A RECURSIVE EDGE DETECTOR FOR COLOR FILTER ARRAY IMAGE

B.Magnier

IMT Mines Alès
6 av. de Clavière 30319 Alès, France

A. Aberkane, N. Gorrity

Audensiel, 88 r. du Dôme Pôle R&D
92100 Boulogne-Billancourt, France

ABSTRACT

Most of embedded cameras use a single sensor to capture images through a color filter. They produce special images with only one color component per pixel. Missing data are usually estimated through a demosaicking process, but this takes undesirable computation time and may generate undesirable color artifacts. Many embedded systems under real-time constraints could use this type of camera for purposes like edge detection. In this paper, a new edge detection method is proposed for the computation of partial derivative images. This algorithm is tested on a large set of synthetic images where edge ground truth is unquestionable. The exploitation of the raw data of images allows not only to drastically reduce computational time, but also to get edge detection results even more precise than using certain demosaicking-based methods.

Index Terms— Edge detection, Bayer CFA, rotated filter.

1. INTRODUCTION AND MOTIVATIONS

The most of digital devices such as smartphones and tablets embed single-CCD (CMOS) cameras. They use only one charge-coupled device, to which a Color Filter Array (CFA) is superposed [1]. CFAs are mosaics of color filters placed over each pixel sensor. Different configurations exist, the most widely used in literature remains the Bayer filter [2]. The disposition is arranged on a base of 2×2 patterns, favoring the green channel, with 50% of green data(G), 25% of red (R) and 25% of blue (B). Each pixel of the CFA image only contains one color component out of three (see Fig.1).

The inversion process of the subsampling corresponds to a demosaicking. This allows to visualize color images, so the missing data can be estimated, creating a fully-defined RGB color image. Many demosaicking algorithms have been presented in literature [1, 3, 4, **?**]. Despite their efficiency to produce visually satisfying images, their failure to reconstruct high-frequency data by interpolations is detrimental to low-level analysis like edge detection. This was shown by Losson et al. [5] and confirmed by Breier et al. [6]. Moreover, those algorithms inevitably consume additional computation time.

Aberkane et al. [7] showed that directly edge detection in those images is possible with a very close or even better precision than segmentation after demosaicking. They introduced two approaches to use the raw data of CFA images for edge detection: (i) Deriche-based Luminance (DL) technique estimates partial derivative images of luminance thanks to Deriche recursive filters [8], so that a gradient image is computed, (ii) Deriche-based Color (DC) derivative method also uses Deriche filters to compute partial derivatives of red, green and blue channels so that a structure tensor is applied [9]. Despite the best results offered by DL, methods computing luminance derivatives from Bayer CFA were not studied deeply enough. Aberkane et al. [7] showed the failure to detect isoluminant edges. Moreover, a raw image of luminance could be directly captured with a monochrome camera. As a Bayer CFA image offers color data, calculating color derivatives is more relevant. This paper presents a new algorithm for the computation of color partial derivatives directly from Bayer CFA image, with a low computational complexity. The proposed algorithm offers a better overall performance compared to edge detection schemes applied to democaicking image or to CFA image with fewer operations per pixel.

2. DERICHE FILTERS AND COLOR GRADIENT

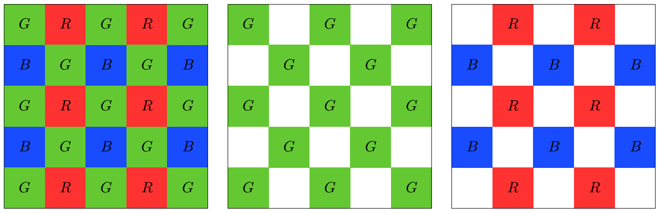
2.1. The recursive Deriche filter

Deriche [10] created an optimal edge detection filter according to Canny's criteria [11]: unicity, good detection and good localization. This filter can be implemented recursively allowing reducing computation time [8]. Deriche used a gradient approach based on two separable filters for the computation of partial derivative images: (i) a smoothing filter $s_\alpha(t)$ approximating a Gaussian function, (ii) a derivative filter $d_\alpha(t)$ which is the derivative of $s_\alpha(t)$:

$$s_\alpha(t) = m_s \cdot (1 + \alpha \cdot |t|)\, e^{-\alpha|t|} \tag{1}$$

$$d_\alpha(t) = -m_d \cdot \alpha^2 \cdot t \cdot e^{-\alpha \cdot |t|}, \tag{2}$$

where α represent the standard deviation of the filter and



(a) I^{CFA} CFA image (b) $\ddot{I}^{G}$ green channel (c) $\ddot{I}^{R,B}$, red and blue channels

Fig. 1: Decomposition of Bayer filter array.

(m_s, m_d) are two normalization parameters: $m_d = \frac{\left(1-e^{-\alpha}\right)^2}{\alpha^2 \cdot e^{-\alpha}}$ and $m_s = \frac{\left(1-e^{-\alpha}\right)^2}{1+2\cdot\alpha\cdot e^{-\alpha}-e^{-2\alpha}}$. Thus, the partial derivative image in x direction is obtained by: (i) applying the smoothing filter $s_\alpha(t)$ vertically on the original image, (ii) processing this smoothened image horizontally with the derivative filter $d_\alpha(t)$. Afterward, the partial derivative image in y direction can be obtained using the same process perpendicularly.

These filters can be implemented recursively [10]. Coefficients defining the deviation of both smoothing and derivative functions $s_\alpha(t)$ and $d_\alpha(t)$ depend on a single real positive parameter α. High values of α (α>1) enhance edge localization quality, whereas small values of α (α<1) enhance edge detection quality with poorer localization.

2.2. Structure Tensor: Di Zenzo gradient

A basic approach to compute a gradient of a color image is to combine first order derivatives issued of each channel. The best known color gradient was introduced by Di Zenzo [9] which uses the partial derivatives along x and y of each red, green and blue component. Those partial derivatives are gathered in two vectors $\mathbf{I}_x = (I_x^R, I_x^G, I_x^B)$ and $\mathbf{I}_y = (I_y^R, I_y^G, I_y^B)$. For each pixel, the gradient direction θ is chosen so the first fundamental form $d\mathbf{I}^2$ is maximized:

$$d\mathbf{I}^2(\theta) = f \cdot \cos^2(\theta) + 2 \cdot g \cdot \cos(\theta)\sin(\theta) + h \cdot \sin^2(\theta) \quad (3)$$

where $f{=}\mathbf{I}_x \cdot \mathbf{I}_x{=}\sum_{k=R,G,B}\left(I_x^k\right)^2$, $g{=}\mathbf{I}_x \cdot \mathbf{I}_y{=}\sum_{k=R,G,B}\left(I_x^k I_y^k\right)$ and $h{=}\mathbf{I}_y \cdot \mathbf{I}_y{=}\sum_{k=R,G,B}\left(I_y^k\right)^2$. This structure tensor provides in every pixel the gradient direction $\theta^* = \underset{\theta\in[-\pi,\pi]}{\operatorname{argmax}}\left(d\mathbf{I}^2(\theta)\right)$ and its tied norm $|\nabla\mathbf{I}| = |d\mathbf{I}^2\left(\theta^*\right)|$.

3. EDGE DETECTION DIRECTLY ON CFA IMAGES

Starting from Bayer [2] observation: green channel can be assimilated to luminance, red and blue channels to chrominance. SEDD (Smoothed component Estimation and Deriche Derivative) computes a gradient by estimating bilinear interpolations [7]. Here, we propose to decompose the CFA image into two channels $\ddot{I}^G$ and $\ddot{I}^{R,B}$ which both contain half of all pixels arranged in a quincunx lattice (see Figs.1(b) and 1(c)). The algorithm introduced in this paper process pixels in diagonal directions. We introduce a rotated Deriche filter directly adapted from eq. (1) and (2). As diagonal pixels are separated by a distance of $\sqrt{2}$ instead of 1 in usual directions (the gap between each pixel of a color channel in a CFA image), the parameter α must be modified so $\alpha' = \sqrt{2}\alpha$, [7].

Each channel $\ddot{I}^G$ and $\ddot{I}^{R,B}$ is diagonally smoothed using eq. 1. The diagonal derivatives images $I^G_{X,\alpha'}(x,y)$ and $I^{RB}_{X,\alpha'}(x,y)$ are obtained by filtering the smoothed images using eq. 2 in the perpendicular direction. Fig. 2 shows the rotation process for compute those diagonal derivatives.

Those partial derivatives must be combined as with the structure tensor using horizontal and vertical partial derivatives. The rotated partials derivatives of a pixel $P(x,y)$ can be gathered in a vector $D^{\mathrm{rot}}(x,y)$:

$$D^{\mathrm{rot}}(x,y) = \begin{pmatrix} I_{X,\alpha'}(x,y) \\ I_{Y,\alpha'}(x,y) \end{pmatrix}$$

The equivalent vector $D(x,y)$ in the straight system is computed by rotating the original coordinates system by an angle $\phi = -45°$. To respect spatial ratio between both straight and rotated systems, a homothecy must be done with a factor $\sqrt{2}$. The required transformation matrix is easily calculated by:

$$D(x,y) = \sqrt{2} \cdot \begin{pmatrix} \cos(\phi) & \sin(\phi) \\ -\sin(\phi) & \cos(\phi) \end{pmatrix} \cdot D^{\mathrm{rot}}(x,y) \quad (4)$$

Going through this process for ($I^{\mathrm{G}}_{X,\alpha}$, $I^{\mathrm{G}}_{Y,\alpha}$) and ($I^{\mathrm{RB}}_{X,\alpha}$, $I^{\mathrm{RB}}_{Y,\alpha}$) gives the images of straight partial derivatives $I^{\mathrm{G}}_{x,\alpha}$, $I^{\mathrm{G}}_{y,\alpha}$, $I^{\mathrm{RB}}_{x,\alpha}$ and $I^{\mathrm{RB}}_{y,\alpha}$ in half pixels for the G and RB componants. These partial derivatives in odd pixels (for green) and partial derivatives in even pixels (for others) must be estimated. Thereafter, missing values are calculated by bilinear interpolation in the direction perpendicular to the derivation direction. The fully-defined partial derivative images $\check{I}^{\mathrm{G}}_{x,\alpha}$, $\check{I}^{\mathrm{G}}_{y,\alpha}$, $\check{I}^{\mathrm{RB}}_{x,\alpha}$ and $\check{I}^{\mathrm{RB}}_{y,\alpha}$ are computed by:

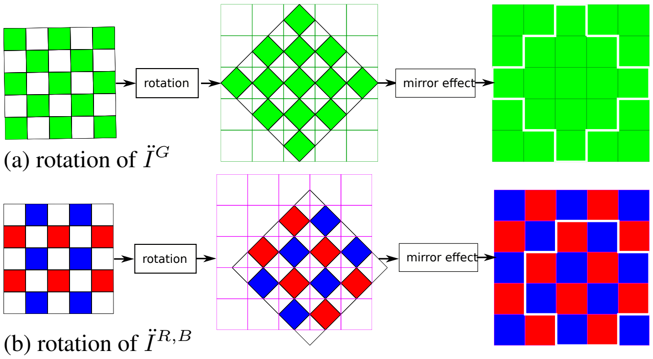
$$\check{I}_{x,\alpha}(x,y){=}\begin{cases} I_{x,\alpha}(x,y) \text{ where } P(x,y) \text{ is defined (not empty)} \\ \frac{1}{2}\left[I_{x,\alpha}(x,y-1) + I_{x,\alpha}(x,y+1)\right] \text{ elsewhere.} \end{cases} \quad (5)$$

$$\check{I}_{y,\alpha}(x,y){=}\begin{cases} I_{y,\alpha}(x,y) \text{ where } P(x,y) \text{ is defined (not empty)} \\ \frac{1}{2}\left[I_{y,\alpha}(x-1,y) + I_{y,\alpha}(x+1,y)\right] \text{ elsewhere.} \end{cases} \quad (6)$$

Color gradient is computed using the structure tensor (detailed in Section 2.2) by considering that red and blue derivative images are equal to the chrominance derivative images.

3.1. Application to edge detection

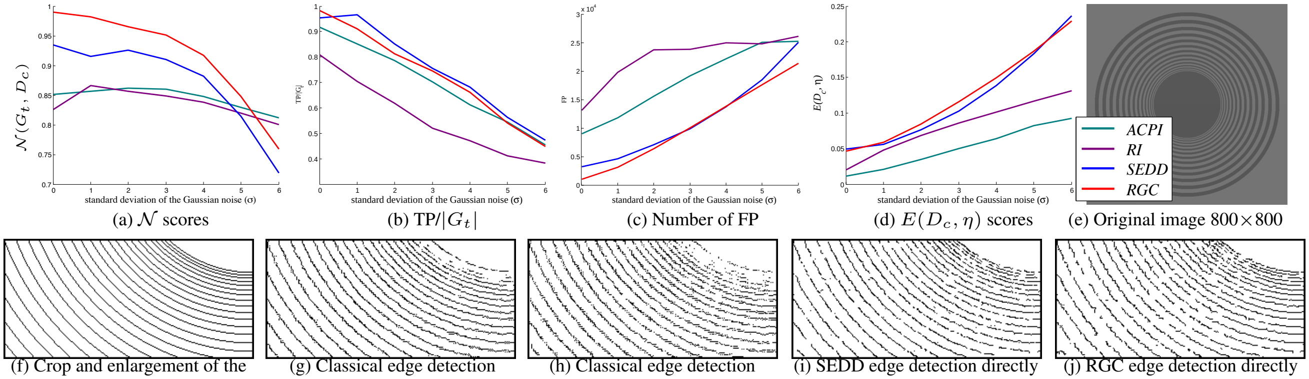
In both methods, Di Zenzo gradient (detailed in Section 2.2) is calculated from the estimated color partial derivative images $\check{I}^{\mathrm{R}}_x$, $\check{I}^{\mathrm{R}}_y$, $\check{I}^{\mathrm{G}}_x$, $\check{I}^{\mathrm{G}}_y$, $\check{I}^{\mathrm{B}}_x$ and $\check{I}^{\mathrm{B}}_y$. Thus, edges can be extracted by thresholding local maxima of the gradient norm in the gradient direction [11]. Finally, to eliminate undesirable edges due, for example, to noise, the image of edges must be thresholded either with a classic thresholding process or by hysteresis [11].



(a) rotation of $\ddot{I}^G$

(b) rotation of $\ddot{I}^{R,B}$

Fig. 2: The rotation process of $\ddot{I}^G$ and $\ddot{I}^{R,B}$.



(a) $\mathcal{N}$ scores (b) TP/$|G_t|$ (c) Number of FP (d) $E(D_c, \eta)$ scores (e) Original image 800×800

(f) Crop and enlargement of the ground truth (image inverted) (g) Classical edge detection after ACPI demosaicking, (h) Classical edge detection after ARI demosaicking, (i) SEDD edge detection directly on the Bayer CFA image, (j) RGC edge detection directly on the Bayer CFA image.

Fig. 3: Comparison and evaluation of edge detection using α=2.01. Detected edges are tied to the noisy image with $\sigma = 4$.

4. EVALUATION AND EXPERIMENTAL RESULTS

Experiments are carried out on synthetic and real images. Smoothed component Estimation and Deriche Derivative (SEDD), applied to CFA image, propose to use raw data directly to estimate the image partial derivatives with a adaptive Deriche filter for edge detection [12]. Deriche filter is also applied on color images after demosaicking process. Adaptive Color Plane Interpolation (ACPI) [13], applied to demosaicking image, is a simple demosaicking procedure consisting in making a bilinear interpolation in the direction perpendicular to the gradient. Adaptative Residual Interpolation (ARI) [14], interpolates colors in a residual domain and produces visually satisfying images, but has a very high computational complexity which makes it time-consuming. Evaluations are reported using synthetic images where the true positions of the edges are perfectly known [15]. Let G_t be the reference contour map corresponding to the ground truth and D_c the detected contour map of an image I. Comparing pixel by pixel G_t and D_c, a basic evaluation is composed of statistics:

- True Positive (TP), commun points of G_t and D_c,
- False Positive (FP), spurious detected edges of D_c,
- False Negative (FN), missing boundary points of D_c,
- True Negative (TN), common non-edge points.

Thus, described in [16], the normalized $\mathcal{N}$ edge detection evaluation measure is, for $FN > 0$ or $FP > 0$:

$$\mathcal{N}(G_t, D_c) = \frac{1}{FP + FN} \cdot \left[\frac{FP}{|D_c|} \cdot \sum_{p \in D_c} \frac{1}{1 + \kappa_{FP} \cdot d^2_{G_t}(p)} + \frac{FN}{|G_t|} \cdot \sum_{p \in G_t} \frac{1}{1 + \kappa_{FN} \cdot d^2_{D_c}(p)}\right], \tag{7}$$

where $(\kappa_{FP}, \kappa_{FN}) \in]0, 1]^2$ represent two scale parameters [16], $| \cdot |$ denotes the cardinality of a set, and $d_A(p)$ is the minimal Euclidian distance between a pixel p and a set A [15]. Therefore, the measure $\mathcal{N}$ calculates a standardized dissimilarity score; the closer the evaluation score is to 1, the more the segmentation is qualified as suitable. On the contrary, a score close to 0 corresponds to a poor detection of contours, finally, if FP=FN=0, then $\mathcal{N}$ =1.

The objective here is to get the best contour map in a supervised way. For that, the contours are extracted after a suppression of the local non-maxima, then a threshold by hysteresis is applied to obtain a binary segmentation [11]. Theoretically, to be objectively compared, the ideal contour map of a measure must be a D_c at which the supervised evaluation gets the highest score [17, 16]. For each better segmentation tied to $\mathcal{N}$, the false positives (FP) and percentage of true positives relative to the total number of edge pixels of G_t are also displayed (TP/G_t). The ground truth image is available in Fig. 3(f) and the original Bayer CFA image is available the Fig. 3(e). Also, the last evaluation measure concerns the gradient angle, η [18]. Once D_c is created, considering $\mathcal{C}_{D_c}$, the set of contour chains in D_c (i.e., at least 2 pixels per chain), the gradient evaluation is computed as follows:

$$E(\mathcal{C}_{D_c}, \eta) = \frac{1}{|\mathcal{C}_{D_c}|} \cdot \sum_{p \in \mathcal{C}_{D_c}} \sum_{d_k \in W} \left[1 - \frac{|90^\circ - |\overrightarrow{\eta_p} - \overrightarrow{\eta_{d_k}}||}{90^\circ}\right] / c_k,$$

where d_k represents a contour pixel belonging to W, a 3×3 window centered on p, $\overrightarrow{\eta_{d_k}}$ the gradient orientation of d_k and c_k the number of contour pixels in W, minus the central pixel. This evaluation linearly ranges from 0 for identical angles of $\overrightarrow{\eta_p}$ and $\overrightarrow{\eta_{d_k}}$ to 1 for angles that differs. Note that $\overrightarrow{\eta_{d_k}}$ and $\overrightarrow{\eta_p}$ angles belong to $[0; 180^\circ[$ and when one direction approximates 0 and the other direction 180°, the evaluation of these two directions remains close to 0. These scores are presented in Figs.3 (a-d) according to the standard deviation of the white Gaussian noise (σ) added to the CFA image for each channel. Therefore, regarding the proposed RGC method, the scores of $\mathcal{N}$ outperform other scores.The segmentation available in Fig. 3 (j) on the top right illustrates a better edge extraction than detected edge after demosaicking process. The proposed method obtains also good statistics in terms of FPs and TPs. Finally, the RGC method obtains more errors for the gradient directions because the gradient is computed regarding an image with empty pixels (the CFA image), contrary to edge detection after ARI and ACPI process which is computed on full channels.

Results presented in Fig.4 illustrates the interest to extract directly features instead of demosaicking before segmentation. Indeed, as illustrated in Figs.4(c) and (d), demosaicking creates undesirable colors, especially for edges or rupture points. Each result is displayed with the same threshold. Thus, obtained edges with the proposed technique are straighter regarding the vertical boards whereas contours of the details in the gray object on the right are well extracted whereas demosaicking methods create a checkerboard effect.

4.1. Computational complexity

For real-time constraints, less time-consuming algorithms are preferable. This parts reports complexity of the compared algorithms: ACPI, ARI, SEDD and RGC using and the same color edge detection process. Their complexity only differ in (i) demosaicking process for ACPI and ARI, and (ii) partial color derivative computation for SEDD and RGC.

Tab. 1 shows the computational complexity of those algorithms. The complexity is evaluated as the number of required elementary operations per pixel: addition, subtraction, multiplication and division, each one being weighted similarly, as detailed in [7]. The expressions of the recursive Deriche (recursive of order 2) equations require seven additions and eight multiplications per pixel for the smoothing filter, seven additions and nine multiplications per pixel for the derivative filter [10]. So 15 operations for left to right, and 15 from right to left (causal + anti-causal), i.e., 30 operations for the smoothing filter (for each (RGB) channels, i.e., 90 operations). In the same way, 96 operations are required per pixels for the derivative filter using a full color image (after demosaicking).

In RGC, a diagonal Deriche smoothing filter is processed on the CFA image, which takes 15 operations for each X and Y smoothened images. The diagonal derivative filter requires $2 \cdot 16$ operations per pixel. Vector conversion in eq. (4) takes two multiplications and one addition per coordinates, which makes $2 \cdot 3$ operations per pixel. Interpolation of missing derivative values in eq. (5) and (6) takes 2 operations per coordinate in half pixels of both channels, which is equivalent to $2 \cdot \frac{2 \cdot 2}{2} = 4$ operations per pixel. Hence derivation step of RGC takes $2 \cdot 21$ operations per pixel. ARI demosaicking is far more complex because its complexity as 1245 operations to estimate each missing green value and 534 operations to estimate a red or blue value. Thus, RGC allows to dramatically reduce the computational complexity. The number of elementary operations required is (i) three times lower than ACPI, (ii) almost twenty times lower than RI, and (iii) twice lower than SEDD which is the method with no demosaicking proposed in [7] that offered the best edge quality with the lowest complexity among color derivative methods.

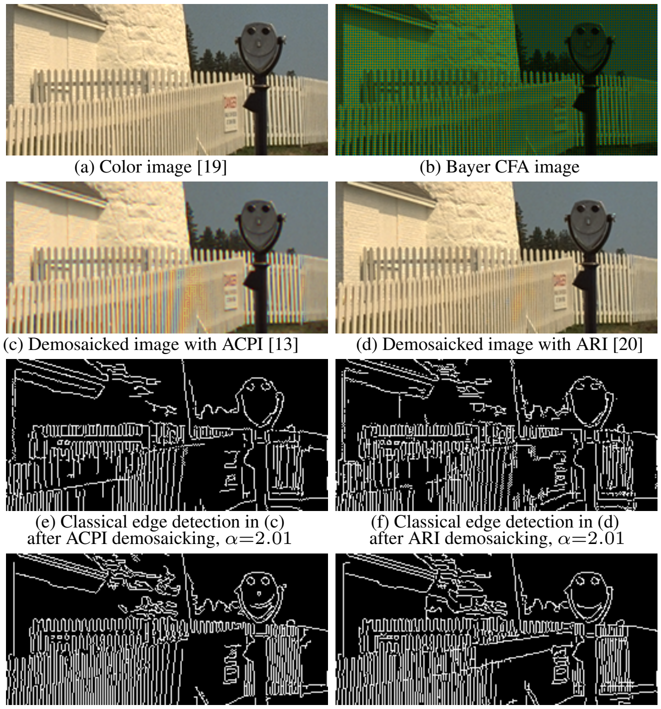

(a) Color image [19]
(b) Bayer CFA image
(c) Demosaicked image with ACPI [13]
(d) Demosaicked image with ARI [20]
(e) Classical edge detection in (c) after ACPI demosaicking, α=2.01
(f) Classical edge detection in (d) after ARI demosaicking, α=2.01
(g) SEDD edge detection on (b), α=2.01
(h) RGC edge detection on (b), α=2.01

Fig. 4: Comparison of classical edge detection using color Deriche filter and detection directly on Bayer CFA images.

Algorithms	ACPI [13]	ARI [20]	SEDD [7]	RGC
demosaicking	30	1779	0	0
Deriche smoothing	90	90	40	30
Deriche derivation	96	96	96	42
Total	216	1965	136	**72**

Table 1: Number of elementary operations per pixel required for the different color partial derivative computation methods.

5. CONCLUSION

Aberkane et al. in [7] made a first investigation of gradient-oriented edge detection methods that exploit raw CFA images and bypass the demosaicking step, with the main goal of reducing computational time. Their best color partial derivative computation method SEDD still offered an average performance lower than ACPI, which uses standard Deriche filters after a simple but efficient demosaicking process. This work aimed to propose other CFA-based multi-channel partial derivative computation methods with better performances and lower computational time than those using demosaicking. A new approach is presented in this paper, computing the partial derivatives of the green channel and the union of red and blue channels. This innovative approach is the use of diagonal Deriche filters also implementable with standard Deriche filters on a rotated image. This method is tested on a dataset of synthetic images. and the influence the Gaussian noise outperforms other methods, even though the gradient and its direction are estimated on images containing empty pixels. Eventually computational complexity showed that RGC allows to considerably reduce the number of elementary operations per pixel compared to demosaicking process. Several applications may be investigated using the proposed method, such that develop a demosaicking technique using contour orientations [21] or up-sample images [22].

6. REFERENCES

[1] R. Lukac and K. N. Plataniotis, “Color filter arrays: Design and performance analysis,” *IEEE TCE*, vol. 51, no. 4, pp. 1260–1267, 2005.

[2] B. E. Bayer, “Color imaging array,” in *U.S. Patent 3971065*, 1976.

[3] D. Menon and G. Calvagno, “Color Image Demosaicking: An Overview,” *Image Communication*, vol. 26, no. 8-9, pp. 518–533, Oct. 2011.

[4] H. Maître, *From photon to pixel: the digital camera handbook*, John Wiley & Sons, 2017.

[5] T. Ehret and G. Facciolo, “A study of two cnn demosaicking algorithms,” *Image Processing On Line*, vol. 2, pp. 220–230, 2019.

[6] O. Losson, L. Macaire, and Y. Yang, “Comparison of color demosaicing methods,” *Advances in Imaging and Electron Physics*, vol. 162, pp. 173–265, July 2010.

[7] M. Breier, C. Haas, W. Li, and D. Merhof, “Color filter arrays revisited – Evaluation of Bayer pattern interpolation for industrial applications,” in *IEEE INDIN*, 2016, pp. 52–57.

[8] A. Aberkane, O. Losson, and L. Macaire, “Edge detection from Bayer color filter array image,” *J. of Elec. Imag.*, vol. 27, no. 1, pp. 011006, Jan. 2018.

[9] R. Deriche, “Fast algorithms for low-level vision,” *IEEE TPAMI*, vol. 12, no. 1, pp. 78–87, 1990.

[10] S. Di Zenzo, “A note on the gradient of a multi-image,” *CVGIP*, vol. 33, no. 1, pp. 116 – 125, 1986.

[11] R. Deriche, “Using canny's criteria to derive a recursively implemented optimal edge detector,” *IJCV*, vol. 1, no. 2, pp. 167–187, 1987.

[12] J. Canny, “A computational approach to edge detection,” *IEEE TPAMI*, pp. 679–698, 1986.

[13] A. Aberkane, *Edge detection in CFA images*, Ph.D. thesis, Université Lille 1 Sciences et Technologies, 2017.

[14] J. F. Hamilton and J. E. Adams, “Adaptive color plan interpolation in single sensor color electronic camera,” US patent 5,629,734, to Eastman Kodak Co., Patent and Trademark Office, Washington D.C., 1997.

[15] Yusuke Monno, Daisuke Kiku, Masayuki Tanaka, and Masatoshi Okutomi, “Adaptive residual interpolation for color and multispectral image demosaicking,” *Sensors*, vol. 17, no. 12, 2017.

[16] H. Abdulrahman, B. Magnier, and P. Montesinos, “From contours to ground truth: How to evaluate edge detectors by filtering,” *J. of WSCG*, vol. 25, no. 2, pp. 133–142, 2017.

[17] B. Magnier, “Edge detection evaluation: A new normalized figure of merit,” in *IEEE ICASSP*, 2019, pp. 2407–2411.

[18] B. Magnier, H. Abdulrahman, and P. Montesinos, “A review of supervised edge detection evaluation methods and an objective comparison of filtering gradient computations using hysteresis thresholds,” *Journal of Imaging*, vol. 4, no. 6, pp. 74, 2018.

[19] B. Magnier, “An objective evaluation of edge detection methods based on oriented half kernels,” *ICISP*, pp. 80–89, 2018.

[20] Eastman Kodak and Various photographers, “Kodak Photo CD PCD0992, Access Software & Photo Sampler, Final version 2.0,” [CD-ROM, Part No. 15-1132-01], 1991.

[21] D. Kiku, Y. Monno, M. Tanaka, and M. Okutomi, “Beyond color difference: Residual interpolation for color image demosaicking,” *IEEE TIP*, vol. 25, no. 3, pp. 1288–1300, 2016.

[22] P. Getreuer, “Image demosaicking with contour stencils,” *Image Processing On Line*, vol. 2, pp. 22–34, 2012.

[23] J. Seiler, M. Jonscher, M. Schöberl, and A. Kaup, “Resampling images to a regular grid from a non-regular subset of pixel positions using frequency selective reconstruction,” *IEEE TIP*, vol. 24, no. 11, pp. 4540–4555, 2015.