SVM for Edge-Preserving Filtering*

Qingxiong Yang Shengnan Wang Narendra Ahuja
University of Illinois at Urbana Champaign
http://vision.ai.uiuc.edu/~qyang6/

Abstract

In this paper, we propose a new method to construct an edge-preserving filter which has very similar response to the bilateral filter. The bilateral filter is a normalized convolution in which the weighting for each pixel is determined by the spatial distance from the center pixel and its relative difference in intensity range. The spatial and range weighting functions are typically Gaussian in the literature. In this paper, we cast the filtering problem as a vector-mapping approximation and solve it using a support vector machine (SVM). Each pixel will be represented as a feature vector comprising of the exponentiation of the pixel intensity, the corresponding spatial filtered response, and their products. The mapping function is learned via ϵ-SVM regression using the feature vectors and the corresponding bilateral filtered values from the training image. The major computation involved is the computation of the spatial filtered responses of the exponentiation of the original image which is invariant to the filter size given that an IIR $O(1)$ solution is available for the spatial filtering kernel. To our knowledge, this is the first learning-based $O(1)$ bilateral filtering method. Unlike previous $O(1)$ methods, our method is valid for both low and high range variance Gaussian and the computational complexity is independent of the range variance value. Our method is also the fastest $O(1)$ bilateral filtering yet developed. Besides, our method allows varying range variance values, based on which we propose a new bilateral filtering method avoiding the over-smoothing or under-smoothing artifacts in traditional bilateral filter.

1. Introduction

The objective of bilateral filtering is to smooth images. It is done by replacing the intensity (color) value of a pixel by the average of the values of other pixels weighted by their spatial distance and intensity similarity to the original pixel. Zucker and Rosenfeld [37] used this idea by identifying similar pixels by first detecting edges. They iteratively replaced the intensity of a pixel by the average of all the pixels in a small (3×3) neighborhood, and on the same side of the edge as the pixel itself. Davis [8] identified similar pixels differently, by choosing those six of the nine pixels in the neighborhood that are closest in intensity to the original pixel, and used their median to obtain the smoothed value. Ahuja [1] proposed a transform to compute the net similarity between a pixel and all other pixels in the image, as well as the direction in which the largest number of a pixel's most similar pixels are located. The latter was captured by computing a force vector at the pixel. Tabb and Ahuja [22] presented a detailed algorithm for multiscale image segmentation using the force transform and demonstrated the performance advantages of the similarity measure incorporated in the force transform. Tomasi and Manduchi [23] used the same definition of similarity as proposed by Ahuja [1] and used it for image smoothing. They replaced pixel values with similarity-weighted averages and called it bilateral filtering. Other than image smoothing and segmentation, bilateral filtering has found many other applications including denoising [4, 35, 27, 3, 2], texture editing and relighting [14], tone management [10, 17], demosaicking [20], stylization [26], optical-flow estimation [28, 21], stereo matching [36, 32, 29, 31].

Bilateral filter is known to be computationally intensive. Recently, several methods [10, 11, 18, 25] enable it to be computed at either $O(r)$ or $O(log(r))$ runtime in the radius of the filter r. By filtering on the down-sampled image, Paris and Durand [16] prove that the runtime of the Durand and Dorsey's method [10] decreases as the filter size increases because the down-sampling factor can be increased without significantly impacting the accuracy of the result. This method is relatively slow when the filter size is small. Chen *et al.* [7] later show that the GPU implementation of [10] can achieve video rate.

Two $O(1)$ bilateral filtering methods have been proposed recently. Porikli [19] presents three types of $O(1)$ bilateral filters. (1) Box spatial and arbitrary range kernels. The problem with this method is that the spatial variances are not taken into account. Box-like artifacts may appear due

*The support of HP Labs under an Innovation Research Award is gratefully acknowledged.

to the imperfect frequency response of the spatial box filter. (2) Arbitrary spatial and polynomial range kernels. This method has not yet been demonstrated to be useful in practice due to poor to preserve edge. (3) Arbitrary spatial and Gaussian range kernels. Taylor series is used to approximate the Gaussian range function up to the fourth order derivatives. However, this method is invalid for small Gaussian variances due to the limited approximation. In this paper, we only compare the third method proposed in [19] since Gaussian range function is typically used in the literature. Yang *et al.* [30] show that using recursive Gaussian [9], Durand and Dorsey's method [10] takes constant time by decomposing bilateral filter into a number of constant time spatial filters (defined as Principle Bilateral Filtered Image Component (PBFIC) in the paper). A $O(1)$ median filtering method is proposed based on the $O(1)$ bilateral filter in the paper. The computational complexity of the bilateral filter is independent of the filter size but depends on the number of PBFIC used. The smaller the range variance value, the larger number of PBFIC required, and the slower the method.

In this paper, we model bilateral filtering problem as a vector-mapping approximation and solve it using SVM regression. To our knowledge, this is the first learning-based bilateral filtering method. There are two steps involved in our method: training and predicting. The training step is processed off-line, in which an original image and the corresponding bilateral filtered image are used as input. We first apply Gaussian smoothing to the exponentiation (including exponent=0) of the original image. For each pixel, a corresponding feature vector is then constructed using the exponentiation of the image, the Gaussian filtered responses, and their products. The target value of the feature vector is the corresponding pixel value in the bilateral filtered image. The prediction step is processed on-line using the model produced in the training step. Each pixel is processed independently which enables parallel implementation. The computational complexity of the prediction step is invariant to the filter kernel size, *i.e.*, it is $O(1)$ or constant time.

Compared with the other two $O(1)$ bilateral filtering methods [19, 30] , our method has the following advantages:

1. Most of the bilateral filter based applications require edge-preserving filtering in which low range variance Gaussian is indispensable. However, [19] is invalid for low range variance, and the runtime of [30] increases as the range variance decreases. Our method is valid for both high and low range variances and the runtime is independent of the range variance value.

2. Traditional bilateral filtering methods use the same range variance value across the image while our method can smooth the image in constant time with varying range variance values for every pixel. Based on this property, we propose a new bilateral filtering method (Sec. 2.1) avoiding the over-smoothing or under-smoothing artifacts in traditional bilateral filters.

3. To our knowledge, our method is the most efficient $O(1)$ bilateral filtering method yet developed. For a 1 MB grayscale image, the speed of our GPU implementation is about 473 frames per second on an Nvidia Geforce 8800GTX GPU. The computational complexity of our method is about half of [19]. [30] exhibits quantization artifacts for low range variance Gaussian with the same computation complexity.

Additionally, we quantitatively evaluate our method and the other two $O(1)$ methods presented in [19] and [30] using 3638 images from six categories [13] (Sec. 3) and 485 images from two video sequences [5] (Sec. 4). This is different from the evaluation in [19] and [30], in which only a couple of images are tested. We believe that such a careful comparison is important to be clarified to the community since bilateral filtering has so many applications.

2. Approach

In this section, we present the details of our learning-based bilateral filtering method. The bilateral filter is a normalized convolution in which the weighting for each pixel q is determined by the spatial distance from the center pixel p, as well as its relative difference in intensity. Let I_p be the intensity at pixel p and J_p be the filtered value,

$$J_p = \sum_{q\in\Omega} \mathcal{F}(p,q)\mathcal{G}(I_p,I_q)I_q / \sum_{q\in\Omega} \mathcal{F}(p,q)\mathcal{G}(I_p,I_q). \quad (1)$$

The spatial and range weighting functions $\mathcal{F}$ and $\mathcal{G}$ are often Gaussian in the literature [1, 23, 10]

$$\mathcal{F}(p,q) = exp(-(p-q)^2/(2\sigma_S^2)), \quad (2)$$
$$\mathcal{G}(I_p,I_q) = exp(-(I_p-I_q)^2/(2\sigma_R^2)), \quad (3)$$

where σ_S and σ_R are the spatial and range variances, respectively. In this paper, we also confine the two weighting functions to be Gaussian.

Figure 1 (b) shows the bilateral filtering result J of the original image I in (a), while (c) is obtained by setting the range weighting function $\mathcal{G}$ to be a constant, which is the Gaussian spatial filtered response $G(I)$ of the original image I in (a). (e) is obtained by combining (a) and (c) using the blending map B in (d), that is

$$J_p^B = B_p \cdot I_p + (1-B_p)\cdot G(I)_p, \quad (4)$$

where p is a pixel in the image. As can be seen, the noise in (a) is greatly reduced in both (b) and (e) while the intensity

(a) Original: I. (b) BF: J. (c) $G(I)$. (d) B. Map: B. (e) Blended: J^B.

Figure 1. Bilateral filtering and image blending. From left to right: original image, bilateral filtered image, Gaussian filtered image, blending map, and blended image. As can be seen, the noise in (a) is greatly reduced in (b) and (e) while the intensity edge is well-preserved. Note: the reader is urged to view these images at full size (224×300), for details may be lost in hard copy.

edges are well-preserved. The PSNR [16] value computed from (b) and (e) is 45 dB, which demonstrate that (b) and (e) are numerically similar to each other (PNSR$>$ 40 dB often corresponds to almost invisible differences as suggested in [16]). This experiment shows that image blending with a carefully-designed blending map can be used to approximate bilateral filtering for simple scenes, and the blending process can be treated as a vector mapping process which maps a 2-dimentional vector $[I_p, G(I)_p]^T$ to a target value J_p^B.

However, for more complex scenes, for instance, the portrait in Figure 2 (a), it is hard to find a good blending map. Additionally, a simple combination between the original image (a) and the corresponding Gaussian filtered image (b) may not fit the bilateral filtered response correctly.

In this paper, we also use the Gaussian filtered responses of the exponentiation (powers of pixel-wise intensities in the image) of the original image as shown from Figure 2 (b) to (e). [1] Similar to the image blending problem which can be cast as a 2D vector-mapping problem, we formulate the new problem as a N-dimensional vector-mapping problem. That is from each pixel, a N-dimensional vector comprised of the values of exponentiation of the original image, the corresponding Gaussian filtered responses (as shown in Figure 2 (b) to (e)), and their products is obtained. Assuming that the exponent is up to n, $N = (n+1)^2 - 1$. For instance, assume $n = 2$, the feature vector of pixel p is a 8-dimensional vector: $\mathbf{x}_p = [I_p, I_p^2, G(I)_p, G(I^2)_p, I_pG(I)_p, I_p^2G(I)_p, I_pG(I^2)_p, I_p^2G(I^2)_p]^T$. The N-dimensional feature vector is then mapped to the bilateral filtered value. In this paper, the $\mathbb{R}^N$ to $\mathbb{R}^1$ mapping is denoted $\mathcal{M}$, that is: $\mathbb{R}^N \xrightarrow{\mathcal{M}} \mathbb{R}^1$. In practice, the mapping function $\mathcal{M}$ is unknown. In this paper, we learn a robust mapping function from ϵ-Support Vector Regression [24]. Given a set of data points, $\{(\mathbf{x}_1, z_1); \cdots; (\mathbf{x}_l, z_l)\}$, such that $\mathbf{x}_p \in \mathbb{R}^N$ is an input and $z_p \in \mathbb{R}^1$ is a target output, the standard form of support vector regression [24] is used to solve the following optimization problem

$$\begin{array}{ll} \min_{\mathbf{w},b,\xi,\xi^*} & 0.5\mathbf{w}^T\mathbf{w} + C\sum_{p=1}^{l}(\xi_p + \xi_p^*) \\ \text{subject to} & \mathrm{w}^T\phi(\mathbf{x}_p) + b - z_p \leq \epsilon + \xi_p \\ & z_p - \mathrm{w}^T\phi(\mathbf{x}_p) - b \leq \epsilon + \xi_p^* \\ & \xi_p, \xi_p^* \geq 0, p = 1, \cdots, l. \end{array}$$

where ξ_p and ξ_p^* are slack variables that control the upper error bound, and C is a constant penalty factor to penalize data point $(\mathbf{x}_p, z_p)$ that do not satisfy $\mathrm{w}^T\phi(\mathbf{x}_p) + b - z_p \leq \epsilon$. Predictions within the error bound ϵ of the true target are not penalized. For all our experiments, $C = 10$ and $\epsilon = 0.05$, and the SVM-library [6] implementation with linear basis function kernels are used for speed purpose.

2.1. $O(1)$ Bilateral Filter with Non-Uniform Range Variance

Traditional bilateral filtering methods depend on two fixed parameters (spatial and range variances: σ_S and σ_R) that indicate the size of the filter and the contrast of the features to preserve. However, fixing range variance σ_R results in either under-smoothing or over-smoothing artifacts as shown in Figure 3 (a) and (b), respectively. As can be seen, the unwanted wrinkles are not removed in (a) while the details around lip, eyes, and hair are lost in (b). Unlike previous methods, we fix the spatial variance σ_S but use varying range variance value σ_R for each pixel. We normalize the Gaussian filtered response of original image (Figure 2 (b)), such that the maximum value is a pre-defined value, and then use it as the range variance map. Note that our method is capable of filtering the image in constant time with non-uniform range variance values across the image by training different SVMs for each desired σ_R, thus can be di-

[1] The use of the powers of the image intensity is based on Taylor expansion approximation presented in [19].

(a) Original: I. (b) $G(I)$. (c) $G(I^2)$. (d) $G(I^3)$. (e) $G(I^4)$.

Figure 2. Visual comparison of Gaussian filtered responses of the exponentiation of the original image.

(a) Exact BF ($\sigma_R = 0.05$). (b) Exact BF ($\sigma_R = 0.3$). (c) Our method.

Figure 3. $O(1)$ bilateral filter with non-uniform range variance. From left to right: filtered image of Figure 2 (a) using the exact bilateral filter with uniform range variance $\sigma_R = 0.05$, filtered image using the exact bilateral filter with uniform range variance $\sigma_R = 0.3$, filtered image using our non-uniform bilateral filtering method by setting the maximum range variance to 0.3. As can be seen, our method improves the lip and hair details. Note: the reader is urged to view these images at full size (512×683), for details may be lost in hard copy.

rectly integrated with range variance map for non-uniform edge-preserving smoothing. The filtered image using our method with range variance $\sigma_R \in [0.01, 0.30]$ is presented in Figure 3 (c). That is a total of 30 SVR functions corresponding to 30 σ_R values were trained, and at each pixel location, one of these SVR functions will be selected and used. Apparently, using our method, the wrinkles around the eyes are removed and the details of the lip, eyes, and

hair are preserved.

2.2. Computational Complexity

The main computation involved in our method is the Gaussian filtering of the exponentiation of the original image. Figure 4 presents the numerical analysis of the behavior of our method with exponents from 2 to 5 *w.r.t* the range variance using PSNR value. Figure 2 (a) is used as the training and testing image. For exponents less than 5, the computational complexity of our method is about half of Porikli's method [19] with Taylor series approximation up to 3rd derivative, because at most four-pass Gaussian filtering of the exponentiation of the original image (exponent equal to 2 to 5) is required in our method but seven passes in [19]. The computational complexity of the method in [30] is linear to the number of PBFIC used. [30] can be as efficient as our method if only two PBFIC (requires four-pass Gaussian filtering) are used. However, using only two PBFIC results in visible quantization artifacts as will be shown in Sec. 3 and 4.

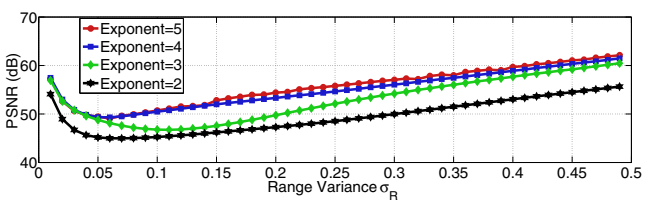


Figure 4. Quantitative evaluation of our method with exponents from 2 to 5 *w.r.t* the range variance using PSNR value. Note that the behavior of our method converges for exponents larger than 3.

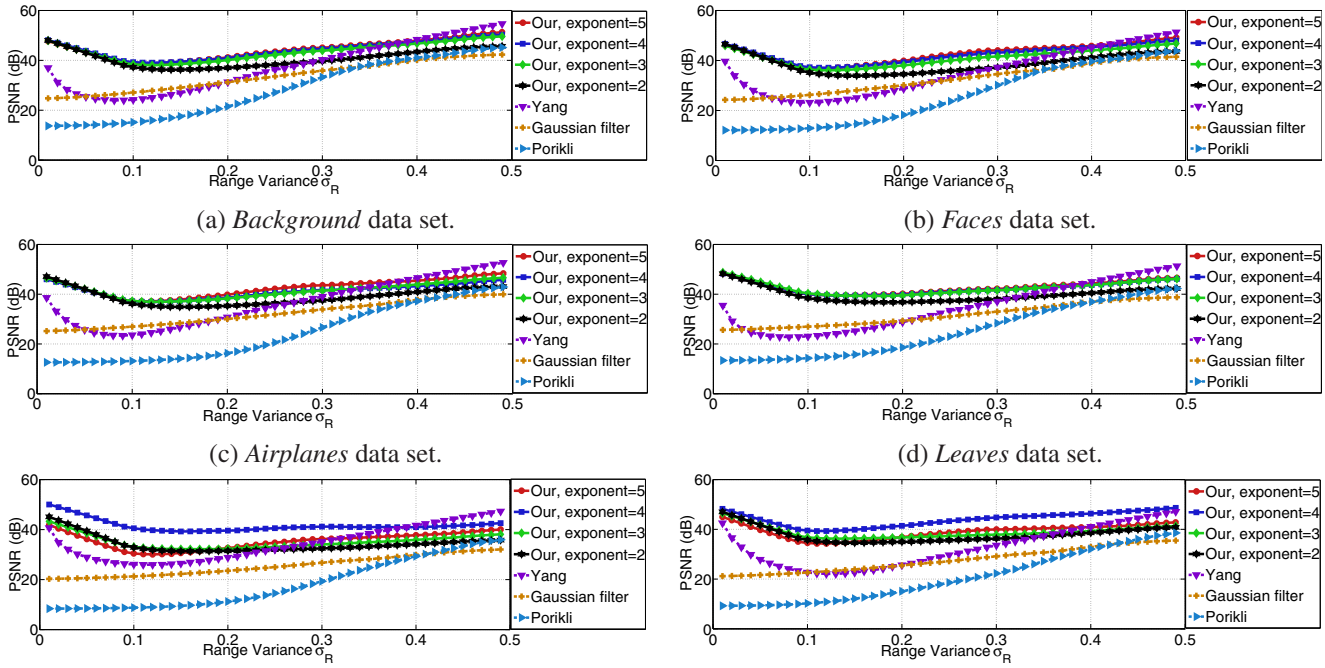
3. Experiments

In this section, we quantitatively compare our learning-based bilateral filtering method with the exact bilateral filtering using Caltech dataset [13]. All the experiments conducted in this section and Sec. 4 use uniform range variance value for standard comparison purpose. A model is first obtained via SVM training using one of the 450 images in the *background* category in [13], and then tested on a total of 3638 images from six categories in [13]. The mean PSNR values for different categories in [13] are presented in Figure 5. As discussed in Sec. 2.2, up to the 3rd derivative Taylor series expansion is used for Porikli's method [19] and two PBFICs are used for Yang's method [30]. As can be seen in Figure 5, our method outperforms the other methods for low range variance values, where Porikli's method is invalid and Yang's method has quantization errors. For high range variance values, Gaussian filtered responses (orange curves) are very similar to bilateral filtered responses, which degrades the importance of all $O(1)$ bilateral filtering methods. Note that the accuracy of our method generally increases as the exponent increases. However, for exponents up to 5, our method has the over-fitting problem for the *motorcycles* and *cars* categories as shown in the red curves in Figure 5 (e) and (f). The sensitivity of our method to the choice of training image is relatively low as proved in Fig. 5. The same training image is used for six different categories data sets containing 3638 different images. The blue curves in (d-f) show that the PSNR accuracy of our method with exponent= 4 is generally larger than 40 dB. It is assumed [16] the PSNR values above 40 dB often corresponds to almost invisible differences. The PSNR values in (a-c) are relatively lower than (d-f) but still lager than 37 dB.

Note that it is safe to use a down-sampled version of the original image for computing the Gaussian filtered responses of its powers with almost no quality loss if both down-sampled versions are used for training and predicting. All the experiments presented in the paper use nearest-neighbor down-sampling. The running time for the CPU and GPU implementations of all the $O(1)$ methods are presented in Table 1. Note that for a 1MB image, the speed of our fastest GPU implementation is about 473 frame per second on an Nvidia Geforce 8800GTX GPU, which is much faster than recursive Gaussian (122 fps) as can be seen in Table 1. 32-bit floating point textures and global memory are used for the GPU implementations of our method and the recursive Gaussian filtering.

4. Natural Video Conferencing

As shown in [30], edge-preserving-smoothing methods provide a way to retain the salient features in HD images while removing unwanted details and noise for modern video conference system, e.g., Halo [12]. Our method is especially suitable for this application. As shown in Figure 4, if the testing image is the same as the training image, there is no visible difference between the results obtained using our method and exact bilateral filter. For this application, the training step is processed at the beginning of the conference, that is, the first frame will be used for training. The model obtained is then used to smooth the rest of the video. Because during a video conference, the content of the video streams will not change dramatically, the filtering results using our method and the exact bilateral filter is very similar as shown in Figure 6. The visual evaluation is presented in Figure 7. The spatial variance σ_S is 0.03 and the range variance σ_R is 0.15 in this experiment. But of course, the purpose of the filtering is edge-preserving smoothing but not obtaining exact response as bilateral filtering. The video demos provided in [33] and [34] are obtained with **non-uniform range variance values (Sec. 2.1)**. The video demos demonstrate that our method is very suitable for natural video conferencing even if the response of our method



(a) *Background* data set. (b) *Faces* data set.

(c) *Airplanes* data set. (d) *Leaves* data set.

(e) *Motorcycles* data set. (f) *Cars* data set.

Figure 5. PSNR accuracy *w.r.t* the range variance for the Caltech data sets [13]. The training process is performed using one of the 450 images in the *background* data set. The trained model is then tested on the 450 *background* images in (a), 450 *face* images in (b), 1074 *airplane* images in (c), 186 *leave* images in (d), 826 *motorcycle* images in (e), and 652 *car* images in (f), respectively. Obviously, our learning-based filtering method has the most similar response as the exact bilateral filter on average. Also note that the response of Gaussian filter is very similar to bilateral filter for large range variances, which degrades the importance of all $O(1)$ bilateral filtering methods with large range variance.

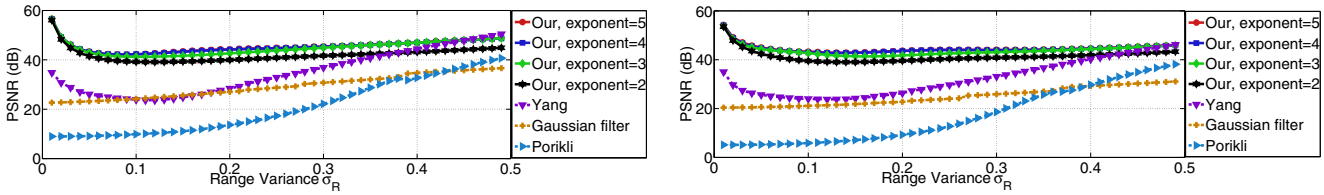
	Recursive Gaussian	Porikli [19]	Yang [30]	Our			
				exponent = 2	exponent = 3	exponent = 4	exponent = 5
CPU	5	3	1	12	8	4	3
GPU	122	NA	120	473	308	222	166

Table 1. Quantitative comparison of the speed (frame per second) of the $O(1)$ bilateral filtering methods. The speed of the CPU implementation of [19] and the GPU implementation of [30] is taken from the papers wherever it was mentioned, and the speed of the CPU implementation of [30] is tested using the source code provided on the author's website. [19] is tested with a P4 3.2 GHz processor, and the others with a 1.8 GHz Intel Core 2 Duo processor. Speed on different processor is compared here because our implementation of the method presented in [19] is much slower than the reported number, and we don't have a P4 3.2 GHz processor. Nevertheless, the performance of the two processors is similar for single-threaded applications [15]. The GPU implementations of recursive Gaussian, [30] and ours are tested on an Nvidia Geforce 8800GTX GPU. 32-bit floating point textures and global memory are used for the GPU implementations of our method and the recursive Gaussian filtering, and [30] uses 8-bit integer which is about $4\times$ faster. Apparently, our method is the fastest method. As will be discussed in the next section, even with small exponent, our method is still suitable for a number of applications requiring edge-preserving smoothing.

is not exactly the same as bilateral filter.

5. Conclusion

We describe a new $O(1)$ edge-preserving-smoothing method in the paper. As far as our knowledge goes, it is the first learning-based method, and is the most efficient method reported so far. Our method overcome the common difficulties encountered in the previous $O(1)$ methods, for instance, either invalid for low range variance Gaussian or the computation time depends on the range variance value. Additionally, we propose a new bilateral filtering method avoiding the over-smoothing or under-smoothing artifacts in traditional bilateral filter by allowing varying range variance values crossing the image. We show that our learning-based filtering method can be directly adapted to the new bilateral filtering method since its computation complexity



(a) IU data set. (b) MS data set.

Figure 6. PSNR accuracy *w.r.t* the range variance for two of *Microsoft i2i* video sequences [5]. The training process is performed using the first frame from each video. The obtained model is then tested for the rest of the video. Our learning-based filtering method has very similar responses as the exact bilateral filter as shown in the figure even with small exponent values. Visual comparison of the methods is provided in Figure 1 in the **supplementary material** due to page limit.

is independent of the range variance value.

References

[1] N. Ahuja. A transform for multiscale image segmentation by integrated edge and region detection. *PAMI*, 18:1211–1235, 1996.

[2] E. P. Bennett, J. L. Mason, and L. McMillan. Multispectral bilateral video fusion. In *Transactions on Image Processin*, volume 16, pages 1185–1194, 2007.

[3] E. P. Bennett and L. McMillan. Video enhancement using per-pixel virtual exposures. In *Siggraph*, volume 24, pages 845–852, 2005.

[4] A. Buades, B. Coll, and J.-M. Morel. A review of image denoising algorithms, with a new one. In *Multiscale Modeling and Simulation*, volume 4, pages 490–530, 2005.

[5] M. R. Cambridge. Microsoft i2i dataset. http://research.microsoft.com/en-us/projects/i2i/data.aspx.

[6] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[7] J. Chen, S. Paris, and F. Durand. Real-time edge-aware image processing with the bilateral grid. In *Siggraph*, volume 26, 2007.

[8] L. S. Davis and A. Rosenfeld. Noise cleaning by iterated local averaging. *IEEE Trans.Systems, Man, Cybernetics SMC*, 8:703–710, 1978.

[9] R. Deriche. Recursively implementing the gaussian and its derivatives. In *ICIP*, pages 263–267, 1992.

[10] F. Durand and J. Dorsey. Fast bilateral filtering for the display of high-dynamic-range images. In *Siggraph*, volume 21, 2002.

[11] M. Elad. On the bilateral filter and ways to improve it. *IEEE Transactions On Image Processing*, 11(10):1141–1151, 2002.

[12] Hewlett-Packard. Hp halo telepresence and video conferencing solutions. http://www.hp.com/halo.

[13] C. V. Lab. Caltech dataset. http://www.vision.caltech.edu/html-files/archive.html.

[14] B. M. Oh, M. Chen, J. Dorsey, and F. Durand. Image-based modeling and photo editing. In *Siggraph*, 2001.

[15] PADT. Ansys benchmark. http://www.padtinc.com/support/benchmark/bm05/.

[16] S. Paris and F. Durand. A fast approximation of the bilateral filter using a signal processing approach. In *ECCV*, 2006.

[17] G. Petschnigg, M. Agrawala, H. Hoppe, R. Szeliski, M. Cohen, and K. Toyama. Digital photography with flash and no-flash image pairs. In *Siggraph*, volume 23, 2004.

[18] T. Q. Pham and L. J. van Vliet. Separable bilateral filtering for fast video preprocessing. In *International Conference on Multimedia and Expo*, 2005.

[19] F. Porikli. Constant time o(1) bilateral filtering. In *CVPR*, 2008.

[20] R. Ramanath and W. E. Snyder. Adaptive demosaicking. In *Journal of Electronic Imaging*, volume 12, pages 633–642, 2003.

[21] P. Sand and S. Teller. Particle video: Long-range motion estimation using point trajectories. In *ECCV*, 2006.

[22] M. Tabb and N. Ahuja. Unsupervised multiscale image segmentation by integrated edge and region detection. *TIP*, 6:642–655, 1997.

[23] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *ICCV*, pages 839–846, 1998.

[24] V. N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, September 1998.

[25] B. Weiss. Fast median and bilateral filtering. In *Siggraph*, volume 25, pages 519–526, 2006.

[26] H. Winnemoller, S. C. Olsen, and B. Gooch. Real-time video abstraction. In *Siggraph*, volume 25, pages 1221–1226, 2006.

[27] W. C. K. Wong, A. C. S. Chung, and S. C. H. Yu. Trilateral filtering for biomedical images. In *International Symposium on Biomedical Imaging*, 2004.

[28] J. Xiao, H. Cheng, H. Sawhney, C. Rao, and M. Isnardi. Bilateral filtering-based optical flow estimation with occlusion detection. In *ECCV*, 2006.

[29] Q. Yang, C. Engels, and A. Akbarzadeh. Near real-time stereo for weakly-textured scenes. In *BMVC*, pages 80–87, 2008.

[30] Q. Yang, K.-H. Tan, and N. Ahuja. Real-time o(1) bilateral filtering. In *CVPR*, 2009.

[31] Q. Yang, L. Wang, and N. Ahuja. A constant-space belief propagation algorithm for stereo matching. In *CVPR*, 2010.

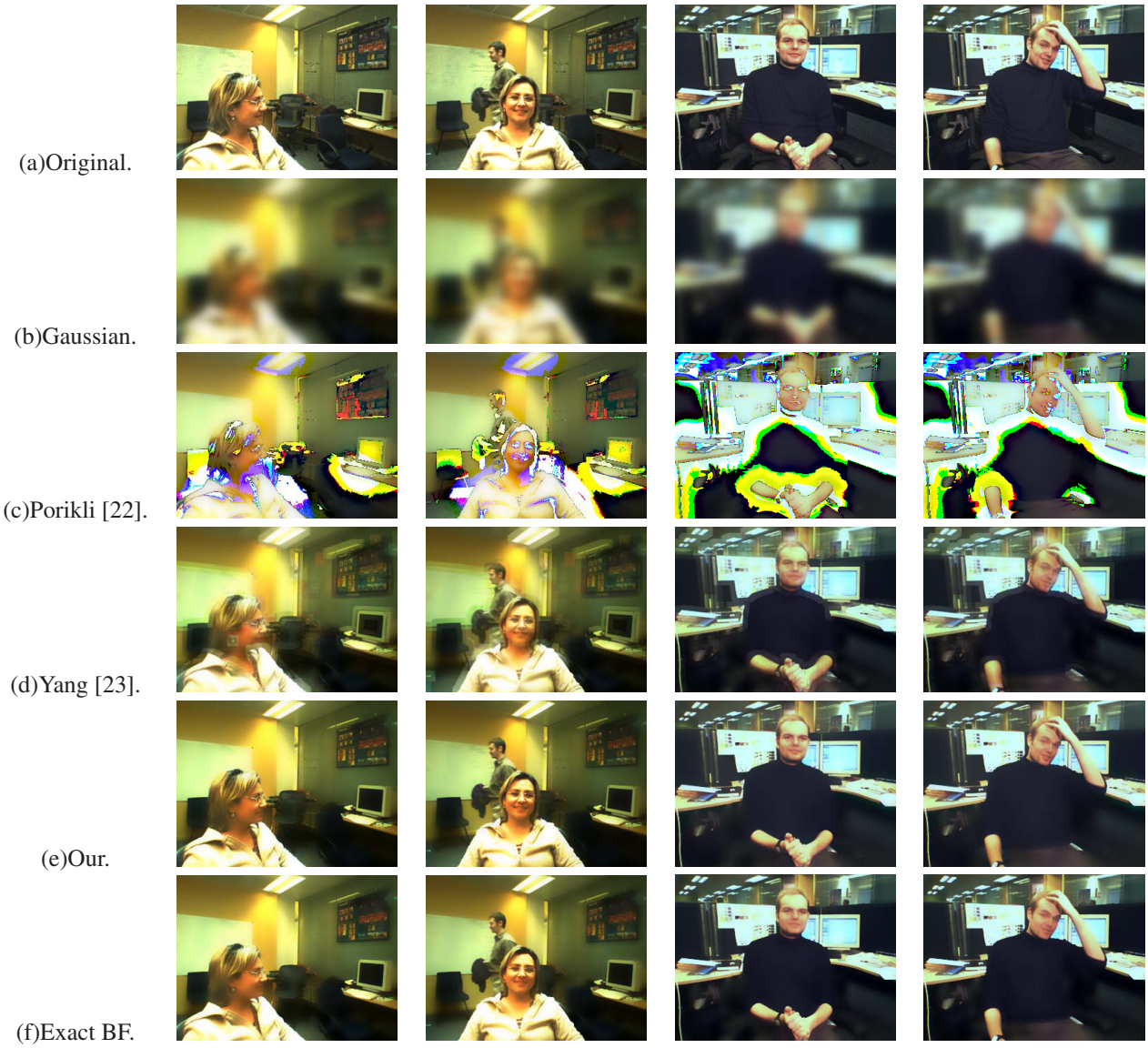


Figure 7. Visual comparison of the $O(1)$ bilateral filtering methods. From left to right: a single frame extracted from the *Microsoft i2i* video sequences [5], results obtained from Gaussian filtering, from Porikli's method [19], from Yang's method [30], from our method, and from the exact bilateral filtering. From up to bottom: the 1st frame (used for training) from *IU* data set, the 58th frame from *IU* data set, the 1st frame (used for training) from *MS* data set, and the 169th from *MS* data set. The spatial variance value is 0.03, and the range variance value is 0.15. As shown in (c) and (d), Porikli's method is invalid for such low range variance, and Yang's method encounters visible quantization artifacts around the color edges. Our method is more robust than Porikli and Yang's methods as shown in (e). The results obtained from our method is visiblely very similar to the exact bilateral filtering results in (f). Note: the reader is urged to view these images at full size (320×240), for details may be lost in hard copy.

[32] Q. Yang, L. Wang, R. Yang, H. Stewénius, and D. Nistér. Stereo matching with color-weighted correlation, hierarchical belief propagation and occlusion handling. *PAMI*, 2008.

[33] Q. Yang, S. Wang, and N. Ahuja. Experiments on iu dataset. http://vision.ai.uiuc.edu/˜qyang6/publications/videos/cvpr-10-svmbf/iu-dataset.avi.

[34] Q. Yang, S. Wang, and N. Ahuja. Experiments on ms dataset. http://vision.ai.uiuc.edu/˜qyang6/publications/videos/cvpr-10-svmbf/ms-dataset.avi.

[35] Q. Yang, R. Yang, J. Davis, and D. Nistér. Spatial-depth super resolution for range images. In *CVPR*, 2007.

[36] K.-J. Yoon and I.-S. Kweon. Adaptive support-weight approach for correspondence search. *PAMI*, 28(4):650–656, 2006.

[37] A. Zucker, S. Lev, and A. Rosenfeld. Iterative enhancement of noisy images. *IEEE Trans.Systems, Man, Cybernetics SMC*, 7:435–441, 1977.