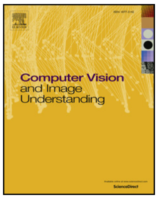
Computer Vision and Image Understanding 171 (2018) 34–47

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu



Multiple view image denoising using 3D focus image stacks

Shiwei Zhou[1], Zhengyang Lou, Yu Hen Hu*, Hongrui Jiang

Department of Electrical and Computer Engineering, University of Wisconsin, Madison, WI 53706, USA

ARTICLE INFO

Keywords:
Multi-view denoising
3D focus image stacks
Image reconstruction
Disparity map
Occlusion handling
Low rank minimization

ABSTRACT

In this paper, we introduce a novel multi-view image denoising algorithm using 3D focus image stacks (3DFIS) to exploit image redundancy within and across views. Robust disparity map is first estimated using the 3DFIS with texture-based view selection and patch-size variation scheme. Leveraging both 3DFIS and the estimated disparity map, the proposed algorithm effectively denoises the target view from multiple views through a low rank minimization approach that incorporates robust similarity metrics and occlusion handling techniques. The paper combs through a number of existing image denoising methods, including the preliminary results in our earlier research efforts, and then details the ways, means and merits of our proposed algorithm. With extensive experiments, we conclude that this novel algorithm is superior over various existing state-of-the-art approaches in terms of both visual and quantitative performance.

1. Introduction

With increasing popularity of camera networks and multi-camera imaging devices, multi-view image processing has become a key enabling tool for applications such as 3D scene reconstruction, object tracking and recognition, environmental surveillance, and 3DTV (Zhang, 2007). For example, while the prevalence of stereoscopic display has greatly enriched people's entertainment life, the application of multi-camera system in laparoscopic surgery has also expedited surgical tasks and improved safety of surgical procedures (Kanhere et al., 2014). In these applications, the availability of multiple views of the scene significantly extends the capability of imaging system by exploiting the 3D information behind the observed scene, and substantially enhances the performance.

Multi-view images are often captured with a camera array that consists of multiple cameras con as a regular geometric array. Compared with those expensive bulky digital single-lens reflex (DSLR) cameras, cameras in a camera array often have limited exposure and small aperture, for its flexibility and portability. As such, images captured by such cameras may be degraded due to image sensor granular noise, lower resolution, and even geometric distortion. The presence of noise not only degrades the perceptual image quality, but may also impede further image processing procedures including segmentation, super resolution, object detection, etc. Furthermore, the processing of 3D information from multi-view images, such as stereo matching and 3D reconstruction, may also be restrained.

Traditional denoising methods (described in more details in Section 2) generally call for reconstructing a cleaner patch from neighboring blocks of the same image to replace the noisy patch. These methods aim to search for intra-view similarities within image for removing the noise. However, as some unique textures in the image may not have corresponding similar patches, such methods often suffer from patch mismatches. With a set of multi-view images, both intra-view and inter-view similarities can be exploited to reduce the noise and reconstruct the clean image.

In this paper, the task of multi-view image denoising is considered. Given a set of noisy multi-view images, our goal is to remove the noise from the target view, which is one of the multi-view images, using noisy images from other viewpoints. Taking advantage of the 3D focus image stacks (3DFIS) and occlusion handling techniques in our previous works (Zhou et al., 2015,2017), this paper introduces a number of novel improvements in both disparity estimation and denoising. Specifically, our main contributions include:

- A new algorithm that appropriately selects views and patch sizes with respect to the texture map to enhance the accuracy of estimated disparity map, resulting in an improved denoising performance. (Section 4).
- A novel robust similarity metric and searching strategy based on patch volumes to reduce patch mismatches, and improve computational efficiency. (Section 5.2).
- A low-rank minimization denoising scheme applied to selected noisy multi-view image patches that yields superior denoising performance. (Section 5.3).

* Corresponding author.
E-mail addresses: szhou45@wisc.edu (S. Zhou), yhhu@wisc.edu (Y.H. Hu).
[1] Mail to first author (Shiwei Zhou).

https://doi.org/10.1016/j.cviu.2018.05.007
Received 28 October 2017; Received in revised form 3 March 2018; Accepted 13 May 2018
Available online 02 June 2018

- A set of new optimization strategies including book-keeping that have remarkably accelerated the computational speed of the algorithm. (Section 5).

Through extensive experiments, we demonstrate that these novel procedures greatly enhance the denoising performance over state-of-the-art single-view and multi-view image denoising methods, including our preliminary works (Zhou et al., 2015; Zhou et al., 2017).

The rest of this paper is organized as follows. The related work in multi-view image denoising is discussed in Section 2. A comprehensive introduction of 3DFIS is provided in Section 3. A robust multi-view disparity estimation algorithm is presented in Section 4. Detailed multi-view denoising procedures are described in Section 5. Experimental results and discussions are presented in Section 6. Conclusion and future works are in Section 7.

2. Related work

Existing image denoising methods process images in either the spatial domain or the transform domain. The former directly manipulates pixels in the spatial domain and estimates pixel values utilizing those of neighboring pixels. Examples of spatial domain denoising techniques include Gaussian filtering (Shapiro and Stockman, 2001), anisotropic filtering (Yang et al., 1995), bilateral filtering (Tomasi and Manduchi, 1998), and least-mean-square filtering (Widrow and Haykin, 2003), etc. These spatial filters attain high efficiency when image noise is low. As noise level increases, the performance of these filters degrades rapidly. In addition to above local filters, non-local approaches have been introduced to improve the denoising quality under high level of noise. Buades et al. (2005) proposed a non-local mean (NLM) filtering method that searches for similar patches from the entire image, instead of local neighborhood, and then uses the weighted average of these similar patches to estimate the denoised patch. K-SVD (Aharon et al., 2006) is another kind of spatial denoising technique that employs sparse coding and dictionary learning. The transform domain methods convert the image into transform domain using various transforms like DCT, wavelets, curvelets, etc., accompanied with denoising operations on transform domain coefficients. Typical examples of transform domain denoising methods include Wiener filtering (Wiener, 1949), wavelet-based techniques (Portilla et al., 2003; Eslami and Radha, 2003), and dictionary-based techniques (Elad and Aharon, 2006; Yan et al., 2013). The state-of-the-art performance is achieved with the combination of spatial and transform domain techniques. Dabov et al. (2007) proposed a block-matching 3D (BM3D) algorithm that groups similar 2D patches into 3D arrays and transforms the 3D array into 3D transform domain. Then collaborative filtering is performed in the 3D transform domain using Wiener filter, followed by an inverse 3D transform that produces the estimation of grouped blocks. Zhang et al. (2010) grouped similar local pixels in their LPG-PCA method, and then ran coefficient shrinkage in the principal component analysis (PCA) domain. Gu et al. (2014) proposed to exploit the image non-local similarity by solving a weighted nuclear norm minimization (WNNM) problem.

Distinct from above single image denoising methods, multi-view image denoising approaches may leverage depth information inferred from disparities among multi-view images to further enhance performance. Zhang et al. (2009) proposed to collaboratively denoise grouped similar patches by applying PCA or tensor analysis, and then restore the denoised image from denoised patches. These patches are grouped by considering similarity between corresponding patches in all other views using depth estimation. Based on the principle of Zhang et al. (2009), Thite and Zhang (2014) proposed to improve the denoising performance and computational complexity by using NLM to the multi-view images that is intuitively parallel and available for GPU acceleration. Similarly, Luo et al. (2013) presented an adaptive NLM that also adopts a robust joint-view distance metric to measure the similarity of patches and estimates an optimal number of patches to be used for denoising. Xue et al. (2010) employed BM3D and applied a patch-based multi-view stereo (Furukawa and Ponce, 2010) model reconstruction algorithm to identify feature points and facilitate search of similar patches in other views. Patches with smallest geodesic distances are selected for Wiener filtering. Yue et al. (2015) developed a two-stage strategy by exploring internal and external patch correlations for both single image and multi-view denoising.

Above multi-view denoising methods require exhaustive searching for similar patches across all views and are extremely computationally expensive. Instead, Miyata et al. (2014) introduced a fast multi-view image reconstruction algorithm based on plane sweeping (PS) (Collins, 1996). Using PS, the multi-focus images (MFI) and disparity map can be estimated, and denoising is achieved by selecting the in-focus pixels from the MFIs. Along this direction, Kodama and Kubota (2014) proposed to transform the synthesized MFIs into 2D frequency domain and apply a linear filter to suppress noise. Denoised images are then obtained by taking the inverse 2D transform. Since no exhaustive block matching is required, these algorithms promise significant computation reduction compared to the block matching methods. However, this speed improvement comes at the expense of degrading denoising quality. For example, the quality of the denoised image obtained by Miyata et al. (2014) does not show noticeable improvement over the conventional NLM method, not to mention the state-of-the-art BM3D.

Noticing the various limitations existing in the related approaches, our earlier efforts focused on developing the 3DFIS data structure and applying it to multi-view image denoising (Zhou et al., 2015, 2017). To advance what we have achieved, in this work, we present a comprehensive development and analysis of a new 3DFIS-based multi-view image denoising algorithm that is much improved from our earlier preliminary results. This new algorithm not only achieves better performance compared to existing state-of-the-art denoising algorithms, but also attains dramatic computation time reduction compared to those reported in Zhou et al. (2017).

3. 3D focus image stacks

In this section, we give a general background of multi-view image model and introduce the notion of 3D focus image stacks (3DFIS) under the multi-view image settings.

3.1. Multi-view images and noise model

We assume the multi-view images are acquired from a dense, planar rectangular array of cameras. Cameras are placed at grid points $(s, t) \in \mathbf{Z}^2$ which is a set of 2D indices of the camera array, and the grid size (in units of world coordinates) is $L_x \times L_y$. The camera located at (0, 0) will be designated as the reference camera (target view) against which the multi-view images taken by neighboring cameras in the array will be aligned. The optical axis of each camera is parallel to the normal vector of the plane where the cameras are placed. The images are taken using identical focal length f. We further denote l_x and l_y to be the sizes of a pixel in x and y directions.

The intensity of the pixel located at (x, y) coordinate of the image taken from camera (s, t) is expressed as:

$$I_{s,t}(x, y) = I'_{s,t}(x, y) + n_{s,t}(x, y) \tag{1}$$

where $I'_{s,\ t}$ is the noiseless image and $n_{s,t}$ is i.i.d. zero-mean Gaussian noise with variance σ^2, i.e. $n_{s,t}(x, y) \sim N(0, \sigma^2)$. In this work, we assume the noise variance is readily known, since multiple literature (Rank et al., 1999; Liu et al., 2006; Liu et al., 2013) have already been proposed for accurate noise estimation. Our objective is to obtain a denoised image I_{est} at the target view (0, 0) given the set of multi-view images $\{I_{s,t}(x, y), (s, t) \in \mathbf{Z}^2\}$. Furthermore, we assume the knowledge of

$I'_{0,0}(x, y)$ in our experiments, so that the quality of denoising can be measured using

$$PSNR = 10\log_{10}\frac{255^2}{\frac{1}{N}\sum_{x,y}\|I'_{0,0}(x,y) - I_{est}(x,y)\|^2} \tag{2}$$

where N is the total number of pixels in the image.

To denoise an image using multiple views taken from different viewpoints, the intuitive perspective is to gather redundant information, usually in form of similar patches, from the multiple views, and then perform denoising procedures to reduce the noise. In the process of gathering similar patches, conventional denoising algorithms, like NLM and BM3D, mostly conduct exhaustive searching in a certain region and compare patch appearance (in terms of Euclidean norm) to find the most similar patches. Such exhaustive searching is extremely time consuming, especially when searching range is extended to multiple views.

In this work, we choose to construct an image data structure called 3D focus image stacks (3DFIS) (Zhou et al., 2015) to facilitate efficient similar patch gathering. In the 3DFIS, corresponding pixels in different views are aligned as a column in the corresponding image stack, thus enabling efficient similar patch grouping. The disparity is the key to locate the correct 3DFIS stack for each pixel such that patch grouping can be carried out without exhaustive searching over all stacks. Therefore, the general framework of our denoising algorithm can be described as follows: first a series of 3DFIS corresponding to various candidate disparity values are constructed. Next, we estimate the disparity map of the target view from the 3DFIS. For each pixel, its corresponding 3DFIS is extracted using the disparity value and similar patches can be gathered without exhaustive searching. Finally, we perform the denoising operations on the grouped similar patches. Details of the algorithm and principles behind it will be elaborated in the following sections.

3.2. 3D focus image stack

Since the set of multi-view images are acquired from a camera array, the pixel locations corresponding to a common 3D point p at a pair of images on the (0, 0) and (s, t) views will differ by an amount known as the *disparity*. In particular, if pixel (x, y) on the reference view corresponds to point p on the object surface at *depth Z*, and (x', y') is its corresponding point on a different view (s, t), then using similar triangles, we have $I_{0,0}(x, y) \approx I_{s,t}(x', y')$, if

$$x' = x + s\cdot[L_x\cdot f/(Z\cdot l_x)] = x + s\cdot d_x, \quad \text{and} \tag{3a}$$

$$y' = y + t\cdot[L_y\cdot f/(Z\cdot l_y)] = y + t\cdot d_y \tag{3b}$$

where L_x, L_y and l_x, l_y stand for distances between cameras and pixel sizes in the x, y direction as defined previously, and d_x, d_y are disparity values (in units of number of pixels) in the x, y direction. The $[\cdot]$ operator rounds its content to nearest integer. For simplicity of presentation, we assume that the cameras are placed at equal distances and pixels are square, i.e. $L_x = L_y = L$ and $l_x = l_y = l$, then

$$d_x = d_y = [f\cdot(L/l)/Z] = d \tag{4}$$

Eq. (4) indicates that disparity d is inversely proportional to the depth Z. Since disparities are integer values, each d is quantized from a range of values $[f\cdot(L/l)/Z - 0.5, f\cdot(L/l)/Z + 0.5)$. In other words, each disparity d corresponds to a collection of depth values, which has the range

$$\Delta Z = \frac{f\cdot(L/l)}{d-0.5} - \frac{f\cdot(L/l)}{d+0.5} = \frac{f\cdot(L/l)}{d^2-0.25}, \quad d > 0 \tag{5}$$

According to Eqs. (4) and (5), as the disparity d increases, the object gets closer to the camera and the range of depth covered becomes smaller, and vice versa. Since normal lenses cannot focus if an object is too close, it is reasonable to set up a maximum value $d_{\max}$ for disparity d such that $d \le d_{\max}$.

For each disparity value d $(1 \le d \le d_{\max})$, we stack up the set of multi-view images $\{I_{s,t}(x, y); (s, t) \in \mathbf{Z}^2\}$ as follows: Assume there are K cameras in the camera array and each one corresponds to a grid point (s, t), i.e. there is a unique mapping from (s, t) to an integer k such that $1 \le k \le K$, then for each $1 \le d \le d_{\max}$, each image $I_{s,t}(x, y)$ at view (s, t) is shifted by (sd, td) and stacked upon each other to form a three-dimensional matrix

$$F^d(x, y, k) = I_{s,t}(x + sd, y + td) \tag{6}$$

The 3D matrix $F^d(x, y, k)$ is called a *3D focus image stack (3DFIS)* with respect to disparity value d. According to Eqs. (3) and (4), for a pixel (x, y) in the reference view, its corresponding points in other views at (s, t) coordinates are displaced by $-sd$ and $-td$ in the x and y direction. Therefore, if the (x, y) pixel at the reference view $I_{0,0}$ has the true disparity value equal to d, its corresponding points in other views will be shifted to the same position in the 3DFIS F^d. Consequently, the entire column of the F^d at position (x, y), denoted by $F^d(x, y,:)$, should have the same pixel value, that is,

$$F^d(x, y, :) = I_{0,0}(x, y)\cdot\mathbf{1}_{K\times1} \tag{7}$$

where $1\,K_{\times 1}$ is a vector consisting of all 1 s. We call such a pixel as an *in-focus* pixel because the true focal plane at this pixel has a disparity value d. On the other hand, if the true focal plane's disparity value is not d, then entries in the column vector $F^d(x, y,:)$ may not have the same value. Likewise, we call such a pixel as an *out-of-focus* pixel.

For demonstration purpose, we adopt a simple three-view system, as shown in Fig. 1, to illustrate the process. The reference point (x_0, y_0) in the center view (shown in black, $s = 0$) has the true disparity value d, and its corresponding points in the two side views (shown in red and blue) are denoted as (x_{-1}, y_{-1}), (x_{+1}, y_{+1}). The views are then shifted using Eq. (6) to form the 3DFIS. If the views are shifted by d, the three corresponding points will be moved to the same position and form a stacked column of pixels in the 3DFIS, which means the pixel is in-focus

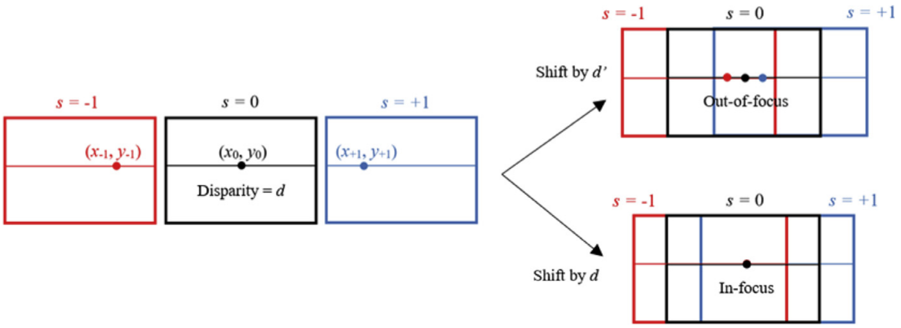


Fig. 1. Illustration of 3DFIS construction using a three-view system. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

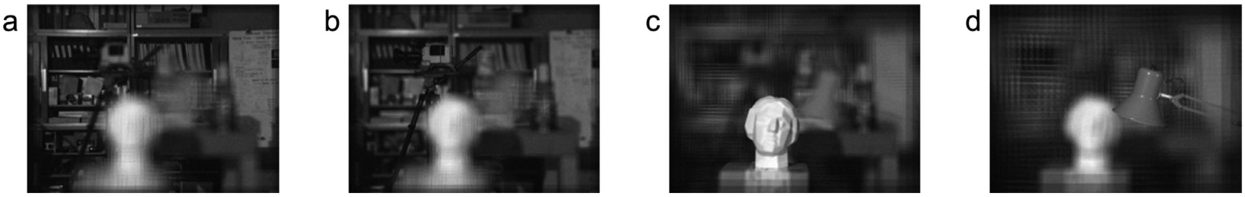


Fig. 2. 2D visualization of 3D focus image stacks: (a) disparity = 5; (b) disparity = 6; (c) disparity = 10; (d) disparity = 14.

in this 3DFIS. On the other hand, if the views are shifted by some other amount, namely *d'*, then the three corresponding points will not be in the same position and they become out of focus.

Derivatives of 3DFIS, such as the *multi-focus image* (*MFI*), have been applied to multi-view/light-field restoration and rendering (Miyata et al., 2014; Kodama and Kubota, Oct. 2014; Takahashi and Naemura, 2006). A MFI with respect to a disparity *d* is an average of $F^d(x, y, k)$ over all views for each pixel (x, y):

$$I^d(x, y) = \frac{1}{K}\sum_{k=1}^{K} F^d(x, y, k) \tag{8}$$

As shown in Fig. 2, MFI can be regarded as a 2D visualization of the corresponding 3DFIS. Clearly, regions containing in-focus pixels appear to be crisp and clear while regions corresponding to out-of-focus pixels appear to be blurred.

The MFI computed from 3DFIS can be used to estimate disparity map using a procedure called *plane sweep (PS)* (Collins, 1996), which is an approach of analyzing multi-view images by projecting them onto multiple focal planes in the 3D scene using the homography induced by the focal planes. To apply PS, the matching cost for each pixel (x, y) and each disparity *d* is computed using the sum of absolute difference (SAD) between each MFI and the reference view, as shown below

$$C(x, y, d) = \frac{1}{n_p}\sum_{(i,j)\in N(x,y)} |I^d(i, j) - I_{0,0}(i, j)|, \tag{9}$$

where $N(x, y)$ is the square patch centered at (x, y), and n_p is the number of pixels in $N(x, y)$. The disparity value of pixel (x, y) can then be estimated as

$$\hat{d}\,(x, y) = \arg\min_{d} C(x, y, d) \tag{10}$$

4. Disparity map estimation

Disparity map is an essential tool to extract the corresponding 3DFIS for each pixel during denoising process, so that patch matching can be conducted in selected image stack. Different from previous approaches (Zhang et al., 2009; Thite and Zhang, 2014; Miyata et al., 2014; Zhou et al., 2015; Takahashi and Naemura, 2006), we propose an improved multi-view disparity estimation algorithm that is robust to noise and yields superior disparity map under noisy conditions. This algorithm incorporates a robust matching cost, a texture-based view selection, and patch size variation scheme.

4.1. Disparity estimation with robust matching cost

Existing MFI-based multi-view depth estimation methods Miyata et al., 2014; Takahashi and Naemura, 2006) first compute a set of multi-focus images (MFI) with a known set of candidate disparity values. Then for each pixel, the disparity value is estimated using a procedure described in Eqs. (9) and ((10). This approach, while simple, often yields noisy, spurious disparity map with diminishing utilities as noise level increases. To improve the robustness of disparity estimation, we introduce a matching cost that is robust to noise.

For each pixel (x, y) and the supporting window *W* centered at it, let us denote by $\mathbf{v}^d_k$ a patch vector containing all pixel values within window *W* of *k*th view in the focus image stack F^d. For convenience, let $\mathbf{v}^d{}_1$ be the patch vector corresponding to the reference view. We then compute the vector difference between the *k*th view ($k > 1$) and reference view ($k = 1$):

$$\tilde{\mathbf{v}}^d_k = \mathbf{v}^d_k - \mathbf{v}^d_1, \quad 2 \le k \le K \tag{11}$$

Next, we sort the sequence $\{||\tilde{\mathbf{v}}^d_k||_1;\ 1 \le k \le K\}$ in increasing order such that $k \to k'$, $||\tilde{\mathbf{v}}^d_{k'}\,||_1 \le ||\tilde{\mathbf{v}}^d_{k'+1}||_1$. Here, $||\tilde{\mathbf{v}}^d_k||_1$ is the sum of absolute values of each element in $\tilde{\mathbf{v}}^d_k$. Define the matching cost as the mean absolute difference of *h* best $||\tilde{\mathbf{v}}^d_{k'}\,||_1$ $(1 < h \le K)$ as

$$C^*(x, y, d) = \frac{1}{n_p(h-1)}\sum_{k'=2}^{h} ||\tilde{\mathbf{v}}^d_{k'}||_1, \tag{12}$$

where n_p is the number of pixels in patch vector $\mathbf{v}^d_k$. The use of a patch vector instead of a single pixel in computing the cost function is based on an assumption that the disparity map is piecewise planar. Therefore, neighboring pixels are very likely to have the same disparity value, except at object boundaries where disparity values may change. Among all cost functions computed, the disparity value for pixel (x, y) is estimated as

$$\hat{d}^*(x, y) = \arg\min_{d} C^*(x, y, d) \tag{13}$$

In Appendix A, we show that the matching cost C^* in Eq. (12) is greater than or equal to the *C* in Eq. (9), if $h = K$, with equality holds when *d* is the true disparity, i.e. Eq. (7) holds. In other words, patch mismatches tend to produce higher cost in Eq. (12), and thus the proposed matching cost makes it easier to distinguish the true disparity from all other candidates, making the disparity estimation more robust to noise.

In Eq. (12), the choice of *h* is critically important to the accuracy of the estimated disparity map. If a view is occluded by another object due to discontinuities in the disparity map, serious bleeding artifacts as illustrated in Fig. 5(a) and (b) may degrade the quality of the estimated disparity map. Previously, Kang et al. (2001) proposed to use the best 50% of the frames (views) in computing the matching cost. Assuming the ground truth disparity map is available, we conducted an experiment comparing the number of erroneous pixels in estimated disparity maps against the number of views used, with the results plotted in Fig. 3. Note that the number of views that yields minimum disparity

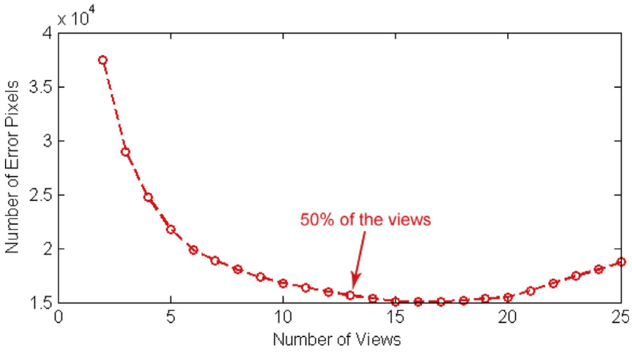


Fig. 3. Number of erroneous pixels in disparity maps estimated using different number of views.

estimation error is 16, which is greater than 12 or 13, i.e. 50% of 25 views. Hence, the value of h needs to be dynamically estimated for each pixel location. In practice, it is impossible to precisely identify the occluded regions and determine which view should be excluded. However, intuitively, occlusion often occurs at object boundaries and edges that may be identified using texture information of the image. Therefore, we propose a dynamic view selection procedure based on the texture analysis of the reference view.

Apart from view selection, we also consider the patch size used in computing the matching score. If the patch size is too small, details of edge formations may be revealed, but low-texture/flat regions are more corrupted with noisy artifacts due to the lack of features. With a larger patch size, the estimation is more robust to noise but at the cost of losing details at edges and boundaries. In this work, we propose a method to vary patch size selections based also on the texture analysis of the image.

4.2. Texture map estimation from multi-view images

Previously in Section 3.2, it is mentioned that the column vector $F^d(x, y,:)$ should have the same intensity value if the true disparity value at (x, y) is d, and vice versa. In other words, the variance of vector $F^d(x, y,:)$ is small only when the true disparity at pixel location (x, y) is d. However, this is only true for high-texture regions, where edges and pixel intensity variations are prevalent. In low-texture regions, due to the homogeneity of pixel values within the neighborhood, column vector $F^d(x, y,:)$ tends to contain similar values no matter whether the true disparity is d or not. This in turn makes the variance of $F^d(x, y,:)$ remain relatively small for all disparity values. This phenomenon inspires us to represent the strength of textures from the variance or standard deviation of 3DFIS.

Given 3DFIS $F^d(x, y, k)$, the standard deviation $\sigma^d(x, y)$ is obtained for each pixel (x, y) as

$$\sigma^d(x, y) = \sqrt{\frac{1}{K}\sum_{k=1}^{K}\left(F^d(x, y, k) - \overline{F}^d(x, y)\right)^2} \tag{14}$$

where $\overline{F}^d(x, y)$ is the mean value of vector $F^d(x, y,:)$. Then the strength of textures at (x, y) is defined as

$$\Sigma(x, y) = \frac{1}{d_{\max}}\sum_{d=1}^{d_{\max}}\sigma^d(x, y) \tag{15}$$

where $d_{\max}$ is the maximum of candidate disparity values, as mentioned in Section 3.2. To further reduce the impact of noise, we also apply a smoothing filter (e.g. Gaussian) to each standard deviation σ^d. Fig. 4 shows an example of texture map of a multi-view dataset, where bright colors (large values) represent high textures, while low textures are identified as dark colors (small values).

4.3. Texture-based view selection and patch size variation

With the texture map, both patch sizes and number of views to be selected can be estimated accordingly. Assume the patch size L may range from $L_{\min} \times L_{\min}$ to $L_{\max} \times L_{\max}$. Intuitively, the texture strength at pixel (x, y), $\Sigma(x, y)$, as defined in Eq. (15), decreases in flat areas, meaning a larger patch size is needed to capture the intensity fluctuation of patches. In textured regions, $\Sigma(x, y)$ tends to increase and a relatively small patch size is sufficient for disparity estimation. Consequently, we are seeking a linear relationship between $L(x, y)$ and $\Sigma(x, y)$ such that $L(x, y)$ increases as $\Sigma(x, y)$ decreases and vice versa, whilst being bounded by $L_{\min}$ and $L_{\max}$ when $\Sigma(x, y)$ reaches some pre-determined upper and lower thresholds (denoted by Σ_u and Σ_l). The maximum patch size $L_{\max}$ is used when $\Sigma(x, y) \le \Sigma_l$, and the minimum patch size $L_{\min}$ is used when $\Sigma(x, y) \ge \Sigma_u$. These two thresholds Σ_u and Σ_l are dependent on image noise level σ since noise may have an influence on the texture estimation. Further discussions on Σ_u and Σ_l are addressed in Section 6.1. Based on above considerations, the patch size $L(x, y)$ can be expressed as a linear function of the texture strength $\Sigma(x, y)$:

$$L(x, y) = \begin{cases} L_{\max}\,,\ \Sigma(x, y) \le \Sigma_l \\ \frac{L_{\max} - L_{\min}}{\Sigma_l - \Sigma_u}\cdot\Sigma(x, y) + \frac{\Sigma_l\cdot L_{\min} - \Sigma_u\cdot L_{\max}}{\Sigma_l - \Sigma_u},\ \Sigma_l < \Sigma(x, y) < \Sigma_u \\ L_{\min}\,,\ \Sigma(x, y) \ge \Sigma_u \end{cases} \tag{16}$$

Similarly, view selection can also be defined as a function of texture strength $\Sigma(x, y)$. On the basis of Eq. (16), we may directly estimate the number of selected views $V(x, y)$ from the patch size $L(x, y)$ as shown in Eq. (17) below. In this equation, the value of $V(x, y)$ varies from K, the total number of views, down to 0.5 K.

$$V(x, y) = \begin{cases} 0.5K\,,\ L(x, y) = L_{\min} \\ \frac{K}{2(L_{\max} - L_{\min})}\cdot L(x, y) + \frac{K\cdot L_{\max} - 2K\cdot L_{\min}}{2(L_{\max} - L_{\min})},\ L_{\min} < L(x, y) < L_{\max}. \\ K\,,\ L(x, y) = L_{\max} \end{cases} \tag{17}$$

The evaluation criteria for measuring the quality of disparity map is the error percentage, which is defined as

$$Err(d_{est}) = \frac{1}{N}\sum_{i=1}^{N}\left(d_{est}(i) \ne d_{gt}(i)\right) \tag{18}$$

where d_{est} is the estimated disparity map, d_{gt} is the ground truth, and N is the total number of pixels in the image. The effectiveness of dynamic patch size variation and view selection is illustrated in Fig. 5(a)–(d). With the robust view selection and patch size variation process implemented, we see the error percentage of estimated disparity maps dropping from over 33% to below 20%.

In Fig. 6, the proposed method is compared with other conventional stereo and multi-view disparity estimation algorithms (Miyata et al., 2014; Zhou et al., 2015; Taniai et al., 2016; Lee et al., 2015; Klaus et al., 2006). In this experiment, images are corrupted by noise with a noise level $\sigma = 20$. Fig. 6(a) is the top stereo matching method in Middlebury stereo benchmark proposed by Taniai et al. (2016), while (b) and (c) are also state-of-the-art stereo algorithms in recent years. The results of previous multi-view algorithms are shown in Fig. 6(d) and (e), including our preliminary results (Zhou et al., 2015). The result of proposed method is shown in Fig. 6(f), which demonstrates clear improvement in terms of both visual and quantitative quality. The significance of accurate disparity map estimation to multi-view image

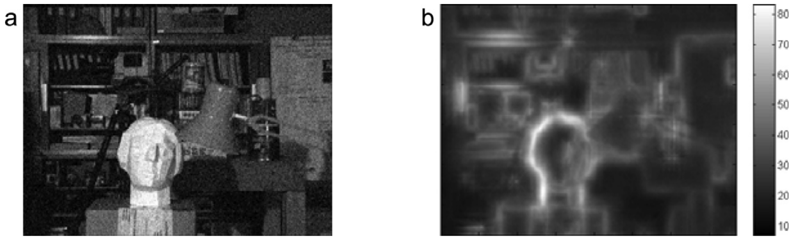


Fig. 4. (a) Noisy image; (b) texture map.

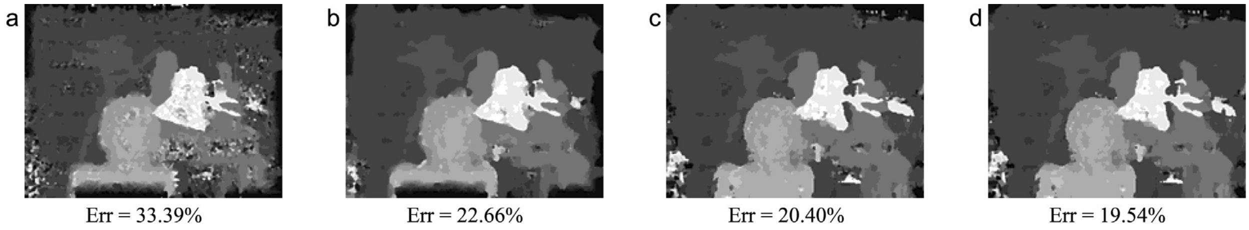


Fig. 5. Disparity maps using (a) fixed patch size (5 × 5) + all views; (b) variable patch size + all views; (c) variable patch size + 50% views; (d) variable patch size + view selection.

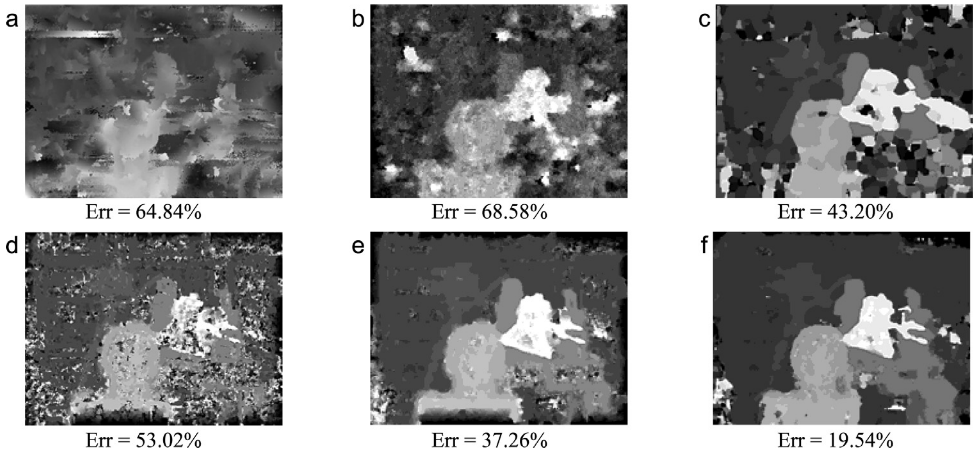


Fig. 6. Comparison of disparity estimation methods: (a) Taniai et al. (2016); (b) Lee et al. (2015); (c) Klaus et al. (2006); (d) Miyata et al. (2014); (e) Zhou et al. (2015); (f) proposed.

denoising will be discussed in Section 6.4. The proposed multi-view disparity estimation algorithm is summarized in Algorithm 1.

5. Multi-view image denoising

In this section, we present the proposed multi-view denoising algorithm using disparity map and 3DFIS. For each pixel (x, y) in the target view, its value is estimated from aggregation of multiple denoised patches that cover this pixel. The value of a denoised patch will be computed using low-rank minimization of a set of "similar" patches judiciously selected from the 3DFIS corresponding to the estimated disparity of the patch. Compared with previously reported results (Zhou et al., 2015,2017), the proposed multi-view denoising algorithm employs the following procedures that result in significant performance enhancement:

- A depth (disparity) guided adaptive window selection procedure with book-keeping strategy is implemented to select patches having consistent texture with the reference patch from the 3DFIS.
- A robust similarity metric that is resilient to noise interference while facilitating more efficient similar patch searching in the 3DFIS is incorporated.
- A low-rank minimization procedure is applied to yield a denoised patch from multi-view similar patches.

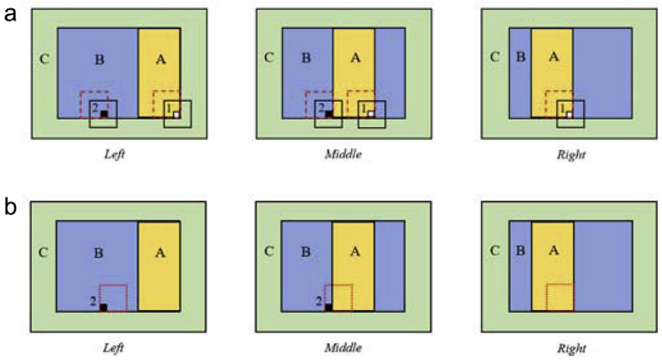


Fig. 7. Illustration of adaptive windows: (a) centered window (solid line) and top-left window (dashed line); (b) top-right window (dotted line).

5.1. Depth-guided adaptive window selection

For a pixel (x, y) in the target view, a patch of surrounding pixels in a square window is considered as the reference patch in the following denoising procedure. Traditionally, this pixel is at the center of the window. However, when the pixel is positioned near a discontinuity of the disparity map, the appearances of patches in the corresponding 3DFIS stack column are often inconsistent and provide no good candidates for denoising. This is illustrated in Fig. 7(a), in which A, B, C represent regions with different disparity values. Positions of region A varies because it has a different disparity value than those of B and C. When the window is centered at the pixel (solid line), patches #1 and #2 tend to make incorrect matches across the left, middle and right views. However, if the top-left window (dashed line) is used, the corresponding patches are matched correctly. In this work, we consider five different window definitions with index $j = 0$ indicating pixel being at the center of the window, while indices $j = 1$ to 4 representing pixel locations at each of the four corners of the window, as shown in Fig. 8(a). These windows are called *adaptive windows*. The notion of adaptive window has been discussed in literatures (Kang et al., 2001; Nakamura et al., 1996; Tao et al., 2001) with selection criteria solely based on intensity values. In this work, both intensity values and disparity values are considered.

To see the difference of our approach, consider an example shown in Fig. 7(b), in which the left view is supposed to be selected for pixel #2 (same as Fig. 7(a)). The dotted line refers to the top-right window. If

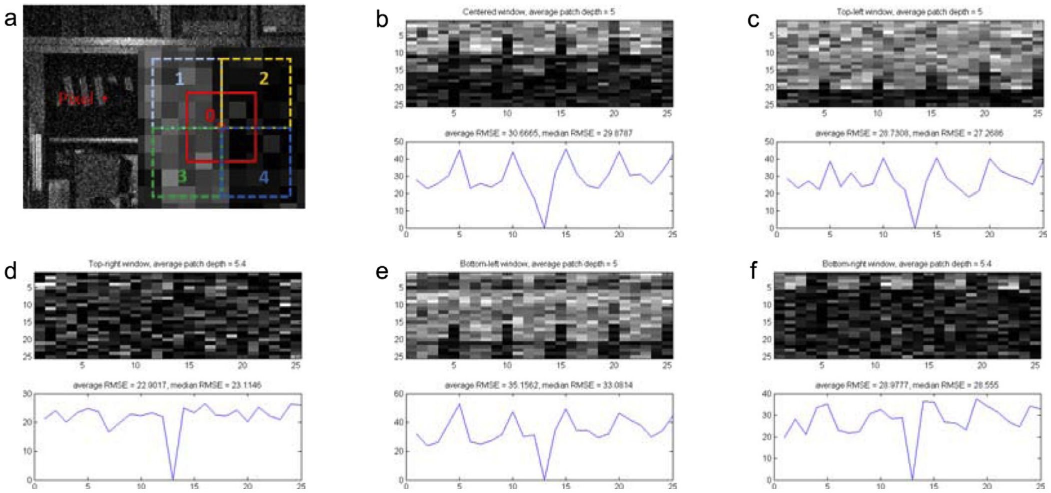


Fig. 8. (a) sample pixel and five windows; (b)-(f) vectorized patches across all views and RMSE plots for $j = 0, 1, 2, 3, 4$.

intensity is solely used for window selection, the right view could have been selected as it has more similarity to the patch at the middle (reference) view, and future denoising procedure tends to use the incorrect views for denoising. Thus, we propose to use depth information to eliminate such ambiguities. For this purpose, define the average disparity value of each candidate window as

$$\overline{d}_j(x, y) = \frac{1}{N_j} \sum_{(x', y') \in W_j} d(x', y') \tag{19}$$

where W_j ($j = 0, 1, 2, 3, 4$) is one of the five windows that cover (x, y), and N_j denotes the number of elements in W_j. The window selection is based on two criteria **C1** and **C2**:

C1. Define $J = \{j \mid j = \arg\min_j \mid \overline{d}_j(x, y) - d(x, y)|\}$. If there is only one element in J, i.e. $J = \{j_1\}$, then select window $j^* = j_1$.

C2. If J consists of two or more elements in **C1**, the median root-mean-square error (RMSE) between patches in each view and the reference view is used to break the tie. The use of median instead of mean is to reduce the impact of significantly biased outliers in the RMSE values. Specifically, for window j, the median RMSE is defined as

$$\delta_j(x, y) = \text{median}[\delta_{j,k}(x, y)] \tag{20}$$

where

$$\delta_{j,k}(x, y) = \frac{1}{N_j} \sqrt{\sum_{(x', y') \in W_j} |F^d(x', y', k) - F^d(x', y', 1)|^2} \tag{21}$$

is the RMSE between patches in the kth view and the reference view. In Eq. (21), $F^d(x', y', k)$ is the kth view in the corresponding 3DFIS with $k = 1$ being the reference view. Select window j^* such that $j^* = \arg\min_j \delta_j(x, y)$.

An example of deploying these two criteria in adaptive window selection is shown in Fig. 8, which visualizes the vectorized patches and plots the RMSE values across all views. The corresponding $\overline{d}_j(x, y)$ and $\delta_j(x, y)$ are shown in Table 1. Using criterion **C1**, we have $J = \{0, 1, 3\}$. Using **C2**, the values of $\delta_j(x, y)$ for $j = 0, 1, 3$ are compared and one has $j^* = 1$, meaning the top-left window should be chosen. Note that the window $j = 2$ actually has the lowest δ_j value, but it is positioned on the wrong object which has a different disparity than the reference pixel.

Table 1
Median disparity and mean RMSE for window selection in Fig. 8.

j	0	1	2	3	4
$\overline{d}_j(x, y)$	5	5	5.4	5	5.4
$\delta_j(x, y)$	29.88	27.27	23.11	28.56	33.08

When performing adaptive window selection, we found that some pixel locations may share the same adaptive window, resulting in many repeated calculations. To save computing time, we introduce the bookkeeping method that records the center pixel location of each adaptive window whenever it is processed. If the window for current pixel has already been computed according to the record, the algorithm will reuse the window and moves to the next pixel.

5.2. 3DFIS-based robust patch volume grouping

Once the patch window at the target view is selected, the algorithm proceeds to search for similar patches over the 3DFIS to be used for denoising. Previously, Zhang et al. (2009) proposed a disparity-guided searching strategy that maps patches from the reference view to all other views and computes the sum of squared differences of all mapped patches as the similarity metric. This similarity metric is based on the assumption that if two patches are similar in the target view, their corresponding patches in all other views should also be similar. Although it improves the patch matching by a significant level compared to Euclidean norm, the searching process is time consuming and the results are dependent on the accuracy of disparity map. In this work, we also incorporate another assumption, that is, if two similar patches are on the same depth plane, then their spatial relationship should also remain across all views. This assumption is intuitive and easy to understand, since most objects in our multi-view scenarios are stationary and rigid body. Combining these two assumptions, and with the help of 3DFIS, we developed novel similarity metric as well as a very efficient way of patch matching that avoids exhaustive searching in the three-dimensional space.

For each pixel (x, y), we have its disparity d and corresponding 3DFIS F^d. If the centered window ($j = 0$) is selected in the previous subsection, define a *patch volume* P_0 as the column stack of patches in F^d centered at (x, y) such that

$$P_0(x, y) = F^d(x - r: x + r, y - r: y + r, :), \tag{22}$$

where r is the patch radius, i.e. half-length of patch's side. For other adaptive windows, the location of the patch volume can be adjusted accordingly which is trivial. In the 2D neighborhood of (x, y) in F^d, the algorithm searches among other patch volumes, namely P_i ($i = 1, 2, \ldots$), for the M best ones that have smallest distances with P_0, defined as

$$\Phi(P_i, P_0) = \frac{1}{m_p} \|P_i - P_0\|_1, \tag{23}$$

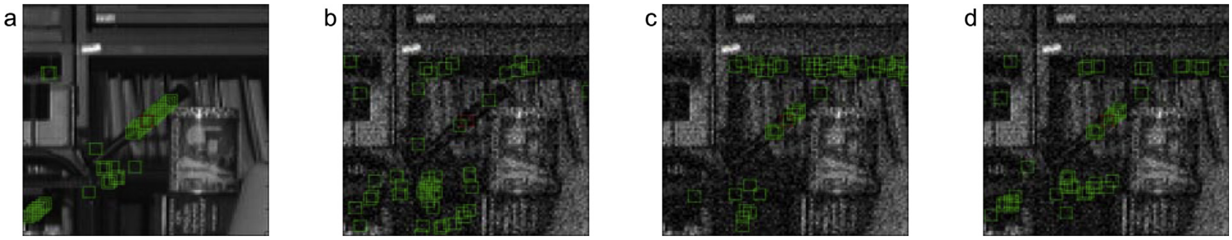


Fig. 9. Similar patch searching using different metrics: (a) block matching (clean); (b) block matching (noisy); (c) Zhang et al. (2009); (d) proposed. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

where $||\cdot||_1$ denotes the sum of absolute values as defined in Eq. (12), and m_p is the total number of elements in P_i (or P_0). By incorporating the two assumptions discussed above, Eq. (23) exhibits a more robust behavior to noise while reducing the amount of computations by simplifying exhaustive searching in 3D space to 2D searching. Fig. 9 shows a comparison of different similarity metrics, with red box representing reference patch and green boxes representing similar patches. We can see that the proposed similarity metric has a performance closer to ideal searching in clean image.

Using the procedures described above, we are able to identify a number of patches similar to the reference patch. A few patches within the patch volumes may still be inconsistent with the majority due to proximity to disparity discontinuities. These patches, also called *outliers*, are removed using an outlier rejection criterion that involves median absolute deviation (MAD). Conventional outlier rejection often uses standard deviation around the mean as a threshold. However, this method is unreliable as both mean and standard deviation are affected by the outliers. In contrast, the absolute deviation around the median is a more robust measure (Leys et al., 2013). For each pixel location, denote p_1 to be the reference patch in the target view, and p_i $(i > 1)$ to be other selected patches. The algorithm first computes the patch difference for each i as

$$\varphi_i = \|p_i - p_1\|_1, \, i = 1, \, \ldots, n \tag{24}$$

Let $\boldsymbol{\varphi} = [\varphi_1, \ldots, \varphi_i]$, then the median absolute deviation (MAD) is defined as

$$\mathrm{MAD}(\varphi) = \mathrm{median}[|\varphi_i - \mathrm{median}[\varphi]|] \tag{25}$$

Patches p_i with $\varphi_i > \mathrm{median}(\boldsymbol{\varphi}) + 3{\cdot}\mathrm{MAD}(\boldsymbol{\varphi})$ will be designated as an outlier and discarded. Here coefficient 3 in the threshold follows conventions in outlier rejection (Leys et al., 2013).

5.3. Patch denoising using low rank minimization

Low rank (LR) minimization has been widely used in image processing and has been shown to generate state-of-the-art denoising performance (Gu et al., 2014; Cai et al., 2010; Ji et al., 2010; Hu et al., 2015) that can compare with spatial domain or transform domain methods like BM3D. Due to the image redundancy in many natural images, similar patches discovered in the previous steps can be assumed to form a low rank matrix. Therefore, a noise-free patch can be recovered using low rank minimization. Given the observed data Y, the estimate of the latent data X, namely $\hat{X}$, is the solution of

$$\arg\min_X \|Y - X\|_F^2 + \lambda \mathrm{rank}(X) \tag{26}$$

where $||\cdot||_F$ denotes the Frobenius norm. Note that Eq. (26) is NP-hard, but can be relaxed as a convex optimization problem as

$$\arg\min_X \|Y - X\|_F^2 + \lambda \, \|X\|_* \tag{27}$$

where $||\cdot||_*$ is the nuclear norm. This convex relaxation can be solved using singular value decomposition (SVD) (Cai et al., 2010) as

$$\widehat{X} = US_\lambda V^T \tag{28}$$

where $Y = USV^T$ is the SVD of Y, and S_λ is the hard-thresholding result of diagonal matrix S such that

$$S_\lambda(i, i) = \max\{S(i, i) - \lambda, 0\} \tag{29}$$

The threshold λ in Eq. (29) plays an influential role in determining the denoising performance. Too large threshold would result in over-smoothing, while too small threshold tends to maintain a few noises in the estimated image. Hu et al. (2015) determined the optimal value of λ by minimizing the mean squared error of estimated values of vector patches. In this work, we choose the same threshold for our LR minimization, i.e. $\lambda = 1.5\sigma\sqrt{N_p}$, where σ is the standard deviation of noise and N_p is the number of similar patches.

For each pixel, similar patches $p_1, \ldots, p_n$ are grouped and the above patch denoising procedure is performed, resulting in a group of denoised patches $p'_1, \ldots, p'_n$. The denoised patch $\hat{p}$ is then computed as the weighted average of p'_I, $i = 1, \ldots, n$,

$$\widehat{p} = \frac{\sum_{i=1}^{n} w(p'_i, p'_1) p'_i}{\sum_{i=1}^{n} w(p'_i, p'_1)} \tag{30}$$

The weight is computed as a non-increasing function

$$w(p'_i, p'_1) = e^{-\frac{\|p'_i - p'_1\|^2}{\rho^2}} \tag{31}$$

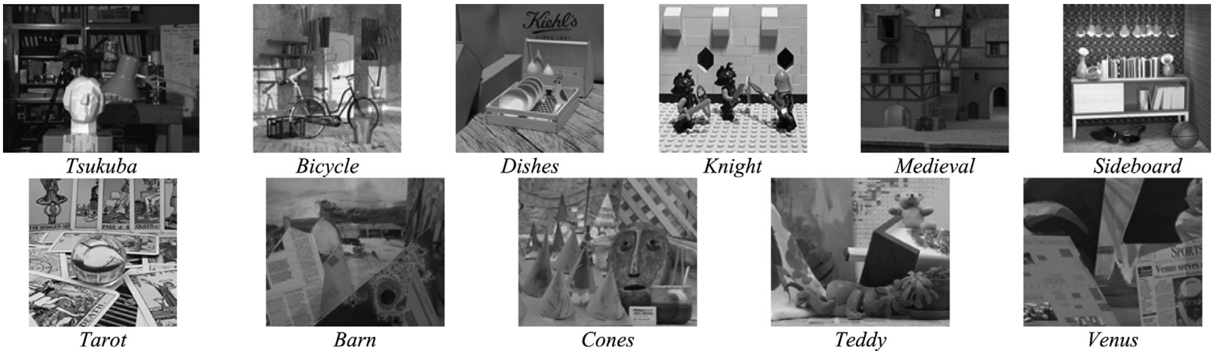
where ρ is the filtering parameter controlling the decaying of the weighting function.

Finally, we take an aggregation step to reconstruct the denoised image from denoised patches. Each pixel is covered by multiple denoised patches, and to determine the value of the pixel in the denoised image, we can take an average of all denoised patches that cover this pixel. The summary of the proposed multi-view denoising algorithm is shown in Algorithm 2.

6. Experiments and discussions

We used eleven datasets from different databases for performance evaluation. Fig. 10 shows one image from each dataset. “Tsukuba” is a 5 × 5 dataset from Middlebury Multi-view Stereo Datasets, (2018). “Knight” and “Tarot” are 17 × 17 datasets from The (New) Stanford Light Field Archive (2018), while “Bicycle”, “Dishes”, “Medieval”, and “Sideboard” are 9 × 9 datasets from 4D Light Field Benchmark from Universität Konstanz (4D Light Field Benchmark, 2018), and we will use the 5 × 5 subset of them. Four additional smaller image sets, namely “Barn”, “Cones”, “Teddy”, and “Venus”, from Middlebury Multi-view Stereo Datasets (2018) are used for specific comparison with the multi-view denoising algorithm proposed in Luo et al. (2013). All datasets from The (New) Stanford Light Field Archive (2018) and 4D Light Field Benchmark (2018) are resized to 256 × 256, while “Tsukuba”, “Barn”, “Cones”, “Teddy”, and “Venus” keep their sizes unchanged. For all datasets, white Gaussian noise with noise levels $\sigma = 20, 30, 40, 50$ are added. We evaluate the denoising performance using peak signal-to-noise ratio (PSNR) which is defined in Eq. (2).

The hardware platform consists of an Intel® Core™ i7-4700MQ CPU



Tsukuba Bicycle Dishes Knight Medieval Sideboard

Tarot Barn Cones Teddy Venus

Fig. 10. Image datasets for experiment.

(2.40 GHz) with a NVIDIA GForce® GT750M graphic card. The programs are implemented with MATLAB® R2014.

6.1. Parameter setting

Several parameters are tuned in experiments for best performance. The same parameters are applied to all image datasets under the same experiment conditions. In the disparity map estimation procedure, the maximum and minimum patch sizes are set to $L_{\max} = 15$, $L_{\min} = 5$. This is because patches smaller than 5 × 5 contain insufficient number of pixels to represent the photo-consistency while oversized patches tend to smooth disparity boundaries. The upper and lower thresholds Σ_u and Σ_l in Eq. (16) are defined as $\Sigma_u = 0.5\sigma + 19$, $\Sigma_l = 0.75\sigma + 5$, where σ is the noise standard deviation. Such a choice is motivated by the fact that texture strength for both flat and textured regions increases as noise level rises, but at a different rate. The flat regions are more likely to be influenced by the noise, hence we are setting a relatively faster change rate for Σ_l. If the noise level is too high (e.g. $\sigma > 50$), the default 50% of the views and patch size $L_{\text{def}} = 5$ will be used for all patches regardless of textures, due to the fact that textures are severely corrupted by noise in this situation.

In the denoising procedure, the maximum disparity range $d_{\max}$ is set to 15, which is sufficiently large for most multi-view images in the testing set. The patch size is fixed to 5, i.e. $r = 2$ in Eq. (22). As described in Section 5.3, the threshold λ in Eq. (29) is set to $1.5\sigma\sqrt{N_p}$ according to Hu et al. (2015). Another important parameter is the decaying parameter ρ in Eq. (31), which plays a similar role in the corresponding part of NLM algorithm. Large ρ tends to average all patches, acting as an averaging operator, while small ρ puts more weights on similar patches. The original NLM took value of 0.35σ or 0.4σ for this parameter, with respect to different levels of noise. With experiments on several datasets, we found that $\rho = 0.4\sigma$ achieves sufficiently satisfactory performance.

The number of similar patch volumes n and the radius (half size) of searching window R are two flexible parameters that affect both denoising performance and computational time. In Table 2, the PSNR and the run time comparisons using different values of n, R on the “Tsukuba” dataset are reported. When R is very small, e.g. $R \le 3$, there may not be enough similar patches existing in the searching window, and corresponding entries are labeled as N/A in the table. Note that run time increases with both n or R, with R having more prominent impacts, while PSNR exhibits a slower change rate. In these experiments, we choose $n = 4$, $R = 4$ for all 5 × 5 datasets considering the trade-off between performance and complexity. When the number of views are reduced, n and R should be increased accordingly to ensure that there are sufficient number of similar patches. Therefore, for the four smaller datasets that are used to compare with Luo et al. (2013), we set $n = 7$, $R = 10$.

6.2. Comparison with single-image denoising

In the first experiment, we compared the proposed algorithm against some state-of-the-art single-view denoising algorithms including NLM (Buades et al., 2005), BM3D (Dabov et al., 2007), and WNNM (Gu et al., 2014). These results are summarized in Table 3 where the best result (highest PSNR) in each row is highlighted in bold face and the second best result is underlined. The results corresponding to the proposed algorithm is listed in the last column in Table 3.

From Table 3, several observations can be made: 1) The proposed method consistently outperforms single-view denoising algorithms by 2–4 dBs with the only exception when $\sigma = 50$ for the “Tsukuba” dataset; 2) As noise variance σ increases, the PSNR decreases accordingly, which is expected. However, the proposed algorithm remains its performance edge in most cases.

A subjective visual comparison of above results is depicted in Fig. 11 for the case $\sigma = 20$. In this figure, part of each image is enlarged for ease of viewing. Note that the denoised images using the proposed algorithm preserve more edges and textures in the original true image. In contrast, single-image denoising methods tend to over-smoothen these regions, possibly due to the lack of inter-view image redundancy information.

6.3. Comparison with multi-view denoising

Next, we compared our proposed algorithm against other multi-view denoising algorithms, including Miyata's fast multi-view denoising (Miyata et al., 2014), VBM4D (Maggioni et al., 2012), and our previous work Zhou et al. (2017). VBM4D is a video filtering algorithm that

Table 2
PSNR and run time for different values of n and R on “Tsukuba”.

	$R = 3$	$R = 4$	$R = 5$	$R = 6$	$R = 7$	$R = 8$
$n = 1$	33.18 dB / 208 s	33.19 dB / 282 s	33.20 dB / 384 s	33.21 dB / 476 s	33.21 dB / 590 s	33.22 dB / 762 s
$n = 2$	33.45 dB / 222 s	33.46 dB / 298 s	33.47 dB / 388 s	33.48 dB / 486 s	33.48 dB / 622 s	33.49 dB / 768 s
$n = 3$	33.54 dB / 263 s	33.56 dB / 316 s	33.57 dB / 399 s	33.57 dB / 510 s	33.58 dB / 635 s	33.58 dB / 802 s
$n = 4$	N/A / N/A	**33.62 dB / 326 s**	33.62 dB / 403 s	33.63 dB / 518 s	33.63 dB / 652 s	33.63 dB / 810 s
$n = 5$	N/A / N/A	33.63 dB / 357 s	33.64 dB / 432 s	33.65 dB / 528 s	33.65 dB / 667 s	33.66 dB / 817 s
$n = 6$	N/A / N/A	33.64 dB / 373 s	33.65 dB / 449 s	33.66 dB / 539 s	33.66 dB / 696 s	33.66 dB / 819 s
$n = 8$	N/A / N/A	33.61 dB / 375 s	33.64 dB / 459 s	33.65 dB / 553 s	33.65 dB / 724 s	33.66 dB / 841 s

Table 3
Denoising performance (PSNR) compared with other methods (best – **bold**; second best – underlined).

$\sigma = 20$							
Image	NLM	BM3D	WNNM	Miyata et al.	VBM4D	Zhou et al.	Proposed
Tsukuba	29.13	31.41	31.72	28.36	30.41	32.43	**33.63**
Bicycle	27.15	28.54	28.31	27.08	30.42	30.53	**31.55**
Dishes	28.61	29.79	30.44	28.22	30.97	32.59	**33.66**
Knight	28.16	29.87	30.61	28.62	31.54	32.35	**33.57**
Medieval	29.59	30.99	31.05	29.33	32.67	32.70	**33.55**
Sideboard	26.01	27.53	28.59	26.87	29.27	30.40	**31.28**
Tarot	25.11	26.38	26.91	25.53	28.07	29.73	**30.73**
$\sigma = 30$							
Image	NLM	BM3D	WNNM	Miyata et al.	VBM4D	Zhou et al.	Proposed
Tsukuba	27.19	29.25	29.47	25.78	28.22	29.91	**30.82**
Bicycle	25.22	26.21	26.32	24.93	28.33	28.49	**29.41**
Dishes	26.40	27.51	28.19	25.86	28.42	30.01	**31.20**
Knight	25.93	27.55	28.09	26.20	29.25	29.85	**30.97**
Medieval	27.49	29.47	29.59	26.85	30.67	30.46	**31.58**
Sideboard	23.88	25.18	26.16	24.90	26.93	28.01	**28.91**
Tarot	23.07	23.82	24.30	23.36	25.64	27.51	**28.39**
$\sigma = 40$							
Image	NLM	BM3D	WNNM	Miyata et al.	VBM4D	Zhou et al.	Proposed
Tsukuba	25.33	27.73	27.92	24.16	26.71	27.74	**28.51**
Bicycle	23.64	24.75	24.97	23.34	26.86	26.76	**27.88**
Dishes	24.59	25.53	26.59	24.08	26.65	27.97	**29.34**
Knight	24.09	25.74	26.39	24.41	27.58	27.71	**29.11**
Medieval	25.89	28.28	28.34	24.99	29.20	28.49	**30.11**
Sideboard	22.29	23.48	24.33	23.25	25.28	26.24	**27.24**
Tarot	21.40	21.95	22.68	21.87	23.85	25.50	**26.58**
$\sigma = 50$							
Image	NLM	BM3D	WNNM	Miyata et al.	VBM4D	Zhou et al.	Proposed
Tsukuba	24.04	26.54	**26.80**	22.83	25.54	25.82	26.60
Bicycle	22.47	23.69	23.90	22.09	25.72	25.57	**26.64**
Dishes	23.26	24.57	25.43	22.70	25.36	26.20	**26.24**
Knight	22.83	24.45	25.01	22.89	26.31	25.83	**27.47**
Medieval	24.76	27.50	27.48	23.53	28.02	26.81	**28.77**
Sideboard	21.22	22.44	23.12	21.74	24.05	24.81	**26.01**
Tarot	20.20	20.80	21.47	20.64	22.50	23.88	**25.14**

exploits temporal and spatial redundancy in video sequence. In our experiments, we fed in the multiple views as input video sequence.

The results are also summarized in Table 3. Again, the proposed algorithm outperforms these multi-view denoising algorithms with significant margin. Our previous work (Zhou et al., 2017) achieves the second best results in most of these cases with a couple of exceptions. Similar visual comparison in Fig. 11 also reveals the superior visual quality of the denoised images using the proposed algorithms. In particular, existing multi-view denoising algorithms exhibit various kinds of artifacts, mostly due to errors in the disparity map estimation. Near object boundaries, visual quality is also degraded due to occlusion of views or parallax. In comparison, the proposed algorithm is able to reduce the impact of such problems to achieve better performance.

We also compared our proposed algorithm with a recent multi-view adaptive NLM algorithm proposed by Luo et al. (2013). Since we have no access to the code of Luo et al. (2013), we experimented on a different set of images (i.e. “Barn”, “Cones”, “Teddy”, “Venus”) that were used in Luo et al. (2013) for the comparison. The results are listed in Table 4, in which the proposed algorithm shows a slight advantage over the adaptive NLM algorithm with the margin varying between 0.3 to 1 dB.

In Table 5, we have also demonstrated the PSNR improvement of each contribution proposed in Sections 4 and 5 using the “Tsukuba” dataset. Starting from our previous work (Zhou et al., 2017), the PSNR increases 0.44 dB with the implementation of improved disparity map, 0.42 dB with the use of the robust similarity metric, and another 0.34 dB when the low rank minimization scheme is used. These improvements yields a total of 1.2 dB PSNR improvement over our previous work.

6.4. Impacts of accuracy of disparity map on denoising performance

The accuracy of the disparity map estimated from the noisy images has significant impact on the denoising performance. If a disparity value is estimated incorrectly, the denoising algorithm is prone to extract the wrong focus image stack and further denoising procedure is likely to make errors when matching similar patches. To illustrate such an impact, we applied the proposed denoising algorithm to “Tsukuba” dataset using three different disparity maps: the ground truth, the proposed disparity map estimated in Section 4, and the one implemented by Miyata et al. (2014).

In Fig. 12, the three chosen disparity maps with their corresponding denoised images are depicted. Two specific regions, indicated by red and green boxes, are selected and enlarged for a closer inspection. Focusing on the red box that covers a depth transition at the boundary of the lamp, the proposed algorithm better captures the disparity values than Miyata's method, and hence produces cleaner denoised image. On the other hand, for the green box that covers the thin arm of the lamp, disparity maps estimated from both the proposed and Miyata's methods are not satisfactory, due to the similar intensity with the background. As a result, part of the lamp arm is missing in both denoised images. However, in general, the proposed disparity map yields a denoised result closer to that obtained using the ground truth, especially along the disparity discontinuities, thanks to better handling of occlusion.

Meanwhile, the PSNR values of the denoised images, excluding the dark borders surrounding the ground truth disparity map, are also listed at the bottom of the figure. In these border regions, the ground truth disparity values are unavailable. Excluding such regions allows a fair comparison of the impacts of these three different disparity maps. Quantitatively, as indicated in Fig. 12, the denoising performance using the proposed disparity is slightly inferior to the one using ground truth, but still better than that using Miyata's disparity map. The experiment suggests that the denoising procedure does not require perfect disparity map to achieve satisfactory denoising performance in most regions, and slight bias in disparity map does not affect the visual appearance of final results too much. Therefore, our proposed algorithm has certain tolerance to bias accumulation from disparity estimation.

6.5. Impacts of number of views on denoising performance

Here we investigate the question: how does the number of views affect the denoising performance? Zhang et al. (2009) mentioned that the PSNR steadily improves as the number of views increases until 15–20, after which it flattens. We also conducted experiments using camera arrays of different sizes ranging from 2 × 2 to 9 × 9. The PSNR corresponding to different camera array sizes on four multi-view image datasets, “Bicycle”, “Dishes”, “Knight”, and “Sideboard”, when $\sigma = 20$ are displayed in Fig. 13. To focus on the impacts on the denoising procedure exclusively, we use the same disparity map estimated using 5 × 5 array for denoising on all the experiments.

As shown in Fig. 13, the plots of PSNR versus array size curves for the four datasets are very similar. The PSNR values increase initially as more views are used and then start decreasing after reaching a maximum. Three of the curves reach the maximum for an array size of 5 × 5 and one reaches the maximum with a 7 × 7 array size. The initial increase of PSNR values as number of views increases is easily understood: more views imply that more candidate similar patches may be

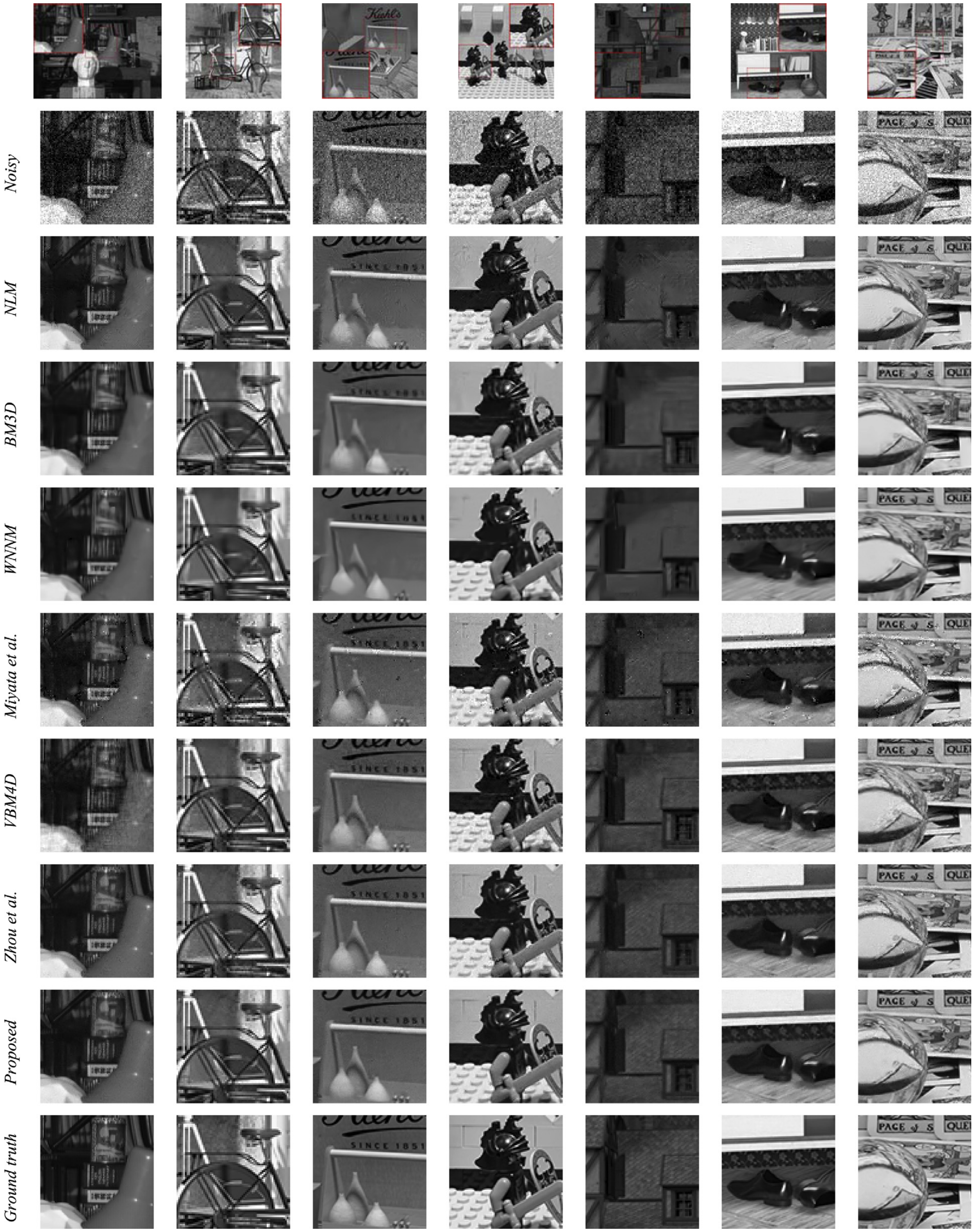


Fig. 11. Qualitative comparison of different denoising methods when $\sigma = 20$.

selected and hence lead to an increasing PSNR. However, as the number of views continue to increase, the reason that the PSNR values start to flatten or decrease needs further investigation.

One of the plausible explanations is that as more views are included in the multi-view system, the side views that are too far from the reference view tend to have large translation such that there are less overlapping regions between the distant views, i.e. less redundant information across views, which is the key to multi-view denoising. Moreover, more regions in one view could be occluded from another view due to parallax. This large disparity with more occluded regions might also have negatively affected the denoising quality. To confirm our conjecture on the distant views, we conducted experiments on the original dataset "Knight", which has 17 × 17 views (i.e. $s, t = 1, 2, \ldots, 17$). In experiment 1, we selected 5 × 5 subset of the views that are

Table 4
Denoising performance (PSNR) compared with Luo et al. (2013).

Image	BM3D	Luo et al.	Proposed
Barn	28.97	30.67	**31.23**
Cones	28.90	30.04	**30.34**
Teddy	30.18	31.11	**32.13**
Venus	30.51	32.00	**32.66**

Table 5
PSNR (dB) improvement of each contribution (step-by-step improvement shown in parentheses).

Step	Original method of Zhou et al. (2017)	Improved disparity map	Robust similarity metric	Low rank minimization (proposed)
PSNR	32.43	32.87 (0.44)	33.29 (0.42)	33.63 (0.34)

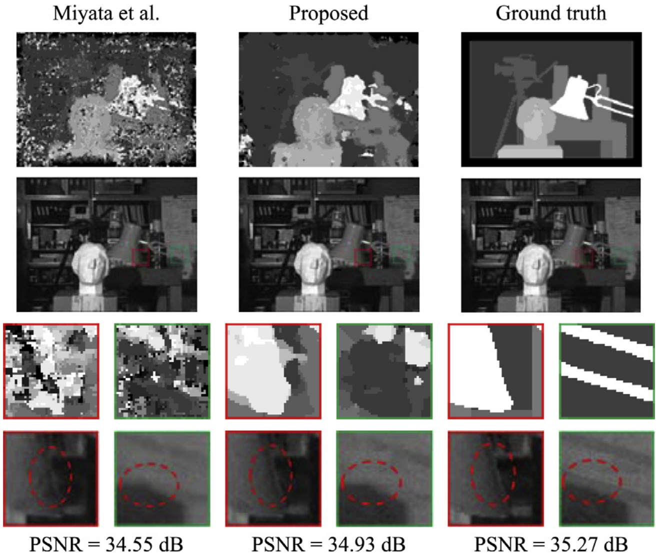


Fig. 12. Denoising performance using different disparity maps. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

close to each other (s, t = 7, 8, 9, 10, 11), with the reference view being at $s_0 = 9$, $t_0 = 9$. In experiment 2, the reference view was kept unchanged, and a same number of more distant views (s, t = 5, 7, 9, 11, 13) were used. In experiment 3, views that are even farther apart (s, t = 1, 5, 9, 13, 17) were selected. The comparison of the three experiments is shown in Table 6, which indicates that faraway views indeed tend to produce inferior denoising results than close views.

6.6. Computational cost

The complexity of the proposed denoising algorithm is $O(NKnR^2r^2)$, where N is the number of pixels in the images, K is the total number of views, n is the number of similar patch volumes, and r, R are radius (half size) of patch and searching window. The book-keeping strategy in window selection reduced computational cost by a great amount by avoiding a lot of repeated computations, including searching for similar patches and performing SVD. Table 7 shows the run time of different

Table 6
Denoising performance using close and distant views on “Knight”.

Experiment	1 (close)	2	3 (distant)
PSNR	33.67 dB	33.34 dB	32.23 dB

Table 7
Run time of different denoising methods on “Tsukuba”.

Algorithm	WNNM	Miyata et al.	Zhou et al.	Proposed
Run time	322 s	12 s	1725 s	326 s
PSNR	31.72 dB	28.36 dB	32.43 dB	33.63 dB

Algorithm 1
Disparity estimation.

```
Input: 3DFIS $F^d$, $d = 1, \dots, d_{\max}$
Output: Disparity map $d_{est}$
for $d = 1$: $d_{\max}$
  for each pixel location $(x, y)$
    Compute $\sigma^d(x, y)$ using Eq. (14);
  end
  Apply Gaussian filter to $\sigma^d$;
end
Estimate texture map $\Sigma$ using Eq. (15);
Estimate patch size $L$ and number of views $V$ using Eqs. (16) and (17);
for $d = 1$: $d_{\max}$
  for each pixel location $(x, y)$
    Compute matching cost $C^*(x, y, d)$ using Eq. (12), with $h = V(x, y)$, and patch size = $L(x, y)$;
  end
end
Compute estimated disparity map $d_{est}$ using Eq. (13);
```

Algorithm 2
Multi-view image denoising.

```
Input: 3DFIS $F^d$, $d = 1, \dots, d_{\max}$, disparity map $d_{est}$
Output: Denoised image for reference view $I_{est}$
for each pixel location $(x, y)$
    Find its disparity value $d(x, y)$ and corresponding 3DFIS $F^d$;
    Compute average disparity and median RMSE using Eqs. (19) and (20) for each candidate window ($j$ = 0, 1, 2, 3, 4);
  Follow procedure C1, C2 to find the best adaptive window $j^*$;
    if the window has been recorded then
      Skip to the next pixel location;
  end
  Search for similar patches $\mathbf{p} = \{p_1, \dots, p_n\}$ using similarity metric in Eq. (23);
  Exclude outliers from $\mathbf{p}$;
    Compute SVD of $\mathbf{p} = USV^T$;
    Compute $S_\lambda$ using Eq. (29);
    Estimate the denoised patches $\mathbf{p}' = \{p'_1, \dots, p'_n\}$ using Eq. (28);
    Compute denoised patch using Eq. (30);
    Record the window by book-keeping its center pixel location;
end
    Aggregate denoised patches to get denoised image $I_{est}$;
```

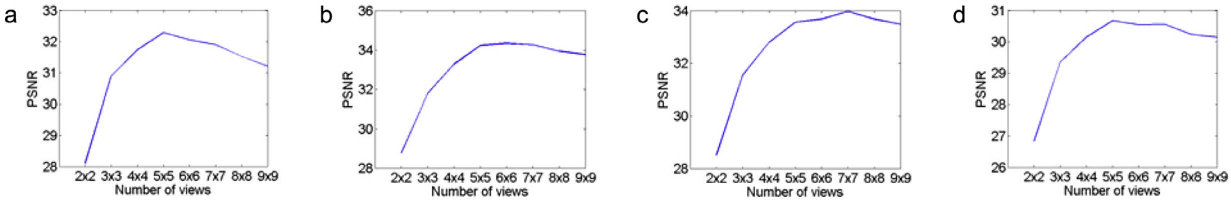


Fig. 13. PSNR vs number of views on dataset (a) “Bicycle”; (b) “Dishes”; (c) “Knight”; (d) “Sideboard”.

denoising algorithms that are written in MATLAB on the 5 × 5 “Tsukuba” dataset on our machine. As can be observed, the proposed algorithm achieves a computational time of 326 s. This is comparable with the state-of-the-art single-image denoising algorithm WNNM (Gu et al., 2014), while exhibiting relatively better denoising performance. Although Miyata's algorithm (Miyata et al., 2014) is faster, the corresponding denoising performance is much worse than the proposed algorithm, due to the over-simplified procedures handling occlusions. Compared to our previous approach Zhou et al. (2017), the proposed algorithm reduces the run time by about 81% while yielding higher PSNR. As discussed in Section 6.3, these performance enhancements are primarily due to the innovations introduced in this work, namely, more accurate disparity map estimation, robust similarity metric, and the low-rank minimization denoising procedure.

7. Conclusion

In this paper, we proposed a multi-view image denoising algorithm based on 3DFIS with improved disparity estimation and enhanced denoising performance. The 3DFIS contains important inter-view image redundancy of the scene that is the key for robust disparity map estimation under the noise. A multi-view low-rank based denoising algorithm is empowered with robust 3DFIS-based similarity metric and enhanced occlusion handling techniques using depth-guided adaptive window selection. These improvements together demonstrated significant denoising performance enhancement over existing approaches. Future work will focus on a more integrated approach that can simultaneously aim at understanding 3D structure of the scene and achieving denoising. Moreover, the theoretical impact of number of views on the denoising performance will be thoroughly investigated.

Acknowledgment

This work was supported by the U.S. National Science Foundation through the cyber-physical systems program [grant number: CNS 1329481].

Appendix A. Proof of $C^*(x, y, d) \geq C(x, y, d)$

In Eq. (12), if $h = K$, then

$$C^*(x, y, d) = \frac{1}{n_p(K-1)} \sum_{k'=2}^{K} \|\tilde{\mathbf{v}}_{k'}^{d}\|_1 = \frac{1}{n_p(K-1)} \sum_{k'=2}^{K} \|\mathbf{v}_k^d - \mathbf{v}_1^d\|_1 \tag{A.1}$$

Eq. (9), if written in terms of patch vectors $\mathbf{v}_k^d$, can be expressed as

$$\begin{aligned} C(x, y, d) &= \frac{1}{n_p} \left\| \frac{1}{K-1} \sum_{k=2}^{K} \mathbf{v}_k^d - \mathbf{v}_1^d \right\|_1 = \frac{1}{n_p} \left\| \frac{1}{K-1} \sum_{k=2}^{K} \mathbf{v}_k^d - \frac{1}{K-1} \sum_{k=2}^{K} \mathbf{v}_1^d \right\|_1 = \frac{1}{n_p(K-1)} \left\| \sum_{k=2}^{K} (\mathbf{v}_k^d - \mathbf{v}_1^d) \right\|_1 \\ &\leq \frac{1}{n_p(K-1)} \sum_{k=2}^{K} \|\mathbf{v}_k^d - \mathbf{v}_1^d\|_1 = C^*(x, y, d) \end{aligned} \tag{A.2}$$

The last inequality comes from the theorem that sum of absolute values is always greater than or equal to the absolute value of sum, i.e. $|x| + |y| \geq |x + y|$, with equality holds when $x \geq 0$, $y \geq 0$. In our multi-view scenario, it means that the equality holds when each element of patch vector $\mathbf{v}_k^d$ is greater than or equal to the corresponding element of $\mathbf{v}^d{}_1$, which is possible only when d is the true disparity, i.e. $\mathbf{v}_k^d = \mathbf{v}^d{}_1$ for $k = 2, \ldots, K$.

References

Aharon, M., Elad, M., Bruckstein, A., 2006. K-SVD: an algorithm for designing overcomplete dictionaries for sparse represention. IEEE Trans. Image Process. 54 (11), 4311–4322.

Buades, A., Coll, B., Morel, J., 2005. A non-local algorithm for image denoising. In: Proc. IEEE Conf. Comput. Vision Pattern Recognit., June, pp. 60–65.

Cai, J.F., Candès, E.J., Shen, Z., 2010. A singular value thresholding algorithm for matrix completion. SIAM J. Optim. 20 (4), 1956–1982.

Collins, R.T., 1996. A space-sweep approach to true multi-image matching. In: Proc. IEEE Conf. Comput. Vision Pattern Recognit., June, pp. 358–363.

Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K., 2007. Image denoising by sparse 3-D transform-domain collaborative filtering. IEEE Trans. Image Process. 16 (8), 2080–2095.

Elad, M., Aharon, M., 2006. Image denoising via sparse and redundant representations over learned dictionaries. IEEE Trans. Image Process. 15 (12), 3736–3745.

Eslami, R., Radha, H., 2003. The contourlet transform for image denoising using cycle spinning. In: Proc. Asilomar Conf. on Signals, Systems and Computers, vol. 2, November, pp. 1982–1986.

Furukawa, Y., Ponce, J., 2010. Accurate, dense, and robust multiview stereopsis. IEEE Trans. Pattern Anal. Mach. Intell. 32 (8), 1362–1376.

Gu, S., Zhang, L., Zuo, W., Feng, X., 2014. Weighted nuclear norm minimization with application to image denoising. In: Proc. IEEE Conf. Comput. Vision Pattern Recognit. pp. 2862–2869.

H. Hu, J. Froment, and Q. Liu, “Patch-based low-rank minimization for image denoising,” arXiv preprint arXiv:1506.08353, June 2015.

Ji, H., Liu, C., Shen, Z., Xu, Y., June 2010. Robust video denoising using low rank minimization using low rank matrix completion. In: Proc. IEEE Conf. Comput. Vision Pattern Recognit, pp. 1791–1798.

Kang, S.B., Szeliski, R., Chai, J., 2001. Handling occlusions in dense multi-view stereo. In: Proc. IEEE Conf. Comput. Vision Pattern Recognit., June. 1. pp. I–103.

Kanhere, A., Van Grinsven, K.L., Huang, C.C., Lu, Y.S., Greensberg, J.A., Heise, C.P., Hu, Y.H., Jiang, H., 2014. Multicamera laparoscopic imaging with tunable focusing capability. J. Microelectromech. Syst. 23 (6), 1290–1299.

Klaus, A., Sormann, M., Karner, K., 2006. Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. In: Proc. 18th IEEE Int. Conf. Pattern Recognit. 3. pp. 15–18.

Kodama, K., Kubota, A., 2014. Linear view/image restoration for dense light fields. In: Proc. IEEE Int. Conf. Image Process., October, pp. 5462–5466.

Lee, S., Lee, J.H., Lim, J., Suh, H.I., 2015. Robust stereo matching using adaptive random walk with restart algorithm. Image Vis. Comput. 37, 1–11.

Leys, C., Ley, C., Klein, O., Bernard, P., Licata, L., 2013. Detecting outliers: do not use standard deviation around the mean, use absolute deviation around the median. J. Exp. Soc. Psycol. 49 (4), 764–766.

4D Light Field Benchmark [Online]. Available: http://hci-lightfield.iwr.uni-heidelberg.de/.

Liu, C., Freeman, W.T., Szeliski, R., Kang, S.B., 2006. Noise estimation from a single image. In: Proc. IEEE Conf. Comput. Vision Pattern Recognit. 1. pp. 901–908.

Liu, X., Tanaka, M., Okutomi, M., 2013. Single-image noise level estimation for blind denoising. IEEE Trans. Image Process. 22 (12), 5226–5237.

Luo, E., Chan, S.H., Pan, S., Nguyen, T., 2013. Adaptive non-local means for multiview image denoising: searching for the right patches via a statistical approach. In: Proc. IEEE Int. Conf. Image Process., September, pp. 543–547.

Maggioni, M., Boracchi, G., Foi, A., Egiazarian, K., 2012. Video denoising, deblocking, and enhancement through separable 4-D nonlocal spatiotemporal transforms. IEEE Trans. Image Process. 21 (9), 3952–3966.

Middlebury Multi-view Stereo Datasets [Online]. Available: http://vision.middlebury.edu/stereo.

Miyata, M., Kodama, K., Hamamoto, T., 2014. Fast multiple-view denoising based on image reconstruction by plane sweeping. In: IEEE Conf. Visual Commun. Image Process., December, pp. 462–465.

Nakamura, Y., Matsuura, T., Satoh, K., Ohta, Y., 1996. Occlusion detectable stereo-occlusion patterns in camera matrix. In: Proc. IEEE Conf. Comput. Vision Pattern Recognit., June, pp. 371–378.

Portilla, J., Strela, V., Wainwright, M.J., Simoncelli, E.P., 2003. Image denoising using scale mixture of gaussians in the wavelet domain. IEEE Trans. Image Process. 12 (11), 1338–1351.

Rank, K., Lendl, M., Unbehauen, R., 1999. Estimation of image noise variance. In: IEE Proc. Vision, Image and Signal Process. 146. pp. 80–84.

Shapiro, L., Stockman, G., 2001. Computer Vision. Prentice-Hall, Englewood Cliffs, NJ, USA.

Takahashi, K., Naemura, T., 2006. Layered light-field rendering with focus measurement. Signal Process. 21 (6), 519–530.

T. Taniai, Y. Matsushita, Y. Sato, and T. Naemura, “Continuous stereo matching using local expansion moves,” arXiv preprint arXiv:1603.08328, March 2016.

Tao, H., Sawhney, H.S., Kumar, R., 2001. A global matching framework for stereo computation. In: Proc. 8th IEEE Int. Conf. Comput. Vision. 1. pp. 532–539.

The (New) Stanford Light Field Archive [Online]. Available: http://lightfield.stanford.edu/lfs.html.

Thite, A., Zhang, L., May 2014. A Parallel Algorithm for Multi-View Image Denoising. Department of Computer Science, University of Wisconsin-Madison, Madison, WI Tech. Rep. TR1809.

Tomasi, C., Manduchi, R., 1998. Bilateral filtering for gray and color images. In: Proc. 6th Int. Conf. Comput. Vision, pp. 839–846.

Widrow, B., Haykin, S., 2003. Least-Mean-Square Adaptive Filters. Wiley-IEEE, New York, NY, USA.

Wiener, N., 1949. Extrapolation, Interpolation, and Smoothing of Stationary Time Series. Wiley, New York, NY, USA.

Xue, Z., Yang, J., Dai, Q., Zhang, N., June 2010. Multi-view image denoising based on graphical model of surface patch. In: 3DTV-Conference, pp. 1–4.

Yan, R., Shao, L., Liu, Y., 2013. Nonlocal hierarchical dictionary learning using wavelets for image denoising. IEEE Trans. Image Process. 22 (12), 4689–4698.

Yang, G.Z., Burger, P., Firmin, D.N., Underwood, S.R., 1995. Structure adaptive anisotropic image filtering. In: Proc. IEEE Int. Conf. Image Process. Applicat, pp. 717–721.

Yue, H., Sun, X., Yang, J., Wu, F., 2015. Image denoising by exploring external and internal correlations. IEEE Trans. Image Process. 24 (6), 1967–1982.

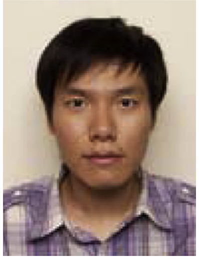
Zhang, C., 2007. Multiview imaging and 3DTV. IEEE Signal Process. Mag. 1053 (5888/07).

Zhang, L., Dong, W., Zhang, D., Shi, G., 2010. Two-stage image denoising by principal component analysis with local pixel grouping. Pattern Recognit. 43 (4), 1531–1549.

Zhang, L., Vaddadi, S., Jin, H., Nayar, S., 2009. Multiple view image denoising. In: Proc. IEEE Conf. Comput. Vision Pattern Recognit., June, pp. 1542–1549.

Zhou, S., Hu, Y.H., Jiang, H., 2015. Multiple view image denoising using 3D focus image stacks. In: IEEE Global Conf. Signal Info. Process. (GlobalSIP), pp. 1052–1056.

Zhou, S., Hu, Y.H., Jiang, H., 2017. Patch-based multiple view image denoising with occlusion handling. In: Proc. IEEE Int. Conf. Accoustic, Speech, Signal Process. (ICASSP), pp. 1782–1786.



Shiwei Zhou received B.Eng. degree from The Hong Kong Polytechnic University, Hong Kong, in 2010, and M.S. degree from University of Colorado, Boulder, CO, USA, in 2012, respectively. He is currently pursuing Ph.D. degree at the Department of Electrical and Computer Engineering at University of Wisconsin, Madison, WI, USA. His research interests include multiple view image enhancement and restoration for flexible micro-camera array.

Zhengyang Lou received B. S. from the University of Wisconsin, Madison WI, USA, in 2017. He is now pursuing M. S. degree at the University of Wisconsin-Madison in signal processing. His research interests include multiple view images denoising, disparity estimation, and super resolution.

Yu Hen Hu received BSEE degree from National Taiwan University, Taiwan ROC in 1976, and Ph.D. degree from University of Southern California, Los Angeles, CA, USA in 1982. He was a faculty member in the Electrical Engineering Department of Southern Methodist University, Dallas, Texas from 1983 to 1987. Since 1987, he has been with the Department of Electrical and Computer Engineering, University of Wisconsin, Madison where he is currently a professor. Dr. Hu's research interests range from design and implementation of signal processing algorithms, computer aided design and physical design of VLSI, pattern classification and machine learning algorithms, and image and signal processing in general.

Hongrui Jiang received the B.S. degree in physics from Peking University, Beijing, China, and the M.S. and Ph.D. degrees in electrical engineering from Cornell University, Ithaca, NY, in 1999 and 2001, respectively. From 2001 to 2002, he was a Postdoctoral Researcher with the Berkeley Sensor and Actuator Center, University of California at Berkeley. He is currently the Vilas Distinguished Achievement Professor and the Lynn H. Matthias Professor in Engineering with the Department of Electrical and Computer Engineering, a Faculty Affiliate with the Department of Biomedical Engineering, the Department of Materials Science and Engineering, and the Department of Ophthalmology and Visual Sciences, and a Member of the McPherson Eye Research Institute with the University of Wisconsin–Madison. His research interests are in microfabrication technology, biological and chemical microsensors, microactuators, optical microelectromechanical systems, smart materials and micro-/nanostructures, lab-on-chip, and biomimetics and bioinspiration.