



HAL
open science

Semi-Gradient for Color Pixel Reconstruction in a RGBZ CMOS Sensor

Valentin Rebiere, Antoine Drouot, Bertrand Granado, Arnaud Bourge,
Andrea Pinna

► **To cite this version:**

Valentin Rebiere, Antoine Drouot, Bertrand Granado, Arnaud Bourge, Andrea Pinna. Semi-Gradient for Color Pixel Reconstruction in a RGBZ CMOS Sensor. IEEE Sensors 2020, Oct 2020, Rotterdam (virtual), Netherlands. hal-02981089v1

HAL Id: hal-02981089

<https://hal.science/hal-02981089v1>

Submitted on 27 Oct 2020 (v1), last revised 17 Dec 2020 (v2)

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.

Semi-Gradient for Color Pixel Reconstruction in a RGBZ CMOS Sensor

Valentin Rebiere^{*†}, Antoine Drouot[†], Bertrand Granado^{*}, Arnaud Bourge[†], Andrea Pinna^{*}

^{*}Sorbonne Université, CNRS, LIP6, F-75005 Paris, France.

[†]STMicroelectronics, Montrouge, France.

Email: valentin.rebiere@lip6.fr

Abstract — In this paper, we introduce a novel color pixel reconstruction algorithm for a RGBZ CMOS imager. A RGBZ imager is a Vision System on Chip (VSoC) sensor that captures simultaneously color and depth information with a hybrid pixel matrix. The imager pattern is based on a Bayer filter array where the Z pixel covers an equivalent area of a 2x2-color-pixel size, and one quarter of the total color information is missing. We propose an original adaptive algorithm to reconstruct the color information using a new operator called semi-gradient. The results show improvements on edges, corners, and narrow lines reconstruction, and a reduction of color and structural artefacts compared to classical reconstruction algorithms.

Index Terms — RGBZ CMOS Imager, edge-directed algorithm, local adaptive weight, color depth filter array, VSoC.

I. INTRODUCTION

Systems acquiring both color (RGB) and depth (Z) information are more and more used [1][2]. Everybody knows the Microsoft Kinect [3] or the facial recognition system of the Apple iPhone10. Both of these systems are composed of two sensing devices. One for the color acquisition and one for the depth acquisition based on using InfraRed light (IR). In order to have a more compact solution, without needing a calibration step, merging these two devices into a monolithic RGBZ sensor has emerged in the past few years [4]. New scientific issues are raised to realize these new sensors, at several technical levels, leading to different possible RGBZ matrix architectures: layout of the heterogeneous pixel matrix, energy consumption, readout circuitry, color/IR on-chip filters. Among these, color reconstruction is a problem at the Z pixel locations, where no color information is present. The color filter array (CFA) is incomplete compared to a classical Bayer RGB scheme. The demosaicing step [5] of an image signal processing chain thus cannot be directly applied. Our paper focuses on this color reconstruction challenge. In [6], Shi et al. used a RGBZ pattern that is an alternation of four rows of RGB pixels and two rows of Z depth pixels. They proposed an edge-directed based approach derived from de-interlacing techniques and a color-selective adaptive demosaicing. Their pattern has a spatial distribution structured in rows. In order to have a homogeneous color distribution around a Z pixel for a given

5x5 kernel, we have chosen to work on an RGBZ structure (Fig.1) based on Bayer pattern with a 2-by-2 Z pixel size, where one-quarter of the color pixels are replaced by Z pixels. To our knowledge, there is no dedicated reconstruction method for this RGBZ pattern. Therefore, we studied different universal reconstruction methods: the bilinear interpolation (BI) and an edge directed interpolation (EDI) inspired from [7], where A. Horé and D. Ziou introduced an edge-directed demosaicing method for various CFA patterns. But applying them to our RGBZ matrix, even after some functional adaption, we observed a lot of structural and color artefacts (Fig.4). Consequently, we worked on an original reconstruction algorithm based on an operator that estimates more finely the missing color information in the CFA image.

II. PROPOSED ALGORITHM

First, we implemented an adaptive algorithm using a classical Sobel-based gradient [8][9] that reduces the visibility of the Z pixels pattern across the matrix. However, some structural artefacts are still present along narrow edges (width edges less than 2 pixels). We propose a new operator called semi-gradient to improve the reconstruction. The idea of the semi-gradient is to find for a given orientation, to which side the missing pixel belongs.

A. Gradients computation

The gradients are computed by the sum of pairwise differences between known color pixels. The color pixels used for each difference are located on either side of a given axis of symmetry of the kernel. Fig.2 represents the masks used for the four gradients computation. The mask depends on the kernel configuration (Fig.1.b): Z pixel values are set to 0; only color pixels are used to compute the gradient. The correlation between the R, G and B channels [10][11] is used

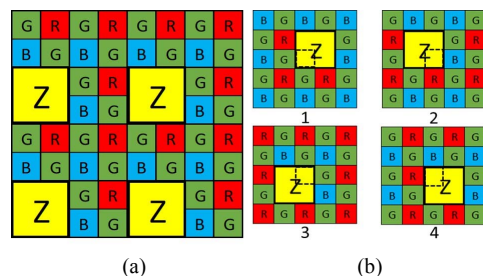


Figure 1. (a) RGBZ pattern. (b) The four kernels depending on the central missing pixel: 1. Blue pixel. 2. Green pixel. 3. Red pixel. 4. Green pixel.

to compute inter-channel differences. This allows detecting edges in any of the three channel and increasing the number of terms, which reduces noise impact. The four gradients (Grads) are normalized following the number of terms used to compute them. Normalized gradients are then discriminated with a gaussian function to compute weights ω for each direction according to the four gradients:

$$\omega_{dir} = e^{-\frac{Grad(dir)^2}{2(\min(Grad)+1)^2}} \quad (1)$$

Where, $\min(Grad)$ corresponds to the minimum of the four gradients and $Grad$ represents a given gradient. The weights $\omega'(i, j)$ are normalized:

$$\omega'(i, j) = \omega(i, j) / \sum_{(g,h) \in \xi} \omega(g, h) \quad (2)$$

Finally, the missing pixel $I(p, q)$ is interpolated with:

$$I(p, q) = \sum_{(g,h) \in \xi} \omega'(g, h) I(g, h) \quad (3)$$

Where ξ is the computation kernel centered on the missing pixel (p, q) and (g, h) the coordinates of the weighted pixel.

The new operator, the semi-gradient, is described in the next section.

B. The semi-gradient operator

Semi-gradient (SG) is a new local operator designed to improve the reconstruction of the missing information, using both intra and inter-channel information. It is firstly used to enhance the gradient computation in order to detect narrow edges (1-2 pixels width). It is also used during the weights computation step to promote weights on the preferred side along an edge. The aim is to evaluate how close neighbors of the missing pixel are similar to pixels in the eight directions. These evaluations allow refining the identification of structures and identifying in which direction there is a continuity of the texture. Hence, we can interpolate the missing pixel along the detected direction.

When the missing pixel is located on an edge, we must interpolate along this edge. To do so, we introduce two values: a Discriminated Semi-Gradient (DSG) and a Discriminated Improved gradient (DIG). The weights are computed as weighted sums between these two values. The weights of the pixels along the edge are mainly defined by the DIG and, conversely, the weights of the pixels located on either side of the edge are mainly defined by the DSG. As we have no information about the missing pixel, the DSGs allow considering image information on each side of the edge. We compare a given DIG to its orthogonal DIG_{\perp} to determine the

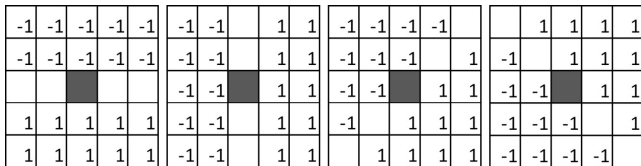


Figure 2. Computation mask centered on the missing pixel, for the four directions of the gradient. Left to right: North-South, West-East, Northwest-Southeast and Southwest-Northeast.

edge and then adapt the weight computation. For example, to compute the edge-directed weight of the North area, we compare DIG_{N-S} with DIG_{E-W} to identify if the edge-directed weight ω needs to be mainly based on the DIG_{N-S} (edge along the North-South axis) or the DSG_N (edge along the East-West axis). The comparison between the two orthogonal gradients for a given orientation ori is defined by:

$$\lambda_{ori} = \frac{DIG_{ori}}{DIG_{ori} + DIG_{ori_{\perp}}} \quad (4)$$

And the edge-directed weight ω_{dir} for a given direction dir is computed by:

$$\omega_{dir} = \lambda_{ori} * DIG_{ori} + (1 - \lambda_{ori}) * DSG_{dir} \quad (5)$$

Finally, weights $\omega(i, j)$ are normalized (2) before the interpolation (3).

DIG and DSG are obtained using semi-gradients. Two SGs are computed for each orientation, e.g. North SG and South SG are computed for the North-South orientation. Each SG results in the summation of pair-wise differences between color pixels: a pixel close to the missing one and a second pixel located near the boundary of the kernel, both located on the same side versus the central point. Pixels on the axis of symmetry for a given direction are considered as close pixels. Two arrays of ten values are computed corresponding to the vertical and the horizontal differences (Fig.3):

$$diff_{vert}[2j] = |I(0, j-2) - I(-2, j-2)|, \text{ if } j \neq 2 \text{ else } 0 \quad (6)$$

$$diff_{vert}[2j+1] = |I(0, j-2) - I(2, j-2)|, \text{ if } j \neq 2 \text{ else } 0 \quad (7)$$

$$diff_{hori}[2j] = |I(j-2, 0) - I(j-2, -2)|, \text{ if } j \neq 2 \text{ else } 0 \quad (8)$$

$$diff_{hori}[2j+1] = |I(j-2, 0) - I(j-2, 2)|, \text{ if } j \neq 2 \text{ else } 0 \quad (9)$$

$I()$ are color pixels in the kernel. Pixels corresponding to a part of the Z pixel are not considered in these equations. The index $j \in [0, 4]$ is the index of both arrays. North differences correspond to (6), South to (7), West to (8), and East to (9). Horizontal and vertical differences are then associated to form the semi-gradients according to their direction. For example, semi-gradient of the North area (10) is computed using the vertical North differences, and the semi-gradient of the North-West area (11) is computed using vertical North differences and horizontal West differences:

$$SG_N = \sum_{j \in \{0; 2; 6; 8\}} diff_{vert}[j] \quad (10)$$

$$SG_{NW} = \sum_{j \in \{0; 2\}} diff_{vert}[j] + diff_{hori}[j] \quad (11)$$

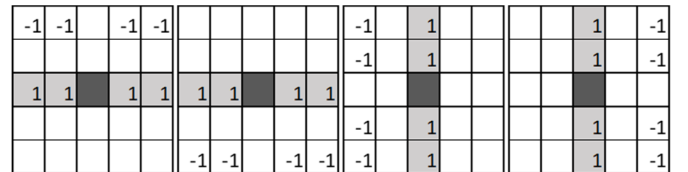


Figure 3. Computation masks of the arrays. Close pixels are marked in grey. Left to right: North differences, South differences, West differences and East differences.

For each SG, the number of color differences could vary due to the Z pixel location inside the kernel (Fig.1.b). They are then used to improve the four gradients Grads. The improved gradient (IG) is defined by:

$$IG = \frac{1}{k_{tot}}(Grad + SG_1 + SG_2) \quad (12)$$

Where k_{tot} is the normalization factor corresponding to the number of terms used to compute the Grad and both SGs, and SG_1 and SG_2 are the two semi-gradients associated to the gradient Grad. For example, for $Grad_{NW}$, $SG_1=SG_N$ and $SG_2=SG_W$. We use a scaled version of the SGs to compensate for the missing differences due to lack of color pixels at Z location (13):

$$SG_{scaled} = \frac{SG}{N} \quad (13)$$

Where SG_{scaled} is the scaled semi-gradient, and $N \in [3,4]$ the number of terms used to compute the corresponding semi-gradient. Before the computation of the interpolation weights, both IGs and scaled SGs are separately discriminated using (1) to obtain the four discriminated improved gradients (DIGs) and the eight discriminated semi-gradients (DSGs).

III. EXPERIMENTAL RESULTS

In this section, we present a series of tests made on the 18 images from the McMaster dataset [12] and on the 24 images from the Kodak dataset [13]. To assess the algorithms, a virtual test bench has been implemented. The RGBZ images are generated from color images using a RGBZ sensor model based on a noise model described in [14]. The missing pixel reconstruction algorithms are included into a typical image sensor processor (ISP) to generate the full color images. Classical steps as the white balance correction, demosaicing and gamma correction are computed here. The demosaicing algorithm used is described in [15]. A reference image corresponding to the original image sampled with the Bayer pattern and processed with the same chain is also built to assess the quality of the reconstructed image. Four methods have been implemented: the bilinear interpolation (BI), the edge-directed interpolation (EDI) [7], the proposed solution without the semi-gradient improvement (PS) and the proposed solution with the semi-gradient improvement (PS-

TABLE I
PERFORMANCE COMPARISON OF RECONSTRUCTION METHODS IN TERMS OF PSNR, C-PSNR AND SSIM

Kodak	BI	EDI	PS	PS-SG (our solution)
PSNR	27.00	27.61	28.21	28.70
C-PSNR	31.72	32.65	33.12	33.62
SSIM	0.944	0.950	0.953	0.957
McM	BI	EDI	PS	PS-SG (our solution)
PSNR	27.53	28.46	28.90	29.31
C-PSNR	31.78	33.07	33.46	33.88
SSIM	0.943	0.952	0.956	0.958

SG). They have been evaluated using the peak signal-to-noise ratio (PSNR), the color-peak signal-to-noise ratio (CPSNR) [16] and the Structural Similarity index (SSIM) [17]. The PSNR metric is used to evaluate the quality of the CFA images (prior to the demosaicing step) while both C-PSNR and SSIM metrics are applied on the final color images. We have reported results in Table I. As we can see, the proposed solution using the semi-gradient improvement leads to the best reconstruction. Structural and color artifacts are visually reduced (Fig.4). Some structural artifacts due to the Z pixel pattern are noticeable for the BI, EDI and the PS while they are significantly reduced with the PS-SG algorithm.

IV. CONCLUSION

In this paper, we have described a new algorithm to fill the missing color information for a RGBZ matrix in order to reconstruct a full Bayer image. Due to the unique structure of the RGBZ matrix where one quarter of the color information is missing and each Z pixel has a size of a 2x2-color-pixel kernel, the reconstruction becomes a challenging problem. Three classical methods were studied, and we developed an original reconstruction algorithm based on a new operator: the semi-gradients. The experimental results have shown that our algorithm is at the state of the art in terms of visual aspect and objective quality measures, compared to classical interpolation algorithms adapted to our RGBZ matrix. To go further, we will investigate joint reconstruction methods using both color information and amplitude information from depth sensing.



Figure 4. Crops (118*112 pixels) of Kodak8 (768*512) and Kodak19 (512*768).

REFERENCES

- [1] R. J. Gove, *CMOS image sensor technology advances for mobile devices*, 2nd ed. Elsevier Ltd., 2020.
- [2] Y. M. Wang *et al.*, “A Low-Power Low-Cost High-Speed 2D / 3D Camera for Virtual Reality Headsets , Mobile Devices and Automobiles,” pp. 270–273, 2017.
- [3] Microsoft, “Xbox 360 Kinect Sensor Manual,” 2010.
- [4] W. Kim *et al.*, “A 1.5Mpixel RGBZ CMOS image sensor for simultaneous color and range image capture,” *Dig. Tech. Pap. - IEEE Int. Solid-State Circuits Conf.*, vol. 55, no. June 2011, pp. 392–393, 2012.
- [5] B. K. Gunturk, J. Glotzbach, Y. Altunbasak, R. W. Schafer, and R. M. Mersereau, “Demosaicking: Color filter array interpolation,” *IEEE Signal Process. Mag.*, vol. 22, no. 1, pp. 44–54, 2005.
- [6] L. Shi, I. Ovsianikov, D. Min, Y. Noh, W. Kim, and S. Jung, “Demosaicking for RGBZ Sensor,” *SPIE Proc.*, vol. 8657, pp. 1–9, 2013.
- [7] A. Horé and D. Ziou, “An edge-sensing generic demosaicking algorithm with application to image resampling,” *IEEE Trans. Image Process.*, vol. 20, no. 11, pp. 3136–3150, 2011.
- [8] S. Jin, W. Kim, and J. Jeong, “Fine directional de-interlacing algorithm using modified Sobel operation,” *IEEE Trans. Consum. Electron.*, vol. 54, no. 2, pp. 857–862, 2008.
- [9] Z. Xiangxi, Z. Yonghui, Z. Shuaiyan, and Z. Jian, “FPGA implementation of edge detection for Sobel operator in eight directions,” *2018 IEEE Asia Pacific Conf. Circuits Syst. APCCAS 2018*, pp. 520–523, 2019.
- [10] S. Pei and I.-K. Tam, “Effective color interpolation in ccd color filter arrays using signal correlation,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 6, pp. 503–513, 2003.
- [11] R. Kimmel, “Demosaicing: Image reconstruction from color CCD samples,” *IEEE Trans. Image Process.*, vol. 1406, no. 9, pp. 610–622, 1998.
- [12] L. Zhang, X. Wu, A. Buades, and X. Li, “Color Demosaicking by Local Directional Interpolation and Nonlocal Adaptive Thresholding,” *Electron. Imaging*, pp. 1–29, 2011.
- [13] “Kodak color image dataset.” [Online]. Available: <http://r0k.us/graphics/kodak/>.
- [14] J. Alakarhu, “Image Sensors and Image Quality in Mobile Phones,” pp. 2–5, 2007.
- [15] L. Alacoque, “United States Patent Application Publication: Method of demosaicing a digital raw image, corresponding computer program and graphics or imager circuit,” 0063682, 2012.
- [16] D. Alleysson, S. Süsstrunk, and J. Héroult, “Linear demosaicing inspired by the human visual system,” *IEEE Trans. Image Process.*, vol. 14, no. 4, pp. 439–449, 2005.
- [17] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: From error visibility to structural similarity,” *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, 2004.