

# Algorithme de Dématriçage pour la Photographie Numérique : Architecture Dédiée et Implantation sur FPGA

HAROLD PHELIPPEAU<sup>1</sup>, MOHAMED AKIL<sup>1</sup>, HUGUES TALBOT<sup>1</sup>, STEFAN BARA<sup>2</sup>

<sup>1</sup> Université Paris-Est, Laboratoire d'Informatique Gaspard-Monge, Equipe A3SI, ESIEE Paris  
Cité Descartes, 2 boulevard Blaise Pascal, BP 99, 93162 Noisy Le Grand cedex, France

<sup>2</sup> NXP Semiconductors

2 Esplanade Anton Philips, Campus Effiscience, Colombelles BP 2000 14906, Caen Cedex 9, France

[phelipp@esiee.fr](mailto:phelipp@esiee.fr), [akilm@esiee.fr](mailto:akilm@esiee.fr), [talboth@esiee.fr](mailto:talboth@esiee.fr)  
[stefan.bara@nxp.com](mailto:stefan.bara@nxp.com)

**Résumé** – Dans cet article, nous proposons un nouvel algorithme de dématriçage fonctionnant par interpolations directionnelles, dédié à l'arrangement des couleurs proposé par Bayer. L'algorithme proposé associe une qualité d'image possédant peu ou pas d'artefacts avec une faible complexité algorithmique. Nous présentons conjointement, une architecture matérielle dédiée qui l'implémente et les résultats de sa synthèse FPGA.

**Abstract** – In this Paper, we propose a novel edge directed demosaicing algorithm dedicated for the Bayer color arrangement. The proposed algorithm associates an excellent image quality and has a weak computational complexity. We present here a dedicated architecture implementing this algorithm and the results corresponding to its FPGA synthesis.

## 1 Introduction

Pour des raisons de coût et d'espace, les appareils photos/vidéos numériques sont équipés d'un capteur unique. L'information de couleur est introduite par la superposition de filtres colorés à la surface des photorécepteurs (voir figure 1). Cela permet le multiplexage spatial des trois canaux de couleurs primaires rouge, vert et bleu. C'est généralement la mosaïque de filtres proposée par Bayer [1] qui est utilisée. Une étape de reconstruction ou dématriçage est nécessaire pour produire une image en couleur. Cette étape a une grande importance concernant la qualité des images produites, elle est susceptible d'introduire de nombreux artefacts tels que des fausses couleurs, des effets de moiré, du flou, des effets de grille et l'introduction de structures en forme de labyrinthes. Ce travail s'intéresse tout d'abord à mettre au point un nouvel algorithme de dématriçage n'introduisant pas ou peu d'artefacts. La méthode que nous proposons doit prendre en compte, d'une part les contraintes de temps imposées pour les applications de la photographie et de la vidéo numérique et d'autre part, les limitations de l'architecture système des plates formes portables (appareils photo numérique, téléphonie mobile). Dans cet article nous présentons l'algorithme de dématriçage proposé ainsi que l'étude et la validation de l'architecture matérielle dédiée qui l'implémente.

## 2 Présentation de l'algorithme GEDI (Green Edge Directed Interpolation)

Nous proposons un nouvel algorithme de dématriçage que nous appelons GEDI (pour Green Edge directed Interpolation) (voir [7, 8]). Cet algorithme utilise un nouvel estimateur de directions des détails de l'image appelé GED (pour Green Edge Directions). Cet estimateur est basé sur la comparaison des gradients du canal vert interpolé dans les directions verticales et horizontales. Considérons  $G_v(\cdot)$  le canal vert interpolé verticalement et  $G_h(\cdot)$  le canal vert interpolé horizontalement. Les interpolations sont calculées en utilisant les termes de corrections des gradients de second ordre des canaux bleu et rouge, comme le propose Hamilton dans [10]. Considérons les

coordonnées  $(i, j) \in X$  d'un pixel de l'image, où  $X$  est un

ensemble de positions 2-D et  $G(\cdot)$ , le canal vert interpolé. L'estimateur GED est présenté dans

l'équation (2), où  $\Delta hG(i, j)$  et  $\Delta vG(i, j)$  sont

respectivement définis dans les équations (3) et (4).

$$G(i, j) = \begin{cases} G_h(i, j) & \text{if } [\nabla hG_h(i, j) + \nabla hG_v(i, j)] \\ & \leq [\nabla vG_h(i, j) + \nabla vG_v(i, j)] \\ G_v(i, j) & \text{sinon} \end{cases} \quad (2)$$

$$\nabla hG(i, j) = |G(i-1, j) - G(i, j)| + |G(i+1, j) - G(i, j)| \quad (3)$$

$$\nabla vG(i, j) = |G(i, j-1) - G(i, j)| + |G(i, j+1) - G(i, j)| \quad (4)$$

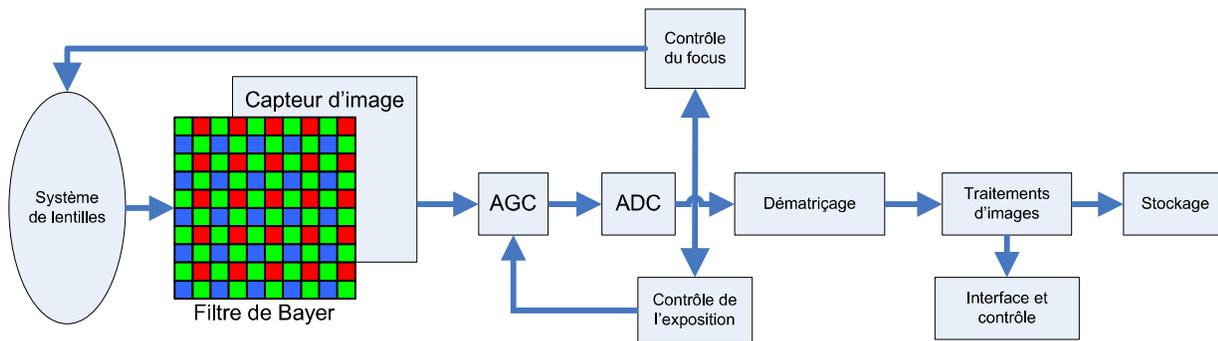


Figure 1: Schéma général d'un appareil photo numérique, il est constitué d'un système de lentilles, d'un capteur d'image numérique, d'un contrôle automatique du gain (AGC), d'un contrôle automatique du focus, d'une conversion analogique numérique (ADC), ainsi que d'une chaîne de traitement d'image, laquelle peut être implantée sur architecture mixte de type DPS/circuit dédié.

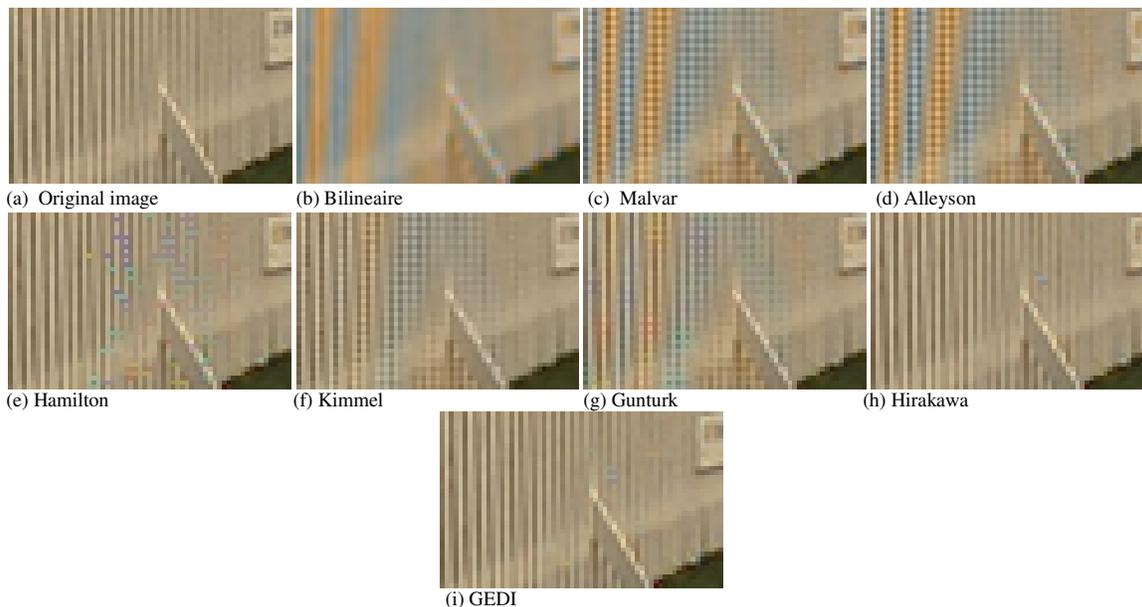


Figure 2: Comparaison de la qualité visuelle des différents algorithmes étudiés sur un motif de bandes verticales.

Une homogénéisation des choix de directions des objets par la méthode de LMDC (Local Majority Direction Choice) (voir [6]) est ensuite appliquée. Une fois que le canal vert est interpolé, les canaux rouge et bleu sont reconstruits par la méthode de constante des teintes proposée par Cock dans [5]. Finalement, une étape de réduction des artefacts de couleurs est appliquée comme le propose Cock dans [4]. Pour résumer, l'algorithme GEDI s'applique en six étapes, (1) Interpoler le plan vert verticalement et horizontalement ; (2) Estimer les directions d'interpolations : verticale ou horizontale avec GED ; (3) Corriger les erreurs d'estimation de directions des interpolations par la méthode de LMDC ; (4) Interpoler les plans verts avec les directions d'interpolations choisies ; (5) Interpoler les plans rouge et bleu par la méthode de constance des teintes ; (6) Réduire les artefacts de couleur. Nous avons comparé la qualité des images produites par l'algorithme GEDI avec les qualités des images produites par les principaux algorithmes de la littérature: bilinéaire, Malvar, Alleyson, Hamilton, Hirakawa, Kimmel et Gunturk [9, 3, 10, 11, 12, 2]. La qualité des images produites est mesurée en utilisant l'appréciation visuelle sur des

motifs problématiques (voir figure 2) et le calcul du MSE dans les régions de contours et dans les régions lisses, comme le propose Li dans [26], on utilise une bibliothèque de 24 images de référence (voir tableau 1). La comparaison montre, que l'algorithme GEDI permet de produire une très bonne qualité d'image. De plus, en regardant le tableau 2, on voit que GEDI possède une complexité de calculs très inférieure aux algorithmes de sa catégorie (en termes de qualité d'image), par exemple l'algorithme d'Hirakawa.

Tab 1 : Calculs de l'erreur quadratique moyenne dans les régions de contours et dans les régions lisses sur 24 images de référence

Algorithmes	Régions de contours	Régions lisses	Image entière
Bilinear 3x3	168,06	23,07	47,86
Malvar	56,70	8,23	17,97
Alleyson	58,34	8,43	19,30
Kimmel	48,48	8,56	16,37
Hibbard	44,66	7,55	14,74
Hamilton	38,21	6,95	13,06
Hirakawa	31,53	5,90	11,34
GEDI	32,91	5,73	11,30

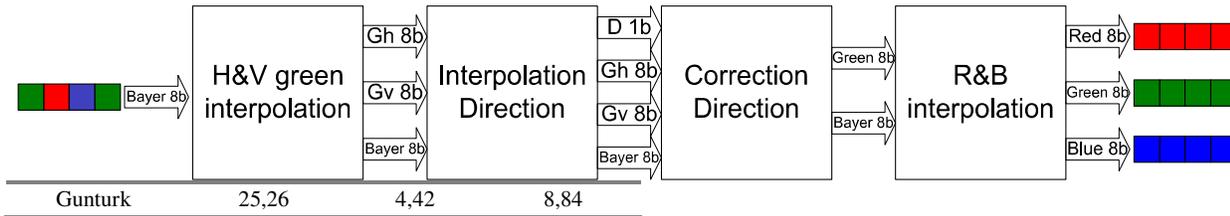


Figure 3 Architecture fonctionnelle de l'algorithme GEDI

Tab 2 : Complexité de calcul des différents algorithmes étudiés

	Multipli- cations	Multip- lications fixes	Addit- ions	Comp- arai- sons	Valeurs absolues
Bilinear	0	4	3	0	0
Malvar	0	19	21	0	0
Alleyson	0	61	70	0	0
Kimmel	0	120	180	0	0
Hibbard	0	6	12	1	2
Hamilton	0	13/2	14	1	4
Hirakawa	24	24	88	71	12
GEDI	0	4	28	1	8
Gunturk	0	480	480	0	0

### 3 Architecture dédiée pour l'algorithme GEDI

L'architecture proposée est composée de quatre blocs. Chacun de ces blocs correspond à une étape de l'algorithme. Le bloc «*H&V green interpolation*», calcule les interpolations verticales et horizontales du canal vert. Il reçoit en entrée le flux de *Bayer*, qui correspond au flux de pixels issus du capteur et délivre trois flux de pixels: le flux de *Bayer*, le flux  $G_H$ , qui correspond aux pixels verts interpolés horizontalement et le flux  $G_V$ , qui correspond aux pixels verts interpolés verticalement. Le bloc «*Interpolation direction*», estime la direction des détails dans l'image. Il reçoit les trois flux de pixels sortant du bloc «*H&V green interpolation*» et délivre ses trois flux entrant avec un flux de direction  $D$ . La fonction «*Correction direction*», implémente la méthode de correction des directions par LMDC et fournit en sortie le flux *Bayer* et le flux des pixels verts interpolés *Green*. La fonction «*R&B interpolation*», interpole les canaux rouge et bleu par la méthode de constance des teintes. Elle délivre les trois flux de pixels rouge, vert et bleu. Pour chacune de ces fonctions, on présente l'étude de la complexité en termes de latence ( $L$  en cycles), d'encombrement mémoire (RAM en bits) et de nombre de bascules Flip-Flop utilisées (appelés éléments séquentiels en bits dans le tableau Tab 3). Cette complexité est donnée en fonction de la largeur  $M$  de l'image à traiter.

Prenons par exemple la fonction «*H&V green interpolation*», celle-ci se décompose de deux blocs : le

bloc de mise en forme du masque de traitement ( $5 \times 5$ ) à

partir du flux de *Bayer* issue du capteur, qui est illustré sur la figure 4 et le bloc d'interpolations horizontales et

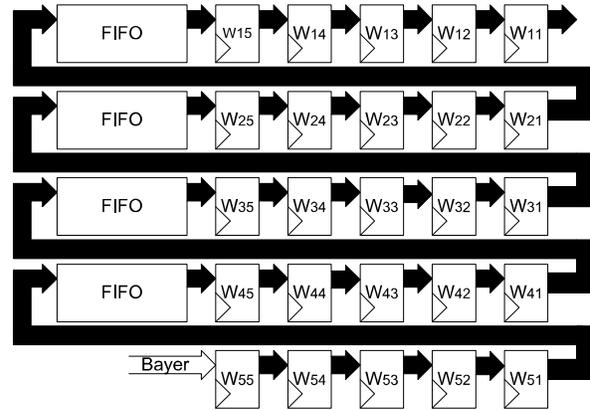


Figure 4 : Bloc de construction du masque de traitement: les pixels issus du capteur sont accumulés jusqu'à la dernière position du masque. Ils se décalent à chaque coup d'horloge vers la position suivante à travers les FIFO et les registres, jusqu'au remplissage total des registres. Une fois que ceux-ci sont pleins, le masque est déplacé vers le pixel suivant à chaque coup d'horloge.

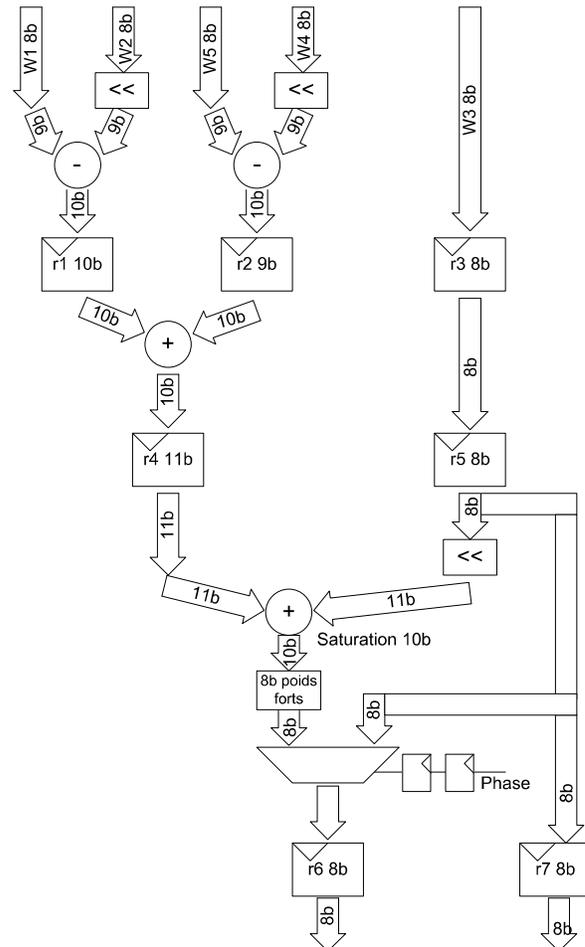


Figure 5: Bloc «*H&V computation*»: Architecture de calcul de l'interpolation des pixels verts manquants. Ce bloc possède une latence de 3 cycles.

verticale du canal vert, qui est illustré sur la figure 5. Le temps de latence pour la mise en œuvre du masque de

traitement de taille (5×5) (cas où  $n = 5$ ) est de :  $[M \times (n-$

$l) + n]$  cycles, où  $M$ , est la largeur de l'image et  $n$  est la taille du masque. La latence totale de cette fonction est  $([M \times (n-1) + n] + 3)$  cycles, 3 étant le nombre de cycles utilisés par la fonction «*H&V computation*», où chaque cycle correspond à une étape de calcul de la fonction.

**Tab 3 : Analyse de la complexité en termes de : latence, mémoire RAM et bascules FF pour les différentes fonctions de l'architecture**

Fonctions	Latences (cycles)	RAM (bits)	Éléments séquentiels
<i>H&amp;V</i>	$4M+8$	$16(M-5)$	330
<i>Interpolation</i>			
<i>Interpolation direction</i>	$2M+7$	$48(M-3)$	444
<i>Correction direction</i>	$2M+4$	$41(M-3)$	266
<i>R&amp;B</i>			
<i>interpolation</i>	$2M+5$	$24(M-3)$	218
<i>Total</i>	$10M+24$	$16M-562$	1258

#### 4 Résultats de l'implantation de l'architecture sur FPGA

Nous avons modélisé, optimisé et simulé sous l'environnement logiciel Synplify, l'architecture dédiée pour l'algorithme de dématricage GEDI. Nous avons synthétisé l'architecture pour deux cibles de FPGA : Xilinx Virtex 5 (XC5VLX30FF324-2) et Altera Stratix IV (EP4SGX70B). Les résultats de cette synthèse sont donnés respectivement par les tableaux 4 et 5. Les pourcentages des ressources occupées sont : 13% de registres et 18 % de LUT dans le cas d'un Xilinx Virtex 5 et 11% de blocs mémoire (M9K) dans le cas d'un Altera Statix IV.

**Tab 4: Résultats de la synthèse Xilinx Virtex 5 – cible XCVLX30FF324-2**

Fréquence estimée en MHz	237,2 MHz
Période estimée en ns	4.217 ns
Nombre total de registres en bits	2510 bits
Nombre de DUAL Port Ram (RAM32X1D)	288
Nombre total de LUTs	3580

**Tab 5: Résultats de la synthèse Altera Stratix IV – cible EP4SGX70B**

Fréquence estimée en MHz	176,6 MHz
Période estimée en ns	5.664 ns
Nombre total de registres en bits	3942 bits
M9Ks (Altesyncram)	54
Nombre total de LUTs	4740

#### Références

- [1]B.E.Bayer. *Color Imaging Array*. United State Patent, 3,971,065, 1976.
- [2]B.K.Gunturk, Y.Altunbasak, and R.M.Mesereau. Color plane interpolation using alternating projections. *IEEE transactions on image processing*, 11:997–1013, 2002.
- [3]D.Alleyson, S.Süstrunk, and J.Herault. Color demosaicing by estimating luminance and opponent chromatic signals in the Fourier domain. *Proc. Color Imaging Conf: Color Science, Systems, Applications*, pages 331–336, 2002.
- [4]D.R.Cock. Single-chip electronic color camera with color dependent birefringent optical spatial frequency filter and red and blue signal signal interpolating circuit. *United States Patent number 4,605,956*, 1986.
- [5]D.R.Cock. Reconstruction of CCD images using template matching. *IS&T's 47th Annual Conference / ICPS*, pages 380–385, 1994.
- [6]H.Phelippeau, H.Talbot, S.Bara, and M.Akil. Local majoritary direction choice enhancement mehod for edge directed demosaicing. *STMICROELECTRONICS*, 2008.
- [7]H.Phelippeau, S.Bara, M.Akil, and H.Talbot. Green edge directed interpolation. *STMICROELECTRONICS*, 2008.
- [8]M.Akil S.Bara H.Phelippeau, H.Talbot. Green edge directed demosaicing algorithm igm2009-06.
- [9]H.S.Malvar, L.W. He, and Y.Yacobi. High-quality linear interpolation for demosaicing of bayer-patterned color images. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2004.
- [10]J.F.Hamilton and J.E.Adams. *Adaptive color plan interpolation in single sensor color electronic camera*. US Patent 5,629,734 to Eastman Kodak Compagny, Patent and Trademark office, Washington, 1997.
- [11]K.Hirakawa and T.Parks. Adaptive homogeneity-directed demosaicing algorithm. *Image Processing, IEEE Transactions on*, 14:360–369, 2005.
- [12]R. Kimmel. Demosaicing: image reconstruction from color ccd samples. *Image Processing, IEEE Transactions on*, 8(9):1221–1228, 1999.