Dématriçage Robuste aux Filtres et à leurs Arrangements via un Algorithme Déroulé

Matthieu MULLER[1] Daniele PICONE[1] Mauro DALLA MURA[1, 2] Magnus O. ULFARSSON[3]

[1]Université Grenoble Alpes, CNRS, Grenoble INP, GIPSA-lab, 38000 Grenoble, France

[2]Institut Universitaire de France (IUF), France

[3]Université d'Islande, Faculté d'ingénierie électrique et informatique, 101 Reykjavík, Islande

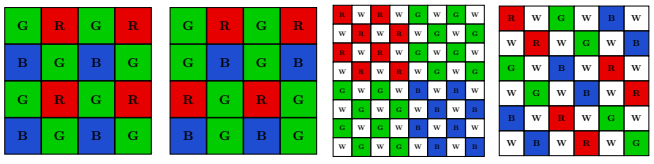
Résumé – Pour obtenir des images en couleurs, la plupart des appareils photo commerciaux s'appuient sur des matrices de filtres spectraux (MFSs), constituées d'un motif de filtres placés sur le plan focal du capteur. Le dématriçage consiste à reconstruire une image RVB (rouge, vert, bleu) complète à partir des pixels du capteur. La majorité des méthodes de dématriçage est conçue pour une MFS spécifique et n'est pas adaptable à d'autres MFSs. Cet article propose un algorithme de dématriçage générique, tant pour le choix des filtres que pour leur arrangement. Elle repose sur le déroulement d'un algorithme itératif secondé d'un réseau de neurones entraîné avec une nouvelle fonction de coût employant des transformations géométriques des MFSs. Des tests préliminaires sur diverses MFSs et filtres spectraux montrent que cette approche se révèle compétitive et rivalise avec les méthodes spécifiques.

Abstract – To obtain color images, most commercial cameras rely on Color Filter Arrays (CFAs), consisting of a pattern of filters placed on the sensor's focal plane. Demosaicking consists in the reconstruction of an RGB image from the sensor's pixels. Most demosaicking methods are designed for a specific CFA and cannot be adapted to other CFAs. This article proposes a generic demosaicking algorithm, both in terms of the choice of filters and their arrangement. It is based on algorithm unrolling assisted by a neural network trained with a new loss function employing geometric transformations of the CFAs. Preliminary tests on various CFAs and spectral filters show that this approach proves competitive and rivals methods tailored on specific CFAs.

1 Introduction

La plupart des appareils photo commerciaux et téléphones doivent être capables d'acquérir des images RVB (rouge, vert, bleu) dans un espace contraint. Une pratique courante est d'utiliser un petit capteur en nuances de gris, couplé à la technologie des matrices de filtres spectraux (MFS). La MFS définit un motif de filtres spectraux qui est superposé aux pixels du capteur. Comme chaque pixel ne contient l'information que pour une seule bande, un processus de dématriçage est nécessaire pour reconstituer les bandes manquantes.

Chaque constructeur utilise des filtres spectraux et une MFS qui leur sont propres, ce qui complique la reconstruction de l'image RVB et rend les algorithmes de dématriçage dépendants de la MFS et des filtres utilisés. Le motif de Bayer (Figure 1a), le plus répandu, utilise des filtres RVB. De plus, une famille plus récente de MFSs ajoute un filtre panchromatique RVB+P (e.g. Figure 1c) qui améliore le rapport signal sur bruit (SNR), en particulier dans des conditions de faible luminosité.



(a) Bayer (b) Lukac (c) Hamilton (d) Kaizu

FIGURE 1 : Quelques exemples de MFSs, RVB et RVB+P.

Cette grande variété de configurations rend le dématriçage d'images acquises par différentes MFSs difficile, car la plupart des méthodes sont adaptées à une MFS spécifique. Il existe de nombreuses catégories de techniques de dématriçage. Par exemple, l'interpolation classique et ses variantes [6]. Ces méthodes devraient être adaptées à chaque MFS et manquent donc d'invariance vis-à-vis du choix de la MFS sur laquelle travailler.

Les méthodes basées sur des modèles s'appuient sur un modèle direct représentant l'opération de la MFS [9] et sur un terme de régularisation, encodant des connaissances sur l'image à reconstruire. Ces méthodes sont invariantes par rapport à la MFS car elles prennent en compte la MFS pendant la résolution du problème, mais dépendent fortement de la qualité du modèle direct et la régularisation, ce qui peut constituer un facteur limitant.

Les méthodes d'apprentissage profond ont récemment émergé [5]. Elles offrent d'excellentes performances grâce à un entraînement de bout en bout, avec des paires d'images de référence et leurs observations associées. Cependant, ce même entraînement engendre une spécialisation sur la MFS sur laquelle le réseau est entraîné et peut nécessiter de vastes quantités de données. En pratique, ces méthodes ne sont pas applicables au dématriçage d'images acquises avec une MFS différente de celle considérée par l'entrainement et déploient des architectures trop grandes pour être ré-entraînées sur chaque nouvelle MFS.

Enfin, un domaine de recherche en expansion vise à fusionner les méthodes basées sur des modèles avec celles pilotées par les données. Le Deep Image Prior (DIP) [14] utilise une architecture de réseau de neurones comme régularisation. Cette méthode peut être considérée comme non supervisée et fonc-

tionne comme un algorithme itératif, ce qui procure une bonne invariance par rapport au choix de la MFS. Néanmoins, ces méthodes nécessitent un modèle direct et sont très coûteuses en calcul et leurs performances dépendent du choix de l'architecture.

Le déroulement d'algorithme [8] est une autre possibilité qui s'est avérée efficace en traitement d'image. Il s'appuie sur un algorithme itératif basé sur un modèle et apprend une régularisation à partir d'un ensemble de données tout en utilisant un modèle direct. Ces méthodes devraient conserver une grande robustesse aux changements de MFS tout en bénéficiant de l'entraînement de la régularisation. Cependant, l'expérience montre que les algorithmes déroulés tendent à surapprendre l'ensemble de données utilisé pour l'entraînement, et la régularisation apprise devient dépendante du type de modèle direct utilisé, perdant ainsi l'avantage de l'invariance fourni par les approches conventionnelles basées sur des modèles.

Par conséquent, le présent travail propose une méthode de dématriçage basée sur le déroulement d'algorithme, robuste par rapport au choix de la MFS et des filtres utilisés. Cet article présente les contributions suivantes :

- Une méthode basée sur un algorithme déroulé spécialisée, faisant appel à une propriété du dématriçage pour l'inversion matricielle pour réduire le coup de calcul.
- Une nouvelle fonction d'entraînement favorisant l'apprentissage d'un réseau de neurones robuste au choix de MFSs et des filtres spectraux composant les MFSs.

2 Méthode proposée

2.1 Formulation du problème

Soit $\boldsymbol{x} \in \mathbb{R}^{3n}$ une image RVB vectorisée. L'acquisition de l'observation $\boldsymbol{y} \in \mathbb{R}^n$ suppose généralement un modèle linéaire [10]. Le processus prend l'image RVB et produit l'observation comme suit :

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}, \tag{1}$$

où $\boldsymbol{A} \in \mathbb{R}^{n\times 3n}$ est le modèle direct qui encode une MFS. $\boldsymbol{A}$ échantillonne l'observation $\boldsymbol{y}$ à partir de l'image RVB $\boldsymbol{x}$. L'objectif est de reconstruire $\boldsymbol{x}$ en utilisant $\boldsymbol{y}$ et la connaissance de $\boldsymbol{A}$.

Le problème est abordé en minimisant une fonction de coût convexe qui est la somme de deux termes. Le premier est le terme de fidélité aux données $F(\boldsymbol{x}) = \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|^2$, garantissant la cohérence avec l'observation. Le second terme est un régularisateur $R : \mathbb{R}^{3n} \to \mathbb{R}$ qui encode des contraintes sur la reconstruction :

$$\hat{\boldsymbol{x}} \in \arg\min_{\boldsymbol{x}} \; F(\boldsymbol{x}) + R(\boldsymbol{x}). \tag{2}$$

Le choix arbitraire de R est une limitation des approches basées sur des modèles, car il est difficile de trouver le régularisateur optimal pour un couple modèle direct / observation donné. Néanmoins, certains régularisateurs sont largement utilisés, comme la Variation Totale (TV) qui donne de bons résultats dans un large éventail de tâches de traitement d'image.

2.2 Algorithme déroulé

Pour éviter de définir un régularisateur, nous utilisons un algorithme déroulé [8] basé sur ADMM (méthode des multiplicateurs alternés) [4]. Celle-ci consiste à itérer K fois les trois étapes suivantes :

$$\begin{cases} \boldsymbol{x}^{k+1} & = \arg\min_{\boldsymbol{w}}(F(\boldsymbol{w}) + \frac{\rho}{2}\|\boldsymbol{w} - \boldsymbol{z}^k + \boldsymbol{u}^k\|^2) \\ \boldsymbol{z}^{k+1} & = \mathcal{N}_{\boldsymbol{\theta}_k}(\boldsymbol{x}^{k+1} + \boldsymbol{u}^k) \\ \boldsymbol{u}^{k+1} & = \boldsymbol{u}^k + \boldsymbol{x}^{k+1} - \boldsymbol{z}^{k+1} \end{cases}, \tag{3}$$

où $k \in \mathbb{N}$ est l'indice d'itération. $\mathcal{N}_{\boldsymbol{\theta}_k}$ est un réseau de neurones dont l'architecture reste identique à chaque itération, mais dont les poids $\boldsymbol{\theta}_k$ peuvent être différents. $\rho, \tau \in \mathbb{R}_*^+$ sont appris avec les $\mathcal{N}_{\boldsymbol{\theta}_k}$.

Cependant, le réseau tend à perdre sa propriété d'invariance et à surapprendre sur les MFSs vues durant l'entraînement.

Mise à jour de $\boldsymbol{x}_k$ Le calcul de $\boldsymbol{x}_k$ est un sous-problème de minimisation qui possède une solution analytique :

$$\boldsymbol{x}_{k+1} = (\rho_k \boldsymbol{I} + \boldsymbol{A}^T\boldsymbol{A})^{-1}(\rho_k(\boldsymbol{z}_k - \boldsymbol{u}_k) + \boldsymbol{A}^T\boldsymbol{y}), \tag{4}$$

où $\boldsymbol{I}$ est la matrice identité, $\rho_k \in \mathbb{R}_*^+$ est appris et $\boldsymbol{A}$ est l'opérateur de dégradation présenté dans (1). Le calcul de $(\rho_k\boldsymbol{I} + \boldsymbol{A}^T\boldsymbol{A})^{-1}$ est infaisable sur un ordinateur, car la matrice est trop grande. Cependant, dans le cas du dématriçage, la matrice $(\boldsymbol{I} + \frac{1}{\rho_k}\boldsymbol{A}\boldsymbol{A}^T)$ est diagonale, ce qui permet d'utiliser la formule de Sherman-Morrison-Woodbury [2] :

$$(\rho_k\boldsymbol{I} + \boldsymbol{A}^T\boldsymbol{A})^{-1} = \frac{1}{\rho_k}(\boldsymbol{I} - \boldsymbol{A}^T(\boldsymbol{I} + \frac{1}{\rho_k}\boldsymbol{A}\boldsymbol{A}^T)^{-1}\boldsymbol{A}). \tag{5}$$

Même si la matrice $(\boldsymbol{I} + \frac{1}{\rho_k}\boldsymbol{A}\boldsymbol{A}^T)$ est encore grande pour être explicitement représentée, son inversion est simple puisqu'il s'agit uniquement de l'inversion élément par élément de ses éléments diagonaux. La combinaison de (4) et (5) donne la forme finale du bloc de données.

Mise à jour de $\boldsymbol{z}_k$ Pour limiter le nombre de poids appris, nous implémentons $\mathcal{N}_{\boldsymbol{\theta}_k}$ comme étant une version réduite du réseau U-Net [11]. Il s'agit d'un réseau de neurones basé sur des convolutions dont les noyaux sont appris. Ainsi, la mise à jour de $\boldsymbol{z}_k$ est la sortie de ce réseau de neurones avec les poids correspondants à l'itération k :

$$\boldsymbol{z}_{k+1} = \mathcal{N}_{\boldsymbol{\theta}_k}(\boldsymbol{x}_{k+1} + \boldsymbol{u}_k).$$

Il est important de noter que même si l'architecture est la même pour toutes les itérations, les poids $\boldsymbol{\theta}_k$ sont différents. Forcer les poids à être identiques à travers les itérations réduirait considérablement le pouvoir expressif du réseau.

Mise à jour de $\boldsymbol{u}_k$ Le calcul de $\boldsymbol{u}_k$ consiste en une combinaison linéaire entre $\boldsymbol{x}_{k+1}$, $\boldsymbol{z}_{k+1}$ et $\boldsymbol{u}_k$, où η_k est appris :

$$\boldsymbol{u}_{k+1} = \boldsymbol{u}_k + \eta_k(\boldsymbol{x}_{k+1} - \boldsymbol{z}_{k+1}).$$

Cette variable permet de suivre les résidus entre $\boldsymbol{x}_{k+1}$ et $\boldsymbol{z}_{k+1}$ et aide les itérations suivantes à affiner les autres variables.

Nombre de poids La dernière itération ne comporte que la mise à jour de $\boldsymbol{x}_k$ puisque les deux autres étapes ne sont utiles qu'à l'itération suivante. Ainsi, le nombre total de poids apprenables est donné par :

$$|\boldsymbol{\theta}| = (K-1)(2+|\boldsymbol{\theta}_k|)+1, \tag{6}$$

où K est le nombre d'itérations et $|\boldsymbol{\theta}_k|$ le nombre de poids dans la mise à jour de $\boldsymbol{z}_k$ à l'itération k.

2.3 Fonction d'entraînement robuste à la MFS

Les algorithmes déroulés sont susceptibles de surapprendre sur la MFS sur laquelle ils ont été entraînés, limitant leur robustesse lorsqu'ils sont appliqués à des images acquises avec d'autres MFSs. En effet, chaque nouvelle MFS constitue un problème en soi, ce qui explique la spécialisation des méthodes de dématriçage trouvées dans la littérature. Pour atténuer ce problème, nous proposons une nouvelle fonction d'entraînement qui donnera à l'algorithme une vision globale de la tâche de dématriçage. Les MFSs (voir Figure 1) peuvent être considérées comme différentes dispositions spatiales de motifs. En considérant un ensemble de transformations géométriques $\mathcal{T}$, on peut générer de nouvelles MFSs, augmentant ainsi leur diversité. Par exemple, Figure 2 montre de telles transformations géométriques appliquées à une MFS. Chaque MFS initiale vue durant l'entraînement génère plusieurs nouvelles MFSs qui sont des transformations géométriques de la MFS initiale. Cela assure que la méthode ne surapprendra pas sur des MFSs spécifiques et augmente sa capacité à traiter des images acquises avec d'autres MFSs. En d'autres termes, cette fonction d'entraînement favorise la robustesse face à de nouvelles MFSs ainsi qu'une reconstruction invariante aux transformations géométriques comme suit :

$$f_{\boldsymbol{\theta}}(T(\boldsymbol{A}), \boldsymbol{y}_{T(\boldsymbol{A})}) = f_{\boldsymbol{\theta}}(\boldsymbol{A}, \boldsymbol{y}_{\boldsymbol{A}}) = \boldsymbol{x}, \quad \forall T \in \mathcal{T},$$

où $f_{\boldsymbol{\theta}}$ est l'algorithme considéré, $\boldsymbol{A}$ est une MFS vue durant l'entraînement, $\boldsymbol{y}_{\boldsymbol{A}} = \boldsymbol{A}\boldsymbol{x}$ l'observation de $\boldsymbol{A}$ et $\boldsymbol{y}_{T(\boldsymbol{A})} = T(\boldsymbol{A})\boldsymbol{x}$ l'observation de $T(\boldsymbol{A})$. Dans ce travail, ce concept est lié à l'idée d'équivariance [3] et est adapté pour favoriser que l'opérateur de reconstruction d'image soit agnostique par rapport à la MFS.

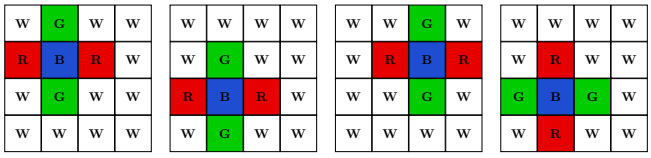


FIGURE 2 : MFS Luo et quelques transformations possibles.

Étant donné Ξ le jeu de données d'entraînement d'images RVB et Ψ l'ensemble des MFSs sur lesquelles s'entraîner, la propriété de reconstruction robuste aux MFSs est imposée par la fonction d'entraînement :

$$\mathcal{L}_{\boldsymbol{\theta}}(\Xi, \Psi, \mathcal{T}) = \frac{1}{|\Xi||\Psi||\mathcal{T}|} \sum_{i=1}^{|\Xi|} \sum_{j=1}^{|\Psi|} \sum_{l=1}^{|\mathcal{T}|} \|\boldsymbol{x}^i - \hat{\boldsymbol{x}}^i_{j,l}\|^2,$$

où $\boldsymbol{x}^i \in \Xi$, $\boldsymbol{A} \in \Psi$ et $\hat{\boldsymbol{x}}^i_{j,l} = f_{\boldsymbol{\theta}}(T_l(\boldsymbol{A}_j), T_l(\boldsymbol{A}_j)\boldsymbol{x}^i)$ est la sortie de l'algorithme.

3 Résultats expérimentaux

Cette section présente les résultats expérimentaux de notre méthode comparée à la version classique de déroulement d'algorithme (sans notre fonction d'entraînement, mais appris de la même façon que notre méthode), à Deep Image Prior (DIP) [14], la méthode TV-ADMM [9] sur laquelle notre méthode est basée, utilisant les Variations Totales pour la régularisation et la méthode d'interpolation bilinéaire [12].

Notre méthode est entraînée sur 300 images du jeu de données BSD500 [1] et 13 MFSs. Les expériences sont réalisées sur le jeu de données multispectral CAVE [15]. Pour mesurer l'efficacité de notre méthode avec plusieurs filtres RVB, chacune des 31 images du jeu de données est échantillonnée avec les réponses spectrales RVB provenant de deux banques de réponses spectrales : [7], contenant les réponses de 28 appareils photo ; et [13], contenant les réponses de 20 téléphones. Enfin, ces 1488 images RVB créées servent de référence pour les calculs de PSNR et SSIM, et chacune va être observée avec 5 MFSs non-vues durant l'entrainement et qui ne sont pas des transformations des MFSs vues pendant l'entrainement.

Table 1 montre que notre méthode est performante sur l'ensemble des MFSs vues durant l'entraînement, mais aussi sur des MFSs qui n'ont pas été vues, à l'inverse du déroulement classique a surappris sur les MFSs d'entraînement. Enfin, Figure 3 présente les images RVB reconstruites par les différentes méthodes étudiées.

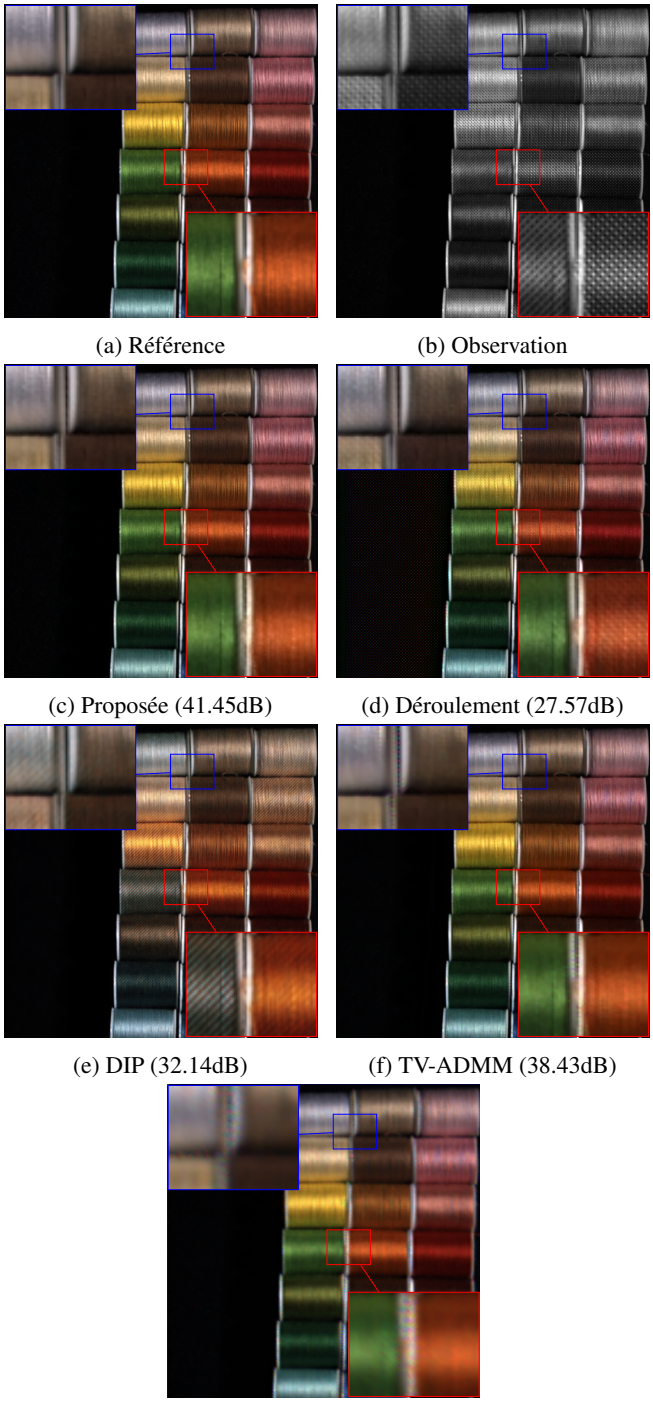
TABLE 1 : Moyenne et écart-type du PSNR et SSIM sur les MFSs vues et non-vues durant l'entraînement, sur les 48 réponses spectrales testées. Meilleurs résultats en **gras**, seconds soulignés.

Méthode	MFSs vues		MFSs non-vues	
	PSNR (dB) ↑	SSIM ↑	PSNR (dB) ↑	SSIM ↑
Méthode proposée	$\mathbf{42.17 \pm 03.90}$	$\mathbf{0.98 \pm 0.01}$	$\mathbf{38.39 \pm 03.64}$	$\underline{0.96 \pm 0.03}$
Déroulement [8]	$\underline{40.82 \pm 02.43}$	$\mathbf{0.98 \pm 0.01}$	32.68 ± 02.97	0.93 ± 0.04
DIP [14]	34.93 ± 02.42	0.95 ± 0.01	36.89 ± 02.57	$\mathbf{0.97 \pm 0.01}$
TV-ADMM [9]	37.06 ± 03.44	$\underline{0.97 \pm 0.02}$	$\underline{37.17 \pm 03.39}$	$\mathbf{0.97 \pm 0.02}$
Bilinéaire [12]	30.88 ± 03.35	0.92 ± 0.06	32.64 ± 03.51	0.94 ± 0.05

4 Conclusion

Cet article propose une méthode générique pour résoudre le problème de dématriçage sur un ensemble de MFSs, avec ou sans filtre panchromatique. De plus, la méthode est robuste aux différents filtres spectraux venant de 48 téléphones et appareils photo, confirmant ainsi l'utilité de la méthode avec des appareils existants. La méthode repose sur une version déroulée d'ADMM spécifique au dématriçage, et est entrainée avec une nouvelle fonction d'entraînement favorisant un apprentissage robuste aux multiples configurations de MFSs existantes.

De futurs travaux permettront de réduire la différence de performance entre les MFSs vues et non vues durant l'entrainement. Une piste serait d'utiliser d'une architecture de réseau de neurones spécialisée pour le dématriçage, tirant parti de la périodicité des observations.



(a) Référence (b) Observation

(c) Proposée (41.45dB) (d) Déroulement (27.57dB)

(e) DIP (32.14dB) (f) TV-ADMM (38.43dB)

(g) Bilinéaire (28.90dB)

FIGURE 3 : Comparaison des méthodes avec la MFS Sony (non-vue) et les filtres spectraux de l'appareil Canon 20D.

Références

[1] P. ARBELAEZ, M. MAIRE, C. FOWLKES et J. MALIK : Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):898–916, 2011.

[2] S. BOYD et L. VANDENBERGHE : *Convex Optimization*. Cambridge University Press, 2004.

[3] D. CHEN, J. TACHELLA et M. E. DAVIES : Robust equivariant imaging : a fully unsupervised framework for learning to image from noisy and partial measurements. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5647–5656. arXiv, 2021.

[4] L. CONDAT, D. KITAHARA, A. CONTRERAS et A. HIRABAYASHI : Proximal splitting algorithms for convex optimization : A tour of recent advances, with new twists. *SIAM Review*, 65(2):375–435, 2023.

[5] K. FENG, Y. ZHAO, J. C-W CHAN, S. KONG, X. ZHANG et B. WANG : Mosaic convolution-attention network for demosaicing multispectral filter array images. *IEEE Transactions on Computational Imaging*, 7:864–878, 2021.

[6] B.K. GUNTURK, J. GLOTZBACH, Y. ALTUNBASAK, R.W. SCHAFER et R.M. MERSEREAU : Demosaicking : color filter array interpolation. *IEEE Signal Processing Magazine*, 22(1):44–54, janvier 2005.

[7] Jun JIANG, Dengyu LIU, Jinwei GU et Sabine SÜSSTRUNK : What is the space of spectral sensitivity functions for digital color cameras ? *In 2013 IEEE Workshop on Applications of Computer Vision (WACV)*, pages 168–179. IEEE, 2013.

[8] V. MONGA, Y. LI et Y. C. ELDAR : Algorithm unrolling : Interpretable, efficient deep learning for signal and image processing. *IEEE Signal Processing Magazine*, 38(2):18–44, mars 2021.

[9] M. MULLER, D. PICONE, M.a DALLA MURA et M. O. ULFARSSON : Model-based demosaicking for acquisitions by a rgbw color filter array. *In IGARSS 2023 - 2023 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2023.

[10] D. PICONE, M. DALLA MURA et L. CONDAT : Joint demosaicing and fusion of multiresolution coded acquisitions : A unified image formation and reconstruction method. *IEEE Transactions on Computational Imaging*, 9:335–349, 2023.

[11] Olaf RONNEBERGER, Philipp FISCHER et Thomas BROX : U-net : Convolutional networks for biomedical image segmentation, 2015.

[12] P.R. SMITH : Bilinear interpolation of digital images. *Ultramicroscopy*, 6(2):201–204, 1981.

[13] Shoji TOMINAGA, Shogo NISHI et Ryo OHTERA : Measurement and estimation of spectral sensitivity functions for mobile phone cameras. *Sensors*, 21(15):4985, 2021.

[14] D. ULYANOV, A. VEDALDI et V. LEMPITSKY : Deep image prior. *International Journal of Computer Vision*, 128(7):1867–1888, mars 2020.

[15] F. YASUMA, T. MITSUNAGA, D. ISO et S. K. NAYAR : Generalized assorted pixel camera : Postcapture control of resolution, dynamic range, and spectrum. *IEEE Transactions on Image Processing*, 19(9):2241–2253, 2010.