TARGETED VIDEO DENOISING FOR DECOMPRESSED VIDEOS

Shibin Parameswaran[†], *Enming Luo*[⋆], *Truong Q. Nguyen*[†]

[†]Department of Electrical and Computer Engineering
University of California-San Diego, La Jolla, CA 92093, USA
[⋆]Facebook, Menlo Park, CA 94025, USA

ABSTRACT

The paradigm of using clean patches from a *targeted* external database to design optimal denoising filters, called Targeted Image Denoising (TID), has been shown to outperform state-of-the-art denoising algorithms such as BM3D. In this paper, we introduce Targeted Video Denoising algorithm that extends the TID algorithm to denoise decompressed video without adding complexity. Our algorithm leverages the motion vectors generated during compression to establish temporal coherency between patches in consecutive frames. We test our algorithm on three decompressed video sequences with different foregrounds, backgrounds and movement patterns, and different noise level settings. Experimental results show that our approach is effective and performs better than original TID and the state-of-the-art video denoising algorithm.

Index Terms— decompressed video, targeted video denoising, patch-based enhancement, denoising, motion vectors

1. INTRODUCTION

Patch-based denoising algorithms such as non-local means (NLM) [1] and BM3D [2], originally introduced to denoise images, have been successfully adapted for video denoising [3, 4, 5]. These approaches assume that images (or videos) contain repeated structures which promises the existence of mutually similar patches within an image/video. For image denoising, similar patches are obtained by searching a 2D neighborhood around the pixel (or patch) being processed. However, for videos, a 3D region around a patch involving its spatial and temporal neighborhoods is searched to find similar patches [3, 4, 5, 6]. Since the similar patches used during denoising are obtained from within an image (or video), these methods fall into a category known as *internal denoising* algorithms. Although very efffective, internal denoising approaches do have some limitations when denoising rare patches [7, 8] and when operating under high noise scenarios. Rare patch effect is nominally alleviated in videos by using patches from multiple frames. However, since similar patches are selected from within the noisy video, the limitations that arise under high noise scenarios still remain a challenge.

In the case of image denoising, researchers have shown that these issues can be ameliorated by using *external* denoising algorithms where similar patches are obtained from a large external database of clean patches [9, 10, 11, 12]. Recently, Luo *et al.* [13, 14] argued for the use of a *targeted* database that incorporated prior knowledge about the scene in an image instead of using large *generic* databases. Their method, known as targeted image denoising (TID), achieved impressive gains in denoising performance by using targeted databases over state-of-the-art internal denoising algorithms like NLM and BM3D, and other external denoising algorithms that use generic databases [13, 14].

In this paper, our focus is on denoising decompressed videos corrupted during transmission. To this end, we adapt TID algorithm to perform video denoising, thus incorporating the advantages of the state-of-the-art external denoising into the video denoising domain. Temporal information contained in a video sequence is leveraged by re-purposing the motion vectors available in a compressed video to establish correspondences between patches in neighboring frames. We evaluate the extended TID algorithm on multiple decompressed video sequences and show that our adaptation makes the already powerful denoising filter more suited for video denoising with virtually *no* added overhead.

Contributions: The contributions of this work are as follows. First, we extend the state-of-the-art TID algorithm to video denoising paradigm. Second, we avoid introducing additional complexity per frame while finding the temporal neighborhood by using the sparse motion vectors extracted from the bitstream of the compressed video. Use of sparse motion vectors instead of dense optical flow or predictive block matching keeps the per-frame complexity of the video denoising algorithm the *same* as that of the underlying image denoising algorithm with only minor reduction in denoising performance.

2. RELATED WORK

2.1. Establishing temporal coherence in video denoising

A critical step in adapting an image denoising algorithm to videos is exploiting the temporal coherency among video frames while searching for similar patches. Taking advan-

978-1-5090-2175-8/17/$31.00 ©2017 IEEE

tage of this similarity between consecutive frames has been key to the success of existing patch-based video denoising algorithms [3, 4, 6, 5, 15, 16, 17]. Temporal coherence was loosely enforced in the original NLM extension [3] by simply searching for patches in a 3D spatio-temporal volume centered around the query patch. Some researchers have argued that explicit motion estimation using optical flow to incorporate temporal coherence provides better results for real-world videos [4, 17]. Video extension of BM3D [5] avoids explicit motion estimation. It uses a predictive-search block matching scheme that searches for similar patches in spatio-temporal volumes that are adapted based on the query patch (data-adaptive). All these approaches of involving the temporal domain adds more complexity to the denoising process. For compressed videos, we show that this added complexity can be avoided by re-using the already available motion vectors present in the bitstream of the compressed video.

2.2. Targeted Image Denoising

TID is an external denoising algorithm that utilizes a *targeted* database of clean patches to denoise an image. A *targeted* database, unlike a *generic* one, is built using prior knowledge of the scene contained in the image being denoised. For instance, while denoising an image of a face, the targeted database is constructed using other face images. In addition to consisting of only relevant patches, using domain knowledge allows for keeping the size of the database smaller, leading to better computational speed.

TID designs data-adaptive optimal denoising filters maximally utilizing the information contained in patches in the given targeted database. First, TID retrieves clean patches similar to the noisy patch under consideration (query patch). Then, the retrieved similar patches are used to learn an optimal denoising filter by solving a group sparsity minimization problem and using a localized Bayesian prior. Please refer to [14] for a detailed explanation and derivation of the TID filter.

The data-adaptive nature of the TID denoising filter makes selection of similar patches from the database very important. We improve the efficacy of TID on videos by leveraging temporal coherence between video frames during patch matching.

As assumed by Luo et al. [13, 14], we also assume that a targeted database is given. Choosing an appropriate database can be a challenging problem in itself and is out of the scope of this study. However, please note that in the absence of a targeted database, a very large generic database of clean patches can be used without any change to the underlying algorithm.

3. TARGETED VIDEO DENOISING

Let $v(i)$ and $u(i)$ be the observed video signal and original signal, respectively, at the spatio-temporal index $i = (x, y, t)$. Here, we consider the case where the noise signal $\eta(i)$ is i.i.d. Gaussian with zero mean and variance σ^2. That is, $v(i) = u(i) + \eta(i)$, where $\eta(i) \sim \mathcal{N}(0, \sigma^2)$. The variance σ^2 is assumed to be known *a priori*.

Algorithm 1 Targeted Video Denoising

```
INPUT: Query patch ($q$), Motion vectors ($(v_X, v_y)$), Database of clean patches ($\mathbf{D}$), Noise variance ($\sigma^2$)
OUTPUT: Denoised patch ($\hat{p}$)
1. Obtain the temporal neighbors of the query patch $q$ based on motion vectors (e.g. $q^{t\pm1}$)
2. Find $n$ reference patches $p_1, p_2, \dots p_n$ from the database using $q$ and its temporal neighbors
3. Form data matrix by concatenating patches retrieved using $q$ and using its temporal neighbors ($q^{t\pm1}$)
        $\mathbf{P} = \left[\mathbf{P^{t-1}}, \mathbf{P^{t}}, \mathbf{P^{t+1}}\right] = [p_1, p_2, \dots p_n]$
4. Form weight matrix: $\mathbf{W} = \frac{1}{\alpha}$ diag$\{w_1, w_2, \dots, w_n\}$ where $w_i = \exp\left(-\frac{\|q-p_i\|^2}{h^2}\right)$, $h$ is user-tunable bandwidth parameter and $\alpha$ is normalization parameter so that the weights add up to 1.
5. Perform eigen-decomposition: $[\mathbf{U}, \mathbf{S}] = \text{eig}\left(\mathbf{PWP}^T\right)$
6. Perform shrinkage: $\mathbf{\Lambda} = \left(\text{diag}\left(\mathbf{S} + \sigma^2\mathbf{I}\right)\right)^{-1} \text{diag}(\mathbf{S})$
7. Denoise $q$: $\hat{p} = \mathbf{U\Lambda U}^T\ q$
```

An outline of the extension of TID for videos, Targeted Video Denoising (TVD) algorithm, is shown in Algorithm 1. For the remainder of this section we provide detailed explanation of the modifications made to the original TID algorithm.

We start our adaptation of TID by establishing and identifying temporal correspondences between patches in consecutive frames (**step 1** of algorithm 1). Previously, temporal correspondence between patches of a frame and its neighboring frames were calculated using optical flow [4] or data-adaptive predictive-search block matching [5] or block matching in a non-adaptive fixed window [3]. These methods add considerable overhead since these operations are carried out for each video frame. To circumvent this problem we propose using motion vectors that are available in a compressed video stream, which are generated during video compression.

Video compression schemes such as MPEG-4 rely heavily on motion estimation to exploit the high spatial and temporal redundancy present in consecutive frames of a video. The estimated motion vectors are used to perform motion-compensated frame differencing to obtain better compression of information contained in videos. Compressed video contains frames that fall into three different picture types: P, B and I frames. The P frames use data from previous frame and have motion vectors that relate the current frame to its previous frame. Likewise, the B frames use data from both its previous and next frames and hence have motion vectors in both temporal directions. Since these motion vectors are encoded in compressed video representations, they can be accessed during decompression with no additional cost. We propose using these motion vectors to identify the temporal neighbors of patches in P and B frames. The I frames are intra-coded

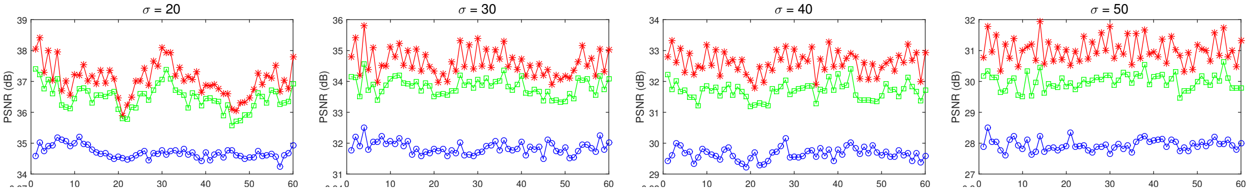

Fig. 1. **Miss America** sequence (First 60 frames): Per-frame PSNR comparison of the proposed TVD algorithm ('$*$' marker) with the VBM3D algorithm ('$\circ$' marker) for different noise levels. For reference, results obtained using the original TID applied on a per-frame basis with no temporal assistance is also shown ('□' marker).

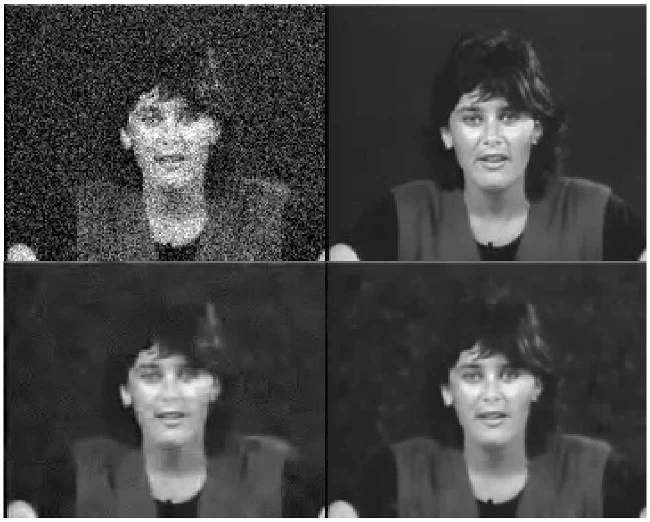

Fig. 2. Frame 5 of the **Miss America** sequence: Visual comparison of a frame corrupted by AWGN of $\sigma = 30$ (top left), the original frame (top right), the VBM3D result (bottom left) and the output of the proposed method (bottom right)

and not dependent on past or future frames. They are similar to static images, with no motion vector information. These frames can be either denoised as an image without using temporal information or subjected to optical flow calculation.

Unlike the dense optical flow, which is defined for all pixels, motion vectors used in a video compression scheme are usually sparse and are defined per-macroblock instead of per-pixel – e.g. a single motion vector for each 16x16 block. Therefore, every pixel in a macroblock has the same motion vector. Although these coarse motion fields may not capture the true motion of every single pixel of a frame, they provide a good approximation and can be easily incorporated into the denoising pipeline with no additional complexity.

After establishing a temporal correspondence between pixels of consecutive frames, each patch can be associated to its counterpart in temporal directions using the motion vectors of its central pixel. This patch correspondence is then used during patch matching as follows (**steps 2-3** of algorithm 1): let the set of patches retrieved from the targeted database using the query patch q of frame t be $\mathbf{P^t}$ and the sets retrieved using q's temporal neighbors ($q^{t\pm1}$) at time $t-1$ and $t+1$ be $\mathbf{P^{t-1}}$ and $\mathbf{P^{t+1}}$ respectively. Then, the data matrix $\mathbf{P} = \left[\mathbf{P^{t-1}}, \mathbf{P^{t}}, \mathbf{P^{t+1}}\right]$; that is, the data matrix used to design the denoising filter is formed by taking the union of the sets of similar patches retrieved using q and all of its temporal neighbors. In addition to providing a certain degree of temporal consistency, this augmentation utilizing patch correspondence also provides robustness to random noise.

The rest of the denoising algorithm (**steps 4-7** of algorithm 1) identical to the original TID algorithm [13, 14].

4. EXPERIMENTAL RESULTS

We tested our algorithm on three grayscale videos with different characteristics. For our first set of experiments we use the well-known "Miss America" and "Coastguard" benchmark sequences. Miss America is a fairly static scene featuring small movements, whereas Coastguard features object movement and tracking. The third video sequence tested is footage of a diver (video # 007) taken from the UCF sports action dataset [18, 19]. The test sequences were created by adding varying amounts of Gaussian noise to the clean video frames.

Targeted database: We assume that either three noise-free images (frames) from the long video or video of the same setting captured on a different occasion are available. For example, three images shot while a surveillance video is being recorded which are sent or saved for offline enhancement of the video in the future. With this application scenario in mind, we use the Miss America and Coastguard sequences to demonstrate the efficacy of TVD under the more *ideal* of these two scenarios. That is, for these videos, the targeted database was created using three random frames from their respective clean videos. We use the diving sequence to demonstrate a more realistic scenario where a targeted database is created using an earlier video shot in a similar setting. In particular, for the diving video (video # 007), we use a *different* video featuring a *different* diver as the targeted database (video # 005 from UCF sports). This is a more realistic setting where we have access to clean videos that are *similar* but *different* from the noisy video. To minimize the database size for the diving sequence, we pick a moving window of three frames from video # 005 to be the targeted database.

Parameter settings: We use the same parameter settings for all the experiments. The targeted video denoising algorithm is run twice where the result from the first iteration is used for finding similar patches in the second iteration. The patch size is set to 8x8 pixels. We retrieve a total of 120 sim-

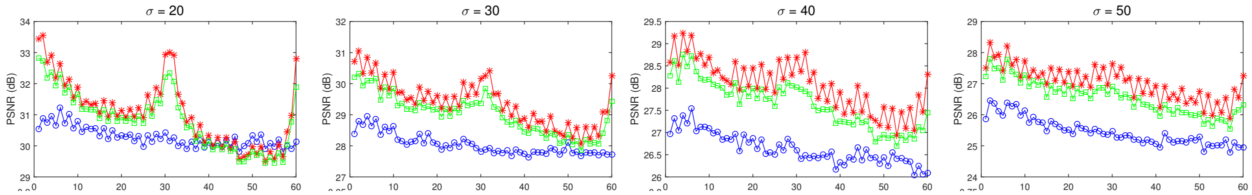

Fig. 3. **Coastguard** sequence (First 60 frames): Per-frame PSNR comparison of the proposed TVD algorithm ('$*$' marker) with the VBM3D algorithm ('$\circ$' marker) for different noise levels. For reference, results obtained using the original TID applied on a per-frame basis with no temporal assistance is also shown ('$\square$' marker).

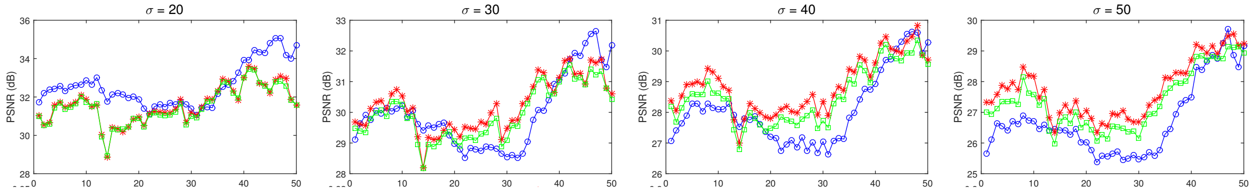

Fig. 4. **Diving** sequence (55 frames): Per-frame PSNR comparison of the proposed TVD algorithm ('$*$' marker) with the VBM3D algorithm ('$\circ$' marker) for different noise levels. For reference, results obtained using the original TID applied on a per-frame basis with no temporal assistance is also shown ('$\square$' marker).

ilar patches from the database for every noisy patch ($n = 120$). Therefore, if a query patch has only one temporal neighbor then both of them will be used to retrieve 60 reference patches each. We compare our results with the state-of-the-art VBM3D algorithm [5]. We use the VBM3D implementation from the BM3D website (http://www.cs.tut.fi/ foi/GCF-BM3D/). For all the experiments, we used the default parameters except for the temporal window size which is set to 3 (by default this is set to 9 in the VBM3D package). To demonstrate the improvement obtained using temporal correspondences, we also compare our results obtained from TID algorithm, with $n = 120$, applied on a per-frame basis without using any temporal knowledge.

The **Miss America** sequence is of QCIF resolution (144x180). The per-frame denoising results obtained on this video sequence are shown in figure 1. Qualitatively and quantitatively, the proposed algorithm clearly outperforms VBM3D and TID on enhancing this video in all the noise settings we tested. TVD achieves a global PSNR (calculated on the whole sequence) improvement of around 2.4-3.1 dB and global SSIM improvement of 0.04-0.1 over VBM3D. For visual comparison, one of the frames from the $\sigma = 30$ noise setting is shown in figure 2.

The **Coastguard** video contains moving objects and also features egomotion due to the camera tracking a moving boat. The resolution of this video is 144x176 pixels. The per-frame comparison of the PSNR values obtained after denoising this sequence with TVD, TID and VBM3D are presented in figure 3. The spikes in the performance curves of the proposed algorithm and TID are due to the location of the frames used as the targeted database. However, this advantage is canceled out in the high noise scenarios (e.g. $\sigma = 50$) because of the difficulty in finding true matches using severely corrupted patches. The global PSNR improvements shown by TVD over VBM3D range from 0.8-1.6 dB (Δglobal-SSIM = 0.04-0.09) as the noise is increased from $\sigma = 20$ to 50.

We downsampled the **diving** sequence from the UCF sports action dataset [18, 19] from a spatial resolution of 404x720 to 101x180. Both the testing sequence and the database sequence have 55 frames each. The per-frame denoising results obtained on this video sequence are presented in figure 4. Under the low noise setting ($\sigma = 20$), although TVD has a slightly higher global SSIM than VBM3D (0.003), global PSNR of VBM3D is better than TVD by 1.1 dB. More importantly, the proposed approach outperforms VBM3D for higher noise settings with $\sigma \geq 30$ yielding global PSNR improvements of 0.4-1.2 dB (Δglobal-SSIM = 0.02-0.07).

Runtime: The proposed TVD algorithm has the same runtime complexity as TID since the extraction of motion vectors does not add any noticeable overhead. Our current implementation of TID denoising algorithm takes approximately 60s per frame in the diving sequence (101x180). The runtimes of TID and TVD can be optimized by using large-scale approximate nearest neighbor algorithms for patch matching such as the RIANN [20] or PatchTable [21] algorithms.

5. CONCLUSION

We have introduced a patch-based video denoising algorithm by extending the TID algorithm [14, 13]. In order to take advantage of the similarity among video frames in a temporal neighborhood, we augmented the localized Bayesian prior of TID with temporal coherency prior. Using the motion vectors present in the compressed bitstream, we denoise decompressed videos with no added complexity. Our results demonstrate that targeted video denoising consistently outperforms its image processing counterpart, TID, and the state-of-the-art internal video denoising algorithm VBM3D in mid- and high-noise conditions.

6. REFERENCES

[1] Antoni Buades, Bartomeu Coll, and Jean-Michel Morel, "A review of image denoising algorithms, with a new one," vol. 4, no. 2, pp. 490–530, 2005.

[2] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2095, August 2007.

[3] Antoni Buades, Bartomeu Coll, and Jean-Michel Morel, "Denoising image sequences does not require motion estimation," in *Advanced Video and Signal Based Surveillance, 2005. AVSS 2005. IEEE Conference on*, Sept 2005, pp. 70–74.

[4] Ce Liu and William T. Freeman, "A high-quality video denoising algorithm based on reliable motion estimation," in *Proceedings of the 11th European Conference on Computer Vision Conference on Computer Vision: Part III*, Berlin, Heidelberg, 2010, ECCV'10, pp. 706–719, Springer-Verlag.

[5] Kostadin Dabov, Alessandro Foi, and Karen Egiazarian, "Video denoising by sparse 3d transform-domain collaborative filtering," in *Signal Processing Conference, 2007 15th European*, Sept 2007, pp. 145–149.

[6] Mona Mahmoudi and Guillermo Sapiro, "Fast image and video denoising via nonlocal means of similar neighborhoods," *IEEE Signal Processing Letters*, vol. 12, pp. 839–842, 2005.

[7] Priyam Chatterjee and Peyman Milanfar, "Is Denoising Dead?," *IEEE Transactions on Image Processing*, vol. 19, no. 4, pp. 895–911, April 2010.

[8] Anat Levin and Boaz Nadler, "Natural image denoising: Optimality and inherent bounds," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'11)*. June 2011, pp. 2833–2840, IEEE.

[9] Harold Christopher Burger, Christian Schuler, and Stefan Harmeling, "Learning how to combine internal and external denoising methods," in *Pattern Recognition*, Joachim Weickert, Matthias Hein, and Bernt Schiele, Eds., vol. 8142 of *Lecture Notes in Computer Science*, pp. 121–130. Springer Berlin Heidelberg, 2013.

[10] Stanley H Chan, Todd Zickler, and Yue M Lu, "Monte Carlo non-local means: random sampling for large-scale image filtering.," *IEEE transactions on image processing*, vol. 23, no. 8, pp. 3711–25, August 2014.

[11] Inbar Mosseri, Maria Zontak, and Michal Irani, "Combining the Power of Internal and External Denoising," in *IEEE International Conference on Computational Photography (ICCP)*, 2013, pp. 1–9.

[12] Daniel Zoran and Yair Weiss, "From learning models of natural image patches to whole image restoration," in *International Conference on Computer Vision*. Nov. 2011, pp. 479–486, IEEE.

[13] Enming Luo, Stanley H. Chan, and Truong Q. Nguyen, "Image denoising by targeted external databases," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. May 2014, pp. 2450–2454, IEEE.

[14] Enming Luo, Stanley H Chan, and Truong Q Nguyen, "Adaptive image denoising by targeted databases.," *IEEE transactions on image processing*, vol. 24, no. 7, pp. 2167–81, July 2015.

[15] Matteo Maggioni, Vladimir Katkovnik, Karen Egiazarian, and Alessandro Foi, "Nonlocal transform-domain filter for volumetric data denoising and reconstruction," *IEEE Transactions on Image Processing*, vol. 22, no. 1, pp. 119–133, Jan 2013.

[16] Matteo Maggioni, Giacomo Boracchi, Alessandro Foi, and Karen Egiazarian, "Video denoising, deblocking, and enhancement through separable 4-d nonlocal spatiotemporal transforms," *IEEE Transactions on Image Processing*, vol. 21, no. 9, pp. 3952–3966, Sept 2012.

[17] Antoni Buades, Jose-Luis Lisani, and Marko Miladinovc, "Patch-based video denoising with optical flow estimation," *Transactions on Image Processing*, vol. 25, no. 6, pp. 2573–2586, June 2016.

[18] Mikel D. Rodriguez, Javed Ahmed, and Mubarak Shah, "Action mach: a spatio-temporal maximum average correlation height filter for action recognition," in *In Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, 2008.

[19] Khurram Soomro and Amir R. Zamir, *Computer Vision in Sports*, chapter Action Recognition in Realistic Sports Videos, pp. 181–208, Springer International Publishing, Cham, 2014.

[20] Nir Ben-Zrihem and Lihi Zelnik-Manor, "Riann: Approximate nearest neighbor fields in video," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'15)*, 2015.

[21] Connelly Barnes, Fang-Lue Zhang, Liming Lou, Xian Wu, and Shi-Min Hu, "Patchtable: Efficient patch queries for large datasets and applications," in *ACM Transactions on Graphics (Proc. SIGGRAPH)*, Aug. 2015.