAGORODNOV : Adaptive filtering for color filter array demosaicking. *IEEE Transactions on Image Processing*, 16(10):2515–2525, octobre 2007.

- [47] Olivier LOSSON et Ludovic MACAIRE : CFA local binary patterns for fast illuminant-invariant color texture classification. *Journal of Real-Time Image Processing*, pages 1–15, décembre 2012.
- [48] Olivier LOSSON et Ludovic MACAIRE : Colour texture classification from colour filter array images using various colour spaces. *IET Image Processing*, 6(8):1192–1204, novembre 2012.
- [49] Olivier LOSSON, Alice POREBSKI, Nicolas VANDENBROUCKE et Ludovic MACAIRE : Color texture analysis using CFA chromatic co-occurrence matrices. *Computer Vision and Image Understanding*, 117(7):747–763, juillet 2013.
- [50] Rastislav LUKAC et Konstantinos N. PLATANIOTIS : Color filter arrays : Design and performance analysis. *IEEE Transactions on Consumer Electronics*, 51(4):1260–1267, novembre 2005.
- [51] Rastislav LUKAC, Bogdan SMOLKA, Karl MARTIN, Konstantinos N. PLATANIOTIS et Anastasios N. VENETSANOPOULOS : Vector filtering for color imaging. *IEEE Signal Processing Magazine*, 22(1):74–86, mars 2005.
- [52] Henrique S. MALVAR, Lee-Wei HE et Ross CUTLER : High-quality linear interpolation for demosaicing of Bayer-patterned color images. In *Procs. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'04)*, volume 3, pages 485–488, Montreal, Quebec, Canada, mai 2004.
- [53] David MARR et Ellen HILDRETH : Theory of edge detection. *Proceedings of the Royal Society of London B : Biological Sciences*, 207(1167):187–217, 1980.
- [54] Enrico De MICHELI, Bruno CAPRILE, Pasquale OTTONELLO et Vincent TORRE : Localization and noise in edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(10):1106–1117, octobre 1989.
- [55] Yusuke MONNO, Masayuki TANAKA et Masatoshi OKUTOMI : Multispectral demosaicking using guided filter. In *Procs. SPIE Electronic Imaging Annual Symposium (SPIE'12) : Digital Photography VII*, volume 8299, pages 82990O–82990O–7, Burlingame, California, USA, janvier 2012.
- [56] Philippe MONTESINOS et S. DATTENNY : Sub-pixel accuracy using recursive filtering. In *Procs. 10th Scandinavian Conference on Image Analysis (SCIA'97)*, pages 523–530, Lappeenranta, Finland, juin 1997.

- [57] Dmitriy PALIY, Vladimir KATKOVNIK, Radu BILCU, Sakari ALENIOUS et Karen EGIAZARIAN : Spatially adaptive color filter array interpolation for noiseless and noisy data. *International Journal of Imaging Systems and Technology*, 17(3):105–122, octobre 2007.
- [58] C. PALM : Color texture classification by integrative co-occurrence matrices. *Pattern Recognition*, 37(5):965–976, mai 2004.
- [59] Sung Hee PARK, Hyung Suk KIM, Steven LANSEL, Manu PARMAR et Brian A. WANDELL : A case for denoising before demosaicking color filter array data. *In Procs. 43rd Asilomar Conference on Signals, Systems and Computers*, pages 860–864, Pacific Grove, CA, USA, novembre 2009.
- [60] Soo-Chang PEI et Io-Kuong TAM : Effective color interpolation in CCD color filter arrays using signal correlation. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(6):503–513, juin 2003.
- [61] Ibrahim PEKKUCUKSEN et Yucel ALTUNBASAK : Edge strength filter based color filter array interpolation. *Image Processing, IEEE Transactions on*, 21(1):393–397, Jan 2012.
- [62] Ibrahim PEKKUCUKSEN et Yucel ALTUNBASAK : Multiscale gradients-based color filter array interpolation. *IEEE Transactions on Image Processing*, 22(1):157–165, janvier 2013.
- [63] Harold PHELIPPEAU : *Méthodes et algorithmes de dématricage et de filtrage du bruit pour la photographie numérique*. Thèse de doctorat, Université Paris-Est, avril 2009.
- [64] Harold PHELIPPEAU, Mohamed AKIL, Stefan BARA et Hugues TALBOT : Algorithme de dématricage pour la photographie numérique : Architecture dédiée et implantation sur FPGA. *In XXIIe colloque sur le Traitement du Signal et des Images (GRETSI 2009)*, Dijon, France, septembre 2009.
- [65] Austin ROORDA, Andrew B. METHA, Peter LENNIE et David R. WILLIAMS : Packing arrangement of the three cone classes in primate retina. *Vision Research*, 41:1291–1306, janvier 2001.
- [66] Caroline A. SCHNEIDER, Wayne S. RASBAND et Kevin W. ELICEIRI : NIH image to imagej : 25 years of image analysis. *Nature Methods*, 9:671–675, 2012.

- [67] Jun SHEN et Serge CASTAN : An optimal linear operator for step edge detection. *Computer Vision, Graphics, and Image Processing*, 54(2):112–133, 1992.
- [68] Alain TRÉMEAU, Christine FERNANDEZ-MALOIGNE et Pierre BONTON : *Image numérique couleur - De l'acquisition au traitement*. Dunod, Paris, France, 2004.
- [69] Nicolas VANDENBROUCKE : *Segmentation d'images couleur par classification de pixels dans des espaces d'attributs colorimétriques adaptés. Application à l'analyse d'images de football*. Thèse de doctorat, Université Lille 1 – Sciences et Technologies, décembre 2000.
- [70] Patrick S.-P. WANG : *Pattern recognition, machine intelligence and biometrics*. Numéro 173. Springer-Verlag, Berlin, Heidelberg, 2011.
- [71] Xiang XIE, GuoLin LI, XiaoWen LI, XinKai CHEN, Kun YANG, Chun ZHANG et Zhihua WANG : A novel method of lossy image compression for digital image sensors with Bayer color filter arrays. In *Procs. IEEE International Symposium on Circuits and Systems (ISCAS 2005)*, pages 4995–4998, Kobe, Japan, mai 2005.
- [72] Yanqin YANG : *Contribution à l'évaluation objective de la qualité d'images couleur estimées par dématricage*. Thèse de doctorat, Université Lille 1, Sciences et Technologies, octobre 2009.
- [73] Wei YE et Kai-Kuang MA : Color image demosaicing using iterative residual interpolation. *IEEE Transactions on Image Processing*, 24(12):5879–5891, décembre 2015.
- [74] Fan ZHANG, Xiaolin WU, Xiaokang YANG, Wenjun ZHANG et Lei ZHANG : Robust color demosaicking with adaptation to varying spectral correlations. *IEEE Transactions on Image Processing*, 18(12):2706–2717, décembre 2009.
- [75] Lei ZHANG, Rastislav LUKAC, Xiaolin WU et David ZHANG : PCA-based spatially adaptive denoising of CFA images for single-sensor digital cameras. *IEEE Transactions on Image Processing*, 18(4):797–812, avril 2009.
- [76] Lei ZHANG, Xiaolin WU et David ZHANG : Color reproduction from noisy CFA data of single sensor digital cameras. *IEEE Transactions on Image Processing*, 16(9):2184–2197, septembre 2007.

-
- [77] Dengwen ZHOU, Xiaoliu SHEN et Weiming DONG : Colour demosaicking with directional filtering and weighting. *IET Image Processing*, 6(8):1084–1092, 2012.

Table des figures

| | | |
|------|---------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 1.1 | Couleurs primaires, secondaires et complémentaires. | 16 |
| 1.2 | Schéma simplifié de la technologie tri-capteur. | 17 |
| 1.3 | Schéma simplifié de la technologie mono-capteur. | 18 |
| 1.4 | Différentes configurations du CFA. | 19 |
| 1.5 | Fonction d'efficacité lumineuse normalisée en vision photopique et sensibilité spectrale relative. | 20 |
| 1.6 | Motifs locaux dans une image CFA. | 21 |
| 1.7 | Image CFA de Bayer et sous-ensembles de pixels par composante couleur. | 22 |
| 1.8 | Acquisition d'une image couleur selon le type de caméra | 25 |
| 1.9 | <i>Principe du dématricage.</i> | 25 |
| 2.1 | Voisinage d'un pixel P dans une image CFA. | 29 |
| 2.2 | Comparaison des méthodes RI et DCC | 32 |
| 2.3 | Cartes des différences de composantes horizontale et verticale [62]. | 35 |
| 2.4 | Correction des erreurs de choix des directions d'interpolations par la méthode de LMDC | 37 |
| 2.5 | Répartition de l'énergie d'une image CFA dans le plan de Fourier. | 39 |
| 2.6 | Voisinages utilisés pour la classification des textures dans I^{CFA} | 43 |
| 2.7 | Compression/décompression des images couleur et CFA | 44 |
| 2.8 | Calcul des composantes YC_bC_r pour un bloc 2×2 de l'image CFA. | 45 |
| 2.9 | Séparation des deux sous-composantes Y pour la compression JPEG. | 46 |
| 2.10 | Rotation de la composante de luminance Y de 45° (Lee et Ortega [44]). | 46 |
| 2.11 | Voisinages utilisés par Chen <i>et al.</i> [13] pour le lissage gaussien de l'image CFA (avec coefficients du filtre pour $\sigma = 1$). | 48 |
| 2.12 | Voisinages utilisés par le filtre laplacien de Chen <i>et al.</i> [13] appliqué à l'image CFA lissée. | 48 |
| 2.13 | Détection des contours sur une image CFA comme proposé par Chen <i>et al.</i> [13]. | 49 |

| | | |
|-----|----------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 3.1 | Calcul de la dérivée partielle selon x par différentes méthodes vectorielles. | 57 |
| 3.3 | Décalage provoqué par un filtre de lissage décalé verticalement et un filtre de dérivation décalé horizontalement sur une image I avec $d = 1$. | 64 |
| 3.4 | Décalage provoqué par un lissage décalé verticalement et une dérivation décalée horizontalement dans une image I^{CFA} avec $d = 2$. | 65 |
| 3.5 | Calcul des trois composantes de la dérivée partielle selon x à partir de I^{CFA} par la méthode SDD. | 66 |
| 3.6 | Calcul des dérivées partielles de l'image I^{CFA} en utilisant les filtres de Deriche classiques et les filtres de Deriche décalés. | 68 |
| 3.7 | Calcul des composantes lissées et dérivation selon x par la méthode SDS. | 70 |
| 3.8 | Calcul des composantes lissées décalées de l'image I^{CFA} . | 71 |
| 3.9 | Calcul de la dérivée partielle selon x par les méthodes scalaires. | 72 |
| 4.1 | Synthetic test image. | 79 |
| 4.2 | Détection des contours et évaluation de leur qualité selon les différentes méthodes. | 82 |
| 4.3 | Évolution du FoM en fonction du paramètre α . | 86 |
| 4.4 | Résultats de détection des contours à partir de l'image CFA bruitée. | 89 |
| 4.5 | Evolution de FoM en fonction du niveau du bruit gaussien et la hauteur des transition couleur. | 90 |
| 4.6 | Évolution du FoM en fonction du paramètre α de Deriche dans les images isoluminantes. | 92 |
| 4.7 | Résultats de détection des contours à partir de l'image CFA iso-luminante bruitée. | 94 |
| 4.8 | Performance des différentes méthodes de détection des contours par rapport à leurs complexité algorithmique. | 97 |

Table des tableaux

| | | |
|-----|---------------------------------------------------------------------------------------------------------------------------|----|
| 4.1 | Corrélation spectrale inter-composantes dans nos images synthétiques en fonction de l'écart-type du bruit ajouté. | 81 |
| 4.2 | Performance des différentes méthodes de détection des contours sur notre base d'images synthétiques. | 87 |
| 4.3 | Performance des différentes méthodes de détection des contours sur des images synthétiques iso-luminantes. | 93 |
| 4.4 | Nombre d'opérations élémentaires par pixel nécessaire pour estimer les dérivées partielles. | 95 |