Fast Bilateral Filter for Multi-channel Images via Soft-assignment Coding

Kenjiro Sugimoto*, Norishige Fukushima† and Sei-ichiro Kamata*
*Graduate School of Information, Production and Systems, Waseda University, Kitakyushu, Japan.
E-mail: ksugimoto@aoni.waseda.jp, kam@waseda.jp
†Faculty of Engineering, Nagoya Institute of Technology, Nagoya, Japan.
E-mail: fukushima@nitech.ac.jp

***Abstract*—This paper presents an acceleration method of the bilateral filter (BF) for multi-channel images. In most existing acceleration methods, the BF is approximated by an appropriate combination of convolutions. A major purpose under this framework is to achieve sufficient approximate accuracy by as few convolutions as possible. However, state-of-the-art methods for multi-channel images still requires hundreds of (e.g., 256) convolutions to achieve sufficient accuracy. The proposed method reduces the number of convolutions without a loss in accuracy via soft-assignment coding. This approach enables us to take two major advantages that two state-of-the-art methods (scalar quantization with linear interpolation and vector quantization) have individually provided. Experiments show that the proposed method can produce sufficiently-accurate resulting images by using 64–80 convolutions only.**

I. Introduction

The bilateral filter (BF) [1]–[3] has played a fundamental role as an edge-preserving smoother in image processing, computer vision and computer graphics. The edge-preserving effect is achieved by the following approach. Traditional linear filters determine filter weights only from the spatial distance between current pixel and its neighboring pixels; the BF additionally introduces their intensity difference. Although it was originally focuses on single-channel images, we discusses its generalization for multi-channel ones such as RGB or hyper-spectral images. We distinguish between them as single-channel BF (ScBF) and multi-channel BF (McBF). The McBF can produce more natural smoothing results (e.g., less pseudo colors around edges) than the ScBF.

A major drawback of the BF is high computational complexity. In order to overcome this problem, many accelerated algorithms for the ScBF have been proposed in the past [4]–[10]. Most of them approximate the ScBF by an appropriate combination of convolutions. This framework has a tradeoff between approximate accuracy and the number of convolutions. In short, we aim at obtaining sufficient accuracy by using as few convolutions as possible. The existing methods for the ScBF have shown successful results in many tasks. Likewise, accelerating the McBF has been actively studied recent years [11]–[15]. However, they still require hundreds of convolutions to satisfy sufficient accuracy. From a practical viewpoint, it is demanded for real applications to drastically reduce the number of convolutions required.

This paper presents an accelerated algorithm for the McBF that requires much fewer convolutions. Our method utilizes the soft-assignment coding [16], [17], which is a well-known technique for general object recognition tasks. This approach enables us to combine major advantages of two state-of-the-art methods: the Yang method [13] (scalar quantization with linear interpolation) and the Mozerov method [12] (vector quantization). Experiments show that, as compared with the Yang method, our method reduces the number of convolutions by approximately 75% without a loss of accuracy.

II. Existing Work

A. Multi-channel Bilateral filter

Although the original BF [1]–[3] treats single-channel images, this paper focuses on its natural extension for multi-channel images. Let us consider filtering a D-dimensional M-channel image with N pixels. Let $\boldsymbol{f} : \mathcal{S} \to \mathcal{R}$ be a target image and $\tilde{\boldsymbol{f}} : \mathcal{S} \to \mathcal{R}$ its smoothed image where $\mathcal{S} \subset \mathbb{Z}^D$ and $\mathcal{R} \subset \mathbb{R}^M$ denote the spatial domain (all pixel positions) and the range domain (possible color vectors) of the images, respectively. By describing the current pixel as $\boldsymbol{p} \in \mathcal{S}$ and its neighboring pixels as $\mathcal{N}(\boldsymbol{p}) \subset \mathcal{S}$, the McBF is defined by

$$\tilde{\boldsymbol{f}}(\boldsymbol{p}) := \frac{\sum_{\boldsymbol{q} \in \mathcal{N}(\boldsymbol{p})} w_s(\boldsymbol{p}, \boldsymbol{q})\, w_r(\boldsymbol{f}(\boldsymbol{p}), \boldsymbol{f}(\boldsymbol{q}))\, \boldsymbol{f}(\boldsymbol{q})}{\sum_{\boldsymbol{q} \in \mathcal{N}(\boldsymbol{p})} w_s(\boldsymbol{p}, \boldsymbol{q})\, w_r(\boldsymbol{f}(\boldsymbol{p}), \boldsymbol{f}(\boldsymbol{q}))}, \quad (1)$$

where $w_s : \mathcal{S} \times \mathcal{S} \to \mathbb{R}_+$ and $w_r : \mathcal{R} \times \mathcal{R} \to \mathbb{R}_+$ are called a spatial kernel and a range kernel, respectively. The both kernels are selected according to computational complexity, noise model or so on. Our discussion is basically applicable to arbitrary spatial/range kernels. We here show Gaussian spatial/range kernels as an common example:

$$w_s(\boldsymbol{x}, \boldsymbol{y}) := e^{-\frac{\|\boldsymbol{y}-\boldsymbol{x}\|^2}{2\sigma_s^2}}, \quad w_r(\boldsymbol{x}, \boldsymbol{y}) := e^{-\frac{\|\boldsymbol{y}-\boldsymbol{x}\|^2}{2\sigma_r^2}}, \quad (2)$$

where σ_s and σ_r are spatial and range scale parameters, respectively, and $\|\cdot\|$ denotes ℓ_2-norm of a vector. An important point is that this naive McBF has computational complexity proportional to the filtering window size $|\mathcal{N}(\boldsymbol{p})|$, which depends on D and σ_s. In general, this theoretical characteristic is a severe problem for real-time processing applications.

B. Accelerated bilateral filters and remaining problems

In the ScBF, many accelerated algorithms have been proposed to reduce computational complexity [4]–[10]. Basically,

these algorithms share the general framework that it is approximated by an appropriate combination of convolutions. Since this framework has a tradeoff between approximate accuracy and the number of convolutions, we aim at achieving sufficient accuracy by as few convolutions as possible. Unfortunately, their natural extensions to multi-channel images generally suffer from *the curse of dimensionality*. Specifically, the number of convolutions increases exponentially with increasing M. This property causes unacceptable computational time even in the case of RGB images ($M = 3$).

In order to address this problem, accelerations for the McBF have been actively studied recent years [12]–[15]. All of them mainly discussed how to simplify range domain $\mathcal{R}$ because the smaller $\mathcal{R}$ requires the fewer convolutions in general. Yang *et al.* [13] aggressively quantized $\mathcal{R}$ (and also spatial domain $\mathcal{S}$) by employing scalar quantization with linear interpolation. Mozerov et al. [12] simplified $\mathcal{R}$ by means of vector quantization in view of color sparseness of images. The other approaches [14], [15] mainly approximated distance computation in $\mathcal{R}$ by random projection. However, these state-of-the-art algorithms still requires at least 100 convolutions to achieve sufficient accuracy, even if RGB images ($M = 3$) are targeted. This is because the size of $\mathcal{R}$ expands exponentially with increasing M. Consequently, it is required to reduce much more convolutions for real-time applications.

III. Proposed Method

This section presents an accelerated McBF that treats range domain $\mathcal{R}$ via soft-assignment coding, which is a successful technique for general object recognition tasks [16], [17].

A. Separable range kernel with Kronecker delta

We decompose the McBF into an appropriate combination of convolutions by describing (2) as the separable form

$$w_r(\boldsymbol{x}, \boldsymbol{y}) = \sum_{\boldsymbol{t} \in \mathcal{R}} \delta(\boldsymbol{x}, \boldsymbol{t})\, w_r(\boldsymbol{t}, \boldsymbol{y}), \tag{3}$$

where $\delta(\boldsymbol{x}, \boldsymbol{y}) = [\boldsymbol{x} = \boldsymbol{y}] \in \{0, 1\}$ indicates the Kronecker delta. Substituting (3) for (1), the McBF is rewritten as

$$\tilde{\boldsymbol{f}}(\boldsymbol{p}) = \frac{\sum_{\boldsymbol{t} \in \mathcal{R}} \delta(\boldsymbol{f}(\boldsymbol{p}), \boldsymbol{t})\, \boldsymbol{\xi}_{\boldsymbol{t}}(\boldsymbol{p})}{\sum_{\boldsymbol{t} \in \mathcal{R}} \delta(\boldsymbol{f}(\boldsymbol{p}), \boldsymbol{t})\, \zeta_{\boldsymbol{t}}(\boldsymbol{p})} = \sum_{\boldsymbol{t} \in \mathcal{R}} \delta(\boldsymbol{f}(\boldsymbol{p}), \boldsymbol{t}) \frac{\boldsymbol{\xi}_{\boldsymbol{t}}(\boldsymbol{p})}{\zeta_{\boldsymbol{t}}(\boldsymbol{p})}, \tag{4}$$

where $\boldsymbol{\xi}_{\boldsymbol{t}} : \mathcal{S} \to \mathcal{R}$ and $\zeta_{\boldsymbol{t}} : \mathcal{S} \to \mathbb{R}$ are defined by

$$\boldsymbol{\xi}_{\boldsymbol{t}}(\boldsymbol{p}) = \sum_{\boldsymbol{q} \in \mathcal{N}(\boldsymbol{p})} w_s(\boldsymbol{p}, \boldsymbol{q}) \left\{ w_r(\boldsymbol{t}, \boldsymbol{f}(\boldsymbol{q}))\, \boldsymbol{f}(\boldsymbol{q}) \right\}, \tag{5}$$

$$\zeta_{\boldsymbol{t}}(\boldsymbol{p}) = \sum_{\boldsymbol{q} \in \mathcal{N}(\boldsymbol{p})} w_s(\boldsymbol{p}, \boldsymbol{q}) \left\{ w_r(\boldsymbol{t}, \boldsymbol{f}(\boldsymbol{q})) \right\}, \tag{6}$$

We call $\boldsymbol{\xi}_{\boldsymbol{t}}(\cdot)$ and $\zeta_{\boldsymbol{t}}(\cdot)$ *component images* corresponding to color vector $\boldsymbol{t}$. Obviously, they are generated by convolutions to the M-channel image $\{w_r(\boldsymbol{t}, \boldsymbol{f}(\boldsymbol{q}))\, \boldsymbol{f}(\boldsymbol{q})\}$ and the single-channel image $\{w_r(\boldsymbol{t}, \boldsymbol{f}(\boldsymbol{q}))\}$, respectively. If we generate the component images to all the possible color vectors $\forall \boldsymbol{t} \in \mathcal{R}$, (4) yields exactly the same results as (1). In terms of computational complexity, it is required to reduce the number of component images without an unacceptable loss in accuracy.

B. Soft-assignment coding of color vectors

We approximate (4) by using component images corresponding to some dominant color vectors only. Moreover, the approximate accuracy is enhanced by utilizing linear combinations of them derived from soft-assignment coding. Let $\boldsymbol{c}_k \in \mathcal{R}$, $(k = 1 \ldots, K)$ be dominant color vectors of a target image where K indicates the number of dominant color vectors. Instead of the summation of $\delta(\cdot)$ in (4), we use the weights derived by soft-assignment coding

$$\alpha_k(\boldsymbol{x}) := \frac{\exp\left(-\lambda \left\| \boldsymbol{x} - \boldsymbol{c}_k \right\|^2\right)}{\sum_{l=1}^{K} \exp\left(-\lambda \left\| \boldsymbol{x} - \boldsymbol{c}_l \right\|^2\right)}, \tag{7}$$

where λ is a smoothing parameter. Then, (4) is described as

$$\tilde{\boldsymbol{f}}(\boldsymbol{p}) \approx \sum_{k=1}^{K} \alpha_k(\boldsymbol{f}(\boldsymbol{p})) \frac{\boldsymbol{\xi}_{\boldsymbol{c}_k}(\boldsymbol{p})}{\zeta_{\boldsymbol{c}_k}(\boldsymbol{p})}, \tag{8}$$

Since natural images generally consist of some dominant colors and their gradations, our method can achieve sufficient accuracy if K is sufficiently-large. As a result, (8) contains $K(M + 1)$ convolutions where the number of convolutions is counted independently for each channel (i.e. M-channel convolution is counted as M convolutions). Note that all the $\boldsymbol{\xi}_{\boldsymbol{c}_k}(\cdot)$ and $\zeta_{\boldsymbol{c}_k}(\cdot)$ are precomputed before performing (8).

C. Comparison with state-of-the-art methods

Our method outperforms the Yang method [13] in terms of computational complexity. Conceptually, the Yang method samples $\boldsymbol{c}_k$ from in a scalar quantization manner and then interpolate unsampled points in $\mathcal{R}$ from their neighboring $\boldsymbol{c}_k$. However, an image have non-uniform color distribution in general. In other words, we non-uniformly access $\mathcal{R}$ in (4) in practice. Scalar quantization cannot exploit the tendency of color distribution. Moreover, it suffers from *the curse of dimensionality*. The Yang method contains $B^M(M + 1)$ convolutions where B is the number of bins/channel and its most common choice is $B = 4$. Evidently, this approach works well for single-channel images [6] but shows limitation for multi-channel images (especially for hyper-spectral images).

Our method also shows a clear advantage over the Morerov method [12]. This method determines $\boldsymbol{c}_k$ in a vector quantization manner. However, it does not provide an alternative of interpolation between $\boldsymbol{c}_k$ (i.e., hard-assignment coding). This approach has a limitation of approximate accuracy. In order to reveal this fact, we examine a toy problem of color reduction using the image "lenna". Let us consider replacing each pixel color of the image to the linear combination

$$\boldsymbol{t} \approx \sum_{k=1}^{K} \alpha_k(\boldsymbol{t})\, \boldsymbol{c}_k.$$

Figure 1 exhibits results of hard- and soft-assignment codings where we used the k-means algorithm for clustering. The approximate accuracy between original and color-reduced images is quantified as the peak signal-to-noise ratio (PSNR). The two results of hard-assignment coding reveal many pseudo

(a) Input image (b) HA ($K = 8$) : 26.9 dB (c) HA ($K = 16$) : 29.7 dB (d) **SA** ($K = 8$) : 29.1 dB

Fig. 1. Color reduction of the image "lenna" where the soft-assignment coding uses $\lambda = 0.5$.

edge and texture regions even if $K = 16$. By contrast, the result of soft-assignment coding preserves edges and the most natural visual appearance. This is because soft-assignment coding can accurately represent gradation regions composed of some dominant colors. Evidently, these results support that soft-assignment coding outperforms hard-assignment coding.

D. Extension to constant-time algorithms

In both the ScBF and the McBF, their acceleration algorithms are easily extended to constant-time algorithms where *constant-time* means that computational complexity does not depend on filter window size (i.e. $O(1)$ time per pixel). This is because, if the convolutions of (5) and (6) are operated by an constant-time filtering algorithm, it can be a constant-time BF. In our method, (8) can be operated in constant-time by precomputing $\boldsymbol{\xi}_{\boldsymbol{c}_k}$ and $\zeta_{\boldsymbol{c}_k}$. For example, Gaussian spatial kernel and box spatial kernel, which are widely-used spatial kernels in the BF, can be convolved in constant-time such as the integral image [18], [19], recursive filtering [20]–[22], and short-time spectral approaches [23], [24]. Hence, our method employs the constant-time Gaussian filter proposed in [24].

IV. Experiments and Discussion

This section evaluates the computational complexity and the approximate accuracy of our method through several experiments using natural images. The test environment mounts on Intel Xeon CPU 3.70GHz and 32GB main memory. All the comparators are implemented in C++ and our implementations do not explicitly use parallel architecture such as multicore processing and vector computing. Test images are "Kodak Photo CD", which contains 24 RGB images with the size of 512×768 or 768×512. Note that $D = 2$, $M = 3$, and each channel has 8-bit the dynamic range (i.e., $\mathcal{R} \in \{0, 1, \ldots, 255\}^M$).

Firstly, we confirm approximate accuracy of our method. The accuracy is quantified as the PSNR between the naive McBF of (1) and our method of (8). Figure 2 plots the relationship between the number of convolutions and the PSNR where the parameters (σ_s, σ_r) are (5,30), (10,30), and (10,45). Our method achieves sufficient accuracy (i.e. 40 [dB]) at 64–80 convolutions (i.e. $K = 16$ to 20). Under the same parameters and test images, the Yang method achieves almost the same PSNR by using 256 convolutions (i.e. $B = 4$). Specifically, our method can run approximately 3.5 to 4.0 times faster than the Yang method without a loss of accuracy. Note that the randomized approaches [14], [15] also requires hundreds of convolutions. These results also show that $\lambda = \frac{0.5}{255}$ shows the best performance tradeoff and we can reduce more convolutions when σ_r is larger.

Secondly, Fig. 3 shows the image "kodim04" in the test set and its resulting images of the naive McBF, the Yang method, and our method ($K = 10$) where all the images are zoomed to facilitate visual assessment. The Yang method lost some edges (see her eyes) due to scalar quantization that neglects the color distribution. Our method more correctly preserved edge and also flat regions than the Yang method. On the other hand, if a target image consists of many dominant colors, both of the Yang method and our method showed pseudo contours of colors. How to find the correct K for each image is a remaining problem for our method.

Lastly, we mention the computational time of the naive McBF and our method. Under the parameters $(\sigma_s, \sigma_r) = (10, 30)$, the naive McBF consumed approximately 7.5 [s]; by contrast, our method ($K = 16$) took less than 1.0 [s] including clustering process. The naive McBF has the computational time proportional to σ_s but our method can run at a constant speed regardless of σ_s. Consequently, our method can be an alternative of the naive McBF for many image processing applications.

V. Conclusions

This paper presented an acceleration algorithm for the McBF based on soft-assignment coding. Our method provided advantages of vector quantization and linear interpolation, which two state-of-the-art methods had individually provided. Many existing methods required hundreds of convolutions; by contrast, our method succeeded in reducing to 64–80 convolutions without a loss in approximate accuracy. This performance improvement will contribute to expand the range of applications of the McBF. Future work will generalize our proposed ideas to multi-lateral filtering or non-local mean filtering [25].

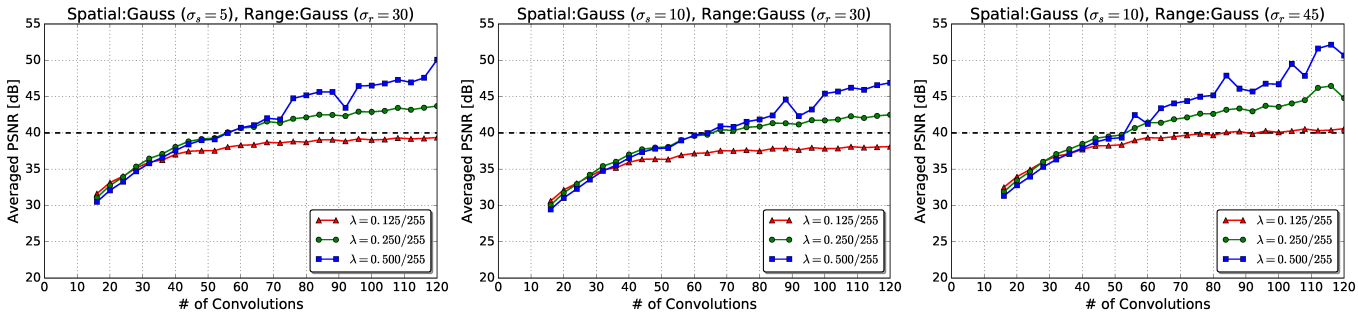

Fig. 2. The number of convolutions versus approximate accuracy (PSNR [dB] averaged over the 24 test images).

(a) Input image (b) Naive McBF (∞ [dB]) (c) Yang *et al.* (35.4 dB) (d) **Ours** (34.4 dB)

Fig. 3. Zoomed results of the image "kodim04". Parameters: (b,c,d) $\sigma_s = 3$, $\sigma_r = 30$, (c) $B = 4$, and (d) $K = 10$, $\lambda = 0.5$.

Acknowledgment

REFERENCES

[1] V. Aurich and J. Weule, "Non-linear Gaussian filters performing edge preserving diffusion," in *Mustererkennung 1995, 17. DAGM-Symposium*, 1995, pp. 538–545.

[2] S. M. Smith and J. M. Brady, "SUSAN — a new approach to low level image processing," *Int. J. Comput. Vis. (IJCV)*, vol. 23, no. 1, pp. 45–78, 1997.

[3] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Jan. 1998, pp. 839–846.

[4] F. Durand and J. Dorsey, "Fast bilateral filtering for the display of high-dynamic-range images," *ACM Trans. Graph. (Proc. SIGGRAPH)*, vol. 21, no. 3, pp. 257–266, July 2002.

[5] S. Paris and F. Durand, "A fast approximation of the bilateral filter using a signal processing approach," *Int. J. Comput. Vis. (IJCV)*, vol. 81, no. 1, pp. 24–52, Jan. 2009.

[6] Q. Yang, K. H. Tan, and N. Ahuja, "Real-time O(1) bilateral filtering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2009, number 1, pp. 557–564.

[7] S. Yoshizawa, A. Belyaev, and H. Yokota, "Fast Gauss bilateral filtering," *Computer Graphics Forum*, vol. 29, no. 1, pp. 60–74, Mar. 2010.

[8] K. N. Chaudhury, "Acceleration of the shiftable O(1) algorithm for bilateral filtering and nonlocal means," *IEEE Trans. Image Process.*, vol. 22, no. 4, pp. 1291–1300, Apr. 2013.

[9] K. Sugimoto and S. Kamata, "Compressive bilateral filtering," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 3357–3369, Nov. 2015.

[10] K. Sugimoto, T. Breckon, and S. Kamata, "Constant-time bilateral filter using spectral decomposition," in *Proc. IEEE Int. Conf. Image Process. (ICIP) (to appear)*, Sept. 2016.

[11] H. Peng, R. Rao, and S. A. Dianat, "Multispectral image denoising with optimized vector bilateral filter," *IEEE Trans. Image Process.*, vol. 23, no. 1, pp. 264–273, Jan. 2014.

[12] M. G. Mozerov and J. van de Weijer, "Global color sparseness and a local statistics prior for fast bilateral filtering," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5842–5853, Dec. 2015.

[13] Q. Yang, N. Ahuja, and K.-H. Tan, "Constant time median and bilateral filtering," *Int. J. Comput. Vis. (IJCV)*, vol. 112, no. 3, pp. 307–318, May 2015.

[14] C. Karam, C. Chen, and K. Hirakawa, "Stochastic bilateral filter for high-dimensional images," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sept. 2015, vol. 2, pp. 192–196.

[15] S. Ghosh and K. N. Chaudhury, "Fast bilateral filtering of vector-valued images," in *Proc. IEEE Int. Conf. Image Process. (ICIP) (to appear)*, Sept. 2016.

[16] J. C. van Gemert, C. J. Veenman, A. W. M. Smeulders, and J. M. Geusebroek, "Visual word ambiguity," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 7, pp. 1271–1283, July 2010.

[17] L. Liu, L. Wang, and X. Liu, "In defense of soft-assignment coding," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Nov. 2011, pp. 2486–2493.

[18] F. C. Crow, "Summed-area tables for texture mapping," *ACM Trans. Graph. (Proc. SIGGRAPH)*, vol. 18, no. 3, pp. 207–212, July 1984.

[19] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Dec. 2001, vol. 1, pp. 511–518.

[20] R. Deriche, "Fast algorithms for low-level vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 1, pp. 78–87, 1990.

[21] I. T. Young and L. J. van Vliet, "Recursive implementation of the Gaussian filter," *Signal Process.*, vol. 44, no. 2, pp. 139–151, June 1995.

[22] L. J. van Vliet, I. T. Young, and P. W. Verbeek, "Recursive Gaussian derivative filters," in *Proc. Int. Conf. Pattern Recognition (ICPR)*, 1998, vol. 1, pp. 509–514.

[23] K. Sugimoto and S. Kamata, "Fast Gaussian filter with second-order shift property of DCT-5," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sept. 2013, pp. 514–518.

[24] K. Sugimoto and S. Kamata, "Efficient constant-time Gaussian filtering with sliding DCT/DST-5 and dual-domain error minimization," *ITE Trans. Media Technol. Appl.*, vol. 3, no. 1, pp. 12–21, 2015.

[25] A. Buades, B. Coll, and J. M. Morel, "A non-local algorithm for image denoising," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2005, vol. 2, pp. 60–65.