Fast Weighted Histograms for Bilateral Filtering and Nearest Neighbor Searching

Shengfeng He, Qingxiong Yang, *Member, IEEE,* Rynson W.H. Lau, *Senior Member, IEEE,* and Ming-Hsuan Yang, *Senior Member, IEEE*

***Abstract*—The locality sensitive histogram (LSH) injects spatial information into the local histogram in an efficient manner, and has been demonstrated to be very effective for visual tracking. In this paper, we explore the application of this efficient histogram in two important problems. We first extend the LSH to linear time bilateral filtering, and then propose a new type of histogram for efficiently computing edge-preserving nearest neighbor fields (NNF). While existing histogram-based bilateral filtering methods are the state-of-the-art for efficient grayscale image processing, they are limited to box spatial filter kernels only. In our first application, we address this limitation by expressing the bilateral filter as a simple ratio of linear functions of the LSH, which is able to extend the box spatial kernel to an exponential kernel. The computational complexity of the proposed bilateral filter is linear in the number of image pixels. In our second application, we derive a new bilateral weighted histogram (BWH) for NNF. The new histogram maintains the efficiency of LSH, which allows approximate NNF to be computed independent of patch size. In addition, BWH takes into account both spatial and color information, and thus provides higher accuracy for histogram-based matching, especially around color edges.**

***Index Terms*—locality sensitive histograms, bilateral weighted histograms, bilateral filtering, nearest neighbor searching, edge-preserving smoothing.**

I. Introduction

A Local histogram contains a large amount of information and has been shown to be useful for variety applications in image processing, computer vision and computer graphics. Bilateral filtering is one of the applications which efficient implementations reside in the local histogram.

Bilateral filters [34], [27] are ubiquitous in vision, graphics and multimedia applications, including stereo vision [40], image denoising [8], [39] and tone management [12], [28], [3]. The main obstacle of applying bilateral filters to realtime applications is that the brute-force implementation is slow when the kernel is large. Several techniques have been proposed to improve the efficiency of bilateral filters.

Histogram-based bilateral filters [35], [31] are the state-of-the-art for processing grayscale images. While recent histogram-based bilateral filtering methods [35], [31] perform well, the adopted spatial filters are limited to box kernels. However, the box spatial filter kernel can only be effectively used in a number of applications as the spatial information is not exploited. For example, the bilateral filter can be used in colorization as shown in Figure 1. Nevertheless, the box spatial filter kernel has a finite impulse response. When used in propagating color information from user strokes to the rest of the image, the filter radius needs to be large, which may cause ambiguities around edges as shown in Figure 1b.

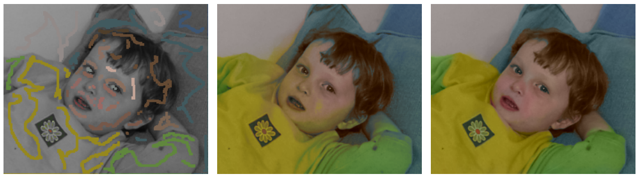

(a) Input (b) Box filter (c) LSH

Fig. 1: Colorization using a bilateral filter with different spatial kernels. (a) Input image and several color strokes. (b) Colorization result using a box filer kernel. (c) Colorization result using locality sensitive histogram.

On the other hand, the local histogram can also be applied to matching tasks [30]. Histogram matching is independent of patch size and can thus be performed very efficiently, especially for large patches. However, histograms throw away the structural information of the local region, they are not suitable for the content-aware tasks, which usually require high matching accuracy like image editing using PatchMatch [6].

In this paper, we explore the application of the local histogram in addressing the problems of bilateral filtering and nearest neighbor searching. First, a new bilateral filtering method is proposed to overcome the restriction on using a box spatial filter kernel in the histogram-based bilateral filter [35], [31] while maintaining efficiency. The proposed bilateral filtering algorithm is derived from our previous locality sensitive histogram (LSH), which was originally proposed for visual tracking [21]. LSH is computed at each pixel location and a suitable floating-point value is added to the corresponding bin for each occurrence of an intensity value. The floating-point value reduces exponentially with respect to the distance from the center point. We make use of the LSHs to find the bilateral convolution response

Shengfeng He, Qingxiong Yang, and Rynson W.H. Lau are with the Department of Computer Science, City University of Hong Kong, Hong Kong. E-mail: shengfeng_he@yahoo.com, {qiyang, rynson.lau, }@cityu.edu.hk

Ming-Hsuan Yang is with the School of Engineering, University of California, Merced, CA 95344. E-mail: mhyang@ucmerced.edu

and overcome the limitation of the state-of-the-art methods [31], [35], enabling an exponential kernel to be used as a spatial kernel. The computational complexity of the proposed bilateral filter is independent of the kernel size and linear in the number of image pixels. It is comparable to Porikli's method [31], which uses integral histograms. As each bin of the LSHs can be computed separately, the memory cost is independent of the number of bins but linear in the number of image pixels. Our method only requires three times the image memory (including the memory for the images before and after filtering). It can be extended to joint/cross bilateral filtering, but it will require four times the size of the image memory. It can also be directly extended to filter multi-dimensional signals. While the space complexity remains the same, the computational complexity will be exponential in dimensionality.

We further extend the LSH to a new bilateral weighted histogram (BWH), which takes into account both the spatial and color information to recover the structural information of a local histogram while maintaining the efficiency of LSH. Similar to LSH, BWH can be recursively implemented by introducing the recursive approximation of the range kernel. The proposed BWH is especially effective for nearest neighbor searching, and can be directly integrated with the efficient approximate nearest neighbor field (ANNF) algorithm, such as PatchMatch [6] (or other ANNF algorithms [25], [33]). Experiments on image reconstruction, optical flow, and colorization demonstrate that the proposed method achieves better performance than the original PatchMatch algorithm in terms of speed, accuracy and visual quality.

The rest of this paper is organized as follows. Section II reviews related works on bilateral filtering and ANNF, and briefly summarizes the locality sensitive histogram. Section III presents bilateral filtering and joint bilateral filtering using LSH. Section IV presents the bilateral weighted histogram for ANNF. Section V evaluates the proposed methods with a number of experiments. Finally, Section VI concludes this paper with remarks on our future work.

II. Related Work

In this section, we briefly review the most relevant works on bilateral filtering and PatchMatch. We also briefly summarizes the locality sensitive histogram. A comprehensive literature review on bilateral filtering can be found in [27].

A. Bilateral Filtering

A bilateral filter has two filter kernels, one spatial kernel and one range filter, for measuring radiometric distances between the center pixel and its neighbors for preserving sharp edges while reducing noise. The brute-force implementation of bilateral filters is slow when the kernel size is large and several techniques have been proposed to improve the efficiency.

Pham and van Vliet [29] propose to separate the 2D bilateral filter into two 1D bilateral filters and apply one after the other. Although this method is efficient, it generates good results only on uniform regions, horizontal edges and vertical edges. Durand and Dorsey [12] propose to linearize the bilateral filter by quantizing the range domain to a small number of discrete values. Paris and Durand [26] extend this method by representing the grayscale image with a volumetric grid, and bilateral filtering is then equivalent to convolving the grid with a 3D Gaussian filter. The use of bilateral grid helps improve the accuracy of the fast algorithm [12] when the spatial domain quantization is used. However, it entails significant memory requirement when the filter kernel is small. Yang et al. [38] further demonstrate that the fast algorithm [12] can be implemented using a recursive Gaussian filter such that its computational complexity is independent of the filter kernel size. In addition, Adams et al. propose to use the permutohedral lattice [1] and Gaussian KD-trees [2] for efficient high-dimensional Gaussian filtering, which can be directly integrated with the approximate algorithm [26] for fast bilateral filtering. Yang [36] proves that the bilateral filter can be recursively implemented without quantization; thus the computational complexity will be linear in both image size and dimensionality.

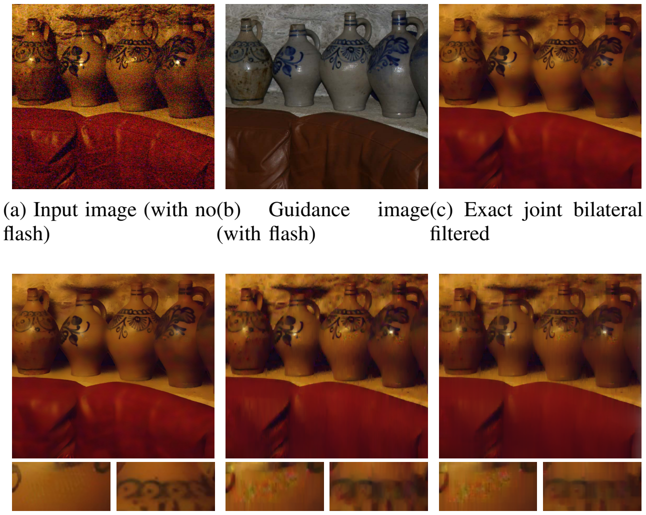

(a) Input image (with no flash)
(b) Guidance image (with flash)
(c) Exact joint bilateral filtered
(d) Ours joint bilateral filtered
(e) Joint filtered by [16]
(f) Joint filtered by [36]

Fig. 2: Joint filtering comparison. The edge-preserving filters like (e) [16] and (f) [36] are limited to particular applications, hence extra artifacts are introduced while applying to joint filtering. In contrast, the result of the proposed joint bilateral filtering (d) is very close to the ground truth (c).

Inspired by the bilateral filter, a number of edge-preserving filtering methods are proposed with low computational complexity recently, e.g., Fattal's EAW method [14], He et al.'s guided filtering method [18], and Gastal and Oliveira's domain transform filtering method [16]. Nevertheless, unlike regular bilateral filter that can have arbitrary spatial kernel and arbitrary range kernel, these fast bilateral filter approximations or efficient edge-preserving filters may not be adjusted for different applications. For example, [36] and [16] are not suitable for joint filtering as shown in Figure 2 and [18] may completely ignore the color differences between pixels in some special cases as demonstrated in [17].

Histogram-based bilateral filters [31], [35] have also been shown to be effective for processing grayscale images. The computation of intensity differences between the center pixel and its neighbors in a local patch can be independent of the kernel size as long as the intensity statistics are pre-computed. Weiss [35] shows that a bilateral filter with a box spatial filter kernel can be expressed as a simple ratio of two linear functions of a local histogram, which is computed based on the square neighborhoods of the input image. An efficient algorithm is then proposed to compute the local histograms, which has a computational complexity of $O(\log r)$ per pixel where r is the filter radius. Porikli [31] further reduces its complexity to $O(1)$ per pixel by showing that local histograms can be computed in constant time using the integral histogram [30]. However, the main limitation of histogram-based bilateral filtering is that the spatial kernel is limited to the box filter, which is addressed in this work.

B. Approximate Nearest Neighbor Fields

Patch-based methods have been shown to be very useful in numerous computer vision and graphics applications. The core of patch-based methods is to find ANNF that describes the corresponding patches between image I and J, where I and J can be the same image. KD-tree [15] and locality sensitive hashing [23] are two classical algorithms for ANNF. They both aim at reducing the query time by partitioning the space either deterministically or randomly. However, computing a dense and global corresponding map is too computational expensive for realtime applications.

Computing ANNF has been remarkably advanced by the PatchMatch algorithm [6], [7]. The main idea of PatchMatch is to utilize image coherence: the neighborhood of a good matched pair in image I will likely form good matched pairs with the neighborhood of the matched patch in image J. Initially, PatchMatch randomly assigns patches in image I to patches in image J. These random assignments may include good matches and bad matches. Only the good matches are propagated to nearby patches to provide good initial guesses and refine the matching results. This process is performed iteratively, and the algorithm converges after a small number of iterations. The PatchMatch algorithm allows computing ANNF in realtime, enabling lots of image editing applications like image completion and image reshuffling. PatchMatch is further improved in speed and accuracy through locality sensitive hashing [25] and KD-tree [33], respectively. However, the patch size is still the decisive factor of the computation overhead, especially for applications requiring large patches. In addition, the PatchMatch algorithm compares two patches using pixel-by-pixel differences, which leads to over-smoothing artifacts. These problems are addressed by integrating PatchMatch with the proposed BWH.

C. Locality Sensitive Histograms

The conventional image histogram is a 1D array where each value is an integer indicating the frequency of occurrence of a particular intensity value. In contrast, LSH [21] is constructed by floating point as it takes into account the spatial distance from the center pixel. Given an image $\mathbf{I}$, the LSH, $\mathbf{H}_p^E$, at pixel p is computed by:

$$\mathbf{H}_p^E(b) = \sum_{q=1}^{W} \alpha^{|p-q|} \cdot Q(\mathbf{I}_q, b), \quad b = 1, \ldots, B, \tag{1}$$

where W is the number of pixels, B is the number of bins, $Q(\mathbf{I}_q, b)$ is zero except when intensity value $\mathbf{I}_q$ (at pixel location q) belongs to bin b, and $\alpha \in (0, 1)$ is a parameter controlling the decreasing weight as a pixel moves away from the target center. LSH can be computed efficiently when the input image is 1D:

$$\mathbf{H}_p^E(b) = \mathbf{H}_p^{E,left}(b) + \mathbf{H}_p^{E,right}(b) - Q(\mathbf{I}_p, b), \tag{2}$$

where

$$\mathbf{H}_p^{E,left}(b) = Q(\mathbf{I}_p, b) + \alpha \cdot \mathbf{H}_{p-1}^{E,left}(b), \tag{3}$$
$$\mathbf{H}_p^{E,right}(b) = Q(\mathbf{I}_p, b) + \alpha \cdot \mathbf{H}_{p+1}^{E,right}(b). \tag{4}$$

Based on Eq. (3) and (4), pixels on the right of pixel p do not contribute to the LSH on the left hand side $\mathbf{H}_p^{E,left}$, while pixels on the left of pixel p do not contribute to the LSH on the right hand side $\mathbf{H}_p^{E,right}$. The summation of $\mathbf{H}_p^{E,left}$ and $\mathbf{H}_p^{E,right}$, however, combines the contribution from all pixels and the weight of the contribution drops exponentially with respect to the distance to pixel p. Clearly, only B multiplication and B addition operations are required at each pixel location in order to compute $\mathbf{H}_p^{E,left}$ (or $\mathbf{H}_p^{E,right}$). Thus, the computational complexity of the locality sensitive histogram is reduced to $O(B)$ per pixel. Eq. (2) can be easily extend to multi-dimensional images by simply performing the 1D algorithm separately and sequentially in each dimension. This efficient algorithm is used and extended throughout this paper.

III. Linear Time Bilateral Filtering

Porikli [31] shows that bilateral filtering can be accelerated by the local histograms. However, its main limitation is that the spatial kernel is limited to box filters. In this section, we show that the proposed LSH is able to address this problem.

The bilateral filter is an edge-preserving smoothing operator with a spatial filter kernel $\mathcal{F}(\cdot)$ and a range filter kernel $\mathcal{G}(\cdot)$. Let $\mathbf{I}$ be a grayscale image, and $\mathbf{I}_p$ and $\mathbf{I}_q$ be the intensity values of pixels p and q, respectively. The bilateral filtered value of pixel p is:

$$\mathbf{I}_p^I = \sum_{q=1}^{W} \mathcal{F}(p, q)\mathcal{G}(\mathbf{I}_p, \mathbf{I}_q)\mathbf{I}_q / \kappa_p, \tag{5}$$

where κ_p is a normalizing parameter defined as:

$$\kappa_p = \sum_{q=1}^{W} \mathcal{F}(p, q)\mathcal{G}(\mathbf{I}_p, \mathbf{I}_q), \tag{6}$$

such that $\sum_{q=1}^{W} \mathcal{F}(p,q)\mathcal{G}(\mathbf{I}_p,\mathbf{I}_q)/\kappa_p$ =1.

From Eq. (5), the computational complexity of the bilateral filter is $O(W^2)$ with a straightforward implementation. Using the locality sensitive histograms with B bins as described in Section II-C, the complexity can be reduced to $O(WB)$ because of the use of an exponential kernel in LSHs and the bilateral filter can be expressed as a simple ratio of two linear functions of the computed LSHs.

Let h_b be the intensity value corresponding to the b-th bin of the histograms. According to Eq. (6),

$$\kappa_p = \sum_{q=1}^{W} \mathcal{F}(p,q)(\sum_{b=1}^{B} Q(\mathbf{I}_q,b)\cdot\mathcal{G}(\mathbf{I}_p,h_b)). \tag{7}$$

Let

$$\mathbf{H}_p^E(b) = \sum_{q=1}^{W} \mathcal{F}(p,q)\cdot Q(\mathbf{I}_q,b) \tag{8}$$

which is actually the LSHs presented in Section II-C,

$$\kappa_p = \sum_{b=1}^{B} \mathbf{H}_p^E(b)\mathcal{G}(\mathbf{I}_p,h_b). \tag{9}$$

As shown in Section II-C, the spatial kernel $\mathcal{F}(p,q)$ here is an exponential kernel. If B is sufficiently large to cover all possible pixel intensity values. The spatial kernel $\mathcal{F}(p,q)$ is now embedded in the histograms. By substituting Eq. (9) into Eq. (5), we obtain:

$$\mathbf{I}_p^I = \frac{\sum_{b=1}^{B}\mathbf{H}_p^E(b)\mathcal{G}(\mathbf{I}_p,h_b)h_b}{\sum_{b=1}^{B}\mathbf{H}_p^E(b)\mathcal{G}(\mathbf{I}_p,h_b)}. \tag{10}$$

The computational complexity now depends only on the resolution of the LSHs, $\mathbf{H}_p^E$, which is $O(WB)$.

A. Linear Time Joint Bilateral Filtering

If the range filter kernel $\mathcal{G}(\cdot)$ in Eq. (5) is computed using an additional image $\mathbf{J}$, where the intensity values of pixels p and q are $\mathbf{J}_p$ and $\mathbf{J}_q$, respectively, the resulting filter becomes a joint/cross bilateral filter [13], [28]. The joint bilateral filtered value of pixel p is:

$$\mathbf{I}_p^J = \sum_{q=1}^{W} \mathcal{F}(p,q)\mathcal{G}(\mathbf{J}_p,\mathbf{J}_q)\mathbf{I}_q/\kappa_p^J, \tag{11}$$

where $\kappa_p^J = \sum_{q=1}^{W} \mathcal{F}(p,q)\mathcal{G}(\mathbf{J}_p,\mathbf{J}_q)$ is a normalization parameter. This joint bilateral filter can also be computed in $O(WB)$ with a slight change in the histograms. From Eq. (9):

$$\kappa_p^J = \sum_{b=1}^{B} \mathbf{H}_p^J(b)\mathcal{G}(\mathbf{J}_p,h_b), \tag{12}$$

where $\mathbf{H}_p^J$ are the LSHs computed from image $\mathbf{J}$:

$$\mathbf{H}_p^J(b) = \sum_{q=1}^{W} \mathcal{F}(p,q)\cdot Q(\mathbf{J}_q,b). \tag{13}$$

Using Eq. (11) and (12):

$$\begin{aligned}\kappa_p^J\mathbf{I}_p^J &= \sum_{q=1}^{W} \mathcal{F}(p,q)(\sum_{b=1}^{B} Q(\mathbf{J}_q,b)\cdot\mathcal{G}(\mathbf{J}_p,h_b))\mathbf{I}_q \\ &= \sum_{b=1}^{B}(\sum_{q=1}^{W}\mathcal{F}(p,q)\cdot Q(\mathbf{J}_q,b)\mathbf{I}_q)\cdot\mathcal{G}(\mathbf{J}_p,h_b). \qquad (14)\end{aligned}$$

Let

$$\mathbf{H}_p^K(b) = \sum_{q=1}^{W} \mathcal{F}(p,q)\cdot Q(\mathbf{J}_q,b)\mathbf{I}_q, \tag{15}$$

We then have:

$$\begin{aligned}\mathbf{I}_p^J &= \sum_{b=1}^{B}\mathbf{H}_p^K(b)\mathcal{G}(\mathbf{J}_p,h_b)/\kappa_p^J \\ &= \frac{\sum_{b=1}^{B}\mathbf{H}_p^K(b)\mathcal{G}(\mathbf{J}_p,h_b)}{\sum_{b=1}^{B}\mathbf{H}_p^J(b)\mathcal{G}(\mathbf{J}_p,h_b)}. \qquad (16)\end{aligned}$$

When used for joint bilateral filtering, a new B dimensional vector, $\mathbf{H}_p^K(b)$, in addition to the LSHs, $\mathbf{H}_p^J(b)$, needs to be computed at each pixel location. This doubles the memory cost, while the computation overhead is negligible.

B. Multi-Dimensional Filtering

The proposed bilateral filtering is not limited to grayscale images. For an N-channel image $\vec{\mathbf{J}}$, two N-dimensional histograms, which are extensions of the LSHs in Eq. (13) and (15), are computed at each pixel p as:

$$\begin{aligned}\mathbf{H}_p^{\vec{J}}(b_1,\cdots,b_N) &= \sum_{q=1}^{W}\mathcal{F}(p,q)\cdot\prod_{c=1}^{N}Q(\vec{\mathbf{J}}_q^c,b_c), \qquad (17)\\ \mathbf{H}_p^{\vec{K}}(b_1,\cdots,b_N) &= \sum_{q=1}^{W}\mathcal{F}(p,q)\cdot\prod_{c=1}^{N}Q(\vec{\mathbf{J}}_q^c,b_c)\mathbf{I}_q, \qquad (18)\end{aligned}$$

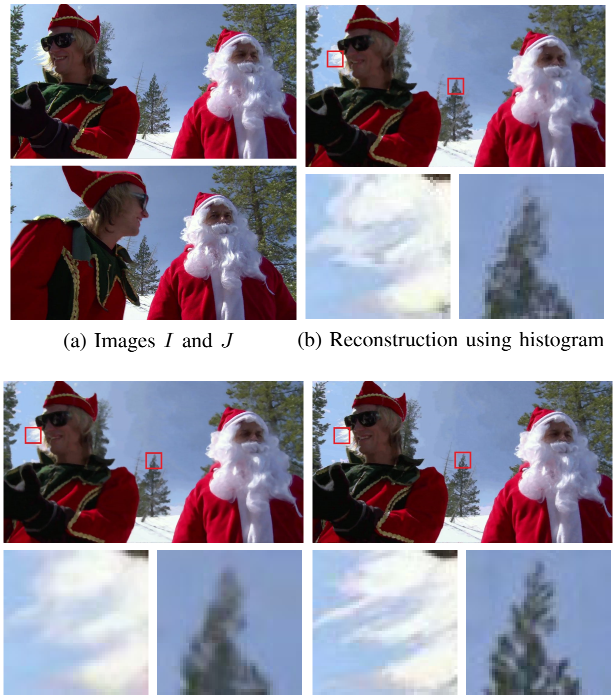
where $\vec{\mathbf{J}}^c$ is the c-th channel of image $\vec{\mathbf{J}}$. Let $\vec{h}_b = [h_{b_1},\cdots,h_{b_N}]^\top$ and $\vec{b} = [b_1,\cdots,b_N]^\top$. The joint bilateral filter in Eq. (16) can be updated using Eq. (17) and (18) for multi-dimensional filtering:

$$\begin{aligned}\mathbf{I}_p^{\vec{J}} &= \frac{\sum_{b_1=1}^{B}\cdots\sum_{b_N=1}^{B}\mathbf{H}_p^{\vec{K}}(\vec{b})\mathcal{G}(\vec{\mathbf{J}}_p,\vec{h}_b)}{\sum_{b_1=1}^{B}\cdots\sum_{b_N=1}^{B}\mathbf{H}_p^{\vec{J}}(\vec{b})\mathcal{G}(\vec{\mathbf{J}}_p,\vec{h}_b)} \\ &= \frac{\sum_{\vec{b}=[1,\cdots,1]^\top}^{[B,\cdots,B]^\top}\mathbf{H}_p^{\vec{K}}(\vec{b})\mathcal{G}(\vec{\mathbf{J}}_p,\vec{h}_b)}{\sum_{\vec{b}=[1,\cdots,1]^\top}^{[B,\cdots,B]^\top}\mathbf{H}_p^{\vec{J}}(\vec{b})\mathcal{G}(\vec{\mathbf{J}}_p,\vec{h}_b)}. \qquad (19)\end{aligned}$$

C. Complexity Analysis

According to Eq. (17) and (18), the computational complexity of constructing the locality sensitive histograms is $O(WBN)$, where W is the number of image pixels, B is the number of bins, and N is the number of channels. However, the complexity of the joint bilateral filter in Eq. (19) is exponential in N. Thus, the computational complexity of the whole algorithm is linear in the image size but exponential in dimensions.

The memory cost mainly resides in the locality sensitive histograms (and the additional histograms in the case of



(a) Images I and J

(b) Reconstruction using histogram

(c) Reconstruction using LSHs

(d) Reconstruction using BWHs

Fig. 3: Histogram matching examples. Image I is reconstructed from image J and the dense patch map from I to J. Traditional histogram in (b) is not suitable for content-aware applications. LSH in (c) obtains better matching results but still not distinct enough. The proposed BWH in (d) maintains the efficiency of histogram and achieves the best results.

joint bilateral filter), which is a $(2+N)$-dimensional array. The memory cost is huge for a straightforward implementation when N is large. Let

$$L_p^1 = \sum_{\vec{b}=[1,\cdots,1]^\top}^{[B,\cdots,B]^\top} \mathbf{H}_p^{\vec{K}}(\vec{b})\mathcal{G}(\vec{J}_p,\vec{h}_b), \tag{20}$$

$$L_p^2 = \sum_{\vec{b}=[1,\cdots,1]^\top}^{[B,\cdots,B]^\top} \mathbf{H}_p^{\vec{J}}(\vec{b})\mathcal{G}(\vec{J}_p,\vec{h}_b). \tag{21}$$

Eq. (19) becomes:

$$\mathbf{I}_p^{\vec{J}} = \frac{L_p^1}{L_p^2}. \tag{22}$$

In the above equations, L_p^i can be initialized with all zeros, and then accumulated w.r.t. each $\vec{b}$. As a result, it is not necessary to maintain all the histograms but only $\mathbf{H}_p^{\vec{J}}(\vec{b})$ and $\mathbf{H}_p^{\vec{K}}(\vec{b})$ at each pixel location p. The minimum memory requirement is thus twice the image size for joint bilateral filtering (or same as the image size for bilateral filtering), excluding the memory for the images before and after filtering.

IV. Bilateral Weighted Histograms for ANNF

In this section, we further propose a new locally weighted histogram with application to image matching. Patch-based methods have achieved great success in various applications. Comparing two patches is an inevitable step in these algorithms. The straightforward way is first represented a s-by-s patch in a $3s^2$-dimensional space, and the similarity is defined as the L_2 distance between two s^2 vectors. However, the computation overhead relies on the patch size, which is usually required to be large in order to eliminating matching ambiguities. On the other hand, pixel-by-pixel difference between two large patches cannot help distinguish patches from small scale details (e.g., repetitive patterns), and hence the resulting reconstructed image cannot preserve edges well.

Histogram matching can be an efficient replacement for comparing two large patches, and is mainly used in applications that are insensitive to image structure, such as tracking [11], [21]. However, the lost of structural information makes histogram matching not suitable for content-aware applications, such as colorization [24] and image editing [6]. Figure 3b shows an example of using histograms to compute ANNF for image reconstruction. (The details of image reconstruction will be discuss in Section V-B). The pixels near to the boundaries cannot be reconstructed due to the lost of structural information. On the other hand, adding spatial information along (i.e., LSHs) is not distinctive enough to obtain good patch correspondences, as shown in Figure 3c.

A. Bilateral Weighted Histograms

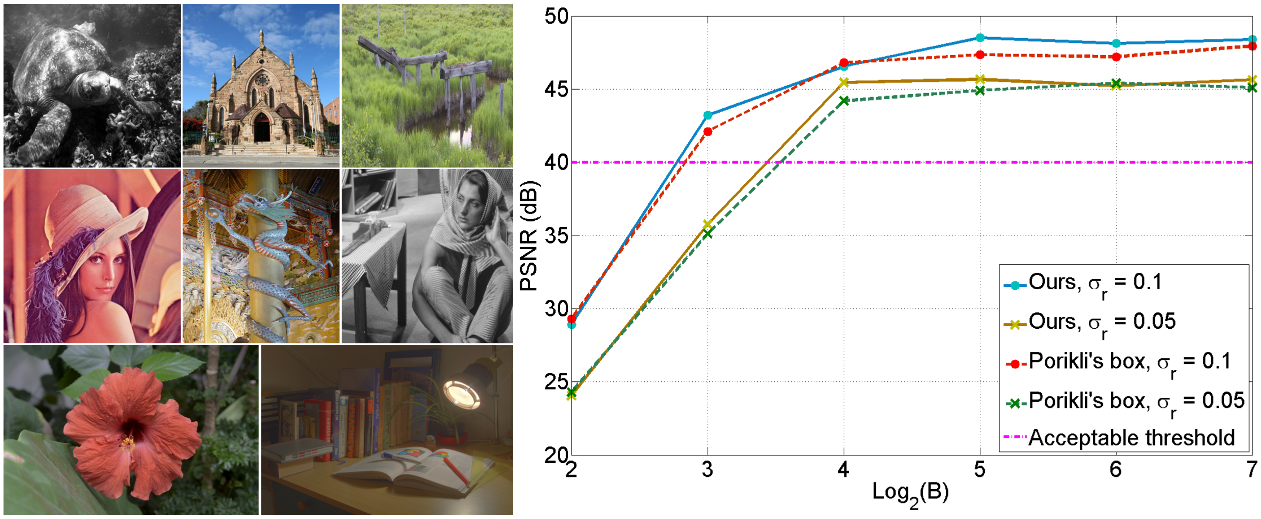
Here, we propose a new type of histogram, referred to as bilateral weighted histograms, to address both the efficiency and accuracy problems. The proposed BWH provides an alternative way for patch matching. Pixel-by-pixel difference aims to locate the patch with similar image structure, while BWH is relatively invariant to rotation (i.e., pixel contributions of BWH depend only on spatial and color distances) and it takes into consideration of both spatial and color information to preserve details around edge areas. In particular, the introduced range kernel provides a geodesic based similarity to recover image structure in an efficient manner.

For image $\mathbf{I}$, the BWH, $\mathbf{H}_p^W$, at pixel p is defined as:

$$\mathbf{H}_p^W(b) = \sum_{q=1}^{W} \alpha^{|p-q|} \cdot \mathcal{G}(\mathbf{I}_p,\mathbf{I}_q) \cdot Q(\mathbf{I}_q,b), \quad b=1,\ldots,B, \tag{23}$$

where $\mathcal{G}(\mathbf{I}_p,\mathbf{I}_q)$ is a range kernel as defined in Section III. We can see that Eq. (23) is an extension of LSH by further considering color similarity in the histogram.

One natural question is whether the efficient implementation of LSH holds in BWH. Inspired by a recently proposed recursive approximation of bilateral filtering [36], BWH can also be implemented in an efficient way by involving recursive range kernel.



(a) Sample images (b) PSNR

Fig. 4: Quantitative evaluation of the proposed bilateral filter using PSNR. (a) Eight images used in the experiment. (b) Average PSNR values computed from the filtered images ($\alpha = 0.91$) and the ground truth w.r.t. the number of bins B.

For pixels p and q in 1D image I, a recursive range filtering kernel $\mathcal{G}_r(\mathbf{I}_p, \mathbf{I}_q)$, is defined as the accumulation of the range distance between every two neighboring pixels on the path of two p and q:

$$\begin{aligned}\mathcal{G}_r(\mathbf{I}_p, \mathbf{I}_q) &= \mathcal{G}_r(\mathbf{I}_q, \mathbf{I}_p)\\ &= \mathcal{G}_r(\mathbf{I}_p, \mathbf{I}_{p+1})\cdots\mathcal{G}_r(\mathbf{I}_{q-2}, \mathbf{I}_{q-1})\mathcal{G}_r(\mathbf{I}_{q-1}, \mathbf{I}_q)\\ &= \prod_{k=p}^{q-1}\mathcal{G}_r(\mathbf{I}_k, \mathbf{I}_{k+1}).\end{aligned}\tag{24}$$

Gaussian kernel is typically used as range kernel, and Eq. (24) becomes:

$$\begin{aligned}\mathcal{G}_r(\mathbf{I}_p, \mathbf{I}_q) &= \prod_{k=p}^{q-1}\mathcal{G}_r(\mathbf{I}_k, \mathbf{I}_{k+1})\\ &= \prod_{k=p}^{q-1}\exp\left(-\frac{|I_k - I_{k+1}|^2}{2\sigma_w^2}\right)\\ &= \exp\left(-\frac{\sum_{k=p}^{q-1}|I_k - I_{k+1}|^2}{2\sigma_w^2}\right).\end{aligned}\tag{25}$$

As can be seen in Eq. (25), the recursive approximation of the range kernel is indeed a geodesic filtering kernel. Although the behavior of geodesic kernel is different from Gaussian kernel in highly-textured areas, geodesic kernel preserves edges quite well as demonstrated in [37].

The recursive implementation of Eq. (25) holds for histogram, as we can consider BWH as a combination of filtering B binary images. Similar to LSH, Eq. (23) can be written as:

$$\begin{aligned}\mathbf{H}_p^{W,left}(b) &=\\ Q(\mathbf{I}_p, b) &+ \alpha\cdot\mathcal{G}_r(\mathbf{I}_p - 1, \mathbf{I}_p)\cdot\mathbf{H}_{p-1}^{W,left}(b),\\ \mathbf{H}_p^{W,right}(b) &=\\ Q(\mathbf{I}_p, b) &+ \alpha\cdot\mathcal{G}_r(\mathbf{I}_p, \mathbf{I}_p + 1)\cdot\mathbf{H}_{p+1}^{W,right}(b).\end{aligned}\tag{26}$$

As the range distance $\mathcal{G}_r(\mathbf{I}_p, \mathbf{I}_q)$ can be pre-computed and stored as a lookup table, Eq. (26) requires only B extra multiplications than LSHs, thus totally $2B$ multiplication and B addition operations are needed at each pixel for computing $\mathbf{H}_p^{W,left}(b)$ (or $\mathbf{H}_p^{W,right}(b)$).

Extending the proposed BWH from 1D to 2D can be achieved by performing Eq. (26) horizontally first, and then apply Eq. (26) to the horizontal pass result vertically (or vice-versa). As a result, the range kernel has two possible paths according to the order of traveling directions. The optimal way is to choose the path with the lowest traveling cost at each pixel. However, this double the computational effort. Similar to [37], we perform horizontal first for all the pixels.

The proposed BWH is an edge-preserving histogram which enabling patch matching independent of patch size, meanwhile removing the influence from outliers (e.g., due to occlusions). We demonstrate its effectiveness by making the original PatchMatch algorithm to be edge-preserving.

B. Edge-Preserving PatchMatch

Given a pair of images I and J, PatchMatch first assigns the corresponding patches randomly. Typically some of these random matches are good (defined by the distance function). Only the good matches are propagated to the neighboring patches across image plane. Some random patch assignments are additional performed for each pixel in order to avoid trapping into local minima, and only the best match is kept.

For two patches that locate at p in image I and q in image J respectively, the distance function between two patches $d(p, q)$ of original PatchMatch is defined as:

$$d(p, q) = \sum_{k=-s/2}^{s/2}||I_{p+k} - J_{q+k}||^2,\tag{27}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2015.2430671, IEEE Transactions on Circuits and Systems for Video Technology

where s is the patch size. We employ the proposed BWH as the patch descriptor. Specifically, the distance function is defined as follows,

$$d^H(p,q) = \sum_{b=1}^{3B} |H_p^{W,I}(b) - H_q^{W,J}(b)|, \tag{28}$$

where $3B$ is the total number of bins of three color channels. The new distance function leads to an efficient edge-preserving PatchMatch algorithm, and it is qualitatively and quantitatively examined in the experiments by three applications. As can be seen in Figure 3d, edges are well preserved in the reconstructed image.

V. Experiments

In this section, we extensive evaluate the proposed bilateral filtering and edge-preserving PatchMatch in Section V-A and Section V-B respectively. The proposed methods are implemented in C++ on a PC with an Intel i5 3.3 GHz CPU (using only a single core) and 8 GB RAM.

A. Evaluation on Bilateral Filtering

To assess the performance of our algorithm, we have implemented a bilateral filter using our locality sensitive histograms, and the state-of-the-art method in terms of time complexity [31]. We evaluate a bilateral filter using locality sensitive histograms on several images by the peak signal-to-noise ratio (PSNR), where each one is normalized such that the intensity values range between 0 and 1. Given two grayscale images, $\mathbf{I}, \mathbf{J} \in [0,1]$, the PSNR is defined as:

$$\text{PSNR} = 10\log_{10}\left(\frac{h \cdot w}{\sum_p |\mathbf{I}_p - \mathbf{J}_p|^2}\right), \tag{29}$$

where h and w are the height and width of $\mathbf{I}$ and $\mathbf{J}$. As suggested by [27], the differences between two images may not be easily visible if the PSNR values are above 40 dB. The Gaussian kernel is selected as the range filter kernel in all our experiments:

$$\mathcal{G}(\mathbf{I}_p, h_b) = \exp\left(-\frac{|\mathbf{I}_p - h_b|^2}{2\sigma_r^2}\right), \tag{30}$$

where σ_r is the standard deviation of the Gaussian kernel.

Quantitative Evaluations. Figure 4(a) shows eight images that are used for numerical evaluation. The average PSNR values of our method and the exact method (with 256 bins) are shown in Figure 4(b). Our method is able to obtain high PSNR values even using as few as 16 bins, i.e., $\log_2(B) = 4$, and the results are as accurate as those by the state-of-the-art method [31] with box spatial filter kernel.

Figure. 5 shows the PSNR curves of the proposed method with respect to parameter σ_r defined in Eq. (30). We can observe that the propose bilteral filter suits for wide-ranging application requirements, as the accuracy of our method is high even for very small σ_r values.

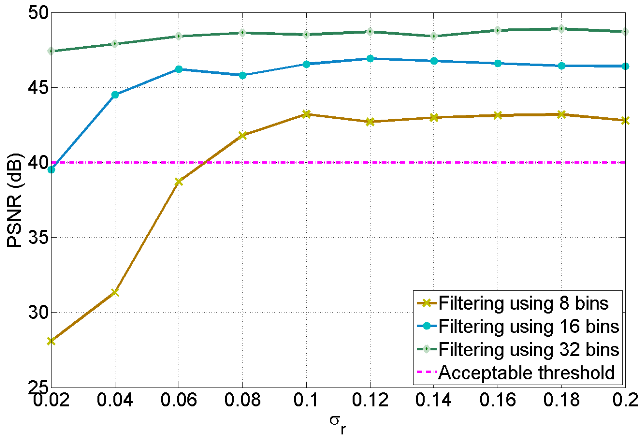


Fig. 5: Accuracy of the proposed bilateral filter in comparison to the exact filter w.r.t. different σ_r values.

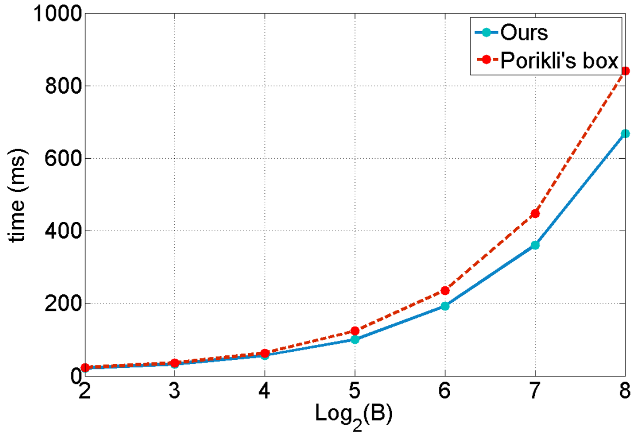


Fig. 6: Exact runtime of [31] and the proposed LSH-based bilateral filter on an 1 MP grayscale image.

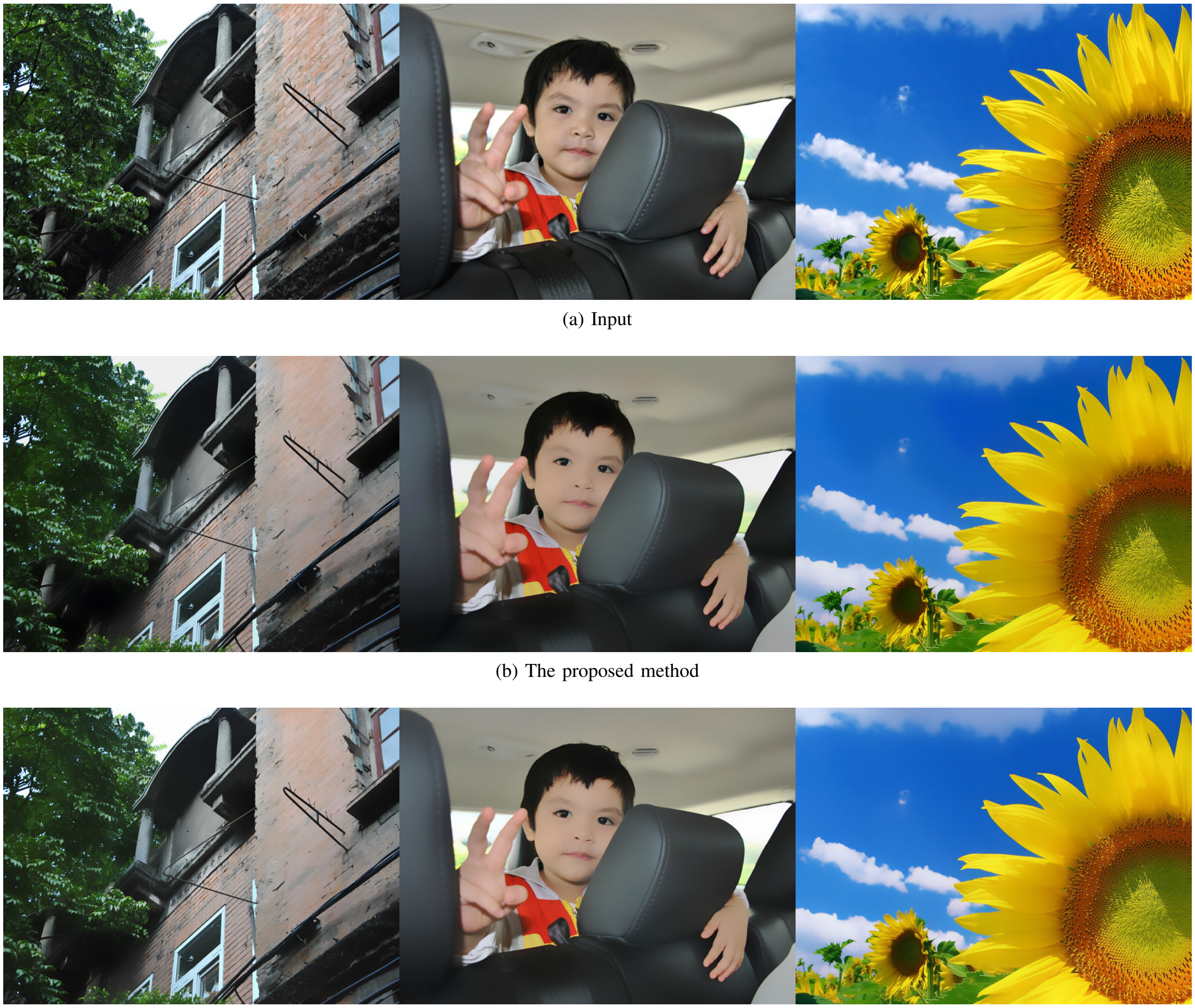
Figure 6 shows the execution time of the two histogram-based bilateral filter methods with respect to the number of bins B on an 1 MP grayscale image. The proposed LSH-based bilateral filtering method is a bit faster than the state-of-the-art constant-time histogram-based approach [31]. In addition, the computational complexity of our method is linear in both the number of bins and the image size.

Table I compares the speed of the proposed bilateral filter with the existing bilateral filters or approximations. We choose the number of bins to 16 for histogram-based methods as satisfactory PSNR can be obtained, as shown in Figure 4. As can be seen, the proposed bilateral filter outperforms all the existing method for processing grayscale image. One drawback is that the computational complexity of histogram-based methods is exponential in dimensions, and thus these methods are not suitable for high-dimensional filtering.

Qualitative Evaluations. Figure 7 shows the filtered image obtained from our method and the exact method (with 256 bins). The result obtained from our method is visually close to that obtained from the exact method, even using a relatively small number of bins (B = 16).

Figure 8 demonstrates the effectiveness of the proposed multi-dimensional filter. As shown in Figure 8(b), our multi-dimensional filter does not have the color bleeding

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2015.2430671, IEEE Transactions on Circuits and Systems for Video Technology



(a) Input

(b) The proposed method

(c) Ground truth

Fig. 7: Qualitative evaluation of the proposed bilateral filter. Locality sensitive histograms are computed with $\alpha = 0.91$. The standard deviation of the range filter kernel is $\sigma_r = 0.05$, and $B = 16$. Each channel is processed independently in this experiment.

TABLE I: Speed comparisons with existing edge-preserving filters on an 1 MP grayscale image. We use 16 bins for the histogram-based methods. As can be seen, the proposed bilateral filter is the fastest method for processing grayscale image.

Method	Runtime (ms)
Proposed	57
Porikli's box [31]	63
Bilateral Grid [9]	356
AM filter [17]	105
RTBF [38]	155

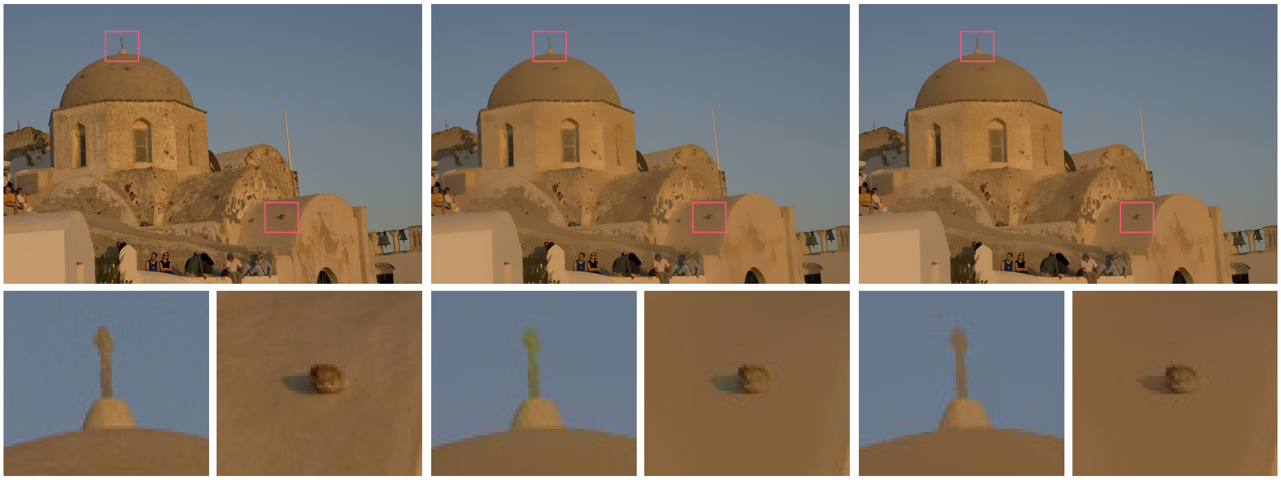
problem presented in Figure 8(a).

Figure 9 presents a stylization example of the proposed method, in which Figure 9c is produced using the illumination invariant features [21]. Note that the illumination invariant features are also based on LSHs.

B. Evaluation on PatchMatch

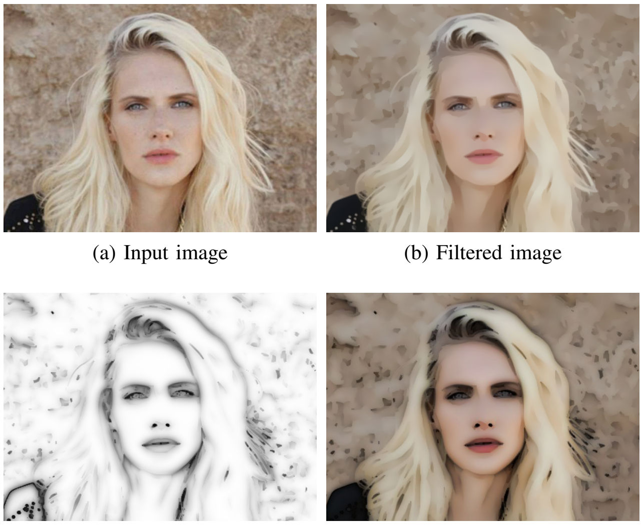
We evaluate the proposed edge-preserving PatchMatch by comparing it with the original PatchMatch. The proposed method is examined by three applications: image reconstruction, optical flow and example-based colorization. As the proposed BWH takes into account the whole image, we set the spatial cutoff to be the patch size s, and adjust the spatial coefficient α to force the smallest weight within the patch to be 0.1.

Image Reconstruction. The properties of NNF are typically examined by image reconstruction, which is the main step in image editing such image retargeting and inpainting. Given a source image I, it can be reconstructed by a target image J and a dense patch map from I to J. The experiment is conducted on the public data set VidPairs [25], but



(a) Input (b) Filtering each channel separately (c) Multi-dimensional filtering

Fig. 8: Multi-dimensional bilateral filtering results. (b) Combining three separately filtered color channels. (c) Multi-dimensional filtering. Note that in (b), the top of the building and the shadow of a small object merge with their backgrounds when filtering the three color channels separately. Our multi-dimensional filter is able to avoid this problem as shown in (c). (Best viewed at full size on a high-resolution display).



(a) Input image (b) Filtered image

(c) Edge image (d) Stylized image

Fig. 9: Stylization example of the proposed bilateral filtering. Edge image is produced using the illumination invariant features in [21].

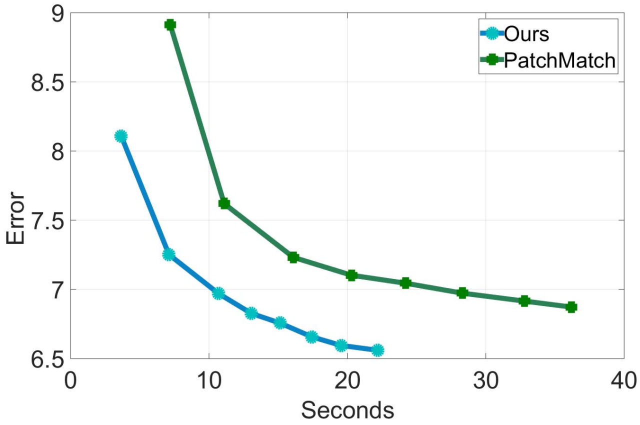
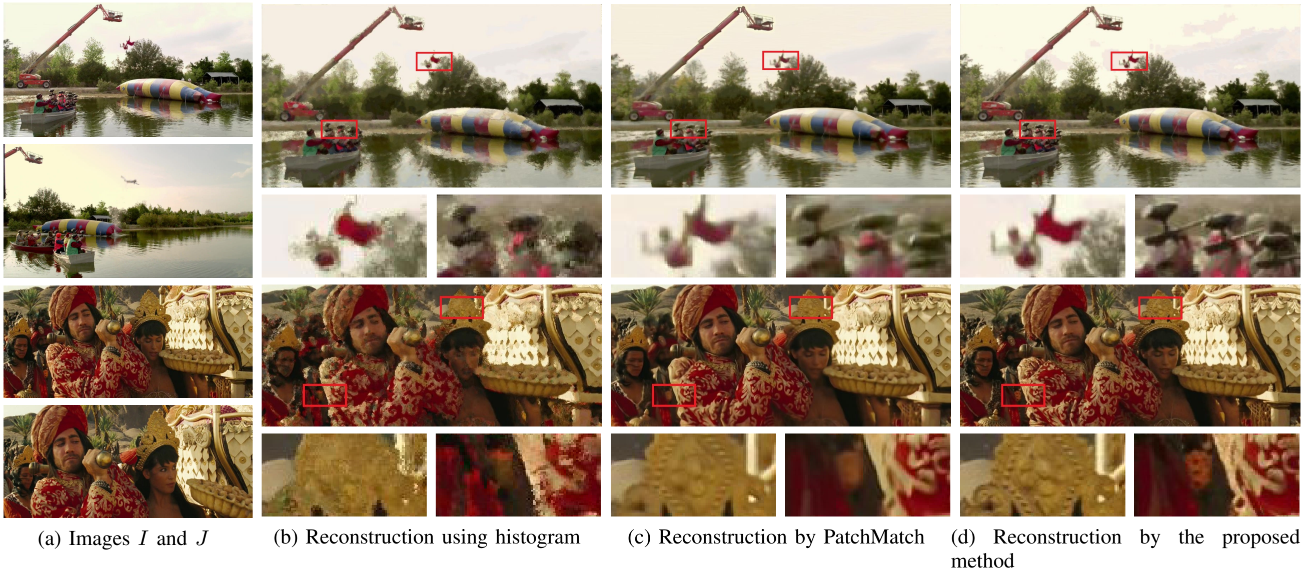


Fig. 10: Reconstruction error-time tradeoffs averaged on 133 image pairs. Root-mean-square error (RMSE) is used to measure the error between the original image and the reconstructed image. Each iteration is indicated by markers on the curves. Overall, the proposed method is able to achieve about $5-10\%$ lower in reconstruction errors and about 2 to 3 times faster than the original PatchMatch with similar accuracy.

resized to 0.4 MP. We follow [25] to reconstruct the image under the bidirectional similarity framework [32]. For each pixel, the reconstructed RGB values are determined by averaging the corresponding pixels in all the patches that cover it. The reconstruction error is measured by root-mean-square error (RMSE) between the original image and the reconstructed image.

Figure 10 shows the error-time curves using 8×8 patches. The error and running time are averaged over 133 image pairs in the data set. As can be seen in Figure 10, the proposed edge-preserving PatchMatch has two advantages. First, the proposed method is more accurate. It achieves about $5-10\%$ lower reconstruction errors than the original PatchMatch. Second, our method is about 2 to 3 times faster to achieve similar accuracy than the original PatchMatch. For example, the proposed method only takes 3 iterations to reach a similar accuracy as the original PatchMatch after 6 iterations. Meanwhile, computing the BWHs (16 bins) for an 1MP grayscale image takes about 60ms, and the time is tripled for color images, which is negligible comparing to the time of performing PatchMatch.

Figure 11 shows two examples of image reconstruction.

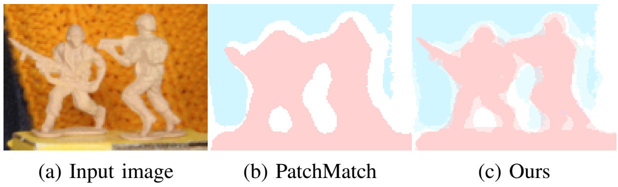

(a) Images I and J

(b) Reconstruction using histogram

(c) Reconstruction by PatchMatch

(d) Reconstruction by the proposed method

Fig. 11: Image reconstruction results. Image I is reconstructed from image J and the dense patch map from I to J. The proposed method in (d) preserves edges, yielding better results in the textured regions than the original PatchMatch in (c).


(a) Input image

(b) PatchMatch

(c) Ours

Fig. 12: Optical flow comparison of the original PatchMatch and the proposed method. The result (cropped) is tested on the "Army" dataset from Middleburry benchmark [4]. The proposed method preserves details better than the original PatchMatch.

The results shown in Figure 11b are reconstructed by the traditional histogram, which is not able to retain fine details. The original PatchMatch (Figure 11c) captures details well but produces over-smoothing artifacts. In contrast, the proposed method (Figure 11d) preserves the finest details, especially around edges.

Optical Flow. PatchMatch has been shown to be effective in large displacement optical flows [10], [22], [5]. We simply replace the original PatchMatch in [5] to evaluate the effectiveness of our method. As can be seen in Figure 12, the proposed method preserves edges much better than the original PatchMatch. Note that our distance function (i.e., Eq. (28)) is similar to the one used in [5] in spirit. The main difference is that [5] involves bilateral weights when computing pixel-by-pixel differences, while the bilateral weights are contained in the proposed histograms in an efficient manner.

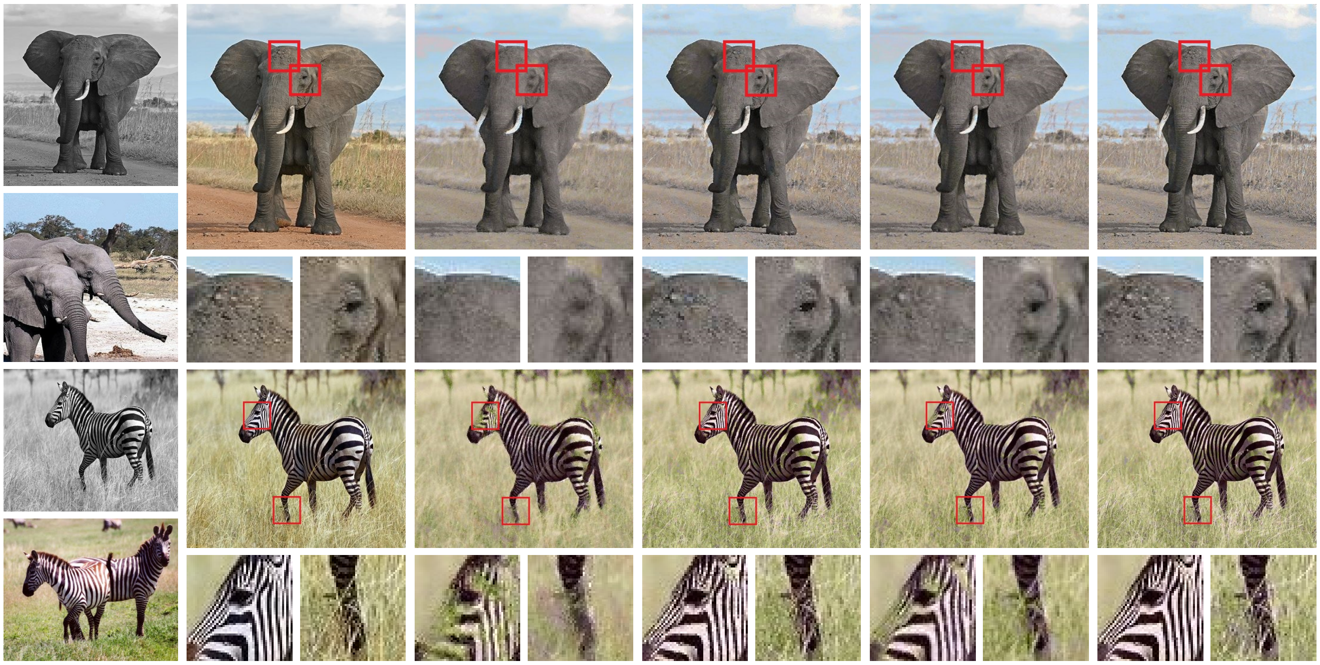
Example-based Colorization. Another intuitive example is example-based colorization. We simply compute the ANNF between the source grayscale image and the reference image, which is also converted to grayscale image. For each pixel in the source image, the color is assigned from the color of the corresponding patch center. Figure 13 shows the colorization results. We can see that the proposed method computes a better ANNF and preserves edge much better, while the original PatchMatch cannot distinguish patches from small scale details. The same situation occurs even with small patch size.

VI. Conclusions

In this paper, we further explore the LSH and propose a new BWH. Accordingly, we propose an efficient bilateral filtering algorithm based on LSHs and an edge-preserving PatchMatch based on BWHs. The proposed bilateral filtering algorithm overcomes the box spatial kernel restriction of existing histogram-based methods, enabling the use of an exponential kernel, and the new BWH leads to an accurate and efficient way for image matching. Experimental results show that the proposed bilateral filter and edge-preserving PatchMatch perform favorably against state-of-the-art algorithms. One of our future work is to extend the spatial kernel of LSH, which is currently an exponential kernel, to a general kernel. On the other hand, as many vision problems can be modeled with histograms, another future work will focus on exploring more potential applications (e.g., saliency detection [20], [19]) of two efficient histograms.

Acknowledgments

We would like to thank the anonymous reviewers for their insightful comments and constructive suggestions. The work described in this paper was partially supported by a GRF grant and an ECS grant from the RGC of Hong



(a) Target and reference images (b) Groundtruth (c) PatchMatch with 8 × 8 patch size (d) Proposed method with 8 × 8 patch size (e) PatchMatch with 4 × 4 patch size (f) Proposed method with 4 × 4 patch size

Fig. 13: Example-based colorization results with different patch sizes. The proposed method preserves edges much better than PatchMatch with either 8×8 or 4×4 patch size.

Kong (RGC Ref.: CityU 115112 and CityU 21201914). M.-H Yang is supported in part by the NSF CAREER Grant #1149783 and NSF IIS Grant #1152576.

References

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

[2] A. Adams, N. Gelfand, J. Dolson, and M. Levoy. Gaussian KD-trees for fast high-dimensional filtering. *ACM TOG*, **28**(3), July 2009.

[3] S. Bae, S. Paris, and F. Durand. Two-scale tone management for photographic look. *ACM TOG*, **25**:637–645, Jan. 2006.

[4] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. Black, and R. Szeliski. A database and evaluation methodology for optical flow. *IJCV*, 92(1):1–31, 2011.

[5] L. Bao, Q. Yang, and H. Jin. Fast edge-preserving patchmatch for large displacement optical flow. *IEEE TIP*, 23(12):4996–5006, Dec 2014.

[6] C. Barnes, E. Shechtman, A. Finkelstein, and D. Goldman. PatchMatch: A randomized correspondence algorithm for structural image editing. *ACM TOG*, 28(3), Aug. 2009.

[7] C. Barnes, E. Shechtman, D. Goldman, and A. Finkelstein. The generalized PatchMatch correspondence algorithm. In *Proc. ECCV*, Sept. 2010.

[8] E. Bennett and L. McMillan. Video enhancement using per-pixel virtual exposures. *ACM TOG*, **24**(3):845–852, 2005.

[9] J. Chen, S. Paris, and F. Durand. Real-time edge-aware image processing with the bilateral grid. *ACM TOG*, **26**(3), 2007.

[10] Z. Chen, H. Jin, Z. Lin, S. Cohen, and Y. Wu. Large displacement optical flow from nearest neighbor fields. In *Proc. IEEE CVPR*, pages 2443–2450, June 2013.

[11] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE TPAMI*, 25(5):564 – 577, 2003.

[12] F. Durand and J. Dorsey. Fast bilateral filtering for the display of high-dynamic-range images. *ACM TOG*, **21**(3), 2002.

[13] E. Eisemann and F. Durand. Flash photography enhancement via intrinsic relighting. *ACM TOG*, **23**(3):673–678, 2004.

[14] R. Fattal. Edge-avoiding wavelets and their applications. *ACM TOG*, **28**(3), 2009.

[15] J. Friedman, J. Bentley, and R. Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM TOMS*, 3(3):209–226, Sept. 1977.

[16] E. Gastal and M. Oliveira. Domain transform for edge-aware image and video processing. *ACM TOG*, **30**(4), 2011.

[17] E. Gastal and M. Oliveira. Adaptive manifolds for real-time high-dimensional filtering. *ACM TOG*, 31(4), 2012.

[18] K. He, J. Sun, and X. Tang. Guided image filtering. In *Proc. ECCV*, pages 1–14, 2010.

[19] S. He, R. Lau, W. Liu, Z. Huang, and Q. Yang. Supercnn: A superpixelwise convolutional neural network for salient object detection. *IJCV*, pages 1–15, 2015.

[20] S. He and R. W. Lau. Saliency detection with flash and no-flash image pairs. In *Proc. ECCV*, pages 110–124, 2014.

[21] S. He, Q. Yang, R. Lau, J. Wang, and M.-H. Yang. Visual tracking via locality sensitive histograms. In *Proc. IEEE CVPR*, 2013.

[22] M. Hornek, F. Besse, J. Kautz, A. Fitzgibbon, and C. Rother. Highly overparameterized optical flow using patchmatch belief propagation. In *Proc. ECCV*, pages 220–234, 2014.

[23] P. Indyk and R. Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *ACM STOC*, pages 604–613, 1998.

[24] R. Irony, D. Cohen-Or, and D. Lischinski. Colorization by example. In *Proc. EGSR*, pages 201–210, 2005.

[25] S. Korman and S. Avidan. Coherency sensitive hashing. In *Proc. IEEE ICCV*, pages 1607–1614, 2011.

[26] S. Paris and F. Durand. A fast approximation of the bilateral filter using a signal processing approach. *IJCV*, **81**:24–52, Jan. 2009.

[27] S. Paris, P. Kornprobst, and J. Tumblin. Bilateral filtering: Theory and applications. *Foundations and Trends in Computer Graphics and Vision*, 2009.

[28] G. Petschnigg, M. Agrawala, H. Hoppe, R. Szeliski, M. Cohen, and K. Toyama. Digital photography with flash and no-flash image pairs. *ACM TOG*, **23**(3), 2004.

[29] T. Pham and L. van Vliet. Separable bilateral filtering for fast video preprocessing. In *Proc. IEEE ICME*, 2005.

[30] F. Porikli. Integral histogram: a fast way to extract histograms in cartesian spaces. In *Proc. IEEE CVPR*, pages 829–836, 2005.

[31] F. Porikli. Constant time O(1) bilateral filtering. In *Proc. IEEE CVPR*, 2008.

[32] D. Simakov, Y. Caspi, E. Shechtman, and M. Irani. Summarizing

visual data using bidirectional similarity. In *Proc. IEEE CVPR*, pages 1–8, June 2008.

[33] J. Sun and K. He. Computing nearest-neighbor fields via propagation-assisted kd-trees. In *Proc. IEEE CVPR*, pages 111–118, 2012.

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

[35] B. Weiss. Fast median and bilateral filtering. *ACM TOG*, 25(3):519–526, July 2006.

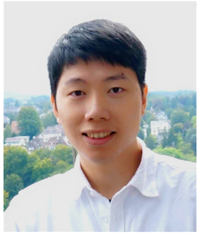
[36] Q. Yang. Recursive bilateral filtering. In *Proc. ECCV*, pages 399–413, 2012.

[37] Q. Yang. Recursive approximation of the bilateral filter. *IEEE TIP*, 2015.

[38] Q. Yang, K. Tan, and N. Ahuja. Real-time O(1) bilateral filtering. In *Proc. IEEE CVPR*, pages 557–564, 2009.

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

[40] K. Yoon and I. Kweon. Adaptive support-weight approach for correspondence search. *IEEE TPAMI*, **28**(4):650–656, 2006.



Shengfeng He is a Ph.D. student at City University of Hong Kong. He obtained his B.Sc. degree and M.Sc. degree from Macau University of Science and Technology. His research interests include computer vision, image processing, computer graphics, and machine learning.

Qingxiong Yang is an Assistant Professor in the Department of Computer Science at City University of Hong Kong. He obtained his BEng degree in Electronic Engineering & Information Science from University of Science & Technology of China (USTC) in 2004 and PhD degree in Electrical & Computer Engineering from University of Illinois at Urbana-Champaign in 2010. His research interests reside in Computer Vision and Computer Graphics. He won the best student paper award at MMSP 2010 and best demo at CVPR 2007.

Rynson W.H. Lau received his Ph.D. degree from University of Cambridge. He was on the faculty of Durham University and The Hong Kong Polytechnic University. He is now with City University of Hong Kong.

Rynson serves on the Editorial Board of Computer Animation and Virtual Worlds and IEEE Trans. on Learning Technologies. He has served as the Guest Editor of a number of journal special issues, including *ACM Trans. on Internet Technology*, *IEEE Trans. on Multimedia*, *IEEE Trans. on Visualization and Computer Graphics*, and *IEEE Computer Graphics & Applications*. In addition, he has also served in the committee of a number of conferences, including Program Co-chair of *ACM VRST 2004*, *ACM MTDL 2009*, *IEEE U-Media 2010*, and Conference Co-chair of *CASA 2005*, *ACM VRST 2005*, *ACM MDI 2009*, *ACM VRST 2014*.

Ming-Hsuan Yang is an associate professor in Electrical Engineering and Computer Science at University of California, Merced. He received the PhD degree in computer science from the University of Illinois at Urbana-Champaign in 2000. Prior to joining UC Merced in 2008, he was a senior research scientist at the Honda Research Institute working on vision problems related to humanoid robots. He coauthored the book Face Detection and Gesture Recognition for HumanComputer Interaction (Kluwer Academic 2001) and edited special issue on face recognition for Computer Vision and Image Understanding in 2003, and a special issue on real world face recognition for IEEE Transactions on Pattern Analysis and Machine Intelligence. Yang served as an associate editor of the IEEE Transactions on Pattern Analysis and Machine Intelligence from 2007 to 2011, and is an associate editor of the Image and Vision Computing. He received the NSF CAREER award in 2012, the Senate Award for Distinguished Early Career Research at UC Merced in 2011, and the Google Faculty Award in 2009. He is a senior member of the IEEE and the ACM.