Chapter 2
High-Quality Video Denoising for Motion-Based Exposure Control

Li Zhang, Travis Portz and Hongrui Jiang

Abstract New digital cameras, such as Canon SD1100 and Nikon COOLPIX S8100, have an autoexposure (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 Dabov et al. TIP 16:2007, [1], or assume reliable motion estimation as input, such as Liu, ECCV, 2010, [2], our method evaluates the reliability of flow at each pixel and uses the "lifespan" of reliable flow trajectories 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 Dabov et al. TIP 16:2007, Liu, ECCV, 2010, [1, 2].

L. Zhang (✉)
Google, 651 North 34th Street, Seattle, WA 98103, USA
e-mail: zhl@google.com

T. Portz
Amazon.com, Seattle, WA, USA
e-mail: travis@travisportz.com

H. Jiang
University of Wisconsin, Engineering Hall, 1415 Engineering Drive,
Madison, WI 53706, USA
e-mail: hongrui@engr.wisc.edu

G. Hua and X.-S. Hua (eds.), *Mobile Cloud Visual Media Computing*,
DOI 10.1007/978-3-319-24702-1_2

2.1 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 autoexposure (AE). Traditionally, AE is mainly determined by environment *brightness*: bright scenes lead to short exposure time. 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.

As more computing power is put in digital cameras, new cameras, such as Canon SD1100 and Nikon COOLPIX S8100, have an AE function that is based on motion estimation. In a nutshell, the apparent motion estimated from two consecutive frames is used to guide the exposure time and ISO setting for the next frame, so that blur is minimized. In the captured video, most frames do not have blur, but those with short exposure time will be noisy due to the high ISO setting. This AE function largely turns video enhancement into a denoising problem.

This chapter presents a research work on 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 Fig. 2.1 shows; it is also difficult in its own ways.

- Within a sequence captured using motion-based AE, 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 chapter, 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 [1] 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,

[1] For 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.

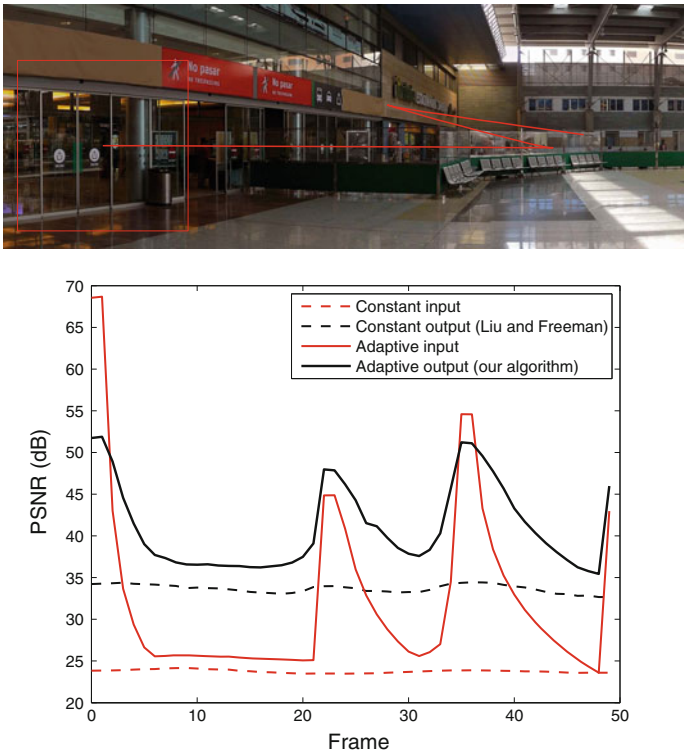

Fig. 2.1 The benefit of denoising videos captured with motion-based exposure control. *Top*: A panoramic image from which we generate a synthetic video whose viewport (*red box*) moves along the red zigzag curve with varying speed. *Bottom*: If a constant short exposure is applied to each frame to minimize blur, the captured video has constant low PSNR (*dashed red curve*), and a state-of-the-art video denoising [2] improves its PSNR to about 34dB (*dashed black curve*). If exposure time is set *adaptively* based on motion estimation, the input video has higher PSNR (*solid red 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

our idea is to detect the length of reliable flow trajectories for each pixel and use the length 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 [1], or assume reliable motion estimation as input, such as [2], our method selectively operates in whichever regime works best. As a result, our algorithm performs better than both VBM3D [3] and the latest video denoising algorithm [2].

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.

2.2 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 [1, 4] for excellent reviews of the literature. An incomplete list of recent works includes [1, 4–9]. In particular, the methods that are based on local self-similarity, such as nonlocal means [4] and BM3D [1], are particularly notable because of their simple ideas and impressive results. The nonlocal means and BM3D methods do not perform well when local image patterns are unique.

Video denoising [2, 3, 10, 11] can address this limitation as the temporal dimension provides additional redundant data. Liu and Freeman [2] showed that the spatial regularization in the optical flow can be used to ensure temporal coherence in removing structured noise. Multi-view denoising [12–14] is another way of addressing this limitation, which exploits noisy measurements from multiple viewpoints to reconstruct a clean image. Zhang et al. [14] 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 [2], in which the authors integrate robust optical flow into a nonlocal 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. [15] and Schubert et al. [16] proposed an approach to enhance low-resolution videos of a static scene using multi-view stereo to compute correspondences between low-resolution video and high-resolution images; Gupta et al. [17] use optical flow to compute correspondences and can therefore handle dynamic scenes as well. Watanabe et al. [18] propagate high-frequency information in high-resolution frames to low-resolution frames using motion compensation. Nagahara et al. [19] 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 lifespan of reliable flow trajectory to combine spatial denoising and temporal denoising.

2.3 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 [1] to perform our spatial denoising:

$$\hat{I}_S(\mathbf{z}) = \mathrm{CBM3D}(I, \mathbf{z}), \tag{2.1}$$

where I is the input image and $\mathbf{z}$ 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 CVMB3D 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 nonlocal means [4], can also be used instead.

Temporal Denoising Along Reliable Flow

Let I_t be the frame we are currently denoising. We compute the optical flow over a temporal window of $\pm H$ frames, where we use $H = 5$ as in [2]. 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 (1 and 3 are used in our synthetic and real experiments, respectively).

For each pixel in frame I_t, we determine the number of frames of consistent forward flow up to at most frame $t + H$, and backward flow down to at most $t - H$. If the per pixel flow is not consistent at frame $t + 1$, we do not consider frame $t + 2$ for that pixel. The number of consistent frames in the forward and backward directions are denoted H_f and H_b, respectively. H_f and H_b are functions of the pixel under consideration; however, we omit the function notation for simplicity.

Once we have determined the "lifespan" $[t - H_b, t + H_f]$ of a reliable flow, the temporal pixel estimate is computed by filtering along the optical flow:

$$\hat{I}_T(\mathbf{z}) = \frac{1}{Z} \sum_{i=t-H_b}^{t+H_f} W(\mathbf{z}_i) \cdot I_i(\mathbf{z}_i), \tag{2.2}$$

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

$$W(\mathbf{z}_i) = \left(\beta_i^2 + \beta_t^2\right)^{-\frac{3}{2}} \exp\left\{-\frac{\|P(\mathbf{z}) - P(\mathbf{z}_i)\|^2}{\beta_i^2 + \beta_t^2}\right\}, \tag{2.3}$$

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. (2.3), 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 is computed using a weighted SSD as in [2].

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

$$\tau_t = m \cdot \beta_t + \tau_0, \tag{2.4}$$

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 $m = 0.40$ and $\tau_0 = -5.3 \cdot 10^{-4}$.

Combining Spatial and Temporal Denoising

To combine the spatial and temporal denoising results, we linearly interpolate using the number of consecutive frames of flow consistency $H_b + H_f$ as the weight:

$$\hat{I}(\mathbf{z}) = \frac{H_f + H_b}{2H} \hat{I}_T(\mathbf{z}) + \left(1 - \frac{H_f + H_b}{2H}\right) \hat{I}_S(\mathbf{z}). \tag{2.5}$$

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.

2.3.1 *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 [20], which typically handle flow magnitudes of 10 or fewer pixels. For example, we tried the flow algorithm [21] used in [2] as input for video denoising. The algorithm does

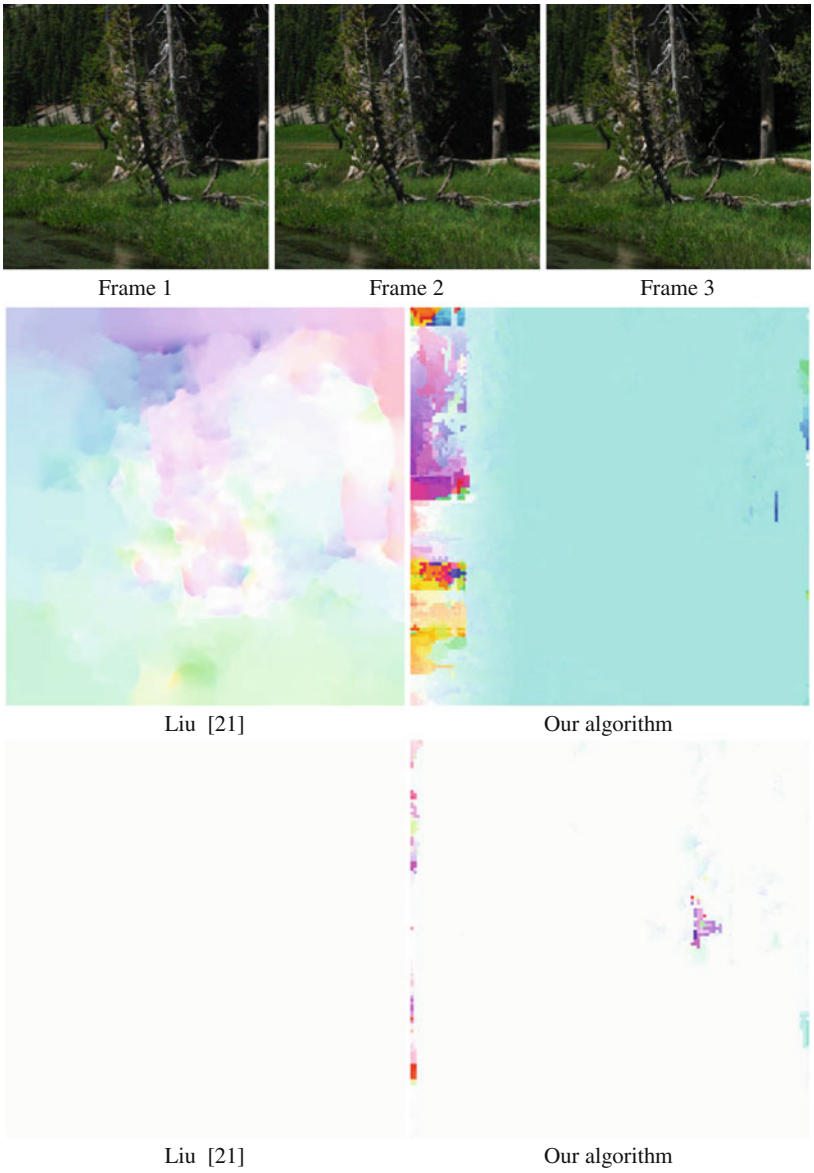

Fig. 2.2 Optical flow results for three consecutive frames in the *mountain* scene. *Top*: The displacement between frames 1 and 2 is large, whereas the displacement between frames 2 and 3 is small. *Middle*: Our optical flow outperforms the optical flow in [21] for large displacements. The left to right 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 [21] outperforms our method for small displacements by producing a smoother flow. Best viewed electronically in color

not produce correct flow for a typical pair of frames with large motion as shown in Fig. 2.2. 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 fall back to a traditional hierarchical block matching technique to compute our optical flow.

Suppose we are computing the optical flow from frame i to j. We start by constructing image pyramids of downsampled versions of I_i and I_j with L levels, where the coarsest level has been downsample by a factor of 2^{L-1}. We then compute a flow field for the coarsest level by performing block matching between the two downsampled images using search windows of size $M \times M$. The choice of M determines the size of motion the algorithm can handle. Performing the primary search at the coarsest level effectively reduces the size of the search space necessary to find matches for large motions. We use $L = 3$ and $M = 61$, allowing us to handle displacements of up to 120 pixels between consecutive frames.

Next, we upsample the flow field by a factor of two and refine it by searching within the next coarsest level of the pyramid. If $\mathbf{v}$ was a flow vector in the coarsest level, then $2\mathbf{v}$ is the flow vector in the next level. This upsampling and refinement step is repeated until we have a flow field that is the same size as our original images. The refinement step is necessary to obtain better resolution and accuracy in our flow field than is possible using only the coarsest level. However, the search window used during refinement can be much smaller than the window used at the coarsest level; we use a 7×7 search window for refinement.

We first compute flow between neighboring frames, then concatenate the flow to initialize motion estimation between the reference frame t to any other frame i between $[t - H, t + H]$, and finally refine the initialization by block matching in the finest resolution only. We found this simple method works quite well for handling large motion; an example of the flow result is shown in Fig. 2.2.

2.4 Experimental Results

Our results are best viewed electronically in color. More results, including videos, are available in the supplementary material.

2.4.1 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 Fig. 2.3. 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

City scene Mountain scene

Fig. 2.3 Our synthetic video sequences are generated from panoramic images. A 512 × 512 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

each frame. If d is the displacement between the previous two frames and f is the frame rate, then

$$T = \frac{1}{d \cdot f}. \tag{2.6}$$

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 set the gain to:

$$g = \frac{T_{\max}}{T} \tag{2.7}$$

such that the video sequence maintains 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.

We run the input sequences through CBM3D and Liu and Freeman [2] using the known σ parameters for each frame. For our algorithm, we use $\beta_0 = 0.1$ in Eq. (2.3) (with pixel intensities in the range [0,1]) and specify the gain values for the individual frames. 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 Fig. 2.4 for the *city* and *mountain* sequences and in Fig. 2.1 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. Our results do have lower PSNR for frames that were very clean to begin with. However, the difference is imperceptible with our results having a mean square error of only about 10^{-3} of an intensity level in the cleaner frames.

The improvements in our results over CBM3D are primarily made in the regions with unique texture and structure, as can be seen in Figs. 2.5 and 2.6. In these regions, the optical flow is reliable, thus temporal denoising is effective. The weights between the temporal and spatial estimates are shown in Fig. 2.7. In smooth regions where our optical flow is unreliable, our denoising algorithm falls back on CBM3D which performs well on smooth regions.

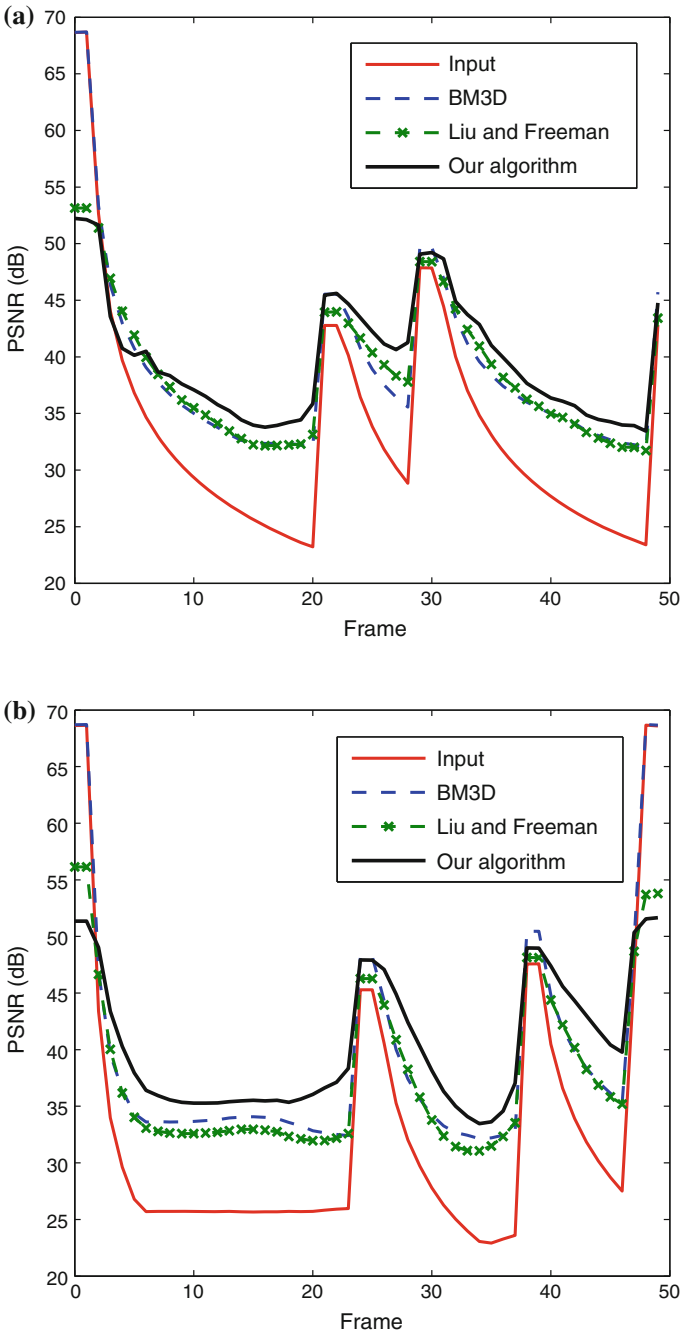

Fig. 2.4 PSNR results for the synthetic video sequences. In frames with significant noise levels, our algorithm outperforms other state-of-the-art denoising algorithms. Best viewed electronically in color. **a** City scene. **b** Mountain scene

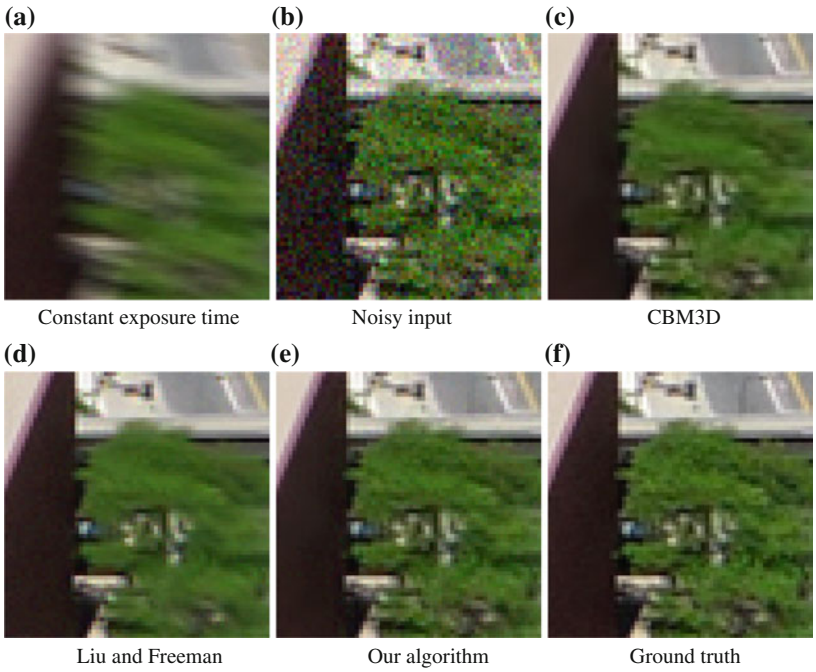

Fig. 2.5 A close-up of results from the *city* sequence. The motion-based AE provides a sharp but noisy image, shown in (**b**), as opposed to the blurry image captured with a constant exposure time, shown in (**a**). Our denoising algorithm outperforms CBM3D [1] (applied to each individual frame using corresponding frame noise variance) and Liu and Freeman [2] (using the known noise variance for each individual frame). More detail is preserved in the tree while the building is still properly smoothed. Best viewed electronically in color

2.4.2 Real Video

To test our system on a real video sequence, we first needed motion-based exposure control. We implemented the motion estimation portion of the exposure control algorithm using a standard hierarchical image registration technique. The remainder of the exposure control 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. We used one camera, a Canon EOS 7D, to capture a video sequence with a constant exposure time of 1/30 s and the other camera, a Point Grey Grasshopper, to capture a video sequence with motion-based exposure control. As shown in Fig. 2.8, our algorithm preserves detail better than [2], because optical flow is hard to be estimated 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.

More results are available in the supplementary material.

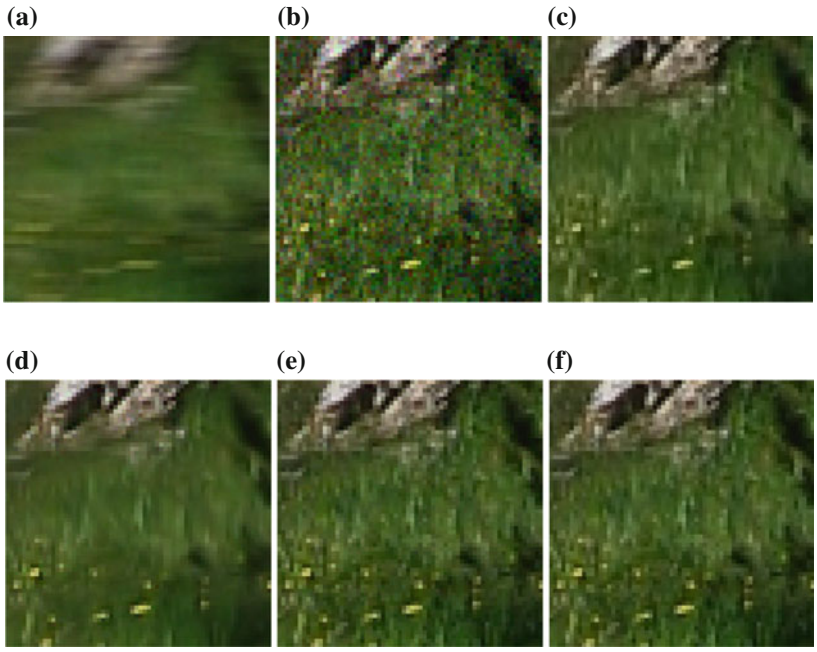

Fig. 2.6 A close-up of results from the *mountain* sequence. CBM3D [1] loses some detail in the yellow flowers, while [2] over-smooths the grass. Our algorithm performs better at denoising both the flowers and the grass. Best viewed electronically in color. **a** Constant exposure time. **b** Noisy input. **c** CBM3D. **d** Liu and Freeman. **e** Our algorithm. **f** Ground truth

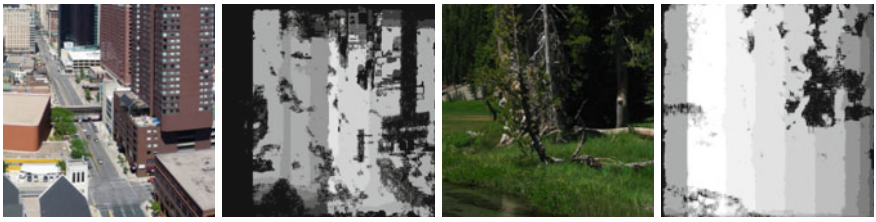

Fig. 2.7 Two weight maps from the synthetic sequences. Lighter colors denote pixels that rely more on temporal denoising than spatial denoising. The darker regions in the weight maps correspond to smooth regions of the image where optical flow trajectory is less reliable. The horizontal motion in the 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

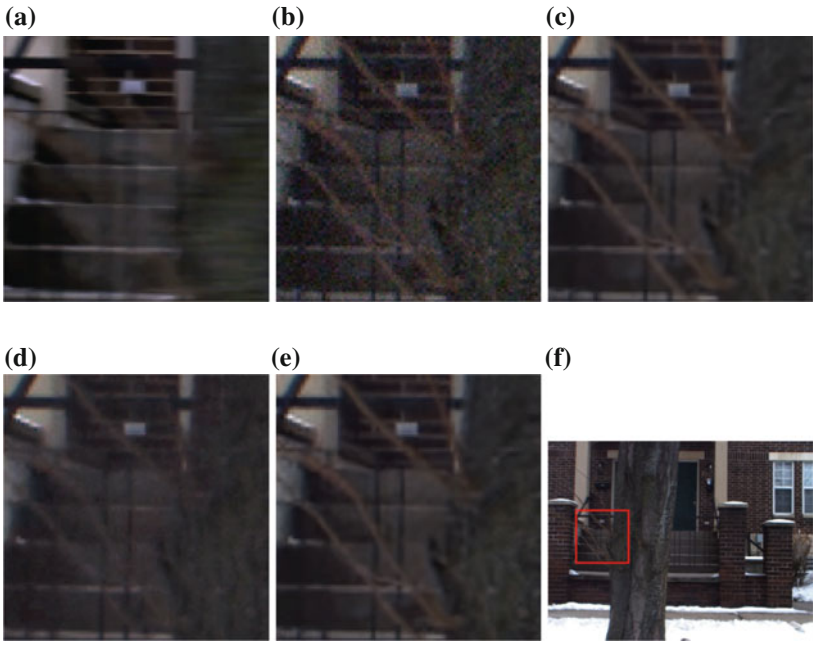

Fig. 2.8 Results from the driving sequence. Our results are comparable to CBM3D [1], which preserves the detail of the tree reasonably well. The tree branches and some of the other fine details were over-smoothed by Liu and Freeman [2] due to inaccurate flow in the presence of large motion, multiple depth layers, and thin structure. Best viewed electronically in color. **a** Constant exposure time. **b** Noisy input. **c** CBM3D. **d** Liu and Freeman. **e** Our algorithm. **f** Full denoised frame

2.5 Conclusion

In this chapter, we have proposed a high-quality video denoising algorithm in the context of motion-based exposure control. Unlike previous denoising works which either avoid using motion estimation, such as BM3D [1], or assume reliable motion estimation as input, such as [2], our method uses the "lifespan" of reliable flow trajectory 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 the lifespan of a reliable flow is zero, the algorithm resorts to CBM3D; in this case, temporal coherence is not enforced. This differs from [2], which uses smooth optical flow to obtain temporal consistency in the presence of structural noise. However, as Figs. 2.5, 2.6, and 2.8 show, this temporal consistency is obtained at the

expense of sacrificing texture details. Therefore our method is suited for higher quality cameras with little compression artifact, while [2] is more suited for low-quality cameras with strong structured noise and compression artifacts. 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, a better weighting scheme would address this limitation.

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.

References

1. Dabov, K., Foi, R., Katkovnik, V., Egiazarian, K., Member, S.: Image denoising by sparse 3D transform-domain collaborative filtering. TIP **16**, 1395–1411 (2007)
2. Liu, C., Freeman, W.T.: 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 (ECCV). Springer, Berlin (2010). http://portal.acm.org/citation.cfm?id=1927006.1927061
3. Dabov, K., Foi, A., Egiazarian, K.: Video denoising by sparse 3D transform-domain collaborative filtering. In: Proceedings of the 15th European Signal Processing Conference (2007)
4. Buades, A., Coll, B., Morel, J.M.: A review of image denoising algorithms, with a new one. Simulation **4**, 490–530 (2005)
5. Roth, S., Black, M.J.: Fields of Experts: A framework for learning image priors. CVPR **2**, 860–867 (2005)
6. Elad, M., Aharon, M.: Image Denoising Via Learned Dictionaries and Sparse representation. CVPR pp. 895–900 (2006)
7. Lyu, S., Simoncelli, E.P.: Statistical modeling of images with fields of gaussian scale mixtures. NIPS (2006)
8. Tappen, M.F., Liu, C., Adelson, E.H., Freeman, W.T.: Learning gaussian conditional random fields for low-level vision. CVPR, pp. 1–8. IEEE Computer Society, CA, USA (2007). http://doi.ieeecomputersociety.org/10.1109/CVPR.2007.382979
9. Foi, A., Katkovnik, V., Egiazarian, K.: Pointwise shape-adaptive DCT for high-quality denoising and deblocking of grayscale and color images. TIP **16**(8), 2080–2095 (2007)
10. Bennett, E.P., McMillan, L.: Video enhancement using per-pixel virtual exposures. SIGGRAPH pp. 845–852. ACM, California, New York (2005). http://doi.acm.org/10.1145/1186822.1073272
11. Chen, J., Tang, C.K.: Spatio-temporal markov random field for video denoising. CVPR (2007)
12. Vaish, V., Levoy, M., Szeliski, R., Zitnick, C.L., Kang, S.B.: Reconstructing occluded surfaces using synthetic apertures: stereo, focus and robust measures. CVPR **2**, 1063–6919. IEEE Computer Society, CA, USA (2006). http://doi.ieeecomputersociety.org/10.1109/CVPR.2006.244

13. Heo, Y.S., Lee, K.M., Lee, S.U.: Simultaneous depth reconstruction and restoration of noisy stereo images using Non-local pixel distribution. CVPR pp. 1–8. IEEE Computer Society, CA, USA (2007)
14. Zhang, L., Vaddadi, S., Jin, H., Nayar, S.: Multiple view image denoising. CVPR, IEEE Computer Society, CA, USA (2009). http://doi.ieeecomputersociety.org/10.1109/CVPRW.2009.5206836
15. Bhat, P., Zitnick, C.L., Snavely, N., Agarwala, A., Agrawala, M., Curless, B., Cohen, M., Kang, S.B.: Using photographs to enhance videos of a static scene. In: Kautz J., Pattanaik S. (eds.) Proceedings of the Eurographics Symposium on Rendering (2007). http://www.cs.washington.edu/homes/pro/papers/videoEnhancement/videoEnhancement.htm
16. Schubert, F., Mikolajczyk, K.: Combining high-resolution images with low-quality videos. BMVC08 pp. 1–10 (2008) http://www.visionbib.com/bibliography/match-pl503.html#TT48849
17. Gupta, A., Bhat, P., Dontcheva, M., Curless, B., Deussen, O., Cohen, M.: Enhancing and experiencing spacetime resolution with videos and stills. In: International Conference on Computational Photography (2009) http://grail.cs.washington.edu/projects/enhancing-spacetime/
18. Watanabe, K., Iwai, Y., Nagahara, H., Yachida, M., Suzuki, T.: Video synthesis with high spatio-temporal resolution using motion compensation and spectral fusion. IEICE—Trans. Inf. Syst. **E89–D**, pp. 2186–2196, Oxford University Press, Oxford, UK (2006). http://portal.acm.org/citation.cfm?id=1184860.1185056
19. Nagaharaf, H., Matsunobuf, T., Iwaif, Y., Yachidaf, M., Suzuki, T.: High-resolution video generation using morphing. ICPR **4**, 338–341 (2006) doi:10.1109/ICPR.2006.626
20. Baker, S., Scharstein, D., Lewis, J., Roth, S., Black, M.J., Szeliski, R.: A database and evaluation methodology for optical flow. ICCV (2007)
21. Liu, C.: Beyond pixels: Exploring new representations and applications for motion analysis. Dissertation, Massachusetts Institute of Technology, Cambridge (2009)

http://www.springer.com/978-3-319-24700-7

Mobile Cloud Visual Media Computing
From Interaction to Service
Hua, G.; Hua, X.-S. (Eds.)
2015, X, 353 p., Hardcover
ISBN: 978-3-319-24700-7