Shen-Castan Based Edge Detection Methods for Bayer CFA Images

Zian Li, Arezki Aberkane, Baptiste Magnier

SHEN-CASTAN BASED EDGE DETECTION METHODS FOR BAYER CFA IMAGES

Zian Li [⋆,†]*, Arezki Aberkane*[†] *and Baptiste Magnier*[⋆]

[⋆] EuroMov Digital Health in Motion, Univ Montpellier, IMT Mines Ales, Ales, France
[†] Technical Innovation Team, Audensiel Technologies, Boulogne-Billancourt, France

ABSTRACT

Color Filter Array (CFA) represents a mosaic of incomplete color information from a digital image. This paper presents two edge detection methods performing directly on CFA images, without the necessity of the demosaicing process, thus saving significant computation steps. First, existing methods for CFA images based on well-known Deriche recursive filters are revisited. Then, new algorithms based on Shen-Castan filters design are proposed. They correspond to recursive first-order filters, outperforming the complexity of other edge detection techniques. Finally, quantitative assessments based on synthesized images using normalized Figure of Merit evaluate the performances of the edge detection methods, while qualitative results based on real images are also reported, illustrating the new methods reliability.

Index Terms— CFA Image, Filtering, Edge Detection

1. INTRODUCTION AND MOTIVATIONS

In digital imaging, each pixel on a single-sensor camera contains a color filter array (CFA) capturing only one wavelength of light that only lets through one color [1]. Indeed, a captured image records the brightness of the red, green, and blue pixels separately. A very common CFA is the Bayer matrix (see Fig.1). An image of this pattern is called a CFA image. To create full color information, the demosaicing process is applied to the CFA image, several algorithms have been proposed [2][3], amongst them the simplest ones are nearest neighbor and bilinear interpolations.

Cameras equipped with CFA filters are portable and lightweight, and are often used in embedded devices. Usually, feature extraction from images requires full-color information, thus the CFA images are mostly demosaiced before being processed. However, the demosaicing algorithm requires extra computation, resulting in an undesirable inefficiency for machines with low performance. Consequently, for simplicity and convenience, features may be extracted using grayscale vision sensors indicating only the brightness of a pixel. In response to the demand of feature extraction efficiency directly on CFA images, Aberkane *et al.* [4] and Magnier *et al.* [5] have proposed edge detection algorithms that operate based on Deriche edge detector and bilinear interpolations [6].

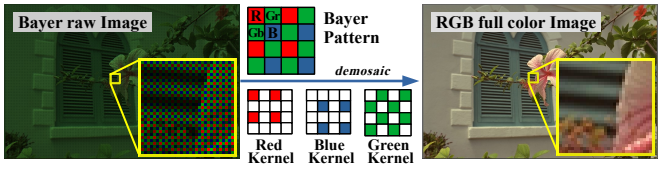


Fig. 1: Bayer patterns on a CFA image and full-color image.

In this paper, two edge detection approaches adjusting Shen-Castan edge detector [7] are developed. They require less computation steps than methods proposed in the literature. Then, the new algorithms are evaluated with synthetic images and their performances versus complexity are reported. Eventually, results on real-world images showed that the proposed approaches achieved satisfying detection quality while reducing considerably the algorithmic complexity.

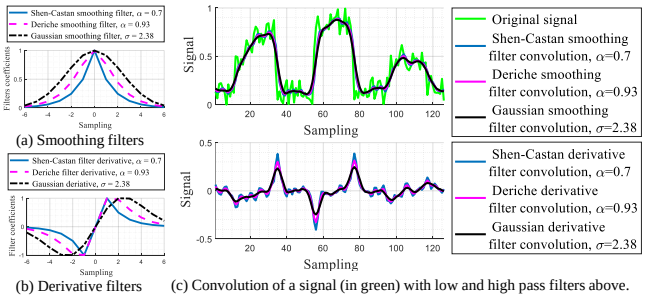
2. THE SHEN-CASTAN EDGE DETECTOR

2.1. Recursive and separable filters for edge detection

In the field of image analysis and computer vision, edge detection and extraction represent important image processing tasks and are related to a wide range of application [8]. Edges are associated with intensity changes in the gray level image and constitute an efficient descriptor of the image structure. In this context, several operators were developed in the literature [9]. Canny presented criteria for measuring the quality of optimal edge detectors regarding one-dimensional (1D) Finite Impulse Response (FIR) filter [10]. The performances of a detector are essentially characterized by:

- Detection with low error rate: the operator must give a response in the vicinity of a contour.
- Localization: the contour must be precisely located.
- Unicity: an outline must elicit a single response from the extraction operator.

Edge detection methods differ in the types of optimal filters and one solution is the Gaussian filter $G_\sigma(t) = \frac{1}{\sqrt{2\pi}\sigma} \cdot \mathrm{e}^{-t^2/2\sigma^2}$, where σ represents the standard deviation of the Gaussian. Based on this design, several approaches were proposed to edge detection utilizing Infinite Impulse Response (IIR) filters which can be recursively implemented with high efficiency



(a) Smoothing filters (b) Derivative filters (c) Convolution of a signal (in green) with low and high pass filters above.

Fig. 2: Comparison of 1D filters in (a)-(b) and convolution with a noisy signal in (c)-top-. In (a), the parameters of the different filter equations are chosen such that the smoothing filter support contains 98% of the filtered information, here: support = 13 sample points.

owing to low algorithmic cost. Indeed, Shen and Castan [7] proposed an operator optimizing detection and localization criteria. Its implementation is based on a first order recursive exponential filter. The obtained smoothing filter is written by:

$$\mathcal{S}_\alpha(x) = c \cdot e^{-\alpha|x|}, \tag{1}$$

with $c = \frac{1-e^{-\alpha}}{1-e^{+\alpha}}$ the normalization coefficient. The α parameter defines the "width" of the filter: the smaller α is, the greater the filter performs the smoothing. Thereafter, the high pass filter $\mathcal{D}_\alpha$ is tied to the derivative of $\mathcal{S}_\alpha$ with $\mathcal{D}_\alpha(0)=0$ (Fig.2 (a)-(b)) for symmetrical reasons:

$$\mathcal{D}_\alpha(x) = \begin{cases} c \cdot e^{-\alpha x} & \text{if} \quad x > 0 \\ 0 & \text{if} \quad x = 0 \\ -c \cdot e^{\alpha x} & \text{if} \quad x < 0. \end{cases} \tag{2}$$

Thus, the first order discontinuity at point 0 constitutes a sharp filter, as illustrated in Fig. 2(c). Thanks to the separability property, the smoothed image is given by:

$$I_{\mathcal{S}}(x,y) = (I(x,y) * \mathcal{S}_\alpha(x)) * \mathcal{S}_\alpha(y), \tag{3}$$

where "$*$" represents the product of convolution, I the original image and (x,y) the pixel coordinates. The low-pass filter $\mathcal{S}_\alpha$ is applied successively horizontally and vertically (or inversely). Elsewhere, the image derivative using these filters is equivalent to using the 2D derivative filter (Fig. 3(c)); it can also be obtained by filter separability using a horizontal derivative with $\mathcal{D}_\alpha(x)$ and a vertical smoothing with $\mathcal{S}_\alpha(y)$:

$$I_x(x,y) = (I(x,y) * \mathcal{D}_\alpha(x)) * \mathcal{S}_\alpha(y), \tag{4}$$

for a x-derivative of the image. The y-derivative is computed by applying a vertical $\mathcal{D}_\alpha$ and a horizontal $\mathcal{S}_\alpha$ filters on a gray level image. Regarding two dimensions, the Shen-Castan filter preserves its sharp shape, as illustrated in Fig. 3(c). Hence, it makes it possible to avoid a significant delocalization of the contours in the smoothed image, even with low α values. Both $\mathcal{S}_\alpha$ and $\mathcal{D}_\alpha$ represent first ordered recursive filters, allowing the (very) fast computation of image derivatives.

Otherwise, based on Canny's design, the IIR Deriche filter has analytical expression allowing an exact recursive implementation of order 2 [6]. The smoothing part is given by:

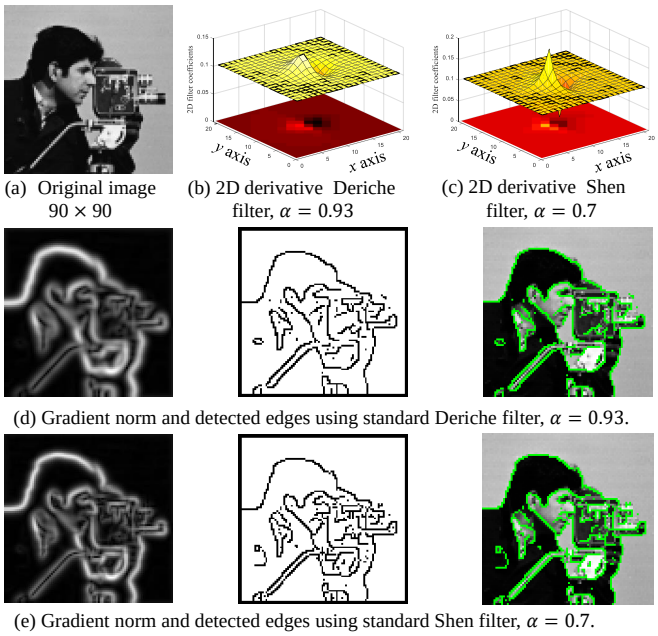
$$\mathcal{H}_\alpha(x) = a \cdot (\alpha|x| + 1) \cdot e^{-\alpha|x|}, \tag{5}$$

It is also separable and the derivative part corresponds to:

$$\mathcal{K}_\alpha(x) = b \cdot x \cdot e^{-\alpha|x|}, \tag{6}$$

with a and b two normalization coefficients [6]. The α parameter tunes these filters as for $\mathcal{S}_\alpha$ and $\mathcal{D}_\alpha$: the bigger α is, the less the filter smoothes the signal (thin and sharp filter) and prioritizes the edge localization. On the contrary, when α is decreasing, the filter appears increasingly more like a Gaussian (see Fig. 2(a)). Consequently, edges of small object are more precisely extracted using Shen-Castan algorithm, as illustrated in Fig. 2(a)-(b) and Fig. 3(d)-(e). Moreover, $\mathcal{S}_\alpha$ filter computes a gradient norm less extended and offers the possibility to detect separately two close contours without any delocalization caused by the regularization filter, contrary to Gaussian or Deriche filters.

Despite its inability to respond to all of Canny's criteria, the Shen-Castan filter offers a lower algorithmic cost than the Deriche filter, along with satisfying results on contour localization regarding gray level images. These characteristics may be attractive in domains as embedded systems, where the algorithm complexity must be adjusted as a function of the desired task and the machine power.



(a) Original image 90 × 90 (b) 2D derivative Deriche filter, $\alpha = 0.93$ (c) 2D derivative Shen filter, $\alpha = 0.7$

(d) Gradient norm and detected edges using standard Deriche filter, $\alpha = 0.93$.

(e) Gradient norm and detected edges using standard Shen filter, $\alpha = 0.7$.

Fig. 3: Comparison of 2D filter shapes and detected edges using standard Deriche and Shen-Castan filters on a gray level image. Thresholds are selected by Rosin technique [11].

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 with each channel. The best known color gradient was introduced by Di Zenzo [12] 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:

$$d\mathbf{I}^2(\theta) = f \cdot \cos^2(\theta) + 2 \cdot g \cdot \cos(\theta)\sin(\theta) + h \cdot \sin^2(\theta), \quad (7)$$

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 $|\nabla \mathbf{I}| = |d\mathbf{I}^2\left(\theta^*\right)|$.

Eventually, edges can be extracted by thresholding local maxima of the gradient norm in the θ^* direction [10].

3. EDGE DETECTION FOR CFA IMAGES

Different from full-color image, the CFA image lacks information at each pixel. Therefore, the edge detector cannot directly operate on CFA images as on full-color images, and extra steps are necessary for the estimation of missing colors.

3.1. Deriche based approaches - SEDD and SDDE

In the works of Aberkane *et al.* [4] and Magnier *et al.* [5], Deriche-based algorithms were proposed (Eqs. (5) and (6)) directly on the CFA image to extract contours.

As illustrated in Fig. 1, there are discontinuities (holes) between pixels of the same colors in CFA images. However, it is possible to obtain a seamless image in exchange for the image resolution by down-sampling the pixels of the same color channel from a CFA image and then reconstructing the image without empty values. To solve this, Aberkane *et al.* [4] introduced a spatial distance parameter (d) directly into the Deriche filter implementation (Eqs. (5) and (6)), so that the filters can be applied on CFA images the same way as on a full image (image without empty values). Although the filter is applied discretely using a spatial distance parameter, the filter response remains discrete because for every pixel there are still two colors missing.

Therefore, inspired by bilinear interpolation, Aberkane *et al.* [4] proposed an empty-pixel estimation method to obtain an integral filter response with no missing components, and two methods were conceived: SEDD *(Smoothed component Estimation of Deriche Derivative)* and SDDE *(Smoothed Deriche Derivative Estimation)*. This is achieved by finding 4 types of neighborhoods of a pixel P in a Bayer CFA image: $\mathbb{N}^H$, $\mathbb{N}^V$, $\mathbb{N}^+$ and $\mathbb{N}^\times$. The superscript H (or V) indicates that the two neighboring horizontal (or vertical) pixels have the same color component. The superscript $+$ (or $\times$) indicates that the four neighboring horizontal and vertical (or diagonal) pixels have the same color component (related to Fig. 1). As an example, considering the red pixels of a CFA image, the estimated full-color pixel (i.e., a triple) is expressed by: $\mathbf{I}(x,y) = (I^{CFA}(x,y), I^{CFA,+}(x,y), I^{CFA,\times}(x,y))$, where $I^{CFA,+}(x,y) = \frac{1}{4} \cdot \sum_{(i,j)\in\mathbb{N}^+(x,y)} I_{CFA}(i,j)$. Other components are as well easily computed, see [4].

On another note, the DRGC *(Deriche derivative of Rotated Green-Chrominance)* method proposed by Magnier *et al.* [5] operates on Bayer CFA images in the diagonal directions in order to obtain pixel continuity (see Fig. 1). Here, the full color gradients are restored using bilinear interpolation. Tab.1 reported the evaluation and complexity of Deriche filters adapted to CFA images. The results showed that without demosaicing process, the edge detection remains very efficient with low computation time. Thereafter, the proposed methods offers a reduction in complexity.

3.2. Proposed Algorithms

For the sake of convenience, some notations are defined. First, the notations of images are expressed as follows:

- Red Pixels and Blue Pixels respectively: P_R and P_B.
- Green Pixels to the right of the Red Pixels: P_{Gr}.
- Green Pixels to the left of the Blue Pixels: P_{Gb}.

Considering an image with three color channels, Shen-Castan's smoothing and derivation filters (Eqs.(1) and (2)) are implemented by first-order recursive filtering. However, CFA images have only one channel, so each pixel carries only one color information, and the other two colors are considered as empty values. The empty values are skipped by specifying a spatial parameter d when applying the filter (see Sec. 3.1). Hence, by taking into account the d parameter, the new smoothing recursive filter is built with the causal and anti-causal parts $I_{S^-_{x,\alpha,d}}$ and $I_{S^+_{x,\alpha,d}}$:

$$I_{S^-_{x,\alpha,d}}(x,y) = c \cdot I(x,y) + e^{-\alpha} \cdot I_{S^-_{x,\alpha,d}}(x-d,y), \quad (8)$$

$$I_{S^+_{x,\alpha,d}}(x,y) = c \cdot I(x+d,y) + e^{-\alpha} \cdot I_{S^+_{x,\alpha,d}}(x+d,y). \quad (9)$$

The smoothed CFA image along x i.e., $I_{\mathcal{S},x}$, is computed by:

$$I(x,y) * \mathcal{S}_{x,\alpha,d}(x,y) = I_{S^+_{x,\alpha,d}}(x,y) + I_{S^-_{x,\alpha,d}}(x,y), \quad (10)$$

where $\mathcal{S}_{x,\alpha,d}(x,y)$ indicates applying the horizontal smoothing on a CFA image with a spatial distance d. The same procedure is applied vertically, considering the vertical spatial distance of d. Then, the horizontal derivative filter of Eq. (2) is recursively implemented with the causal and anti-causal parts in the following equations:

$$\begin{aligned} I_{\mathcal{D}^-_{x,\alpha,d}}(x,y) &= (1-e^{-\alpha}) \cdot I(x,y) + e^{-\alpha} \cdot I_{\mathcal{D}^-_{x,\alpha,d}}(x-d,y), \\ I_{\mathcal{D}^+_{x,\alpha,d}}(x,y) &= (e^{-\alpha}-1) \cdot I(x+d,y) + e^{-\alpha} \cdot I_{\mathcal{D}^+_{x,\alpha,d}}(x+d,y). \end{aligned} \quad (11)$$

Thus, the derivative image along x, i.e. $I_{\mathcal{D},x}$ is computed by:

$$I(x,y) * \mathcal{D}_{x,\alpha,d}(x,y) = I_{\mathcal{D}^+_{x,\alpha,d}}(x,y) + I_{\mathcal{D}^-_{x,\alpha,d}}(x,y), \quad (12)$$

where $\mathcal{D}_{x,\alpha,d}(x,y)$ represents the horizontal derivative filter on a CFA image with a spatial distance d (cf. Eq. 4). Lastly, the same strategy is applied vertically to obtain a vertical derivative, considering the vertical spatial distance d.

Once different notations have been properly defined, two edge detection methods based on exponential Shen-Castan filter are proposed and adapted to Bayer CFA images.

3.2.1. *First method: SESCD*

SESCD (Smoothed components Estimation Shen-Castan Derivative) applies first a smoothing filter with d=2 and then estimates the missing components (empty pixels), immediately after smoothing with Eqs. (9) and (8). Thus, the smoothed component in the x-direction is obtained by:

$$I^{CFA}_{\mathcal{S},x}(x,y) = I^{CFA} * \mathcal{S}_{x,\alpha,2}(x,y). \tag{13}$$

The y-direction is calculated the same way by replacing x with y. The order of filters is not invertible in this case since the estimation step is after the smoothing step. Here, the three red, green and blue smoothed full channels are estimated for each position of the P_R, P_{Gr}, P_{Gb} and P_B horizontally and vertically by considering respectively $\mathbf{I}_{\mathcal{S},x}$ and $\mathbf{I}_{\mathcal{S},y}$:

$$\mathbf{I}_{\mathcal{S},x}=\begin{cases}(I^{CFA}_{\mathcal{S},x}, I^{CFA,H}_{\mathcal{S},x}, I^{CFA,\times}_{\mathcal{S},x}),\\ (I^{CFA,H}_{\mathcal{S},x}, I^{CFA}_{\mathcal{S},x}, I^{CFA,V}_{\mathcal{S},x}),\\ (I^{CFA,V}_{\mathcal{S},x}, I^{CFA}_{\mathcal{S},x}, I^{CFA,H}_{\mathcal{S},x}),\\ (I^{CFA,\times}_{\mathcal{S},x}, I^{CFA,H}_{\mathcal{S},x}, I^{CFA}_{\mathcal{S},x}),\end{cases} \mathbf{I}_{\mathcal{S},y}=\begin{cases}(I^{CFA}_{\mathcal{S},y}, I^{CFA,V}_{\mathcal{S},y}, I^{CFA,\times}_{\mathcal{S},y}), \text{for } P_R\\ (I^{CFA,H}_{\mathcal{S},y}, I^{CFA}_{\mathcal{S},y}, I^{CFA,V}_{\mathcal{S},y}), \text{for } P_{Gr}\\ (I^{CFA,V}_{\mathcal{S},y}, I^{CFA}_{\mathcal{S},y}, I^{CFA,H}_{\mathcal{S},y}), \text{for } P_{Gb}\\ (I^{CFA,\times}_{\mathcal{S},y}, I^{CFA,V}_{\mathcal{S},y}, I^{CFA}_{\mathcal{S},y}), \text{for } P_B.\end{cases} \tag{14}$$

Once obtained the fully smoothed image for the 3 color channels, the Shen-Castan derivative filter is applied with d=1. Finally, the full derivatives are simply computed by $\mathbf{I}_x(x,y)=\mathbf{I}_{\mathcal{S},y} * \mathcal{D}_{x,\alpha,1}(x,y)$ and $\mathbf{I}_y(x,y)=\mathbf{I}_{\mathcal{S},x} * \mathcal{D}_{y,\alpha,1}(x,y)$ for each channel before applying Di Zenzo tensor, as in Sec. 2.2.

3.2.2. *Second method: SSCDE*

In the SSCDE (Smoothed Shen-Castan Derivative Estimation) algorithm, the derivative filter is applied with d=2, according to Eq. (11). Then, the missing components are estimated before smoothing. Once obtained the full color information, the smoothing filter with d=1 is applied to the image derivative. An example is presented in Fig. 4.

SSCDE differs from SESCD (Sec. 3.2.1) in the order of missing component estimation. As described earlier, in SESCD, the full-color information is restored after the smoothing process, then the derivative is calculated. On the contrary, in SSCDE, before applying the smoothing filter, the full-color restoration is performed immediately after the image derivative by:

$$I^{CFA}_{\mathcal{D},x}(x,y) = I^{CFA} * \mathcal{D}_{x,\alpha,2}(x,y). \tag{15}$$

The y-derivative is obtained by applying $\mathcal{D}_{y,\alpha,2}$ vertically.

Note that the estimation formulae of SSCDE are identical to SESCD (Eq. (14)). The full-color derivative $\mathbf{I}_{\mathcal{D},x}$ and $\mathbf{I}_{\mathcal{D},y}$ are calculated simply by replacing $I_{\mathcal{S}}$ with $I_{\mathcal{D}}$ in the missing-components estimation formulae. Thus, the fully defined derivatives of a CFA image are estimated by Eq. (14).

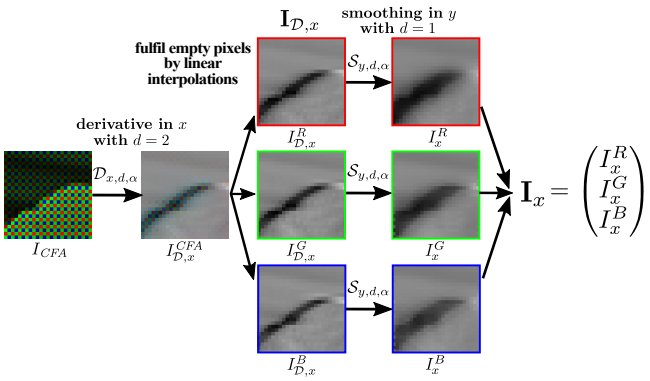


Fig. 4: Flowchart of SSCDE algorithm.

Thereafter, the smoothing filter is applied with the classic parameters d=1 to calculate the fully defined gradient components in x-direction $\mathbf{I}_x(x,y) = \mathbf{I}_{\mathcal{D},x} * \mathcal{S}_{y,\alpha,1}(x,y)$ and in y-direction $\mathbf{I}_y(x,y) = \mathbf{I}_{\mathcal{D},y} * \mathcal{S}_{x,\alpha,1}(x,y)$ respectively for the 3 color components. At last, Di Zenzo tensor is applied to compute the gradient norm and to extract edges (see Sec. 2.2).

4. EDGE DETECTORS EVALUATION AND RESULTS

4.1. Algorithms Complexity

For evaluating the performances of the proposed algorithms, implementations were made in Python and C API. Consequently, the complexities of the algorithms are easily compared since they are implemented in the same way. The algorithmic complexity per pixel is calculated according to the number of arithmetic operations (addition, subtraction, multiplication and division) without taking into account branching instructions or memory accesses. The compared methods involve 3 algorithmic procedures: filtering, missing components estimation (Interp.) and image rotation (for DRGC).

Standard Deriche's filters require 15 smoothing and 16 derivations per pixel (see [4]). Let $\boldsymbol{H}$ denote the image height and $\boldsymbol{L}$ the image width, SEDD *(Smoothed Estimation of Deriche Derivatives)* [4] and SESCD are estimated to require $\mathbf{20}\frac{\boldsymbol{H\times L}}{\mathbf{4}}$ operations per pixel according to Eq. (14).

In that respect, the number of operations for a smoothing or derivation is calculated by $M\times N\times Q\times T$, with:

- M, the number of image channels,
- N, the number of operations per pixel,
- Q, the number of times the filter is applied,
- T, the number of applied filter directions.

The results are summarized in Tab.1. Standard Deriche and Shen-Castan algorithms adapted on full-color images (Sec. 2.2) involves the demosaicing process, corresponding to 30 to 1212 steps per pixel depending on the algorithm [3][4]. Note also that the DRGC method has 2×3 rotation steps [5].

(a) Synthesized image (b) Bayer CFA image (c) True Edge, (Ground truth) (d) SEDD[4], $\mathcal{N}$=0.82 (e) SDDE[4], $\mathcal{N}$=0.82 (f) SESCD, $\mathcal{N}$=0.79 (g) SSCDE, $\mathcal{N}$ = 0.81

Fig. 5: Edges detected from image corrupted by a Gaussian blur and a Gaussian white noise ($\sigma = 8$)

4.2. Edge detection evaluation

To evaluate the edge detection, the *Normalized Figure of Merit* method proposed in [13] is employed. Let G_t be the reference contour map corresponding to the ground truth and D_c the detected contour map of an image. Comparing pixel by pixel G_t and D_c, a simple evaluation based on pixel-wise comparison leads to the definition of the following indicators:

- True Positive (TP), common 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, as described in [13], the normalized $\mathcal{N}$ edge detection evaluation measure is, for $FN{>}0$ or $FP{>}0$:

$$\mathcal{N}(G_t,D_c)=\frac{FP}{Er\cdot|D_c|}\sum_{p\in D_c}\frac{1}{1+\delta d^2_{G_t}(p)}+\frac{FN}{Er\cdot|G_t|}\sum_{p\in G_t}\frac{1}{1+\kappa d^2_{D_c}(p)},$$

with $Er=(FP+FN)$ and where $(\delta,\kappa)\in]0,1]^2$ represent two scale parameters [13]. Meanwhile, $|\cdot|$ denotes the cardinality of a set, and $d_A(p)$ is the minimal Euclidian distance between a pixel p and a set A. So, if there are no error, i.e., $FP{=}FN{=}0$, then it corresponds to a perfect score: $\mathcal{N}{=}1$. Therefore, the measure $\mathcal{N}$ calculates a standardized dissimilarity score; the closer the evaluation score is to 1, the more the edge detection is qualified as suitable.

Evaluation of experiments is carried out on 486 synthetic CFA images, knowing the true edges [14]. Then a Gaussian blur and noises are added (Fig. 5(a)) to assess the robustness of the algorithms. For each filter width (parameter α), the threshold that gives the best evaluation score is recorded. Then, the average score over the 486 images is calculated and recorded as a function of α ranged from 0.1 to 3.0 by a step of 0.1. Finally, the maximum average score $\mathcal{N}$ as well as its corresponding filter parameter is estimated. As an example, the average score of SSCDE method reaches its maximum at $\alpha{=}0.7$, see $\mathcal{N}$ in Tab. 1 and plotted curves in Fig. 6(a). Visually in Fig. 5, SSCDE detects more precisely edges than SESCD, especially on the most interior circle.

The evaluation process above is applied on SESCD and SSCDE methods, as well as on SEDD and SDDE. It appears that SEDD achieves the most reliable result. Nevertheless, the proposed approach SSCDE obtains a better score than SESCD (Tab. 1) and a less noisy edge map as shown in Fig. 5(g). Generally, the number of operations versus the best $\mathcal{N}$ scores are plotted in Fig. 6(b). The upper-leftmost algorithm obtains the most robust performance while requires the lowest complexity. Hence, the Shen-Castan adapted Bayer CFA approaches (SESCD and SSCDE) require a nearly half number of operations of those considering Deriche filters.

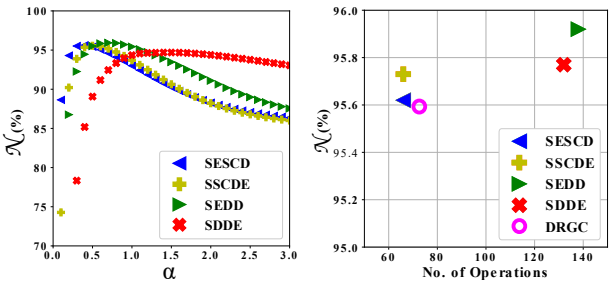
Method	Smoothing	Derivation	Interp.	Total	$\mathcal{N}$ Score
Deriche [6]	3×15×1×2	3×16×1×2	-	186+Demosaic	97.66%
Shen-Castan[7]	3×7×1×2	3×7×1×2	-	84+Demosaic	97.27%
SEDD [4]	1×15×1×2	3×16×1×2	5×2	**136**	**95.92%**
SDDE [4]	3×15×1×2	1×16×1×2	5×2	132	95.77%
DRGC [5]	1×15×1×2	1×16×1×2	2×2	72+6 (Sec.4.1)	95.56%
SESCD	1×7×1×2	3×7×1×2	5×2	66	95.62%
SSCDE	3×7×1×2	1×7×1×2	5×2	**66**	**95.73%**

Table 1: Algorithmic complexity of edge detection filters.

4.3. Experiments on Real Images

Regarding real images presented in Fig. 7, the automatic Rosin thresholding is applied here [11], giving satisfying visual results in term of continuity of detected edges. Otherwise, the classical Shen-Castan edge detector applied to full-color images is used as a reference for the proposed algorithms (SSCDE) which detect the edges directly from CFA images. The selected filter parameter is tied to the highest evaluation score: $\alpha{=}0.7$ (Fig. 6). In this respect, edges detected by SSCDE are visually of the same quality as those obtained by Shen-Castan algorithm for full-color images, illustrating the reliability of the proposed method. In another respect, note that false edges may be created when a contour is blurred, due to the filter's sensitivity or interpolations, and that very close edges may create checkerboard binary pixels.

Similarly, other experiments are presented in additional results file, containing comparison with SESCD technique.



(a) $\mathcal{N}$ versus filter parameter α (b) $\mathcal{N}$ versus Complexity

Fig. 6: Evaluation and complexity of tested algorithms.

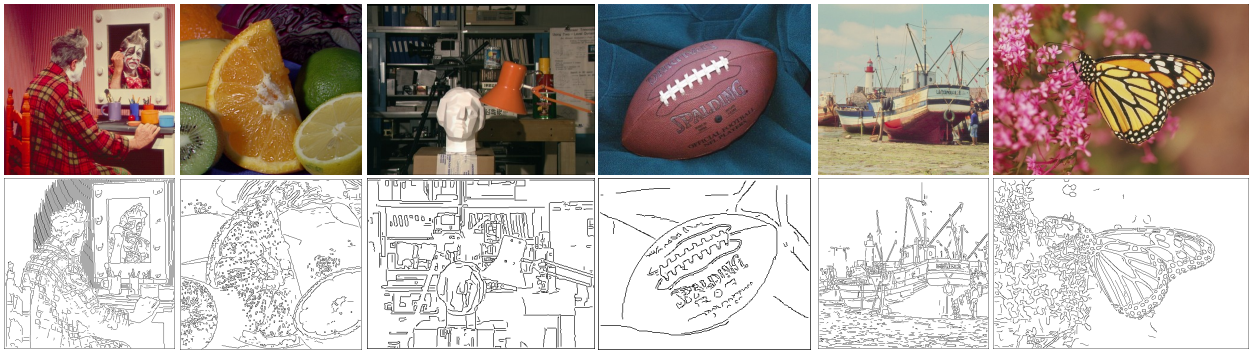
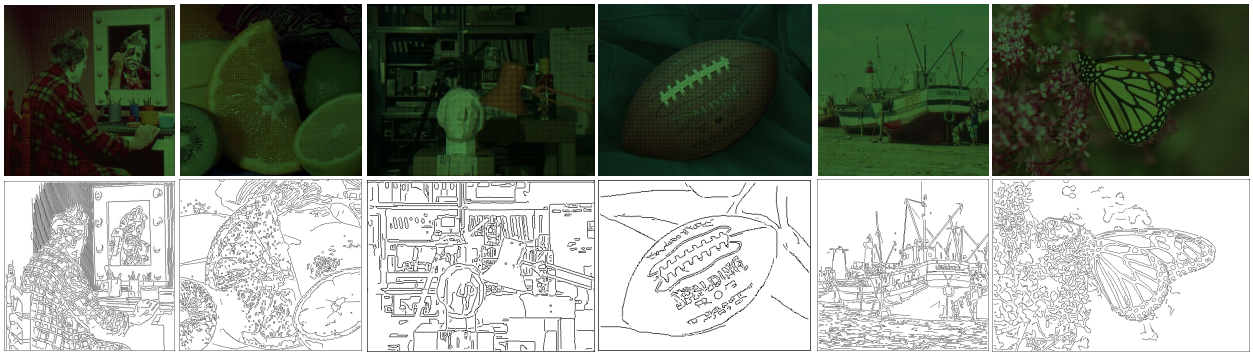


Image 512×512 Image 512×480 Image 384×288 Image 320×256 Image 512×512 Image 768×512

(a) Edge detection on full colour images using Shen-Castan classical filter and Di Zenzo structure tensor.



(b) Edge detection on CFA Bayer images using SSCDE algorithm followed by Di Zenzo structure tensor.

Fig. 7: Visual comparison between classical Shen-Castan versus SSCDE method with automatic thresholding [11].

5. CONCLUSION

Two recursive edge detection filters of order 1 are proposed in this study. Inspired by existing literature, two Shen-Castan based edge detection algorithms have been developed, namely SESCD and SSCDE. They differ from each other only in the order in which the missing components are estimated. The recursive implementation performs with great efficiency, reducing the time of calculation to half of the Deriche approach.

Besides, edge detection score of SSCDE is slightly higher than SESCD, for the same complexity. For real-world CFA images, edges detected directly are equivalent to those obtained after a demoisaicing process, while requiring much less computational complexity. As the computational complexity per pixel is low, several applications may employ the gradient or edges calculated by SSCDE. Features extracted using this filtering technique should be very useful for applications on embedded devices.

6. REFERENCES

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

[2] P. Getreuer, "Image demosaicking with contour stencils," *IPOL*, vol. 2, pp. 22–34, 2012.

[3] Y. Yang O. Losson, L. Macaire, "Comparison of color demosaicing methods," *Adv. in Imaging and Electron Physics, Elsevier*, , no. 162, pp. 173–265, 2010.

[4] 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, 2018.

[5] B. Magnier, A. Aberkane, and N. Gorrity, "A recursive edge detector for color filter array image," in *IEEE ICASSP*, 2020, pp. 2543–2547.

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

[7] J. Shen and S. Castan, "An optimal linear operator for step edge detection," *CVGIP*, vol. 54, no. 2, pp. 112–133, 1992.

[8] D. Marr and E. Hildreth, "Theory of edge detection," *Proc. of the Roy. Soc. of London*, vol. 207, pp. 187–217, 1980.

[9] G. Papari and N. Petkov, "Edge and line oriented contour detection: State of the art," *IVC*, vol. 29, no. 2, pp. 79–103, 2011.

[10] John Canny, "A computational approach to edge detection," *IEEE TPAMI*, vol. 8, no. 6, pp. 679–698, Nov. 1986.

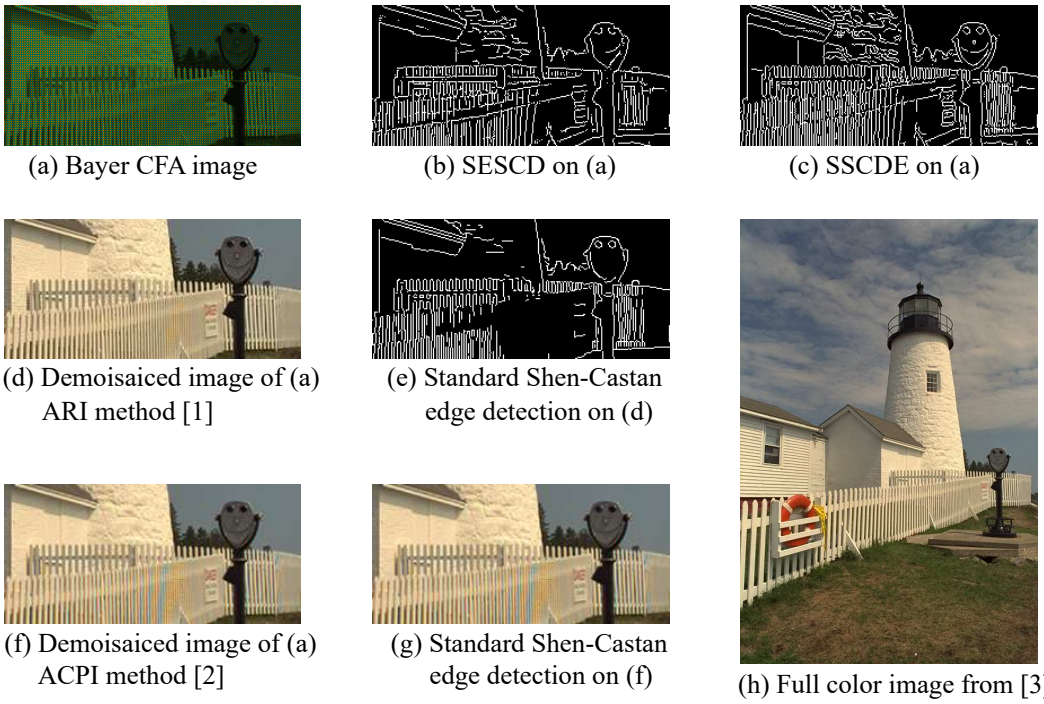
[11] P.L. Rosin, "Unimodal thresholding," *Patt. Rec.*, vol. 34, no. 11, pp. 2083–2096, 2001.

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

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

[14] "Synthetic image dataset for color edge detection," http://color.univ-lille.fr/datasets/color-edge, 2015.

SHEN-CASTAN BASED EDGE DETECTION METHODS FOR BAYER CFA IMAGES: ADDITIONAL RESULTS



(a) Bayer CFA image (b) SESCD on (a) (c) SSCDE on (a)

(d) Demoisaiced image of (a) ARI method [1] (e) Standard Shen-Castan edge detection on (d)

(f) Demoisaiced image of (a) ACPI method [2] (g) Standard Shen-Castan edge detection on (f)

(h) Full color image from [3]

Fig. 1: Comparison of CFA methods and traditional methods on demosaiced Kodak image No.19 [3]. Adaptive Color Plane Interpolation (ACPI) [2] is a simple demosaicking procedure consisting in making a bilinear interpolation in the direction perpendicular to the gradient. Adaptative Residual Interpolation (ARI) [1] interpolates colors in a residual domain and produces visually satisfying images, but has a very high computational complexity. BE CAREFUL: compressed files.

References:

[1] 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

[2] 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.

[3] Kodak image database: `http://www.cs.albany.edu/~xypan/research/snr/Kodak.html`

Fig. 2: Additional Results (arranged in rows): binary images are calculated by Rosin method (see original paper).
From top to bottom: Original Full-Color Image, Bayer CFA Image, Edge detected by SESCD, Edge detected by SSCDE. BE CAREFUL: compressed files.