

Extending Guided Image Filtering for High-Dimensional Signals

Shu Fujita¹ and Norishige Fukushima²

¹ Nagoya University, Nagoya, Japan

² Nagoya Institute of Technology, Nagoya, Japan

Abstract. We extend guided image filtering for high-dimensional signals. The guided image filter is one of edge-preserving filtering, and various applications are proposed with the filter. The guided image filter can compute in constant time, which means that the computational time is constant to the size of the filtering kernel. The filter assumes the local linear model in each kernel. The local linear model and constant time property are convenient for various applications. The guided image filter, however, suffers from noises when the kernel radius is large. The noises are caused by violating a local linear model. Moreover, unexpected noises and complex textures often deteriorate the condition of the local linearity. Therefore, we propose high-dimensional guided image filtering to overcome the problems. Our experimental results show that our high-dimensional guided image filtering and a novel framework which utilize the high-dimensional guided image filtering can work robustly and efficiently for various image processing.

1 Introduction

Edge-preserving filtering has recently attracted attention and becomes fundamental tool in image processing. The filtering techniques such as bilateral filtering [3, 32, 35] and guided image filtering [18] are used for various applications including image denoising [5], high dynamic range imaging [9], detail enhancement [4, 11], flash/no-flash photography [28, 10], up-sampling/super resolution [24], depth map denoising [25, 15], guided feathering [18, 23] and haze removing [20].

A representative formulation of edge-preserving filtering is weighted averaging, i.e., finite impulse response (FIR) filtering, based on space and color weights that are computed from distances among neighborhood pixels. When the distance and the weighting function are Euclidean and Gaussian respectively, the formulation becomes the bilateral filter [35], which is a representative edge-preserving filter. The bilateral filter has useful properties but is known as time-consuming; thus, a number of acceleration methods have been actively proposed [29, 30, 38, 27, 6, 37, 14, 33]. As the other formulation, there is a formulation using geodesic distance. The representative examples are domain transform filtering [16] and recursive bilateral filtering [36, 39]. They are formulated as infinite

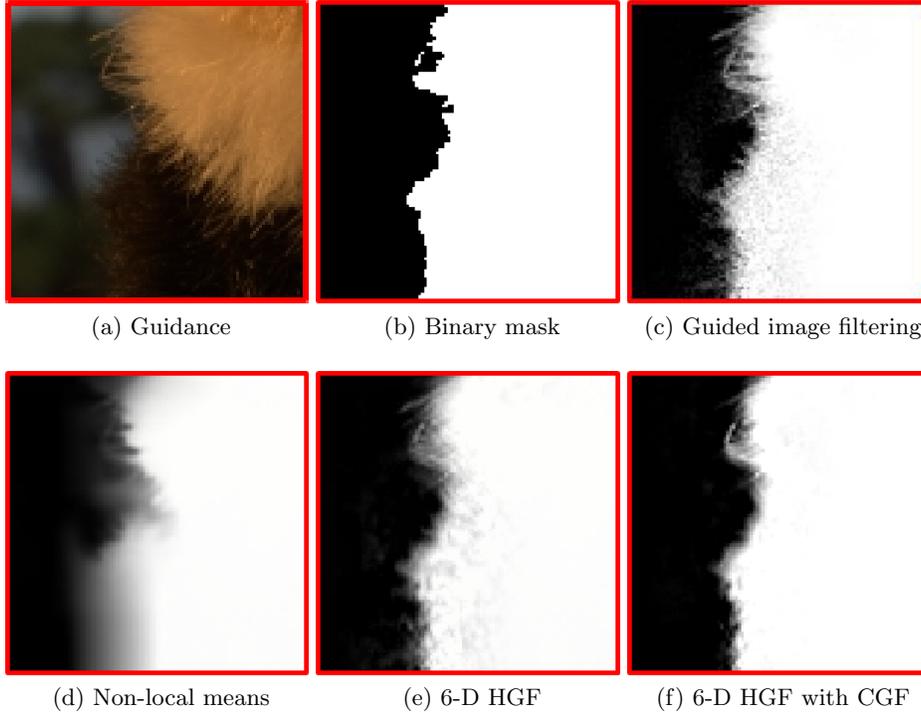


Fig. 1: Guided feathering results. (c) contains noises around object boundaries, while our results (e) and (f) can suppress such noises.

impulse response (IIR) filtering and represented by the combination of horizontal and vertical 1D filtering. These methods, therefore, can efficiently smooth images.

The guided image filter [18, 19], which is one of the efficient edge-preserving filters, has a different assumption from the previously introduced filtering methods. The guided image filter assumes a local linear model in each local kernel. Its property is convenient and essential for several applications in computational photography [9, 28, 24, 18, 20]. Furthermore, the guided image filter can efficiently compute in constant time, which means that the computational cost is independent of the size of filtering kernel. This fact is also useful for fast visual corresponding problems [21]. The local linear model is, however, violated by unexpected noises such as Gaussian noises and multiple kinds of textures. Such situation often happens when the size of the kernel is large. Then, the resulting image may contain noises. Figure 1 demonstrates feathering [18]. The result of guided image filtering (Figure 1 (c)) contains noises around border of the object.

For noise-robust implementation, a number of studies employed patch-wise processing such as non-local means filtering [5] and DCT denoising [40, 13].

Patch-wise processing gathers intensity or color information in each local patch to channels or dimensions of a pixel. In particular, non-local means filtering obtains filtering weights from the gathered information between target and reference pixels. Since patch-wise processing utilizes the richer information, it can robustly work for noisy information compared to pixel-wise processing. The extension has been also discussed as high-dimensional representation such as high-dimensional Gaussian filtering [2, 1, 17, 14]. However, these previous filters for the high-dimensional signals cannot support guided image filtering. Figure 1 (d) shows the result of non-local means filtering that is extended to joint filtering for feathering. The result has been over-smoothed because of the loose of the characteristic of the local linearity.

Therefore, we propose a high-dimensional extension of guided image filtering for obtaining robust property. We call the extension as high-dimensional guided image filtering (HGF). We firstly extend the guided image filtering so that the filter can handle high-dimensional information. In this regard, letting d be the number of dimensions of the guidance image, the computational complexity of HGF becomes $O(d^{2.807\dots})$ as pointed in [17]. Consequently, we also introduce a dimensionality reduction technique for HGF to suppress the computational cost. Furthermore, we introduce a novel framework for HGF, named as *combining guidance filtering (CGF)*. The novel framework builds a new guidance image by combining the HGF output with the guidance image, and then re-executes HGF using the combined guidance image. This framework exploits the characteristics of HGF that can utilize high-dimensional information, and can give the more robust performance to HGF. Figures 1 (e) and (f) indicate our results. Our HGF suppresses the noises, and HGF with CGF further improves the noise problem.

Note that this paper is an extension version of our conference paper [12]. The main extended part is proposed part of the CGF and associated experimental results.

2 Related Works

We discuss several acceleration methods of high-dimensional filtering in this section.

Paris and Durand [27] introduced the high-dimensional space, called as the bilateral grid [7], that is defined by adding the intensity domain to the spatial domain. We can obtain edge-preserving results by linear filtering on the bilateral grid. The bilateral grid is, however, computationally inefficient because the high-dimensional space is huge. As a result, the bilateral grid requires down-sampling of the space for efficient filtering, but the computational resource and the memory footprints are expensive especially when the dimension of guidance information is high. The Gaussian kd-trees [2] and the permutohedral lattice [1] focus on representing the high-dimensional space with point samples to overcome the problems. These methods have succeeded to alleviate the computational complexity when the filtering dimension is high. However, since these works still

require a significant amount of calculation and memory, they are not sufficiently for real-time applications.

The adaptive manifold [17] is a slightly different approach. The three methods described above focus on how represents and expands each dimension. By contrast, the adaptive manifold samples the high-dimensional space at scattered manifolds adapted to the input signal. This fact means that the method avoids that pixels are enclosed into cells to perform barycentric interpolation. This property enables us to compute a high-dimensional space efficiently and reduces the memory requirement. The property is the reason that the adaptive manifold is more efficient than other high-dimensional filtering methods [27, 2, 1]. On the other hand, the accuracy is lower than them. The adaptive manifold causes quantization artifacts depending on the parameters.

3 High-Dimensional Guided Image Filtering

We introduce our high-dimensional extension techniques for guided image filtering [18, 19] in this section. We firstly extend the guided image filtering to high-dimensional information. Next, a dimensionality reduction technique is introduced for efficient computing. We finally present combining guidance filtering, which is a new framework for HGF, to further suppress noises caused by violation of the local linear model.

3.1 Definition

Guided image filtering assumes a local linear model between an input guidance image \mathbf{I} and an output image q . The assumption of the local linear model is also invariant for our HGF. Let \mathbf{J} denote a n -dimensional guidance image. We assume that \mathbf{J} is generated from the guidance image \mathbf{I} using a function f :

$$\mathbf{J} = f(\mathbf{I}). \quad (1)$$

The function f constructs a high-dimensional image from the low-dimensional image signal \mathbf{I} ; for example, the function is utilizing square neighborhood centered at a focusing pixel, discrete cosine transform (DCT) or principle components analysis (PCA) of the guidance image \mathbf{I} .

HGF utilizes this high-dimensional image \mathbf{J} as the guidance image; thus, the output q is derived from a linear transform of \mathbf{J} in a square window ω_k centered at a pixel k . When we let p be an input image, the linear transform is represented as follows:

$$q_i = \mathbf{a}_k^T \mathbf{J}_i + b_k, \quad \forall i \in \omega_k. \quad (2)$$

Here, i is a pixel position, and \mathbf{a}_k and b_k are linear coefficients. In this regard, \mathbf{J}_i and \mathbf{a}_k represent $n \times 1$ vectors. Moreover, the linear coefficients can be derived

by the solution used in [18, 19]. Let $|\omega|$ denote the number of pixels in ω_k , and let U be a $n \times n$ identical matrix. The linear coefficients are computed by:

$$\mathbf{a}_k = (\Sigma_k + \epsilon U)^{-1} \left(\frac{1}{|\omega|} \sum_{i \in \omega_k} \mathbf{J}_i p_i - \boldsymbol{\mu}_k \bar{p}_k \right) \quad (3)$$

$$b_k = \bar{p}_k - \mathbf{a}_k^T \boldsymbol{\mu}_k, \quad (4)$$

where $\boldsymbol{\mu}_k$ and Σ_k are the $n \times 1$ mean vector and the $n \times n$ covariance matrix of \mathbf{J} in ω_k , ϵ is a regularization parameter, and $\bar{p}_k (= \frac{1}{|\omega|} \sum_{i \in \omega_k} p_i)$ represents the mean of p in ω_k .

Finally, we compute the filtering output by applying the local linear model to all local windows in the whole image. Note that q_i in each local window including a pixel i is not same. Therefore, the filter output is computed by averaging all the possible values of q_i as follows:

$$q_i = \frac{1}{|\omega|} \sum_{k: i \in \omega_k} (\mathbf{a}_k \mathbf{J}_i + b_k) \quad (5)$$

$$= \bar{\mathbf{a}}_i^T \mathbf{J}_i + \bar{b}_i, \quad (6)$$

where $\bar{\mathbf{a}}_i = \frac{1}{|\omega|} \sum_{k \in \omega_i} \mathbf{a}_k$ and $\bar{b}_i = \frac{1}{|\omega|} \sum_{k \in \omega_i} b_k$.

Computational time of HGF does not depend on the kernel radius that is the inherent ability of guided image filtering. HGF consists of many times of box filtering and per-pixel small matrix operations. The box filtering can compute in $O(1)$ time [8], however, the number of times of box filtering linearly depends on the dimensions of the guidance image. Also, the order of the matrix operations depends on exponentially in the dimensions.

3.2 Dimensionality Reduction

For efficient computing, we utilize PCA for dimensionality reduction. The dimensionality reduction technique has been proposed in [34] for non-local means filtering or high-dimensional Gaussian filtering. The approach aims for finite impulse response filtering with Euclidean distance. We extend the dimensionality technique for HGF.

For HGF, the guidance image \mathbf{J} is converted to new guidance information that is projected onto the lower dimensional subspace determined by PCA. Let Ω be a set of all pixel positions in \mathbf{J} . To conduct PCA, we should firstly compute the $n \times n$ covariance matrix Σ_Ω for the set of all guidance image pixel \mathbf{J}_i . The covariance matrix Σ_Ω is computed as follows:

$$\Sigma_\Omega = \frac{1}{|\Omega|} \sum_{i \in \Omega} (\mathbf{J}_i - \bar{\mathbf{J}})(\mathbf{J}_i - \bar{\mathbf{J}})^T, \quad (7)$$

where $|\Omega|$ and $\bar{\mathbf{J}}$ are the number of all pixels and the mean of \mathbf{J} in the whole image, respectively. After that, pixel values in the guidance image \mathbf{J} are projected

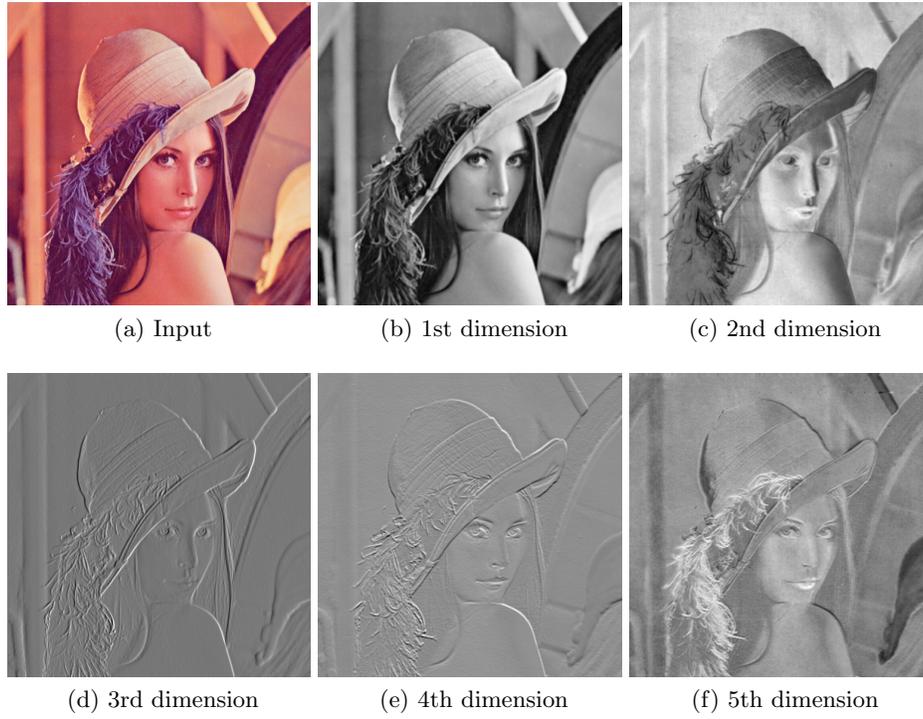


Fig. 2: PCA result. We construct the color original high-dimensional guidance image from 3×3 square neighborhood in each pixel of the input image. We reduce the dimension $27 = (3 \times 3 \times 3)$ to 5.

onto d -dimensional PCA subspace by the inner product of the guidance image pixel \mathbf{J}_i and the eigenvectors \mathbf{e}_j ($1 \leq j \leq d, 1 \leq d \leq n$, where d is a constant value) of the covariance matrix Σ_Ω . Let \mathbf{J}^d be a d -dimensional guidance image, then the projection is performed as:

$$J_{ij}^d = \mathbf{J}_i \cdot \mathbf{e}_j, \quad 1 \leq j \leq d, \quad (8)$$

where J_{ij}^d is the pixel value in the j -th dimension of \mathbf{J}_i^d , and $\mathbf{J}_i \cdot \mathbf{e}_j$ represents the inner product of the two vectors. We show an example of the PCA result of each eigenvector \mathbf{e} in Fig. 2.

In this way, we can obtain the d -dimensional guidance image \mathbf{J}^d . This guidance image \mathbf{J}^d is used by replacing \mathbf{J} in Eqs. (2), (3), (5), and (6). Moreover, each dimension in \mathbf{J}^d can be weighed by the eigenvalues $\boldsymbol{\lambda}$, where is a $d \times 1$ vector, of the covariance matrix Σ_Ω . Note that the eigenvalue elements from the $(d+1)$ -th to n -th are discarded because HGF only use d dimensions. Hence, the identical matrix U in Eq. (3) can be weighed as to the eigenvalues $\boldsymbol{\lambda}$. Then, we

take the element-wise inverse of the eigenvalues $\boldsymbol{\lambda}$:

$$\mathbf{E}_d = U \boldsymbol{\lambda}^{inv} \quad (9)$$

$$= \begin{bmatrix} \frac{1}{\lambda_1} & & \\ & \ddots & \\ & & \frac{1}{\lambda_d} \end{bmatrix}, \quad (10)$$

where \mathbf{E}_d represents a $d \times d$ diagonal matrix, $\boldsymbol{\lambda}^{inv}$ represents the element-wise inverse eigenvalues, and λ_x is the x -th eigenvalue. Note that we take the logarithm of the eigenvalues $\boldsymbol{\lambda}$ depending on applications and normalize the eigenvalue based on the 1st eigenvalue λ_1 . Taking the element-wise inverse of $\boldsymbol{\lambda}$ is to use the small ϵ for the dimension having the large eigenvalue as compared to the small eigenvalue. The reason is that the elements of $\boldsymbol{\lambda}$ satisfy $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_d$, and the eigenvector whose eigenvalue is large is more important. As a result, we can preserve the characters of the image in the principal dimension.

Therefore, we can obtain the final coefficient \mathbf{a}_k instead of using Eq. (3) in the case of high-dimensional case as follows:

$$\mathbf{a}_k = (\Sigma_k^d + \epsilon \mathbf{E}_d)^{-1} \left(\frac{1}{|\omega|} \sum_{i \in \omega_k} \mathbf{J}_i^d p_i - \boldsymbol{\mu}_k^d \bar{p}_k \right), \quad (11)$$

where and $\boldsymbol{\mu}_k^d$ and Σ_k^d are the $d \times 1$ mean vector and the $d \times d$ covariance matrix of \mathbf{J}^d in ω_k .

3.3 Combining Guidance Filtering

Our extension of the HGF can utilize high-dimensional signals. In other words, HGF can use multiple single-channel images as the guidance information by using the function f as merging multiple image channels. By utilizing this property and extending HGF, we present a novel framework—named as *combining guidance filtering (CGF)*.

The overview of CGF is shown in Fig. 3. Our CGF contains three main steps; (1) computing a filtered result using initial guidance information $J^{(0)}$, (2) generating new guidance information $J^{(t)}$ by combining the filtered result $q^{(t)}$ with the initial guidance information $J^{(0)}$, and (3) re-executing HGF using the combined guidance image $J^{(t)}$. Here, the steps (2) and (3) are repeated, and t represents the number of iterations. According to our preliminary experiments, 2–3 iterations is appropriate to obtain adequate results. Note that the filtering target image is not changed from the initial input image for avoiding an over-smoothing problem. This framework works well in recovering edges from additional guidance information as guided feathering [18]. It results from the fact that the additional guidance information is not discarded and is added to new guidance information. Moreover, the filtered guidance image added to new guidance information plays an important role to suppress noises.

Our CGF framework is similar to the framework of rolling guidance image filtering proposed by Zhang *et al.* [41]. The rolling guidance image filtering is

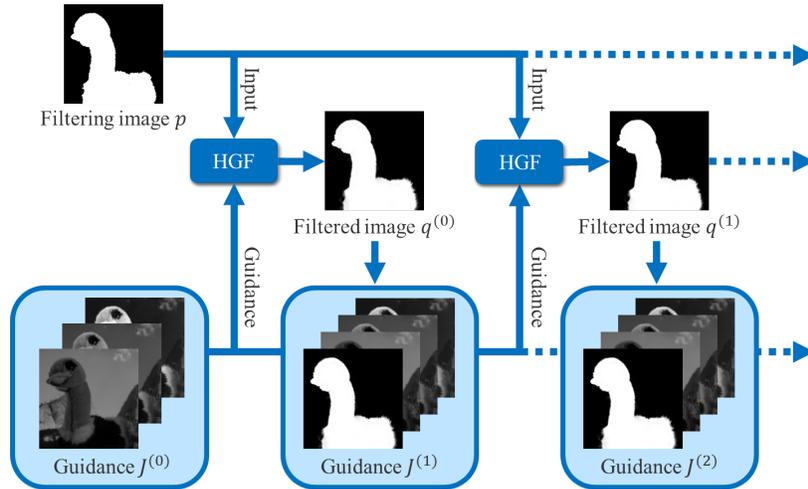


Fig. 3: Overview of CGF using our HGF. This figure shows the case of $d = 3$, i.e., the initial guidance images are three.

iterative processing with the fixed input image and the updated guidance. The rolling guidance image filtering is limited to direct filtering, i.e., the filter is not utilized joint filtering, such as feathering. Thus, their work aims at image smoothing to remove detailed textures. On the other hand, our work can deal joint filtering and mainly aims at edge recovery from additional guidance information.

4 Experimental Results

In this section, we evaluate the performance of HGF concerning efficiency and also verify the characteristics by using several applications. Note that we use GF for representing the conventional guided image filter [18, 19] in this section. In our experiments, each pixel of high-dimensional images \mathbf{J} has multiple pixel values that consist of a fixed-size square neighborhood around each pixel in original guidance image \mathbf{I} . Note that the dimensionality is reduced by the PCA approach discussed in Sec. 3.2.

We firstly reveal the processing time of HGF. We have implemented our proposed and competition methods written in C++ with Visual Studio 2010 on Windows 7 64 bit. The code is parallelized by OpenMP. The CPU for the experiments is 3.50 GHz Intel Core i7-3770K. The input images whose resolution is 1-megapixel, i.e., 1024×1024 , are grayscale or color images.

Figure 4 shows the result of the processing time. The processing time of HGF exponentially increases as the guidance image dimensionality becomes high. From this cost increasing result, the dimensionality reduction is essential for HGF. Also, the computational cost of PCA is small as compared with

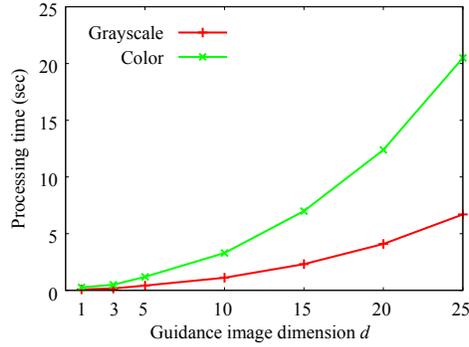


Fig. 4: Processing time of high-dimensional guided image filtering with respect to guidance image dimensions.

the increase of the filtering time by increasing the dimensionality. Therefore, although the computational cost becomes high by increasing the dimensionality, the problem is not significant. Tasdizen [34] also remarked that the performance of the dimensionality reduction peaks at around 6. The fact is also shown in following our experiments.

Figure 5 shows the result of the dimension sensitivity of HGF. Note that we obtain the binary input mask by using GrabCut [31]. We can improve the edge-preserving effect of HGF by increasing the dimension. The amount of the improvement is, however, slight in the case of over 10-D. Thus, we do not need to increase the dimension.

Next, we discuss the characteristics between GF and HGF. As mentioned in Sec. 1, GF can transfer detailed regions such as feathers, but it may cause noises near the object boundary at the same time (see Fig. 1 (c)). By contrast, HGF can suppress the noises while the detailed regions are transferred as shown in Fig. 1 (e). The noise suppression ability can be further improved by CGF as shown in Fig. 1 (f). Note that we apply 2 iterations for CGF, i.e., we set $t = 2$. Therefore, we can apply CGF if we hope the better results.

We also show the detailed results of guided feathering and alpha matting in Fig. 6. The whole guidance images and initial masks used in this experiment are the same in Fig. 5. The result of guided image filtering causes noises and color mixtures near the object boundary. HGF can alleviate these problems and suppress noises and color mixtures. However, some noises and blurs remain near the object boundary. These problems can improve by applying CGF. The result of HGF with CGF further suppress noises and has clear boundaries compared to the other methods as shown in Fig. 6 (c).

Figure 7 shows the image abstraction results. Note that the result takes 3 times iterations of filtering. As shown in Figs. 7 (b) and (d), since the local linear model is often violated in filtering with large kernel, the pixel values are scattered. On the other hands, HGF can smooth the image without such problem (see Figs. 7 (c) and (e)).

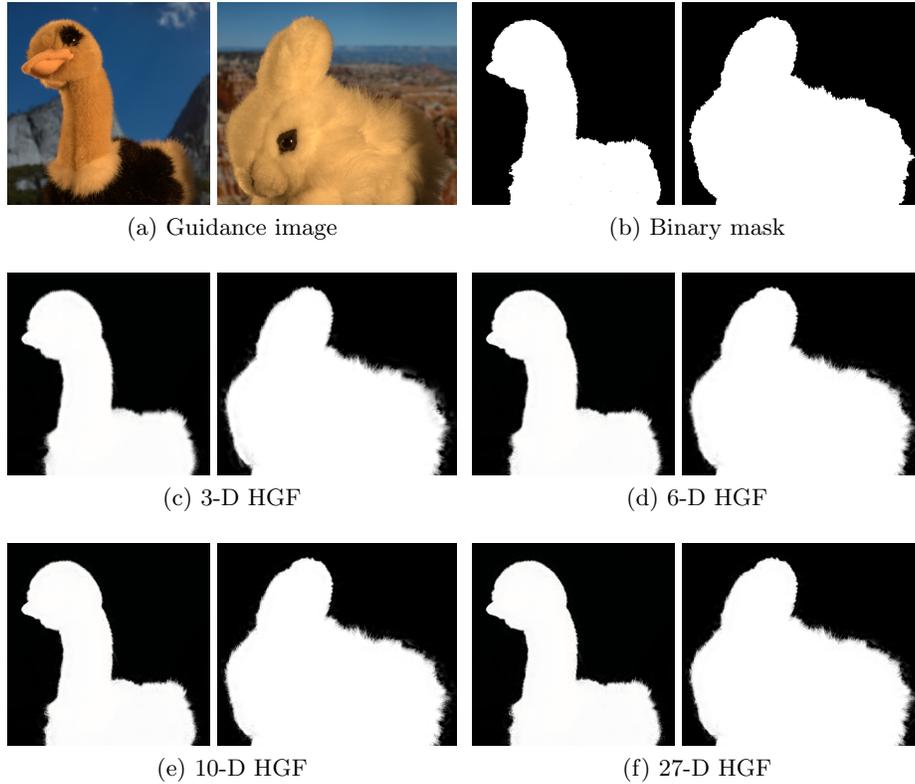
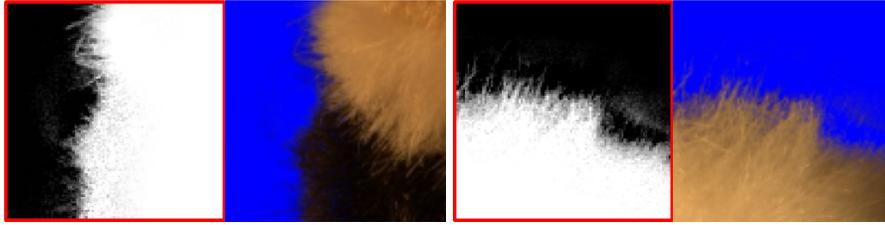


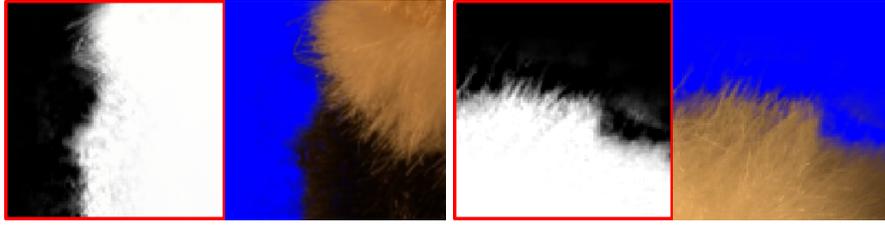
Fig. 5: Dimension sensitivity. The color patch size for high-dimensional image is 3×3 , i.e., the complete dimension is 27. The parameters are $r = 15$, $\epsilon = 10^{-6}$.

HGF also has excellent performance for haze removing [20]. The haze removing results and the transition maps are shown in Fig. 8. In the case of GF, the transition map preserves major textures while there are over-smoothed regions near the detailed regions or object boundaries, e.g., between trees or branches. The over-smoothing effect affects the haze removal in such regions. In our case, the transition map of HGF preserves such detailed texture; thus, HGF can remove the haze better than GF in the detailed regions. For these results, HGF is effective for preserving the detailed areas or textures.

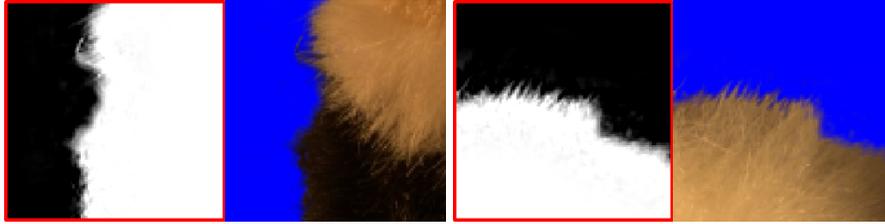
As the other application for high-dimensional guided image filtering, there is an image classification with a hyperspectral image. The hyperspectral image has various wavelength information, which is useful for distinguishing different objects. Although we can obtain a good result by using support vector machine classifier [26], Kang et al. improved the accuracy of image classification by applying guided image filtering [22]. They made a guidance image using PCA from the hyperspectral image, but most of the information was unused because



(a) GF



(b) 6-D HGF



(c) 6-D HGF with CGF

Fig. 6: Guided feathering and matting results using different methods. The parameters are the same as Fig. 5.

GF cannot utilize the high-dimensional data. Our extension has an advantage in such case. Since HGF can utilize high-dimensional data, we can further improve the accuracy of classification by adding the remaining information.

Figure 9 and Tab. 1 show the result of classification of Indian Pines dataset, which was acquired by Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) sensor. We objectively evaluate the classification accuracy by using the three metrics: the overall accuracy (OA), the average accuracy (AA), and the kappa coefficient, which are widely used for evaluating classification. OA denotes the ratio of correctly classified pixels. AA denotes the average ratio of correctly classified pixels in each class. The kappa coefficient denotes the ratio of correctly classified pixels corrected by the number of pure agreements. We can confirm that the HGF result achieves the better result than GF. Especially, the detailed regions are improved in our method. The accuracy is objectively further improved as shown in Tab. 1.

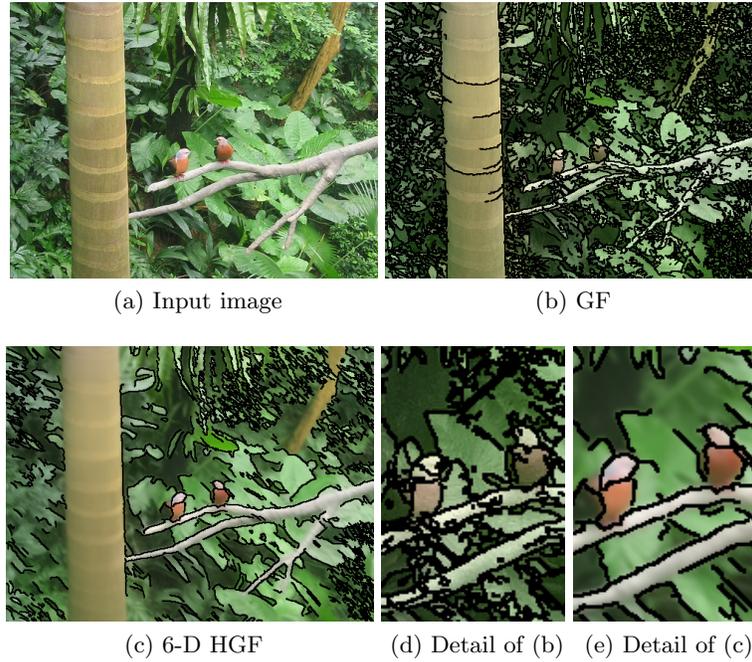


Fig. 7: Image abstraction. The local patch size for high-dimensional image is 3×3 . The parameters for GF and HGF are $r = 25$, $\epsilon = 0.04^2$.

5 Conclusion

We proposed high-dimensional guided image filtering (HGF) by extending guided image filtering [18, 19]. The extension allows the guided image filter to utilize high-dimensional signals, e.g., local square patches and hyperspectral images and obtain the robustness for unexpected textures, which is a limitation of guided image filtering. Our high-dimensional extension has a limitation that the computational cost becomes high as the number of dimensions increases. To alleviate this limitation, we simultaneously introduce a dimensionality reduction technique for efficient computing. Furthermore, we also presented a novel framework named as combining guidance filtering (CGF) in this study. We have proposed this framework to more exploits the characteristics of HGF that can utilize high-dimensional information. As a result, HGF with CGF obtains robustness and can further suppress noises caused by violation of the local linear model. Experimental results showed that HGF can work robustly in noisy regions and transfer detailed regions. In addition, HGF can compute efficiently by using the dimensionality reduction technique.

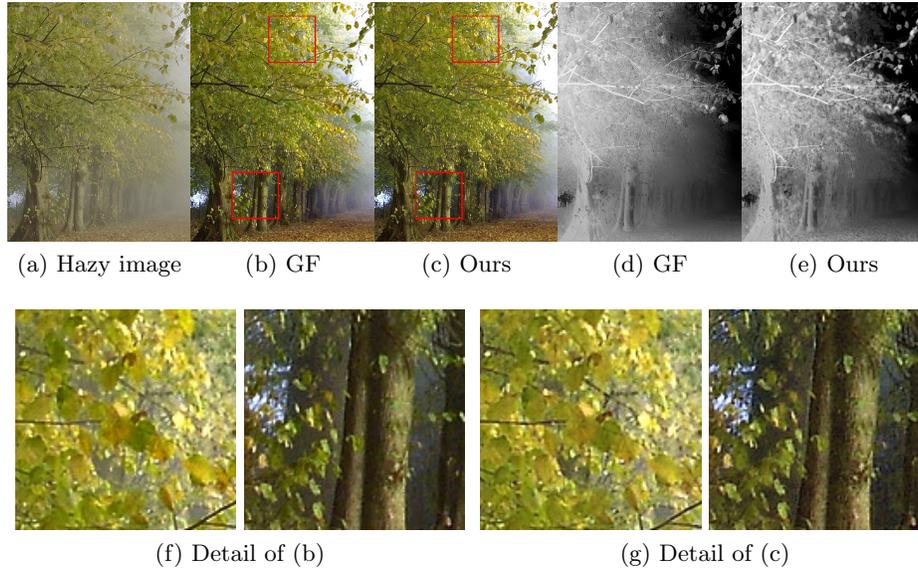


Fig. 8: Haze removing. The bottom row images represent transition maps of (b) and (c). The local patch size for high-dimensional image is 5×5 . The parameters for GF and HGF are $r = 20$, $\epsilon = 10^{-4}$.

Table 1: Classification accuracy [%] of the classification results shown in Fig. 9.

Method	OA	AA	Kappa
SVM	81.0	79.1	78.3
GF	92.7	93.9	91.6
HGF	92.8	94.1	91.8

Acknowledgment This work was supported by JSPS KAKENHI Grant Number 15K16023.

References

1. Adams, A., Baek, J., Davis, M.A.: Fast high-dimensional filtering using the permutohedral lattice. *Computer Graphics Forum* 29(2), 753–762 (2010)
2. Adams, A., Gelfand, N., Dolson, J., Levoy, M.: Gaussian kd-trees for fast high-dimensional filtering. *ACM Trans. on Graphics* 28(3) (2009)
3. Aurich, V., Weule, J.: Non-linear gaussian filters performing edge preserving diffusion. In: *Mustererkennung*. pp. 538–545. Springer (1995)
4. Bae, S., Paris, S., Durand, F.: Two-scale tone management for photographic look. *ACM Trans. on Graphics* 25(3), 637–645 (2006)

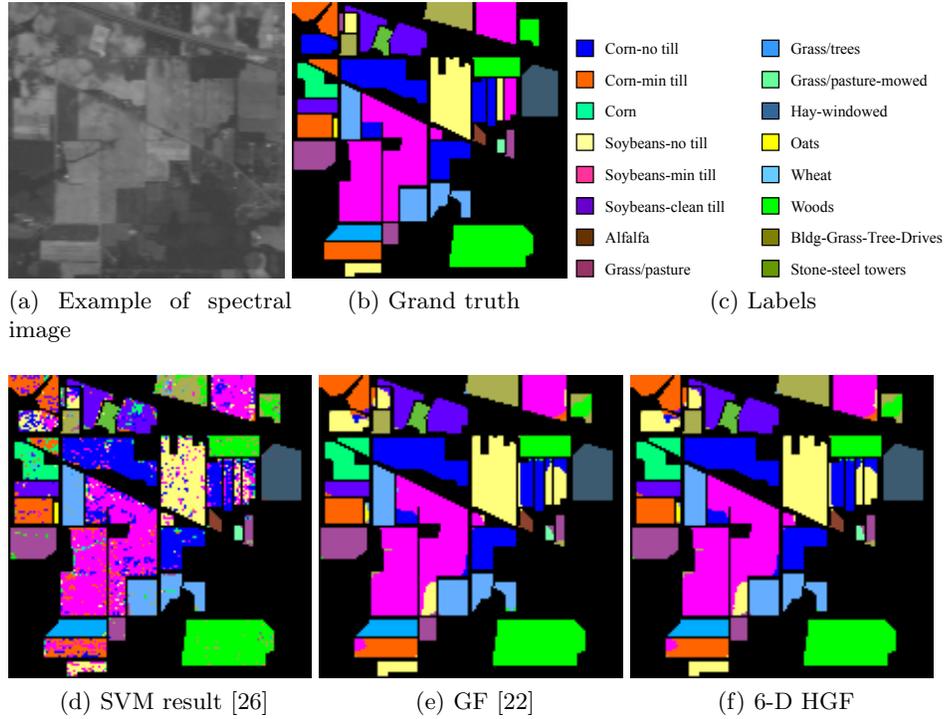


Fig. 9: Classification result of Indian Pines image. The image of (a) represents a spectral image that the wavelength is $0.7\mu\text{m}$. The parameters for GF and HGF are $r = 4$, $\epsilon = 0.15^2$.

5. Buades, A., Coll, B., Morel, J.M.: A non-local algorithm for image denoising. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2005)
6. Chaudhury, K.: Acceleration of the shiftable mbiO(1) algorithm for bilateral filtering and nonlocal means. IEEE Trans. on Image Processing 22(4), 1291–1300 (2013)
7. Chen, J., Paris, S., Durand, F.: Real-time edge-aware image processing with the bilateral grid. ACM Trans. on Graphics 26(3) (2007)
8. Crow, F.C.: Summed-area tables for texture mapping. In: Proc. ACM SIGGRAPH. pp. 207–212 (1984)
9. Durand, F., Dorsey, J.: Fast bilateral filtering for the display of high-dynamic-range images. ACM Trans. on Graphics 21(3), 257–266 (2002)
10. Eisemann, E., Durand, F.: Flash photography enhancement via intrinsic relighting. ACM Trans. on Graphics 23(3), 673–678 (2004)
11. Fattal, R., Agrawala, M., Rusinkiewicz, S.: Multiscale shape and detail enhancement from multi-light image collections. ACM Trans. on Graphics 26(3) (2007)
12. Fujita, S., Fukushima, N.: High-dimensional guided image filtering. In: Proc. International Conference on Computer Vision Theory and Applications (VISAPP)

- (2016)
13. Fujita, S., Fukushima, N., Kimura, M., Ishibashi, Y.: Randomized redundant dct: Efficient denoising by using random subsampling of dct patches. In: Proc. ACM SIGGRAPH Asia Technical Briefs (2015)
 14. Fukushima, N., Fujita, S., Ishibashi, Y.: Switching dual kernels for separable edge-preserving filtering. In: Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (2015)
 15. Fukushima, N., Inoue, T., Ishibashi, Y.: Removing depth map coding distortion by using post filter set. In: Proc. IEEE International Conference on Multimedia and Expo (ICME) (2013)
 16. Gastal, E.S.L., Oliveira, M.M.: Domain transform for edge-aware image and video processing. *ACM Trans. on Graphics* 30(4) (2011)
 17. Gastal, E.S.L., Oliveira, M.M.: Adaptive manifolds for real-time high-dimensional filtering. *ACM Trans. on Graphics* 31(4) (2012)
 18. He, K., Shun, J., Tang, X.: Guided image ffiltering. In: Proc. European Conference on Computer Vision (ECCV) (2010)
 19. He, K., Shun, J., Tang, X.: Guided image filtering. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 35(6), 1397–1409 (2013)
 20. He, K., Sun, J., Tang, X.: Single image haze removal using dark channel prior. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2009)
 21. Hosni, A., Rhemann, C., Bleyer, M., Rother, C., Gelautz, M.: Fast cost-volume filtering for visual vorrespondence and beyond. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 35(2), 504–511 (2013)
 22. Kang, X., Li, S., Benediktsson, J.: Spectral-spatial hyperspectral image classification with edge-preserving filtering. *IEEE Trans. on Geoscience and Remote Sensing* 52(5), 2666–2677 (2014)
 23. Kodera, N., Fukushima, N., Ishibashi, Y.: Filter based alpha matting for depth image based rendering. In: IEEE Visual Communications and Image Processing (VCIP) (2013)
 24. Kopf, J., Cohen, M., Lischinski, D., Uyttendaele, M.: Joint bilateral upsampling. *ACM Trans. on Graphics* 26(3) (2007)
 25. Matsuo, T., Fukushima, N., Ishibashi, Y.: Weighted joint bilateral filter with slope depth compensation filter for depth map refinement. In: International Conference on Computer Vision Theory and Applications (VISAPP) (2013)
 26. Melgani, F., Bruzzone, L.: Classification of hyperspectral remote sensing images with support vector machines. *IEEE Trans. on Geoscience and Remote Sensing* 42(8), 1778–1790 (2004)
 27. Paris, S., Durand, F.: A fast approximation of the bilateral filter using a signal processing approach. *International Journal of Computer Vision* 81(1), 24–52 (2009)
 28. Petschnigg, G., Agrawala, M., Hoppe, H., Szeliski, R., Cohen, M., Toyama, K.: Digital photography with flash and no-flash image pairs. *ACM Trans. on Graphics* 23(3), 664–672 (2004)
 29. Pham, T.Q., Vliet, L.J.V.: Separable bilateral filtering for fast video preprocessing. In: Proc. IEEE International Conference on Multimedia and Expo (ICME) (2005)
 30. Porikli, F.: Constant time $o(1)$ bilateral filtering. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2008)
 31. Rother, C., Kolmogorov, V., Blake, A.: Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM Trans. on Graphics* 23(3), 309–314 (2004)
 32. Smith, S.M., Brady, J.M.: Susan—a new approach to low level image processing. *International Journal of Computer Vision* 23(1), 45–78 (1997)

33. Sugimoto, K., Kamata, S.I.: Compressive bilateral filtering. *IEEE Trans. on Image Processing* 24(11), 3357–3369 (2015)
34. Tasdizen, T.: Principal components for non-local means image denoising. In: *Proc. IEEE International Conference on Image Processing (ICIP)* (2008)
35. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: *Proc. IEEE International Conference on Computer Vision (ICCV)* (1998)
36. Yang, Q.: Recursive bilateral filtering. In: *Proc. European Conference on Computer Vision (ECCV)* (2012)
37. Yang, Q., Ahuja, N., Tan, K.H.: Constant time median and bilateral filtering. *International Journal of Computer Vision* 112(3), 307–318 (2014)
38. Yang, Q., Tan, K.H., Ahuja, N.: Real-time $o(1)$ bilateral filtering. In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2009)
39. Yang, Q.: Recursive approximation of the bilateral filter. *IEEE Trans. on Image Processing* 24(6), 1919–1927 (2015)
40. Yu, G., Sapiro, G.: Dct image denoising: A simple and effective image denoising algorithm. *Image Processing On Line* 1 (2011)
41. Zhang, Q., Shen, X., Xu, L., Jia, J.: Rolling guidance filter. In: *Proc. European Conference on Computer Vision (ECCV)*. pp. 815–830 (2014)