TẠP CHÍ KHOA HỌC ĐẠI HỌC TÂN TRÀO
ISSN: 2354 - 1431
http://tckh.daihoctantrao.edu.vn/

DISCUSSION ON LOG - BASED OPERATORS FOR REAL-TIME TEXT DETECTION

Dinh Cong Nguyen[1,*], *PhD*

[1] *Faculty of Information Technologies and Communication, Hong Duc University.*

No 565 Quang Trung Street - Dong Ve Ward - Thanh Hoa City.

[*] *Email: nguyendinhcong@hdu.edu.vn*

Article info

Recieved:
20/9/2020
Accepted:
10/12/2020

Keywords:
Text detection, LoG operator, stroke model, almost-Gaussian.

Abstract:

In this paper methods for real-time text detection in camera-based images are presented, having a particular focus on the Laplacian of Gaussian (LoG) operators. These methods are discussed with a specific focus on the aspects of computational complexity and robustness. Some illustrative results and baseline experiments are given to characterize the methods. Moreover, we provide comments on the improvements of the methods to the text detection problem.

1. Introduction

The problem of text processing in natural images is a core topic in the fields of image processing (IP) and pattern recognition (PR). Recent state-of-the-art methods and international contests can be found in [1] and [2], respectively. A key problem is to make the methods being time-efficient in order to embed into devices to support real-time processing [3] [4] [5].

The real-time systems in the [1] [3], [4] [6], [7], [8], [9], [10] apply the strategy of two stages composing of detection and recognition. The detection localizes the text components at a low complexity level and groups them into text candidate regions before classification. The objective is to get a perfect recall for the detection with a maximum precision for optimization of the recognition. The two-stage strategy differs from the end-to-end strategy, that applies template/feature matching with classification using high-level models for text entities [11]. The text elements in natural images present specific shapes with elongation, orientation and stroke width variation, etc. as illustrated in Figure 1. This makes difficult the detection problem. Therefore, various approaches have been investigated in the literature to design real-time and robust methods.

The recent works on the topic drive the text processing as a blob detection problem with the maximally stable extremal regions (MSER) [3], [5] and the LoG-based operators [6], [8], [10], [4], [12]. MSER looks for the local intensity extrema and applies a watershed-like segmentation algorithm for detection. The algorithm is processed in a linear time complexity. It copes well with background/foreground regions but is sensitive to blurring. The Laplacian of Gaussian (LoG) operator is a blob detector, but can be tuned to a stroke detector with scale and orientation for better characterization of text elements [10], [4]. Recently, LoG estimators have been proposed at a linear-time complexity [13], [14] making the operator competitive with MSER.

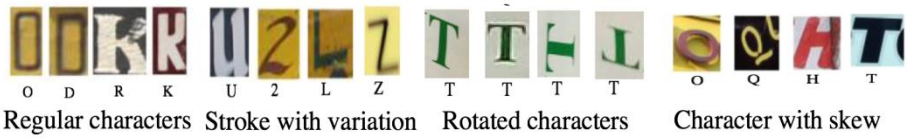


Figure 1. Example of text elements/characters in images [12]

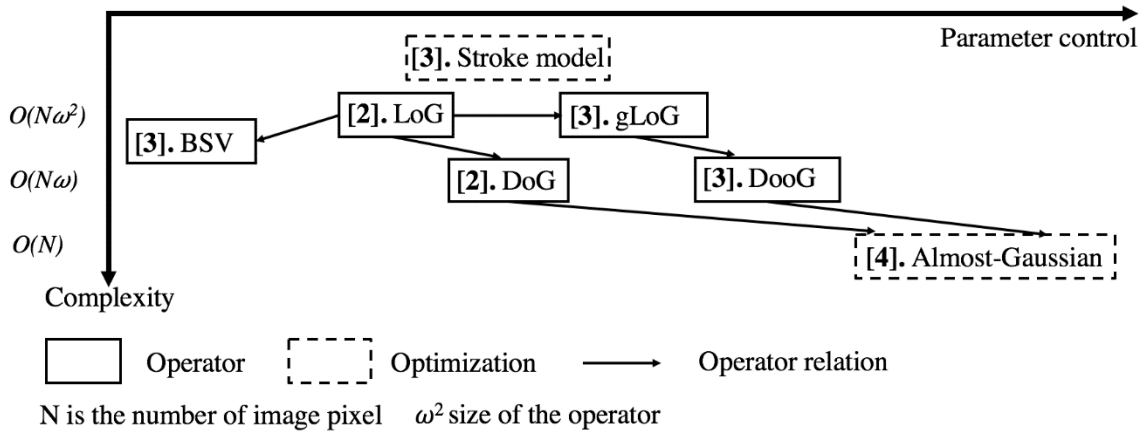


Figure 2. A characterization of different methods in the paper.

This paper gives several key contributions.

- We focus only on text detection phase, we bring together all the recent trends of the LoG-based operators dealing with adaptation to the text detection problem.
- We discuss and concentrate on how to optimize these operators with real-time constraints. Figure 2 characterizes different methods in the paper with key sections.
- The baseline LoG operator is reformulated into the stroke model paradigm and generalized LoG (gLoG) for scale and adaptive rotation. Optimization is obtained with the difference of Gaussian (DoG) and difference-of-offset-Gaussian (DooG) reformulation of the operators, then estimation with almost-Gaussian components.

The rest of this paper illustrated in Figure 2 is as follows. Section 2 gives an introduction to LoG operators for blob detection. The adaptation of the LoG operator to stroke/text detection will be introduced in section 3. In section 4, real-time LoG operators will be discussed. At last, section 5 gives the conclusions and perspectives. Figure 3 gives the meaning of symbols used in the paper.

Symbols	Meaning	Symbols	Meaning
f	An image function	σ_e	The standard deviation for edge detection
g	A Gaussian function	θ	Directional orientations
Π	A step function / box filter	a,b,c	Trigonometric functions
$\otimes$	The global convolution	$\wedge$	Estimator
$\nabla_x \nabla_y$	Offsets of Gaussian kernels	w	Stroke widthe parameter
σ, σ_s	The standard/optimum deviations	∇^2	Laplacian
h, h_s	Responses with stroke model	$df = f_x + f_y$	Total differentiation
g_x, g_{xx}	First/second derivatives	O	Complexity
ω	Size of operator	N	Size of image

2. Baseline LoG Operators

One of the standard approaches for differential blob detector is found by LoG based on the Gaussian function. The multivariate Gaussian function, with a vectorial notation, is given in Eq. (1).

$$g(p|\mu,\Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}}\sqrt{|\Sigma|}} e^{-\frac{1}{2}(p-\mu)^T\Sigma^{-1}(p-\mu)} \quad (1)$$

In the two-dimensional case, n = 2, *p* is a point and μ *is a* centroid. Σ is the diagonal covariance matrix with Σ^{-1} the inverse and |Σ| the determinant, where σ_x, σ_y are the standard deviations in *x*, *y*. Considering $\sigma_x = \sigma_y$, μ is null and a scalar notation, the Gaussian function Eq. (1) becomes Eq. (2).

$$g(x,y|\sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (2)$$

The LoG is a compound operator resulting of the Laplacian ∇^2 of $g(x,y|\sigma)$ Eq. (3).

$$\nabla^2 g(x,y|\sigma) = g_{xx}(x,y|\sigma) + g_{yy}(x,y|\sigma) = \frac{1}{2\pi\sigma^4}\left(\frac{x^2+y^2}{\sigma^2} - 2\right) e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (3)$$

The LoG-filtered image *h(x, y)* Eq. (4) is obtained by the global convolution between the initial image *f(x, y)* and the LoG operator $\nabla^2 g(x,y|\sigma)$.

$$h(x,y) = \nabla^2\big(g(x,y|\sigma) \otimes f(x,y)\big) = \nabla^2 g(x,y|\sigma) \otimes f(x,y) \quad (4)$$

LoG function can be approximated by means of DoG as Eq. (5) with relation among $(\sigma, \sigma_1, \sigma_2)$ as Eq. (6).

$$\nabla^2 g(x,y|\sigma) \approx g(x,y|\sigma_1) - g(x,y|\sigma_1) \quad (5)$$

$$\sigma^2 = \frac{\sigma_1^2\sigma_2^2}{\sigma_1^2 - \sigma_2^2} \ln\left(\frac{\sigma_1^2}{\sigma_2^2}\right) \quad (6)$$

where σ_1, σ_2 can be presented as $\sigma_2 = k\sigma_1$ with *k* a parameter, resulting in the DoG formulation Eq. (7).

$$g(x,y|\sigma_1) - g(x,y|k\sigma_1) = \frac{1}{2\pi\sigma_1^2}\left(e^{-\frac{x^2+y^2}{2\sigma_1^2}} - \frac{1}{k^2} e^{-\frac{x^2+y^2}{2(k\sigma_1)^2}}\right) \quad (7)$$

As the scale σ of LoG is relatively low, we tend to use LoG in order to detect edges with zero-crossing. In contrast, blob-like structures will be converged at some scales to local extrema when the scale σ increases [15]. As illustrated in Figure 4, this motivates application of the LoG operator for text [10] [4].

Original image Blob-based detection

Figure 4. Blob-based detection for text detection with a LoG operator with σ = 2.3.

3. The LoG Operators for Text Detection

The LoG operator has been applied in different works for text detection in [10] [4] [12] [14]. In this paper, we will explore recent trends on this topic dealing with adaptation of the operator to the text detection problem. This includes of the control of standard deviation parameters σ (stroke model [6] [10]) and LoG kernel reformulation [4].

3.1. The Stroke Model

A crucial problem with the LoG operator for blob detection is the control of the scale parameter σ [12]. When the object to detect is a text element/ character, the LoG operator can be driven as a stroke detector where the parameter σ is able to be

derived from the stroke width parameter *w*. This is presented as the stroke model in literature.

Figure 5 illustrates the model. The general idea is to look for the convolution response between a LoG-based operator and a stroke signal model as unit step function. We can express then the minimal/maximal derivatives of the convolution product. Assuming that these minimum/maximums are located at the center of the stroke *w/2*, we can present the standard deviation σ as a function $\sigma = f(w)$. These aspects will be developed here.

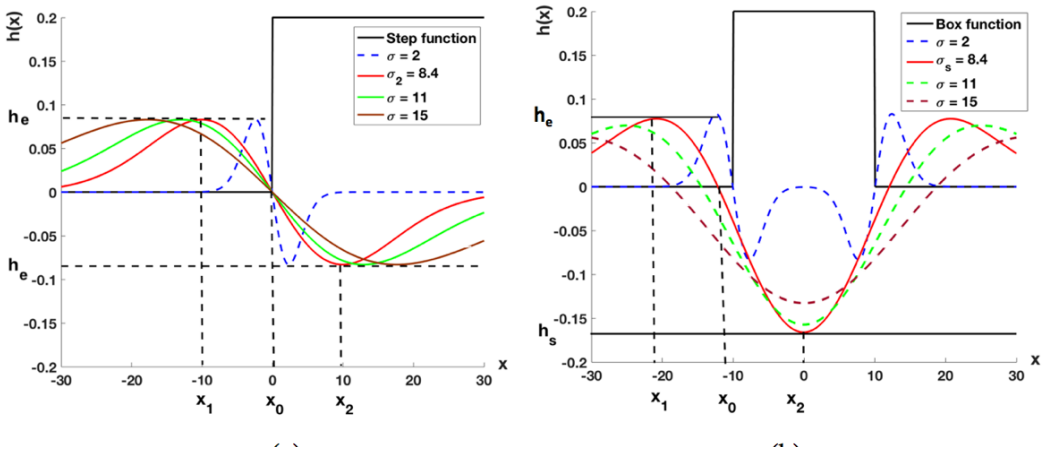


Figure 5. LoG responses at different scales to (a) a step function (b) a boxcar function [14].

Assuming the image signal as a function $a \otimes \Pi(x)$ (considering 1-D case as discussed in [10]) $\Pi(x)$ the step function Eq. (9) and *a* as a constant parameter, the convolution product with the LoG operator $\nabla^2 g(x)$ is given in Eq. (8).

$$h(x_0) = a(\Pi \otimes g)(x_0) = \int_{-\infty}^{+\infty} a\Pi(x_0 - x)\,\nabla^2 g(x)dx \quad (8)$$

$$\Pi(x_0 - x) = \begin{cases} 0 & if\ x < x_0 \\ 1 & otherwise \end{cases} \quad (9)$$

As $\Pi(x_0 - x)$ is located at x_0, the convolution product $\Pi(x_0 - x) \otimes \nabla^2 g(x)$ over x equals the summation $\nabla^2 g(x)$ at centered at x_0. Approximately $\nabla^2 g(x) = g(x|\sigma_1) - g(x|\sigma_2)$ as DoG function, the result of Eq. (8) is reformulated into Eq. (10).

$$h(x_0) = \int_{-\infty}^{+\infty} a(g(x|\sigma_1) - g(x|\sigma_2))\,dx \quad (10)$$

From derivative $h_x(x_0)$ of Eq. (10), the local extremal optimum is obtained as Eq. (11) with *k* a parameter.

$$x_{1,2} = \pm k\sigma\sqrt{\frac{2\ln k}{k^2 - 1}} \quad (11)$$

Discussion:

As given in Eq. (11) and shown in Figure 5(a), it is seen that $x_{1,2}$ locations are dependent on the σ parameter. With $x_2 = x_0 + w/2$ the middle of the stroke and goes to Eq. (11), we can get the optimum scale and operator response Eq. (12)

$$\sigma_s = \frac{w}{2k}\sqrt{\frac{k^2 - 1}{2lnk}} \qquad h_s = \frac{a}{2}\left(\text{erf}\left(\sqrt{\frac{lnk}{k^2 - 1}}\right) - \text{erf}\left(k\sqrt{\frac{lnk}{k^2 - 1}}\right)\right) \quad (12)$$

where erf(x) is the Gauss error function erf(x) = $\frac{2}{\pi}\int_0^x e^{-t^2}dt$. The optimum/extremal responses

(these aspects are not proven in the paper [10], but illustrated with experiments) of the DoG operator appear at the middle of the stroke *w/2* with a accurate scaling parameter σ_S. This response decreases while shifting the scaling parameter σ around σ_S optimum Figure 5(b).

3.2 The Generalized LoG Operator

The LoG (either DoG) operator has good performances in locating the middle of 2-D near circular blobs, with a proper standard deviation setting parameter σ_S. However, the operator is limited in detecting blobs with general elliptical shapes and is not able to estimate the orientation of the detected blobs. Indeed, the conventional LoG operator is rotational symmetric, i.e., the σ is set to be equal for both x and y coordinates. The Figure 6(a) illustrates this problem, as the character is rotated, variations appear in the stroke width resulting in the lowest responses of the operator

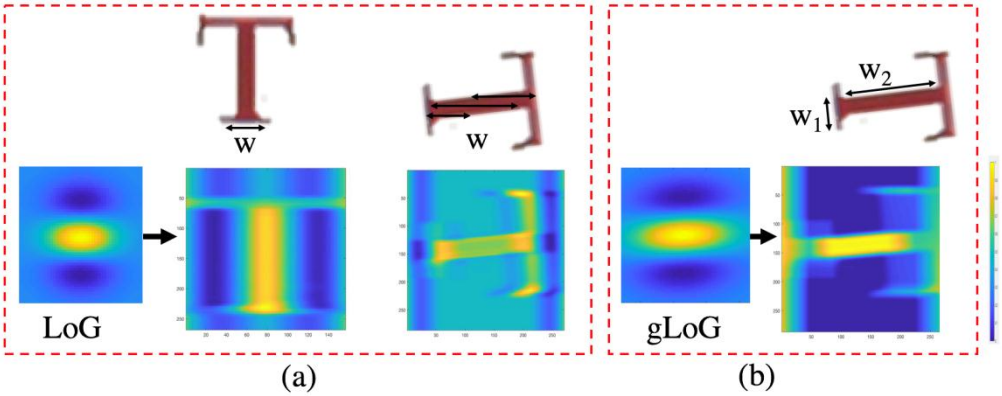


Figure 6. (a) LoG responses at scale σ_S = f(w) with a regular and a rotated character (b) gLoG response at scale σ_x = f(W_1), σ_y = f((W_1) with a rotated character.

To address this problem the LoG operator is generalized to detect elliptical and rotated shapes Figure 6(b). This makes the operator robust to the detection cases with rotation and shifts the operator for detection of Haar-like features. For simplification, we refer the generalized operator as gLoG as suggested in [15]. At best of our knowledge, only the paper [16] has investigated this issue for text detection. Recent contributions on the gLoG detector for natural images are found in [15].

Let us g(x, y| σ_x, σ_y, θ) as 2-D oriented Gaussian function with form as Eq. (13),

$$g(x,y|\sigma_x,\sigma_y,\theta) = \frac{1}{2\pi\sigma_x\sigma_y} e^{-(ax^2+2bxy+cy^2)} \qquad (13)$$

with *a*, *b* trigonometric functions to control the shape and the orientation with standard deviations σ_x, σ_y and orientation θ. The gLoG $\nabla^2 g(x,y|\sigma_x,\sigma_y,\theta)$ is obtained by Eq. (14) resulting from Eq. (13). The convolution products of gLoG with the given image will be used to determine the shape and the orientation of blobs.

$$\nabla^2 g(x,y|\sigma_x,\sigma_y,\theta) = g_{xx}(x,y|\sigma_x,\sigma_y,\theta) + g_{yy}(x,y|\sigma_x,\sigma_y,\theta) \qquad (14)$$

Discussion

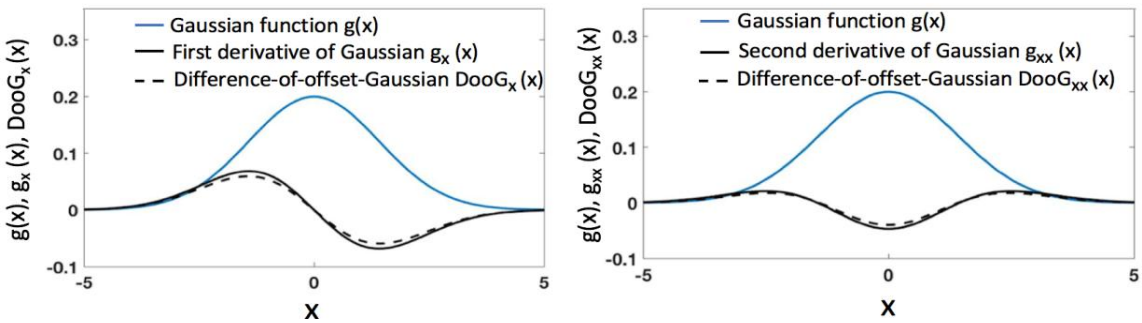
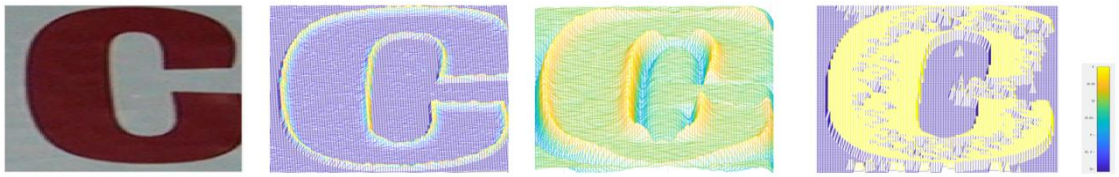


Figure 7. Approximations of (a) g_x with $DooG_x$ (b) $g_{xx}$with $DooG_{xx}$ reformulations.

For optimization, difference-of-offset-Gaussian (DooG) operator is considered, which was first introduced by Young [17]. Basically, DooG function is designed by using Eq. (13) with offset valuesΔ_x, Δ_y as the distance between two Gaussian kernels [18]. It could be explained that the derivatives of a Gaussian function are mathematically closely equal to discrete difference between Gaussian functions with relatively small offset distances in Figure 7. The first derivative in x dimension of the 2-D oriented Gaussian function Eq. (13) is given in Eq. (15), where *a, b, c* parameters are defined in Eq. (13). The DooG function Eq. (16) can approximate the Gaussian derivative function Eq. (15).



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

Figure 8. (a) a character, responses in color map of (b) the LoG operator (c) the BSV operator (d) the BSV after hysteresis thresholding.

$$g_x(x,y|\sigma_x,\sigma_y,\theta) = -2(ax+by)g(x,y|\sigma_x,\sigma_y,\theta) \quad (15)$$

$$DooG_x(x,y|\sigma_x,\sigma_y,\theta) = g(x+\Delta_x,y|\sigma_x,\sigma_y,\theta) - g(x-\Delta_x,y|\sigma_x,\sigma_y,\theta)$$
$$\approx \qquad\qquad \approx g_x(x,y|\sigma_x,\sigma_y,\theta) \quad (16)$$

The DooG operator can be extended to the second derivative from the x or y dimensions Eq. (17). These operators approximate the second order derivatives of Gaussian g_{xx}, g_{yy}.

$$DooG_{xx}(x,y|\sigma_x,\sigma_y,\theta) = g(x+2\Delta_x,y|\sigma_x,\sigma_y,\theta) - 2g(x+2\Delta_x,y|\sigma_x,\sigma_y,\theta) + g(x-2\Delta_x,y|\sigma_x,\sigma_y,\theta) \quad (17)$$

With $DooG_{xx}(x,y|\sigma_x,\sigma_y,\theta)$ and $DooG_{yy}(x,y|\sigma_x,\sigma_y,\theta)$ formulations, we can approximate the gLoG operator Eq. (14) as given in Eq. (18).

$$\nabla^2 g(x,y|\sigma_x,\sigma_y,\theta) = DooG_{xx}(x,y|\sigma_x,\sigma_y,\theta) + DooG_{yy}(x,y|\sigma_x,\sigma_y,\theta) \quad (18)$$

3.3 The BSV Operator

The BSV operator [4] is a LoG look-like operator for stroke detection. It differs from the blob-based strategy with LoG, that targets optimum response h_s Eq.(10) with the scale parameter σ_s Eq. (12). The operator processes as an edge detector with a zero-crossing operation, where the optimum scale for edge detection $\sigma_e \ll \sigma_s$. Whereas the LoG operator produces a strong response at an edge location and a null response in the in-between edge area Figure 8(b), the BSV operator still guaranties a no null response Figure 8(c). Then, similar to edge detector the stroke elements can be obtained with hysteresis thresholding Figure 8(d).

The BSV operator is close to Laplacian formulation Eq. (3). It results in the total differential d of an image function *f(x, y)* convolved with a *δ(x, y)* operator Eq. (19).

$$h(x,y) = d(\delta(x,y)) \otimes f(x,y) \approx \nabla^2 f(x,y) \quad (19)$$

Using the linearity property, the compound operator *BSV(x, y) = d(δ(x, y))* can be achieved in Eq. (20) with $\delta_x(x,y), \delta_y(x,y)$ as defined in Eq. (21). This operator is expressed from the the formulation of Biot-Savart law into an image convolution operator as described from original paper [4] in detail.

$$BSV(x,y) = d(\delta(x,y)) = \delta_x(x,y) + \delta_y(x,y) \quad (20)$$

$$\delta_x(x,y) = \frac{-x}{(x^2+y^2)^{3/2}} \qquad \delta_y(x,y) = \frac{-y}{(x^2+y^2)^{3/2}} \quad (21)$$

Discussion

A convolution with the *BSV* operator is close to a derivative product, but with specific steps and averaging. When a Gaussian averaging product is embedded Eq. (22), the BSV operator tends to produce a LoG look-like function as Eq. (23) with $g_x(x,y|\sigma),\ g_x(x,|\sigma)$ the Gaussian derivatives. Compared to the LoG, the BSV operator enhances the central part of the kernel that maintains a response in the in-between edge area

$$h(x,y) = d(g(x,y|\sigma) \otimes \delta(x,y)\ ) \otimes f(x,y) = BSV(x,y|\sigma) \otimes f(x,y) \quad (22)$$

$$BSV(x,y|\sigma) = g_x(x,y|\sigma) \otimes \delta_x(x,y) + g_y(x,y|\sigma) \otimes \delta_y(x,y) \quad (23)$$

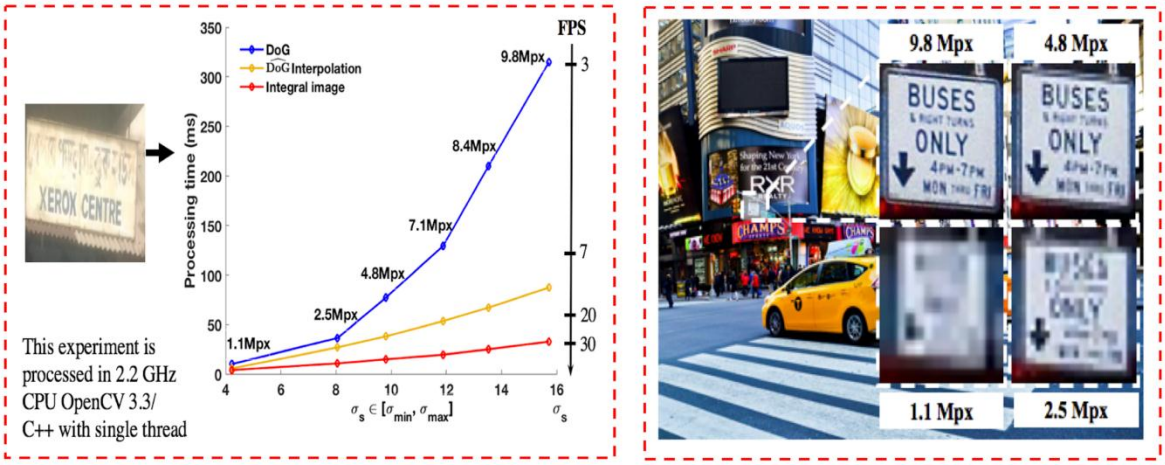
The compound operator *BSV(x, y)* of Eq. (20) is not separable. The real-time property is coming from the operator size, as we have $\sigma_e \ll \sigma_s$. However, optimization could be obtained with the non-compound form of the operator (these aspects are not discussed in [4]). The Gaussian derivatives $g_x(x,y|\sigma), g_y(x,y|\sigma)$ can be approximated with DooG operators Eq. (16) then almost-Gaussian function (see section 4).The $\delta_x(x,y), \delta_y(x,y)$ are functions close to Haar-like features that could be approximated with boxcar operators [13].

4 Discussion on Real-time LoG Operators

The baseline approach to process a LoG operator is the convolution product. The LoG function (3) is discretized to get a mask g of size ω × ω, applied in the product $h = f \otimes g$. The size of the mask ω^2 is dependent on the σ parameter (the typical size is 6σ for a full coverage of the function [19]), requiring a complexity O(Nω^2) with N the image size (in pixels). Optimization is obtained with the DoG function Eq. (5) that can be implemented with separable filters of size 1 × ω such as $h = f \otimes g_1 \otimes g_2$ shifting the complexity to O(Nω).

If the DoG operator introduces a main optimization compared to the LoG operator, however the complexity O(Nω) is not parameter-free. The recent trends with camera devices (e.g. smartphones, tablets) are to process up to 10-Mpx for image streaming at 30 to 60 frames per second (FPS). However, as illustrated in Figure 9(a) the DoG operator can guarantee the frame rate at a low resolution only (less then 2-Mpx). If a low resolution is sufficient for simple text scene image Figure 9(a), it introduces character degradations with complex scene images Figure 9(b).

For optimization, the DoG operator can be estimated with almost-Gaussian functions [13] [20]. This enters in an estimator cascade methodology $LoG \approx DoG \approx \widehat{DoG}$, where $\widehat{DoG}$ is the DoG estimator. Specifically, repeated filtering with the averaging filters can be used to approximate a Gaussian filter, as given below Eq. (24) and shown in Figure 10(a), with a desired standard deviation [19].

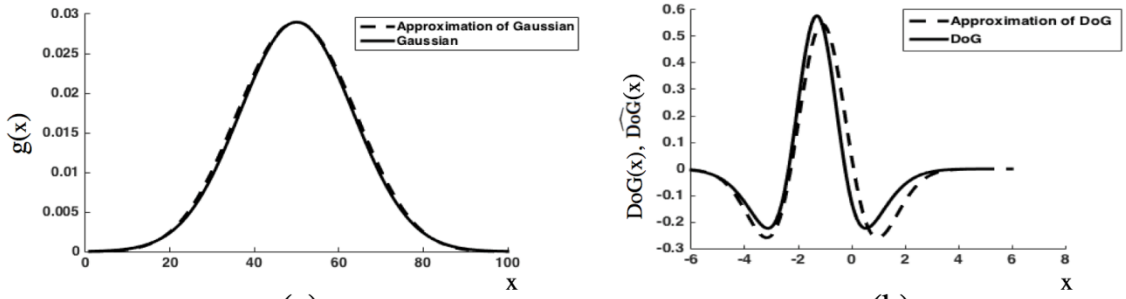


(a) (b)

Figure 9. (a) image with text from with processing time /FPS of DoG/almost- Gaussian operators at different resolutions with parameters σ_s (11) (b) degradations of text/characters at low resolutions with a complex scene image.

$$\hat{g}(x,y|\sigma) = \sum_{i \in [1,n]} \Pi_i(x,y) \qquad (24)$$

I



n

Figure 10. Approximation process (a) approximation of Gaussian function after the successive averaging (b) DoG can be obtained from approximation of Gaussian.

the Eq. (24) $\Pi_i(x,y)$ is a given box filter function having a predefined size. The quality of approximation is based on the number of repeated filtering *n*, certainly no more than 6. It can be justified by Eq. (25) in order to obtain approximation of a Gaussian, as presented in [19], where ω is the width of the averaging filter.

$$\frac{n(\omega-1)}{2} \approx 3\sigma \qquad (25)$$

From approximation of Gaussian in Eq. (24), it becomes possible to approximate the DoG operator by $\widehat{DoG}$ in (26) with two sets of box filter function. Figure 10(b) gives a plot of Eq. (26).

$$\begin{aligned}\widehat{DoG} &= \hat{g}(x,y|\sigma_1) \otimes f(x,y) - \hat{g}(x,y|\sigma_2) \otimes f(x,y) \\ &= \sum_{i \in [1,n]} \Pi_i(x,y) \otimes f(x,y) - \sum_{J \in [1,n]} \Pi_J(x,y) \otimes f(x,y) \end{aligned} \qquad (26)$$

Obviously, the $\Pi_i(x,y) \otimes f(x,y)$ products from Eq. (26) is able to be obtained with integral image at complexity *O(N)*. As a result, approximation of DoG is possibly achieved with *2n* accesses of integral image, it therefore is parameter free.

The DoG filter is then approximated as a linear combination of several box filters Π_i. Then, box coefficients must be found to minimize the approximation error. In [13], this is presented as an L1 regularized least-square problem that can be solved with an optimization algorithm (e.g. LASSO as detailed on the optimization aspects). The experiments in [13] report that DoG estimator achieves an acceleration $\approx\times 2$ at low scales $\sigma_s \in$ [1.5, 3.1], while maintaining a low average mean square error compared to the DoG. Figure 9(a) gives the processing time of the estimator over the different image resolutions and scales σ_s.

The BSV operator [4] is the edge-based operator while applying a hybrid strategy that generates a blob detection from an edge detection using a LoG look-like function. Although they get a sake of time-efficiency, the edge-based operators perform a poor detection as an average. The LoG operator is controlled through the stroke model paradigm for scale-invariance. The gLoG operator [15] guaranties the rotation and contrast-invariance. All these operators are symmetric except the gLoG operator. The symmetric operators detect the medical axes of characters that produces an important number of keypoint candidates. These keypoints must be post-processed for grouping. The gLoG operator relaxes this constraint, it the processes with a full primitive detection. Therefore, it is a time-consuming operator and is minimally compatible with a real-time strategy. However, it could be approximated by the DooG operator, even with the $\widehat{DooG}$ operator. This point has been little explored in the literature, it then could be a promising solution.

5 Conclusions and Perspectives

This paper has presented how the LoG operators can be set and adapted for text detection problem and made real-time with an estimator cascade methodology. Some main perspectives and challenges remain. Firstly, the LoG operators for text detection have mainly been investigated with symmetric model. However, little work exists on the generalization case (i.e. gLoG operator). The

generalization can turn the operator into a stroke detection for a better detection accuracy. Next, the real-time methodology with estimator cascade offers intermediate acceleration factors (≃ ×2 to ×4). It processes as a Full-Search (FS) method in the spatial domain with the fast estimation of the operator product. Similar to template matching, further acceleration could be obtained with FS-equivalent methods.

Bibliography

[1] Q. Ye and D. Doermann, "A survey Text detection and recognition in imagery," *PAMI*, vol. 37.7, pp. 1480-1500, 2015.

[2] R. Gomez and B. Shi, "ICDAR2017 robust reading challenge on COCO-Text," *ICDAR*, pp. 1435-1443, 2017.

[3] H. Yang and C. Wang, "An Improved System For Real-Time Scene Text Recognition," *Proc. Mul.*, pp. 657-660, 2015.

[4] X. Girones and C. Julia, "Real-Time Text Localization in Natural Scene Images Using a Linear Spatial Filter," *ICDAR*, pp. 1261-1268, 2017.

[5] S. Deshpande and R. Shriram, "Real time text detection and recognition on hand held objects to assist blind people," *Proc. Dyn. Opt. Tech*, pp. 1020-1024, 2016.

[6] B. Epshtein, E. Ofek and Y. Wexler, "Detecting text in natural scenes with stroke width transform," *CVPR*, pp. 2963-2970, 2010.

[7] L. Neumann and J. Matas, "Real-time scene text localization and regconition," *CVPR*, pp. 3538-3545, 2012.

[8] L. Neumann and J. Matas, "Scene text localization and regconition with oriented stroke detection," *ICCV*, pp. 97-104, 2013.

[9] L. Gomez and D. Karatzas, "MSER-based real-time text detection and tracking," in *ICPR*, 2014.

[10] Y. Liu, D. Zhang, Y. Zhang and S. Lin, "Real-time scene text detection based on stroke model," *ICPR*, pp. 3116-3120, 2014.

[11] J. Matas and L. Neumann, "Real-time lexicon-free scene text localization and recognition," *PAMI*, vol. 38.9, pp. 1872-1885, 2016.

[12] D. Nguyen, M. Delalandre, D. Conte and T. Pham, "Perfor- mance evaluation of real-time and scale-invariant LoG operators for text detection.," *VISAPP*, pp. 344-353, 2019.

[13] V. Fragoso, G. Srivastava, A. Nagar, Z. Li, K. Park and M. Turk, "Cascade of Box (CABOX) Filters for Optimal Scale Space Approximation," *CVPR*, pp. 126-131.

[14] D. Nguyen, M. Delalandre, D. Conte and T. Pham, "Fast RT-LoG operator for scene text detection," *JRTIP*, 2020.

[15] H. Kong, H. Akakin and S. Sarma, "A generalized Laplacian of Gaussian filter for blob detection and its applications," *Cyber*, vol. 43.6, pp. 1719-1733, 2013.

[16] N. Makhfi and O. Bannay, "Scale-space approach for character segmentation in scanned images of arabic document. J. . : 444 (2016)," *Theo. App. Infor. Tech*, vol. 94.2, 2016.

[17] R. Young, "Gaussian derivative theory of spatial vision: analysis of cortical cell receptive field line-weighting profiles," *Motors Research Laboratories*, 1985.

[18] W. Ma and M. B.S., "EdgeFlow: a technique for boundary detection and image segmentation," *TIP*, vol. 9.8, pp. 1375-1388, 2000.

[19] P. Kovesi, "Fast almost-gaussian filtering," *Dig. Ima. Comp. Tech*, pp. 21-125, 2010.

[20] M. Grabner, H. Grabner and H. Bischof, "Fast approximated SIFT," *ACCV*, pp. 918-927, 2006.

[21] D. Sen and S. Pal, "Gradient histogram: Thresholding in a region of interest for edge detection," *IVC*, vol. 28.4, pp. 677-695, 2010.

THẢO LUẬN VỀ CÁC TOÁN TỬ DỰA TRÊN LoG ĐỂ PHÁT HIỆN VĂN BẢN THEO THỜI GIAN THỰC

Dinh Cong Nguyen PhD

Thông tin bài viết

Ngày nhận bài:
20/9/2020
Ngày duyệt đăng:
10/12/2020

Từ khóa:

Phát hiện văn bản, toán tử LoG, mô hình đột quỵ, almost-Gaussian.

Tóm tắt

Trong bài báo này trình bày các phương pháp phát hiện văn bản thời gian thực trong hình ảnh dựa trên máy ảnh, tập trung đặc biệt vào toán tử Laplacian of Gaussian (LoG). Các phương pháp này được thảo luận với sự tập trung cụ thể vào các khía cạnh của tính phức tạp và tính mạnh mẽ. Một số kết quả minh họa và các thí nghiệm cơ bản được đưa ra để mô tả đặc điểm của các phương pháp. Hơn nữa, bài báo cũng cung cấp nhận xét về những cải tiến của các phương pháp đối với vấn đề phát hiện văn bản.