Principal Component Analysis for Accelerating Color Bilateral Filtering

Kazuya Ishikawa, Sou Oishi, and Norishige Fukushima

Nagoya Institute of Technology, Gokiso-Cho, Showa-Ku, Nagoya, Aichi, 466–8555, Japan.

ABSTRACT

In this paper, we propose to speed up bilateral filtering by principal component analysis (PCA)-based dimensionality compression method with constant-time bilateral filtering. Constant-time bilateral filtering speeds up the filtering by representing it as a summation of the multiple Gaussian filters. However, a simple implementation is of the order of $O(K^3)$ for color and suffers from the curse of dimensionality. A clustering-based approximation speedup solves this problem with an order of $O(K)$ or $O(K^2)$. PCA can provide a more informative signal relative to the transformation matrix. We have accelerated the process by converting the color information of the input image from 3-channel to 1-channel by PCA, considering the constant-time bilateral filter as joint bilateral filtering, and using the transformed image as a guide image. This dimensional reduction allowed us to filter images with sufficient accuracy at a higher speed.

Keywords: Bilateral Filter, PCA, Constant-time Bilateral Filter

1. INTRODUCTION

Bilateral filtering (BF)[1] is an edge-preserving smoothing filter used in various situations. Deblurring,[2,3] detail emphasis,[4] high- dynamic range imaging (HDR),[5] Dehaze,[6,7] Depth map correction,[8] Stereo-matching[9] are examples.

BF uses a composite kernel generated according to the distance between pixels (spatial kernel) and the difference in luminance values (range kernel). Since the shape of the composite kernel is different for each pixel of interest, the computational complexity is more significant than that of a linear filter such as Gaussian filtering (GF). In this case, the computational complexity depends on the kernel radius of the filter, resulting in a significant processing time.

To solve this problem, a constant-time BF (O(1)BF)[10] has been proposed, which speeds up BF by decomposing it into multiple spatially invariant filters, and its speed does not depend on its kernel radius. However, while a grayscale convolution works as fast as $O(K)$, the color implementation suffers from the curse of dimensionality, as in $O(K^3)$, where K is the approximate order and $K << r$. Clustering-based approaches[11,12] solve this curse, and the order is $O(K)$ or $O(K^2)$. However, the accuracy of the filter approximation depends on the clustering results, and the randomness of the clustering makes the filter results unstable. In addition, the clustering process itself is an overhead. The color information for the weight calculation is the curse of the color BF dimension.

Suppose the number of guide images can be reduced from 3 channels to 1 channel. In that case, the conventional color BF can be significantly accelerated because we can use the acceleration approaches for grayscale images. In this paper, we show that, in many cases, sufficient accuracy can be achieved by converting the guide image to a lower dimension using principal component analysis (PCA).

Further author information: (Send correspondence to N. Fukushima)
N. Fukushima: Web: https://fukushima.web.nitech.ac.jp/en/. This work was supported by JSPS KAKENHI (21H03465).

2. RELATED WORKS

2.1 Constant-time Bilateral Filter

First, the formulation of BF is shown below.

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

where $f_{\boldsymbol{p}}$ is the image luminance value, S is neighbor pixels. $w_s = e^{-\frac{\|\boldsymbol{q}-\boldsymbol{p}\|_2^2}{2\sigma_s^2}}$, $w_r = e^{-\frac{\|\boldsymbol{b}-\boldsymbol{a}\|_2^2}{2\sigma_r^2}}$ are weight functions called spatial and range kernel functions, respectively, and are defined by Gaussian functions. Note that σ_s and σ_r are smoothing parameters for each kernel.

Constant-time BF achieves speedup by decomposing BF into multiple Gaussian filters, shown below.

$$w_r(f_{\boldsymbol{p}}, f_{\boldsymbol{q}}) = \sum_{k=0}^{K-1} \phi_k(f_{\boldsymbol{p}}) \psi_k(f_{\boldsymbol{q}}) \tag{2}$$

$$\hat{f}_{\boldsymbol{p}} \approx \frac{\sum_{k=0}^{K-1} \phi_k(f_{\boldsymbol{p}}) \sum_{\boldsymbol{q}\in\mathcal{S}} w_s(\boldsymbol{p},\boldsymbol{q}) \{\psi_k(f_{\boldsymbol{q}}) f_{\boldsymbol{q}}\}}{\sum_{k=0}^{K-1} \phi_k(f_{\boldsymbol{p}}) \sum_{\boldsymbol{q}\in\mathcal{S}} w_s(\boldsymbol{p},\boldsymbol{q}) \{\psi_k(f_{\boldsymbol{q}})\}}. \tag{3}$$

K is the order of approximation. Since $\{\cdot\}$ in this expression can be regarded as an intermediate image, and in addition, the $\sum_{\boldsymbol{q}\in\mathcal{S}}$ part is a convolution operation of it, we can decompose BF into a combination of convolution operations for a fixed spatial kernel for a total of $2K$ times. However, in the case of color processing, the number of GFs to be decomposed increases with the number of channels, resulting in the curse of dimensionality: 8 GFs are required for gray, while 256 are required for color.

The weights in the form have various forms such as interpolation,[13] compressive,[14,15] and SVD.[16]

2.2 Compressive Bilateral Filter

Compressive bilateral filtering[14,15] uses Fourier series expansion to separate the range kernel from the variables. Since the range kernel is a Gaussian function and an even function, it can be approximated as follows:

$$w_r(x) \approx \hat{w_r}(x, K, T) = \alpha_0 + 2\sum_{k=1}^{K} \alpha_k \cos\left(\frac{2\pi}{T} kx\right), \tag{4}$$

where α_k, the Fourier coefficients, can be expressed as:

$$\alpha_k = \frac{1}{T}\int_{-T/2}^{T/2} w_r(x) \cos(\frac{2\pi}{T} kx) dx \tag{5}$$

Here, since a Gaussian function is used as the range kernel, the Fourier coefficients can be approximated as $\alpha_k \approx \frac{2}{T}\exp^{-\frac{1}{2}(\frac{2\pi}{T}k\sigma)^2}$. Substituting the range kernel approximated by the Fourier series expansion into the expression(3) and using the additive theorem of trigonometric functions, the output of the bilateral filter, $\hat{f}_{\boldsymbol{p}}$, can be expressed as follows.

$$\hat{f}_{\boldsymbol{p}} \approx \frac{\alpha_0 \tilde{f}_{\boldsymbol{p}} + 2\sum_{k=1}^{K} \alpha_k (\cos(\omega_k f_{\boldsymbol{p}}) \tilde{C}'_{\boldsymbol{p}} + \sin(\omega_k f_{\boldsymbol{p}}) \tilde{S}'_{\boldsymbol{p}})}{\alpha_0 + 2\sum_{k=1}^{K} \alpha_k (\cos(\omega_k f_{\boldsymbol{p}}) \tilde{C}_{\boldsymbol{p}} + \sin(\omega_k f_{\boldsymbol{p}}) \tilde{S}_{\boldsymbol{p}})} \tag{6}$$

where $\tilde{C}_{\boldsymbol{p}}$, $\tilde{S}_{\boldsymbol{p}}$, $\tilde{C}'_{\boldsymbol{p}}$, $\tilde{S}'_{\boldsymbol{p}}$ represent the convolution for the intermediate image and $\tilde{f}_{\boldsymbol{p}}$ is the output of convolution in the DC component.

$$\tilde{C}_{\boldsymbol{p}} = \sum_{\boldsymbol{q}\in\mathcal{S}} w_s(\boldsymbol{p},\boldsymbol{q}) \cos(w_k f_{\boldsymbol{q}}), \qquad \tilde{S}_{\boldsymbol{p}} = \sum_{\boldsymbol{q}\in\mathcal{S}} w_s(\boldsymbol{p},\boldsymbol{q}) \sin(w_k f_{\boldsymbol{q}})$$

$$\tilde{C'_{\boldsymbol{p}}} = \sum_{\boldsymbol{q}\in\mathcal{S}} w_s(\boldsymbol{p},\boldsymbol{q})\cos(w_k f_{\boldsymbol{q}}) f_{\boldsymbol{q}}, \qquad \tilde{S'_{\boldsymbol{p}}} = \sum_{\boldsymbol{q}\in\mathcal{S}} w_s(\boldsymbol{p},\boldsymbol{q})\sin(w_k f_{\boldsymbol{q}}) f_{\boldsymbol{q}}$$

Also, $w_k = \frac{2\pi}{T}k$, and the period T is determined to minimize the following analytically determined error.

$$E(K,T) = \mathrm{erfc}\left(\frac{\pi\sigma}{\mathrm{T}}(2\mathrm{K}+1)\right) + \mathrm{erfc}\left(\frac{\mathrm{T}-\mathcal{R}}{\sigma}\right). \tag{7}$$

In this method, convolution is performed for four intermediate images and one DC component per order, so a total of $4K+1$ convolutions are required for the approximate order K. Note that the implementation reduces the multiplication by dividing each coefficient in the denominator numerator by α_0 and doubling it to $\alpha'_k = 2\alpha_k/\alpha_0$.

$$\hat{f}_{\boldsymbol{p}} \approx \frac{\tilde{f}_{\boldsymbol{p}} + \sum_{k=1}^{K-1}\alpha'_k(\cos(\omega_k f_{\boldsymbol{p}})\tilde{C'_{\boldsymbol{p}}} + \sin(\omega_k f_{\boldsymbol{p}})\tilde{S'_{\boldsymbol{p}}})}{1 + \sum_{k=1}^{K-1}\alpha'_k(\cos(\omega_k f_{\boldsymbol{p}})\tilde{C'_{\boldsymbol{p}}} + \sin(\omega_k f_{\boldsymbol{p}})\tilde{S'_{\boldsymbol{p}}})} \tag{8}$$

2.3 Clustering-based Constant-time Bilateral Filter

Clustering-based constant-time bilateral filtering (CCBF)[17] used in this paper adopts Nyström approximated acceleration of eigenvalue decomposition (EVD).

First, we explain EVD for CCBF. Let $\boldsymbol{\mathcal{T}} = \{\boldsymbol{f_p} : \boldsymbol{p} \in \mathcal{S}\}$ be a list of color vector values in a signal $\boldsymbol{f} : \mathcal{S} \mapsto [0,R]^3$. Let $\boldsymbol{\mathcal{T}} = \{\boldsymbol{t_1}, \boldsymbol{t_2}, \ldots, \boldsymbol{t_m}\}$ be a ordering of the elements in $\mathcal{T}$, where m is the number of elements. Given an index $l \in [1,m]$, $\boldsymbol{t}_l = \boldsymbol{f_p}$ for some $\boldsymbol{p} \in \mathcal{S}$. An index map $\tau : \mathcal{S} \mapsto [1,m]$ can track the correspondence, where

$$\tau_{\boldsymbol{p}} = l \quad \text{if} \quad \boldsymbol{t}_l = \boldsymbol{f_p}. \tag{9}$$

Next, the kernel matrix $\boldsymbol{W} \in \mathbb{R}^{m\times m}$ is defined. The elements of $\boldsymbol{W}$ is given by

$$\boldsymbol{W}(i,j) = w_r(\boldsymbol{t}_i,\boldsymbol{t}_j) = \exp\left(-\frac{\|\boldsymbol{t}_i - \boldsymbol{t}_j\|_2^2}{2\sigma_r^2}\right), \tag{10}$$

where $i,j \in [1,m]$. Substituting (10) for (25) gives

$$\bar{\boldsymbol{f}}_{\boldsymbol{p}} = \frac{\sum_{\boldsymbol{q}\in N_{\boldsymbol{p}}} w_s(\boldsymbol{p},\boldsymbol{q})\boldsymbol{W}(\tau_{\boldsymbol{p}},\tau_{\boldsymbol{q}})\boldsymbol{f_q}}{\sum_{\boldsymbol{q}\in N_{\boldsymbol{p}}} w_s(\boldsymbol{p},\boldsymbol{q})\boldsymbol{W}(\tau_{\boldsymbol{p}},\tau_{\boldsymbol{q}})}, \tag{11}$$

$\boldsymbol{W}$ is a symmetric matrix; thus, EVD of $\boldsymbol{W}$ is as follows;

$$\boldsymbol{W} = \sum_{k=1}^{m} \lambda_k \boldsymbol{u}_k \boldsymbol{u}_k^T, \tag{12}$$

where $\lambda_k(\lambda_1 \geq \lambda_2 \geq, ..., \geq \lambda_m \in \mathbb{R})$ are the eigenvalues, and $\boldsymbol{u}_k$ is the corresponding eigenvectors. Substituting (12) to (11) gives

$$\bar{\boldsymbol{f}}_{\boldsymbol{p}} = \frac{\sum_{\boldsymbol{q}\in N_{\boldsymbol{p}}} w_s(\boldsymbol{p},\boldsymbol{q}) \sum_{k=1}^{m}\lambda_k \boldsymbol{u}_k(\tau_{\boldsymbol{p}})\boldsymbol{u}_k(\tau_{\boldsymbol{q}})\boldsymbol{f_q}}{\sum_{\boldsymbol{q}\in N_{\boldsymbol{p}}} w_s(\boldsymbol{p},\boldsymbol{q}) \sum_{k=1}^{m}\lambda_k \boldsymbol{u}_k(\tau_{\boldsymbol{p}})\boldsymbol{u}_k(\tau_{\boldsymbol{q}})}, \tag{13}$$

On switching the sums, this becomes

$$\bar{\boldsymbol{f}}_{\boldsymbol{p}} = \frac{\sum_{k=1}^{m}\lambda_k \boldsymbol{u}_k(\tau_{\boldsymbol{p}}) \sum_{\boldsymbol{q}\in N_{\boldsymbol{p}}} w_s(\boldsymbol{p},\boldsymbol{q})\{\boldsymbol{u}_k(\tau_{\boldsymbol{q}})\boldsymbol{f_q}\}}{\sum_{k=1}^{m}\lambda_k \boldsymbol{u}_k(\tau_{\boldsymbol{p}}) \sum_{\boldsymbol{q}\in N_{\boldsymbol{p}}} w_s(\boldsymbol{p},\boldsymbol{q})\{\boldsymbol{u}_k(\tau_{\boldsymbol{q}})\}}. \tag{14}$$

Next, we approximate the filter. Let $\hat{\boldsymbol{W}} \in \mathbb{R}^{K\times K}$ be the matrix of a low-rank approximation of $\boldsymbol{W}$ using the top $K(K \ll m)$ eigenvalues and eigenvectors. Using $\hat{\boldsymbol{W}}$ instead of $\boldsymbol{W}$ in (14), BF can be approximated as

$$\bar{\boldsymbol{f}}_{\boldsymbol{p}} \approx \frac{\sum_{k=1}^{K}\lambda_k \boldsymbol{u}_k(\tau_{\boldsymbol{p}}) \sum_{\boldsymbol{q}\in N_{\boldsymbol{p}}} w_s(\boldsymbol{p},\boldsymbol{q})\{\boldsymbol{u}_k(\tau_{\boldsymbol{q}})\boldsymbol{f_q}\}}{\sum_{k=1}^{K}\lambda_k \boldsymbol{u}_k(\tau_{\boldsymbol{p}}) \sum_{\boldsymbol{q}\in N_{\boldsymbol{p}}} w_s(\boldsymbol{p},\boldsymbol{q})\{\boldsymbol{u}_k(\tau_{\boldsymbol{q}})\}}. \tag{15}$$

(a) input (b) pca 1ch

Figure 1. Image dimensional compression by PCA

For the Nystöm approximation of EVD of $\boldsymbol{W}$, we first construct a small kernel $\boldsymbol{A} \in \mathbb{R}^{K\times K}$ and then extrapolate the eigenvectors of $\boldsymbol{A}$ to approximate those of $\boldsymbol{W}$. $\boldsymbol{A}$ is defined using dominant color vectors $\boldsymbol{\mu}_k$ introduced by clustering as

$$\boldsymbol{A}(i,j) = w_r(\boldsymbol{\mu}_i, \boldsymbol{\mu}_j) \quad i,j \in [1,K]. \tag{16}$$

The size of $\boldsymbol{A}$ is smaller than that of $\boldsymbol{W}$; thus, we can easily compute EVD:

$$\boldsymbol{A} = \sum_{k=1}^{K} \lambda_k \boldsymbol{v}_k \boldsymbol{v}_k^T, \tag{17}$$

where $\lambda_k \in \mathbb{R}$ and $\boldsymbol{v}_k \in \mathbb{R}^K$. A matrix $\boldsymbol{B} \in \mathbb{R}^{K\times m}$ for extrapolation is:

$$\boldsymbol{B}(i,j) = w_r(\boldsymbol{\mu}_i, \boldsymbol{t}_j) \quad i \in [1,K], j \in [1,m]. \tag{18}$$

This matrix is used to extrapolate $\boldsymbol{u}_k$ as follows;

$$\boldsymbol{u}_k = \frac{1}{\lambda_k} \boldsymbol{B}^T \boldsymbol{v}_k. \tag{19}$$

These calculations eliminate the computation of EVD of the large matrix $\boldsymbol{W}$. The computing order of $\boldsymbol{u}_k$ in (19) is $O(K)$ and the computation is required for each convolution; thus, the order of generating decomposed images for filtering is $O(K^2)$. The actual cost is about $2K$ multiply/addition for the construction of the filtering target images of $\boldsymbol{u}_k(\tau_{\boldsymbol{q}})\boldsymbol{f}_{\boldsymbol{q}}$ or $\boldsymbol{u}_k(\tau_{\boldsymbol{q}})$, respectively. The cost is smaller than the convolution cost, in which the order is $O(K)$; thus, we can ignore the order in the small K case. For the large K case, the footprint cannot be ignored.

3. PROPOSED METHOD

3.1 PCA

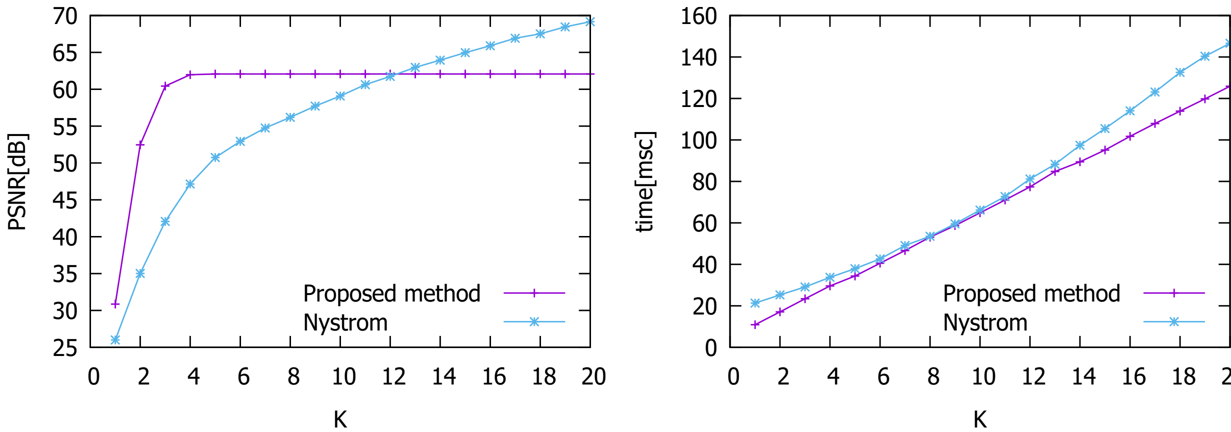
A p-th row of the signals in a matrix representation $\boldsymbol{X} \in \boldsymbol{R}^{|\Omega|\times N}$ with an input data $\boldsymbol{G} \in \boldsymbol{R}^{|\Omega|}$ is defined as

$$\boldsymbol{X}_p = [\boldsymbol{R}, \boldsymbol{G}, \boldsymbol{B}], \tag{20}$$

where $p \in \Omega$ is an index of data, $\Omega \subset \mathbb{N}$ is a set of all data indices.

We compute the covariance matrix of $\boldsymbol{X}$ for PCA. The matrix can describe as

$$\boldsymbol{C} = (\boldsymbol{X} - \bar{\boldsymbol{X}})^\top (\boldsymbol{X} - \bar{\boldsymbol{X}}), \tag{21}$$



(a) PSNR to approximation order.

(b) time to approximation order.

Figure 2. Approximation results for each order.

$$\bar{\boldsymbol{X}} = \mathbb{1}\left(\frac{1}{|\Omega|}\sum_{p\in\Omega}\boldsymbol{X}_p\right), \tag{22}$$

where $\mathbb{1} \in \boldsymbol{R}^{|\Omega|}$ is a vector that all elements are 1.

Third, PCA is based on an EVD of the covariance matrix

$$\boldsymbol{C} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^{-1}, \tag{23}$$

where $\boldsymbol{U}$ is a matrix that each column is eigenvector of the $\boldsymbol{C}$, $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1 \cdots, \lambda_N)$, λ_n is a n-th largest eigenvalue of the $\boldsymbol{C}$.

Finally, the dimensional reduction of the high-dimensional signals $\boldsymbol{X}' \in \boldsymbol{R}^{|\Omega|\times k}$ can described as

$$\boldsymbol{X}'_p = \boldsymbol{X}_p\boldsymbol{U}_k, \tag{24}$$

where each column of the $\boldsymbol{U}_k$ is an eigenvector that corresponds to the upper k eigenvalues.

3.2 Joint Bilateral Filter

The proposed method represents BF by joint bilateral filtering (JBF). JBF uses another guide image instead of the input image to determine the weights of the range kernel. Using the input image converted to a single channel by PCA as the guide image, the curse of dimensionality of CBF can be avoided. JBF and its constant-time one are defined by:

$$\hat{f}_{\boldsymbol{p}} = \frac{\sum_{\boldsymbol{q}\in\mathcal{S}} w_s(\boldsymbol{p},\boldsymbol{q})w_r(g_{\boldsymbol{p}},g_{\boldsymbol{q}})f_{\boldsymbol{q}}}{\sum_{\boldsymbol{q}\in\mathcal{S}} w_s(\boldsymbol{p},\boldsymbol{q})w_r(g_{\boldsymbol{p}},g_{\boldsymbol{q}})}, \tag{25}$$

$$\hat{f}_{\boldsymbol{p}} \approx \frac{\sum_{k=0}^{K-1}\phi_k(g_{\boldsymbol{p}})\sum_{\boldsymbol{q}\in\mathcal{S}} w_s(\boldsymbol{p},\boldsymbol{q})\{\psi_k(g_{\boldsymbol{q}})f_{\boldsymbol{q}}\}}{\sum_{k=0}^{K-1}\phi_k(g_{\boldsymbol{p}})\sum_{\boldsymbol{q}\in\mathcal{S}} w_s(\boldsymbol{p},\boldsymbol{q})\{\psi_k(g_{\boldsymbol{q}})\}}. \tag{26}$$

$g_{\boldsymbol{p}}$ represents the luminance of the 1-channel guide image. The number of Gaussian filters can be significantly reduced.

4. EXPERIMENTAL RESULTS AND DISCUSSION

We used the Nyström method as a comparison for the proposed method. The $\sigma_s = 2$, $\sigma_r = 70$, and the approximation order $K = 1$ to 20, respectively. PSNR was used for image accuracy, and a compressive bilateral filter was used as a constant-time BF method. The CPU was an Intel Core i3-8100, 4 cores/4 threads at 3.6GHz. For the approximation accuracy evaluation, the output image from the naïve implementation was used as the ideal image, and the approximation error was evaluated in terms of PSNR between the filtered image and the ideal image using each method. 24 Kodak image sets (756×512) were filtered as input images, and the average accuracy was estimated. The results show that the proposed method can output sufficient approximation accuracy at low order, and the filtering time is faster than the Nyström method. The approximation accuracy of the proposed method reaches a ceiling of 60dB because the guide image has been set to 1-channel.

REFERENCES

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

[2] Dai, S., Han, M., Wu, Y., and Gong, Y., "Bilateral back-projection for single image super resolution.," in *Proc. IEEE International Conference on Multimedia and Expo (ICME)*, 1039–1042 (2007).

[3] Cho, S. and Lee, S., "Fast motion deblurring," *ACM Transactions on graphics* **28**(5), 145 (2009).

[4] Farbman, Z., Fattal, R., Lischinski, D., and Szeliski, R., "Edge-preserving decompositions for multi-scale tone and detail manipulation," *ACM Transactions on Graphics* **27**(3), 67 (2008).

[5] Durand, F. and Dorsey, J., "Fast bilateral filtering for the display of high-dynamic-range images," *ACM Transactions on Graphics* **21**(3), 257–266 (2002).

[6] Fukushima, N., Sugimoto, K., and Kamata, S., "Guided image filtering with arbitrary window function," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, (2018).

[7] Fukushima, N., Fujita, S., and Ishibashi, Y., "Switching dual kernels for separable edge-preserving filtering," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, (2015).

[8] Matsuo, T., Fukushima, N., and Ishibashi, Y., "Weighted joint bilateral filter with slope depth compensation filter for depth map refinement," in *Proc. The International Conference on Computer Vision Theory and Applications (VISAPP)*, 300–309 (2013).

[9] Matsuo, T., Fujita, S., Fukushima, N., and Ishibashi, Y., "Efficient edge-awareness propagation via single-map filtering for edge-preserving stereo matching," in *Proc. Three-Dimensional Image Processing, Measurement (3DIPM), and Applications*, (2015).

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

[11] Miyamura, T., Fukushima, N., Waqas, M., Sugimoto, K., and Kamata, S., "Image tiling for clustering to improve stability of constant-time color bilateral filtering," in *Proceedings of IEEE International Conference on Image Processing (ICIP)*, (2020).

[12] Oishi, S. and Fukushima, N., "Clustering-based acceleration for high-dimensional gaussian filtering," in *Proc. Signal Processing and Multimedia Applications (SIGMAP)*, (2021).

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

[14] Sugimoto, K. and Kamata, S., "Compressive bilateral filtering," *IEEE Transactions on Image Processing* **24**(11), 3357–3369 (2015).

[15] Sumiya, Y., Fukushima, N., Sugimoto, K., and Kamata, S., "Extending compressive bilateral filtering for arbitrary range kernel," in *Proceedings of IEEE International Conference on Image Processing (ICIP)*, (2020).

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

[17] Nair, P. and Chaudhury, K. N., "Fast high-dimensional kernel filtering," *IEEE Signal Processing Letters* **26**, 377–381 (2019).