A novel edge detection method based on efficient gaussian binomial filter

El Houssain Ait Mansour [a,1,*], Francois Bretaudeau [a,2]

[a] Department of Risks and Prevention, Office of Geophysics and Mines, Orleans, France
[1] eh.aitmansour@brgm.fr; [2] f.bretaudeau@brgm.fr
* corresponding author

ARTICLE INFO

Article history
Received November 2, 2020
Revised April 9, 2021
Accepted April 19, 2021
Available online April 19, 2021

Keywords
Image edge detection
Gaussian filter
Image processing
Gaussian binomial filter
Algorithm design and analysis

ABSTRACT

Most basic and recent image edge detection methods are based on exploiting spatial high-frequency to localize efficiency the boundaries and image discontinuities. These approaches are strictly sensitive to noise, and their performance decrease with the increasing noise level. This research suggests a novel and robust approach based on a binomial Gaussian filter for edge detection. We propose a scheme-based Gaussian filter that employs low-pass filters to reduce noise and gradient image differentiation to perform edge recovering. The results presented illustrate that the proposed approach outperforms the basic method for edge detection. The global scheme may be implemented efficiently with high speed using the proposed novel binomial Gaussian filter.

1. Introduction

Edge detection is a fundamental task in digital image processing and analysis, such as recognition, compression, classification, and segmentation [1]–[6]. While, edges contain useful information and features. Edge detection aims to localize the boundaries and image discontinuities over image intensities or textures changes. It also indicates high-frequency bandwidth. This is crucial to physical image aspects and content understanding, such as objects and motion, in computer vision applications [7]–[12]. Poor edge detection tasks dramatically reduce global system performance. The secondary purpose of edge detection is to make images easy to process by removing useful low-frequency content information.

Edge detection tools can be classified into two categories. The classical methods use spatial linear convolution based on image intensity with a fixed edge operator. There were two types of operators, primary (gradient) and secondary derivatives (Laplacian), such as Sobel, Prewitt, Laplacian, and Canny [13]–[17]. Gradient localizes broad edge peak, while Laplacian detects zero-crossing. On the other hand, Laplacian of Gaussian (LoG) uses a smoothing filter combined with a Laplacian operator to reduce noise. Alternative methods are focused on local statistical computation based on nonlinear transformations. The purpose is to analyze and classify each image pixel as an edge. However, differential techniques are mostly used and easy to implement. Recently, a sophisticated approach to edge detection integrates Convolutional and Deep Neural Networks (CNN/DNN) because of the increasing number of computer vision applications. However, this approach is sensitive to noise, database selection with high complexity implementation in a real-time computer vision system [18]–[22]. In Table 1, we summarize edge detection techniques with applications.

https://doi.org/10.26555/ijain.v7i2.651

Table 1. Application examples of edge detection techniques.

Application	Method
Face Recognition [2]	Line Edge Map
Vision systems [3]	DNN
Machine vision [4]	Multilevel Fuzzy Edge Detection (FMFED)
Medical image and Robotics [5]	LoG
Medical Images [6]	Statistical
Texture Analysis [7]	Roberts, Prewitt, Canny
Brain Tumor Segmentation [8]	Sobel
Robotics, medical imaging [9]	DNN
Identification interpretation [10]	Fractional Derivatives
Optics [11]	Spin Hall Effect
State of forests analysis [12]	Canny
Satellite Images [13]	Prewitt, Kirsch, LoG, morphological
Object extraction [14]	Fuzzy
Mealtime Face Recognition [14]	LoG
Image processing [15]	Robert, Sobel, Prewitt and Canny
MRI Medical Images [16]	DNN
Cataract Detection [18]	Canny
Synthetic Aperture Radar [19]	Oriented Gaussian filter
Human discrimination objects [20]	Prewitt
Computed tomography [21]	DNN
Meta-materials and Meta-surfaces [22]	Optical edge

However, noise is a crucial problem in edge detection and contours recovery. It significantly reduces performances of high-level processing, such as image recognition and classification [1], [2], [5], [6]. It is because conventional edge detectors behave as high-pass filtering, which is insufficient to remove low-pass noised content information. In gradient and Laplacian operators, the edge algorithms focus on pixel intensities differences and zero-crossing detection, respectively. They are much sensitive to noise amplitude resulting in false edge detection. This paper aims to develop a novel and robust approach based on Gaussian binomial filters for edge detection in section 2. The proposed method efficiency, speed, and reproducibility are shown in section 3.

2. Method

2.1. Gaussian Binomial filter

We introduce in this section a novel approximation of Binomial Coefficients (BC) [23] based on the Gaussian function. Firstly, the Euler linearization of the powers of a half-period of the sine function $g_s(t)$ is given by Eq. 1. Which ωk is a pulsation for corresponding harmonic k, ω_0 is a fundamental pulsation expressed by the natural frequency f_0. It leads to

$$g_s(t) = sin^n(\omega_0 t) = \frac{1}{(2i)^n}\sum_{k=0}^{n}(-1)^{n-k}\binom{n}{k}\exp(i\omega_k t), \tag{1}$$

$$\omega_0 = 2\pi f_0,\ \omega_k = (2k-n)\omega_0, 0 \leq t \leq \frac{1}{2f_0}$$

where $\binom{n}{k}$ defines the binomial coefficients, are defined for each n, k positive integers expressed by

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}, n \geq k \geq 0 \in N \tag{2}$$

Let us now compute Fourier Transform (FT) of the powers of a half-period of the cosine function. Where $\delta(f)$ denotes Dirac distribution in the frequency domain, $f_k = \omega_k/2\pi$ is the frequency for given k harmonic. Let $G_s(f)$ denotes FT of $g_s(t)$ expressed by

$$G_s(f) = \frac{1}{(2i)^n}\sum_{k=0}^{n}(-1)^{n-k}\binom{n}{k}\delta(f-f_k), \tag{3}$$

while the Fourier coefficients $a_s(k)$ related to $\mathrm{G}_s(f)$ is determined by

$$G_s(f) = \sum_{k=0}^{n} a_s\,(k)\delta(f - f_k),\, a_s(k) = \frac{(-1)^{n-k}}{(2i)^n}\binom{n}{k} \tag{4}$$

Therefore, the Fourier coefficients Modulus may express only by $\binom{n}{k}$ yields

$$\|a_s(k)\| = \binom{n}{k}/2^n \tag{5}$$

Moreover, as stated in [24], we can obtain a precision approximation of the Gaussian function in terms of a sinusoidal function. On the other hand, it well knows that the FT of the Gaussian function is another Gaussian in the frequency domain. Then, the discrete binomial coefficients $\binom{n}{k}$ with fixed n as a function of k may be approximated by using a discrete Gaussian function. This function featured by (μ, σ^2) parameters, mean value μ and variance $\sigma 2$. In the following, we compute Gaussian parameters for given $\binom{n}{k}$ binomials coefficients. We consider $g(k)$ discrete Gaussian function expressed by

$$g(k) = A exp\left[-\frac{1}{2\sigma^2}(k - \mu)^2\right], A = \frac{1}{\sigma\sqrt{2\pi}}, 0 \le k \le n \tag{6}$$

The $\|\mathrm{a_s(k)}\|$ maximum occurs in the middle of $[0, n]$ for k varying from 0 to n, especially when $k = n/2$ in Fig. 1.

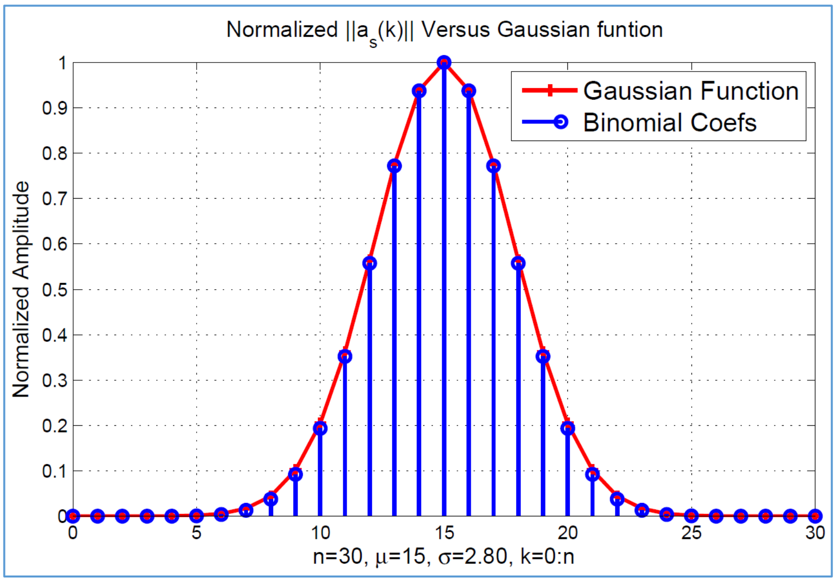

Fig. 1. Example of Gaussian function and normalized binomial coefficients $\binom{n}{k}$ with n=30, $0 \le k \le n$ $\mu = 15, \sigma = 2.80$.

Then, $\mu = n/2$. The (σ) is obtained at $k = [\mu] = [n/2]$ from Fourier coefficients, where [ ] is integer part yields

$$\sigma = \frac{1}{\sqrt{2\pi}\binom{n-1}{[\mu]}2^{n-1}}, \mu = \frac{n}{2} \tag{7}$$

Let $\mathrm{g_n(t)}$denotes the normalized Gaussian function $(A = 1)$ given by

$$g_n(t) = \exp\left[-\frac{1}{2\sigma^2}(t - \mu)^2\right] \tag{8}$$

We, therefore, have the proposed BC approximation based on the Gaussian function given by

$$\binom{n}{k} \approx \begin{cases} \binom{n}{\mu} exp\left(-\frac{k_\mu^2}{2\sigma^2}\right) = \binom{n}{\mu} g_n(k), 0 \leq k \leq n \\ \sigma = \frac{1}{\sqrt{2\pi}\binom{n-1}{[\mu]}2^{n-1}}, \mu = \frac{n}{2}, k_\mu = k - \mu \end{cases} \tag{9}$$

For bi-dimensional binomial formulation, one uses the combination of multiples One-dimensional property of Gaussian. Let $\mathrm{g}_2(\mathrm{k},\mathrm{l})$ denotes the discrete 2D filter and $\mathrm{g}_{\mathrm{b}2}(\mathrm{k},\mathrm{l})$ their binomial approximation. Let (k, l) is Cartesian positive integer indexes with $0 \ \leq k, l \ \leq n. \mathrm{g}_2(\mathrm{k},\mathrm{l}))$ filter is a square matrix with $(n+1) \times (n+1)$ dimensions. Then we have

$$\mathrm{g}_{\mathrm{b}2}(\mathrm{k},\mathrm{l}) = \frac{\binom{\mathrm{n}}{\mathrm{k}}\binom{n}{l}}{\binom{n}{\mu}^2}, 0 \leq k, l \leq n \approx \mathrm{g}_2(\mathrm{k},\mathrm{l}) = \mathrm{g}(\mathrm{k}) \times \mathrm{g}(\mathrm{l}) \approx \exp\left[-\frac{1}{2\sigma^2}\left(k_\mu^2 + l_\mu^2\right)\right], \tag{10}$$

where $k_\mu = k - \mu$, and $l_\mu = l - \mu$.

2.2. Proposed edge detection architecture

This subsection describes the proposed scheme to perform edge detection based on Gaussian binomial filter, which consists of four stages shows in Fig. 2. This involves linear low-pass Gaussian filters, linear normalization, and nonlinear operations such as absolute subtraction and mathematical morphological.

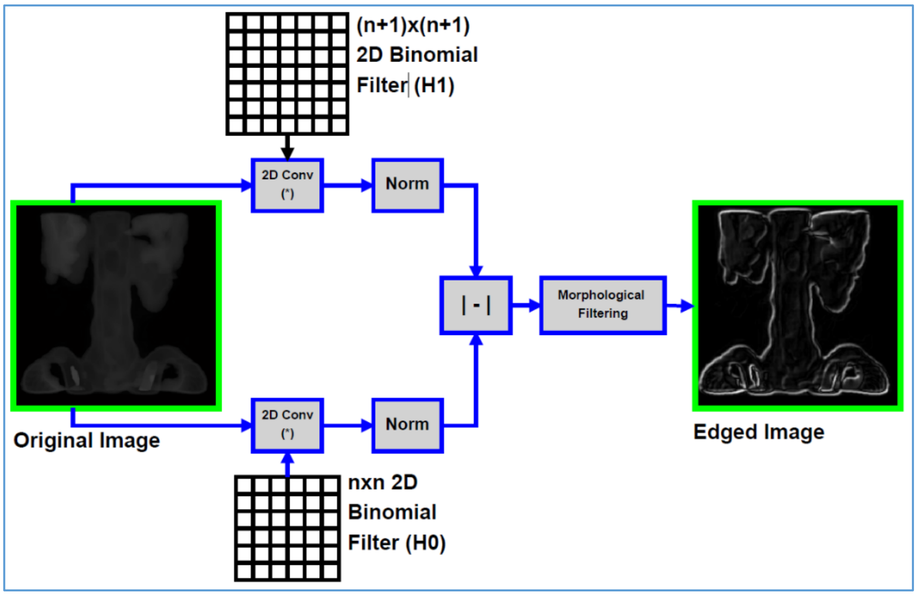

Fig. 2. Proposed edge detection architecture

- **Smoothing filters**: Dual symmetric stages which consist of dual 2D Gaussian binomial filters H_0 and H_1 with n and $n+1$ lengths, respectively. The output $y(i,j)$ is the 2D discrete convolution operator (*) withinput original image (I): $y(i,j) \leftarrow (I * H)(i,j)$.
- **Normalization function**: This stage performs global normalization of smoothed image $y(i,j)$. It divides smoothed image by absolute maximal value of $y(i,j)$: $yn(i,j) \leftarrow \frac{y(i,j)}{\max(|y|)}$. Image contrast is then focused in $[0, 1]$ for both stages.
- **Subtraction**: The aim of this stage is to compute absolute residual images between primary and secondary stages. The absolute difference operation is used to detect image edge and contours.
- **Morphological filtering**: Post-processing for image pooling and single pixels removing. We use in this wok morphological open fallowed by close operation using 7 structural element length. This block reduce snowflakes with a radius of less than 7 pixels by opening it with a disk-shaped structuring element having a 7-pixel radius.

Fig. 3 shows the pseudo-code to compute 2D Gaussian binomial filter $H(k, j)$. This implementation return $(n+1) \times (n+1)$ coefficients computed for (k, j) varying from 0 up to n according to equations (9) and (10).

```
Algorithm 0: 2D Gaussian binomial filter coefficients computation
Input: filter length $n$
Output: the computed 2D Gaussian binomial filter $H$
$\mu \leftarrow \frac{n}{2}$
$\sigma \leftarrow \frac{1}{\sqrt{2\pi}\binom{n-1}{\lceil\mu\rceil}2^{n-1}}, A \leftarrow \frac{1}{\sigma\sqrt{2\pi}}$
for $k \leftarrow 0$ to $n$ do
    $j \leftarrow 0$ to $n$ do
        $H(k,j) \leftarrow A^2 . g_n(k) . g_n(j)$
    end
end
Return $H$
```

Fig. 3. 2D Gaussian binomial filter coefficients computation algorithm.

3. Results and Discussion

3.1. Efficiency

In order to evaluate the efficiency of the proposed Gaussian binomial approximation, we make a comparison between true 1D binomial filter coefficients computation and its Stirling's approximation [25] given by

$$n! \approx \sqrt{2\pi n}\left(\frac{n}{e}\right)^n \tag{11}$$

In Fig. 4, we present the Mean Square Error (MSE) and relative error as 1D filter length n increases. In the result, we calculated the Gaussian binomial filter coefficients in range n=4 to 50, for each value of k rising from zero over n.

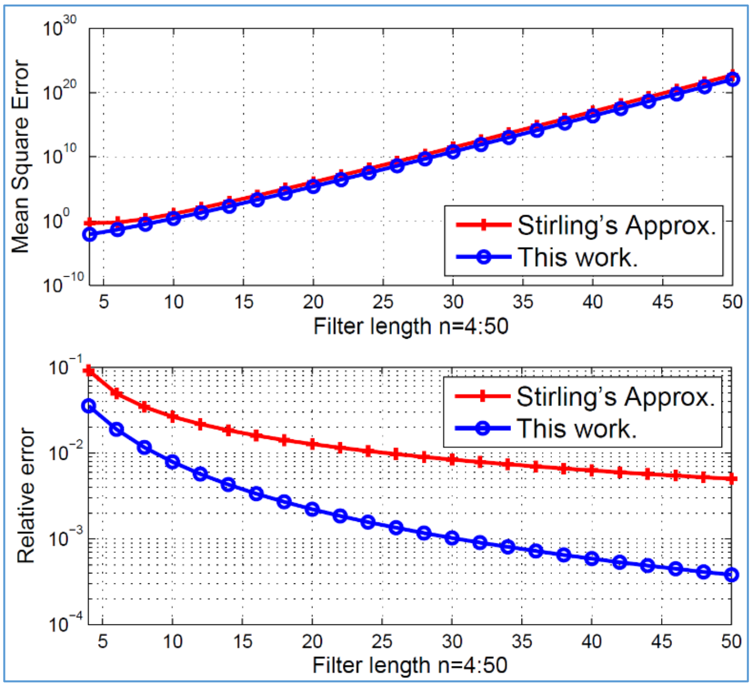

Fig. 4. Efficiency evaluation in calculation the 1D binomial filter coefficients for varying filter length n.

The proposed approximation has the best accuracy compared to Stirling's formulation. The relative error decreases rapidly for increasing filter size; it passed from $n \approx 2 \cdot 10^{-2}$ $(n = 4)$ $to \approx 3 \cdot 10^{-4}$ $(n = 50)$. However, Stirling's error decrease slightly over n: $2 \approx 10^{-3}$ $(n = 50)$. Examples of the results given by Stirling's and the proposed approximations of binomial coefficients $\binom{n}{k}$ are illustrated in Table 2, where n = 4 up to 40 and $k = \frac{n}{2} + 1$. In this results, $round$ () Matlab's function was used to recover approximations integer part. It can be seen in the table that the proposed approximation outperform those of Stirling's in the case of the smallest and largest binomial coefficients.

Table 2. Examples of few values of the stirling's approximation in comparison with the suggested approach.

n	k	True $\binom{n}{k}$	Stirling's	This work
4	3	4	4	4
6	4	15	16	15
8	5	56	58	55
10	6	210	216	208
12	7	792	809	787
14	8	3003	3059	2990
16	9	11440	11624	11401
18	10	43758	44380	43640
20	11	167960	170100	167587
22	12	646646	654116	645447
24	13	2496144	2522518	2492221
26	14	9657700	9751731	9644668
28	15	37442160	37780194	37398312
30	16	145422675	146646636	145273483
32	17	565722720	570182287	565210090
34	18	2203961430	2220299932	2202184616
36	19	8597496600	8657649529	8591290067
38	20	33578000610	33800434261	33556169292
40	21	131282408400	132108164711	131205133369

3.2. Speed and Reproducibility

This subsection shows simulation results of different methods implemented in Matlab in terms of speed and execution time reproducibility. We have implemented the following algorithms shows in Table 3 to calculate $(n + 1)$ coefficients of 1D Gaussian binomial filter for $k = 0$ up to n.

Table 3. Some approaches to calculate the binomial coefficients $\binom{n}{k}$ with corresponding matlab function.

Algorithm	Equation	Matlab function
1	$\frac{n!}{k!\,(n-k)!}$	Factorial()
2	$\prod_{i=n-k+1}^{n} \frac{i}{k!}$	nchoosek()
3	$\frac{n!}{k!\,(n-k)!}$	prod()
Stirling's	(11)	Matlab implementation
This work	(9)	Matlab implementation

Fig. 5 present binomial coefficients $\binom{n}{k}$ algorithms speed evaluation for varying filter length n from 4 up to 200. In this figure, we calculated $(n+1)$ binomial filter coefficients for each value of k from 0 over n (top plot in Fig. 5).

We used a 200000 iterations loop to compute a single binomial coefficient of 10 for each algorithm (bottom plot in Fig. 5). We observe that the proposed algorithm has the lowest execution time, come after Stirling's formulation. Algorithms 1 and 2 present the lowest speed. On the other hand, the proposed approach runtimes increase slightly for increasing filter length n better than Stirling's approximation. Because our approach used only exponential function implementation, multiplication numbers are proportional to filter length n for other approaches presented in Table 3. This makes our implementation speed less sensitive to filter length n.

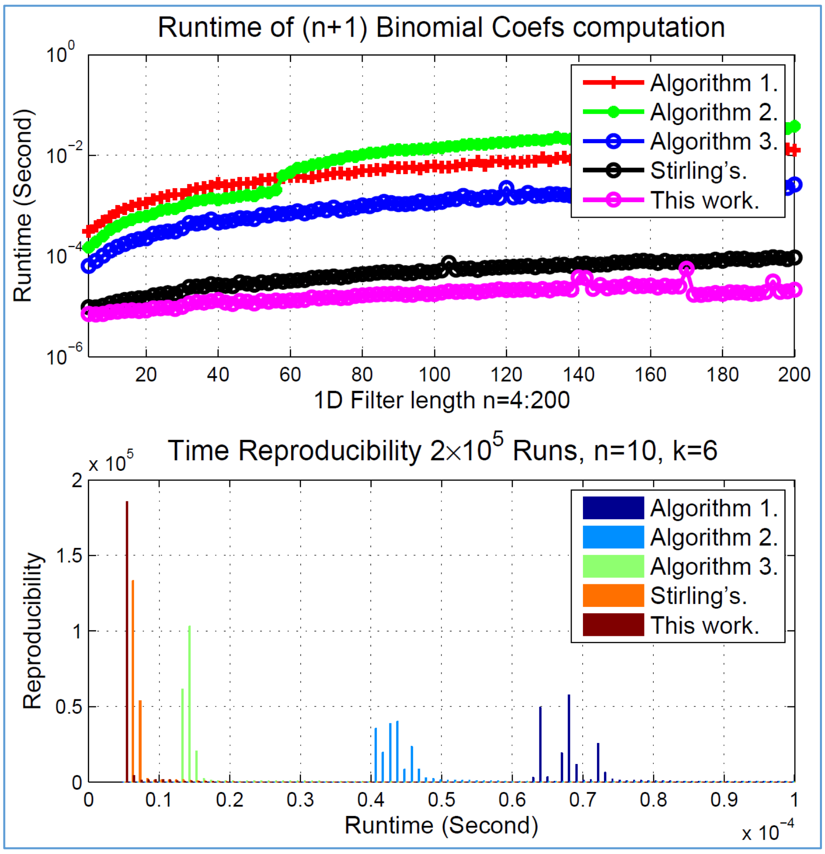

Fig. 5. Runtimes and time reproducibility evaluation for varying binomial filter length n

3.3. Edge Detection

This subsection covers qualitative and comparative analysis to verify the performance of the proposed algorithm. We discussed two parameters, Entropy and PSNR (Peak Signal to Noise Ratio). Firstly, the entropy E(I) indicates the information measurement in the image I introduced by Shannon [26]. On the other hand, it used to integrate pixel values repetition in the image. Increasing numbers of values denote (M) has high entropy and content, and lowest with a few values which are expressed by

$$E(I) = -\sum_{i=0}^{M} p_i \log p_i \tag{12}$$

Secondly, the PSNR is the ratio between the maximum power of the original image and their MSE with the filtered image. We consider I as the original input image, and If it is the output filtered image (M, N) are numbers of rows and columns, respectively. The low MSE induced high PSNR, indicating higher quality edge detection, while low PSNR results in low quality to edge recovering. The PSNR(dB) may be expressed by

$$MSE(I,I_f) = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left(I(i,j) - I_f(i,j)\right)^2,$$

$$PSN\ R(dB) = 10\log\left(\frac{max(|I|^2)}{MSE(I,I_f)}\right), \quad (13)$$

For analysis, we used primary 30 images on the BSDS500 database [27]. Each image has been resized to 128x128 and converted to grayscale with normalizing intensities. The 2D Gaussian binomial filters H_0 and H_1 parameters n and $n + 1$ have been fixed to 4 and 5, respectively, as shown in Fig. 2. While several types of edge detection approaches are applied to the similar original input image. Table 4 compares Prewitt, LoG, Laplacian, Sobel, and our method through entropy and PSNR. It has been shown that both PSNR and entropy parameters are slightly similar for Prewitt and Sobel and relatively higher compared to LoG and Laplacian approaches. It is clear in the table that the proposed algorithm produces the best performances improvements. This makes our algorithm an efficient method to edge recovering. Whereas The PSNR values remain relatively low because the original grayscale image has been used as a reference in PSNR computation, however, it is sufficient to verify algorithm performance.

Table 4. Qualitative evaluation of the proposed algorithm for edge detection through psnr(db) and entropy.

Image	ENTROPY					PSNR(dB)				
	This work	*Prewitt*	*LoG*	*Laplacian*	*Sobel*	*This work*	*Prewitt*	*LoG*	*Laplacian*	*Sobel*
*I*1	3.96	3.91	3.80	3.75	3.91	5.15	3.90	3.58	3.55	3.89
*I*2	3.94	3.33	3.27	3.17	3.33	6.01	4.26	4.00	4.02	4.25
*I*3	3.89	3.50	3.66	3.36	3.50	5.38	4.24	3.85	3.89	4.24
*I*4	3.62	2.95	2.49	2.26	2.95	3.37	2.27	1.99	2.06	2.26
*I*5	4.21	3.96	4.07	4.03	3.98	8.36	5.99	5.42	5.25	5.97
*I*6	4.14	3.78	4.13	4.17	3.76	6.88	5.39	4.90	4.91	5.38
*I*7	4.67	4.17	4.17	3.74	4.14	9.74	6.21	5.94	6.08	6.23
*I*8	4.04	3.22	3.41	3.37	3.25	4.63	2.63	2.24	2.15	2.61
*I*9	3.46	3.52	3.46	3.32	3.51	7.70	6.41	6.22	6.29	6.40
*I*10	4.87	4.50	4.65	4.68	4.51	9.67	6.38	5.62	5.38	6.36
*I*11	4.12	3.78	3.74	3.50	3.78	6.13	4.32	4.14	4.21	4.31
*I*12	4.89	4.15	4.30	4.34	4.18	9.49	6.05	5.50	5.37	6.04
*I*13	4.02	3.57	3.58	3.30	3.59	11.19	7.89	7.40	7.72	7.88
*I*14	4.46	4.05	4.38	4.32	4.09	10.22	7.28	6.57	6.56	7.25
*I*15	3.88	3.33	3.24	3.17	3.34	11.05	8.79	8.55	8.63	8.78
*I*16	4.58	3.86	4.23	4.16	3.89	6.94	4.79	4.16	4.32	4.79
*I*17	4.11	3.58	3.84	3.89	3.62	6.59	5.09	4.75	4.77	5.09
*I*18	4.49	4.57	4.41	4.44	4.59	7.44	5.12	4.69	4.69	5.10
*I*19	4.18	3.83	3.91	3.93	3.84	7.23	5.51	5.17	5.15	5.50
*I*20	4.33	3.43	4.13	4.00	3.46	6.87	5.26	4.78	4.79	5.25
*I*21	3.83	3.98	4.13	4.09	3.96	3.24	2.10	1.74	1.82	2.09
*I*22	4.36	4.02	4.23	4.04	4.04	8.71	6.37	5.86	5.89	6.35
*I*23	4.80	4.12	4.56	4.45	4.16	9.85	6.41	5.34	5.55	6.39
*I*24	4.54	3.98	4.21	4.28	4.04	10.96	7.43	6.54	6.24	7.37
*I*25	4.29	3.94	4.33	4.27	3.96	8.17	6.19	5.46	5.40	6.19
*I*26	4.02	3.89	3.86	3.87	3.94	7.17	5.59	5.30	5.29	5.59
*I*27	4.46	3.91	4.28	4.05	3.89	11.16	8.02	7.04	7.38	8.00
*I*28	3.62	3.41	3.42	3.45	3.39	7.10	5.81	5.60	5.64	5.81
*I*29	4.26	4.29	3.88	3.77	4.28	6.51	4.72	4.58	4.64	4.73
*I*30	3.88	3.80	3.89	3.86	3.82	6.81	5.26	4.94	4.96	5.25

In Fig. 6, we show some examples of edge detection methods in comparison with the proposed approach. Firstly, the original images doesn't corrupt by noise. These images level are normalized, ranged $\in [0, 1]$, and resized to 512x512. The Gaussian binomial filter size n is set to 8. The result shows that the proposed approach and LoG algorithm exhibit the best performances. They provide good sharpness and thickness of the edges because of the Gaussian kernel efficiency and smoothing properties. The proposed approach provides high-quality edge detection. They reduce impulsion noise induced by the subtraction stage in Fig. 2 because of a morphological filter. However, others methods have acceptable quality.

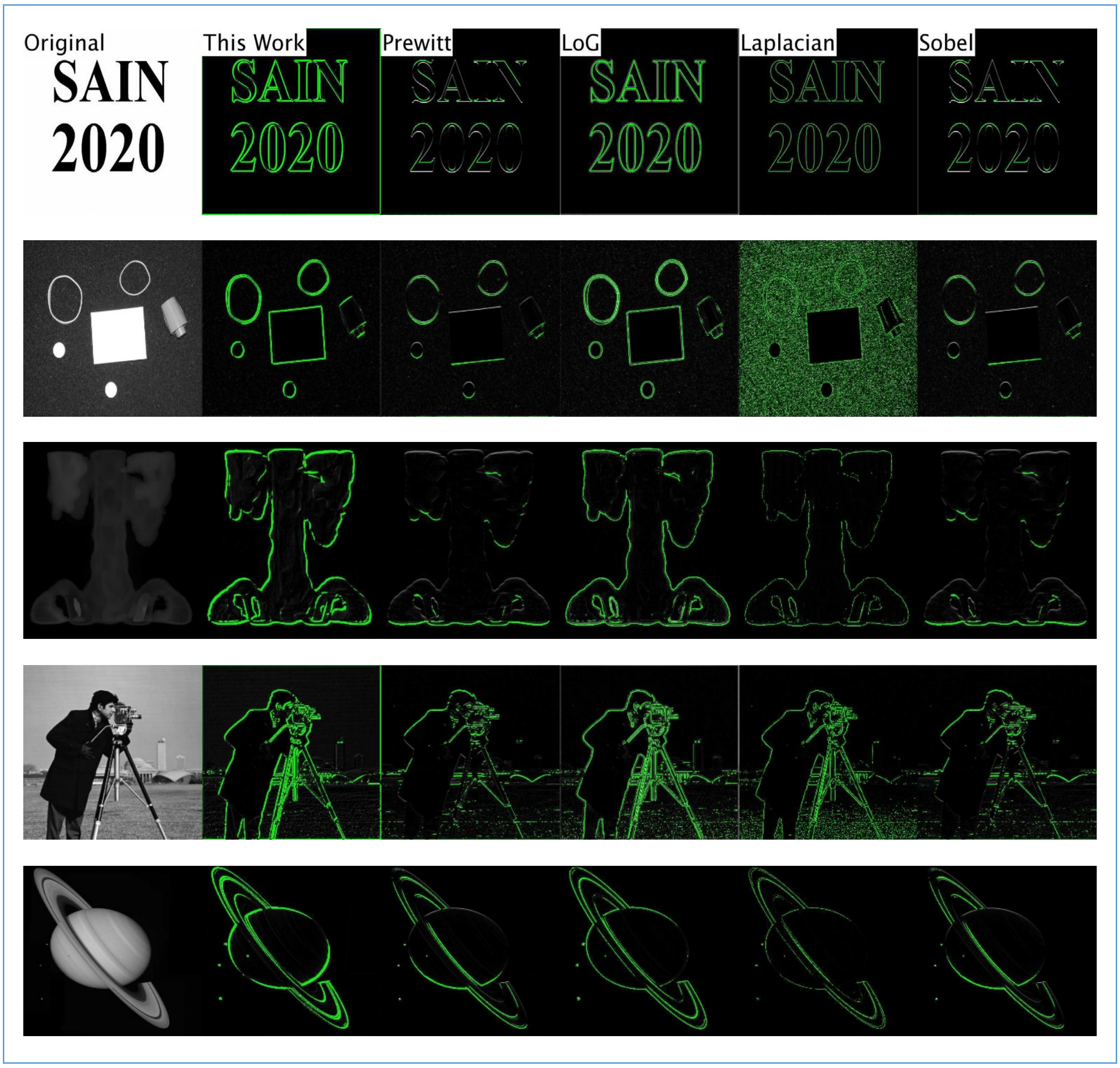

Fig. 6. Edge detection examples with Prewitt, LoG, Laplacian, Sobel operators in comparison with the proposed approach (from left to right).

Secondly, we evaluate algorithm robustness against noise. We use similar parameters, as shown in Fig. 6. Additive Gaussian noise with $\sigma = 10\%$ has been applied to the original images. Simulation results have been illustrated in Fig. 7. It is clear in the figure that the proposed approach exhibit better result followed by LoG method under noisy conditions. In comparison, Prewitt and Sobel provide average performance. Laplacian yields poor quality to edge recovering. Again, the proposed approach maintains a good quality and is less sensitive to noise. This is due to the use of double low-pass Gaussian filters by smoothing the noised image. These filters reduce noise slightly before edge detection operation as shown in Fig. 7. In the other hand, morphological filter remove individual pixel noise after subtraction operation.

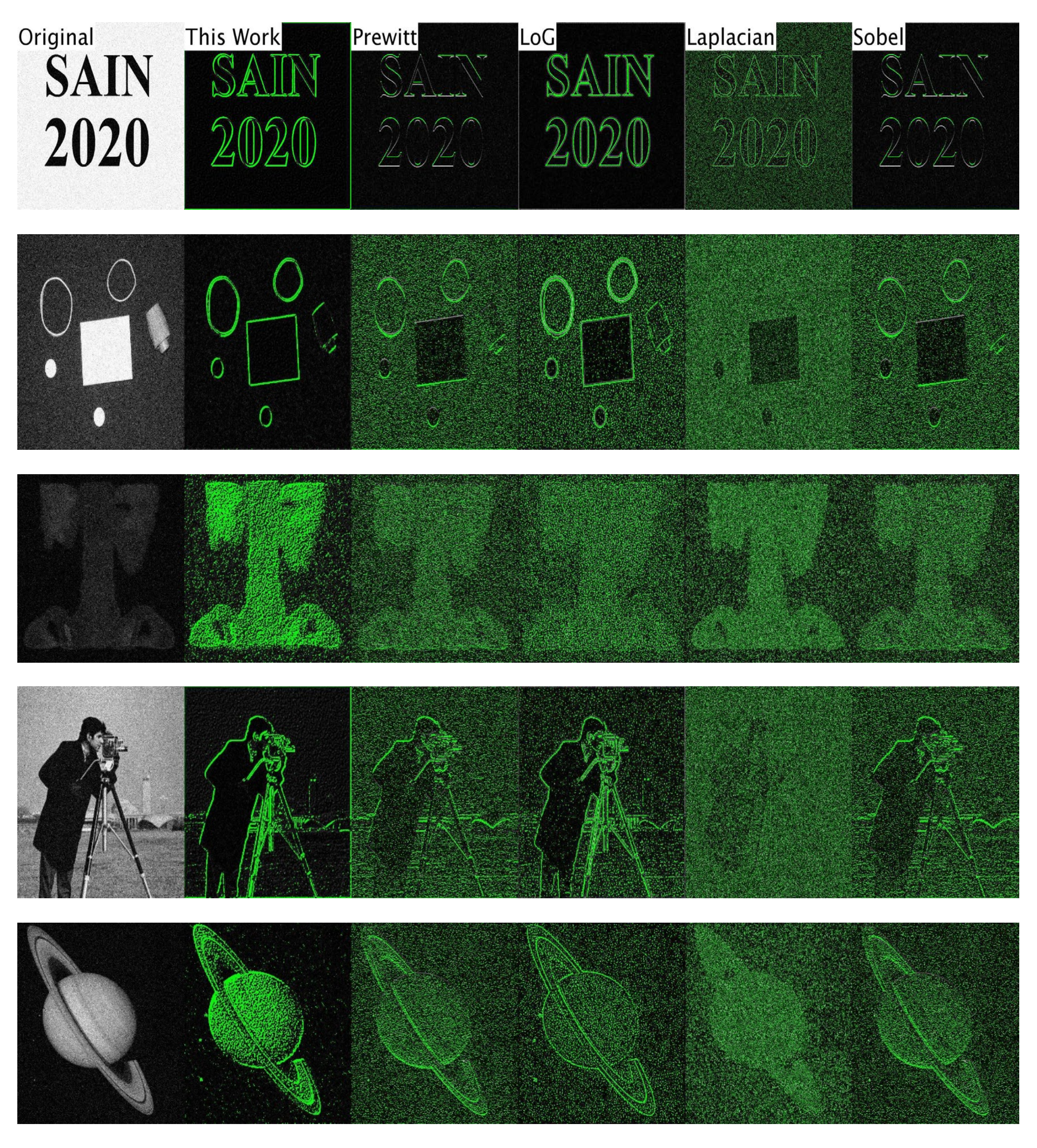

Fig. 7. Noised image edge detection examples ($\sigma = 10\%$).

4. Conclusion

This paper describes a novel image edge detection approach, which may be outlined in four points. First, we perform a low-pass spatial filter using dual 2D Gaussian binomial filters to smooth the image. Second, the normalized filtered images are combined using the absolute difference operator in the spatial domain. Finley, the morphological filter is performed to reduce the impulse noise in images. The proposed approach improved a Significant advantage of the Gaussian binomial filter in terms of speed and efficiency compared to other known methods. Moreover, we show that the suggested edge detection architecture is much simpler and easy to implement. A real-time edge detection implementation based on FPGA (Field-Programmable Gate Array) or GPU (Graphics Processing Unit) is an issue that deserves further investigation.

Acknowledgment

The authors gratefully acknowledge the support of Bureau de Recherche Ge´ologiques et Minie`res (BRGM) for post-doc funding. The authors would like to thank the reviewers for their criticisms, comments, and suggestions.

Declarations

Author contribution. All authors contributed equally to the main contributor to this paper. All authors read and approved the final paper.

Funding statement. None of the authors have received any funding or grants from any institution or funding body for the research.

Conflict of interest. The authors declare no conflict of interest.

Additional information. No additional information is available for this paper.

References

[1] T. X. -, "A Novel Image Edge Detection Algorithm based on Prewitt Operator and Wavelet Transform," *Int. J. Adv. Comput. Technol.*, vol. 4, no. 19, pp. 73–82, Oct. 2012, doi: 10.4156/ijact.vol4.issue19.10.

[2] Yongsheng Gao and M. K. H. Leung, "Face recognition using line edge map," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 6, pp. 764–779, Jun. 2002, doi: 10.1109/TPAMI.2002.1008383.

[3] P. Dollar and C. L. Zitnick, "Structured Forests for Fast Edge Detection," in *2013 IEEE International Conference on Computer Vision*, 2013, pp. 1841–1848, doi: 10.1109/ICCV.2013.231.

[4] J. Wu, Z. Yin, and Y. Xiong, "The Fast Multilevel Fuzzy Edge Detection of Blurry Images," *IEEE Signal Process. Lett.*, vol. 14, no. 5, pp. 344–347, May 2007, doi: 10.1109/LSP.2006.888087.

[5] Xin Wang, "Laplacian Operator-Based Edge Detectors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 5, pp. 886–890, May 2007, doi: 10.1109/TPAMI.2007.1027.

[6] A. Piórkowski, "A Statistical Dominance Algorithm for Edge Detection and Segmentation of Medical Images," 2016, pp. 3–14. doi: 10.1007/978-3-319-39796-2_1

[7] Ş. Öztürk and B. Akdemir, "Comparison of Edge Detection Algorithms for Texture Analysis on Glass Production," *Procedia - Soc. Behav. Sci.*, vol. 195, pp. 2675–2682, Jul. 2015, doi: 10.1016/j.sbspro.2015.06.477.

[8] A. Aslam, E. Khan, and M. M. S. Beg, "Improved Edge Detection Algorithm for Brain Tumor Segmentation," *Procedia Comput. Sci.*, vol. 58, pp. 430–437, 2015, doi: 10.1016/j.procs.2015.08.057.

[9] X. Song, X. Zhao, L. Fang, H. Hu, and Y. Yu, "EdgeStereo: An Effective Multi-task Learning Network for Stereo Matching and Edge Detection," *Int. J. Comput. Vis.*, vol. 128, no. 4, pp. 910–930, Apr. 2020, doi: 10.1007/s11263-019-01287-w.

[10] V. Kothapalli, S. Arora, and M. Hanmandlu, "Edge detection using fractional derivatives and information sets," *J. Electron. Imaging*, vol. 27, no. 05, p. 1, Jun. 2018, doi: 10.1117/1.JEI.27.5.051226.

[11] T. Zhu *et al.*, "Generalized Spatial Differentiation from the Spin Hall Effect of Light and Its Application in Image Processing of Edge Detection," *Phys. Rev. Appl.*, vol. 11, no. 3, p. 034043, Mar. 2019, doi: 10.1103/PhysRevApplied.11.034043.

[12] D. Panagiotidis, A. Abdollahnejad, P. Surový, and K. Kuželka, "Detection of fallen logs from high-resolution UAV images," *New Zeal. J. For. Sci.*, vol. 49, Mar. 2019, doi: 10.33494/nzjfs492019x26x.

[13] R. Bausys, G. Kazakeviciute-Januskeviciene, F. Cavallaro, and A. Usovaite, "Algorithm Selection for Edge Detection in Satellite Images by Neutrosophic WASPAS Method," *Sustainability*, vol. 12, no. 2, p. 548, Jan. 2020, doi: 10.3390/su12020548.

[14] A. Ighoyota Ben, O. Nicholas.O., and O. Charles O., "Optimum Fuzzy based Image Edge Detection Algorithm," *Int. J. Image, Graph. Signal Process.*, vol. 9, no. 4, pp. 44–55, Apr. 2017, doi: 10.5815/ijigsp.2017.04.06.

[15] E. Dong, K. Li, and J. Tong, "FPGA Based Design and Implementation of Improved Edge Detection Algorithm using LOG Operator," in *2018 IEEE International Conference on Mechatronics and Automation (ICMA)*, 2018, pp. 2092–2096, doi: 10.1109/ICMA.2018.8484676.

[16] A. Karnam, D. R. Kulkarni, K. P. Sunagar, N. G. Revankar, and M. M. Dixit, "Analysis of Various Edge Detection Techniques," *Bonfring Int. J. Res. Commun. Eng.*, vol. 6, no. Special Issue, pp. 10–12, Nov. 2016, doi: 10.9756/BIJRCE.8190.

[17] T. S. Gunawan, I. Z. Yaacob, M. Kartiwi, N. Ismail, N. F. Za'bah, and H. Mansor, "Artificial Neural Network Based Fast Edge Detection Algorithm for MRI Medical Images," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 7, no. 1, p. 123, Jul. 2017, doi: 10.11591/ijeecs.v7.i1.pp123-130.

[18] H. H. Abbass and Z. R. Mousa, "Edge detection of medical images using Markov basis," *Appl. Math. Sci.*, vol. 11, pp. 1825–1833, 2017, doi: 10.12988/ams.2017.75160.

[19] R. Sigit, E. Triyana, and M. Rochmad, "Cataract Detection Using Single Layer Perceptron Based on Smartphone," in *2019 3rd International Conference on Informatics and Computational Sciences (ICICoS)*, 2019, pp. 1–6, doi: 10.1109/ICICoS48119.2019.8982445.

[20] W. Jing, T. Jin, and D. Xiang, "SAR image edge detection with recurrent guidance filter," *IEEE Geosci. Remote Sens. Lett.*, vol. 18, no. 6, pp. 1064–1068, 2020. doi: 10.1109/LGRS.2020.2990688

[21] H. Ye, B. Shen, and S. Yan, "Prewitt edge detection based on BM3D image denoising," in *2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, 2018, pp. 1593–1597. doi: 10.1109/IAEAC.2018.8577643

[22] M. Gholizadeh-Ansari, J. Alirezaie, and P. Babyn, "Deep Learning for Low-Dose CT Denoising Using Perceptual Loss and Edge Detection Layer," *J. Digit. Imaging*, vol. 33, no. 2, pp. 504–515, Apr. 2020, doi: 10.1007/s10278-019-00274-4.

[23] F.-Z. Zhao and T. Wang, "Some results for sums of the inverses of binomial coefficients," *Integers*, vol. 5, no. 1, p. A22, 2005. Available at: Google Scholar.

[24] E. H. Ait Mansour and S. Barth, "Efficient Approximation of Gaussian Function for Signal and Image Processing Applications," in *2019 Signal Processing Symposium (SPSympo)*, 2019, pp. 1–6, doi: 10.1109/SPS.2019.8882020.

[25] S. Eger, "Stirling's approximation for central extended binomial coefficients," *Am. Math. Mon.*, vol. 121, no. 4, pp. 344–349, 2014. doi: 10.4169/amer.math.monthly.121.04.344

[26] R. Maini and H. Aggarwal, "Peformance evaluation of various speckle noise reduction filters on medical images," *Int. J. Recent Trends Eng.*, vol. 2, no. 4, p. 22, 2009. Available at: Google Scholar.

[27] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik, "Contour Detection and Hierarchical Image Segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, May 2011, doi: 10.1109/TPAMI.2010.161.

Supplementary Material

Matlab codes of these results are available from the correspond authors upon request.