

High-quality Video Denoising for Motion-based Exposure Control

Travis Portz Li Zhang Hongrui Jiang

University of Wisconsin–Madison

<http://pages.cs.wisc.edu/~lizhang/projects/autoexpo/>

Abstract

New digital cameras, such as Canon SD1100 and Nikon COOLPIX S8100, have an Auto Exposure (AE) function that is based on motion estimation. The motion estimation helps to set short exposure and high ISO for frames with fast motion, thereby minimizing most motion blur in recorded videos. This AE function largely turns video enhancement into a denoising problem. This paper studies the problem of how to achieve high-quality video denoising in the context of motion-based exposure control. Unlike previous denoising works which either avoid using motion estimation, such as BM3D [7], or assume reliable motion estimation as input, such as [13], our method evaluates the reliability of flow at each pixel and uses that reliability as a weight to integrate spatial denoising and temporal denoising. This weighted combination scheme makes our method robust to optical flow failure over regions with repetitive texture or uniform color and combines the advantages of both spatial and temporal denoising. Our method also exploits high quality frames in a sequence to effectively enhance noisier frames. In experiments using both synthetic and real videos, our method outperforms the state of the art [7, 13].

1. Introduction

In most automated vision systems and consumer cameras, it is desirable to automatically determine an appropriate exposure time based on the scene; this function is known as Auto Exposure (AE). Traditionally, Auto Exposure is mainly determined by environment *brightness*: bright scenes lead to a short exposure time and/or a large aperture. This control scheme is simple to implement and has been widely adopted. However, when the brightness level of a scene remains constant, this scheme does not consider camera motion or subject motion and therefore often leads to motion blur. Small consumer cameras and mobile vision systems can benefit from a better control scheme for a variety of reasons.

- These cameras often have small, fixed aperture sizes. In low-light conditions, the AE must resort to a long exposure time or a high ISO setting. A long exposure time will result in significant motion blur when motion is present. On the other hand, a high ISO setting will provide lower quality images than possible if the motion is small.

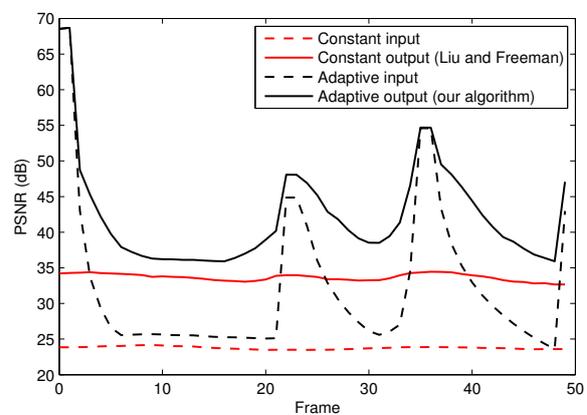


Figure 1. The benefit of denoising videos captured with motion-based exposure control. Top: A panoramic image from which we generate a synthetic video whose viewport (red box) moves along the red zigzag curve with varying speed. Bottom: If a constant short exposure is applied to each frame to minimize blur, the captured video has constant low PSNR (dashed red curve), and a state-of-the-art video denoising [13] improves its PSNR to about 34dB (solid red curve). If exposure time is set *adaptively* based on motion estimation, the input video has higher PSNR (dashed black curve), and our denoising algorithm produces a much higher quality video with a total PSNR of 39 dB (solid black curve). **Best viewed electronically in color.**

- The handheld and/or mobile nature of these camera systems make them susceptible to camera shake and widely variable movement patterns. A fixed exposure time may be excessively short for some frames and result in motion blur for other frames.

As more computing power is put in digital cameras, new cameras, such as Canon SD1100 and Nikon COOLPIX S8100, have an Auto Exposure function that is based on motion estimation. These cameras perform motion estimation during setup time (when the shutter button is half-pressed) and adjust the shutter and ISO setting so that blur is minimized in the captured photograph. This functionality is currently only available in single-shot mode on commer-

cial cameras; however, the same concept can be applied to video capture by using the apparent motion estimated from the previous two frames to select the exposure time and ISO for the next frame. In the captured video, most frames do not have blur, but those with short exposure time will be noisy due to a high ISO setting. This AE function largely turns video enhancement into a denoising problem.

This paper studies the problem of how to achieve high-quality video denoising in the context of motion-based exposure control. This problem is pertinent as motion deblurring in general is a challenging problem; achieving high-quality denoising in this context may greatly *reduce*, although not eliminate, the need of motion deblurring for video enhancement. This problem is promising as Figure 1 shows; it is also difficult in its own ways.

- Within a sequence captured using motion-based Auto Exposure, there are often high quality frames, which correspond to the frames with little apparent motion and captured with relatively long exposure and low ISO.¹ Ideally, we would want to use the high quality frames to better enhance the noisier frames; at the same time, we would not want the noisy frames to compromise the high quality frames during the denoising process.
- Noisy frames are captured with high ISO and short exposure because of fast motion. To exploit high quality frames to enhance noisy frames, we would need robust motion estimation that can handle large displacement. In our experiments, we commonly found displacement of 70 or more, which confound even top-performing optical flow methods that have been adopted in state-of-the-art video denoising.

In this paper, we present a high quality video denoising method in the context of motion-based exposure control, by combining spatial denoising and temporal denoising in a novel way. Our combination is based on an intuitive observation. Specifically, spatial methods like BM3D [7] perform well if the image has abundant locally similar structure. Its performance starts to degrade when the local structure is unique. Motion-compensated filtering on the other hand works best when local patches are unique, because the optical flow can be reliably estimated. Therefore, our idea is to detect the reliability of the flow for each pixel, and use the reliability as a weight to combine the results of BM3D and motion-compensated filtering.

Unlike previous denoising works which either avoid using motion estimation, such as BM3D [7], or assume reliable motion estimation as input, such as [13], our method selectively operates in whichever regime works best. As a

¹For example, although it is hard to hold a camera perfectly still for a long period, it is also rare that our hands would continuously shake a camera; shaky intervals are always intermingled with steady moments.

result, our algorithm performs better than both VBM3D [6] and the latest video denoising algorithm [13].

Our flow reliability evaluation is based on a forward-backward consistency check, which is a widely used technique in stereo and motion estimation. However, this reliability measure of motion estimation has not been exploited for improving video denoising performance in the literature, to the best of our knowledge.

2. Related Work

Our work is most related to image and video denoising and enhancement.

Denoising Image denoising has been studied for several decades. A complete review is beyond the scope of this paper. We refer the readers to the previous work sections in [4, 7] for excellent reviews of the literature. An incomplete list of recent works include [16, 4, 8, 14, 7, 18, 9]. In particular, the methods that are based on local self-similarity, such as non-local means [4] and BM3D [7], are particularly notable because of their simple ideas and impressive results. The non-local means and BM3D methods do not perform well when local image patterns are unique.

Video denoising [2, 6, 5, 13] can address this limitation as the temporal dimension provides additional redundant data. Liu and Freeman [13] showed that the spatial regularization in the optical flow can be used to ensure temporal coherence in removing structured noise. Multi-view denoising [19, 11, 21] is another way of addressing this limitation, which exploits noisy measurements from multiple viewpoints to reconstruct a clean image. Zhang *et al.* [21] observed that 3D depth can be used as a constraint to find more reliable matches to further improve the performance of multi-view image denoising.

Our work is most related to [13], in which the authors integrate robust optical flow into a non-local means framework; their work assumes reliable flow as input. Our work does not assume the flow is reliable. Rather, we evaluate the flow trajectory reliability for each pixel and use the reliability measure as a weight to combine spatial denoising and temporal denoising results.

Video Enhancement using Stills Our work is also related to works that use high quality digital photos to enhance low resolution videos. For example, Bhat *et al.* [3] and Schubert *et al.* [17] proposed an approach to enhance low-resolution videos of a static scene by using multi-view stereo to compute correspondences between low-resolution video and high-resolution images; Gupta *et al.* [10] use optical flow to compute correspondences and can therefore handle dynamic scenes as well. Watanabe *et al.* [20] propagate high frequency information in high-resolution frames to low-resolution frames using motion compensation. Nagahara *et al.* [15] take a similar approach but use morphing

based on feature matching instead of motion compensation. In our work, the frame resolution is the same; what differs is the noise level. We do not assume reliable flow as input; instead, we use the reliability of the flow to combine spatial denoising and temporal denoising.

3. Denoising Algorithm

Our denoising algorithm is based on the following intuition. If an image region has *unique* texture patterns, we would prefer to use temporal denoising, because optical flow can be estimated reliably and spatial denoising usually does not work well. On the other hand, if an image region has repetitive texture or uniform color, we would prefer to use spatial denoising because optical flow is unreliable and self similarity makes spatial denoising work effectively. We do not judge the flow reliability using a binary decision. Instead, we softly combine the spatial and temporal denoising result using our reliability measure as weight. Next we explain our algorithm in detail.

Spatial Denoising We use the single-image denoising method CBM3D [7] to perform our spatial denoising:

$$\hat{I}_S(\mathbf{z}_t) = \text{CBM3D}(I_t, \mathbf{z}_t), \quad (1)$$

where I_t is the current input frame and \mathbf{z}_t is pixel location. We apply this single denoising method to each frame using the corresponding frame noise variance as parameter. We do not use CVBM3D, the video version of CBM3D, because CVBM3D only handles constant noise variance across the whole video volume, which would compromise the high quality frames in the captured video. We choose CBM3D due to its performance, efficiency, and public availability; other spatial denoising methods, such as non-local means [4], can also be used instead.

Temporal Denoising along Reliable Flow We compute the optical flow over a temporal window of $\pm H$ frames, where we use $H = 5$ as in [13]. The flow may not be reliable for every pixel and every frame in the temporal window. We use the forward-backward consistency as a measure of flow reliability. If the flow vector from a pixel in frame i to a pixel in frame j is denoted \mathbf{v}_{ij} , then the flow consistency error is $\|\mathbf{v}_{ij} + \mathbf{v}_{ji}\|^2$. We consider the flow to be consistent if the error is below some threshold (3 is used in both our synthetic and real experiments).

For each pixel in frame I_t , we determine the frames with consistent forward flow up to at most frame $t + H$, and backward flow down to at most $t - H$. The set of frames with consistent flow is denoted H_c . H_c is a function of the pixel under consideration; however, we omit the function notation for simplicity.

Once we have determined the frames with reliable flow, the temporal pixel estimate is computed by filtering along the optical flow:

$$\hat{I}_T(\mathbf{z}_t) = \frac{1}{Z} \sum_{i \in H_c} W(\mathbf{z}_i) \cdot I_i(\mathbf{z}_i), \quad (2)$$

where Z is a normalization factor and $W(\mathbf{z}_i)$ is given by:

$$W(\mathbf{z}_i) = (\beta_i^2 + \beta_t^2)^{-\frac{3}{2}} \exp \left\{ -\frac{\|P(\mathbf{z}_t) - P(\mathbf{z}_i)\|^2}{\beta_i^2 + \beta_t^2} \right\} \quad (3)$$

where $\beta_i = g_i \cdot \beta_0$ with g_i being the gain used to capture frame i and β_0 being proportional to the base noise level of the camera. In Eq. (3), we note

- The first term assigns larger weight to pixels from cleaner frames. This weighting scheme facilitates using the high quality frames to better enhance the noisier frames; at the same time, it discourages using the noisy frames to compromise the high quality frames during the denoising process.
- The exponential term assigns smaller weight to pixels that came from optical flows with poorer block matches. The distance between two patches (where a patch is denoted by $P(\cdot)$) is computed using a weighted SSD as in [13].

In addition to having the exponential term based on the patch distance, we use a threshold,

$$\tau_t = m \cdot \beta_t + \tau_0, \quad (4)$$

to reject pixels with large patch distances. The linear form and parameters for τ_t were determined empirically by maximizing the PSNR of a simulated video sequence. With pixel intensities in the range $[0, 1]$, we used $m = 0.051$ and $\tau_0 = -1.9 \cdot 10^{-3}$. The negative value for τ_0 yields zero or negative patch distance threshold τ_t for clean frames (which have small β_t) and, therefore, prevents the clean frames from being degraded by lower quality neighboring frames and/or inaccurate flows.

Combining Spatial and Temporal Denoising To combine the spatial and temporal denoising results, we linearly interpolate using the number of frames with consistent flow $|H_c|$ as the weight:

$$\hat{I}(\mathbf{z}_t) = \frac{|H_c|}{2H} \hat{I}_T(\mathbf{z}_t) + \left(1 - \frac{|H_c|}{2H}\right) \hat{I}_S(\mathbf{z}_t). \quad (5)$$

When a pixel does not have any consistent flows, we rely purely on the CBM3D estimate. When a pixel has perfectly consistent flows (within the temporal window), we rely purely on the temporal estimate.

3.1. Efficient Flow for Large Motion

Now we describe how we compute optical flow for denoising in our experiments. Optical flow is not our technical contribution; we describe it so that our paper is reproducible.

In real videos, we found that flow vectors can easily be 70 pixels or more. This large motion easily confounds many top performing flow algorithms evaluated in [1], as the benchmark data sets typically only have flow magnitude of 20 pixels or less. For example, we tried the flow algorithm [12] used in [13] as input for video denoising. The algorithm does not produce correct flow for a typical pair of frames with large motion as shown in Figure 2. We believe this is because most flow algorithms use derivative-based continuous optimization which is easily trapped in local minima, even if an image pyramid is used. To handle large motions in our video, we use a traditional hierarchical block matching technique to compute our flow.

We start by constructing image pyramids of the two frames under consideration (with a factor of two between each level). At the coarsest level, we perform block matching with a search window of size $M \times M$. Next, we upsample the flow field and refine it by searching within a smaller window at the next coarsest level of the pyramid. We use a three level image pyramid with $M=61$ for the coarse block matching and a 7×7 search window for refinement. These parameters allow us to handle displacements of up to 120 pixels between consecutive frames.

We concatenate flows between neighboring frames to initialize motion estimation between arbitrary frames, then refine by block matching at the finest resolution only. We found this simple method works well for handling large motion; an example of the flow result is shown in Figure 2.

4. Experimental Results

Our results are best viewed electronically in color. More results, including videos, are available at <http://pages.cs.wisc.edu/~lizhang/projects/autoexpo/>.

4.1. Synthetic Video

We first test our system on three different synthetic video sequences. Each sequence is generated by moving a 512×512 window around a large panoramic image as shown in Figure 3. The motion of the windows have speeds ranging from 0 to 750 pixels per second and undergo two changes of direction. Motion-based exposure control is simulated on the sequences to determine the optimal exposure time T for each frame. If d is the displacement between the previous two frames and f is the frame rate, then $T = 1/(d \cdot f)$. This results in one pixel of motion during the camera’s exposure time. The actual exposure time is clamped between 1 ms and $1/f$, where we use $f = 7.5$ frames per second. Once the exposure time has been set, we adjust the gain to keep a constant brightness level. We then add white Gaussian noise to the current frame with $\sigma = g \cdot \sigma_0$ where σ_0 is chosen such that $\sigma = 25$ (out of 255) for the shortest exposure time. We also generate videos with constant short and long exposure times for comparison.

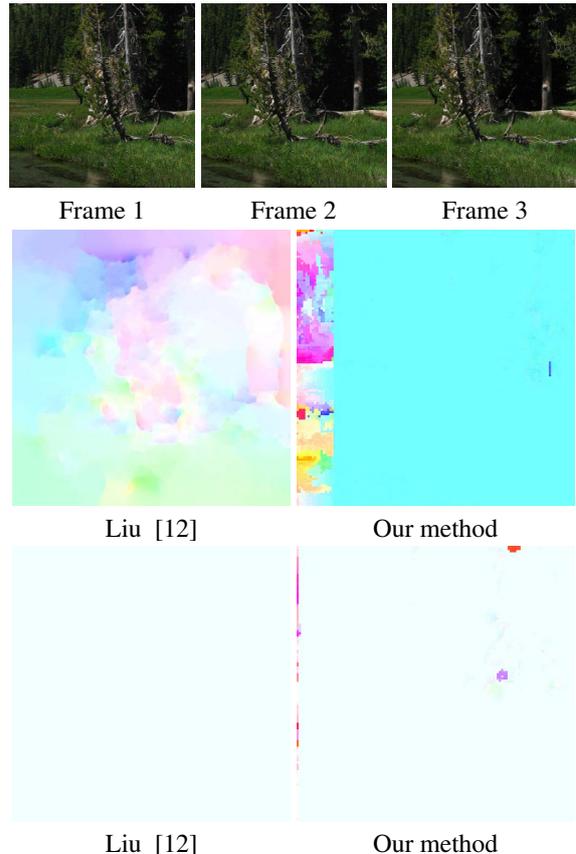


Figure 2. Optical flow results for three consecutive frames in the *mountain* scene. Top: The displacement between frames 1 and 2 is large, whereas the displacement between frames 2 and 3 is small. Middle: Our optical flow outperforms the optical flow in [12] for large displacements. The left to right motion causes the pixels on the left edge of frame 1 to be invisible in frame 2, which is why our flow is inaccurate on that edge. Bottom: The optical flow in [12] outperforms our method for small displacements by producing a smoother flow. **Best viewed electronically in color.**

We run the input sequences through CBM3D and [13] using the known σ values for each frame. Since the flow method used by [13] does not perform well on the large motion in our sequences, we use our flow as input to their denoising algorithm for a fair comparison. For our algorithm, we use $\beta_0 = 0.01$ in Eq. (3) (with pixel intensities in the range $[0, 1]$). The value for β_0 was found empirically to provide full denoising power without sacrificing texture preservation.

The per frame PSNRs can be seen in Figure 4 for the *city* and *mountain* sequences and in Figure 1 for the *station* sequence. Our algorithm provides higher PSNR than the state-of-the-art algorithms for all of the frames containing significant noise levels.

The improvements in our results over CBM3D are primarily made in the regions with unique texture and structure, as can be seen in Figure 5 and Figure 6. In these re-



City scene



Mountain scene

Figure 3. Our synthetic video sequences are generated from panoramic images. A 512x512 pixel window follows the trajectory shown in red. The motion in each sequence has variable speed and undergoes multiple direction changes. **Best viewed electronically in color.**

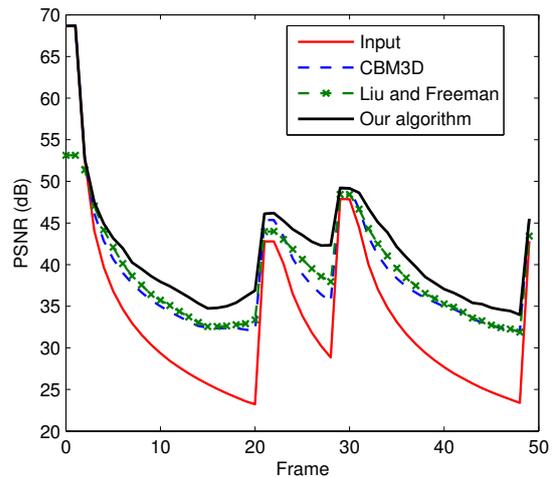
gions the optical flow is reliable, thus temporal denoising is effective. The weights between the temporal and spatial estimates are shown in Figure 7. In smooth regions where our optical flow is unreliable, our denoising algorithm falls back on CBM3D which performs well on smooth regions.

For completeness, we also run our denoising algorithm and [13] on the *city* sequence using both their flow method and the ground truth flow. In all cases, our denoising method outperforms the denoising method in [13]. When using their flow method, the frames with large motion have inaccurate flow. Our method is more robust to the flow inaccuracy (PSNR = 36.70 dB for our method, 33.25 dB for their method). When using the ground truth flow, our method better preserves detail by making use of the good flows and not relying on spatial denoising (PSNR = 39.23 dB for our method, 36.59 dB for their method). Visual comparisons can be seen in the supplementary material.

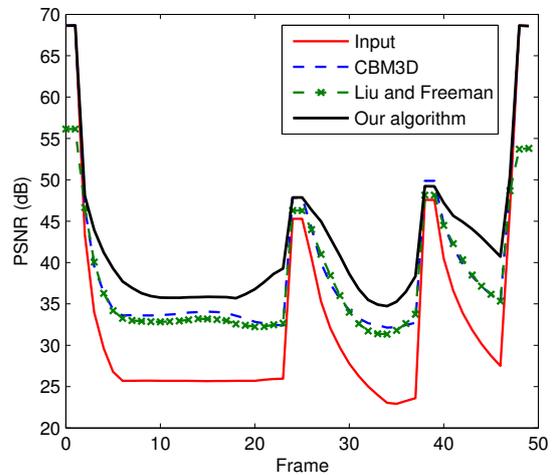
4.2. Real Video

To test our system on a real video sequence, we first needed motion-based AE. We implemented the motion estimation portion of the exposure control algorithm using a standard hierarchical image registration technique. The remainder of the AE algorithm works just as described for the synthetic video. Since the image registration only tracks global translational motion, we designed our real experiment to have primarily translational motion. We set up two cameras facing out the side window of an automobile. One camera, a Canon EOS 7D, used a constant exposure time of 1/30 seconds, and the other camera, a Point Grey Grasshopper, used motion-based AE. As shown in Figure 8, our algorithm preserves detail better than [13], because optical flow is hard to estimate reliably in the presence of large motion, multiple depth layers, and thin structure. Our method measures flow reliability and is robust to inaccurate flow input.

We also ran our denoising algorithm on the videos from [13] using both our flow method and their flow method. These videos were not captured using motion-based AE and contain relatively small motion. As a result, the smoother flow of [13] enables our denoising method to achieve better temporal consistency in smooth regions. See our supplementary material for video comparisons.



(a) City scene



(b) Mountain scene

Figure 4. PSNR results for the synthetic video sequences. In frames with significant noise levels, our algorithm outperforms other state-of-the-art denoising algorithms. **Best viewed electronically in color.**

5. Conclusion

In this paper, we have proposed a high quality video denoising algorithm in the context of motion-based exposure control. Unlike previous denoising works which either

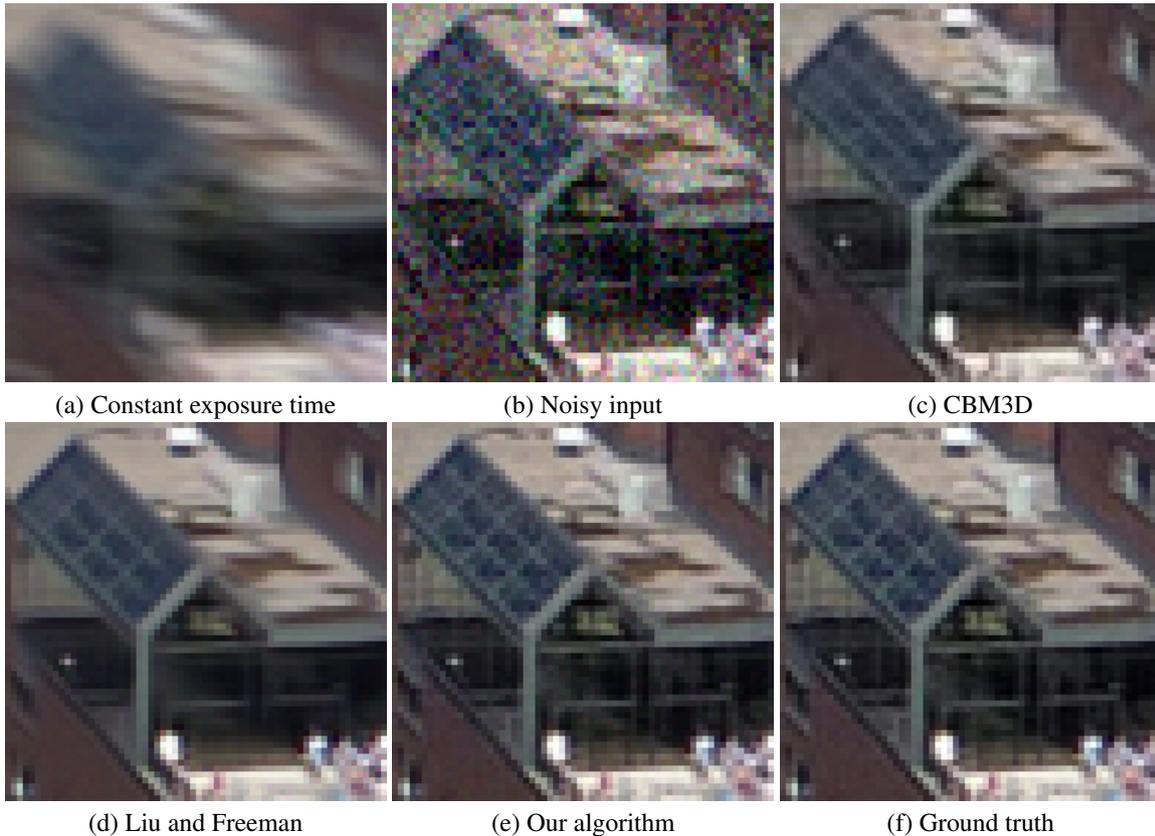


Figure 5. A close-up of results from the *city* sequence. The motion-based AE provides a sharp but noisy image, shown in (b), as opposed to the blurry image captured with a constant exposure time, shown in (a). Our denoising algorithm outperforms CBM3D [7] (applied to each individual frame using corresponding frame noise variance) and Liu and Freeman [13] (using the known noise variance for each individual frame). More detail is preserved in the roof and windows. **Best viewed electronically in color.**

avoid using motion estimation, such as BM3D [7], or assume reliable motion estimation as input, such as [13], our method uses the reliability of the optical flow as a weight to integrate spatial denoising and temporal denoising. This weighted combination scheme (1) makes our method robust to optical flow failures over regions with repetitive texture or uniform color, (2) combines the advantages of both spatial and temporal denoising, and (3) outperform the state of the art. There are several avenues for future research.

First, we would like to investigate better weighting schemes. In the current formulation, when there are no frames with reliable flow, the algorithm resorts to CBM3D; in this case, temporal coherence is not enforced. This differs from [13], which uses smooth optical flow to obtain temporal consistency in the presence of structural noise. However, as Figures 5, 6, and 8 show, this temporal consistency is obtained at the expense of sacrificing texture details. Furthermore, the lack of temporal consistency in our results is not as noticeable since the motion-based exposure control only produces noisy frames when there is large motion. Nevertheless, if video stabilization is applied to the captured video, the temporal inconsistency is still notice-

able. Therefore, more research is needed to obtain temporal consistency while still preserving spatial detail.

Second, although motion-based AE reduces motion blur significantly, it does not completely eliminate motion blur because exposure is set based on the motion of previous frames; there is always a delay. It is desirable to use the noisy frames and/or high quality frames to enhance motion blur in a video captured with motion-based AE.

Third, it will be useful to investigate a real-time implementation of this approach so that denoising can be executed before compression. Our approach has the potential to be implemented in real time as all components are block-based; no complex optimization, such as conjugate gradient, is involved in the optical flow estimation.

6. Acknowledgments

This work is supported in part by NSF EFRI-0937847, NSF IIS-0845916, NSF IIS-0916441, a Sloan Research Fellowship, and a Packard Fellowship for Science and Engineering. Travis Portz is also supported by a University of Wisconsin-Madison, Department of Electrical and Computer Engineering graduate fellowship.

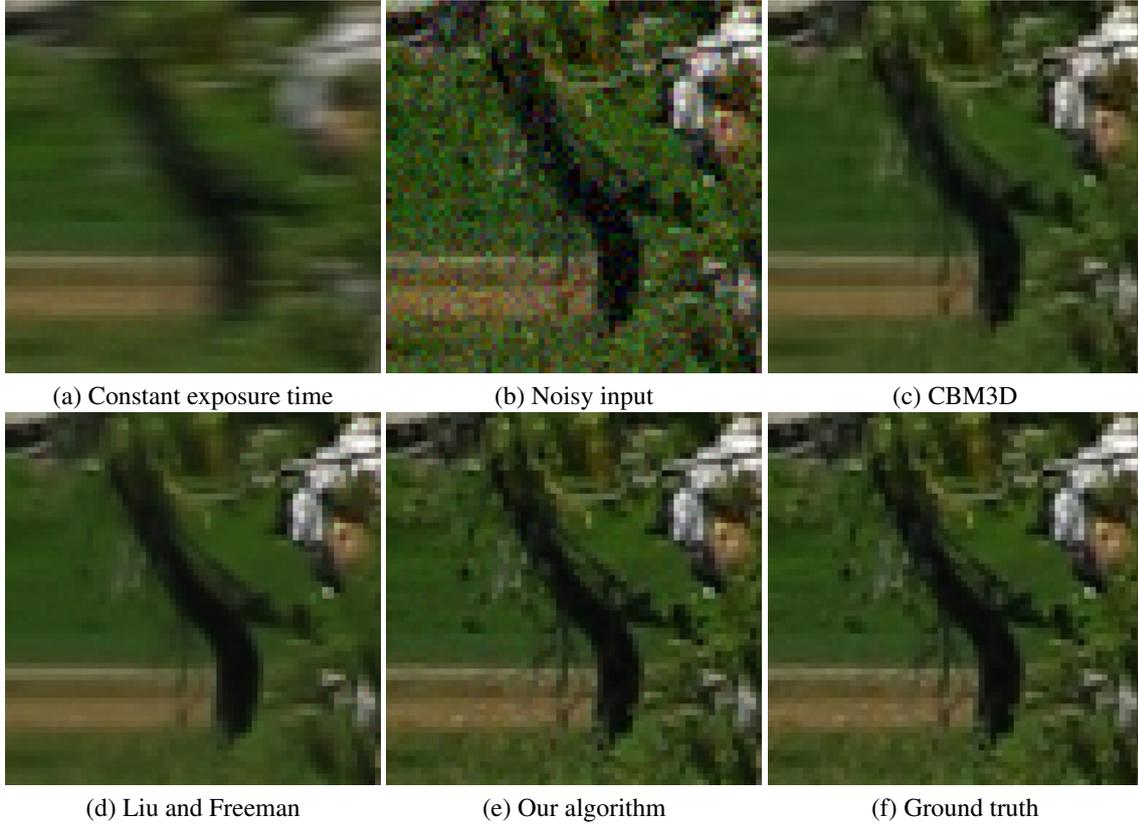


Figure 6. A close-up of results from the *mountain* sequence. Both CBM3D [7] and [13] over-smooth the tree branches and grass. Our algorithm preserves the fine structures and texture. **Best viewed electronically in color.**

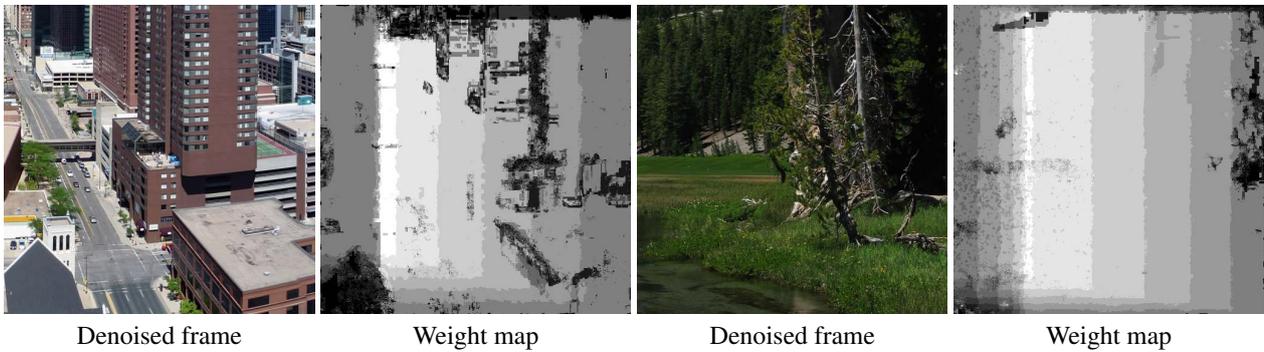


Figure 7. Two weight maps from the synthetic sequences. Lighter colors denote pixels that rely more on temporal denoising than spatial denoising. The darker regions in the weight maps correspond to smooth regions of the image where optical flow trajectory is less reliable. The horizontal motion in the video sequences causes the sides of the image to be invisible in neighboring frames, which is why we see the vertical bands of constant weight. **Best viewed electronically in color.**

References

- [1] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. J. Black, and R. Szeliski. A database and evaluation methodology for optical flow. In *ICCV*, 2007.
- [2] E. P. Bennett and L. McMillan. Video enhancement using per-pixel virtual exposures. In *SIGGRAPH*, 2005.
- [3] P. Bhat, C. L. Zitnick, N. Snavely, A. Agarwala, M. Agrawala, B. Curless, M. Cohen, and S. B. Kang. Using photographs to enhance videos of a static scene. In *Proceedings Eurographics Symposium on Rendering*, 2007.
- [4] A. Buades, B. Coll, and J. M. Morel. A review of image denoising algorithms, with a new one. *Simulation*, 2005.
- [5] J. Chen and C.-K. Tang. Spatio-temporal markov random field for video denoising. In *CVPR*, 2007.
- [6] K. Dabov, A. Foi, and K. Egiazarian. Video denoising by sparse 3d transform-domain collaborative filtering. In *Proc. 15th European Signal Processing Conference*, 2007.

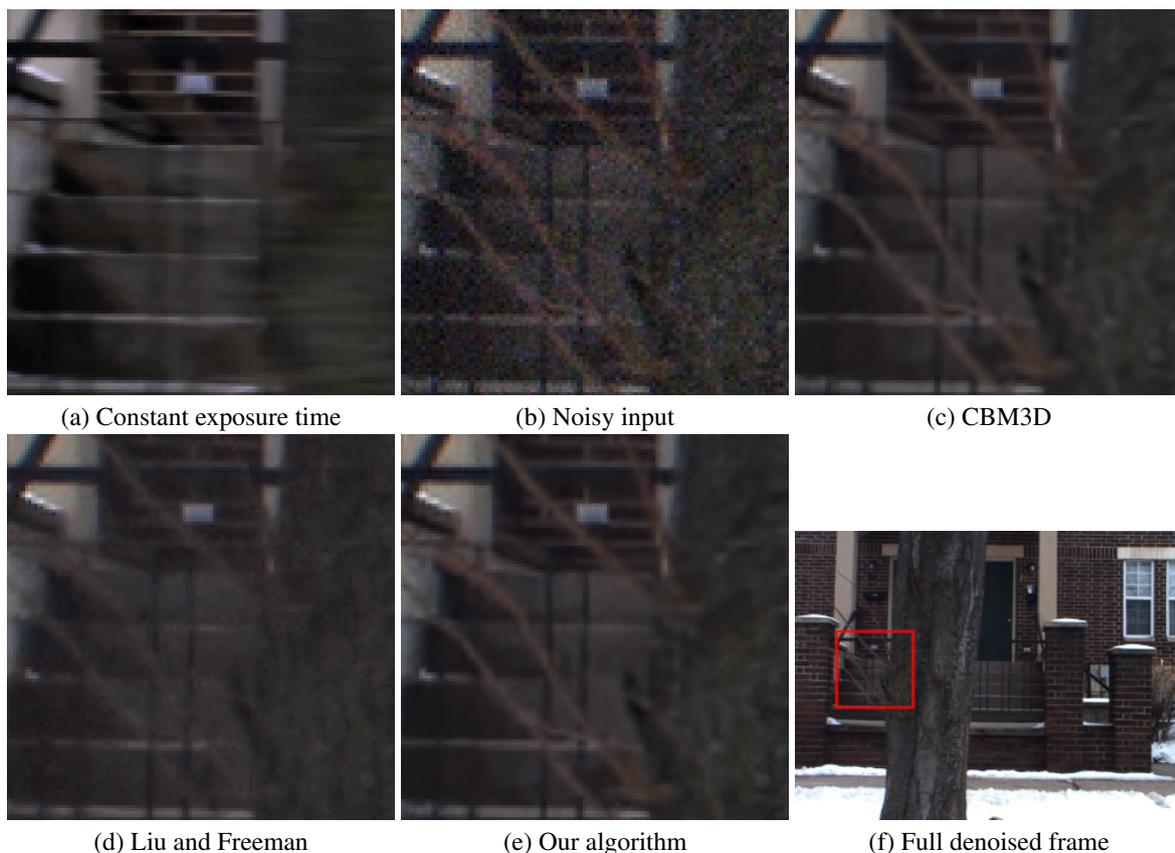


Figure 8. Results from the driving sequence. Our results are comparable to CBM3D [7], which preserves the detail of the tree reasonably well. The tree branches and some of the other fine details were over-smoothed by Liu and Freeman [13] due to inaccurate flow in the presence of large motion, multiple depth layers, and thin structure. **Best viewed electronically in color.**

- [7] K. Dabov, R. Foi, V. Katkovnik, K. Egiazarian, and S. Member. Image denoising by sparse 3d transform-domain collaborative filtering. *TIP*, 16:2007, 2007.
- [8] M. Elad and M. Aharon. Image denoising via learned dictionaries and sparse representation. In *CVPR*, 2006.
- [9] A. Foi, V. Katkovnik, and K. Egiazarian. Pointwise shape-adaptive dct for high-quality denoising and deblocking of grayscale and color images. *TIP*, 2007.
- [10] A. Gupta, P. Bhat, M. Dontcheva, B. Curless, O. Deussen, and M. Cohen. Enhancing and experiencing spacetime resolution with videos and stills. In *International Conference on Computational Photography*, 2009.
- [11] Y. S. Heo, K. M. Lee, and S. U. Lee. Simultaneous depth reconstruction and restoration of noisy stereo images using non-local pixel distribution. In *CVPR*, pages 1–8, 2007.
- [12] C. Liu. *Beyond Pixels: Exploring New Representations and Applications for Motion Analysis*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 2009.
- [13] C. Liu and W. T. Freeman. A high-quality video denoising algorithm based on reliable motion estimation. In *ECCV*, 2010.
- [14] S. Lyu and E. P. Simoncelli. Statistical modeling of images with fields of gaussian scale mixtures. In *NIPS*, 2006.
- [15] H. Nagaharuf, T. Matsunobuf, Y. Iwaif, M. Yachidaf, and T. Suzuki. High-resolution video generation using morphing. In *ICPR*, 2006.
- [16] S. Roth and M. J. Black. Fields of experts: A framework for learning image priors. In *CVPR*, pages 860–867, 2005.
- [17] F. Schubert and K. Mikolajczyk. Combining high-resolution images with low-quality videos. In *BMVC08*, 2008.
- [18] M. F. Tappen, C. Liu, E. H. Adelson, and W. T. Freeman. Learning gaussian conditional random fields for low-level vision. In *CVPR*, pages 1–8, 2007.
- [19] V. Vaish, M. Levoy, R. Szeliski, C. L. Zitnick, and S. B. Kang. Reconstructing occluded surfaces using synthetic apertures: Stereo, focus and robust measures. In *CVPR*, 2006.
- [20] K. Watanabe, Y. Iwai, H. Nagahara, M. Yachida, and T. Suzuki. Video synthesis with high spatio-temporal resolution using motion compensation and spectral fusion. *IEICE - Trans. Inf. Syst.*, E89-D, July 2006.
- [21] L. Zhang, S. Vaddadi, H. Jin, and S. Nayar. Multiple view image denoising. In *CVPR*, 2009.