EXTENDING COMPRESSIVE BILATERAL FILTERING FOR ARBITRARY RANGE KERNEL

Yuto Sumiya[*]*, Norishige Fukushima*[*]*, Kenjiro Sugimoto*[†]*, and Sei-ichiro Kamata*[†]

[*]Nagoya Institute of Technology, Japan
[†]Waseda University, Japan

ABSTRACT

Smoothing filters have been used for pre/post-processing in various fields, such as computer vision and computer graphics. Bilateral filtering (BF) has a typical edge-preserving filter for such applications. The main issue of BF is its computational cost. Constant-time BF (O(1) BF) is one of the solutions to this problem, and compressive BF is a kind of O(1) BF. Compressive BF has, however, a restriction that we can only use Gaussian kernel as a range kernel until now. In this paper, we propose the method to extend compressive BF to handle arbitrary range kernels. Experimental results show that our extension handles arbitrary range kernels, and becomes the number of convolutions into half.

Index Terms— Constant-time bilateral filter, O(1) bilateral filter, compressive bilateral filter, arbitrary range kernel, Fourier series expansion

1. INTRODUCTION

Edge-preserving smoothing is the processing that smooths images while maintaining contours in it. Bilateral filtering (BF) [1] is one of the famous edge-preserving filters. The main drawback of BF is computational cost. Even if we heavily optimize codes of BF [2, 3], BF has more processing time than linear filters, such as Gaussian filtering (GF), because BF has spatially variant kernels. By contrast, GF is spatially invariant. Constant-time BF (O(1) BF) is one of the acceleration methods, and there are various approaches. O(1) BF decomposes BF into multiple GFs, and then we perform the GFs by using constant-time GF [4, 5, 6, 7, 8].

Early works of approximated BF [9, 10, 11, 12, 13, 14, 15] decompose BF by hat functions and then linearly interpolate results. The GF is not constant-time GF; thus, approximated BF is not $O(1)$. The following works [16, 17, 18, 19] extend BF to have constant-time property by using linear interpolation. Next works use raised cosine decomposition [20, 21, 22, 23], cosine transformation [24, 25], and matrix decomposition [26] to improve the approximation accuracy. Furthermore, convolution sharing techniques [22, 27, 28, 29, 30] reduces the number of convolutions into half. Taylor decomposition [22, 27], Chebyshev decomposition [28], fast compressive BF [29], and singular value decomposition (SVD) [30] can reduce the number of convolutions. Now, the SVD approach is state-of-the-arts.

BF has spatial and range kernels. Both kernels are Gaussian distribution. However, more flexibility kernel shape is useful for various signal processing applications, such as audio processing, biological signal processing, and signal processing for various IoT sensors. In this context, the linear interpolation approach [19] and the matrix factorization approach [26, 30] support this property. The SVD approach, especially, can save the number of convolutions. However, the SVD has difficulty in its initialization cost. Solving SVD takes high computational cost; thus, the method is not suitable for interactive processing, such as photo editing. The method is suitable for fixed-parameter cases, such as video processing.

Compressive bilateral filtering reaches the near performance of SVD, and its initialization cost is lower than SVD. The conventional form of compressive BF [24, 25] with slight modification, we can represent arbitrary range kernels by using cosine transform. However, the method is cost-consuming because the paper focuses only on Gaussian distribution. Also, the functionality of the arbitrary range kernel is not justified in the paper. Moreover, the acceleration approach of compressive BF [29], named fast compressive BF, can only accelerate the Gaussian range kernel.

Therefore, we extend compressive bilateral filtering to have arbitrary range kernels and also accelerate the initialization. The contributions of this paper are:

- Extending fast compressive bilateral filtering to have arbitrary range kernel.
- Accelerating the period optimization, which improves the approximation accuracy.

[*]This work was supported by JSPS KAKENHI 17H01764, 18K19813

[†]This work was supported by JSP KAKENHI 17H01764, 18K18076.

2. RELATED WORKS

2.1. Formulation of $O(1)$ BF

The kernel of BF consists of two weights, such as a weight of spatial domain $\mathcal{S}$ and a weight of range domain $\mathcal{R}$. Consider D-dimensional grayscale image with spatial domain $\mathcal{S} \subset \mathbb{Z}^D$ and dynamic range $\mathcal{R} = \{0, \dots, R-1\} \subset \mathbb{Z}$, (D = 2, R = 256 in general). A pixel position is denoted by $\boldsymbol{p} \in \mathcal{S}$ and the intensity at $\boldsymbol{p}$ is expressed by $I_{\boldsymbol{p}} \in \mathcal{R}$. BF is defined as follows;

$$\hat{I}_{\boldsymbol{p}} = \frac{\sum_{\boldsymbol{q}\in\mathcal{S}} w_s(\boldsymbol{p},\boldsymbol{q}) w_r(I_{\boldsymbol{p}}, I_{\boldsymbol{q}}) I_{\boldsymbol{q}}}{\sum_{\boldsymbol{q}\in\mathcal{S}} w_s(\boldsymbol{p},\boldsymbol{q}) w_r(I_{\boldsymbol{p}}, I_{\boldsymbol{q}})}, \tag{1}$$

where $\hat{I}_{\boldsymbol{p}}$ are the output intensity at $\boldsymbol{p}$. $I_{\boldsymbol{p}}$ and $I_{\boldsymbol{q}}$ are the intensity at target and reference pixels, respectively. w_s and w_r are spatial and range weights, respectively. The weights are defined by $w_s(\boldsymbol{p},\boldsymbol{q}) = \exp(-\frac{\|\boldsymbol{q}-\boldsymbol{p}\|_2^2}{2\sigma_s^2})$ and $w_r(\boldsymbol{a},\boldsymbol{b}) = \exp(-\frac{(\boldsymbol{b}-\boldsymbol{a})^2}{2\sigma_r^2})$, where σ_s and σ_r are parameters of spatial and range scales, respectively.

$O(1)$ BF can be formulated by substituting $w_r(\boldsymbol{a},\boldsymbol{b}) \approx \sum_{k=0}^{K-1} \phi_k(\boldsymbol{a})\psi_k(\boldsymbol{b})$ into formula (1):

$$\hat{I}_{\boldsymbol{p}} \approx \frac{\sum_{k=0}^{K-1} \phi_k(I_{\boldsymbol{p}}) \sum_{\boldsymbol{q}\in\mathcal{S}} w_s(\boldsymbol{p},\boldsymbol{q})\{\psi_k(I_{\boldsymbol{q}}) I_{\boldsymbol{q}}\}}{\sum_{k=0}^{K-1} \phi_k(I_{\boldsymbol{p}}) \sum_{\boldsymbol{q}\in\mathcal{S}} w_s(\boldsymbol{p},\boldsymbol{q})\{\psi_k(I_{\boldsymbol{q}})\}}, \tag{2}$$

where K is the approximating order. $\sum_{\boldsymbol{q}\in\mathcal{S}} w_s(\boldsymbol{p},\boldsymbol{q})\{\cdot\}$ can be considered as a spatial convolution of an image. The functions of ϕ and ψ are different for each approximation method, e.g., linear, compressive, and SVD.

2.2. Compressive Bilateral Filtering [24]

Gaussian kernel is an even function; thus, we can represent it as sum of cosine terms by using Fourier series expansion:

$$w_r(x) \approx \hat{w_r}(x,K,T) = \alpha_0 + 2\sum_{k=1}^{K} \alpha_k \cos\left(\frac{2\pi}{T} k x\right), \tag{3}$$

$$\alpha_k = \frac{1}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}} f(x)\cos(w_k x)dx \tag{4}$$

The closed-form approximation holds $\alpha_k \approx \frac{2}{T}\exp^{-\frac{1}{2}(\frac{2\pi}{T}k\sigma)^2}$ for Gaussian kernel [24]. By using the trigonometric addition formulas, BF becomes as follows;

$$\hat{I}_{\boldsymbol{p}} = \frac{\alpha_0 \tilde{I}_{\boldsymbol{p}} + 2\sum_{k=1}^{K} \alpha_k \big(\cos(w_k I_{\boldsymbol{p}})\tilde{C}'_{\boldsymbol{p}} + \sin(w_k I_{\boldsymbol{p}})\tilde{S}'_{\boldsymbol{p}}\big)}{\alpha_0 + 2\sum_{k=1}^{K} \alpha_k \big(\cos(w_k I_{\boldsymbol{p}})\tilde{C}_{\boldsymbol{p}} + \sin(w_k I_{\boldsymbol{p}})\tilde{S}_{\boldsymbol{p}}\big)}, \tag{5}$$

$\tilde{C}_{\boldsymbol{p}} = \sum_{\boldsymbol{q}\in\mathcal{S}} w_s(\boldsymbol{p},\boldsymbol{q})\cos(w_k I_{\boldsymbol{q}})$, $\tilde{S}_{\boldsymbol{p}} = \sum_{\boldsymbol{q}\in\mathcal{S}} w_s(\boldsymbol{p},\boldsymbol{q})\sin(w_k I_{\boldsymbol{q}})$, $\tilde{C}'_{\boldsymbol{p}} = \sum_{\boldsymbol{q}\in\mathcal{S}} w_s(\boldsymbol{p},\boldsymbol{q})\cos(w_k I_{\boldsymbol{q}}) I_{\boldsymbol{q}}$, $\tilde{S}'_{\boldsymbol{p}} = \sum_{\boldsymbol{q}\in\mathcal{S}} w_s(\boldsymbol{p},\boldsymbol{q})\sin(w_k I_{\boldsymbol{q}}) I_{\boldsymbol{q}}$, where $w_k = \frac{2\pi}{T}k$, and $\tilde{I}_{\boldsymbol{p}}$ is result of GF for $I_{\boldsymbol{p}}$. In this case, ϕ and ψ are cos and sin functions. The period of T is optimized by minimizing the following closed-form function:

$$E(K,T) = \mathrm{erfc}\left(\frac{\pi\sigma}{\mathrm{T}}(2\mathrm{K}+1)\right) + \mathrm{erfc}\left(\frac{\mathrm{T}-\mathcal{R}}{\sigma}\right). \tag{6}$$

The function is differential; thus, we can obtain $T_{opt} = \arg\min_T E(K,T)$ by using Newton-Raphson method. The number of convolutions is $4K+1$.

2.3. Fast Compressive Bilateral Filtering [29]

Fast compressive BF shares the convolution results for each numerator and denominator, such as $\tilde{S}_{\boldsymbol{p}}$ and $\tilde{S}'_{\boldsymbol{p}}$, $\tilde{C}_{\boldsymbol{p}}$ and $\tilde{C}'_{\boldsymbol{p}}$. For this sharing, we change the filtering form (1) by subtracting $I_{\boldsymbol{p}}$:

$$\hat{I}_{\boldsymbol{p}} - I_{\boldsymbol{p}} = \frac{\sum_{\boldsymbol{q}\in\mathcal{S}} w_s(\boldsymbol{p},\boldsymbol{q}) w_r(I_{\boldsymbol{p}}, I_{\boldsymbol{q}})(I_{\boldsymbol{q}} - I_{\boldsymbol{p}})}{\sum_{\boldsymbol{q}\in\mathcal{S}} w_s(\boldsymbol{p},\boldsymbol{q}) w_r(I_{\boldsymbol{p}}, I_{\boldsymbol{q}})}. \tag{7}$$

Plugin the differential of Gaussian distribution function $\exp'(\frac{(a-b)^2}{-2\sigma^2}) = -\exp(\frac{(a-b)^2}{-2\sigma^2})\frac{(a-b)}{\sigma^2}$ into this equation (7), fast compressive BF is defined as follows:

$$\hat{I}_{\boldsymbol{p}} = I_{\boldsymbol{p}} - \frac{\sum_{\boldsymbol{q}\in\mathcal{S}} w_s(\boldsymbol{p},\boldsymbol{q}) w'_r(I_{\boldsymbol{p}}, I_{\boldsymbol{q}})}{\sum_{\boldsymbol{q}\in\mathcal{S}} w_s(\boldsymbol{p},\boldsymbol{q}) w_r(I_{\boldsymbol{p}}, I_{\boldsymbol{q}})}. \tag{8}$$

w'_r can be computed from differential of Eq. (3). Therefore, the filtering is defined by:

$$\hat{I}_{\boldsymbol{p}} = I_{\boldsymbol{p}} - \frac{2\sigma^2 \sum_{k=1}^{K} \alpha_k \big(\sin(w_k I_{\boldsymbol{p}})\tilde{C}_{\boldsymbol{p}} - \cos(w_k I_{\boldsymbol{p}})\tilde{S}_{\boldsymbol{p}}\big)}{\alpha_0 + 2\sum_{k=1}^{K} \alpha_k \big(\cos(w_k I_{\boldsymbol{p}})\tilde{C}_{\boldsymbol{p}} + \sin(w_k I_{\boldsymbol{p}})\tilde{S}_{\boldsymbol{p}}\big)}. \tag{9}$$

This form has the same convolution results, i.e., $\tilde{C}_{\boldsymbol{p}}$ and $\tilde{S}_{\boldsymbol{p}}$, in the numerator and denominator terms. Therefore, the number of convolutions is half ($2K$).

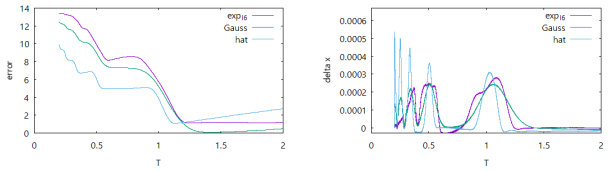
3. PROPOSED METHOD

3.1. Extension for Arbitrary Range Kernel

The compressive BF [24] and the fast method [29] are specialized for Gaussian distribution. In this section, we extend the works to have arbitrary range kernels.

For compressive BF, it is easy to extend to arbitrary range kernels. Computing (3) and (4), the form supports arbitrary range kernels, because Fourier series expansion is adaptable to piece-wise smooth arbitrary functions, and it does not assume that the function is differentiable. To solve (4), we use numerical integration with trapezoidal rule. For the optimizing period, the equation cannot support arbitrary range kernel. This problem is solved in the next subsection.

Next, we consider fast compressive BF. First, we start the form (7). Obviously, we can solve denominator terms in the same way as compressive BF. Note that we cannot use the closed-form solution, because we do not limit the range kernel as Gaussian distribution. For the denominator, the term $w_r(I_{\boldsymbol{p}}, I_{\boldsymbol{q}})(I_{\boldsymbol{q}} - I_{\boldsymbol{p}})$ in (7) is an odd function because w_r is an even function and $I_{\boldsymbol{q}} - I_{\boldsymbol{p}}$ is an odd function. Therefore, we approximate the function of $f(x)x$ by sum of sin terms by Fourier series expansion.



(a) Error (b) Δ Error

Fig. 1. Optimizing period.

(a) Gauss (b) hat (c) $\exp_{L_6}$

Fig. 2. Shape of each kernel.

$$w_r(x)x \approx 2\sum_{k=1}^{K}\beta_k \sin\left(\frac{2\pi}{T}kx\right), \tag{10}$$

$$\beta_k = \frac{1}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}} f(x)x\sin(w_k x)dx \tag{11}$$

To solve (11), we also use numerical integration. By using the trigonometric addition formulas, the approximated function of $f(x)x$ in the range kernel are decomposed into two variables:

$$w_r(a,b)(b-a) \approx 2\sum_{k=1}^{K}\beta_k \sin(\omega_k(b-a)) \tag{12}$$

$$= 2\sum_{k=1}^{K}\beta_k(\sin(\omega_k a)\cos(\omega_k b) - \cos(\omega_k a)\sin(\omega_k b)), \tag{13}$$

After separation, the form of fast compressive BF for arbitrary range kernel is as follows;

$$\hat{I}_{\boldsymbol{p}} = I_{\boldsymbol{p}} - \frac{2\sum_{k=1}^{K}\beta_k\left(\sin(w_k I_{\boldsymbol{p}})\tilde{C}_{\boldsymbol{p}} - \cos(w_k I_{\boldsymbol{p}})\tilde{S}_{\boldsymbol{p}}\right)}{\alpha_0 + 2\sum_{k=1}^{K}\alpha_k\left(\cos(w_k I_{\boldsymbol{p}})\tilde{C}_{\boldsymbol{p}} + \sin(w_k I_{\boldsymbol{p}})\tilde{S}_{\boldsymbol{p}}\right)}. \tag{14}$$

The form can share the convolution terms in the numerator and denominator terms, such as $\tilde{C}_{\boldsymbol{p}}$ and $\tilde{S}_{\boldsymbol{p}}$. The number of convolutions is $2K$.

3.2. Optimization Method of Period T

The period of T is generally 2. However, we usually use a monotonically decreasing function from 0 to the tail for the range kernel. In this case, we can ignore the tail distribution. The tails discarding is realized by minimizing T. The compressive BF [24] optimizes the periods to have more performance. To optimize T, we cannot use the closed-form solution of Eq. (6), because we handle unknown arbitrary functions. The naive kernel error function between ideal kernel of w_r and approximated kernel of $\hat{w}_r$ are defined as follows;

$$E(K,T) = \frac{\sum_{a\in\mathcal{R}}\sum_{b\in\mathcal{R}}\left(w_r(a,b) - \hat{w}_r(a,b,K,T)\right)^2}{|\mathcal{R}| \times |\mathcal{R}|}, \tag{15}$$

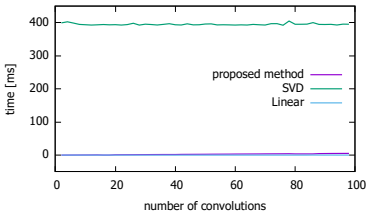


Fig. 3. Initialize time of each method.

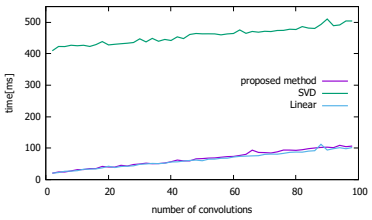


Fig. 4. Total processing time for each method.

where $|\mathcal{R}|$ is the number of elements in intensity, and usually 256. This error function has a double-loop. Also, the function is not guaranteed by a differentiable function. Therefore, the direct solution to obtain $T_{opt} = \arg\min_T E(K,T)$ is linear search algorithm; however, the processing is cost-consuming. Figure 1 shows $E(K,T)$ and $E(K,T) - E(K,T+\epsilon)$, where ϵ is a very small value, for various kernels used in the experimental results section. The figure shows that the functions are not applicable for Newton's method.

Here, $\hat{w}_r(a,b,K,T)$ is a periodic function; thus, the function has shiftability [20]. In this condition, the error function (15) becomes a single-loop.

$$E(K,T) = \frac{\sum_{a\in\mathcal{R}}(w_r(v,a) - \hat{w}_r(v,a,K,T))^2}{|\mathcal{R}|}, \tag{16}$$

where $v \in \mathcal{R}$ is a constant value, i.e., any value from 0 to 255 is possible. The transformation minimizes the search cost.

Further, we utilize the golden section search algorithm to solve the problem more efficiently. We assume that the function w_r is piece-wise smooth for the Fourier series expansion. In this case, the golden section search algorithm works well.

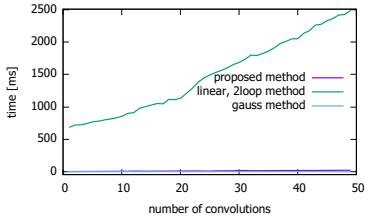
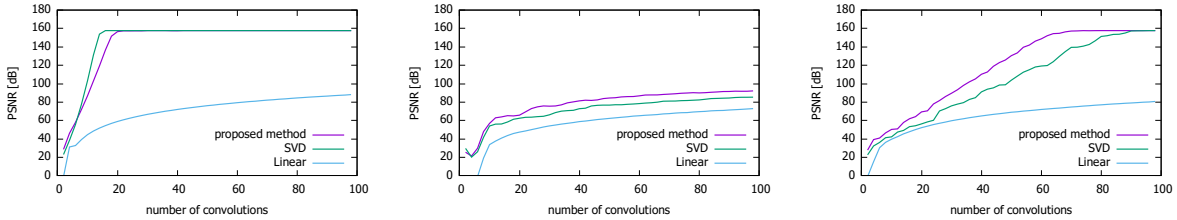


Fig. 5. Processing time of calculating the period. Order K is from 1 to 50.



(a) Gauss (b) hat (c) $\exp_{L_6}$

Fig. 6. PSNR vs. the number of convolutions for each method.

(a) Proposed (383ms) (b) SVD (1117ms) (c) Linear (367ms)

Fig. 7. Filtering results of hat kernel for each method. Differential parts are boosted by 1.0×10^4. The numer of convolutions is 100.

4. EXPERIMENTAL RESULTS

We evaluated compressive BF with arbitrary range kernels, such as Gaussian, exponential function of L_p norm[1], and hat kernels. The definitions are $\mathrm{Gauss}(\mathrm{a}-\mathrm{b}) = \exp(-\frac{1}{2}\left(\frac{\mathrm{a}-\mathrm{b}}{\sigma_\mathrm{r}}\right)^2)$, $\mathrm{hat}(\mathrm{a}-\mathrm{b}) = \max(0, 1-\frac{|\mathrm{a}-\mathrm{b}|}{\sigma_\mathrm{r}})$, and $\exp_{l_p}(x) = \exp(-\frac{1}{p}\left(\frac{|a-b|}{\sigma_r}\right)^p)$. Figure 2 shows the kernel shapes. For $\exp_{L_p}$, $p=1$ is Laplacian distribution, $p=2$ is Gaussian distribution, and $p=\infty$ is box kernel. In this experiment, we use $p=6$. The code was written in C++, parallelized by OpenMP, and vectorization by AVX. CPU was Core i5-4690 3.50GHz. For the entire experiment, we set $\sigma_r = 40$, $\sigma_s = 5$. We used the standard test grayscale images (512×512), such as Lenna, Cameraman, Sailboat on lake, Couple, Peppers, Male, Stream and bridge, and Woman.

Figure 5 shows the processing time to optimize the period of T. We compared three methods, such as linear search with a double-loop (linear, 2loop method), golden section search with a single-loop (proposed method), and closed-form solution of Gaussian distribution (gauss method). Note that we evaluated only the Gaussian kernel for the closed-form solution. The proposed searching accelerates the optimization processing, and the difference between the closed-form is less than 0.001ms.

The following experiments, we compared the proposed method with linear [19] and SVD [30]. All results are averaged for each test image. Figure 3 shows the initializing time of generating the range kernels. The horizontal axis is the number of convolutions. For linear and the proposed method, the number of convolutions is $2K$. For SVD, that is K. Linear is the fastest and is within 1 ms. The processing time of the proposed method within 5.5 ms at worst. By contrast, SVD needs about 400ms for initialization; thus, it is not suitable for interactive processing. Figure 4 shows the time from start to finish all processing, including convolution operations. The processing time of the proposed method and the linear method is almost the same. The proposed method is five times faster than the SVD method. Note that when we process videos with the fixed-parameter, we can ignore the initialization cost.

Figure 6 shows peak signal-to-noise ratio (PSNR) between approximated results and naive filtering results, and Figure 7 shows the result images; the left half is the output image, and the other half is the difference image between approximated results and naive filtering results. Except for the Gaussian kernel, the proposed method has higher accuracy than the other method. For the Gaussian kernel, SVD has better performance. Note that we take within 1 ms for one convolution. However, the SVD requires offset time to generate range kernel; therefore, the proposed method tends to have better performance even when we use the Gaussian kernel in interactive filtering.

5. CONCLUSION

In this paper, we extended compressive BF to have any range kernel. Moreover, we also extended fast compressive BF to reduce the number of convolutions into half. Furthermore, we accelerate the optimization process of the period for trigonometric functions by using shiftability in error function and the golden selection search algorithm.

Experimental results show that we can represent arbitrary range kernels, and the proposed method superior to the state-of-the-art of the SVD method. Also, the accelerated optimization method minimizes the initialization cost method, and the cost reaches the near computational time of the simplest approach of linear interpolation.

[1] $L_p = (|x_1|^p + |x_2|^p + \cdots + |x_n|^p)^{\frac{1}{p}}$

6. REFERENCES

[1] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, 1998.

[2] Y. Maeda, N. Fukushima, and H. Matsuo, "Effective implementation of edge-preserving filtering on cpu microarchitectures," *Applied Sciences*, vol. 8, no. 10, 2018.

[3] Y. Maeda, N. Fukushima, and H. Matsuo, "Taxonomy of vectorization patterns of programming for fir image filters using kernel subsampling and new one," *Applied Sciences*, vol. 8, no. 8, 2018.

[4] R. Deriche, "Recursively implementating the gaussian and its derivatives," in *Proc. IEEE International Conference on Image Processing (ICIP)*, 1992, pp. 263–267.

[5] L. J. van Vliet, I. T. Young, and P. W. Verbeek, "Recursive gaussian derivative filters," in *Proc. International Conference on Pattern Recognition (ICPR)*, 1998.

[6] K. Sugimoto and S. Kamata, "Fast gaussian filter with second-order shift property of dct-5," in *Proc. IEEE International Conference on Image Processing (ICIP)*, 2013.

[7] K. Sugimoto and S. Kamata, "Efficient constant-time gaussian filtering with sliding dct/dst-5 and dual-domain error minimization," *ITE Transactions on Media Technology and Applications*, vol. 3, no. 1, pp. 12–21, 2015.

[8] K. Sugimoto, S. Kyochi, and S. Kamata, "Universal approach for dct-based constant-time gaussian filter with moment preservation," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.

[9] F. Durand and J. Dorsey, "Fast bilateral filtering for the display of high-dynamic-range images," *ACM Transactions on Graphics*, vol. 21, no. 3, pp. 257–266, 2002.

[10] B. Weiss, "Fast median and bilateral filtering," *ACM Transactions on Graphics*, vol. 25, no. 3, pp. 519–526, 2006.

[11] J. Chen, S. Paris, and F. Durand, "Real-time edge-aware image processing with the bilateral grid," *ACM Transactions on Graphics*, vol. 26, no. 3, 2007.

[12] S. Paris and F. Durand, "A fast approximation of the bilateral filter using a signal processing approach," in *Proc. European conference on computer vision (ECCV)*, 2006.

[13] S. Paris and F. Durand, "A fast approximation of the bilateral filter using a signal processing approach," *International Journal of Computer Vision*, vol. 81, no. 1, pp. 24–52, 2009.

[14] A. Adams, N. Gelfand, J. Dolson, and M. Levoy, "Gaussian kd-trees for fast high-dimensional filtering," *ACM Transactions on Graphics*, vol. 28, no. 3, pp. 21, 2009.

[15] A. Adams, J. Baek, and M. A. Davis, "Fast high-dimensional filtering using the permutohedral lattice," *Computer Graphics Forum*, vol. 29, no. 2, pp. 753–762, 2010.

[16] F. Porikli, "Constant time o(1) bilateral filtering," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[17] Q. Yang, K. H. Tan, and N. Ahuja, "Real-time o(1) bilateral filtering," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[18] Q. Yang, N. Ahuja, and K. H. Tan, "Constant time median and bilateral filtering," *International Journal of Computer Vision*, vol. 112, no. 3, pp. 307–318, 2015.

[19] B. K. Gunturk, "Fast bilateral filter with arbitrary range and domain kernels," *IEEE Transactions on Image Processing*, vol. 20, no. 9, pp. 2690–2696, 2011.

[20] K. N. Chaudhury, "Constant-time filtering using shiftable kernels," *IEEE Signal Processing Letters*, vol. 18, no. 11, pp. 651–654, 2011.

[21] K. N. Chaudhury, "Acceleration of the shiftable o(1) algorithm for bilateral filtering and nonlocal means," *IEEE Transactions on Image Processing*, vol. 22, no. 4, pp. 1291–1300, 2013.

[22] K. N. Chaudhury, "Fast and accurate bilateral filtering using gauss-polynomial decomposition," in *Proc. IEEE International Conference on Image Processing (ICIP)*, 2015.

[23] N. Fukushima, K. Sugimoto, and S. Kamata, "Complex coefficient representation for iir bilateral filter," in *Proc. International Conference on Image Processing (ICIP)*, 2017.

[24] K. Sugimoto and S. Kamata, "Compressive bilateral filtering," *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 3357–3369, 2015.

[25] S. Ghosh, P. Nair, and K. N. Chaudhury, "Optimized fourier bilateral filtering," *IEEE Signal Processing Letters*, vol. 25, no. 10, pp. 1555–1559, 2018.

[26] K. Sugimoto, T. Breckon, and S. Kamata, "Constant-time bilateral filter using spectral decomposition," in *Proc. IEEE International Conference on Image Processing (ICIP)*, 2016.

[27] K. N. Chaudhury and S. D. Dabhade, "Fast and provably accurate bilateral filtering," *IEEE Transactions on Image Processing*, vol. 25, no. 6, pp. 2519–2528, 2016.

[28] S. Ghosh and K. N. Chaudhury, "Fast and high-quality bilateral filtering using gauss-chebyshev approximation," in *International Conference on Signal Processing and Communications (SPCOM)*, 2016.

[29] G. Deng, "Fast compressive bilateral filter," *Electronics Letters*, vol. 53, no. 3, pp. 150–152, 2017.

[30] K. Sugimoto, N. Fukushima, and S. Kamata, "200 fps constant-time bilateral filter using svd and tiling strategy," in *Proc. IEEE International Conference on Image Processing (ICIP)*, 2019.