

# Extending Guided Image Filtering for High-Dimensional Signals

Shu Fujita<sup>1,2</sup> and Norishige Fukushima<sup>2</sup>

<sup>1</sup> Nagoya University, Nagoya, Japan

s.fujita@fujii.nuee.nagoya-u.ac.jp

<sup>2</sup> Nagoya Institute of Technology, Nagoya, Japan

fukushima@nitech.ac.jp

**Abstract.** This paper presents an extended method of guided image filtering (GF) for high-dimensional signals and proposes various applications for it. The important properties of GF include edge-preserving filtering, local linearity in a filtering kernel region, and the ability of constant time filtering in any kernel radius. GF can suffer from noise caused by violations of the local linearity when the kernel radius is large. Moreover, unexpected noise and complex textures can further degrade the local linearity. We propose high-dimensional guided image filtering (HGF) and a novel framework named combining guidance filtering (CGF). Experimental results show that HGF and CGF can work robustly and efficiently for various applications in image processing.

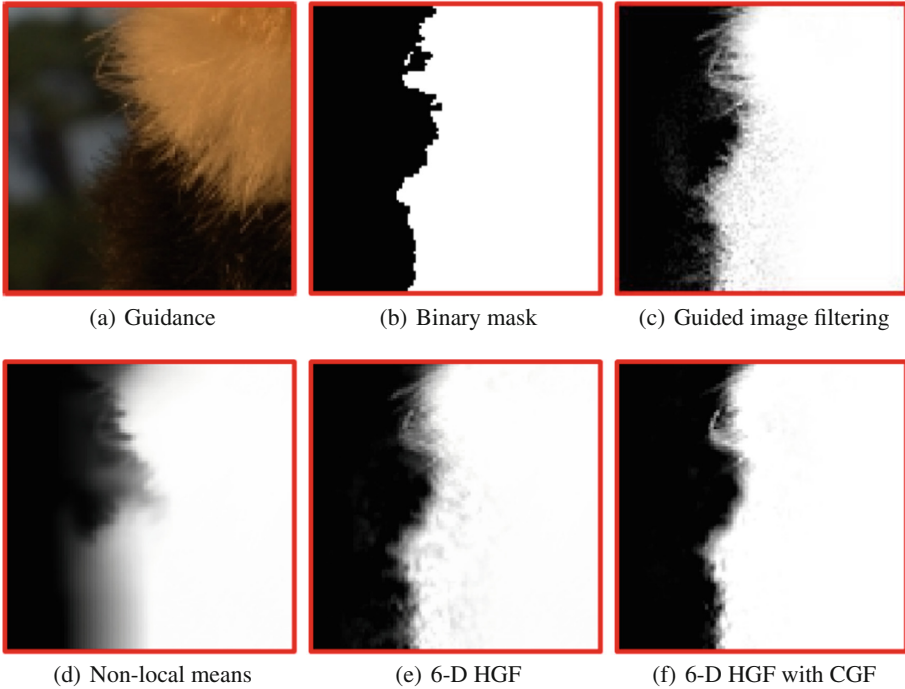
## 1 Introduction

Recently, edge-preserving filtering has been attracting increasing attention and has become as fundamental tool in image processing. The filtering techniques such as bilateral filtering [33] and guided image filtering (GF) [17] are used for various applications including image denoising [4], high dynamic range imaging [8], detail enhancement [3, 10], flash/no-flash photography [9, 27], super resolution [23], depth map denoising [14, 24], guided feathering [17, 22] and haze removal [19].

One representative formulation of edge-preserving filtering is weighted averaging, i.e., finite impulse response filtering, based on space and color weights that are computed from distances among neighborhood pixels. When the distance and weighting function are Euclidean and Gaussian, respectively, the formulation becomes a bilateral filter [33], which is a representative edge-preserving filter. The bilateral filter has useful properties, but is known as time-consuming; thus, several acceleration methods have been proposed [5, 13, 26, 28, 29, 31, 35, 36]. Another formulation uses geodesic distance, representative examples being domain transform filtering [15], and recursive bilateral filtering [34, 37]. These are formulated as infinite impulse response filtering, represented by a combination of horizontal and vertical one-dimensional filtering. These methods can, therefore, efficiently smooth images.

A guided image filter [17,18], which is an efficient edge-preserving filter, is based on an assumption different from those of previously introduced filtering methods. The guided image filter assumes a local linear model in each local kernel. This property is convenient and essential for several applications in computational photography [8,17,19,23,27]. Furthermore, the guided image filter can efficiently compute in constant time, with the result that the computational cost is independent of the size of the filtering kernel. This fact is also useful for fast visual correspondence problems [20]. However, the local linear model can be violated by unexpected noise, such as Gaussian noise, and different types of textures. Such situations often arise when the kernel is large. Then, the resulting image can contain noise. Figure 1 demonstrates feathering [17], where the result of GF (Fig. 1(c)) contains noise around the border of an object.

For noise-robust implementation, several studies have used patch-wise processing, such as non-local means filtering [4] and, discrete cosine transform (DCT) denoising [12,38]. 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 the target and reference pixels. Since patch-wise processing utilizes richer information, it can work more robustly for noisy information compared to pixel-wise processing. Extension to high-dimensional representations



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

such as high-dimensional Gaussian filtering has also been discussed [1, 2, 13, 16]. However, these previous filters for high-dimensional signals cannot support GF. Figure 1(d) shows the result of non-local means filtering extended to joint filtering for feathering. The result has been over-smoothed because of the loose of the local linearity.

Therefore, we propose a high-dimensional extension of GF to obtain robustness. We call the extension high-dimensional guided image filtering (HGF). We first extend GF to ensure the filter can handle high-dimensional information. In this regard, assuming  $d$  as the number of dimensions of the guidance image, the computational complexity of HGF becomes  $O(d^{2.807\dots})$  as noted in [16]. We then introduce a dimensionality reduction technique for HGF to reduce the computational cost. Furthermore, we introduce a novel framework for HGF, called *combining guidance filtering (CGF)*, which 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 provides more robust performance to HGF and utilizes the HGF characteristics that can use high-dimensional information. Figures 1(e) and (f) show our results. HGF suppresses the noise, and HGF with CGF further reduces the noise.

This paper is an extended version of our conference paper [11]. The main extension part is the section on CGF and the associated experimental results.

## 2 Related Works

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

Paris and Durand [26] introduced the bilateral grid [6], which is a high-dimensional space defined by adding the intensity domain to the spatial domain. We can obtain edge-preserving results using 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. However, the computational resources and memory footprints are expensive, especially when the dimension of guidance information is high. Gaussian kd-trees [2] and permutohedral lattice [1] focus on representing the high-dimensional space with point samples to overcome these problems. These methods have succeeded in alleviating computational complexity when the filtering dimension is high. However, since they still require a significant amount of calculation and memory, they are not sufficient for real-time applications.

Adaptive manifolds [16] provides a slightly different approach. The three methods described above focus on how to represent and expand each dimension. In contrast, the adaptive manifolds samples the high-dimensional space at scattered manifolds adapted to the input signal. The method thereby avoids having pixels are enclosed into cells to perform barycentric interpolation. This property enables us to efficiently compute a high-dimensional space and reduces the memory requirement. This is why the use of the adaptive manifolds is more efficient than that of other high-dimensional filtering methods [1, 2, 26]. Its accuracy is,

however, lower than theirs. The adaptive manifolds causes quantization artifacts depending on the parameters.

### 3 High-Dimensional Guided Image Filtering

We introduce our high-dimensional extension techniques for GF [17, 18] in this section. We first extend GF to high-dimensional information. Next, a dimensionality reduction technique is introduced to increasing computing efficiency. We finally present CGF, which is a new framework for HGF, to further suppress noise caused by violation of the local linearity.

#### 3.1 Definition

The guided image filter 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 an  $n$ -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 might use a square neighborhood centered at a focusing pixel, DCT or principle components analysis (PCA) of the guidance image  $\mathbf{I}$ .

HGF uses 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  $\omega_k$  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 using the solution used in [17, 18]. Let  $|\omega|$  denote the number of pixels in  $\omega_k$ , and  $U$  be an  $n \times n$  identity 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  $\Sigma_k$  are the  $n \times 1$  mean vector and  $n \times n$  covariance matrix of  $\mathbf{J}$  in  $\omega_k$ ,  $\epsilon$  is a regularization parameter, and  $\bar{p}_k (= \frac{1}{|\omega|} \sum_{i \in \omega_k} p_i)$  represents the mean of  $p$  in  $\omega_k$ .

Finally, we compute the filtering output by applying the local linear model to all local windows in the entire image. Note that the  $q_i$  values in all local windows

including a pixel  $i$  are not necessarily the same. Therefore, the filter output is computed by averaging all 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  $\bar{b}_i = \frac{1}{|\omega|} \sum_{k \in \omega_i} b_k$ .

The computation time of HGF does not depend on the kernel radius that provides the inherent ability of GF. HGF consists of many instances of box filtering and per-pixel small matrix operations. The box filtering can compute in  $O(1)$  time [7], but the number of instances of box filtering linearly depends on the dimensions of the guidance image. In addition, the order of the matrix operations exponentially depends on the number of dimensions.

### 3.2 Dimensionality Reduction

For efficient computing, we use PCA for dimensionality reduction. The dimensionality reduction technique was proposed in [32] 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 into new guidance information, which is projected onto the lower-dimensional subspace determined by PCA. Let  $\Omega$  be a set of all pixel positions in  $\mathbf{J}$ . To conduct PCA, we must first compute the  $n \times n$  covariance matrix  $\Sigma_\Omega$  for the set of all guidance image pixels  $\mathbf{J}_i$ . The covariance matrix  $\Sigma_\Omega$  is computed as follows:

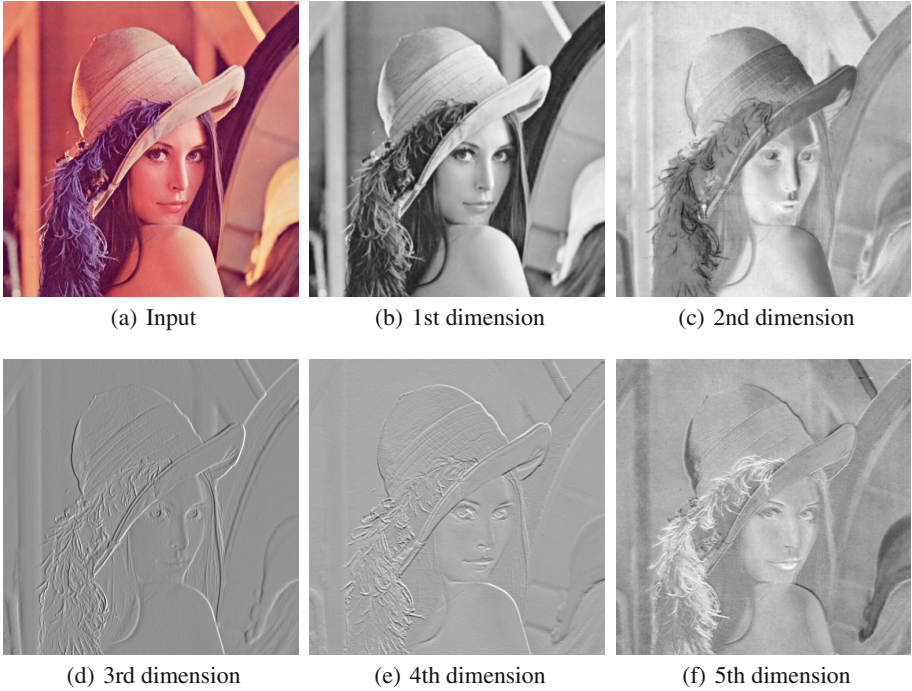
$$\Sigma_\Omega = \frac{1}{|\Omega|} \sum_{i \in \Omega} (\mathbf{J}_i - \bar{\mathbf{J}})(\mathbf{J}_i - \bar{\mathbf{J}})^T, \quad (7)$$

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 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  $\Sigma_\Omega$ . 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 manner, we can obtain the  $d$ -dimensional guidance image  $\mathbf{J}^d$ , which is used to replace  $\mathbf{J}$  in Eqs. (2), (3), (5), and (6). Moreover, each dimension in  $\mathbf{J}^d$  can be weighted by the eigenvalues  $\boldsymbol{\lambda}$ , where is a  $d \times 1$  vector, of the covariance



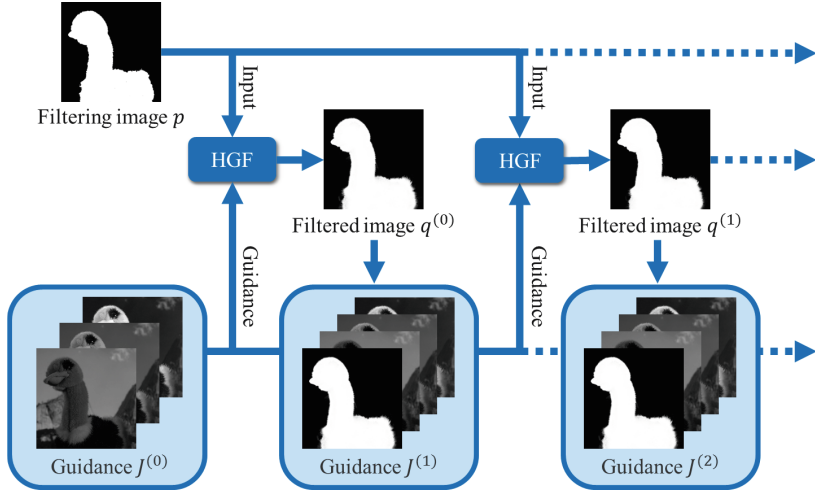
**Fig. 2.** PCA result. We construct the color original high-dimensional guidance image from  $3 \times 3$  square neighborhood in each pixel of the input image. We reduce the dimension  $27 = (3 \times 3 \times 3)$  to 5. (Color figure online)

matrix  $\Sigma_{\Omega}$ . Note that the eigenvalue elements from the  $(d + 1)$ th to  $n$ th are discarded because HGF uses only  $d$  dimensions. Hence, the identity matrix  $U$  in Eq. (3) can be weighted by the eigenvalues  $\lambda$ . Then, we take the element-wise inverse of eigenvalues  $\lambda$ :

$$\mathbf{E}_d = \text{diag}(\lambda^{inv}) \quad (9)$$

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

where  $\mathbf{E}_d$  represents a  $d \times d$  diagonal matrix,  $\lambda^{inv}$  represents the element-wise inverse eigenvalues, and  $\lambda_x$  is the  $x$ th eigenvalue. Note that we take the logarithms of the eigenvalues  $\lambda$  depending on the applications and normalize the eigenvalue based on the first eigenvalue  $\lambda_1$ . We take the element-wise inverse of  $\lambda$  to use the smallest  $\epsilon$  for the dimension having the large eigenvalue as compared to the small eigenvalue, because the elements of  $\lambda$  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.



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

Therefore, we can obtain the final coefficient  $\mathbf{a}_k$  instead of using Eq. (3) in the 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  $\boldsymbol{\mu}_k^d$  and  $\Sigma_k^d$  are the  $d \times 1$  mean vector and the  $d \times d$  covariance matrix of  $\mathbf{J}^d$  in  $\omega_k$ , respectively.

### 3.3 Combining Guidance Filtering

Our extension of 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. Utilizing this property and extending HGF, we present the novel framework CGF.

An overview of CGF is shown in Fig. 3. CGF involves 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, steps (2) and (3) are repeated, and  $t$  is the number of iterations. According to our preliminary experiments, two to three iterations are sufficient to obtain adequate results. Note that the filtering target image is not changed from the initial input image to avoid an over-smoothing problem. This framework works well in recovering edges from additional guidance information as guided feathering [17]. This is because the additional guidance information is not discarded and is added to new guidance information. Moreover, the filtered

guidance image added to the new guidance information plays an important role to suppress noise.

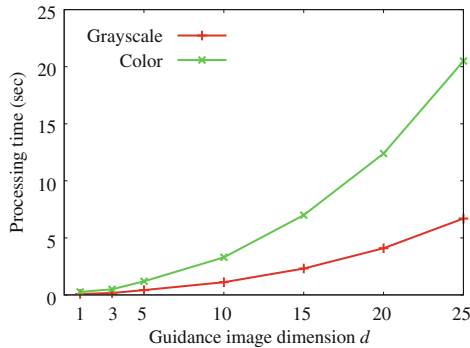
Our CGF framework is similar to the framework of rolling guidance image filtering proposed by Zhang *et al.* [39]. Rolling guidance image filtering is an iterative processing with a fixed input image and updated guidance. This filtering framework is limited to direct filtering, i.e., the filter does not utilize joint filtering, such as feathering. Thus, their work aims at image smoothing to remove detailed textures. In contrast, our work can deal with joint filtering and mainly aims primarily at edge recovery from additional guidance information.

## 4 Experimental Results

In this section, we evaluate the performance of HGF in terms of efficiency and verify its characteristics by using several applications. In our experiments, each pixel of high-dimensional images  $\mathbf{J}$  has multiple pixel values comprising a fixed-size square neighborhood around each pixel in the original guidance image  $\mathbf{I}$ . The dimensionality is reduced using the PCA approach discussed in Sect. 3.2.

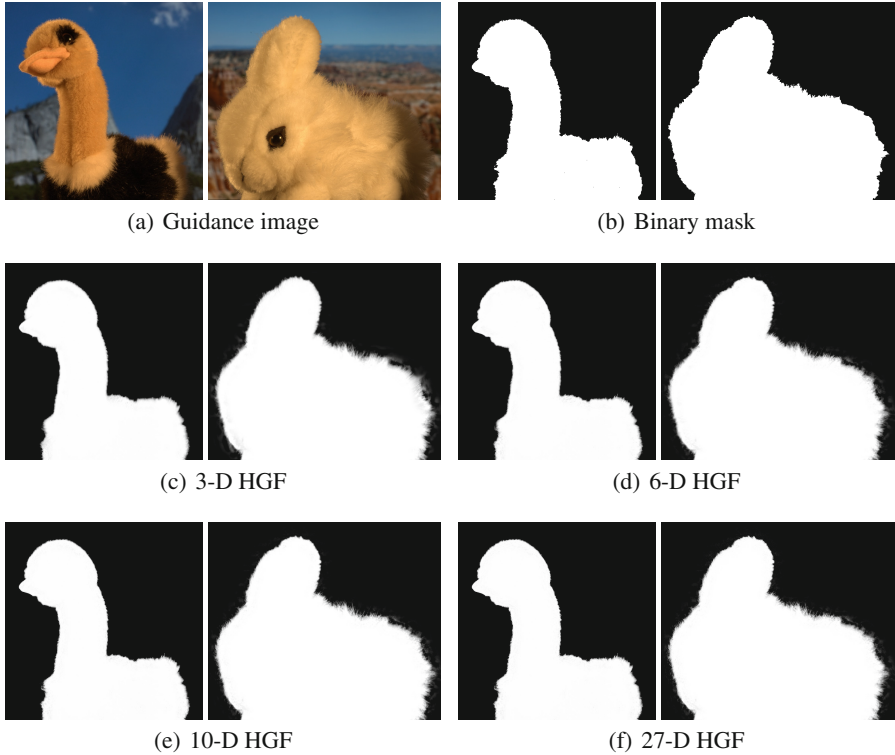
We first reveal the processing time of HGF. Our proposed and competing methods are implemented in C++ with Visual Studio 2010 on Windows 7 64 bit. The code is parallelized using OpenMP. The CPU for the experiments is a 3.50 GHz Intel Core i7-3770K. The input images, with resolution of one-megapixel, i.e.,  $1024 \times 1024$ , are grayscale or color images.

Figure 4 shows the result of the processing time. The processing time of HGF increases exponentially as the guidance image dimensionality increases. This cost increasing result makes the dimensionality reduction essential for HGF. In addition, the computational cost of PCA is small as compared with the increase in the filtering time caused by increase in dimensionality. Therefore, although the computational cost becomes high from increasing the dimensionality, the problem is not significant. Tasdizen [32] remarked that the performance of dimensionality reduction peaks at approximately six. This fact is also shown in our experiments.



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



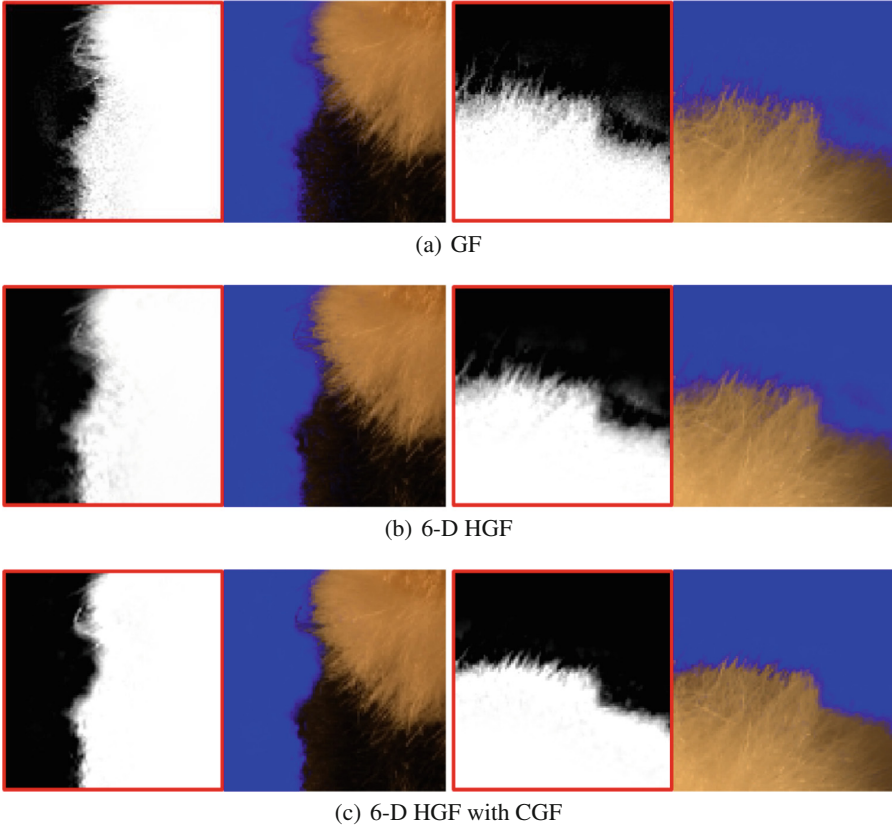


**Fig. 5.** Dimension sensitivity. The color patch size for high-dimensional image is  $3 \times 3$ , i.e., the complete dimension is 27. The parameters are  $r = 15$ ,  $\epsilon = 10^{-6}$ . (Color figure online)

Figure 5 shows the result of the dimension sensitivity of HGF. We obtain the binary input mask using GrabCut [30]. We can improve the edge-preserving effect of HGF by increasing the dimension. However, the amount of the improvement is slight in the case of more than ten dimensions. Thus, we do not need to increase the dimensionality.

Next, we compare the characteristics of GF and HGF. As mentioned in Sect. 1, GF can transfer detailed regions such as feathers, but may simultaneously cause noise near the object boundary (see Fig. 1(c)). In contrast, HGF can suppress noise while the detailed regions are transferred, as shown in Fig. 1(e). This noise suppression ability can be further improved by using CGF, as shown in Fig. 1(f). We use two iterations for CGF, i.e., we set  $t = 2$ . Therefore, we can apply CGF if we desire better results.

We also show the detailed results of guided feathering and alpha matting in Fig. 6. All guidance images and initial masks used in this experiment are the same as those in Fig. 5. The result of GF causes noise and color mixtures near the object boundary. HGF can alleviate these problems and suppress noise and

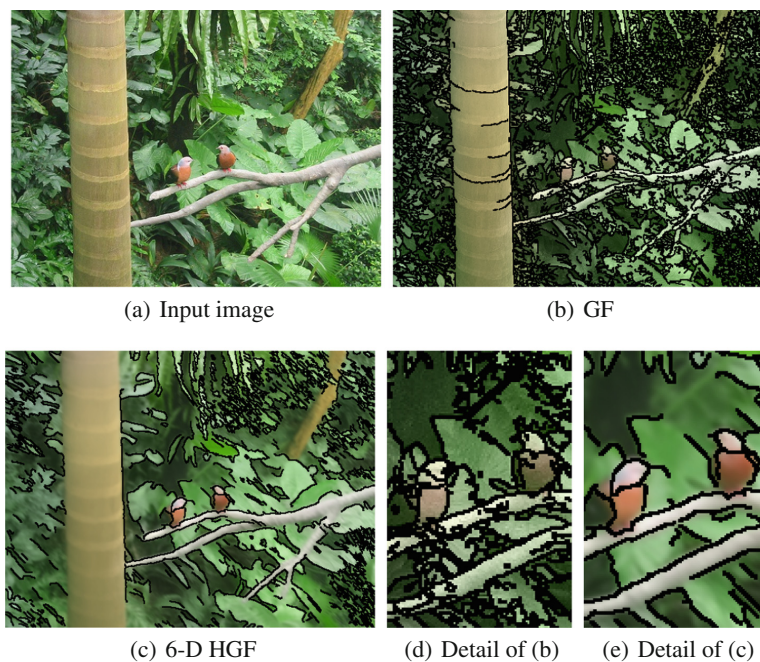


**Fig. 6.** Guided feathering and matting results using different methods. The parameters are the same as Fig. 5.

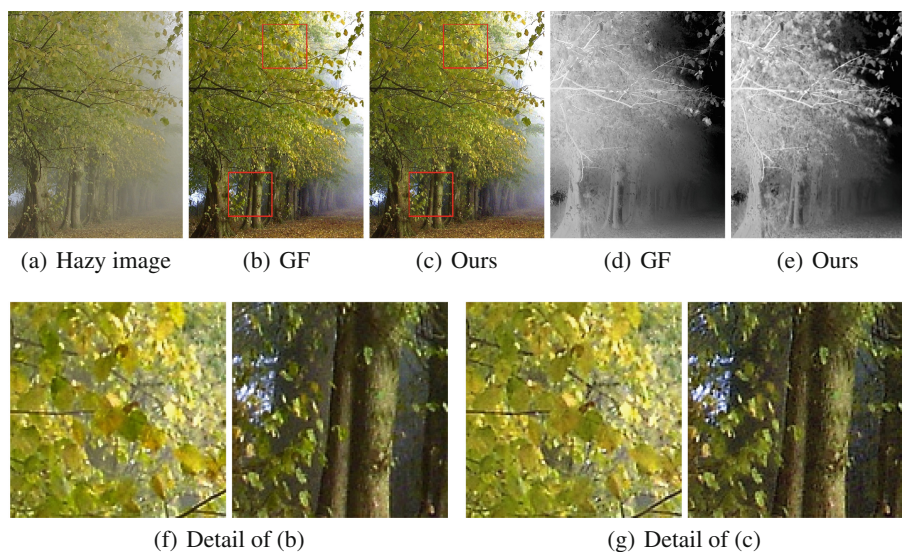
color mixtures. However, some noise and blurs remain near the object boundary. These problems can be improved by using CGF. The use of HGF with CGF further suppresses noise and results in clear boundaries compared to the other methods, as shown in Fig. 6(c).

Figure 7 shows the image abstraction results. The result requires three iterations of filtering. As shown in Figs. 7(b) and (d), since the local linear model is often violated in filtering with large kernels, the pixel values are scattered. In contrast, HGF can smooth the image without such problems (see Figs. 7(c) and (e)).

HGF also shows excellent performance for haze removal [19]. The haze removal 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 haze removal in such regions. In our case, the transition map of HGF preserves such detailed textures; thus, HGF can remove



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

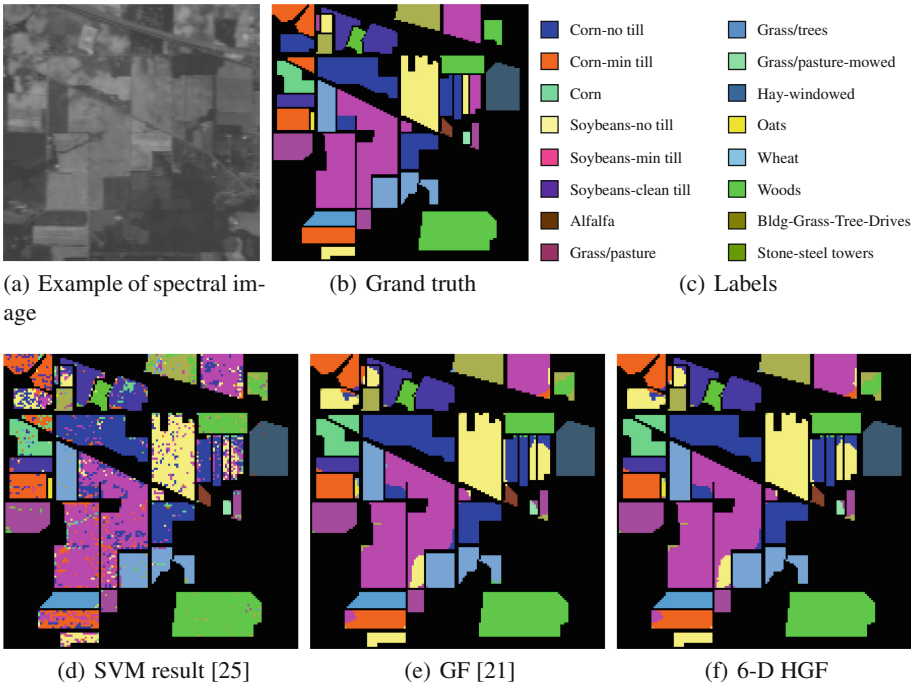


**Fig. 8.** Haze removal. The bottom row images represent transition maps of (b) and (c). The local patch size for high-dimensional image is  $5 \times 5$ . The parameters for GF and HGF are  $r = 20$ ,  $\epsilon = 10^{-4}$ .

the haze more effectively than GF in the detailed regions. These results show that HGF is effective for preserving detailed areas or textures.

An other application for HGF is image classification with a hyperspectral image. A hyperspectral image contains considerable wavelength information, which is useful for distinguishing different objects. Although we can obtain good results using support vector machine classifiers [25], Kang et al. improved the accuracy of image classification by applying GF [21]. They created a guidance image using PCA from a hyperspectral image; however, most of the information was unused because GF cannot utilize high-dimensional data. Our extension has an advantage in such cases. Since HGF can utilize high-dimensional data, we can further improve classification accuracy by adding the remaining information.

Figure 9 and Table 1 show the result of classifying the Indian Pines dataset, which was acquired using an Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) sensor. We objectively evaluate the classification accuracy by using three metrics: the overall accuracy (OA), the average accuracy (AA), and the kappa coefficient, which are widely used for evaluating classification. OA is the ratio of correctly classified pixels, AA is the average ratio of correctly classified pixels in each class, and the kappa coefficient denotes the ratio of correctly



**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  $r = 4$ ,  $\epsilon = 0.15^2$ .

classified pixels corrected by the number of pure agreements. We can confirm that HGF achieves a better result than GF. In particular, the detailed regions are improved using our method. The accuracy is further improved objectively, as shown in Table 1.

**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    | <b>92.8</b> | <b>94.1</b> | <b>91.8</b> |

## 5 Conclusion

We proposed high-dimensional guided image filtering (HGF) by extending guided image filtering [17, 18]. The extension allows the guided image filter to utilize high-dimensional signals, e.g., local square patches and hyperspectral images, and suppress unexpected noise that is a limitation of guided image filtering. Our high-dimensional extension has a limitation that the computational cost increases as the number of dimensions increases. To alleviate this limitation, we simultaneously introduce a dimensionality reduction technique for efficient computing. Furthermore, we presented a novel framework named as combining guidance filtering (CGF) in this study, to further exploit the HGF characteristics that can utilize high-dimensional information. As a result, HGF with CGF obtains robustness and can further suppress noise 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.

**Acknowledgement.** 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. *Comput. Graph. 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. Graph.* **28**(3), 21 (2009)
3. Bae, S., Paris, S., Durand, F.: Two-scale tone management for photographic look. *ACM Trans. Graph.* **25**(3), 637–645 (2006)
4. Buades, A., Coll, B., Morel, J.M.: A non-local algorithm for image denoising. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2005)

5. Chaudhury, K.: Acceleration of the shifttable  $O(1)$  algorithm for bilateral filtering and nonlocal means. *IEEE Trans. Image Process.* **22**(4), 1291–1300 (2013)
6. Chen, J., Paris, S., Durand, F.: Real-time edge-aware image processing with the bilateral grid. *ACM Trans. Graph.* **26**(3), 103 (2007)
7. Crow, F.C.: Summed-area tables for texture mapping. In: *Proceedings of ACM SIGGRAPH*, pp. 207–212 (1984)
8. Durand, F., Dorsey, J.: Fast bilateral filtering for the display of high-dynamic-range images. *ACM Trans. Graph.* **21**(3), 257–266 (2002)
9. Eisemann, E., Durand, F.: Flash photography enhancement via intrinsic relighting. *ACM Trans. Graph.* **23**(3), 673–678 (2004)
10. Fattal, R., Agrawala, M., Rusinkiewicz, S.: Multiscale shape and detail enhancement from multi-light image collections. *ACM Trans. Graph.* **26**(3), 51 (2007)
11. Fujita, S., Fukushima, N.: High-dimensional guided image filtering. In: *Proceedings of International Conference on Computer Vision Theory and Applications (VISAPP)* (2016)
12. Fujita, S., Fukushima, N., Kimura, M., Ishibashi, Y.: Randomized redundant DCT: efficient denoising by using random subsampling of DCT patches. In: *Proceedings of ACM SIGGRAPH Asia Technical Briefs* (2015)
13. Fukushima, N., Fujita, S., Ishibashi, Y.: Switching dual kernels for separable edge-preserving filtering. In: *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2015)
14. Fukushima, N., Inoue, T., Ishibashi, Y.: Removing depth map coding distortion by using post filter set. In: *Proceedings of IEEE International Conference on Multimedia and Expo (ICME)* (2013)
15. Gastal, E.S.L., Oliveira, M.M.: Domain transform for edge-aware image and video processing. *ACM Trans. Graph.* **30**(4), 69 (2011)
16. Gastal, E.S.L., Oliveira, M.M.: Adaptive manifolds for real-time high-dimensional filtering. *ACM Trans. Graph.* **31**(4), 33 (2012)
17. He, K., Shun, J., Tang, X.: Guided image filtering. In: *Proceedings of European Conference on Computer Vision (ECCV)* (2010)
18. He, K., Shun, J., Tang, X.: Guided image filtering. *IEEE Trans. Pattern Anal. Mach. Intell.* **35**(6), 1397–1409 (2013)
19. He, K., Sun, J., Tang, X.: Single image haze removal using dark channel prior. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2009)
20. Hosni, A., Rhemann, C., Bleyer, M., Rother, C., Gelautz, M.: Fast cost-volume filtering for visual correspondence and beyond. *IEEE Trans. Pattern Anal. Mach. Intell.* **35**(2), 504–511 (2013)
21. Kang, X., Li, S., Benediktsson, J.: Spectral-spatial hyperspectral image classification with edge-preserving filtering. *IEEE Trans. Geosci. Remote Sens.* **52**(5), 2666–2677 (2014)
22. Kodera, N., Fukushima, N., Ishibashi, Y.: Filter based alpha matting for depth image based rendering. In: *IEEE Visual Communications and Image Processing (VCIP)* (2013)
23. Kopf, J., Cohen, M., Lischinski, D., Uyttendaele, M.: Joint bilateral upsampling. *ACM Trans. Graph.* **26**(3), 96 (2007)
24. 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)

25. Melgani, F., Bruzzone, L.: Classification of hyperspectral remote sensing images with support vector machines. *IEEE Trans. Geosci. Remote Sens.* **42**(8), 1778–1790 (2004)
26. Paris, S., Durand, F.: A fast approximation of the bilateral filter using a signal processing approach. *Int. J. Comput. Vis.* **81**(1), 24–52 (2009)
27. Petschnigg, G., Agrawala, M., Hoppe, H., Szeliski, R., Cohen, M., Toyama, K.: Digital photography with flash and no-flash image pairs. *ACM Trans. Graph.* **23**(3), 664–672 (2004)
28. Pham, T.Q., Vliet, L.J.V.: Separable bilateral filtering for fast video preprocessing. In: *Proceedings of IEEE International Conference on Multimedia and Expo (ICME)* (2005)
29. Porikli, F.: Constant time  $O(1)$  bilateral filtering. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2008)
30. Rother, C., Kolmogorov, V., Blake, A.: GrabCut: interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.* **23**(3), 309–314 (2004)
31. Sugimoto, K., Kamata, S.I.: Compressive bilateral filtering. *IEEE Trans. Image Process.* **24**(11), 3357–3369 (2015)
32. Tasdizen, T.: Principal components for non-local means image denoising. In: *Proceedings of IEEE International Conference on Image Processing (ICIP)* (2008)
33. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: *Proceedings of IEEE International Conference on Computer Vision (ICCV)* (1998)
34. Yang, Q.: Recursive bilateral filtering. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) *ECCV 2012*. LNCS, vol. 7572, pp. 399–413. Springer, Heidelberg (2012). doi:[10.1007/978-3-642-33718-5\\_29](https://doi.org/10.1007/978-3-642-33718-5_29)
35. Yang, Q., Ahuja, N., Tan, K.H.: Constant time median and bilateral filtering. *Int. J. Comput. Vis.* **112**(3), 307–318 (2014)
36. Yang, Q., Tan, K.H., Ahuja, N.: Real-time  $O(1)$  bilateral filtering. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2009)
37. Yang, Q.: Recursive approximation of the bilateral filter. *IEEE Trans. Image Process.* **24**(6), 1919–1927 (2015)
38. Yu, G., Sapiro, G.: DCT image denoising: a simple and effective image denoising algorithm. *Image Process. On Line* **1**, 292–296 (2011)
39. Zhang, Q., Shen, X., Xu, L., Jia, J.: Rolling guidance filter. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) *ECCV 2014*. LNCS, vol. 8691, pp. 815–830. Springer, Cham (2014). doi:[10.1007/978-3-319-10578-9\\_53](https://doi.org/10.1007/978-3-319-10578-9_53)