Color Pixel Reconstruction for a Monolithic RGB-Z CMOS Imager

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

Noname manuscript No.
(will be inserted by the editor)

Color pixel reconstruction for a monolithic RGB-Z CMOS Imager

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

Received: date / Accepted: date

Abstract In this paper, we introduce the challenges, intrinsic and extrinsic, of color and depth sensors integration in the same matrix for a monolithic RBG-Z CMOS imager system. Due to the fact that the technology to conceive this type of circuit is still under development, the challenge that we address is the extrinsic one. It is a consequence of the heterogeneity of the matrix, where information is missing compared to what can be provided by separate RGB and Z systems. For that a first evaluation is done taking into account how the RGB-Z patterns could impact the demosaicing step. The evaluated pattern are in function of the different sizes between color and depth pixels. For the missing color reconstruction we have evaluated the state of the art algorithms, adapted to the missing information, and we propose an original adaptive algorithm using a new operator called *semi-gradient* (SG).

To fill the lack of a mature technology for which real images are missing for this type of CMOS imager, a test environment was created and then used with three different databases, Kodak, McMaster, HDR+burst. The results show improvements on edges, corners, and narrow lines reconstruction, and a reduction of color and structural artefacts compared to the state-of-the-art reconstruction algorithms.

STMicroelectronics
29 Boulevard Romain Rolland, 92120 Montrouge, France.
E-mail: valentin.rebiere@st.com
E-mail: antoine.drouot@st.com
E-mail: arnaud.bourge@st.com

Sorbonne Université, CNRS, Laboratoire d'Informatique Paris 6 - LIP6,
4, Place Jusssieu, 75005 Paris, France.
E-mail: andrea.pinna@lip6.fr
E-mail: bertrand.granado@lip6.fr

Keywords RGBZ CMOS Imager · edge-directed algorithm, · local adaptive weight · color depth filter array · Vision System on Chip (VSoC).

1 Introduction

The Vision Systems on Chip (VSoC) are becoming more and more complex, in terms of spatial and frequency resolution, frame rates already reached 1000 frames per second [10], but also in terms of quality [8] and integration of new functionalities [5]. A VSoC is based on the close cohabitation between signal acquisition and processing on the same chip. Among the VSoC, the RGB-Z imagers are systems able to capture a flat color image (RGB), and its depth (Z). The RGB-Z imager could be realized as a heterogeneous system based on two different sensors integrated separately. The first one is a classical color image sensor while the second one acquires the depth of the observed scene using different techniques, like as : active stereo-vision, passive stereo-vision and Time of Flight (ToF).

Since the development of the Microsoft Kinect[19], many applications using both color and depth information, like as gesture recognition, virtual reality, 3D mapping and modeling applications[14], have been proposed. Lately, the face recognition and identification function of the iPhoneX [16], has highlighted the need of integrating a RGB-Z system in mobile phones. The smartphone manufacturers [9] are torn between the need to add more and more sensors and cameras for more functionalities on the one hand, and the will to limit the number of slots (as many holes in the shell) for design reasons. The example of the smartphone can be extended to other areas such as autonomous navigation

(drones, robots) and IoT[1], virtual or augmented reality headsets, or even the automobile and medical devices.

Heterogeneous RGB-Z imager needs a calibration step in order to align the data (RGB on one side and Z on the other). This in turn entails the possibility of errors in data acquisition and a greater consumption of energy and resources, thus putting at risk applications where precision and energy autonomy are necessary.

In order to overcome the limitations in terms of size and need for calibration of heterogenous RGB-Z imager, an RGB-Z monolithic integration has begun to be studied [12]. Thanks to advances in semiconductor technology such an imager combines both color and depth-sensing at the pixel level and in the same matrix.

In a monolithic image sensor, one of the problems is to reconstruct the missing information to obtain a full resolution sensor. As in an RGB image sensor where full-color pixels are computed using neighborhood pixels, in an RGB-Z system it is necessary to extrapolate the missing information from the neighborhood.

In this article, we introduce an original method to reconstruct the missing color information in a monolithic RGB-Z imager where RGB and Z pixels are in the same matrix, and we evaluate it in terms of image quality reconstruction. We consider a monolithic RGB-Z sensor composed of color pixels (R, G and B) and pinned photodiodes [24] used to acquire depth information using an indirect time-of-flight (i-ToF) technology.

The rest of this paper is organized as follows : section 2 some mixed matrix and more specifically the monolithic RGB-Z architectures and their limits. Section 3 analyzes the impact of the RGB-Z pattern on the overall architecture. Section 4 describes the algorithm for the color information reconstruction, needed before the demosaicing step. Section 5 shows the tool chain and the simulation results on the RGB-Z pattern study and on the pixel color reconstruction. A brief discussion on the results can be found in section 6. At last, the conclusion is presented in section 7.

2 Mixed Matrix Imagers

A mixed matrix is composed of color pixels (visible wavelength) on the one hand, and other pixels (non-visible wavelength) on the other hand. Those pixels could be range pixels as in a RGB-Z sensor or IR pixels as in a RGB-IR sensor.

[1] Internet of Things

2.1 Monolithic RGB-IR Imager

RGB-IR sensors acquire both RGB and Near Infra-Red (NIR) images. They are composed of a mosaic of RGB and IR pixels. The information acquired by the IR pixels is similar to the information acquired by the color pixels. The difference is that the photons acquired by IR pixels belong to the spectrum of infrared (more than 800 nm). Several RGB-IR filter arrays exist [20]. The density and position of IR pixels vary from one architecture to another. Depending on the available color and NIR information, the reconstruction methods could vary [28]. There are also machine learning based reconstruction methods [21] that seem to show better results than gradient based demosaicing methods. However, these methods are, at this time, costly to implement on embedded devices.

Moreover, most of RGB-IR sensors have no dedicated IR filter for the color pixels. IR light is not blocked and is collected in addition to the visible light: color pixels are then polluted. However, the IR pixels provide a good measure of the IR signal and this is subtracted from the color pixels [13].

2.2 Monolithic RGB-Z Imager

In the case of RGB-Z sensors, Shi et al. [27] propose a missing pixel reconstruction method followed by a demosaicing to reconstruct a full color image from a RGB-Z imager. The two steps are performed consecutively and the missing pixel reconstruction step is adapted to the RGB-Z matrix.

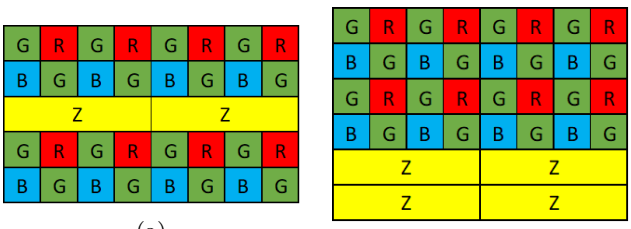


Fig. 1: Representation of (a) the theoretical RGB-Z matrix introduced in [12] and (b) the RGB-Z matrix introduced in [27].

The pixel matrix is similar to the one proposed in [12] (see Fig 1.b) but in this case, there is an alternation of four rows of RGB pixels and two rows of Z pixels (see Fig 1.b). The reconstruction method of the missing information has been patented [26]. The reconstruction process is a two-step algorithm. The first step

consists of reconstructing the missing color information corresponding to the Z-pixel. Which color will be reconstructed depends on which type of filter has been chosen. This kind of filter, named Color Filter Array (CFA), defines the mosaic of the color filters placed over the sensor. The most used CFA is the Bayer filter composed of alternating lines of Green-Red and Blue-Green pixels (like the two first lines in Figure1). For a monolithic RGB-Z imager, the CFA must integrate a new filter for the Z pixel. In this case to define the filter array we use the CDFA[2] acronym, where D is for Depth. During the second step, a demosaicing algorithm is applied in order to reconstruct the complete color image. For a Bayer filter, three color images will be reconstructed, one Red, one Green and one Blue.

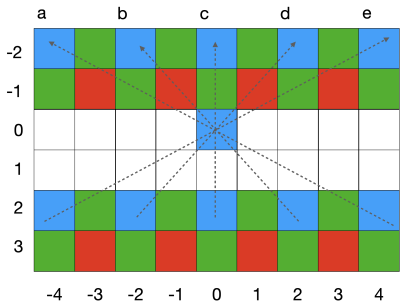


Fig. 2: Example of a directional interpolation of a missing blue pixel based on a 9x6 pixels kernel [27].

In [27], the missing pixel interpolation method is an EDI[3] algorithm based on the directional correlation between the pixels in the adjacent original scanned lines. For a given missing color pixel, five edge directions are evaluated. An example is presented on the Figure 2. It is a blue pixel reconstruction where the five edge directions (a, b, c, d and e) are computed as follow:

$$a = |I(y-2, x-4) - I(y+2, x+4)| \tag{1a}$$
$$b = |I(y-2, x-2) - I(y+2, x+2)| \tag{1b}$$
$$c = |I(y-2, x) - I(y+2, x)| \tag{1c}$$
$$d = |I(y-2, x+2) - I(y+2, x-2)| \tag{1d}$$
$$e = |I(y-2, x+4) - I(y+2, x-4)| \tag{1e}$$

Where $I(y, x)$ represents the pixel value at the location (y, x). The five differences are then evaluated. If the region contain a dominant edge, a median filter is performed in the direction of the highest correlation. Otherwise, for a relatively homogeneous area, a bilinear interpolation along vertical region is performed. Once the Bayer image is fully reconstructed, the demosaicing step can be applied. This is an adaptive interpolation solution mainly inspired from [22]. The weights used in the interpolation are computed following the local pixel similarities and their position in the kernel. Compared to the solution proposed by R. Ramanath al. [22] add a color-selective kernel that specifies neighboring pixels according to the reconstructed pixel. This allows to only use known pixels during the demosaicing step, rather than the newly reconstructed pixels in step one. Reconstruction results are shown in the Figure 3. They show the interest of a dedicated algorithm to interpolate the missing color pixels. There are noticeable structural and color artifacts present in the image reconstructed using a bilinear interpolation (Figure 3.d) compared to the image reconstructed using the proposed EDI algorithm (Figure 3.e).

The implementation of a monolithic RGB-Z imager presents new constraints at different levels [12]. We can summarize them in order to highlight the paper's contributions. The constraints could be splitted into two categories:

- Intrinsic issues at the sensor level. As it is a heterogeneous system, where two different types of sensor share the same pixels matrix, the color filtering system must be adapted. Most of the depth acquisition systems operate with an IR illumination. The color pixels should not be exposed to NIR[4] light and the depth pixels should not be exposed to visible light. In this context, a new filtering system must be designed to apply a local NIR filtering on color pixels and a local filtering of the visible light on depth pixels. The layout and the readout circuit should also be adapted. The overall architecture of the matrix depends on the pixel distribution and the difference between the color and depth pixel pitch sizes.
- Extrinsic issues at the image processing level. New types of defects are induced by the technological limits of the sensor. Due to the matrix of mixed pixels, color information is missing at the Z-pixel location and vice versa. This leads to resolution problems and incomplete information. In addition, the acquisition modes (integration time, frame rate) may differ between the two pixel technologies and therefore the data format and output stream may be different.

The contributions of this paper are mainly focused on the extrinsic issues. It is the extended and the detailed version of our previous paper [23]. The novel

[2] Color Depth Filter Array
[3] Edge Directed Interpolation
[4] Near InfraRed

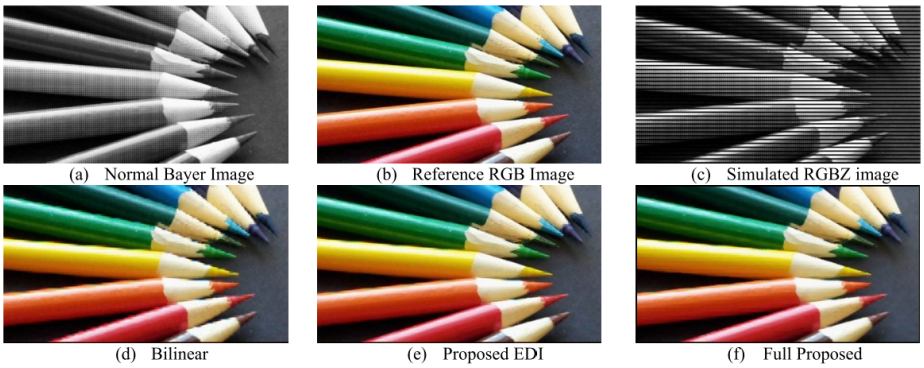


Fig. 3: Results of the reconstruction method applied to a simulated RGBZ image proposed in [27]. (a) Normal Bayer image. (b) Reference RGB image corresponding to (a), demosaiced by Malv ar et al. [18]. (c) Simulated RGBZ image by blacking out rows in (a). (d) Reconstructed image by filling the missing rows with bilinear interpolation (demosaiced by Malvar et al. [18]). (e) Reconstructed image using EDI method [27] (demosaiced by Malvar et al. [18]). (f) reconstructed image using EDI method [27] and color adaptive demosaicing [27].

parts of this paper are as follows: (i) we propose a new RGB-Z matrix using a 1-by-1 Z-pixel, (ii) we adapt the semi-gradient method to the new RGB-Z matrix after the study of its artifacts, (iii) we introduce the simulation chain used to implement and evaluate the different algorithms, and (iv) we report additional results using various high-resolution color images from the HDR+ burst dataset and we compare the two RGB-Z studied matrix.

3 RGB-Z CDFA evaluation

In the context of RGB-Z imager, the objective is to have as few color artifacts as possible while having a sufficient depth map resolution. Shi et al.[27] proposed a reconstruction solution dedicated to a RGB-Z matrix [12] where entire lines of color pixels are missing. A narrow horizontal edge perfectly aligned with these missing pixels lines would be difficult to detect and thus to reconstruct. This shows the importance of the choice of the RGB-Z pattern in order to reconstruct the colors as well as possible beyond the presence of the depth pixel.

In the context where the quality of the color image is one of the first performance criteria of a RGB-Z imager, the main constraint concerns the missing color reconstruction at the Z-pixels location. The implemented solutions to reconstruct the missing color information depend on which CFA pattern is based the CDFA matrix. In this paper we have worked with the Bayer pattern as a starting point for the CFA, on which groups of pixels are replaced by Z pixel(s) to produce the CDFA. From this structure, the impact of the Z-pixel position and dimension on the demosaicing step is explored before to evaluate the best algorithm for the missing color pixel reconstruction.

3.1 Z pixel dimension and sampling

A first constrain is the pixels dimensions. The larger a Z-pixel is compared to a color pixel, the more color information is missing at the Z-pixel location. From the technology point of view the challenge is to shrink as possible the Z pixels dimension. Z pixels cannot be as small as the smallest possible color pixel. This is a key point in order to facilitate the reconstruction of missing color pixels.

Two different Z-pixel dimension have been evaluated:

- The first one correspond to the Z-pixel solution where it has the same dimension of color pixels, we define this pixel as the *1x1-Z-pixel* (Figure 4.a).
- The second one, the Z-pixel has double dimension as the previous one, in this case color pixels are quarter the size of the Z-pixels. We define this second pixel as the *2x2-Z-pixel* (Figure 4.b).

A second constraining aspect is the sampling of the Z-pixel across the matrix. Indeed, the more Z-pixels there are, the less color information is available. However, it is necessary to ensure that the Z resolution is sufficient. To answer this question without real data, we made an assumption based on the recognized *iPhone X TrueDepth* system [6]. It is a well known face recognition system based on a Structured Light technology where 30,000 dots are projected on the target. This implies that a maximum of 30,000 points are recorded

(a) (b)

Fig. 4: Proposed Z-pixels. (a) 1x1-Z-pixel: its dimension is the same as color pixel dimension. (b) 2x2-Z-pixel: its dimension is twice the size than color pixel dimension.

to compute the depth map. Our assumption is that a minimum of 30,000 pixels is needed to have a suitable resolution for the Z-pixel distribution. Considering a 8 Megapixel sensor with a ratio of 1/36 Z-pixels, we have more than $200,000$ depth measurement points, which is much higher than the hypothesis made. However, it can be risky to have a too sparse distribution of the Z pixels on the matrix. Especially concerning the quality of the depth measurement. Denoising methods use kernels of few pixels. If the Z-pixels are too spread out in the matrix , there is not enough information for a given kernel to process denoising properly. The impact of noise on the depth measurement may be too significant. In addition, the Z-pixel's undersampling can induce aliasing effects. This is why we consider a dense distribution RGB-Z matrix. The total area of the Z-pixels represents 1/4 of the total area of the pixel matrix. Below the 1x1-Z-pixel and the 2x2-Z-pixel will be named *Mask1* and *Mask2* respectively.

3.2 CFDA Mask1 and Mask2 impact on Bayer CFA

The demosaicing step reconstructs 2/3 of the total information of a color image. During this step two color values are reconstructed for each pixel. For example, for a known red pixel, the blue and green values are reconstructed during demosaicing. Whereas the reconstruction of the missing color information due to the Z-pixel represents, in the cases presented above, one quarter of the information of the CFA image. This represents 1/12 of the total information of the color image. The demosaicing step should not be neglected in favor of the step of missing color information reconstruction due to the Z-pixel. This is why we have chosen to focus on the study of RGB-Z architectures based on the Bayer pattern. The demosaicing algorithm [3] used in this work is then dedicated to the Bayer pattern. The CFA images (without any Z pixel) reconstructed with this demosaicing algorithm [3] are now considered as reference images in order to evaluate the missing color information algorithms. We now consider these images as a point of comparison and as a quality target to reach. A representation of the unitary elements of the two RGB-Z architectures studied in this work are shown in Figure 4.
For the Mask1, one quarter of the color information is missing in the CDFA image. It is important to note the absence of green pixels on the rows and columns with Z-pixels, it is relevant as the green pixels represent the most important information (luminance).
For the Mask2, four color pixels are missing at a Z-pixel location, one quarter of the color information is missing in the CDFA image. The difference with the Mask1 is the missing color distribution. In this case one blue, one red and two green pixels are missing. However, there is in this case, no absence of complete color information on rows and columns.

In the context of a RGB-Z imager, the choice of one CDFA rather than another (choice between the two masks) may imply to modify the reconstruction solution and thus the quality of this reconstruction. In the same way, the choice of one CFA rather than another impacts the demosaicing step. It is therefore important to study the quality of the reconstructed image at the output of the demosaicing step regarding the CFA.

In the next section, we will present the algorithms study for a Bayer CFA reconstruction from RGB-Z CDFA.

4 CFA reconstruction algorithms evaluation

In the same way as Shi [27], we first explored classical reconstruction methods that we adapted to our RGB-Z matrices presented in the section 3. This exploration highlights the limitations of these methods. A new operator called semi-gradient is then proposed to correct the artifacts and thus improve the image quality. Classical methods such as bilateral interpolation and edge-directed interpolation based on gradient computation are first presented. Then, we will present our proposed method using semi-gradients to correct the highlighted limitations. All algorithms are implemented using the Python 2.7 environment and some parts of the algorithms, which need to be accelerated, are implemented in C.

4.1 Bilateral based algorithm

We have implemented an algorithm inspired from the bilateral interpolation (BI) [29]. Due to the absence of color information at the location of the interpolated pixel, the intensity parameter of a given weight is calculated by evaluating its intensity regarding the pixels

from the same color channel in the considered neighborhood. The computational kernel used corresponds to a 5x5 window ξ centered on the missing pixel. The weights are computed using 2. For each weight, the intensity parameter β, is computed using 4, and the distance parameter α is computed with a 2-dimension Gaussian filter 3.

$$\omega(i,j) = \alpha(i,j) \times \beta(i,j) \tag{2}$$

$$\alpha(i,j) = \exp\left(-\frac{i^2+j^2}{2\sigma^2}\right), \tag{3}$$

$$\beta(i,j) = \frac{1}{1 + \sum\limits_{(g,h)\in\xi_G} |I(i,j)_G - I(g,h)_G|} \tag{4}$$

The red (respectively blue) pixel interpolation is computed as follow :

$$I(p,q)_R = I(p,q)_G + \sum_{(i,j)\in\xi_R} \omega'(i,j) \times (I(i,j)_R - I(i,j)_G) \tag{5}$$

σ was empirically determined according to the kernel size and is equal to 1. The choice of a Gaussian filter allows a faster decrease of the weight according to the distance. Thus, it values the pixels close to the missing pixel. On the Figure 5 are represented two simulated RGB-Z images (Mask1 and Mask2) reconstructed using the bilateral algorithm and demosaiced using [3] compared to the reference Bayer image demosaiced with [3]. On the Mask1, we can see a blurring effect along the edges of the fence. On the Mask2, the RGB-Z pattern is clearly visible. This is similar to a Zipper effect of two pixels wide, which corresponds to the width of a Z-pixel. We see that this filter is not able to prevent interpolation across edges.

4.2 Edge directed algorithm

An edge-directed interpolation (EDI) solution has been implemented to overcome the structural artifacts that are mainly located around the edges. This method uses gradients calculation to determine the direction of the edges. Two methods for calculating the gradients are explored. The first solution is directly inspired from the first step of the algorithm describes in [11]. The gradients are calculated by central symmetry around the missing pixel (EDICentral). For the second one, the gradients are now calculated with respect to the axes of symmetry of the kernel (EDILine). A representation of these axes of symmetry is given in the Figure 6.

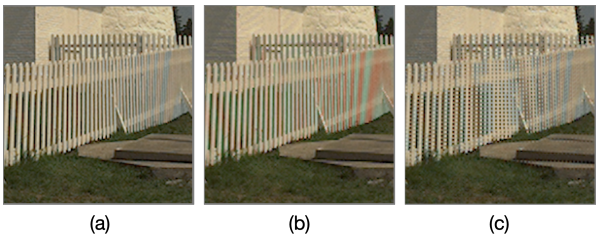


Fig. 5: Color images reconstructed from simulated RGB-Z data using the bilateral solution compared to the reference image. RGB-Z data are simulated using both Mask1 (4a) and Mask2 (4b). The demosaicing algorithm used for the three reconstructed images is [3]. (a) Reference cropped image. (b) simulated RGB-Z image reconstructed based on the Mask1. (c) simulated RGB-Z image reconstructed based on the Mask2.

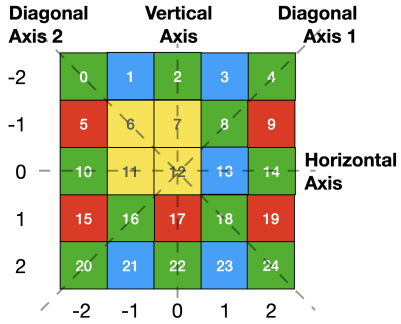


Fig. 6: Representation of a computational kernel of the Mask2. The four axes of symmetry are illustrated by the black lines.

4.2.1 Gradients computed around the missing pixel

To compute the weights used for the interpolation of a missing color pixel, equation 2 need to take into account also the edge parameter γ.

$$\omega(i,j) = \alpha(i,j) \times \beta(i,j) \times \gamma(ori) \tag{6}$$

Equations 3 and 4 are still valid. The calculation of the edge parameter is different. The areas are no longer overlapped to compute it. Indeed, the majority of artifacts in the reconstructed RGB-Z images are located along the edges. Overlapping areas when calculating the edge parameters does not result in sharp edges after interpolation. The overlapping may value the weights of some pixels that are located between two areas that do not necessarily correspond to the main direction determined for the interpolation. The edge parameter is now computed using the four areas illustrated on the Figure 7.

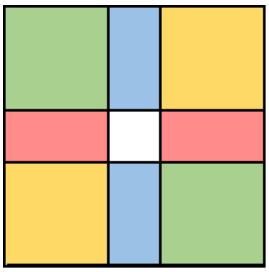


Fig. 7: Representation of the four non-overlapped areas corresponding to the orientations *ori* to compute the four gradients. The blue areas correspond to the vertical gradient, the red areas correspond to the horizontal gradient. The yellow and green areas correspond to the two diagonal gradients.

A Gaussian discrimination function is used for the missing color pixel interpolation because it has a faster slope's decrease of the function. This allows a stronger discrimination of the gradients.

$$\gamma(ori) = \exp\left(-\frac{Grad_{averaged}(ori)^2}{2\sigma^2}\right) \quad (7)$$

In this case, σ corresponds to the minimum gradient between the four averaged gradients. On the Figure 8 are represented two simulated RGB-Z images (Mask1 and Mask2) reconstructed using the presented edge-directed algorithm. There is a significant reduction of structural artifacts for the Mask2. However, there are still many remaining.

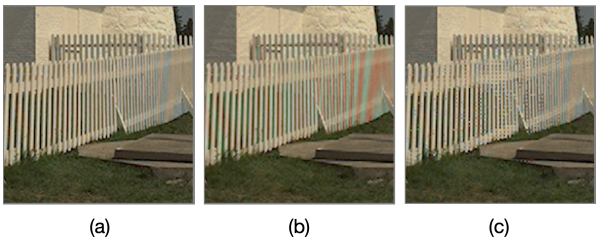


Fig. 8: Color images reconstructed from simulated RGB-Z data using the edge directed solution using central symmetry compared to the reference image. RGB-Z data are simulated using both Mask1 (section 4a) and Mask2 (section 4b). (a) Reference cropped image. (b) Simulated RGB-Z image reconstructed based on the Mask1. (c) Simulated RGB-Z image reconstructed based on the Mask2.

4.2.2 Gradients computed along the kernel axes

A second edge-directed algorithm solution has been implemented. It is a weighted-sum based interpolation using the same three factors (distance factor (3), intensity factor (4) and edge-directed factor (7)). The main difference concerns the edge-directed factor computation. In the previous algorithm, the gradients are computed using a central symmetry with the missing pixel, while in this algorithm the gradients are computed using the axes of symmetry of the kernel.
The method is inspired from the Prewitt edge-detection operator [2]. The Prewitt operator is only sensitive to the vertical and horizontal orientations. It is not sensitive to the diagonal orientations. To address this issue, we propose a gradient computation along the diagonal orientations based on the diagonal axes. The gradients are then 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. The Figure 9 represents the four masks used to compute the gradients. The relationships between the three channels based on the constant color difference assumption are used to compute inter-channel differences. The computation of the four gradients is detailed in the equations 8 to 11. Pixel indices refer to the Figure 6.

$$Grad_{vertical} = \sum_{m\in[-2;2]} [|I_{\eta\neq3}(-2,m) - I_{\eta\neq3}(+2,m)| + |I_{\eta\neq3}(-1,m) - I_{\eta\neq3}(+1,m)|] \quad (8)$$

$$Grad_{horizontal} = \sum_{m\in[-2;2]} [|I_{\eta\neq3}(m,-2) - I_{\eta\neq3}(m,+2)| + |I_{\eta\neq3}(m-1) - I_{\eta\neq3}(m,+1)|] \quad (9)$$

$$Grad_{diagonal1} = \sum_{m\in[-2;1]} \sum_{k\in[0;1]} [|I_{\eta\neq3}(m+1+k,m-1-k) - I_{\eta\neq3}(m-1-k,m+1+k)| + |I_{\eta\neq3}(m+1+k,m-k) - I_{\eta\neq3}(m-k,m+1+k)|] \quad (10)$$

$$Grad_{diagonal2} = \sum_{m\in[-2;1]} \sum_{k\in[0;1]} [|I_{\eta\neq3}(m+1+k,1+k-m) - I_{\eta\neq3}(m-1-k,-m-1-k)| + |I_{\eta\neq3}(m+1+k,k-m) - I_{\eta\neq3}(m-k,-m-1-k)|] \quad (11)$$

Where, $I(i,j)$ represents the intensity value of a pixel (i,j) and η corresponds to the color channel of a given pixel. Namely, $\eta = 0$ corresponds to red (R) channel,

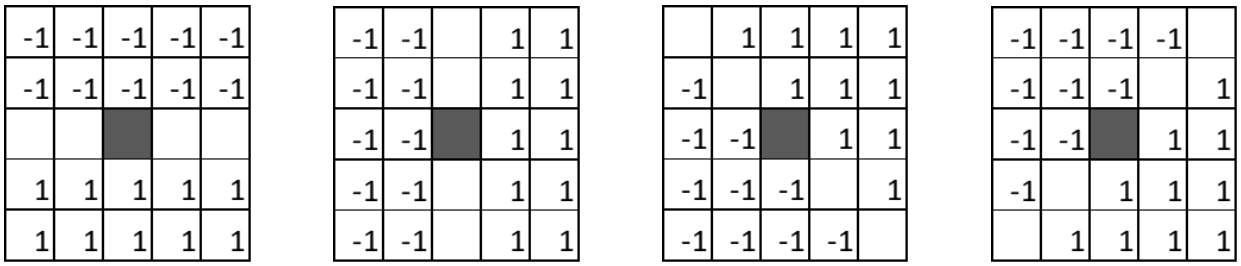


Fig. 9: General computation masks of the centered on the missing pixel, for the four orientations of the gradient. (a) Vertical gradient. (b) Horizontal gradient. (c) and (d) two diagonal gradients.

$\eta = 1$ corresponds to the green (G) channel, $\eta = 2$ corresponds to the blue (B) channel, and $\eta = 3$ correspond to the Z channel. According to the Z-pixel distribution across the kernel, the mask used to compute a gradient can vary. It depends on the kernel configuration: only the color pixels are used to calculate the gradient. The masks can be adjusted according to the current RGB-Z architecture. The Z-pixel values are not taken into account, so the differences including a Z-pixel are not calculated. The number of differences calculated for each orientation can be different. The four gradients are scaled following their number of differences used to compute them. The steps to achieve the interpolation are similar the previous sections: the weights are computed using equation 6, they are normalized with equation 12 and finally the missing color pixel $I_{\eta \neq 3}(p, q)$ is interpolated using a weighted averaged summation of the same color channel pixels. For example to interpolate a missing red pixel (respectively η is adapted following the color channel reconstructed), we use the equation 13.

$$\omega'(i,j) = \omega(i,j) / \sum_{(g,h)\in\xi} \omega(g,h) \tag{12}$$

$$I_{\eta=0}(p,q) = \sum_{(g,h)\in\xi} \omega'(g,h) I_{\eta=0}(g,h) \tag{13}$$

On the Figure 10 are represented two simulated RGB-Z images (Mask1 and Mask2) reconstructed with the presented edge-directed algorithm. They are also compared to the images reconstructed with the bilateral and EDI methods presented above. A reduction of structural artifacts is again observed on the Mask2. The pattern of the RGB-Z architecture remains however clearly visible. About the Mask1, there is no noticeable improvement between the different solutions proposed. Aliasing and false color artifacts are still present.

Fig. 10: Color images reconstructed from simulated RGB-Z data using the edge directed solution using kernel axes presented compared to the reference image. RGB-Z data are simulated using both Mask1 (section 4a) and Mask2 (section 4b). (a) Reference cropped image. (b) Simulated RGB-Z image reconstructed based on the Mask1. (c) Simulated RGB-Z image reconstructed based on the Mask2.

4.3 CFA reconstruction by a semi-gradient approach

The reconstruction methods using EDI presented in the previous sections still have some remaining artifacts. These artifacts are mainly present along the thin edges. A study is first performed for the Mask1 in order to understand the cause of these artifacts and thus to propose a solution that corrects them.

4.3.1 Artifacts study for the Mask1

In order to better understand the artifacts present in an image simulated with the Mask1, it is necessary to look back at the RGB-Z matrix and more precisely at the computational kernel level. On the Figure 11, we can see that the amount of green information available to reconstruct the missing pixel is limited. Indeed, there are only four green pixels in the considered neighborhood. Moreover, the distribution of these pixels does not allow a purely horizontal or purely vertical interpolation. This effect is even more visible when the reconstructed pixel is located along a narrow edge. A visual

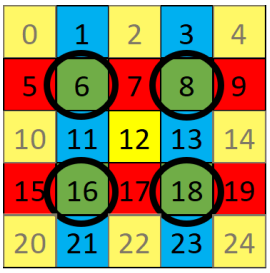


Fig. 11: The 5x5 computational kernel for the Mask1 centered on the missing pixel (pixel number 12). Black circle show the four green pixels used to interpolate the central missing pixel.

example is given on the Figure 12. On this example we can see some false colors artifacts along the vertical edges of the fork. There is a reconstruction error despite the fact that the gradients correctly identify the vertical edge. This error is due to the bad distribution of green pixels in the neighborhood of the missing pixel.

In this case, the four green pixels belong to two different textures of the image. The two left pixels belong to the white fork area and the two right pixels belong to the background area. Moreover, they are not on the vertical axis of the kernel which corresponds, in this case, to the priority axis of interpolation. This is why, during the interpolation the weight distribution is homogeneous between the four pixels (they all are equal to 1/4). In order to solve this issue, we propose to estimate the belonging of the missing pixel between the different textures of the image.

4.3.2 Proposed solution for the Mask1

In order to estimate on which texture the missing pixel belongs among the different potential areas present in a given kernel, we propose to implement a new operator that we call *semi-gradient*. As we do not have color information directly available at the location of the missing pixel, we study its closest neighbors. The objective is to study the local variations of the texture inside the computational kernel. The developed operator corresponds to the calculation of a local gradient in the kernel, while the gradient studies the texture variations through the kernel.

The SG operator is specifically designed to work on a Bayer based architecture. In the same vein as the calculation of the gradient presented in section 4.2.2, the SG is calculated by considering horizontal and vertical symmetry axes of the kernel. The idea is to compare the

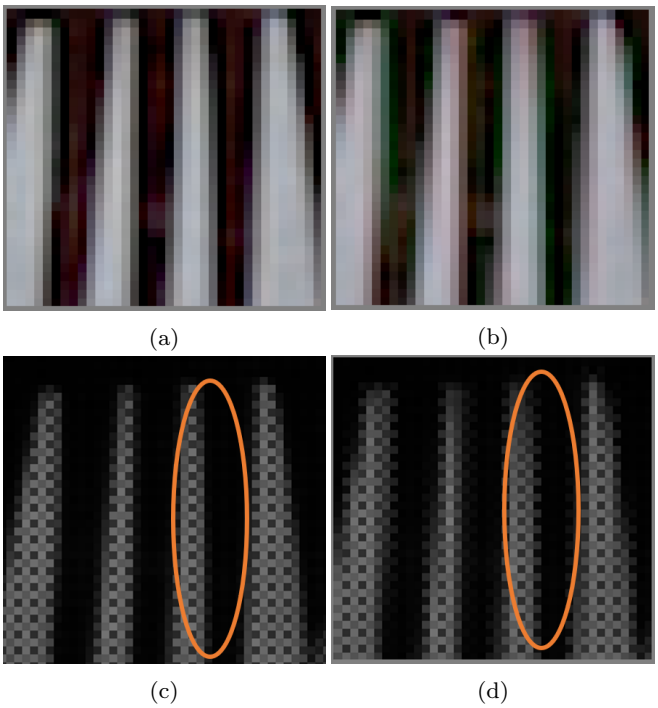


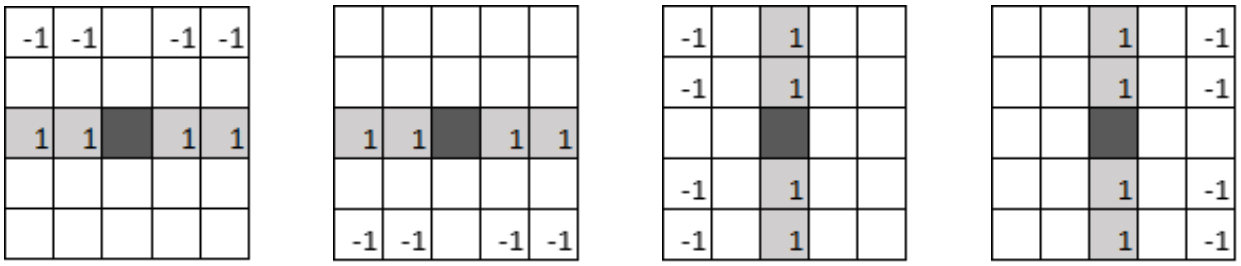
Fig. 12: Example of reconstructed Bayer images from the Mask1 and their corresponding demosaiced images. (a) Reference color image. (b) Reconstructed color image using EDI section 4.2.2. (c) Reference CFA that corresponds to the reference color image. (d) Reconstructed CFA using algorithm presented in section 4.2.2. The orange circles show the pixel reconstruction errors.

differences between the pixels close to the missing pixel and pixels at the kernel borders. If the tested pixels are close, in terms of pixel intensity, to a specific area of the kernel, then the weights of the pixels in this area are valued. Two arrays of eight values are computed corresponding to eight vertical and eight horizontal differences (equations 14, 15, 16, 17). They are computed using masks shown in Figure 13.

$$diff_{vert}[2j] = |I_{\eta_1}(0, j-2) - I_{\eta_2}(-2, j-2)|, \quad with\ \eta_1 = \eta_2 \neq \eta_3 \quad (14)$$

$$diff_{vert}[2j+1] = |I_{\eta_1}(0, j-2) - I_{\eta_2}(2, j-2)|, \quad with\ \eta_1 = \eta_2 \neq \eta_3 \quad (15)$$

$$diff_{hori}[2j] = |I_{\eta_1}(j-2, 0) - I_{\eta_2}(j-2, -2)|, \quad with\ \eta_1 = \eta_2 \neq \eta_3 \quad (16)$$



(a) (b) (c) (d)

Fig. 13: Differences computation masks used to compute SGs. Close pixels are marked in Grey. The missing color pixel is the darkest. (a) Comparison between close pixels and the top of the kernel. (b) Comparison between close pixels and the bottom of the kernel. (c) Comparison between close pixels and the left part of the kernel. (d) Comparison between close pixel and the right part of the kernel. (a) and (b) correspond to the vertical differences. (c) and (d) correspond to the horizontal differences

$$diff_{hori}[2j+1] = |I_{\eta_1}(j-2,0) - I_{\eta_2}(j-2,2)|, \quad with\ \eta_1 = \eta_2 \neq \eta_3 \quad (17)$$

The index $j \in [0,4]$ is the index of both arrays. Horizontal and vertical differences are associated to form the SGs. A total of eight SGs SG are computed (equations 18), two for each orientation. To facilitate the use of SGs, we attribute cardinal points to them: North, South, East, West.

$$SG_N = \sum_{j\in[0;2;6;8]} diff_{vert}[j] \quad (18a)$$

$$SG_S = \sum_{j\in[1;3;7;9]} diff_{vert}[j] \quad (18b)$$

$$SG_W = \sum_{j\in[0;2;6;8]} diff_{hori}[j] \quad (18c)$$

$$SG_E = \sum_{j\in[1;3;7;9]} diff_{hori}[j] \quad (18d)$$

$$SG_{NW} = \sum_{j\in[0;2]} diff_{vert}[j] + diff_{hori}[j] \quad (18e)$$

$$SG_{SE} = \sum_{j\in[7;9]} diff_{vert}[j] + diff_{hori}[j] \quad (18f)$$

$$SG_{NE} = \sum_{j\in[6;8]} diff_{vert}[j] + \sum_{j\in[1;3]} diff_{hori}[j] \quad (18g)$$

$$SG_{SW} = \sum_{j\in[1;3]} diff_{vert}[j] + \sum_{j\in[4;6]} diff_{hori}[j] \quad (18h)$$

They are then discriminated using equation 7.
Finally the weights are computed using only the SGs. There are only four weighted pixels used for the interpolation, while there are eight SGs. Each weight is therefore the linear combination of three SGs. For example, the weight of the pixel number 6, that is located to the top right area of the kernel (Figure 11), is computed by the linear combination of SG_N, SG_W and SG_{NW}. The four weights are computed with equations 19.

$$\omega_6 = SG_N + SG_W + SG_{NW} \quad (19a)$$

$$\omega_8 = SG_N + SG_E + SG_{NE} \quad (19b)$$

$$\omega_{16} = SG_S + SG_W + SG_{SW} \quad (19c)$$

$$\omega_{18} = SG_S + SG_E + SG_{SE} \quad (19d)$$

It is interesting to note that in the case of Mask1, gradients are no longer used to calculate weights. Only SGs are considered. The particular distribution of green pixels useful for interpolation does not allow direct interpolation along a direction. However, the SGs study the belonging of the central pixel and its neighborhood compared to the rest of the kernel. They thus allow to promote some areas in the kernel. An example of an image reconstructed using the SG method is given in Figure 14.c and its corresponding CFA Figure 14.f. The edges of the fork are sharper and there is also a reduction of false color artifacts along the edges. The flowchart of the algorithm presented in this section is described in Figure 15.
The numbers of operations required to reconstruct a missing pixel with the presented methods and for both masks are given in the Table 1.

5 Simulation chain

The evaluation of the algorithms was done in simulation because real data acquired with a RGB-Z test chip are not yet available. The test environment, below named simulation chain, used to evaluate the different algorithms is composed of several elements.

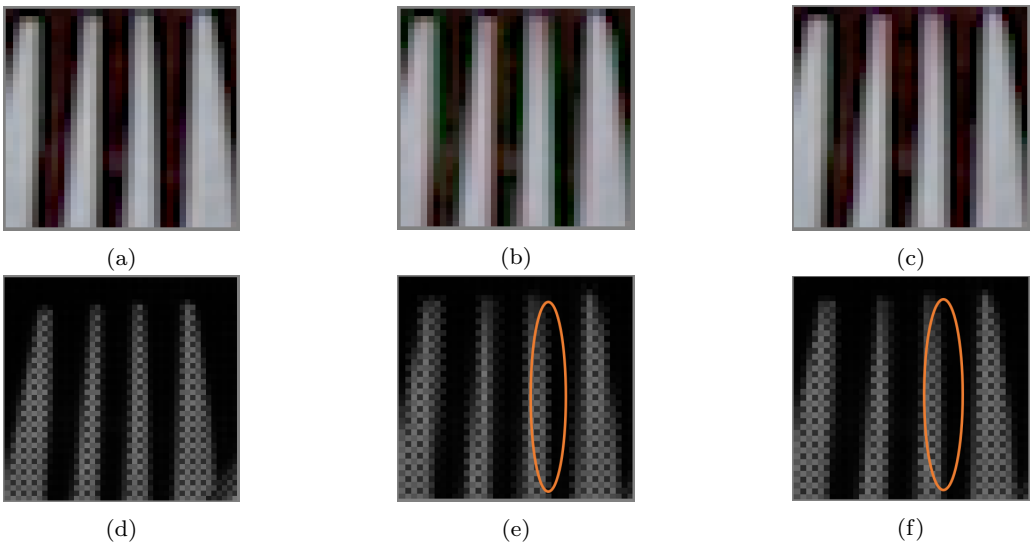


Fig. 14: Example of reconstructed Bayer images from the Mask1 and their corresponding demosaiced image. (a) Reference color image. (b) Reconstructed color image using EDI section 4.2.2. (c) Reconstructed color image using SGs. (d) Reference CFA that corresponds to the reference color image. (e) Reconstructed CFA using algorithm presented in section 4.2.2. (f) Reconstructed CFA using SGs. The orange circles show the improvement of the pixel reconstruction between the two solutions.

Type of operations	BI Mask1	BI Mask2	EDICentral Mask1	EDICentral Mask2	EDILine Mask1	EDILine Mask2	SG Mask1	SG Mask2
Adds and subbs	32	128/242	26	42/51	68	88/97	72	133/142
Multiplications	12	32/44	16	20/23	16	20/23	28	55/61
Divisions	4	8/11	16	24/30	16	24/30	28	36/39
Exponentials	0	0/0	4	4/4	4	4/4	8	12/12

Table 1: Number of operations to reconstruct one missing pixel for each method and both masks. They are two different cases for the Mask2: green pixel reconstruction and red/blue pixel reconstruction. For the first case 11 pixels are used to calculate the weighted sum while for red/blue pixel reconstruction, only 8 pixels are used to calculate the weighted sum. This is why the number of operations varies.

5.1 Simulation Chain description

The RGB-Z sensor model used to simulate the images that would have resulted from a real RGB-Z sensor is based on two sensor models. The first one, called color sensor model, simulates color pixels, while the second one, called i-ToF sensor model, simulates both depth pixels and the IR active illumination source. A RGB-Z ISP model is then used to reconstruct the color image from the CDFA raw image. The color sensor model and the i-ToF sensor model are firstly presented to introduce the RGB-Z sensor model. Then, the RGB-Z ISP model based on two ISP models is presented. In the same way as the RGB-Z sensor model, the first ISP model is used to reconstructed the depth map and the second one is used to reconstruct the full color image. The whole RGB-Z simulation chain is detailed in the Figure 16.

A reference image is also generated with the model. This corresponds to the green path in the Figure 16. The reference image is used to evaluate the reconstructed images. The color pixels located at the Z-pixels location in the reference image represent to the ground truth of the pixels to reconstruct.

5.1.1 Color Sensor Model

The color sensor model is used to simulate the acquisition of a CFA image. It is inspired from [4]. It takes as input a RGB image or a multispectral image set coded on 8 or 16 bits per channel. The output is a raw, noisy image patterned with the selected CFA coded on 11 bits. A noise model [4] is applied to simulate the photon-shot noise and the fix pattern noise. Different characteristics of the pixel and the pixel matrix are taken into

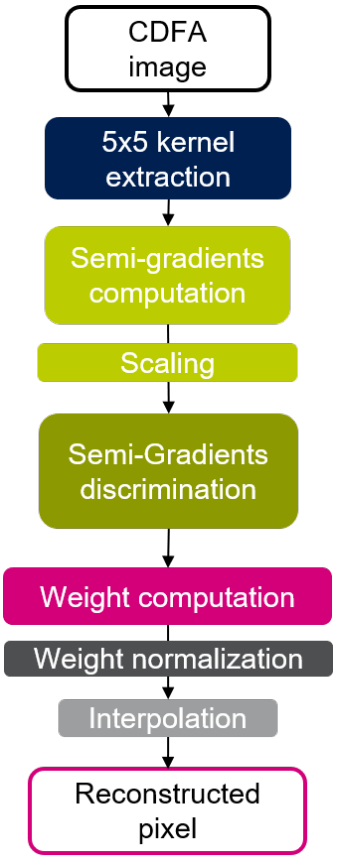


Fig. 15: Flowchart of a pixel reconstruction with the solution using the SG for the Mask1.

account in the model.

5.1.2 i-ToF Sensor Model

The sensor model that is used to simulate the acquisition of an i-ToF sensor takes as input a disparity map and the maximum theoretical distance corresponding to this map. The simulate Z-pixel is composed of 4 shutters (taps) that allow the acquisition of photons out of phase with a phase delay of 90 degrees from each other. The i-ToF Sensor Model takes into account internal and external parameters to compute a multidimensional array that represents the probability distribution of generated electrons for each shutter. Information about the active laser source, such as wavelength, modulation frequency and power are external parameters of this model. It is also necessary to have information about the IR reflectance of the objects in the scene. The electrons maps of the four shutters are then augmented by the application of different noise sources such as photon shot noise, dark current, or fixed pattern noise. Finally they are converted into an output array coded on 11 bits.

5.1.3 RGB-Z matrix build function

A RGB-Z sensor model that combines the two previous sensor models is used to generate the raw RGB-Z data. Both color and i-ToF sensor models are used in parallel to simulate a raw RGB-Z image. However it is necessary to make several acquisitions for the i-ToF sensor model in order to generate a depth map. Indeed, a set of three frequencies is used in order to avoid phase wrapping phenomena and to improve the maximum measurement distance. Once the CFA image and the shutter array of electrons are generated, they are mixed together to form a multidimensional RGB-Z matrix. The color information at the Z-pixel location is therefore removed and vice versa. It is called the *RGB-Z matrix build function*. That is during this step that the RGB-Z patterns studied are implemented.

5.1.4 RGB-Z ISP model

Once the RGB-Z data are simulated, the ISP models reconstruct depth and amplitude information from the Z-pixels and the full color image from color pixels. Concerning the ISP i-ToF, the filter kernel used for the denoising step is adapted to the RGB-Z matrix. The original filter was a 3x3 average filter. It is replaced by an average filter that takes into account the architecture of the matrix. It is now computed with a 5x5 kernel with the distribution represented in Figure 17.

The other functions of the ISP are not modified: the depth calculation is done only on the depth pixels. The missing depth information at the color pixels location is not reconstructed.
The color ISP is also modified in order to reconstruct missing color information. The algorithms presented in the previous chapter are included in the processing chain. However, this reconstruction step should not be placed anywhere in the chain. Indeed, it is imperative to reconstruct the missing information before the demosaicing step. In the same way, as we use an automatic white balance, it is preferable to have a complete color pixel matrix available for this algorithm. On the other hand, in order to avoid using defect pixels for the color pixel reconstruction, we apply the defect correction algorithms before the reconstruction of the missing color pixels. The modified ISP chain is shown in Figure 18.

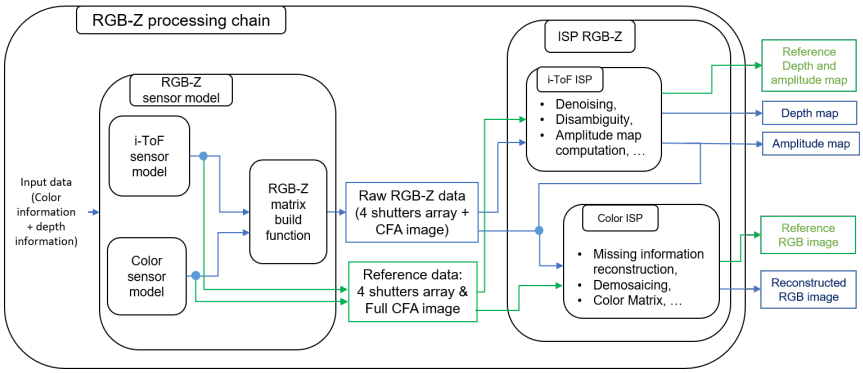
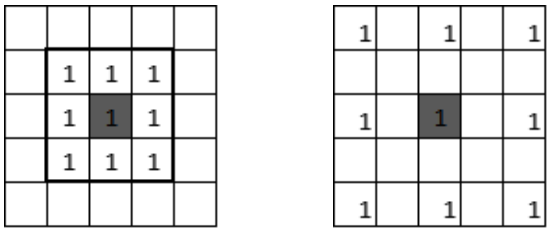


Fig. 16: Complete RGB-Z simulation chain. The RGB-Z Sensor Model generates the pixel arrays; the RGB-Z ISP reconstructs the color image and depth information. The blue path represents RGB-Z data. The green path represents the reference data.



(a) (b)

Fig. 17: Filter kernel used for noise reduction during ISP i-ToF step. (a) Original kernel. (b) Kernel used for the RGB-Z matrix.

5.2 Databases

Several databases with different features are used to evaluate the quality of the reconstructed images. It is important to evaluate the algorithms on several images. It allows to reduce the effect that a particular image could have. To this intent, we use databases with varying characteristics. For example, different resolutions are studied, different frequency contents.

Kodak [1], McMaster [33] and HDR+ [25] burst are used to study color reconstruction without the use of depth pixel information. These three databases have different frequency content and resolutions.

5.2.1 Kodak and McMaster datasets

The Kodak database [1] is wildly used in demosaicing papers. It contains twenty-four color images, of spatial size 768×512, which have been digitized by scanner after being captured on film. This dataset was criticized for having statistics very different from the ones of natural images [33]. This led to the creation of the McMaster dataset, which contains eighteen 500×500 crops of 2310×1814 images. The high resolution of the original McMaster images is an advantage of McMaster over Kodak, as the sensor images have a high resolution too. In comparison, they have more saturated colors than Kodak and contain many sharp structures with abrupt color transitions.

5.2.2 HDR+ burst dataset

In addition to the McMaster and Kodak databases, the HDR+ burst dataset [25] constituted of real images acquired by CFA sensors is used. It consists of 3640 images acquired with different CMOS imaging sensors from Android mobile phones. This database is built to study HDR reconstruction techniques. The images are generally 12-13 Mpixels, depending on the type of camera used for capture. Not all the images in the database are used in this work. An arbitrarily selection of twelve images is made. In addition, the selected images are downsampled in order to improve the computational time. Furthermore, by down sampling the images, we also reduce the distance between structures, thus, we increase the frequency content in the image. Since the most complex structures to reconstruct are located in high frequencies areas, it is also interesting for the assessment of the reconstruction algorithms. The images from the HDR+ burst are down sampled by a factor 2 horizontally and vertically.

5.3 Results

In this section, we present the results obtained with the algorithms presented in 4. Reconstructed images are compared to their corresponding reference image at two different levels of the processing chain presented in Figure 18. Both reference and reconstructed images

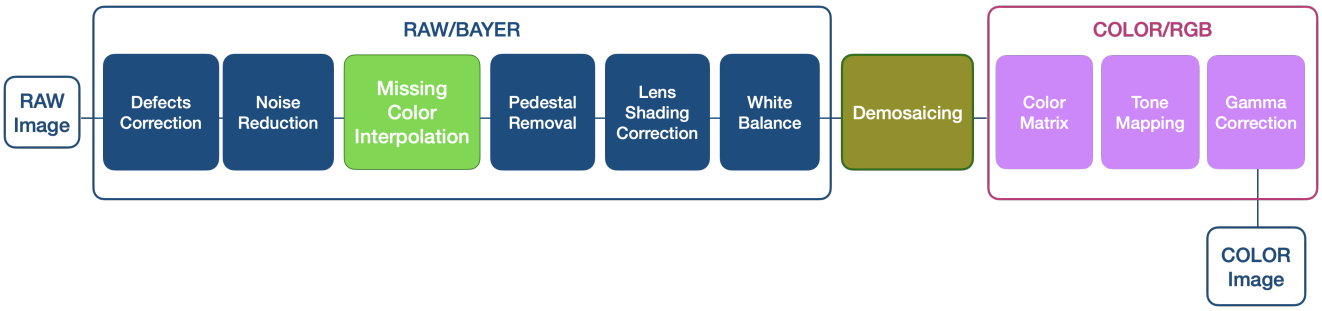


Fig. 18: Typical ISP processing chain adapted to a RGB-Z sensor.

Database	Year	Total Images	Number of images used	Image resolution (used)	Multi-spectral acquisition	Depth map
Kodak [1]	1991	24	24	768×512	No	No
McMaster [33]	2011	18	18	500×500	No	No
HDR+ burst [25]	2016	3640	12	4000×3000 (2000×1500)	No	No

Table 2: Overview of databases used in this work.

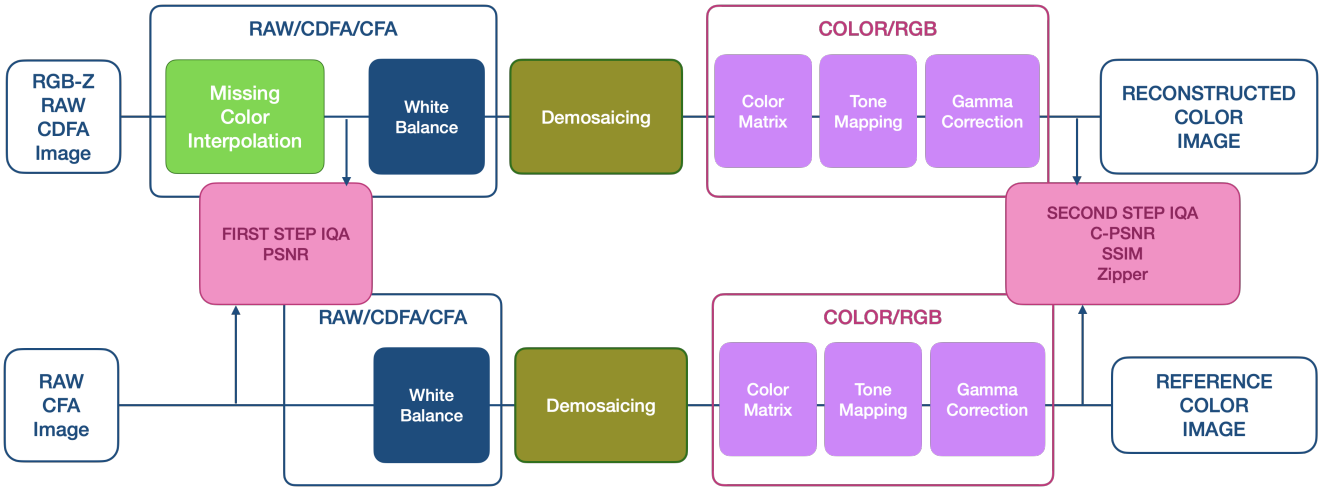


Fig. 19: Diagram of two simplified ISPs, one for the reconstruction of the RGB-Z image (top) and the other for the reconstruction of the color image (bottom). Both pink boxes indicate the two steps of image quality assessment (IQA). The simulated input images are 11-bit per value images acquired with one exposure.

are generated using the same processing chain. In this way, only the reconstruction step of the missing information is evaluated. The parameters of the simulation of the raw RGB-Z and raw RGB images are identical, only the generation of the RGB-Z pattern differs. In the same way for the reconstruction of the two color images, only the reconstruction step of the missing information is added compared to the reference image. Characteristics such as noise level, integration time, illuminant, pixels quantum efficiency, color and IR filters characteristics are common to both images. Only the RGB-Z matrix construction function and the missing information reconstruction algorithm are bypassed for the reference image. Figure 19 illustrates the two ISP chains executed on the reference image and the reconstructed RGB-Z image.

The first level of assessment is applied on the reconstructed CFA images. The PSNR[5] is computed on the reconstructed color pixels only. The PSNR is one of the most commonly used objective image quality metrics.

[5] Peak signal-to-noise-ratio

It is based on the pixel difference between two images: the reconstructed image is compared to the reference one. The computation is based on the Mean Squared Error (MSE) :

$$MSE = \frac{1}{NM}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}(I_d(m,n) - I_r(m,n))^2 \quad (20)$$

where, I_d and I_r represent respectively the pixel intensity of the reconstructed image and the reference image. M and N are the dimension of the given image. The PSNR is then computed by

$$PSNR = 10log_{10}\frac{s^2}{MSE} \quad (21)$$

where, s represents the maximum range of the image, it is defined by $s = 2^b - 1$, where b is the bit depth of the image. For a 8-bit image, $s = 255$.

The second level of assessment is then applied at the end of the ISP chain, on the reconstructed full color image. The C-PSNR[6], the SSIM[7] and M-SSIM[8], and the Zipper metric are used to quantify the reconstruction performance. The C-PSNR is used on the color images. It is computed using Color Mean Squared Error, MSE_C 22 and 23, with k that corresponds to the 3 color channels :

$$MSE_C = \frac{1}{3NM}\sum_{k=1}^{3}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}(I_d(m,n,k) - I_r(m,n,k))^2 \quad (22)$$

$$C - PSNR = 10log_{10}\frac{s^2}{MSE_C} \quad (23)$$

The PSNR and C-PSNR are not the most efficient method to discriminate structural deformations and it does not correlate well with the measure of perceived quality. Its major advantage is the simplicity of calculation.

The SSIM [30] is considered to be correlated with the quality perception of the Human Visual System (HVS). The idea is to measure the similarity of structure between two images, rather than a pixel-to-pixel difference as the PSNR does.

$$SSIM(d,r) = \left[l(d,r)^\alpha \times c(d,r)^\beta \times s(d,r)^\gamma\right] \quad (24)$$

[6] Color Peak signal-to-noise-ratio
[7] Structural Similarity index
[8] Multi scale Structural Similarity index

With,

$$l(d,r) = \frac{2\mu_d\mu_r + c_1}{\mu_d^2 + \mu_r^2 + c_1} \quad (25)$$

$$c(d,r) = \frac{2\sigma_d\sigma_r + c_2}{\sigma_d^2 + \sigma_r^2 + c_2} \quad (26)$$

$$s(d,r) = \frac{\sigma_{dr} + c_3}{\sigma_d + \sigma_r + c_3} \quad (27)$$

where, μ_d and μ_r are respectively the average of d and r, σ_d and σ_r are respectively the variance of d and r, σ_{dr} is the co-variance of d and r. The three variables c_1, c_2 and c_3 are used to stabilize the division with weak denominator. They are defined by $c_1 = (0.01 \times L)^2$, $c_2 = (0.03 \times L)^2$ and $c_3 = c_2/2$, with L represents the maximum range of the image, it is defined by $L = 2^b - 1$, where b is the bit depth of the image. Conventionally, to simplify the expression, we set $\alpha = \beta = \gamma = 1$. To evaluate the overall image quality, we compute the mean of each local window SSIM index.

The M-SSIM[31] is an improvement of the SSIM. It allows to have a different resolution evaluation of the image details. The system iteratively applies a low-pass filter and downsamples the filtered image by a factor of 2. The M-SSIM is computed using 28.

$$M - SSIM(d,r) = [l_M(d,r)]^{\alpha_M} \times \prod_{j=1}^{M}[c_j(d,r)]^{\beta_j} \times [s_j(d,r)^{\gamma_j}] \quad (28)$$

the number of iterations is fixed to 5 and the parameters are determined to be equal to $\beta_1 = \gamma_1 = 0.0448$, $\beta_2 = \gamma_2 = 0.2856$, $\beta_3 = \gamma_3 = 0.3001$, $\beta_4 = \gamma_4 = 0.2363$, and $\alpha_5 = \beta_5 = \gamma_5 = 0.1333$.
Both SSIM and M-SSIM metrics are decimal value between 0 and 1. An value equal to 1 means that the both image are identical and therefore indicates perfect structural similarity.

The zipper-metric[15] aims to detect zipper effects in a reconstructed image compared to the reference. For this purpose, the intensity differences between a pixel and its eight neighbors are studied in the reference image. For each pixel, we identify the neighbor with the smallest difference. We then compute the difference between the two same pixels in the reconstructed image corresponding to the smallest difference in the reference image. We finally study the variation between these two differences, in order to determine if there is a noticeable change between them. If it is confirmed, the pixel is considered to have a zipper effect.

To compute this metric, we use the CIELAB color space [7], a color space that is closer to human perception of color differences. It is specified by the CIE[9]. It is intended as a perceptually uniform space, where a given numerical change corresponds to similar perceived change in color. It is therefore necessary to convert the sRGB images into the CIELAB color space. Once the images are converted into the CIELAB color space, the zipper metric is computed. The difference between two pixels in the CIELAB color space is defined as the Euclidean distance between color values of the pixels, noted ΔE^*_{ab} 29.

$$\Delta E^*_{ab} = \sqrt{(L^*_2 - L^*_1)^2 + (a^*_2 - a^*_1)^2 + (b^*_2 - b^*_1)^2} \quad (29)$$

where, (L^*_1, a^*_1, b^*_1) and (L^*_2, a^*_2, b^*_2) are the CIELAB color values of the two pixels. The pixel I corresponding to the smallest difference between the eight neighbors of a given pixel is given by 30.

$$I = argmin_{i \in \epsilon} \Delta E^*_{ab}(P, i) \quad (30)$$

the variation between the two differences computed with the same two pixels in the reference image, and the reconstructed image is given by 31.

$$\psi = \Delta \tilde{E}^*_{ab}(P, I) - \Delta E^*_{ab}(P, I) \quad (31)$$

$\Delta \tilde{E}^*_{ab}(P, I)$ corresponds to ΔE^*_{ab} between the two considered pixels in the reconstructed image.
A pixel is considered to have a noticeable change in color difference when $\mid \psi \mid > \delta$. The threshold δ is set to 2.3 according to [17]. The final score of the zipper metric corresponds to the percentage of pixels in the reconstructed image for which a noticeable change is noticed compared to the reference one. This is why the higher the score, the lower the quality of the reconstructed image.

The experimental results are presented according to the RGB-Z pattern considered. The results of the reconstruction methods applied to the Mask1 are first presented, then the results of the images reconstruction of the Mask2 are analyzed. Full description of the reconstruction method applied to the Mask2 can be found in [23].
In the following sections, the color pixel reconstruction algorithms are abbreviated. The bilateral based algorithm (Section 4.1) is indicated as *BI*. The edge directed algorithm calculating the gradients by a central symmetry around the missing pixel (Section 4.2.1) is indicated as *EDICentral*. The edge directed algorithm calculating the gradients along the kernel axes s (Section 4.2.2) is indicated as *EDILine*. Finally, the solution using the Semi-gradients (Section 4.3) is indicated as *SG*.

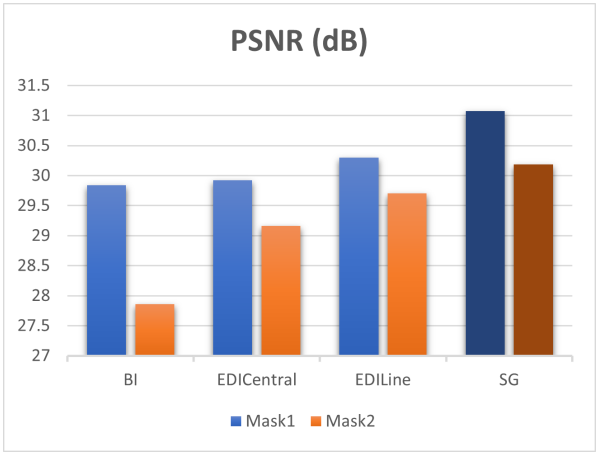
[9] stands for International Commission on Illumination. It is the international authority on light, illumination, color, and color spaces

5.3.1 Quality assessment of the Mask1

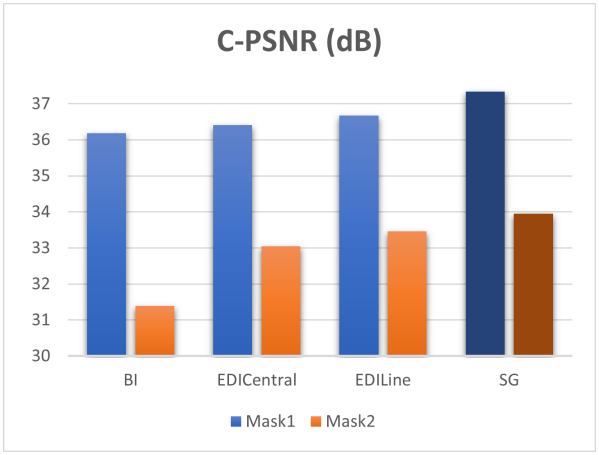
The methodology presented in the previous sections is applied to evaluate the reconstruction quality for the Mask1 (Figure 4a). The results for the Kodak, McMaster and HDR+ burst databases are presented in the Figure 20. We notice that the standard deviation for a given method is relatively high compared to the differences between the four methods. This is mainly due to the variations of the content between the images in the same database. However, it is interesting to note a general trend: the solution using the semi-gradients *SG* has better performances than the *EDILine*, which has better performances than the *EDICentral*. We can also notice that *BI* and *EDICentral* solutions have very similar results. This is mainly due to the fact that the distribution of green pixels useful for interpolation does not allow direct interpolation in all directions. Therefore, despite the use of edge directed methods, it is common that the weights distribution is homogeneous between the four pixels. In this case, the EDI interpolation is similar to the *BI* interpolation. The semi-gradients are designed to compensate this limitation, to improve the weights computation and consequently, the reconstruction of the missing pixel. On average, the semi-gradient *SG* method improves the PSNR of the reconstructed pixels in the CFA image by 1 dB compared to *EDILine*. Similarly, the overall C-PSNR on the color image is improved when the reconstruction of the missing pixels is performed by the semi-gradient method.

5.3.2 Quality assessment of the Mask2

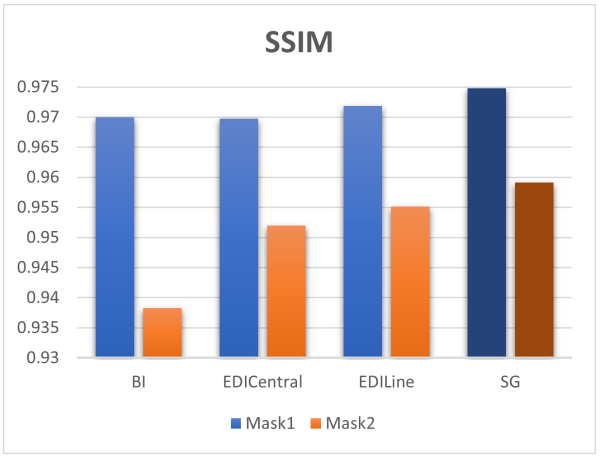
The same evaluation as in the previous section is performed for the Mask2 (Figure 4b). The results are summarized in the Figure 20 for the Kodak, McMaster and HDR+ burst databases. In the same way as the 1x1 Z-Pixel mask, the general trend is similar: the semi-gradient based method shows the better performances. However, we can note that the metrics scores applied on the color images obtained for Mask2 are in average lower than the scores obtained for Mask1, while the PSNR computed on the reconstructed pixels using the CFA images are closer. This suggests that is inherently more difficult to fully reconstruct a color image from a RGB-Z CDFA image based on the Mask2 compared to the Mask1. The semi-gradient based algorithm obtains, on average, a score of 30.60 dB when used to reconstruct



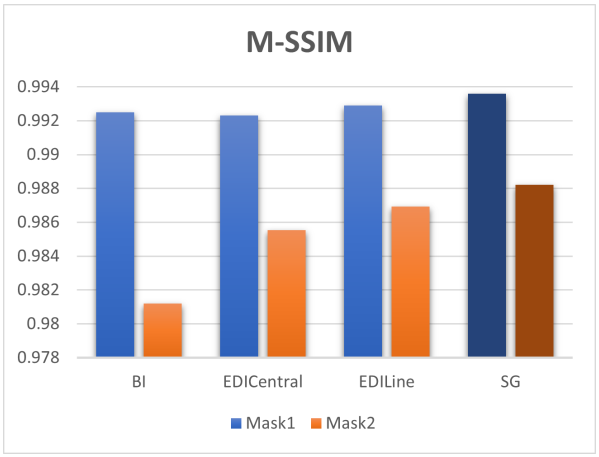
(a)



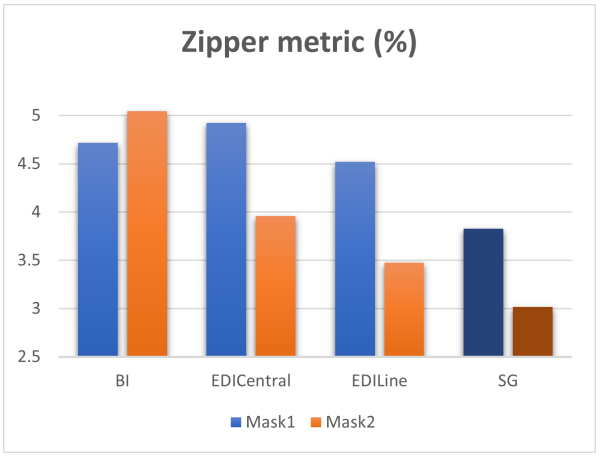
(b)



(c)



(d)



(e)

Fig. 20: Averaged performances of reconstruction methods in terms of (a) PSNR applied on the CFA image and (b) C-PSNR, (c) SSIM, (d) M-SSIM, and (e) Zipper Metric applied on the color image. Results are computed on the Kodak database [1], the McMaster database [33] and the HDR+ burst [25] for the Mask1 and the Mask2. For each mask, the best solution is highlighted in dark.

a RGB-Z architecture based on the Mask1, while it obtains a score of 30.20 for the 2x2 Z-Pixel mask. Those results are relatively close compared to the difference in term of C-PSNR between the two Masks. On average

Fig. 21: Crop of reconstructed KodaK9 images from the Mask1 and Mask2 using the four reconstruction methods.

Mask1	BI	EDICentral	EDILine	SG
PSNR	24.55	24.46	24.97	**26.9**
C-PSNR	30.64	30.81	31.02	**32.99**
SSIM	0.9595	0.9607	0.9636	**0.9751**
M-SSIM	0.9861	0.9861	0.9872	**0.9918**
Zipper metric	12.493	12.087	11.504	**8.591**
Mask2	BI	EDICentral	EDILine	SG
PSNR	22.93	24.83	25.13	**26.13**
C-PSNR	26.70	29.12	29.13	**30.31**
SSIM	0.9163	0.9565	0.9600	**0.9652**
M-SSIM	0.9676	0.9832	0.9835	**0.9868**
Zipper metric	9.810	5.420	5.434	**5.041**

Table 3: Results of the metrics for the kodak9 image using the four reconstruction algorithms applied on the Mask1 and the Mask2. The corresponding images are shown in figure 21. We can note that the zipper metric score is inversely proportional to the quality of the reconstructed image: the higher the score, the lower the quality of the reconstructed image. For each metric and for both masks, the SG method shows the best results.

the C-PSNR score of the semi-gradient based algorithm for the Mask1 is 36.61 dB whereas it is 33.95 dB for the 2x2 Z-Pixel mask.

One possible explanation concerns the demosaicing step. In the case of 2x2 Z-Pixel mask, the missing information represents a cluster of 2x2 color pixels at the Z-pixel location. An incorrect reconstruction of the four color pixels corresponding to a Z-pixel in the 2x2 Z-Pixel mask introduces an error that could strongly impact the demosaicing step. The Z-pixel pattern becomes visible. This effect is particularly visible on the images reconstructed with the *BI* algorithm.

An example is given in the Figure 21. The corresponding metrics results processed are given in the Table 3. The Z-pixel pattern is apparent when we compare the two reconstructed images using the Bilateral interpolation (Figure 21.b and Figure 21.f). Indeed, the artifacts corresponding to the missing information at the Z-pixels location are directly identifiable on the edges of the image in the case of Mask2. They remain visible with the *EDICentral*, *EDIline* and the *SG* solutions, but they are significantly reduced. The objective evaluation of the different methods has the same trend as the subjective analysis of the reconstructed images. In both cases, it is reported that the solution using semi-gradients reduces the artifacts in the final reconstructed image and improves the image quality.

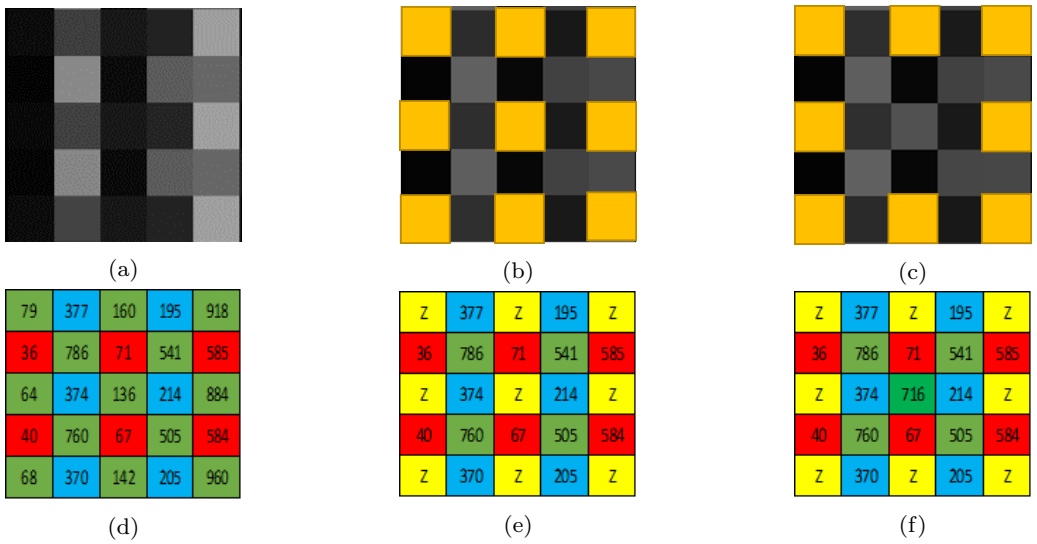


Fig. 22: Example of a narrow edge reconstruction error for Mask1 using the semi-gradient method. a) CFA reference and its corresponding values (d). (b) CDFA and its corresponding values (e). (c) CDFA after the interpolation of the missing pixel and its corresponding values (f). The real value of the missing pixel is 136 while the four green pixels in the neighborhood are all greater than 500. It is impossible to correctly interpolate the missing pixel value using a weighted sum of the four neighboring green pixels. The result of the interpolation is 716. The pixel values are coded on 11 bits. The yellow squares indicate the Z-pixels.

Fig. 23: Example of reconstruction errors along very thin edges. (a) Reference image. (b) Reconstructed image of the Mask1 using semi-gradient method. The red and green vertical stripes along the window are characteristic of too thin edges. The image is a crop from the HDR+ burst database [25].

6 Discussion

We have seen in the previous sections that the use of semi-gradients allows improving the quality of reconstructed images in the case of the studied RGB-Z architectures. However, even with the semi-gradients, some structures are not correctly reconstructed.

6.0.1 Mask1

Regarding the Mask1, for the same reasons explained in the section 4.3.1, it is difficult to reconstruct vertical and horizontal edges due to the absence of green pixel along these axes. Semi-gradients allow to overcome these limitations for some cases by studying the close neighborhood of the missing pixel in order to determine to which texture it belongs. This method works as long as the missing pixel is close to a neighboring texture. However, there are situations with sharp narrow edges with a one pixel width. In these situations, it may happen that the green pixels available for interpolating the missing pixel do not match the value of the missing pixel. A weighted sum of these pixels cannot then correctly reconstruct the missing pixel. The Figure 22 shows an example of a narrow edge that can not be properly reconstructed using semi-gradients.

In this example, we can notice that the semi-gradient based interpolation promotes the use of the green pixels located on the left side of the kernel. Indeed, the red pixels values neighboring the missing pixel are closer to the red pixels values located on the left side of the kernel. An example of erroneous reconstruction along very thin edges is given in Figure 23.

6.0.2 Mask2

Corners are structures that are difficult to reconstruct in the Mask2 architecture. In particular, there are reconstruction errors when a Z-pixel is precisely located on the corner. In a such case, there are often only two textures present in the 5x5 computation kernel: a first one that corresponds to the corner and the second one that corresponds to the background. An example is given on the Figure 24. The color pixels present on the bottom left edges of the calculation kernel belong to the background texture, and the missing color pixels belong to the corner texture that is located in the top right area. Background pixels are shaded on the figure. The two red pixels located on the northwest/southeast diagonal belong to the background texture. The gradient and semi-gradients along this axis are small and imply an interpolation along this one, while the interpolation should be along the southwest/northeast axis with a privileged interpolation along the north east corner of the kernel thanks to the semi-gradient. A known information at the missing pixel location could help to detect the corner properly and thus improve its reconstruction. This suggests that joint reconstruction could be relevant to improve corner reconstruction. Studied on real data, or with a relevant database, joint reconstruction may be an opportunity of improvement in the case of reconstruction based on the Mask2. An example of a reconstruction error of a corner is given in the Figure 25.

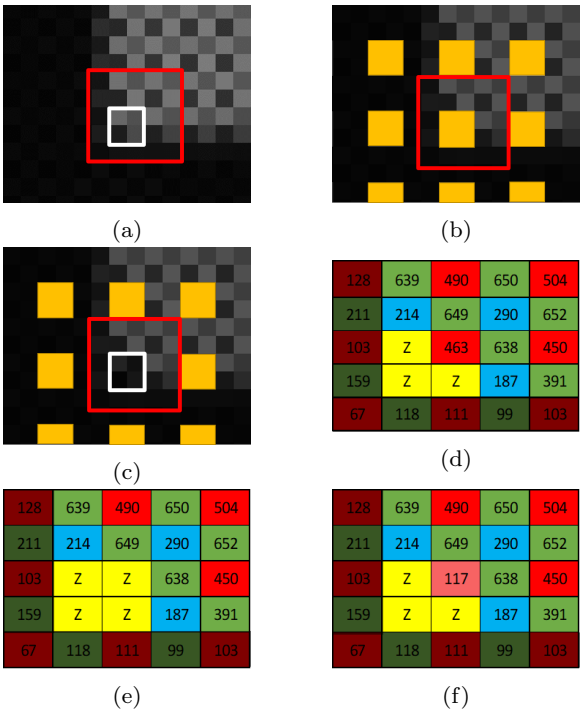


Fig. 24: Example corner reconstruction error for Mask2 using the semi-gradient method. The yellow squares indicate the Z-pixels, the white squares indicate the location of the four reconstructed color pixels, and the red square indicates the computational kernel to reconstruct the missing red pixel shown as example.(a) CFA reference and its corresponding values (d). (b) CDFA and its corresponding values (e). (c) CDFA after the interpolation of the missing pixel and its corresponding values (f). The shaded area represents the texture corresponding to the background. The pixel values are coded on 11 bits. The interpolation is made using the semi-gradients method.

7 Conclusion

In this paper, two monolithic RGB-Z sensor architectures have been introduced. They are based on the Bayer pattern. A method of reconstruction of the missing color information has been proposed for these arrays. This method uses a new operator called semi-gradient. Due to the lack of a mature technology, a simulation chain to generate RGB-Z raw data has been introduced as well as a processing chain to reconstruct the full color images. The experimental results have shown that our algorithm gives better images in terms of visual quality and objective quality measures, compared to the adapted methods from the state of the art. However, even with the semi-gradient method, some structures are not correctly reconstructed. An interesting way of improvement would be to explore solutions using the information acquired by the Z-pixels to reconstruct the missing color information. An other interesting future work would be to study the feasibility of a hardware implementation of our proposed solution in order to integrate it in an ISP chain.

Fig. 25: Example of reconstruction errors of a corner for Mask2 architecture. (a) Reference image. (b) Reconstructed image using semi-gradient method. The image is a crop from the CAVE database [32].

References

1. Kodak color image dataset. http://r0k.us/graphics/kodak/. URL http://r0k.us/graphics/kodak/
2. Adlakha, D., Adlakha, D., Tanwar, R.: Analytical Comparison between Sobel and Prewitt Edge Detection Techniques **7**(1), 1482–1485 (2016)
3. Alacoque, L.: Method for demosaicing a raw digital image, corresponding computer program and imaging or graphic circuit (number EP 2426639 B1) (2015)
4. Alakarhu, J.: Image Sensors and Image Quality in Mobile Phones. International Image Sensor Workshop, Ogunquit pp. 7–18 (2007)
5. Bevilacqua, A., Di Stefano, L., Azzari, P.: People tracking using a Time-of-Flight depth sensor. Proceedings - IEEE International Conference on Video and Signal Based Surveillance (2006). DOI 10.1109/AVSS.2006.92
6. Breitbarth, A., Schardt, T., Kind, C., Brinkmann, J., Dittrich, P.G., Notni, G.: Measurement accuracy and dependence on external influences of the iPhone X TrueDepth sensor (October), 7 (2019). DOI 10.1117/12.2530544
7. Fairchild, M.D.: Color Appearance Models (1997)
8. Gamal, A.E.: High Dynamic Range Image Sensors. International Solid-State Circuits Conference **290** (2002)
9. Gove, R.J.: CMOS image sensor technology advances for mobile devices. In: High Performance Silicon Imaging, 2 edn., pp. 185–240. Elsevier Ltd. (2020). DOI 10.1016/B978-0-08-102434-8.00007-6. URL http://dx.doi.org/10.1016/B978-0-08-102434-8.00007-6
10. Gretchen Alper: 1000 Frames per second cameras: fast frame rates with ultra high resolution industrial cameras (2015). URL https://www.adimec.com/1000-frames-per-second-cameras-fast-frame-rates-with-ultra-high-resolution-industrial-cameras/
11. Horé, A., Ziou, D.: An edge-sensing generic demosaicing algorithm with application to image resampling. IEEE Transactions on Image Processing **20**(11), 3136–3150 (2011). DOI 10.1109/TIP.2011.2159229
12. Kim, W., Yibing, W., Ovsiannikov, I., Lee, S., Park, Y., Chung, C., Fossum, E.: A 1.5Mpixel RGBZ CMOS image sensor for simultaneous color and range image capture. Digest of Technical Papers - IEEE International Solid-State Circuits Conference **55**, 392–393 (2012). DOI 10.1109/ISSCC.2012.6177061
13. Koyama, S., Inaba, Y., Kasano, M., Murata, T.: A Day and Night Vision MOS Imager With Robust **55**(3), 754–759 (2008)
14. Litomisky, K.: Consumer rgb-d cameras and their applications. Tech. rep. (2012). URL http://www.cs.ucr.edu/ klitomis/files/RGBD-intro.pdf%5Cnpapers3://publication/uuid/028500F0-C8D5-49D3-AA1B-5828D3D8D7AA
15. Lu, W., Tan, Y.P.: Color filter array demosaicking: new method and performance measures. IEEE transactions on image processing : a publication of the IEEE Signal Processing Society **12**(10), 1194–1210 (2003). DOI 10.1109/TIP.2003.816004
16. Maggie Tillman: What is Apple Face ID and how does it work? (2020). URL https://www.pocket-lint.com/phones/news/apple/142207-what-is-apple-face-id-and-how-does-it-work
17. Mahy, M., Eycken, L.V., Oosterlinck, A.: Evaluation of Uniform Color Spaces Developed after the Adoption of CIELAB and CIELUV. Color Research and Application **19**(2), 105–121 (1994)
18. Malvar, H., Li-wei He, Cutler, R.: High-quality linear interpolation for demosaicing of Bayer-patterned color images. 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing **3**, iii–485–8 (2004). DOI 10.1109/ICASSP.2004.1326587. URL http://ieeexplore.ieee.org/document/1326587/
19. Microsoft: Microsoft Kinect. URL https://www.xbox.com/en-US/kinect
20. Monno, Y., Teranaka, H., Yoshizaki, K., Tanaka, M., Okutomi, M.: Single-Sensor RGB-NIR Imaging: High-Quality System Design and Prototype Implementation. IEEE Sensors Journal **19**(2), 497–507 (2019). DOI 10.1109/JSEN.2018.2876774
21. Navinprashath, R.R., Radhesh, B.: Learning based demosaicing and color correction for RGB-IR patterned image sensors. Society for Imaging Science and Technology pp. 1–6 (2019). DOI https://doi.org/10.2352/ISSN.2470-1173.2019.15.AVM-045
22. Ramanath, R., E. Snyder, W.: Adaptive demosaicking. journal of electronic imaging **12**, 633–642 (2003). DOI 10.1117/1.1606459
23. Rebiere, V., Drouot, A., Granado, B., Bourge, A., Pinna, A.: Semi-Gradient for Color Pixel Reconstruction in a RGBZ CMOS Sensor. 2020 IEEE SENSORS pp. 6–9 (2020). DOI 10.1109/SENSORS47125.2020.9278887
24. Rodrigues, B., Guillon, M., Billon-pierron, N., Mancini, J.b., Giffard, B., Cazaux, Y., Malinge, P., Waltz, P., Ngoua, A., Taluy, A., Kuster, S., Joblot, S., Roy, F., Lu, G.n., Lyon, N.D., Lyon, B.: Indirect ToF Pixel integrating fast buried-channel transfer gates and gradual epitaxy , and enabling CDS. International Image Sensor Workshop pp. 266–269 (2017)
25. Samuel W. Hasinoff, Dillon Sharlet, Ryan Geiss, A.A., Levoy, Jonathan T. Barron, Florian Kainz, Jiawen Chen, M.L.: Burst photography for high dynamic range and low-light imaging on mobile cameras. ACM Transactions on Graphics **35** (2016). URL http://hdrplusdata.org/dataset.html
26. SHI, L., OVSIANNIKOV, I.: Demosaicing for RGBZ sensor (Nb US 2015/0022869 A1) (2015)
27. Shi, L., Ovsiannikov, I., Min, D.k., Noh, Y., Kim, W., Jung, S.: Demosaicing for RGBZ Sensor. SPIE Proceedings **8657**, 1–9 (2013). DOI 10.1117/12.2001702
28. Teranaka, H., Monno, Y., Tanaka, M., Ok, M.: Single-Sensor RGB and NIR Image Acquisition: Toward Optimal Performance by Taking Account of CFA Pattern, Demosaicking, and Color Correction. Electronic Imaging **2016**(18), 1–6 (2016). DOI 10.2352/ISSN.2470-1173.2016.18.DPMI-256
29. Tomasi, C., Manduchi, R.: Bilateral Filtering for Gray and Color Images. International Conference on Computer Vision (1998)
30. Wang, Z., Bovik, a.C., Sheikh, H.R., Simmoncelli, E.P.: Image quality assessment: form error visibility to structural similarity. Image Processing, IEEE Transactions on **13**(4), 600–612 (2004). DOI 10.1109/TIP.2003.819861
31. Wang, Z., Simoncelli, E.P., Bovik, A.C.: Multi-scale Structural Similarity For image Quality Assessment. IEEE Asilomar Conference on Signals, Systems and Computer **2**, 9–13 (2003)
32. Yasuma, F., Mitsunaga, T., Iso, D., Nayar, S.K.: Generalized Assorted Pixel Camera: Post-Capture Control of Resolution, Dynamic Range and Spectrum. Tech. rep. (2008)
33. Zhang, L., Wu, X., Buades, A., Li, X.: Color Demosaicking by Local Directional Interpolation and Nonlocal Adaptive Thresholding. Electronic Imaging pp. 1–29 (2011)