

# High-dimensional Guided Image Filtering

Shu Fujita and Norishige Fukushima  
*Nagoya Institute of Technology, Nagoya, Japan*

Keywords: High-dimensional Filtering, Constant Time Filtering, Guided Image Filtering.

Abstract: We present high-dimensional filtering for extending guided image filtering. Guided image filtering is one of edge-preserving filtering, and the computational time is constant to the size of the filtering kernel. The constant time property is essential for edge-preserving filtering. When the kernel radius is large, however, the guided image filtering suffers from noises because of violating a local linear model that is the key assumption in the guided image filtering. Unexpected noises and complex textures often violate the local linear model. Therefore, we propose high-dimensional guided image filtering to avoid the problems. Our experimental results show that our high-dimensional guided image filtering can work robustly and efficiently for various image processing.

## 1 INTRODUCTION

Edge-preserving filtering has recently attracted attention in image processing researchers. Such filters, e.g., bilateral filtering (Tomasi and Manduchi, 1998) and non-local means filtering (Buades et al., 2005), are used for various applications including image denoising (Buades et al., 2005), high dynamic range imaging (Durand and Dorsey, 2002), detail enhancement (Bae et al., 2006; Fattal et al., 2007), flash/no-flash photography (Petschnigg et al., 2004; Eisemann and Durand, 2004), up-sampling/super resolution (Kopf et al., 2007), alpha matting (He et al., 2010) and haze removing (He et al., 2009).

Edge-preserving filtering is often represented as weighted averaging filtering by using space and color distances among neighborhood pixels. When the distance is Euclidean, and the kernel weight is Gaussian, this is a representative filter of the bilateral filter (Tomasi and Manduchi, 1998). The bilateral filter has a number of acceleration methods (Porikli, 2008; Yang et al., 2009; Paris and Durand, 2009; Pham and Vliet, 2005; Fukushima et al., 2015). Domain transform filtering (Gastal and Oliveira, 2011) and recursive bilateral filtering (Yang, 2012), which use geodesic distance, can also effectively filter images.

The guided image filter (He et al., 2010) is one of the efficient edge-preserving filters; however, the filtering property is different from the filters smoothing with pixel-wise distance. The guided image filter assumes a local linear model in a kernel. Its property is essential for several applications in computational photography (Durand and Dorsey, 2002;

Petschnigg et al., 2004; Kopf et al., 2007; He et al., 2010; He et al., 2009) and fast visual corresponding problems (Hosni et al., 2013). 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 (He et al., 2010) and the result of guided image filtering contains noises.

For efficient implementation, intensity/color information in each patch is gathered to channels or dimensions in a pixel. Patch-wise processing is effective for handling the noisy information, e.g., non-local means filtering (Buades et al., 2005) and DCT denoising (Fujita et al., 2015). Then, the dimension of the image becomes higher. The representation is efficiently smoothed as high-dimensional Gaussian filtering (Adams et al., 2009; Adams et al., 2010; Gastal and Oliveira, 2012; Fukushima et al., 2015). However, these filters do not have the similar property of the guided image filtering for computational photography. Figure 1 (e) shows the result by non-local means filtering that is extended to joint filtering for feathering. The result has been over-smoothed.

Therefore, we extend the guided image filtering to store patch-wise neighborhood pixels into high-dimensional space. We call this extension as high-dimensional guided image filtering (HGF). We firstly extend the guided image filtering to handle high-dimensional signals. In this regard, letting  $d$  be the number of dimensions of the guidance image, the computational complexity is  $O(d^{2.807\dots})$  as pointed in (Gastal and Oliveira, 2012). Therefore, we also

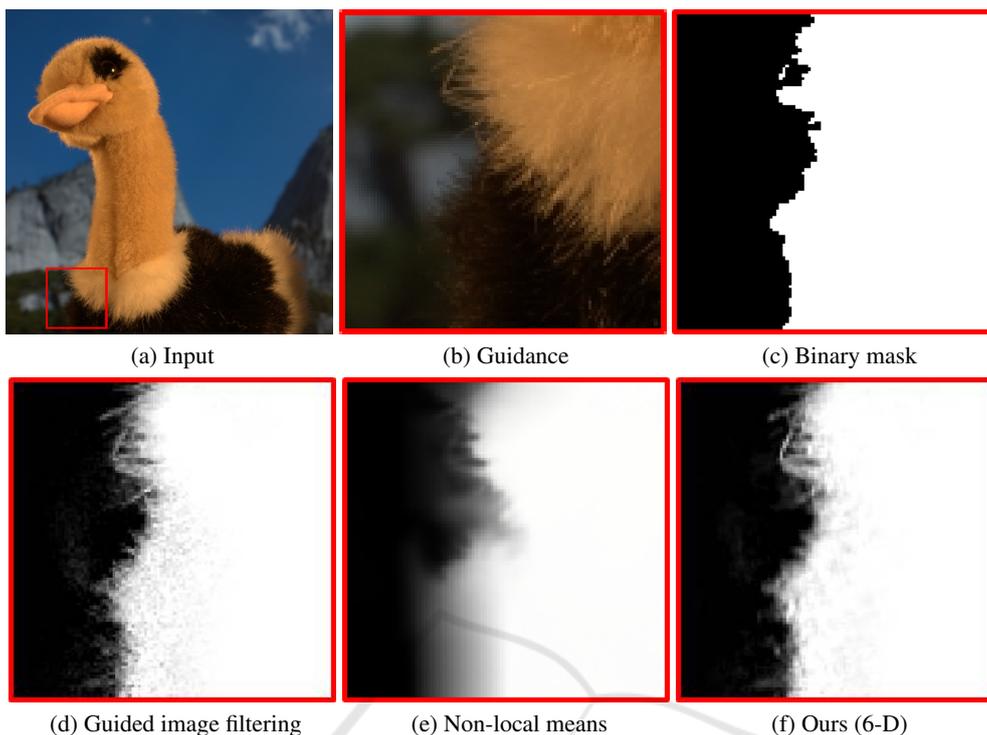


Figure 1: Feathering results from guided image filtering (c) contains noises around object boundaries, while our result from high-dimensional guided image filtering (d) suppresses such noises.

introduce a dimensionality reduction technique for HGF to suppress the computational cost. Figure 1 (f) indicates our result with reduced dimension.

## 2 RELATED WORKS

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

The bilateral grid (Paris and Durand, 2009) is used for color bilateral filtering whose dimension is three. The dimension is not high; however, the simple regular grid is computationally inefficient; thus, we use down-sampling of the grid for efficient filtering. Note that the bilateral grid directly represents the filtering space and is sparse. With this representation, the computational resource and the memory are consumed beyond necessity. Due to this, the Gaussian kd-trees (Adams et al., 2009) and the permutohedral lattice (Adams et al., 2010) focus on representing the high-dimensional space with point samples. 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 (Gastal and Oliveira, 2012)

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. Thus, we can avoid enclosing pixels 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 (Paris and Durand, 2009; Adams et al., 2009; Adams et al., 2010). 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

### 3.1 Definition

We extend guided image filtering (GF) (He et al., 2010) for high-dimensional filtering in this section. GF assumes a local linear model between an input guidance image  $I$  and an output image  $q$ . The as-

sumption of the local linear model is also invariant for our HGF. Let  $\mathbf{J}$  denote a  $n$ -dimensional guidance image.  $\mathbf{J}$  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  $\mathbf{I}$ ; for example, it uses a square neighborhood centered at a 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  $\omega_k$  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 by the solution used in (He et al., 2010). Let  $|\omega|$  denote the number of pixels in  $\omega_k$ , and let  $U$  be a  $n \times n$  identical matrix. The linear coefficients are computed by:

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

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

where  $\boldsymbol{\mu}_k$  and  $\Sigma_k$  are the  $n \times 1$  mean vector and the  $n \times n$  covariance matrix of  $\mathbf{J}$  in  $\omega_k$ ,  $\varepsilon$  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 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 GF. HGF consists of many times of box filtering, which can compute in  $O(1)$  time (Crow, 1984), and per-pixel small matrix operations. However, the number of times of box filtering linearly depends on the dimensions of the guidance image, and the order of the matrix operations depend on exponentially in the dimensions.

### 3.2 Dimensionality Reduction

For efficient computing, we utilize PCA for dimensionality reduction. The dimensionality reduction has been proposed in (Tasdizen, 2008). The approach aims for finite impulse response filtering using Euclidean distance. In this paper, we adopt the technique for HGF to extend GF.

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  $\Omega$  be a set of all pixel positions in  $\mathbf{J}$ . To conduct PCA, we should firstly compute the  $n \times n$  covariance matrix  $\Sigma_\Omega$  for the set of all guidance image pixel  $\mathbf{J}_i$ . 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:

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

where  $\mathbf{J}_{ij}^d$  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  $\Sigma_\Omega$ . 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 weighted 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  $\lambda_x$  is the  $x$ -th eigenvalue. Note that we take the logarithm of the eigenvalues  $\boldsymbol{\lambda}$  as to applications and normalize

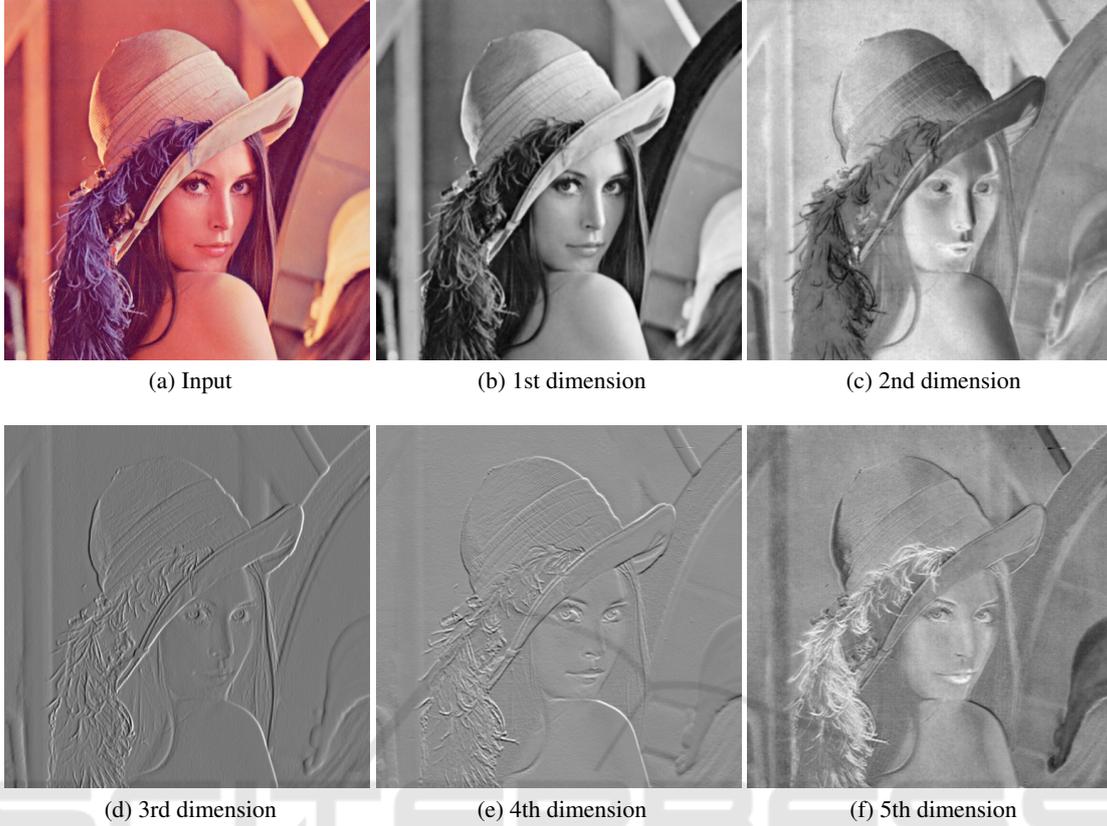


Figure 2: PCA result. We construct the color original high-dimensional guidance image from  $3 \times 3$  square neighborhood in each pixel of the input image. We reduce the dimension  $27 = (3 \times 3 \times 3)$  to 5.

the eigenvalue based on the 1st eigenvalue  $\lambda_1$ . Taking the element-wise inverse of  $\lambda$  is to use the small  $\varepsilon$  for the dimension having the large eigenvalue as compared to the small eigenvalue. The reason is that 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.

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

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

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  $\omega_k$ .

## 4 EXPERIMENTAL RESULTS

In this section, we evaluate the performance of HGF in terms of efficiency and also verify the characteristics by using several applications. In our experiments,

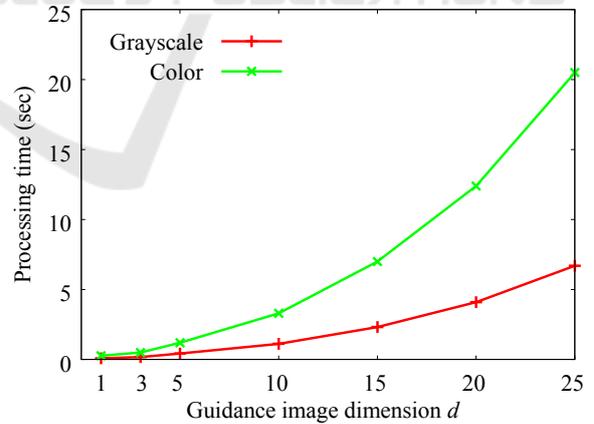


Figure 3: Processing time of high-dimensional guided image filtering with respect to guidance image dimensions.

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 competi-

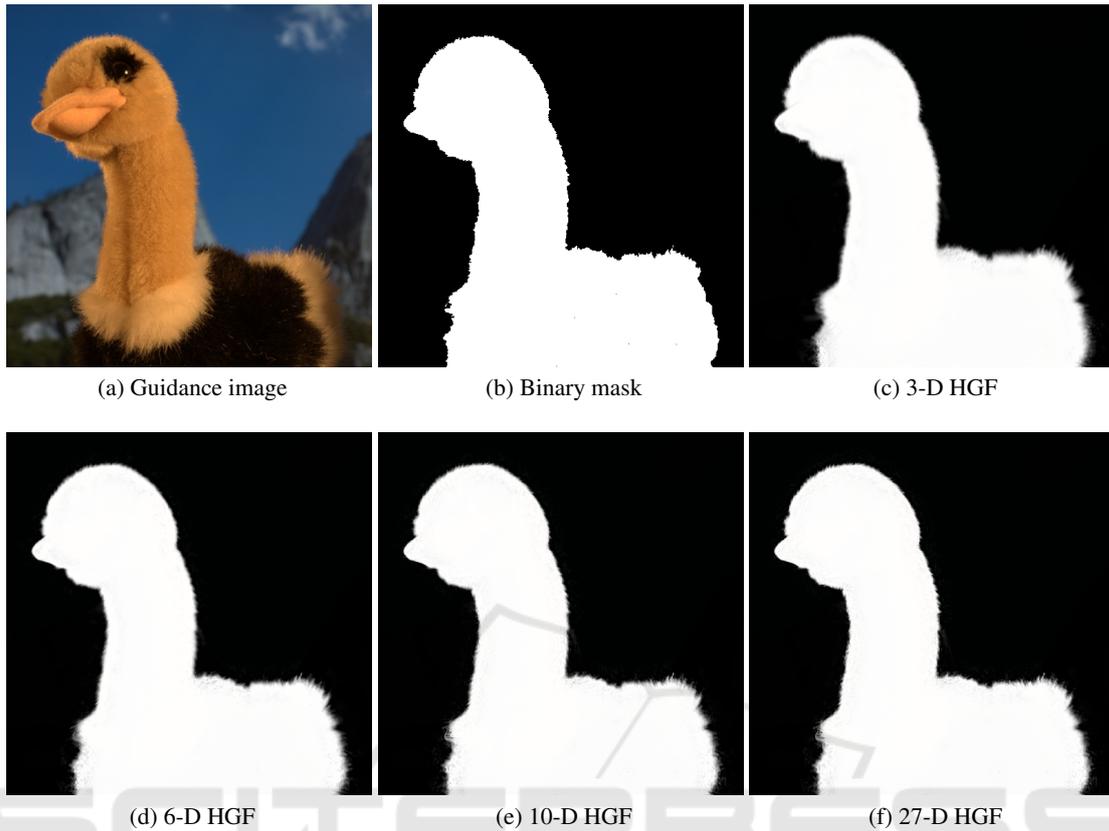


Figure 4: Dimension sensitivity. The color patch size for high-dimensional image is  $3 \times 3$ , i.e., the full dimension is 27. The parameters are  $r = 15$ ,  $\epsilon = 10^{-6}$ .

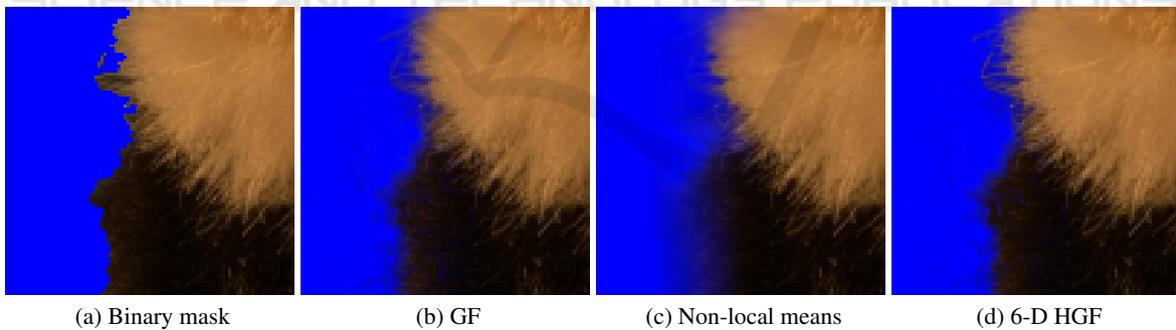


Figure 5: Matting result using alpha masks in Fig. 1.

tion 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.,  $1024 \times 1024$ , are grayscale or color images.

Figure 3 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 computa-

tional cost of PCA is small as compared with 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 (Tasdizen, 2008) also remarked that the performance of the dimensionality reduction peaks at around 6. The fact is shown in following our experiments.

Figure 4 shows the result of the dimension sensitivity of HGF. We can improve the edge-preserving effect of HGF by increasing the dimension. The

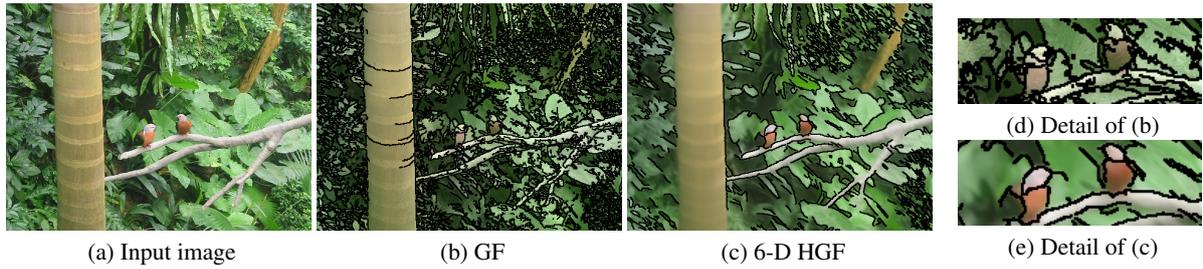


Figure 6: Image abstraction. The local patch size for high-dimensional image is  $3 \times 3$ . The parameters for GF and HGF are  $r = 25$ ,  $\varepsilon = 0.04^2$ .

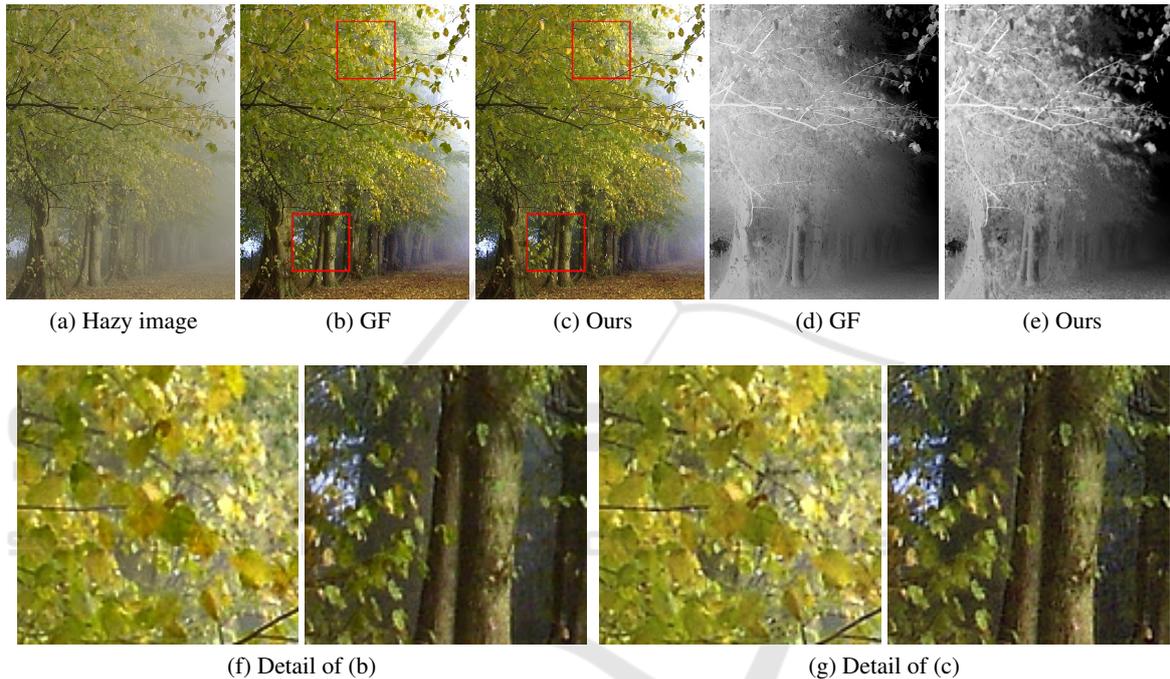


Figure 7: Haze removing. The bottom row images represent transition maps of (b) and (c). The local patch size for high-dimensional image is  $5 \times 5$ . The parameters for GF and HGF are  $r = 20$ ,  $\varepsilon = 10^{-4}$ .

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 (d)). By contrast, HGF can suppress the noises while the detailed regions are transferred as shown in Fig. 1 (f).

We also show the results of alpha matting in Fig. 5. The used alpha masks are shown in Figs. 1 (c)-(f). The result of guided image filtering has noises and color mixtures near the object boundary. The result of non-local means filtering has the blurred edges. These problems are solved in HGF. In our method,

the noises and color mixtures are reduced, the blur is not caused.

Figure 6 shows the image abstraction results. Note that the result takes 3 times iterations of filtering. As shown in Figs. 6 (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. 6 (c) and (e)).

HGF also has an excellent performance for haze removing (He et al., 2009). The haze removing results and the transition maps are shown in Fig. 7. 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

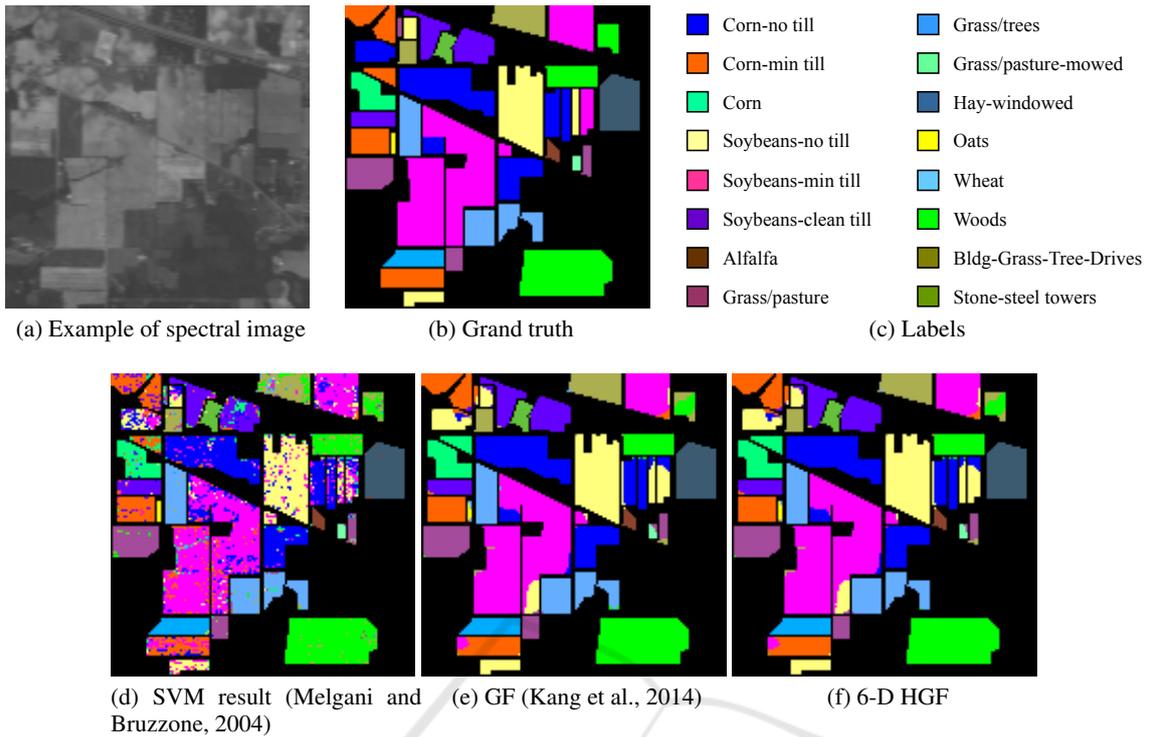


Figure 8: Classification result of Indian Pines image. The image of (a) represents a spectral image that the wavelength is  $0.7\mu\text{m}$ . The parameters for GF and HGF are  $r = 4$ ,  $\epsilon = 0.15^2$ .

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 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 (Melgani and Bruzzone, 2004), Kang et al. improved the accuracy of image classification by applying guided image filtering (Kang et al., 2014). They made a guidance image using PCA from the hyperspectral image, but most of the information was unused because 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 unused information.

Figure 8 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.

Table 1: Classification accuracy [%] of the classification results shown in Fig. 8.

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

In this paper, we proposed high-dimensional guided image filtering (HGF) by extending guided image filtering (He et al., 2010). Due to this extension, guided image filtering obtains the robustness for unexpected noises such as Gaussian noises and multiple textures.

Also our method enable the guided image filter to apply for a high-dimensional image such as a hyper-spectral image. HGF has a limitation that the computational cost becomes high by increasing the number of dimensions. For this reason, we also introduce the dimensionality reduction technique for efficient computing. Experimental results showed that HGF can work robustly in noisy regions and transfer detailed regions. In addition, we can compute efficiently by using the dimensionality reduction technique.

We construct the high-dimensional guidance image from the square neighborhood in each pixel. Therefore, as our future work, we consider the investigation of the generating method for high-dimensional guidance image.

## ACKNOWLEDGEMENT

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

## 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 Trans. on Graphics*, 28(3).
- Bae, S., Paris, S., and Durand, F. (2006). Two-scale tone management for photographic look. *ACM Trans. on Graphics*, 25(3):637–645.
- Buades, A., Coll, B., and Morel, J. M. (2005). A non-local algorithm for image denoising. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Crow, F. C. (1984). Summed-area tables for texture mapping. In *Proc. ACM SIGGRAPH*, pages 207–212.
- Durand, F. and Dorsey, J. (2002). Fast bilateral filtering for the display of high-dynamic-range images. *ACM Trans. on Graphics*, 21(3):257–266.
- Eisemann, E. and Durand, F. (2004). Flash photography enhancement via intrinsic relighting. *ACM Trans. on Graphics*, 23(3):673–678.
- Fattal, R., Agrawala, M., and Rusinkiewicz, S. (2007). Multiscale shape and detail enhancement from multi-light image collections. *ACM Trans. on Graphics*, 26(3).
- Fujita, S., Fukushima, N., Kimura, M., and Ishibashi, Y. (2015). Randomized redundant dct: Efficient denoising by using random subsampling of dct patches. In *Proc. ACM SIGGRAPH Asia Technical Briefs*.
- Fukushima, N., Fujita, S., and Ishibashi, Y. (2015). Switching dual kernels for separable edge-preserving filtering. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.
- Gastal, E. S. L. and Oliveira, M. M. (2011). Domain transform for edge-aware image and video processing. *ACM Trans. on Graphics*, 30(4).
- Gastal, E. S. L. and Oliveira, M. M. (2012). Adaptive manifolds for real-time high-dimensional filtering. *ACM Trans. on Graphics*, 31(4).
- He, K., Shun, J., and Tang, X. (2010). Guided image filtering. In *Proc. European Conference on Computer Vision (ECCV)*.
- He, K., Sun, J., and Tang, X. (2009). Single image haze removal using dark channel prior. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Hosni, A., Rhemann, C., Bleyer, M., Rother, C., and Gelautz, M. (2013). Fast cost-volume filtering for visual vorrespondence and beyond. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 35(2):504–511.
- Kang, X., Li, S., and Benediktsson, J. (2014). Spectral-spatial hyperspectral image classification with edge-preserving filtering. *IEEE Trans. on Geoscience and Remote Sensing*, 52(5):2666–2677.
- Kopf, J., Cohen, M., Lischinski, D., and Uyttendaele, M. (2007). Joint bilateral upsampling. *ACM Trans. on Graphics*, 26(3).
- Melgani, F. and Bruzzone, L. (2004). Classification of hyperspectral remote sensing images with support vector machines. *IEEE Trans. on Geoscience and Remote Sensing*, 42(8):1778–1790.
- 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.
- Petschnigg, G., Agrawala, M., Hoppe, H., Szeliski, R., Cohen, M., and Toyama, K. (2004). Digital photography with flash and no-flash image pairs. *ACM Trans. on Graphics*, 23(3):664–672.
- Pham, T. Q. and Vliet, L. J. V. (2005). Separable bilateral filtering for fast video preprocessing. In *Proc. IEEE International Conference on Multimedia and Expo (ICME)*.
- Porikli, F. (2008). Constant time  $o(1)$  bilateral filtering. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Tasdizen, T. (2008). Principal components for non-local means image denoising. In *Proc. IEEE International Conference on Image Processing (ICIP)*.
- Tomasi, C. and Manduchi, R. (1998). Bilateral filtering for gray and color images. In *Proc. IEEE International Conference on Computer Vision (ICCV)*.
- Yang, Q. (2012). Recursive bilateral filtering. In *Proc. European Conference on Computer Vision (ECCV)*.
- 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)*.