ADAPTIVE NON-LOCAL MEANS FOR MULTIVIEW IMAGE DENOISING: SEARCHING FOR THE RIGHT PATCHES VIA A STATISTICAL APPROACH

Enming Luo[1*], *Stanley H. Chan* [2], *Shengjun Pan*[1] *and Truong Q. Nguyen*[1]

[1]University of California, San Diego, Dept of ECE, 9500 Gilman Drive, La Jolla, CA 92092.
[2]Harvard School of Engineering and Applied Sciences, 33 Oxford Street, Cambridge, MA 02138.

ABSTRACT

We present an adaptive non-local means (NLM) denoising method for a sequence of images captured by a multiview imaging system, where direct extensions of existing single image NLM methods are incapable of producing good results. Our proposed method consists of three major components: (1) a robust joint-view distance metric to measure the similarity of patches; (2) an adaptive procedure derived from statistical properties of the estimates to determine the optimal number of patches to be used; (3) a new NLM algorithm to denoise using only a set of similar patches. Experimental results show that the proposed method is robust to disparity estimation error, out-performs existing algorithms in multiview settings, and performs competitively in video settings.

Index Terms— Non-local means, adaptive filtering, multiview denoising, patch-based denoising

1. INTRODUCTION

We consider the problem of denoising an image using multiple but non-identical observations captured by a multiview system. Defining $\boldsymbol{g}^{(k)} = \boldsymbol{f}^{(k)} + \boldsymbol{n}^{(k)}$ for $k = 1, \ldots, K$ a sequence of K noisy observations, where $\boldsymbol{n}^{(k)}$ is a vector of i.i.d. Gaussian noise of variance σ^2, our goal is to recover $\boldsymbol{f}^{(1)}, \ldots, \boldsymbol{f}^{(K)} \in \mathbb{R}^{N\times 1}$ from $\boldsymbol{g}^{(1)}, \ldots, \boldsymbol{g}^{(K)} \in \mathbb{R}^{N\times 1}$.

The method we consider in this paper is the NLM [1] filtering approach. Letting $\boldsymbol{p}_i^{(k)} \in \mathbb{R}^{P\times 1}$ be a patch centered at the ith pixel in the kth image, NLM computes the weights as

$$W_{ij}^{(kl)} = \exp\left\{-\frac{\|\boldsymbol{p}_i^{(k)} - \boldsymbol{p}_j^{(l)}\|^2}{h^2}\right\}, \tag{1}$$

and denoises the images as

$$\begin{bmatrix}\widehat{\boldsymbol{f}}^{(1)}\\ \vdots \\ \widehat{\boldsymbol{f}}^{(K)}\end{bmatrix} = \begin{bmatrix}\boldsymbol{D}^{(1)} & & \\ & \ddots & \\ & & \boldsymbol{D}^{(K)}\end{bmatrix}^{-1} \begin{bmatrix}\boldsymbol{W}^{(11)} & \ldots & \boldsymbol{W}^{(1K)}\\ \vdots & \ddots & \vdots \\ \boldsymbol{W}^{(K1)} & \ldots & \boldsymbol{W}^{(KK)}\end{bmatrix} \begin{bmatrix}\boldsymbol{g}^{(1)}\\ \vdots \\ \boldsymbol{g}^{(K)}\end{bmatrix}$$

Here, $\boldsymbol{D}^{(k)} = \text{diag}([\boldsymbol{W}^{(k1)}, \ldots, \boldsymbol{W}^{(kK)}]^T \boldsymbol{1}_{KN\times 1})$ is a diagonal matrix for normalization. For notation simplicity and without loss of generality, in the rest of the paper we consider denoising $\boldsymbol{f}^{(1)}$ using $\boldsymbol{g}^{(1)}, \ldots, \boldsymbol{g}^{(K)}$.

One of the biggest issues in the above multiple-image NLM is that the Euclidean distance metric $\|\boldsymbol{p}_i^{(k)} - \boldsymbol{p}_j^{(l)}\|^2$ does not capture the true similarity between $\boldsymbol{p}_i^{(k)}$ and $\boldsymbol{p}_j^{(l)}$. In other words, two patches could be different, but their Euclidean distance might be close because of noise. Therefore, to improve the performance of NLM, one fundamental challenge is how to search for the *right* patches.

1.1. Overview of the Proposed Method

To tackle the problem, our first attempt is to improve the metric. In multiview denoising problems, we consider the robust joint-view distance [2] to measure the similarity. Given a patch $\boldsymbol{p}_i^{(1)}$, we first compute the correspondences (disparity maps) from view 1 to views $2, \ldots, K$ using off-the-shelf algorithms such as block matching [3], optical flow [4], or TV/L1 [5]. Denoting the disparities as $\{q_i^{(1)}, \ldots, q_i^{(K)}\}$ for $i = 1, \ldots, N$, we define the distance between two patches $\boldsymbol{p}_i^{(1)}$ and $\boldsymbol{p}_j^{(1)}$ as

$$D(\boldsymbol{p}_i^{(1)}, \boldsymbol{p}_j^{(1)}) \stackrel{\text{def}}{=} \sum_{k=1}^{K} \left\| \boldsymbol{p}_{i+q_i^{(k)}}^{(k)} - \boldsymbol{p}_{j+q_j^{(k)}}^{(k)} \right\|^2. \tag{2}$$

Replacing $\|\boldsymbol{p}_i^{(1)} - \boldsymbol{p}_j^{(1)}\|^2$ by $D(\boldsymbol{p}_i^{(1)}, \boldsymbol{p}_j^{(1)})$ could improve the robustness of patch similarity, but as observed by Kervrann and Boulanger [6], the presence of many dissimilar patches could still cause unwanted bias in the denoising procedure. Therefore, our second modification is to keep a set of m most similar patches so that there are only m non-zero entries in each row of $\boldsymbol{W}^{(kl)}$.

To obtain a set of m most similar patches (w.r.t. $\boldsymbol{p}_i^{(1)}$), we compute $D(\boldsymbol{p}_i^{(1)}, \boldsymbol{p}_j^{(1)})$ for all possible j's and keep the m best matches. We then define the set of m best matching patches in View 1:

$$\Omega_{i,m}^{(1)} = \left\{ j \mid \text{the indices of } m \text{ smallest } D(\boldsymbol{p}_i^{(1)}, \boldsymbol{p}_j^{(1)}) \right\}, \tag{3}$$

and the sets of m best matching patches in other views:

$$\Omega_{i,m}^{(k)} = \left\{ j + q_j^{(k)} \mid j \in \Omega_{i,m}^{(1)} \right\}, \quad k = 2, \ldots, K. \tag{4}$$

Note that $\Omega_{i,m}^{(k)} \subset \Omega_{i,m+1}^{(k)}$. Restricting $W_{ij}^{(kl)}$ to $\Omega_{i,m}^{(k)}$ yields the following denoising algorithm:

$$\widehat{f}_{i,m}^{(1)} = \frac{\sum_{k=1}^{K} \sum_{j\in\Omega_{i,m}^{(k)}} W_{ij}^{(1k)} g_j^{(k)}}{\sum_{k=1}^{K} \sum_{j\in\Omega_{i,m}^{(k)}} W_{ij}^{(1k)}} \stackrel{\text{def}}{=} \sum_{k=1}^{K} \sum_{j\in\Omega_{i,m}^{(k)}} \widetilde{W}_{ij}^{(1k)} g_j^{(k)}, \tag{5}$$

where $\widetilde{W}_{ij}^{(1k)} \stackrel{\text{def}}{=} W_{ij}^{(1k)} / \sum_{k=1}^{K} \sum_{j\in\Omega_{i,m}^{(k)}} W_{ij}^{(1k)}$.

*Contact author: E. Luo eluo@ucsd.edu. E. Luo and T. Nguyen are supported, in part, by NSF under grant CCF-1160832. S. Chan is supported, in part, by Croucher Foundation Postdoctoral Research Fellowship (2012-2013).

It can be seen that the performance of the above denoising procedure in (5) depends on how m is chosen. Intuitively, m should not be too small or too large, for otherwise insufficient patches or excessive dissimilar patches will be included. Therefore, we propose an adaptive scheme to choose the optimal m. Specifically, we increase from $m-1$ to m until one of the following criteria is met

- (Condition 1) Denoising Consistency:

$$\left|\widehat{f}_{i,m}^{(1)} - \widehat{f}_{i,m'}^{(1)}\right| \geq \gamma, \tag{6}$$

  for any $m' = 1, \ldots, m-1$, which requires that the deviation between the current and the previous estimates to be small.

- (Condition 2) Intersection of Confidence Interval:

$$\bigcap_{t=1}^{m} [\alpha_{i,t},\ \beta_{i,t}] = \emptyset, \tag{7}$$

  which requires that the confidence intervals $[\alpha,\ \beta]$ of the current and previous estimate intersect.

As we shall see, this adaptive scheme will give us a set of m most similar patches that improve the denoising quality.

1.2. Related Works

The literature on single image denoising is rich. However, state-of-the-art single image denoising methods such as NLM [1], BM3D [7], LPG-PCA [8] and many other methods reported in [9] are insufficient for multiview denoising, as these methods assume that similar patches exist at different locations within the image. Extensions of these methods such as [10], [11] are also insufficient for multiview denoising due to similar reasons.

Direct extension of single image denoising methods have been proposed to handle video denoising. In [12], Buades et al. proposed a video denoising method by allowing NLM to search for similar patches in adjacent frames. Similar ideas are applicable to BM3D, yielding the benchmark video denoising method VBM3D [13] and BM4D [14]. One problem of these methods is that displacement across different images is never explicitly used. While the authors of NLM [12], VBM3D [13] and BM4D [14] claim this as an advantage, Liu and Freeman [15] showed that reliable motion vectors are indeed helpful.

Another problem of video NLM [12], VBM3D [13] and BM4D [14] is that the number of patches increases as the number of images increases. This is undesirable because there will be many small but non-zero weights which could reduce the denoising result. Therefore, Kervrann and Boulanger [6] proposed a method to adaptively look for optimal spatial search window size. Later, in [16] they applied similar ideas to videos and demonstrated outstanding performance over other methods [17, 18, 19, 20, 21].

1.3. Contributions and Outline

Our proposed method utilizes the strength of the robust metric proposed in [2], and the spatial adaptivity proposed by Kervrann and Boulanger [6], and V. Katkovnik [22]. The key contributions of this paper are:

- **New Algorithm for Multiple Image Denoising**: Our new denoising scheme in (5) uses only similar patches defined in (3) and (4). As will be discussed in Section 3, the new algorithm out-performs existing methods.
- **Adaptive Neighborhood Selection**: We propose an adaptive scheme to determine the optimal m. The optimal m allows us to denoise the image with the right number of relevant patches, as contrast to classical NLM where all patches are used.

In the following sections, we discuss our proposed method in Section 2 and show experimental results in Section 3. Conclusion is given in Section 4.

2. PROPOSED METHOD

In this section we describe the proposed method. For clarity we present the overall algorithm in Algorithm 1, and discuss the ideas in the following subsections.

Algorithm 1 Proposed Algorithm

1: Input: $\boldsymbol{g}^{(1)}, \ldots, \boldsymbol{g}^{(K)}$.
2: Output: $\widehat{\boldsymbol{f}}^{(1)}$.
3: Pre-denoise $\{\boldsymbol{g}^{(k)}\}$ by single-view methods to obtain $\{\overline{\boldsymbol{g}}^{(k)}\}$.
4: Run optical flows to obtain $\{\boldsymbol{q}^{(1)}, \ldots, \boldsymbol{q}^{(k)}\}$.
5: **for** all i pixels **do**
6: Compute $D(\boldsymbol{p}_i^{(1)}, \boldsymbol{p}_j^{(1)})$ using (2).
7: Compute the sets $\Omega_{i,m}^{(k)}$ using (3) and (4).
8: Compute $\widehat{f}_{i,m}^{(1)}$ using (5).
9: If $\widehat{f}_{i,m}^{(1)}$ does not satisfy Condition 1 or 2, then increase m and repeat Lines 8 – 9.
10: **end for**

2.1. Pre-processing and Optical Flow

In order to compute $D(\boldsymbol{p}_i^{(1)}, \boldsymbol{p}_j^{(1)})$, we first need to determine the disparity maps $\boldsymbol{q}^{(1)}, \ldots, \boldsymbol{q}^{(K)} \in \mathbb{R}^{N\times 2}$. In this paper, we use the classical optical flow [4], with the MATLAB/C++ implementation by Liu [23].

Running optical flow on $\boldsymbol{g}^{(k)}$'s directly is problematic, because $\boldsymbol{g}^{(k)}$'s are noisy images. Therefore, we pre-filter $\boldsymbol{g}^{(k)}$'s to obtain cleaner images before optical flow. The pre-filtering is done using single-image NLM.

2.2. Bias and Variance for Multiview Image NLM

Our proposed adaptive scheme for finding optimal m requires the knowledge of the bias and variance of $\widehat{f}_{i,m}^{(1)}$. To derive the bias and variance of $\widehat{f}_{i,m}^{(1)}$, we substitute $g_i^{(1)} = f_i^{(1)} + n_i^{(1)}$ into (5), and we can show that

$$b_{i,m}^{(1)} \stackrel{\text{def}}{=} \text{Bias}(\widehat{f}_{i,m}^{(1)}) = \sum_{k=1}^{K} \sum_{j\in\Omega_{i,m}^{(k)}} \widetilde{W}_{ij}^{(1k)} f_j^{(1)},$$

$$\left(v_{i,m}^{(1)}\right)^2 \stackrel{\text{def}}{=} \text{Var}(\widehat{f}_{i,m}^{(1)}) = \sum_{k=1}^{K} \sum_{j\in\Omega_{i,m}^{(k)}} \left(\widetilde{W}_{ij}^{(1k)}\right)^2 \sigma^2.$$

We now discuss the conditions in Line 9 of Algorithm 1. For notation simplicity we drop the super-script and let $\widetilde{W}_{ij} = \widetilde{W}_{ij}^{(1k)}$, $\widehat{f}_j = \widehat{f}_j^{(1)}$, $b_{i,m} = b_{i,m}^{(1)}$, and $v_{i,m} = v_{i,m}^{(1)}$.

2.3. Condition 1: Denoising Consistency

The intuition of the denoising consistency is that by increasing $m-1$ to m, the changes $\widehat{f}_{i,m'} - \widehat{f}_{i,m}$ for all $m' = 1, \ldots, m-1$ cannot be too large. In other words, we want the algorithm to terminate when the following probabilistic criterion is satisfied:

$$\Pr\left[\left|\widehat{f}_{i,m'} - \widehat{f}_{i,m}\right| \leq \varepsilon\right] \leq \lambda, \quad m' = 1, \ldots, m-1, \qquad (8)$$

where ε is a threshold, and $\lambda \ll 1$ defines the probability (typically ≈ 0.01). The probability on the left hand side of (8) can be determined through the following proposition.

Proposition 1. *The probability inequality* $\Pr\left[\left|\widehat{f}_{i,m'} - \widehat{f}_{i,m}\right| \leq \varepsilon\right] \leq \lambda$ *holds if and only if*

$$\left|\widehat{f}_{i,m'} - \widehat{f}_{i,m}\right| \leq Q^{-1}\left(\frac{1-\lambda}{2}\right)\Delta v_{i,m'}, \qquad (9)$$

where $Q(\cdot)$ is the Q-function of standard normal distribution, and $\Delta v_{i,m'}^2 \stackrel{\text{def}}{=} v_{i,m}^2 - v_{i,m'}^2$.

Proof. First, we define $\Delta\widehat{f}_{i,m'} = \widehat{f}_{i,m} - \widehat{f}_{i,m'}$. Consequently, the corresponding bias and variance can be defined as

$$\Delta b_{i,m'} \stackrel{\text{def}}{=} b_{i,m} - b_{i,m'} = \sum_{k=1}^{K}\left[\sum_{j\in\Omega_{i,m}^{(k)}} \widetilde{W}_{i,j} f_j - \sum_{j\in\Omega_{i,m'}^{(k)}} \overline{W}_{i,j} f_j\right],$$

$$\Delta v_{i,m'}^2 \stackrel{\text{def}}{=} v_{i,m}^2 - v_{i,m'}^2 = \sum_{k=1}^{K}\left[\sum_{j\in\Omega_{i,m}^{(k)}} \sigma^2\widetilde{W}_{i,j}^2 - \sum_{j\in\Omega_{i,m'}^{(k)}} \sigma^2\overline{W}_{i,j}^2\right],$$

where $\widetilde{W}$ is the normalized version of W (m non-zero entries), and $\overline{W}$ is the normalized version of W (m' non zero entries).

Since $\widehat{f}_{i,m'} = \sum_{k=1}^{K}\sum_{j\in\Omega_{i,m'}^{(k)}} \widetilde{W}_{i,j}(f_j + n_j)$, it can be shown that $\widehat{f}_{i,m'} \sim \mathcal{N}(b_{i,m'}, v_{i,m'}^2)$ and hence $\Delta\widehat{f}_{i,m'} \sim \mathcal{N}(\Delta b_{i,m'}, \Delta v_{i,m'}^2)$. Substitute this into (9) yields

$$\begin{aligned}\Pr\left[\left|\Delta\widehat{f}_{i,m'}\right| \leq \varepsilon\right] &= \Pr\left[\Delta\widehat{f}_{i,m'} \geq -\varepsilon\right] - \Pr\left[\Delta\widehat{f}_{i,m'} \geq \varepsilon\right] \\ &= Q\left(-\frac{-\varepsilon - \Delta b_{i,m'}}{\Delta v_{i,m}}\right) - Q\left(-\frac{\varepsilon - \Delta b_{i,m'}}{\Delta v_{i,m'}}\right).\end{aligned}$$

Assuming that $b_{i,m} = b_{i,m'}$, we have $\Delta b_{i,m'} = 0$ and hence

$$\Pr\left[\left|\Delta\widehat{f}_{i,m'}\right| \leq \varepsilon\right] = 1 - 2Q\left(\frac{\varepsilon}{\Delta v_{i,m'}}\right).$$

Finally, setting $\Pr\left[\left|\Delta\widehat{f}_{i,m'}\right| \leq \varepsilon\right] \leq \lambda$ yields $1-2Q\left(\frac{\varepsilon}{\Delta v_{i,m'}}\right) \leq \lambda$, which in turn requires that

$$\varepsilon \leq Q^{-1}\left(\frac{1-\lambda}{2}\right)\Delta v_{i,m'} \stackrel{\text{def}}{=} \gamma.$$

This implies that $\Pr\left[\left|\Delta\widehat{f}_{i,m'}\right| \leq \varepsilon\right] \leq \lambda$ iff $\left|\Delta\widehat{f}_{i,m'}\right| \leq \gamma$. □

As a consequence of Proposition 1, the algorithm should terminate if $\left|\Delta\widehat{f}_{i,m'}\right| \leq \gamma$. Hence, the optimal m is the smallest integer such that $\left|\Delta\widehat{f}_{i,m'}\right| \geq \gamma$, i.e.,

$$m^* = \underset{m}{\text{argmin}}\left\{\left|\widehat{f}_{i,m} - \widehat{f}_{i,m'}\right| \geq \gamma, \quad 1 \leq m' \leq m\right\}. \qquad (10)$$

2.4. Condition 2: Intersection of Confidence Interval

Our second condition is based on the intuition that m should be increased as long as $\widehat{f}_{i,m}$ has similar confidence interval with $\widehat{f}_{i,m'}$ for $m' = 1, \ldots, m-1$. Since $\widehat{f}_{i,m} \sim \mathcal{N}(b_{i,m}, v_{i,m}^2)$, the confidence interval of $\widehat{f}_{i,m}$ is

$$[\widehat{f}_{i,m} - \tau v_{i,m}, \widehat{f}_{i,m} + \tau v_{i,m}].$$

where τ controls the likelihood that the true value f_i lies in the interval:

$$\alpha_{i,m} \stackrel{\text{def}}{=} \widehat{f}_{i,m} - \tau v_{i,m} \leq f_i \leq \widehat{f}_{i,m} + \tau v_{i,m} \stackrel{\text{def}}{=} \beta_{i,m}. \qquad (11)$$

Therefore, the smallest intersection before having no feasible solution determines the optimal m:

$$m^* = \underset{m}{\text{argmin}}\left\{m \;:\; \bigcap_{t=1}^{m}[\alpha_{i,t},\ \beta_{i,t}] = \emptyset\right\}. \qquad (12)$$

3. EXPERIMENTAL RESULTS

In this section we present the experimental results for multiview image denoising and video denoising using Algorithm 1. Due to limited space, additional results could be found at `http://videoprocessing.ucsd.edu/~eluo/`.

3.1. Multi-view Image Denoising

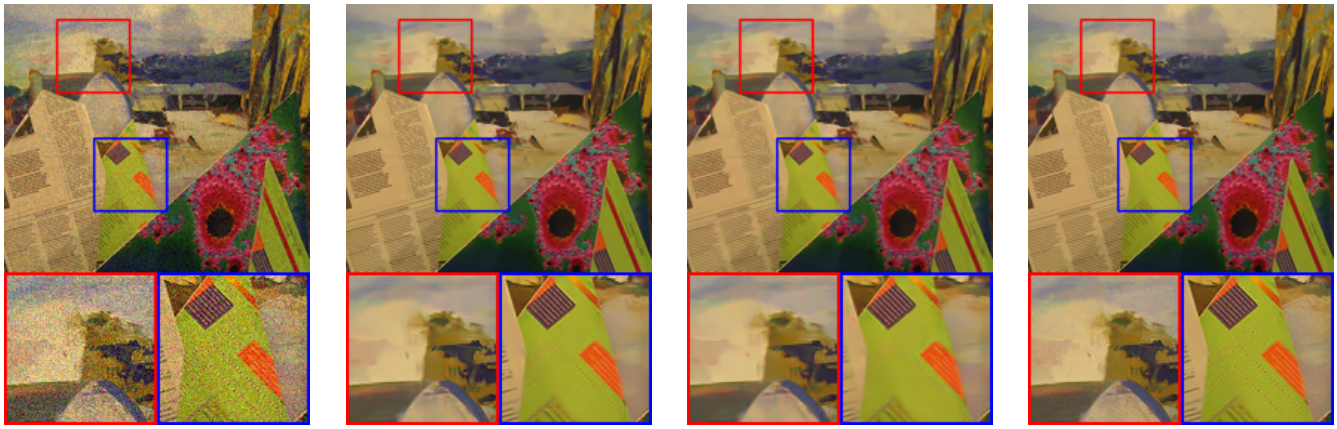
We downloaded 4 sets of images from Middlebury Computer Vision Page `http://vision.middlebury.edu/stereo/`. Each set of images consists of 5 views, so that $K = 5$. In our experiments, we add noise of variance $\sigma = 20$ (out of 255).

We compared our proposed algorithm with four existing methods, namely Video NLM [24], BM3D [7], Video BM3D [13] and BM4D [14]. The results are shown in Table 1 and snapshots of the images are shown in Figure 1.

	Barn	Cone	Teddy	Venus
Video NLM [24]	28.95 (0.8363)	28.81 (0.8358)	29.95 (0.8688)	30.55 (0.8941)
BM3D [7]	28.97 (0.8442)	28.90 (0.8443)	30.18 (0.8844)	30.51 (0.9038)
Video BM3D [13]	30.38 (0.8840)	28.42 (0.8353)	29.70 (0.8755)	31.54 (0.9192)
BM4D [14]	28.70 (0.8368)	27.93 (0.8127)	29.28 (0.8575)	29.96 (0.8920)
Ours - Cond 1	**30.67 (0.9000)**	**30.04 (0.8905)**	**31.11 (0.9000)**	**32.00** (0.9189)
Ours - Cond 2	30.48 (0.8946)	29.91 (0.8850)	31.05 (0.8992)	31.91 **(0.9192)**
Ours - Cond 1 (true disparity)	30.74 (0.9034)	30.13 (0.8977)	31.23 (0.9058)	32.08 (0.9224)
Ours - Cond 2 (true disparity)	30.54 (0.8977)	29.98 (0.8915)	31.14 (0.9042)	31.98 (0.9224)

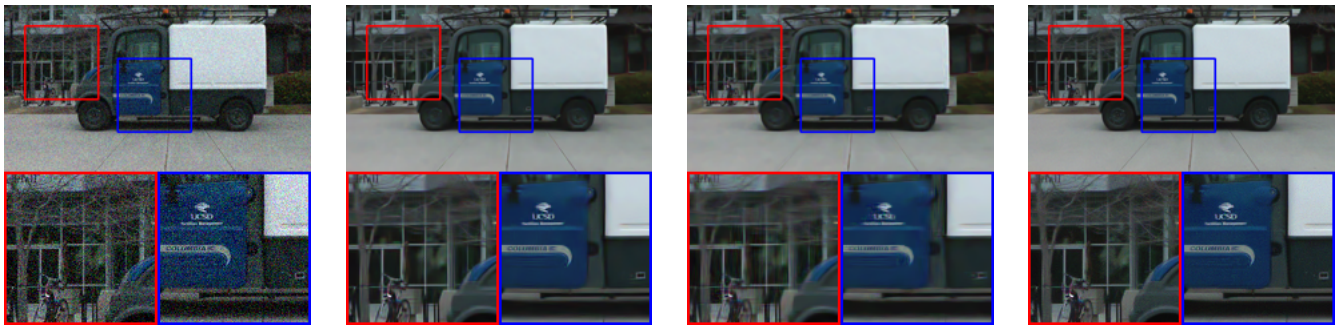
Table 1. PSNR and SSIM (value in the parenthesis) results using Video NLM, BM3D, Video-BM3D, BM4D and the proposed methods for denoising images in Middlebury dataset.

The results suggest the following observations. First, the proposed method (using either conditions) out-performs the competitors by a big margin. Compared with NLM (which uses all patches



(a) Noisy $\sigma = 20$ (b) VBM3D 30.38dB (c) BM4D 28.70dB (d) Ours 30.67dB

Fig. 1. Multiview denoising using VBM3D, BM4D and the proposed method for the image "Barn".



(a) Noisy $\sigma = 20$ (b) VBM3D 32.73dB (c) BM4D 31.07dB (d) Ours 32.84dB

Fig. 2. Multiview denoising using VBM3D, BM4D and the proposed method for the image "Facility Management".

Video Sequences	BM3D [7]	VBM3D [13]	BM4D [14]	Ours Cond 1
Facility Management	31.48 (0.89)	32.73 (0.91)	31.07 (0.88)	**32.84 (0.92)**
Jacob School	25.83 (0.84)	**28.12 (0.91)**	25.23 (0.81)	27.97 **(0.91)**
Market Place	31.69 (0.93)	32.61 **(0.94)**	**32.97** (**0.94**)	32.80 (0.92)
SuperLoop - Big	30.81 **(0.94)**	30.79 (0.93)	30.13 (0.92)	**31.45** (0.93)
SuperLoop - Small	29.04 (0.89)	31.48 (0.92)	**31.95 (0.93)**	31.30 (0.92)
Voigt Drive 1	29.81 (0.92)	31.32 (0.94)	**31.69 (0.95)**	31.19 (0.93)
Voigt Drive 2	29.89 (0.91)	31.80 (**0.94**)	**31.98 (0.94)**	31.48 (0.93)

Table 2. PSNR and SSIM values for video denoising.

in the neighborhood), the results indicate that a large number of dissimilar patches, despite having small weights, could severely reduce the denoising quality as a whole. Compared with BM4D (which groups temporal patches at the same locations of consecutive frames as similar patches), the results indicate that if similar patches cannot be grouped accurately across views, additional patches would deteriorate the denoising performace.

3.2. Video Denoising

Our proposed method is primarily designed for multiview denoising where displacement is large. When the displacement is small, our proposed method still works. However, the marginal gain compared to existing video denoising algorithms becomes smaller. Nevertheless, as indicated in Table 2 and Figure 2, the PSNR values of our proposed method is competitive with existing video denoising algorithms. Averaged over the 7 video sequences we tested, the PSNR improvements of the proposed method over BM3D, VBM3D and BM4D are $+1.5$dB, $+0.05$ and $+0.57$dB, respectively.

4. CONCLUSION

We present an adaptive non-local means denoising method for multiview images in which similar patches are carefully chosen according to the local statistics of the estimates. Dissimilar patches are discarded in order to achieve a trade-off between bias and variance. Experimental results show that the proposed method outperforms state-of-the-art denoising algorithms in multiview denoising settings, and performs competitively in video denoising settings.

5. REFERENCES

[1] A. Buades, B. Coll, and J. Morel, "A review of image denoising algorithms, with a new one," *SIAM Multiscale Model and Simulation*, vol. 4, no. 2, pp. 490–530, 2005.

[2] L. Zhang, S. Vaddadi, H. Jin, and S. Nayar, "Multiple view image denoising," in *Proc. IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2009, pp. 1542–1549.

[3] S. Chan, D. Vo, and T. Nguyen, "Subpixel motion estimation without interpolation," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2010, pp. 722–725.

[4] T. Brox, A. Bruhn, N. Papenberg, and J.Weickert, "High accuracy optical flow estimation based on a theory for warping," in *Proc. European Conference on Computer Vision (ECCV)*, 2004, pp. 25–36.

[5] S. Chan, R. Khoshabeh, K. Gibson, P. Gill, and T. Nguyen, "An augmented lagrangian method for total variation video restoration," *IEEE Trans. Image Process.*, vol. 20, pp. 3097–3111, Nov. 2011.

[6] C. Kervrann and J. Boulanger, "Local adaptivity to variable smoothness for exemplar-based image regularization and representation," *International Journal on Computer Vision*, vol. 79, no. 1, pp. 45–69, 2008.

[7] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3D transform-domain collaborative filtering," *IEEE Trans. Image Process.*, vol. 16, pp. 2080–2095, Aug. 2007.

[8] L. Zhang, W. Dong, D. Zhang, and G. Shi, "Two-stage image denoising by principal component analysis with local pixel grouping," *Pattern Recognition*, vol. 43, pp. 1531–1549, Apr. 2010.

[9] V. Katkovnik, A. Foi, K. Egiazarian, and J. Astola, "From local kernel to nonlocal multiple-model image denoising," *International Journal on Computer Vision*, vol. 86, pp. 1–32, 2010.

[10] Y. Lou, P. Favaro, S. Soatto, and A. Bertozzi, "Nonlocal similarity image filtering," in *Proc. International Conference on Image Analysis and Processing*, 2009, pp. 62–71.

[11] T. Thaipanich, B. Oh, P. Wu, and C. Kuo, "Adaptive nonlocal means algorithm for image denoising," in *Proc. IEEE International Conference on Consumer Electronics (ICCE)*, Jan. 2010, pp. 417–418.

[12] T. Buades, Y. Lou, J. Morel, and Z. Tang, "A note on multi-image denoising," in *Proc. IEEE International Workshop on Local and Non-Local Approximation in Image Processing*, 2009, pp. 1–15.

[13] K. Dabov, A. Foi, and K. Egiazarian, "Video denoising by sparse 3d transform-domain collaborative filtering," in *Proc. European Signal Processing Conference (EUSIPCO)*, Sep. 2007, pp. 145–149.

[14] M. Maggioni, G. Boracchi, A. Foi, and K. Egiazarian, "Video denoising using separable 4-D nonlocal spatiotemporal transforms," in *Proc. SPIE*, 2011, vol. 7870, pp. 787003–787003–11.

[15] C. Liu and W. Freeman, "A high-quality video denoising algorithm based on reliable motion estimation," in *Proc. Euporpean Conference on Computer Vision (ECCV)*, 2010, pp. 706–719.

[16] J. Boulanger, C. Kervrann, and P. Bouthemy, "Space-time adaptation for patch-based image sequence restoration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, pp. 1096–1102, Jun. 2007.

[17] R. Dugad and N. Ahuja, "Video denoising by combining Kalman and Wiener estimates," in *Proc. IEEE International Conference on Image Processing (ICIP)*, 1999, vol. 4, pp. 152–156.

[18] C. Ercole, A. Foi, V. Katkovnik, and K. Egiazarian, "Spatio-temporal pointwise adaptive denoising in video: 3D non parametric approach," in *Proc. International Workshop Video Processing and Quality Metrics for Consumer Electronics*, 2005.

[19] V. Zlokolica and W. Philips, "Motion and detail adaptive denoising in video," in *Proc. SPIE*, 2004, vol. 5298, pp. 403–412.

[20] N. Rajpoot, Z. Yao, and R. Wilson, "Adaptive wavelet restoration of noisy video sequences," in *Proc. IEEE International Conference on Image Processing (ICIP)*, Oct. 2004, pp. 957–960.

[21] D. Rusanovskyy and K. Egiazarian, "Video denoising algorithm in sliding 3D DCT domain," in *Proc. Advanced Concepts for Intelligent Vision Systems*, 2005.

[22] V. Katkovnik, "On adaptive local polynomial approximation with varying bandwidth," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 1998, vol. 4, pp. 2321–2324.

[23] C. Liu, *Beyond Pixels: Exploring New Representations and Applications for Motion Analysis*, Ph.D. thesis, Massachusetts Institute of Technology, 2009.

[24] A. Buades, B. Coll, and J.M. Morel, "Denoising image sequences does not require motion estimation," in *Proc. IEEE Conference on Advanced Video and Signal Based Surveillance*, Sep. 2005, pp. 70–74.