

# CONSTANT-TIME BILATERAL FILTER USING SPECTRAL DECOMPOSITION

Kenjiro Sugimoto<sup>†</sup>, Toby Breckon<sup>‡</sup>, and Sei-ichiro Kamata<sup>†</sup>

<sup>†</sup> Waseda University, Graduate School of Information, Production and Systems, Japan

<sup>‡</sup> Durham University, School of Engineering and Computing Sciences, United Kingdom

## ABSTRACT

This paper presents an efficient constant-time bilateral filter where *constant-time* means that computational complexity is independent of filter window size. Many state-of-the-art constant-time methods approximate the original bilateral filter by an appropriate combination of a series of convolutions. It is important for this framework to optimize the performance tradeoff between approximate accuracy and the number of convolutions. The proposed method achieves the optimal performance tradeoff in a least-squares manner by using spectral decomposition under the assumption that images consist of discrete intensities such as 8-bit images. This approach is essentially applicable to arbitrary range kernel. Experiments show that the proposed method outperforms state-of-the-art methods in terms of both computational complexity and approximate accuracy.

**Index Terms**— Image filtering, Constant-time bilateral filter, Spectral decomposition

## 1. INTRODUCTION

The bilateral filter (BF) [1–3] has played a fundamental role as an edge-preserving smoothing tool in image processing, computer vision and computer graphics, (e.g. for denoising [4–6], high-dynamic range imaging [7–9] and stereo vision [10, 11]; see [12] for more applications). The BF enables us to smooth an image while preserving edges and textures by using filter weights determined from both spatial kernel (pixel position) and range kernel (pixel intensity). Following the original work, many improved methods have been actively proposed to further enhance smoothing quality [4, 5] and to reduce computational complexity [7, 8, 13–21]. A major drawback of the original BF is the computational complexity depending on its filter window size. This causes unacceptable running time in a recent trend toward high-resolutional and volume image processing because they tend to require large filter window size. We therefore focus on *constant-time* ( $O(1)$ ) BF — an approximate BF that can run in  $O(1)$  time per pixel with a slight sacrifice of accuracy.

Many methods of  $O(1)$  BF share the following framework. Firstly, the original BF is approximated by an appropriate combination of a series of convolutions, (e.g. by splatting/slicing technique [8, 15, 16], histogram technique [14] or range kernel decomposition [20, 21]). Secondly, each convolution is operated by an  $O(1)$  method, (e.g. using integral image [22, 23], recursive filter [24–26] or short-time spectra [27, 28]). Under this framework,  $O(1)$  BF involves a performance tradeoff between computational complexity and approximate accuracy. The former can be quantified as the number of the convolutions; the latter indicates how exactly it emulates the output images of the original BF. Naturally, this observation prompts the question about optimality of the performance tradeoff.

Sugimoto and Kamata [21] attempted to answer this question from a viewpoint of compressibility. Based on their discussion,

they developed an efficient  $O(1)$  BF called *compressive BF*, which can achieve a nearly-optimal performance tradeoff in a least-squares manner. However, this method has the following limitations. Firstly, its performance is nearly-optimal only when the number of convolutions are sufficiently-large due to assumptions used for error estimation. Secondly, it focuses only on Gaussian range kernel, not on arbitrary range kernels. Regarding other existing  $O(1)$  BFs, although arbitrary range kernels are supported in some recent work [8, 14–16, 29, 30], none of them have explicitly addressed optimality from a theoretical viewpoint.

This paper presents an  $O(1)$  BF that provides the optimal performance tradeoff in a least-squares manner. We optimize range kernel decomposition by simply employing spectral decomposition under the assumption that images consist of equally-spaced discrete intensities. This assumption is trivial for most practical applications since digital images general satisfy this assumption within an 8-bit representational space. The proposed method eliminates the aforementioned limitations because spectral decomposition guarantees the optimal solution in a least-squares manner and is essentially applicable to arbitrary range kernels. Moreover, we discuss how to reduce the computational complexity of the spectral decomposition, which exploits mathematical properties of real symmetric Toeplitz matrices. Experiments validate that our method outperforms state-of-the-art methods in terms of performance tradeoff.

## 2. EXISTING WORK AND REMAINING PROBLEMS

This section first summarizes the original BF [1–3] and relevant cross/joint extensions [4, 5]. We then show a general framework of the state-of-the-art  $O(1)$  BFs [15, 16, 18–21] from a viewpoint of range kernel decomposition and discuss their remaining problems.

### 2.1. Bilateral Filter

Consider smoothing a  $D$ -dimensional grayscale image on spatial domain  $\Omega \subset \mathbb{Z}^D$ . The BF smooths a target image  $f : \Omega \rightarrow \mathbb{R}$  to yield its smoothed image  $g : \Omega \rightarrow \mathbb{R}$ , by using a guide image  $f_* : \Omega \rightarrow \mathbb{R}$  if needed. The weights around a pixel position  $\mathbf{p} \in \Omega$  are determined from the positions/intensities of its neighboring pixels  $\mathcal{N}(\mathbf{p}) \subset \Omega$ .

**Definition 2.1.1. (Bilateral filter [1–5])**

$$g(\mathbf{p}) := \frac{\sum_{\mathbf{q} \in \mathcal{N}(\mathbf{p})} w_s(\mathbf{p}, \mathbf{q}) w_r(f_*(\mathbf{p}), f_*(\mathbf{q})) f(\mathbf{q})}{\sum_{\mathbf{q} \in \mathcal{N}(\mathbf{p})} w_s(\mathbf{p}, \mathbf{q}) w_r(f_*(\mathbf{p}), f_*(\mathbf{q}))}, \quad (1)$$

where  $w_s : \mathbb{Z}^D \times \mathbb{Z}^D \rightarrow \mathbb{R}$  is spatial kernel and  $w_r : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$  is range kernel. This definition forms the original BF [1–3] if  $f_* = f$  and a cross/joint extension [4, 5] otherwise.

The denominator normalizes the weight so that they amount to unity. The spatial and range kernels are defined according to the intended use with one of the most common choices being the Gaussian kernel.

**Definition 2.1.2. (Gaussian spatial/range kernel)**

$$w_s(\mathbf{p}, \mathbf{q}) := e^{-\frac{\|\mathbf{q}-\mathbf{p}\|^2}{2\sigma_s^2}}, \quad w_r(t, s) := e^{-\frac{(t-s)^2}{2\sigma_r^2}}, \quad (2)$$

where  $\sigma_s, \sigma_r \in \mathbb{R}$  are the scale of Gaussian spatial and range kernels, respectively, and  $\|\cdot\|$  denotes the  $\ell_2$ -norm of a vector.

The BF has the typical problem that the computational complexity depends on the filter window size  $|\mathcal{N}(\mathbf{p})|$ . This causes unacceptable running time in a recent trend towards high-resolution or volume image processing that requires large filter window sizes.

**2.2.  $O(1)$  Bilateral Filters**

In order to address the problem of large filter window size, many  $O(1)$  BFs have been proposed in the past [7, 13–21]. They have attempted to approximate the original BF as accurately as possible. We describe a general framework of  $O(1)$  BFs from a unified perspective of range kernel decomposition. Consider decomposing a range kernel by the separable form

$$w_r(t, s) = \sum_{k=0}^{\infty} \phi_k(t) \psi_k(s), \quad (3)$$

where  $t, s \in \mathbb{R}$  indicate intensity variables. Substituting (3) for (1),

$$g(\mathbf{p}) = \frac{\sum_{k=0}^{\infty} \psi_k(f_*(\mathbf{p})) \Phi_k(\mathbf{p})}{\sum_{k=0}^{\infty} \psi_k(f_*(\mathbf{p})) \bar{\Phi}_k(\mathbf{p})}, \quad (4)$$

where  $\Phi_k(\cdot), \bar{\Phi}_k(\cdot)$  are called *component images*, defined by

$$\Phi_k(\mathbf{p}) = \sum_{\mathbf{q} \in \mathcal{N}(\mathbf{p})} w_s(\mathbf{p}, \mathbf{q}) \phi_k(f_*(\mathbf{q})) f(\mathbf{q}), \quad (5)$$

$$\bar{\Phi}_k(\mathbf{p}) = \sum_{\mathbf{q} \in \mathcal{N}(\mathbf{p})} w_s(\mathbf{p}, \mathbf{q}) \phi_k(f_*(\mathbf{q})). \quad (6)$$

Obviously, (5) and (6) indicate convolutions to transformed image  $\phi_k(f_*(\mathbf{q})) f(\mathbf{q})$  and  $\phi_k(f_*(\mathbf{q}))$ , respectively. These spatial filters can be replaced to an adequate  $O(1)$  method such as [22–28]. After precomputing the component images, (4) is also computable in  $O(1)$  per pixel if we can well-approximate (3) by a few  $K$  terms. It is in essence a matter for this framework to find an efficient way to decompose (3).

We particularize representative examples of the above framework. The splatting/slicing techniques [15, 16] represent a range kernel as  $w_r(t, s) = \int_{\mathbb{R}} w_r(t, r) \delta(s - r) dr$  and then replace the integral to a summation by aggressively quantizing its integral range where  $\delta(\cdot)$  is the Dirac delta function. This result can be identified with (3). Chaudhury [19, 20] discussed the property of (3) as shiftable kernel. For example, Gaussian range kernel can be well approximated by a linear combination of a few cosine terms. This approximation forms (3) since each term can be decomposed into  $\cos(t - s) = \cos(t) \cos(s) + \sin(t) \sin(s)$ . Sugimoto and Kamata [21] showed how to approximate Gaussian range kernel by fewer cosine terms, which are derived from Fourier expansion and optimization of its period length. They also demonstrated empirically that this method provided a nearly-optimal performance tradeoff in a least-squares manner if  $K$  is sufficiently-large.

**2.3. Remaining Problems**

We point out that [21] has the following limitations. First, the performance tradeoff is nearly-optimal, not optimal, due to some assumptions introduced for error estimation of range kernel. For example, the performance tradeoff closely approaches the optimal one if  $K$  is sufficiently-large; by contrast, the accuracy declines significantly otherwise. Second, this method covers Gaussian range kernel only. It should be discussed for arbitrary range kernel because range kernel relates to the noise model of image intensities. Hence, it is important to develop an  $O(1)$  BF that provides the optimal performance tradeoff for arbitrary range kernel over a wide range of parameters.

**3. PROPOSED METHOD**

This section presents an  $O(1)$  BF that provides an optimal performance tradeoff in a least-squares manner by spectral decomposition. This can be achieved by exploiting a typical feature of digital images and is essentially applied to arbitrary range kernel. Due to space limitation, this paper only shows the case of Gaussian range kernel.

**3.1. Spectral Decomposition of Range Kernel**

First of all, we assume that the guide image  $f_*$  consists of  $M$ -step discrete intensities  $\mathcal{Z} = \{0, \dots, M - 1\}$ , i.e.,  $f_* : \Omega \rightarrow \mathcal{Z}$  and  $w_r : \mathcal{Z} \times \mathcal{Z} \rightarrow \mathbb{R}$ . This assumption is natural in most real applications since they generally process 8-bit images ( $M = 256$ ). Under this assumption, all the possible values of the range kernel can be arranged in the matrix

$$\mathbf{W} = \{w_r(t, s)\}_{t, s=0}^{M-1} = [\mathbf{w}_0, \dots, \mathbf{w}_{M-1}] \in \mathbb{R}^{M \times M},$$

which the column vectors  $\mathbf{w}_s \in \mathbb{R}^M$  are called *shifted range kernels*. Note that  $\mathbf{W}$  is a real symmetric Toeplitz matrix. We produce least-squares approximations by applying low-rank matrix factorization to  $\mathbf{W}$ . Its spectral decomposition can be described as

$$\mathbf{W} = \sum_{k=0}^{M-1} \lambda_k \mathbf{u}_k \mathbf{u}_k^\top, \quad (7)$$

where  $\lambda_k \in \mathbb{R}$  are eigenvalues,  $\mathbf{u}_k \in \mathbb{R}^M$  are their corresponding eigenvectors and we assume  $|\lambda_0| \geq \dots \geq |\lambda_{M-1}|$  and  $\|\mathbf{u}_k\|_2 = 1$ . An element of  $\mathbf{W}$  is computed by

$$w_r(t, s) = \mathbf{W}[t, s] = \sum_{k=0}^{M-1} \lambda_k \mathbf{u}_k[t] \mathbf{u}_k[s], \quad (8)$$

where  $[\cdot]$  denotes an index operator for accessing to an element of a vector or a matrix. Obviously, the spectral decomposition of (8) corresponds to the range kernel decomposition of (3). Subsequently,  $w_r(t, s)$  can be well approximated by truncating (8) at index  $K$ . We represent the  $K$ -truncated range kernel matrix as  $\hat{\mathbf{W}}_K = \sum_{k=0}^{K-1} \lambda_k \mathbf{u}_k \mathbf{u}_k^\top$ , which is the optimal solution in a least-squares manner.

Figure 1 illustrates a specific example of the above approach with Gaussian range kernel. Figure 1(a) plots some of shifted Gaussian range kernels. The matrix  $\mathbf{W}$  can be understood as a data matrix that consists of all the shift range kernels  $\mathbf{w}_s$ . Figure 1(b) shows the eigenvectors  $\mathbf{u}_k$ . This result reveals a mathematical property of symmetric Toeplitz matrices [31–34]. Let  $\mathbf{J}$  denote the exchange matrix, which has 1 on the second diagonal and 0 otherwise. We say a vector  $\mathbf{v}$  is symmetric if  $\mathbf{v} = \mathbf{J}\mathbf{v}$  or skew-symmetric if  $\mathbf{v} = -\mathbf{J}\mathbf{v}$ .

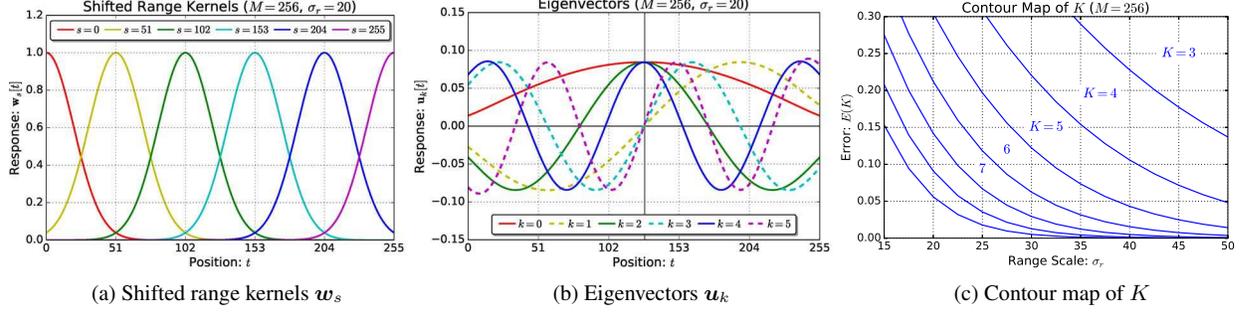


Fig. 1. A specific example of Gaussian range kernels where  $M = 256$  and  $\sigma_r = 20$ .

The eigenvectors  $\mathbf{u}_k$  are symmetric if  $k$  is even and skew-symmetric otherwise. This is Figure 1(c) plots a contour map of  $K$ : curve relations between range scale  $\sigma_r$  and the approximate error of the range kernel for each  $K$ . The approximate error is quantified as a normalized energy loss by

$$E^2(K) = \frac{\|\mathbf{W} - \hat{\mathbf{W}}_K\|_F^2}{\|\mathbf{W}\|_F^2} = \frac{\sum_{k=K}^{M-1} \lambda_k^2}{\sum_{k=0}^{M-1} \lambda_k^2}, \quad (9)$$

where  $\|\cdot\|_F$  indicates the Frobenius norm of a matrix. The curves clarify that the larger  $K$  reconstructs the more exact range kernel and they are well approximated even if  $K$  is small. This contour map enables us to determine  $K$  from a given tolerance  $\tau$ , (e.g. the minimum  $K$  that satisfies  $E(K) \leq \tau$ ). As a result, our method can well approximate the BF by using a few iterations of convolutions.

### 3.2. Acceleration Techniques

It is important for our method to perform the aforementioned spectral decomposition in negligible running time. We first introduce an acceleration technique that exploits a mathematical property of the real symmetric Toeplitz matrix  $\mathbf{W}$ . It has been well studied to efficiently solve eigenvalue problems of symmetric Toeplitz matrices [31–34]. We mention the approach of [31] for the case of even  $M$ . As Fig. 1(b) shows, the eigenvectors  $\mathbf{u}_k$  are symmetric if  $k$  is even or skew-symmetric otherwise. From this fact, the same eigenvectors can be obtained by solving two eigenvalue problems of half-sized matrices. This technique drastically reduces the computational complexity of spectral decomposition. Hence, we compute only the first  $K$  eigenvalues/eigenvectors by combining the power iteration and this matrix decomposition approach.

Another effective technique enables us to eliminate one convolution in (6). If we assume  $\phi_k(t) = \mu$ , i.e., a constant, (6) can be operated as a multiplication of  $\mu$  and the total value of spatial kernel without a convolution. This assumption holds in the compressive BF [21] since its first cosine term is a constant. By contrast, it does not in our method since any  $\mathbf{u}_k$  is not constant in general. In order to utilize this benefit, we redefine  $\mathbf{W} = \{w_r(t, s) - \mu\}_{t, s=0}^{M-1}$  and rewrite (8) as

$$w_r(t, s) = \mathbf{W}[t, s] + \mu + \sum_{k=0}^{M-1} \lambda_k \mathbf{u}_k[t] \mathbf{u}_k[s]. \quad (10)$$

We suggest  $\mu = \frac{1}{M^2} \sum_{t, s=0}^{M-1} w_r(t, s)$  to minimize  $\|\mathbf{W}\|_F$ . Subsequently, spectral decomposition is applied to the new  $\mathbf{W}$ . This technique enhances the approximate accuracy without operating a convolution.

### 3.3. Algorithm Procedure and Advantages

**Algorithm 1** The proposed  $O(1)$  bilateral filter

- 1:  $\triangleright \mathbf{f}$ : target image,  $\mathbf{f}_*$ : guide image
- 2:  $\triangleright \sigma_s$ : spatial scale,  $\sigma_r$ : range scale,  $K$ : truncation index
- 3: **function** PROPOSEDMETHOD( $\mathbf{f}, \mathbf{f}_*, \sigma_s, \sigma_r, K$ )
- 4:  $\triangleright$  Precomputing phase
- 5:  $\mu = \frac{1}{M^2} \sum_{s=0}^{M-1} \sum_{t=0}^{M-1} w_r(t, s)$
- 6:  $\mathbf{W} \leftarrow \{w_r(t, s) - \mu\}_{t, s=0}^{M-1}$
- 7:  $\{\lambda_k, \mathbf{u}_k\}_{k=0}^{K-1} \leftarrow \text{TRUNCATED EVD}(\mathbf{W}, K)$
- 8:  $\triangleright$  Filtering phase
- 9:  $\bar{\mathbf{b}} \leftarrow \text{CONVOLUTION}(\sigma_s, \mu \mathbf{1})$   $\triangleright$  Replaced to multiplication
- 10:  $\mathbf{b} \leftarrow \text{CONVOLUTION}(\sigma_s, \mu \mathbf{f})$
- 11: **for**  $k \leftarrow 0$  to  $K - 1$  **do**
- 12:     **for**  $\forall p$  **do**
- 13:          $\mathbf{x}[p] \leftarrow \mathbf{u}_k[\mathbf{f}_*[p]]$
- 14:          $\bar{\Phi} \leftarrow \text{CONVOLUTION}(\sigma_s, \mathbf{x})$
- 15:          $\Phi \leftarrow \text{CONVOLUTION}(\sigma_s, \mathbf{x} \otimes \mathbf{f})$
- 16:          $\bar{\mathbf{b}} \leftarrow \bar{\mathbf{b}} + \lambda_k \mathbf{x} \otimes \bar{\Phi}$
- 17:          $\mathbf{b} \leftarrow \mathbf{b} + \lambda_k \mathbf{x} \otimes \Phi$
- 18: **return**  $\mathbf{b} \oslash \bar{\mathbf{b}}$

Algorithm 1 describes an algorithm procedure of our method where images are represented as vector forms for simplicity. The operators  $\otimes$  and  $\oslash$  denote element-wise multiplication and division, respectively. The function  $\text{CONVOLUTION}(\cdot)$  requires to be adequately replaced to an arbitrary  $O(1)$  filter for  $O(1)$  BF. Let  $N$  be the number of convolutions. Our method has the computational complexity dominated by  $N = 2K + 1$ , where note that the convolution in line 9 is actually replaced to a multiplication, as mentioned above. Since the precomputing phase depends only on  $\sigma_r$  and  $K$ , not on image content, it is sufficient to run once when the parameters are unchanged, (e.g. for video sequence).

We clarify the major advantages over the existing state-of-the-art methods [20, 21]. Firstly, our method can cover arbitrary range kernels; by contrast, the existing methods mainly discussed Gaussian range kernel only. Secondly, our method provides optimal performance tradeoff in a least-squares manner under any parameter setting. However, [21] imposes a limitation on parameter range, (e.g.  $K$  has to be sufficiently-large and  $\sigma_r \leq 6M$ ). Finally, we can adjust  $N$  more flexibly than the existing methods. In existing methods [20, 21],  $N$  required has a fixed granularity of step-wise scaling by  $4K$ . By contrast, our method offers more flexibility with a granularity of  $2K$  step-wise scaling. Consequently, our method is

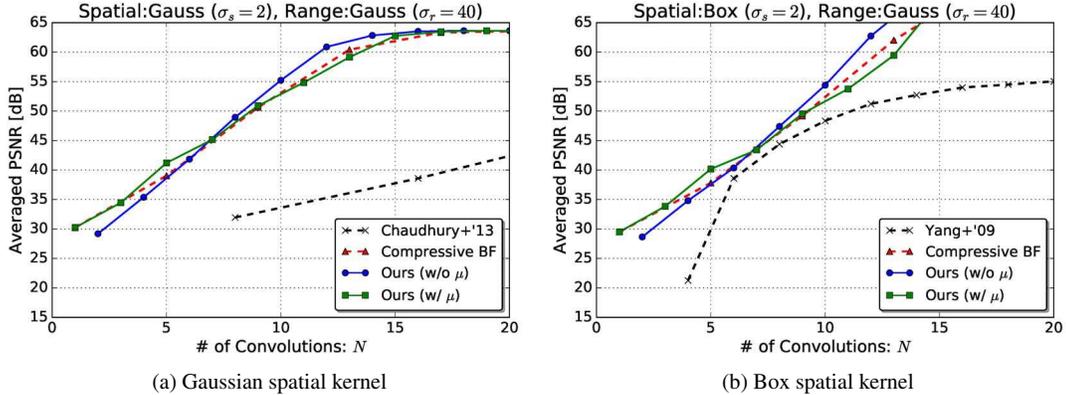


Fig. 2. PSNR comparison using Kodak 24 Images where  $M = 256$ ,  $\sigma_s = 2$  and  $\sigma_r = 40$ .

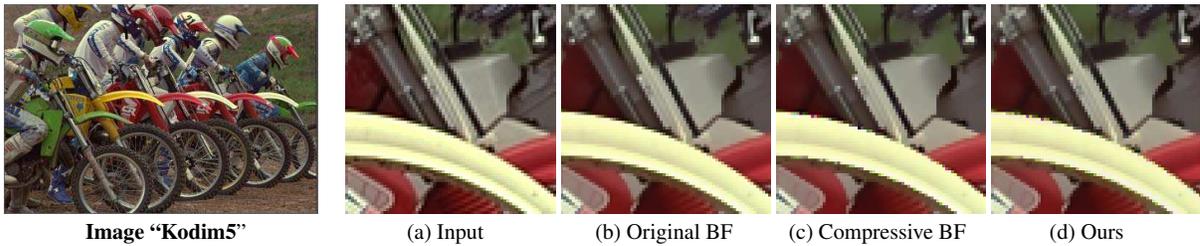


Fig. 3. Zoomed images for visual error assessment where  $\sigma_s = 2$ ,  $\sigma_r = 20$  and  $N = 13$ . The PSNRs are (c) 41.62 [dB] and (d) 41.90 [dB].

applicable to a wider range of applications than prior state-of-the-art methods.

#### 4. EXPERIMENTS AND DISCUSSION

This section evaluates the performance of our method through several experiments. The comparators are Yang et al. [16], Chaudhury [20], the compressive BF [21] and our method. We eliminated any content-dependent techniques such as the MAXFILTER technique in [20] to evaluate their worst-case performance. Their implementations are all written in C++ for a fair comparison. All the methods do not explicitly use parallel processing architecture such as vectorized/multicore computation. Test environment mounts on an Intel Core i5 2.67GHz CPU with main memory 8GB. Test set is the Kodak Photo CD, which contains 24 RGB images with the size of  $512 \times 768$  or  $768 \times 512$ . Each channel is 8-bits ( $M = 256$ ) and independently filtered in this experiments.

Figure 2 shows the relationship between  $N$  required and Peak Signal-to-Noise Ratio (PSNR) [dB] averaged over all the 24 test images. The PSNR is computed between resulting images of each method and the original BF as exact results. Fig. 2(a) reveals that our method significantly outperforms the Chaudhury method and shows slightly higher PSNR than the compressive BF where all the methods use the  $O(1)$  algorithm of [27, 28] for Gaussian convolution. Using  $\mu$  in our method is effective when range kernel is roughly approximated. Note that the Chaudhury method and the compressive BF focus only on Gaussian range kernel but our method is essentially applicable to arbitrary range kernels. In Fig. 2(b), our method achieves higher PSNR than the Yang method when  $N < 6$  or  $10 < N$ . This tendency is observed under any values of  $\sigma_r$ .

Next, we visually assess the output quality of the original BF,

the compressive BF and our method. Figure 3 shows their actual output images zoomed to facilitate visual assessment where  $\sigma_s = 2$ ,  $\sigma_r = 20$  and  $N = 13$ . Our method suppresses spike noise on the edge of the white wing as compared with the compressive BF.

The running time is another important criterion. In the most common case of  $M = 256$ , the proposed spectral decomposition specialized in symmetric Toeplitz matrices took approximately 2 [ms]. On the other hand, naive power iteration took 30 [ms]. Total running time is dominated by a series of convolutions and each Gaussian convolution takes 19 [ms/Mpixels] per iteration in [28]. Hence, the precomputing time is trivial for most applications.

#### 5. CONCLUSIONS

This paper presented an efficient  $O(1)$  BF that provides an optimal performance tradeoff between approximate accuracy and computational complexity in a least-squares manner. The optimality was achieved by spectral decomposition of a matrix generated from range kernel. Our method assumed that images have discrete intensities such as typical 8-bit images but this is natural for most applications. Even if images have continuous tone, most practical cases could be well approximated by a discrete tone with sufficiently-many steps. Future work will thoroughly examine various range kernels such as exponential range kernel.

#### Acknowledgment

This work was supported by JSPS KAKENHI Grant Number JP16K16092. The authors would like to thank Norishige Fukushima for valuable discussions about edge-preserving smoothing filters.

## 6. 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] G. Petschnigg, R. Szeliski, M. Agrawala, M. Cohen, H. Hoppe, and K. Toyama, "Digital photography with flash and no-flash image pairs," *ACM Trans. Graph. (Proc. SIGGRAPH)*, vol. 23, no. 3, pp. 664–672, Aug. 2004.
- [5] E. Eisemann and F. Durand, "Flash photography enhancement via intrinsic relighting," *ACM Trans. Graph. (Proc. SIGGRAPH)*, vol. 23, no. 3, pp. 673–678, Aug. 2004.
- [6] A. Buades, B. Coll, and J. M. Morel, "A review of image denoising algorithms, with a new one," *Multiscale Modeling and Simulation*, vol. 4, no. 2, pp. 490–530, Jan. 2005.
- [7] 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.
- [8] J. Chen, S. Paris, and F. Durand, "Real-time edge-aware image processing with the bilateral grid," *ACM Trans. Graph. (Proc. SIGGRAPH)*, vol. 26, no. 3, pp. 103.1–103.9, July 2007.
- [9] J. Kuang, G. M. Johnson, and M. D. Fairchild, "iCAM06: a refined image appearance model for HDR image rendering," *J. Visual Communication and Image Representation*, vol. 18, no. 5, pp. 406–414, Oct. 2007.
- [10] Q. Yang, R. Yang, J. Davis, and D. Nister, "Spatial-depth super resolution for range images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2007.
- [11] Q. Yang, L. Wang, R. Yang, H. Stewénius, and D. Nistér, "Stereo matching with color-weighted correlation, hierarchical belief propagation, and occlusion handling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 3, pp. 492–504, Mar. 2009.
- [12] S. Paris, P. Kornprobst, J. Tumblin, and F. Durand, "A gentle introduction to bilateral filtering and its applications," in *ACM SIGGRAPH classes*, 2008.
- [13] B. Weiss, "Fast median and bilateral filtering," *ACM Trans. Graph. (Proc. SIGGRAPH)*, vol. 25, no. 3, pp. 519–526, July 2006.
- [14] F. Porikli, "Constant time O(1) bilateral filtering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2008, pp. 1–8.
- [15] 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.
- [16] 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.
- [17] S. Yoshizawa, A. Belyaev, and H. Yokota, "Fast Gauss bilateral filtering," *Computer Graphics Forum*, vol. 29, no. 1, pp. 60–74, Mar. 2010.
- [18] K. N. Chaudhury, D. Sage, and M. Unser, "Fast O(1) bilateral filtering using trigonometric range kernels," *IEEE Trans. Image Process.*, vol. 20, no. 12, pp. 3376–82, Dec. 2011.
- [19] K. N. Chaudhury, "Constant-time filtering using shiftable kernels," *IEEE Signal Processing Letters*, vol. 18, no. 11, pp. 651–654, Nov. 2011.
- [20] 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.
- [21] K. Sugimoto and S. Kamata, "Compressive bilateral filtering," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 3357–3369, Nov. 2015.
- [22] F. C. Crow, "Summed-area tables for texture mapping," *ACM Trans. Graph. (Proc. SIGGRAPH)*, vol. 18, no. 3, pp. 207–212, July 1984.
- [23] 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.
- [24] R. Deriche, "Fast algorithms for low-level vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 1, pp. 78–87, 1990.
- [25] 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.
- [26] 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.
- [27] 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.
- [28] 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.
- [29] B. K. Gunturk, "Fast bilateral filter with arbitrary range and domain kernels," *IEEE Trans. Image Process.*, vol. 20, no. 9, pp. 2690–2696, Sept. 2011.
- [30] S. Pan, X. An, and H. He, "Optimal O(1) bilateral filter with arbitrary spatial and range kernels using sparse approximation," *Mathematical Problems in Engineering*, vol. 2014, pp. 1–11, 2014.
- [31] A. Cantoni and P. Butler, "Eigenvalues and eigenvectors of symmetric centrosymmetric matrices," *Linear Algebra and its Applications*, vol. 13, no. 3, pp. 275–288, 1976.
- [32] G. Cybenko, "The numerical stability of the Levinson-Durbin algorithm for Toeplitz systems of equations," *SIAM J. Sci. and Stat. Comput.*, vol. 1, no. 3, pp. 303–319, 1980.
- [33] W. F. Trench, "Numerical solution of the eigenvalue problem for Hermitian Toeplitz matrices," *SIAM J. Matrix Anal. & Appl.*, vol. 10, no. 2, pp. 135–146, 1989.
- [34] John Makhoul, "On the eigenvectors of symmetric Toeplitz matrices," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 29, no. 4, pp. 868–872, Aug. 1981.