Clustering-based Acceleration for High-dimensional Gaussian Filtering

Sou Oishi and Norishige Fukushima[a]

Nagoya Institute of Technology, Japan
https://fukushima.web.nitech.ac.jp/en/

Abstract: Edge-preserving filtering is an essential tool for image processing applications and has various types of filtering. For real-time applications, acceleration of its speed is also essential. To accelerate various types of edge-preserving filtering, we represent various edge-preserving filtering by high-dimensional Gaussian filtering. Then, we accelerate the high-dimensional Gaussian filtering by clustering-based constant algorithm, which has $O(K)$ order, where K is the number of clusters. The clustering-based method was developed for color bilateral filtering; however, this paper used it for high-dimensional bilateral filtering. Also, cooperating with tiling, k-means++, and principal component analysis, we can further improve the filter's performance. Experimental results show that our method can approximate various edge-preserving filtering by approximated clustering-based high-dimensional Gaussian filtering.

1 INTRODUCTION

Edge-preserving smoothing is an essential tool for image processing and computer vision. The filtering is used in various image processing applications, such as denoising (Zhang and Gunturk, 2008), deblurring (Dai et al., 2007), detail enhancement (Farbman et al., 2008), high-dynamic-range imaging (Durand and Dorsey, 2002), haze removing (Fukushima et al., 2018), stereo matching (Matsuo et al., 2015), and optical flow (Fujita et al., 2015).

A representative edge-preserving filter is bilateral filtering (BF) (Tomasi and Manduchi, 1998). The BF's kernel composites two kernels: a spatial kernel for pixel positions and a range kernel for pixel values. Joint/cross bilateral filtering (JBF) (Eisemann and Durand, 2004; Petschnigg et al., 2004) is a natural extension, which uses an additional image for range kernel computing. Joint bilateral upsampling (Kopf et al., 2007) utilizes the JBF's kernel for upsampling. Dual bilateral filtering (DBF) (Bennett et al., 2007) also extends the JBF, which uses three kernels for space, an RGB image, and an IR image. Joint/cross-trilateral filtering (JTF) has a similar way for depth map filtering, which uses a depth map instead of using an IR image for filtering depth map (Mueller et al., 2010; Matsuo et al., 2013). Vector bilateral filtering (VBF) is an extension for hyperspectral im-

ages (Peng and Rao, 2009). Non-local means filtering (NLM) (Buades et al., 2005) is an extension of BF, which uses patch vectors instead of color intensities.

In this paper, we summarize these variants of BF as high-dimensional Gaussian filtering (HDGF). The HDGF is a convolution filter with a distance not only in space but also in other dimensions. As typical examples, the grayscale BF is a 3-dimensional HDGF for space and intensity value, and the color BF is a 5-dimensional HDGF, which has 3-dimensions in intensity values. The DBF and JTF, including an IR image or a depth map, are also $3+1$-HDGF, respectively. n channels hyperspectral images are an $n+2$-dimensional HDGF. The NLM is a m^2+2 dimensional HDGF in the case of creating an $m \times m$ patch.

The HDGF has several acceleration algorithms. The accelerated grayscale BF, i.e., 3-dimensional HDGF, is proposed early, and it becomes fundamental. The state-of-the-art approaches have $O(K)$ order, where K is the approximating order, and it is smaller than the filtering radius D; $K << D$. We can extend the specialized approaches for grayscale BF to HDGF, e.g., color BF. However, as the number of dimensions increases, the efficiency decreases due to the curse of dimensionality. The computational order of n channels BF is $O(K^n)$. Spatial and range downsampling is a solution; however, these methods do not fundamentally solve the issue.

The clustering-based approaches resolve the curse

---

[a] https://orcid.org/0000-0001-8320-6407

65

of dimensionality. The methods can reduce the computational order $O(K)$, where K is the number of clusters. However, the approximation performance depends on clustering methods. Also, clustering itself is an overhead to the processing time. The sophisticated clustering (Miyamura et al., 2020), which uses a tiling strategy and advanced clustering, improves the performance of 5-dimensional HDGF of color BF, but the evaluation as the HDGF is not sufficient. The filter was only evaluated as the color BF.

We often use dimensionality reduction to compress signals for accelerating processing, but the verification of the clustering-based HDGF is not enough. This paper evaluates the performance of the constant-time algorithm with clustering for various HDGFs. Also, we improve the clustering performance by tiling, K-means++ (Arthur and Vassilvitskii, 2007), and dimensionality reduction. The contributions of this paper are as follows:

- We comprehensively deal with various local smoothing as the HDGF.

- We accelerate various HDGFs by clustering-based constant-time filtering leveraged by tiling and sophisticated clustering techniques.

- We evaluate the dimensionality reduction for various types of the clustering-based HDGF.

2 RELATED WORKS

1) Grayscale BF (3D-HDGF): (Durand and Dorsey, 2002) proposed the early work of accelerating grayscale BF. This approach decomposes BF into multiple Gaussian filtering, and fast Fourier transform (FFT) accelerates it. (Paris and Durand, 2006; Chen et al., 2007) and (Chen et al., 2007) extended it by representing BF as the HDGF cooperating with downsampling. (Weiss, 2006) approximated BF by using histogram, and (Porikli, 2008) reduces the other to $O(K)$. (Yang et al., 2009) extended Durand's approach by constant Gaussian filtering (Deriche, 1992), whose order is also $O(K)$. (Chaudhury et al., 2011; Chaudhury, 2013; Fukushima et al., 2017) proposed a more efficient $O(K)$ method by raised cosine approximation. Further, (Sugimoto and Kamata, 2015; Sumiya et al., 2020) sophisticated the approach by using Fourier series expansion. Also, eigenvalue decomposition improved the performance of grayscale BF (Sugimoto et al., 2016a; Papari et al., 2017) The BF consists of a numerator and denominator, which requires convolution, individually. (Chaudhury and Dabhade, 2016) shared the numerator and denominator's convolution. The idea was extended

for Fourier series expansion (Deng, 2017) and singular value decomposition (Sugimoto et al., 2019).

2) Color BF (5D-HDGF): Changing the grayscale BF to the color BF is not simple. We carefully handle the distance function of color channels, i.e., 1-dimension to 3-dimension. (Paris and Durand, 2009) proposed the natural extension from the grayscale BF. This approach was, however, $O(K^3)$; thus, it utilized downsampling. (Yang et al., 2015) also extended grayscale one. Recent approaches reduced the number of the range kernel convolutions (Karam et al., 2015; Ghosh and Chaudhury, 2016; Tu et al., 2016).

3) nD-HDGF: (Adams et al., 2009; Adams et al., 2010) proposed a suitable data structure for the downsampling-based HDGF, which are Gaussian KD-tree and permutohedral lattice. The order is also $O(k^n)$, but we can more massively downsample data than (Paris and Durand, 2009; Yang et al., 2015). Also, (Gastal and Oliveira, 2012) approximated the HDGF by recursive geodesic filtering, which has Gaussian-like output, but it has high speed.

Clustering-based constant-time BFs (Mozerov and van de Weijer, 2015; Sugimoto et al., 2016b; Nair and Chaudhury, 2019; Miyamura et al., 2020) are aimed for color BF, but we can use them for the HDGF. The order is $O(k)$, but it requires an additional footprint of clustering. (Miyamura et al., 2020) improved the clustering speed and accuracy.

4) Non-local Means: Non-local means filtering(NLM) (Buades et al., 2005; Buades et al., 2005), proposed simultaneously with unsupervised information theoretic adaptive (UINTA) filter (Awate and Whitaker, 2005), is regarded as the HDGF. The NLM requires an entire image for a pixel convolution. Let patch size P, windows size W, image size N. Naïve approach, W=N, $O(PN^2)$. Restricting the search to patches in a local neighborhood (Buades et al., 2005), $O(PWN)$, $W << N$ Random sampling also reduces the cost (Awate and Whitaker, 2005) using pre-selected patches (Buades et al., 2005), $O(PW'N)$, $W' << W$. Searching the patches by keying means and variances, mean, and gradients (Mahmoudi and Sapiro, 2005; Coupé et al., 2006; Kervrann et al., 2007; Gilboa and Osher, 2007) improve the performance. (Brox et al., 2008) used recursive a k-means build-up cluster tree. This approach used clustering, but the order is $O(PK \log(k)N)$. Instead, the clustering-based HDGF is $O(PKN)$. Computing integral images of certain error terms (Wang et al., 2006; Darbon et al., 2008) had $O(PN + WN)$.

Principal component analysis (PCA) was utilized for dimensionality reduction for the NLM's patches for saving distance computation (Tasdizen, 2008; Tas-

dizen, 2009). We can extend the idea for any HDGF.

3 BILATERAL FILTERING AND ITS ACCELERATION

3.1 Definition

We firstly define the BF. Let the D-dimensional R-tone image be $f : S \mapsto \mathcal{R}$, where $S \subset \mathbb{R}^D$ is the spatial domain, $\mathcal{R} \subset [0, R]^c$ is the range domain, and c is the range dimension (generally, $D = 2$, $R = 256$, and $c = 3$), respectively. Denote $p \in S$ as a pixel position, $f_p \in \mathcal{R}$ as its intensity vector, and $N_p \subset S$ as its neighboring pixels. The BF is defined as follows;

$$\bar{f}_p = \frac{\sum_{q \in N_p} w_s(p, q) w_r(f_p, f_q) f_p}{\sum_{q \in N_p} w_s(p, q) w_r(f_p, f_q)}, \quad (1)$$

where $w_s : S \times S \mapsto \mathbb{R}$ is a spatial kernel and $w_r : \mathcal{R} \times \mathcal{R} \mapsto \mathbb{R}$ is a range kernel. The kernels are defined based on the Gaussian distribution;

$$w_s(p, q) = \exp\left(-\frac{\|q - p\|_2^2}{2\sigma_s^2}\right), \quad (2)$$

$$w_r(f_p, f_q) = \exp\left(-\frac{\|f_q - f_p\|_2^2}{2\sigma_r^2}\right), \quad (3)$$

where $\sigma_s \in \mathbb{R}_+$ is a spatial scale and $\sigma_r \in \mathbb{R}_+$ is a range scale.

3.2 Clustering-based Constant-time Bilateral Filter

We introduce a clustering-based constant-time bilateral filtering (CBF). The CBF used in this paper adopts Nyström approximated acceleration of eigenvalue decomposition (EVD) (Nair and Chaudhury, 2019). First, we explain EVD for CBF.

Let $\mathcal{T} = \{f_p : p\}$ be a list and let $\mathcal{T} = \{t_1, t_2, \dots, t_m\}$ be a ordering of the elements in $\mathcal{T}$, where m is the number of elements. This means that, given $l \in [1, m]$, $t_l = f_x$ for some $x \in S$. We track the correspondence by an index map: $S \mapsto [1, m]$, where

$$\tau_p = l \quad \text{if} \quad t_l = f_p. \quad (4)$$

We next define the kernel matrix $W \in \mathbb{R}^{m \times m}$ given by

$$W(i, j) = w_r(t_i, t_j). \quad (5)$$

Substituting (5) for (1) gives

$$\bar{f}_p = \frac{\sum_{q \in N_p} w_s(p, q) W(\tau_p, \tau_q) f_p}{\sum_{q \in N_p} w_s(p, q) W(\tau_p, \tau_q)}, \quad (6)$$

Since it is clear from (5) that W is a symmetric matrix, EVD of W is as follows;

$$W = \sum_{k=1}^{m} \lambda_k u_k u_k^T, \quad (7)$$

where $\lambda_k(\lambda_1 \geq \lambda_2 \geq, \dots, \geq \lambda_m \in \mathbb{R})$ are eigenvalues and u_k is the corresponding eigenvectors. Substituting (7) to (6) gives

$$\bar{f}_p = \frac{\sum_{q \in N_p} w_s(p, q) \sum_{k=1}^{m} \lambda_k u_k(\tau_p) u_k(\tau_q) f_p}{\sum_{q \in N_p} w_s(p, q) \sum_{k=1}^{m} \lambda_k u_k(\tau_p) u_k(\tau_q)}, \quad (8)$$

On switching the sums, this becomes

$$\bar{f}_p = \frac{\sum_{k=1}^{m} \lambda_k u_k(\tau_p) \sum_{q \in N_p} w_s(p, q) \{u_k(\tau_q) f_q\}}{\sum_{k=1}^{m} \lambda_k u_k(\tau_p) \sum_{q \in N_p} w_s(p, q) \{u_k(\tau_q)\}}. \quad (9)$$

Let $\hat{W} \in \mathbb{R}^{m_0 \times m_0}$ be the matrix of a low-rank approximation of W using the top $m_0 (m_0 \ll m)$ eigenvalues and eigenvectors. Using $\hat{W}$ instead of W in (9), the BF can be approximated as

$$\bar{f}_p \approx \frac{\sum_{k=1}^{m_0} \lambda_k u_k(\tau_p) \sum_{q \in N_p} w_s(p, q) \{u_k(\tau_q) f_q\}}{\sum_{k=1}^{m_0} \lambda_k u_k(\tau_p) \sum_{q \in N_p} w_s(p, q) \{u_k(\tau_q)\}}. \quad (10)$$

In the color case, the number of convolutions is $4m_0$, where K in the denominator and $3K$ in the numerator.

In the grayscale case, the size of the matrix is 256×256. However, in the color case, the size of the matrix is 256^3. Therefore, direct EVD of W is difficult. For Nystöm approximation of EVD of W, we first construct a small kernel A and then extrapolate the eigenvectors of A to approximate those of W. $A \in \mathbb{R}^{m_0 \times m_0}$ is defined using dominant color vectors μ_k determined by clustering as

$$A(i, j) = w_r(\mu_i, \mu_j) \quad i, j \in [1, m_0]. \quad (11)$$

The size of A is much smaller than that of W. Thus, we can easily compute EVD:

$$A = \sum_{k=1}^{m_0} \lambda_k v_k v_k^T, \quad (12)$$

where $\lambda_k \in \mathbb{R}$ and $v_k \in \mathbb{R}^{m_0}$. $B \in \mathbb{R}^{m_0 \times m}$ is :

$$B(i, j) = w_r(\mu_i, t_j) \quad i \in [1, m_0], j \in [1, m]. \quad (13)$$

This matrix is used to extrapolate u_k as follows;

$$u_k = \frac{1}{\lambda_k} B^T v_k. \quad (14)$$

These calculations eliminate the computation of EVD of the large matrix W.

4 PROPOSED METHOD

This section summarizes various neighborhood filtering as the HDGF. Also, we accelerate it by clustering-based constant time algorithm. Then we further accelerate it by PCA, tiling, and K-means++.

4.1 High-dimensional Gaussian Filter

The HDGF increases the distance measure's dimensionality in the range weights of the BF. Let the n-dimensional input signal be $I \in \mathbb{R}^n$ and the m-dimensional guide signal be $G \in \mathbb{R}^m$. The HDGF's output of $\bar{I}_p$ at a pixel p is;

$$\bar{I}_p = \frac{\sum_{q \in N_p} w_s(p,q) w_h(G_p, G_q) I_p}{\sum_{q \in N_p} w_s(p,q) w_h(G_p, G_q)}, \qquad (15)$$

where the kernel w_h is defined based on the Gaussian distribution as follows:

$$w_h(G_p, G_q) = \exp\left(-\frac{\|G_q - G_p\|_2^2}{2\sigma_r^2}\right). \qquad (16)$$

The form is similar to (1) that is changing the range kernel's argument. When we use n-dimensional vectors for f for (1), the equation is the HDGF; thus, we can directly use clustering-based CBF as the HDGF.

This form is called joint bilateral filtering (Eisemann and Durand, 2004; Petschnigg et al., 2004), when $m = 1, 3$. When $G = I$, this is the grayscale BF and the color BF. This direct representation of the grayscale BF is the $2+1$-dimensional HDGF, and the color BF is the $2+3$-dimensional HDGF. When we use a n-channels hyperspectral image, it becomes the $2+n$-dimensional HDGF. Therefore, we can represent various neighborhood filtering by changing the guide signal of G:

$$
\begin{aligned}
G_{GBF} &:= \{I_g\}, \\
G_{GJBF} &:= \{G_g\}, \\
G_{CBF} &:= \{I_R, I_G, I_B\}, \\
G_{CJBF} &:= \{G_R, G_G, G_B\}, \\
G_{HBF} &:= \{G_0, G_1, \ldots, G_n\}, \qquad (17)
\end{aligned}
$$

where GBF is the grayscale BF, GJBF is the grayscale JBF, CBF is the color BF, CJBF is the color JBF, HBF is the hyperspectral BF. I_g and G_g are an input grayscale image and an additional guidance image, respectively. $I_{\{R,G,B\}}$ and $G_{\{R,G,B\}}$ are a vector for each RGB components. G_n is a vector of each component of a n-channels hyperspectral image.

The NLM is also the HDGF. For example, for the 5×5 patch size, NLM is a $2+25$ dimensional GF. The NLM is defined as follows

$$\bar{f}_p = \frac{\sum_{q \in N_p} w_s(p,q) w_r(\sum_{r \in R} \|f_{p+r} - f_{q+r}\|_2) f_p}{\sum_{q \in N_p} w_s(p,q) w_r(\sum_{r \in R} \|f_{p+r} - f_{q+r}\|_2)}, \qquad (18)$$

where $R \subset S$ is set of patch neighborhood pixels and r is its element. The form can be represented as neighborhood pixels as high-dimensional vectors. For example, 3×3 patch's NLM is represented by the following HDGF:

$$
\begin{aligned}
G_{NLM} :=\ & \{I_{g(-1,-1)}, \quad I_{g(-1,0)}, \quad I_{g(-1,1)}, \\
& I_{g(0,-1)}, \quad I_{g(0,0)}, \quad I_{g(0,1)}, \\
& I_{g(1,-1)}, \quad I_{g(1,0)}, \quad I_{g(1,1)}\}, \\
G_{CNLM} :=\ & \{I_{B(-1,-1)}, \quad I_{B(-1,0)}, \quad I_{B(-1,1)}, \\
& I_{B(0,-1)}, \quad I_{B(0,0)}, \quad I_{B(0,1)}, \\
& I_{B(1,-1)}, \quad I_{B(1,0)}, \quad I_{B(1,1)}, \\
& I_{G(-1,-1)}, \quad I_{G(-1,0)}, \quad I_{G(-1,1)}, \\
& I_{G(0,-1)}, \quad I_{G(0,0)}, \quad I_{G(0,1)}, \\
& I_{G(1,-1)}, \quad I_{G(1,0)}, \quad I_{G(1,1)}, \\
& I_{R-1,-1)}, \quad I_{R(-1,0)}, \quad I_{R(-1,1)}, \\
& I_{R(0,-1)}, \quad I_{R(0,0)}, \quad I_{R(0,1)}, \\
& I_{R(1,-1)}, \quad I_{R(1,0)}, \quad I_{R(1,1)}\}, \qquad (19)
\end{aligned}
$$

where the G_{NLM} is grayscale NLM and the G_{NLM} is color NLM. The form can also handle joint filtering, i.e., we can replace I as G. Also, (x,y) represents the shift operator, e.g., $I_{(-1,-1)}$ is a vector whose position is shifted by $(-1,-1)$.

We handle dual bilateral filtering (DBF) or joint trilateral filtering (JTF) for the other extension. These filters have an additional kernel for BF expression of (1) and require both input and guidance signals of I and G. Also, the input signal and guidance signal have different characteristics. For example, flash/no-flash image pair, RGB/IR image pair, RGB/depth image pair, RGB/optical flow pair, RGB/alpha matting mask, RGB/normal map image pair. The form is represented by:

$$\bar{I}_p = \frac{\sum_{q \in N_p} w_s(p,q) w_r(G_p, G_q) w_a(I_p, I_q) I_p}{\sum_{q \in N_p} w_s(p,q) w_h(G_p, G_q) w_a(I_p, I_q)}, \qquad (20)$$

where w_a is an additional kernel, and is a usually Gaussian kernel:

$$w_a(I_p, I_q) = \exp\left(-\frac{\|I_q - I_p\|_2^2}{2\sigma_a^2}\right). \qquad (21)$$

For handling different characteristics of the input and guidance signals, the parameters for each kernel is different, i.e., $\sigma_r \neq \sigma_a$.

We can merge the two Gaussian kernels of w_r and w_a by the exponential function's rules:

$$
\begin{aligned}
& w_h(I_p, I_q, G_p, G_q) \\
& = \exp\left(\frac{\|G_q - G_p\|_2^2 + t^2\|I_q - I_p\|_2^2}{-2\sigma_r^2}\right), \\
& = \exp\left(\frac{\|G_q - G_p\|_2^2 + \|t I_q - t I_p\|_2^2}{-2\sigma_r^2}\right), \qquad (22)
\end{aligned}
$$

where $\sigma_r = t\sigma_a$. We can control each parameter by splitting the smoothing parameters between the original image and the additional image by multiplying the pixel value by t. The input signal $I : S \mapsto [0,R]$ is transformed into $I : S \mapsto [0,tR]$. The original paper of DBF handles an RGB image and an IR image to filter the RGB image, and also the CTF paper handles an RGB image and a depth map to filter the depth map. For representing the DBF and CTF as the HDGF, the guidance signals are represented as:

$$G_{DBF} := \{tI_R, tI_G, tI_B, I_{IR}\},$$
$$G_{CTF} := \{I_R, I_G, I_B, tD\}. \quad (23)$$

The DBF and CTF are the $2 + 3 + 1$-dimensional HDGF, including 2D-spatial dimension, RGB range dimensions, and a IR image or depth map dimension. The weight control idea can also be applied for the NLM and the color BF, e.g., using Gaussian patch weighting or handling the important color channels, such as the green channel. For example, we can separately set parameter for each dimension $\sigma_r = t_1\sigma_1 = t_2\sigma_2 = \cdots = t_n\sigma_n$.

4.2 Dimensionality Reduction by PCA

The HDGF utilizes PCA for reducing dimensionality. We compute covariance matrix C for G:

$$C = (G - \bar{G})^T (G - \bar{G}) = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \dots & \sigma_{1n} \\ \sigma_{21} & \sigma_{22} & \dots & \sigma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{n1} & \sigma_{n2} & \dots & \sigma_{nn}, \end{pmatrix} \quad (24)$$

where $\bar{G}$ contains mean vectors of G. For simplicity, $G_{p+r_n} = x_{n,p}, \bar{G_{r_n}} = \bar{x}_n$.

$$\sigma_{mn} = \frac{1}{|\Omega|} \sum_{p \in \Omega} (x_{m,p} - \bar{x}_m)(x_{n,p} - \bar{x}_n) \quad (25)$$

PCA decomposes the matrix by EVD.

$$C = V\Lambda V^{-1}, \quad (26)$$

where, V is matrix, which contains eigenvectors. $\Lambda = diag(\lambda_1, \dots, \lambda_n)$ is a diagonal matrix, which has eigenvalue with decent order. PCA projection is represented as:

$$G'_p = V_k G_p \quad (27)$$

where V_k has k-components of eigenvectors, and G'_p is the projected guidance signal.

4.3 Tiling with Clustering and PCA

For acceleration, we perform image tiling for constant-time filtering (Miyamura et al., 2020). The

previous method, which is for color BF, accelerates clustering processing and accelerates convolution processing in (10). For the clustering method, we used K-means++ (Arthur and Vassilvitskii, 2007) for improving the accuracy.

The proposed method requires PCA for guidance signals; thus, we also perform PCA on tile images instead of entire images. Therefore, tiling accelerates the computational speed, including convolution, clustering, and PCA. Note that PCA for the tiled image does not include full image information, but the effect is not large, verified in experimental results.

5 EXPERIMENTAL RESULTS

We evaluated the approximation accuracy for the NLM, CTF, and DBF. The accuracy was quantified as peak signal-to-noise ratio (PSNR) between approximation and non-approximation results. The codes were implemented in C++, vectorized by AVX, and parallelized by OpenMP. We used Visual Studio 2019's compiler and Intel Core i7-6700K @ 4.00GHz. Also, we divided the image into 4×4 for tiling. In our experiments, we set the parameters to $K = 30$, $\sigma_r = 70$, and $\sigma'_r = 70$, respectively, as default parameters.

Firstly, we evaluated the NLM approximation accuracy. The input image was Lenna (Figure 1a), and the guidance image's channels changed from 4 to 10 by PCA from $3 \times 3 \times 3 = 27$ channels. Figure 1 shows the filtered images by $\sigma_r = 70$, $K = 30$ and 4 channels. Figure 2, 3 (resp. Figure 4) show the accuracy results (resp. the acceleration results) w.r.t. changing the number of clusters K or parameter σ_r with tiling

(a) input   (b) without tiling   (c) with tiling

Figure 1: Input/Output of NLM (Lenna: $\sigma_r = 70$, $K = 30$).
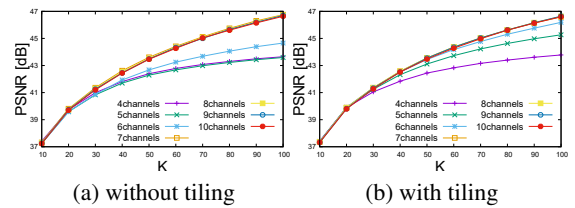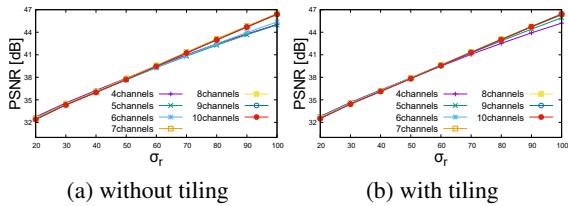

(a) without tiling   (b) with tiling

Figure 2: K w.r.t PSNR [dB] (NLM approx.: $\sigma_r = 70$). Over 7 channels case, Accuracy is saturated in without tiling. With tiling, 5 and 6 channels cases are improved.

(a) without tiling   (b) with tiling

Figure 3: σ_r w.r.t PSNR [dB] (NLM approx.: $K = 30$).

and without tiling cases. Other parameters are default parameters. We use averaged PSNR and the calculation time with 100 trials. From these results, PSNR is improved by increasing K, σ_r, and the dimensionality. Moreover, we can keep PSNR using tiling in the HDGF at the highest level, though PCA uses only local tile information. Also, in the other low-dimensional approximation, our method has a higher PSNR than the method without tiling.
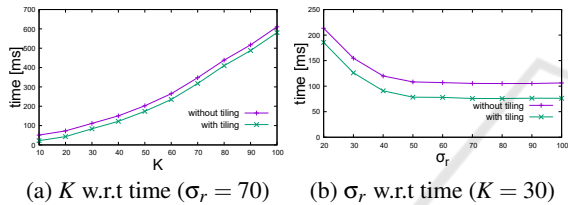

(a) K w.r.t time ($\sigma_r = 70$)   (b) σ_r w.r.t time ($K = 30$)

Figure 4: K and σ_r w.r.t time [ms] (NLM approximation with 4 channels).

Figure 4a is computational time w.r.t. the number of K and Fig. 4b is one w.r.t. σ_r. The calculation time with tiling is faster than without tiling. When σ_r is small, the speed is slow, though it is a constant-time algorithm, because we need to compute quite small numbers, i.e., subnormal numbers. The same problem is found in the normal BF (Maeda et al., 2018) cases.

Secondly, we evaluated the CTF. The filtering input image was the depth map of Tsukuba, and the guidance image was the depth map and the RGB image of Tsukuba (Fig. 5). The depth map is obtained from (Scharstein and Szeliski, 2002; El-Etriby et al., 2007). PCA compressed the 4 channels guide image from 1 to 3 channels. Figure 5 shows the filtered images by $\sigma_r = 70$, $\sigma_r' = 50$ and 3 channels. Figure 6 shows the accuracy results w.r.t. changing σ_r and σ_r'. We used averaged PSNR with 1000 trials. The ground truth image for measuring PSNR was a 4-dimensional HDGF output without PCA. Figure 6a, in the dimensions compressed to 2 and 3, PSNR is improved by increasing the smoothing parameter σ_r for the RGB original image. However, in the dimension compressed to 1, PSNR is constant at a low value. PSNR does not depend on σ_r due to the 1-st component of the guide image is mostly the depth map by PCA. Figure 6b, in the dimensions compressed to 2 and 3, it did not change significantly with increasing

the smoothing parameter of the depth map. In the dimension compressed to 1, PSNR degrades with too large the depth map component.


(a) RGB image   (b) depth map   (c) output

Figure 5: Input/Output of CTF (Tsukuba: $\sigma_r = 70$, $\sigma_r' = 50$, and 3-dimensional filtering result).
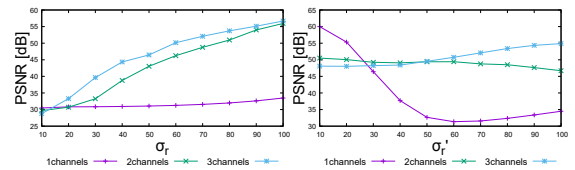
(a) changing RGB's range kernel (fix $\sigma_r' = 70$)   (b) changing depth map's range kernel (fix $\sigma_r = 70$)

Figure 6: Smoothing parameters w.r.t. PSNR [dB] (CTF approximation: $K = 30$).

Finally, we evaluated the DBF. We used Book data for RGB and IR images. The filtering image was the RGB image, and the guidance image is the RGB+IR 4 channels image. Figure 7 shows the filtered images by $\sigma_r = 40$, $\sigma_r' = 70$ and 3 channels. Figure 8 shows the accuracy results w.r.t. changing the smoothing parameters σ_r and σ_r'. We used averaged PSNR with 1000 trials. In the DBF, the compressed guide image in a dimension is mostly the IR image information by PCA. However, since the DBF is filtering for the RGB image, whose color information is mostly determined by the input image, PSNR can be improved by increasing the parameters of the IR image.

(a) RGB image   (b) IR image   (c) output

Figure 7: Input/Output of DBF (Books: $\sigma_r = 40$, $\sigma_r' = 70$, and 3-dimensional filtering result).
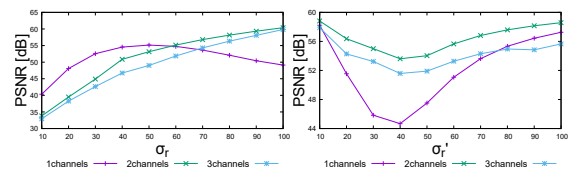
(a) changing RGB's range kernel (fix $\sigma_r' = 70$)   (b) changing IR's range kernel (fix $\sigma_r = 70$)

Figure 8: Smoothing parameters w.r.t. PSNR [dB] (DBF approximation: $K = 30$).

6 CONCLUSIONS

This paper handled several edge-preserving filters as the HDGF, which includes the NLM, DBF, and CTF. We also accelerate the various HDGF by the clustering-based HDGF. Also, we improved the accuracy and computational speed of the HDGF by PCA, K-means++, and tiling. In our experiments, the clustering-based HDGF with tiling showed that its performance is high in the accuracy and speed aspects.

ACKNOWLEDGEMENTS

REFERENCES

Adams, A., Baek, J., and Davis, M. A. (2010). Fast high-dimensional filtering using the permutohedral lattice. *Computer Graphics Forum*, 29(2):753–762.

Adams, A., Gelfand, N., Dolson, J., and Levoy, M. (2009). Gaussian kd-trees for fast high-dimensional filtering. *ACM Transactions on Graphics*, 28(3):21.

Arthur, D. and Vassilvitskii, S. (2007). K-means++: The advantages of careful seeding. In *Proc. Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*.

Awate, S. P. and Whitaker, R. T. (2005). Higher-order image statistics for unsupervised, information-theoretic, adaptive, image filtering. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 44–51 vol. 2.

Bennett, E., Mason, J., and McMillan, L. (2007). Multi-spectral bilateral video fusion. *IEEE Transactions on Image Processing*, 16(5):1185–1194.

Brox, T., Kleinschmidt, O., and Cremers, D. (2008). Efficient nonlocal means for denoising of textural patterns. *IEEE Transactions on Image Processing*, 17(7):1083–1092.

Buades, A., Coll, B., and Morel, J. . (2005). A non-local algorithm for image denoising. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 60–65 vol. 2.

Buades, A., Coll, B., and Morel, J. M. (2005). A review of image denoising algorithms, with a new one. *Multiscale Modeling & Simulation*, 4(2):490–530.

Chaudhury, K. N. (2013). Acceleration of the shiftable o(1) algorithm for bilateral filtering and nonlocal means. *IEEE Transactions on Image Processing*, 22(4):1291–1300.

Chaudhury, K. N. and Dabhade, S. D. (2016). Fast and provably accurate bilateral filtering. *IEEE Transactions on Image Processing*, 25(6):2519–2528.

Chaudhury, K. N., Sage, D., and Unser, M. (2011). Fast o(1) bilateral filtering using trigonometric range kernels. *IEEE Transactions on Image Processing*, 20(12):3376–3382.

Chen, J., Paris, S., and Durand, F. (2007). Real-time edge-aware image processing with the bilateral grid. *ACM Transactions on Graphics*, 26(3).

Coupé, P., Yger, P., and Barillot, C. (2006). Fast non local means denoising for 3d mr images. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 33–40. Springer.

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

Darbon, J., Cunha, A., Chan, T. F., Osher, S., and Jensen, G. J. (2008). Fast nonlocal filtering applied to electron cryomicroscopy. In *IEEE International Symposium on biomedical imaging: from nano to macro*, pages 1331–1334. IEEE.

Deng, G. (2017). Fast compressive bilateral filter. *Electronics Letters*, 53(3):150–152.

Deriche, R. (1992). Recursively implementating the gaussian and its derivatives. In *Proc. IEEE International Conference on Image Processing (ICIP)*.

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

Eisemann, E. and Durand, F. (2004). Flash photography enhancement via intrinsic relighting. *ACM Transactions on Graphics*, 23(3):673–678.

El-Etriby, S., Al-Hamadi, A., and Michaelis, B. (2007). Dense stereo correspondence with slanted surface using phase-based algorithm. In *Proc. IEEE International Symposium on Industrial Electronics (ISIE)*.

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

Fujita, S., Matsuo, T., Fukushima, N., and Ishibashi, Y. (2015). Cost volume refinement filter for post filtering of visual corresponding. In *Proc. Image Processing: Algorithms and Systems XIII*.

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

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

Gastal, E. S. and Oliveira, M. M. (2012). Adaptive manifolds for real-time high-dimensional filtering. *ACM Transactions on Graphics (TOG)*, 31(4):1–13.

Ghosh, S. and Chaudhury, K. N. (2016). Fast bilateral filtering of vector-valued images. In *Proc. IEEE International Conference on Image Processing (ICIP)*.

Gilboa, G. and Osher, S. (2007). Nonlocal linear image regularization and supervised segmentation. *Multiscale Modeling & Simulation*, 6(2):595–630.

Karam, C., Chen, C., and Hirakawa, K. (2015). Stochastic bilateral filter for high-dimensional images. In *Proc. IEEE International Conference on Image Processing (ICIP)*.

Kervrann, C., Boulanger, J., and Coupé, P. (2007). Bayesian non-local means filter, image redundancy and adaptive dictionaries for noise removal. In *International conference on scale space and variational methods in computer vision*, pages 520–532. Springer.

Kopf, J., Cohen, M., Lischinski, D., and Uyttendaele, M. (2007). Joint bilateral upsampling. *ACM Transactions on Graphics*, 26(3).

Maeda, Y., Fukushima, N., and Matsuo, H. (2018). Effective implementation of edge-preserving filtering on cpu microarchitectures. *Applied Sciences*, 8(10).

Mahmoudi, M. and Sapiro, G. (2005). Fast image and video denoising via nonlocal means of similar neighborhoods. *IEEE Signal Processing Letters*, 12(12):839–842.

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

Matsuo, T., Kodera, N., Fukushima, N., and Ishibashi, Y. (2013). Depth map refinement using reliability based joint trilateral filter. *ECTI Transactions on Computer and Information Technology*, 7(2):108–117.

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

Mozerov, M. G. and van de Weijer, J. (2015). Global color sparseness and a local statistics prior for fast bilateral filtering. *IEEE Transactions on Image Processing*, 24(12):5842–5853.

Mueller, M., Zilly, F., and Kauff, P. (2010). Adaptive cross-trilateral depth map filtering. In *Proc. 3DTV-Conference: the True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON)*, pages 1–4.

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

Papari, G., Idowu, N., and Varslot, T. (2017). Fast bilateral filtering for denoising large 3d images. *IEEE Transactions on Image Processing*, 26(1):251–261.

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

Paris, S. and Durand, F. (2009). A fast approximation of the bilateral filter using a signal processing approach. *International Journal of Computer Vision*, 81(1):24–52.

Peng, H. and Rao, R. (2009). Hyperspectral image enhancement with vector bilateral filtering. In *IEEE International Conference on Image Processing (ICIP)*, pages 3713–3716.

Petschnigg, G., Agrawala, M., Hoppe, H., Szeliski, R., Cohen, M., and Toyama, K. (2004). Digital photography with flash and no-flash image pairs. *ACM Transactions on Graphics*, 23(3):664–672.

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

Scharstein, D. and Szeliski, R. (2002). A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1/2/3):7–42.

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

Sugimoto, K., Fukushima, N., and Kamata, S. (2016b). Fast bilateral filter for multichannel images via soft-assignment coding. In *Proc. Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*.

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

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

Sumiya, Y., Fukushima, N., Sugimoto, K., and Kamata, S. (2020). Extending compressive bilateral filtering for arbitrary range kernel. In *Proc. IEEE International Conference on Image Processing (ICIP)*.

Tasdizen, T. (2008). Principal components for non-local means image denoising. In *IEEE International Conference on Image Processing (ICIP)*. IEEE.

Tasdizen, T. (2009). Principal neighborhood dictionaries for nonlocal means image denoising. *IEEE Transactions on Image Processing*, 18(12):2649–2660.

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

Tu, W., Lai, Y., and Chien, S. (2016). Constant time bilateral filtering for color images. In *Proc. IEEE International Conference on Image Processing (ICIP)*.

Wang, J., Guo, Y., Ying, Y., Liu, Y., and Peng, Q. (2006). Fast non-local algorithm for image denoising. In *2006 International Conference on Image Processing*, pages 1429–1432. IEEE.

Weiss, B. (2006). Fast median and bilateral filtering. *ACM Transactions on Graphics*, 25(3):519–526.

Yang, Q., Ahuja, N., and Tan, K. H. (2015). Constant time median and bilateral filtering. *International Journal of Computer Vision*, 112(3):307–318.

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

Zhang, M. and Gunturk, B. K. (2008). Multiresolution bilateral filtering for image denoising. *IEEE Transactions on image processing*, 17(12):2324–2333.