

KERNEL-FREE VIDEO DEBLURRING VIA SYNTHESIS

Feitong Tan, Shuaicheng Liu, Liaoyuan Zeng, Bing Zeng

School of Electronic Engineering
University of Electronic Science and Technology of China, Chengdu, China

ABSTRACT

Shaky cameras often capture videos with motion blurs, especially when the light is insufficient (e.g., dimly-lit indoor environment or outdoor in a cloudy day). In this paper, we present a framework that can restore blurry frames effectively by synthesizing the details from sharp frames. The uniqueness of our approach is that we do not require blur kernels, which are needed previously either for deconvolution or convolving with sharp frames before patch matching. We develop this kernel-free method mainly because accurate kernel estimation is challenging due to noises, depth variations, and dynamic objects. Our method compares a blur patch directly against sharp candidates, in which the nearest neighbor matches can be recovered with sufficient accuracy for the deblurring. Moreover, to restore one blurry frame, instead of searching over a number of nearby sharp frames, we only search from a synthesized sharp frame that is merged by different regions from different sharp frames via an MRF-based region selection. Our experiments show that this method achieves a competitive quality in comparison with the state-of-the-art approaches with an improved efficiency and robustness.

Index Terms— Video deblurring, blur kernel, patch match, synthesis, nearest neighbors.

1. INTRODUCTION

Videos captured from hand-held devices often appear shaky, thus rendering videos with jittery motions and blurry contents. Video stabilization [1] can smooth the jittery camera motions, but leaves the video blurs untouched. Motion blurs tend to happen when videos are captured in a low-light environment. Due to the nature of the camera shakes, however, not all frames are equally blurred [2, 3]. Moderate shakes often deliver relatively sharper frames, while drastic motions yield strong blurs. In this work, we attempt to use sharp frames to synthesize the blurry ones. Please refer the project page for videos.¹

Traditional image deblurring methods estimate a uniform blur kernel [4, 5, 6, 7] or spatially varying blur kernels [8, 9, 10, 11, 12, 13, 14], and deblur the frames using different penalty terms [4, 6, 7] by maximizing the posterior

distribution [5, 15] of the latent image during image deconvolution [15, 16]. However, such a deconvolution can introduce ringing artifacts due to the inaccuracy of blur kernels. Moreover, it is time-consuming to deconvolve every single frame through the entire video [3].

Video frames usually contain complementary information that can be exploited for deblurring [2, 3, 17, 18, 19]. Cho *et al.* [3] presented a framework that transfers sharp details to blurry frames by patch synthesis. Zhang *et al.* [18] jointly estimated motion and blur across multiple frames, yielding deblurred frames together with optical flows. Kim *et al.* [19] focused on dynamic objects. However, all these methods estimate blur kernels and rely on them heavily for deblurring, while the blur kernel estimation on casually captured videos is often challenging due to depth variations, noises, and moving objects.

The nearest neighbor match between image patches, referred to as “patch match”(PM) [20, 21, 22], finds the most similar patch for a given patch in a different image region. In our context, we divide a blurry frame into regular blur patches. For each blur patch, we find the most likely sharp patches in sharp frames to replace it. Therefore, the quality of deblurring is dominated by the accuracy of PM. Traditional approaches [2, 3] estimate the blur kernel from the blurry frame and use it to convolve the sharp frames before PM. We refer this process as “convolutional patch match” (CPM). Our method directly compares a blur patch with sharp patches, which is referred as “direct patch match” (DPM). Intuitively, DPM will deliver inaccurate nearest neighbor matches, as the matched patches are under different conditions - one is blur and the other is sharp. In our work, however, we will provide both empirical and practical evidences of DPM for its high quality and performance in video deblurring.

In this work, we propose a synthesis-based approach that neither estimates kernels nor performs deconvolutions. Specifically, we first locate all blurry frames in a video. For every blurry frame, we find the nearby sharp frames. To deblur one frame, we adopt a process of pre-alignment that roughly aligns all sharp frames to the target blurry frame before the DPM searching. Moreover, instead of search over all sharp frames, we only search over a synthesized sharp frame that is fused from different regions of sharp frames through an “Markov random field” (MRF) region selection that ensures

¹<http://www.liushuaicheng.org/ICIP2016/deblurring/index.html>

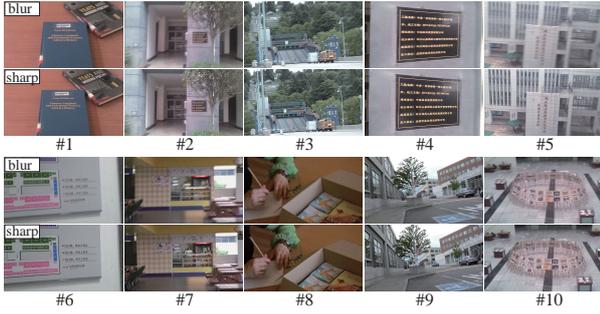


Fig. 1. The thumbnail of 10 pairs of frames selected for experiment. In each pair, the top row and bottom row show the blur and sharp frames, respectively.

the spatial and temporal coherence. Each blur patch will find one sharp patch, which is used to synthesize the deblurred frame. Notably, the key differences between our method and [3] is that we do not estimate blur kernels and only search in merged sharp frames. In summary, our contributions are:

- We propose to use DPM for a synthesis-based video deblurring, which is free from the challenges of blur kernel estimations and image deconvolutions.
- With pre-alignment and region selection, we only search in a limited search space, which highly accelerates the whole system.
- We show that the pre-alignment not only reduces the search space, but also increases the accuracy of DPM.

2. ANALYSIS

We first conduct an experiment to show the difference between CPM and DPM, both visually and numerically. In PM, the “sum of squared differences” (SSD) is the commonly adopted metric [20] in evaluating patch distances. We adopt this metric in our evaluation. We collect 10 blurry frames as well as their neighboring sharp frames from 10 videos, covering static/dynamic scenes, planar/variation depths. Fig. 1 shows the blurry and sharp frame pairs and all frames have resolution of 1280×720 . For CPM, we estimate the blur kernels by the approach [23]. We collect patches with size 21×21 for every two pixel in the blurry frame and assign the search region of 31×31 in the corresponding position at the target sharp frame. We conduct the nearest neighbor search and record the best match index for both methods.

For each blur patch, the L_2 distance is calculated between indexes obtained by two methods. The averaged index differences over all blur patches are shown in Tab. 2. We further record the averaged patch SSD. While the DPM method produces a larger SSD as compared with CPM in all examples, the index difference remains small. In fact, all we want are the correct indexes, instead of the actual SSD. Therefore, we adopt DPM in our system.



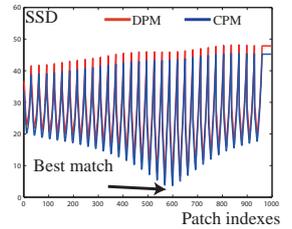
Fig. 2. Results by CPM and DPM. In each pair, the top and bottom row show the result by CPM and DPM, respectively.

	#1	#2	#3	#4	#5
Avg. index diff.	1.80	0.94	1.82	1.51	2.51
Avg. CPM SSD	3.02	3.22	5.97	3.09	5.97
Avg. DPM SSD	4.84	5.91	10.78	4.79	13.25
	#6	#7	#8	#9	#10
Avg. index diff.	1.88	0.87	1.95	0.94	0.87
Avg. CPM SSD	4.80	0.49	0.47	0.32	0.50
Avg. DPM SSD	7.81	0.75	0.57	0.45	0.76

Table 1. Averaged index differences and patch errors.

Fig. 2 shows the visual comparison of the deblurred results by two methods. The small differences in index would not introduce visual differences in the deblurred results, implying that such small index differences are negligible.

Finally, the figure on the right shows what happens during one patch search. Here, the x-axis denotes the sharp patch’s index in the search region and the y-axis shows the SSD value. The CPM method (blue curve) does produce a smaller SSD as compared with the DPM method (red curve), but they both yield the same best match index.



3. OUR METHOD

Fig. 3 shows our system pipeline of deblurring one frame. We first locate the blurry frames and the sharp frames according to the gradient magnitude [3]. Fig. 3 (a) shows a piece of video volume with one blurry frame (red border) and its surrounding sharp frames (blue border). Then, we align all sharp frames to the blurry frame (Fig. 3 (b)) by matching the features and the mesh warping. Notably, the pre-alignment is inaccurate. It can only compensate a rough global camera motion. The accuracy will be improved by DPM. Then, we choose the sharpest regions from all aligned sharp frames to produce a sharp map (Fig. 3 (c) bottom), which is searched by blur patches from the blurry frames (Fig. 3 (c) top). The final result is shown at Fig. 3 (d).

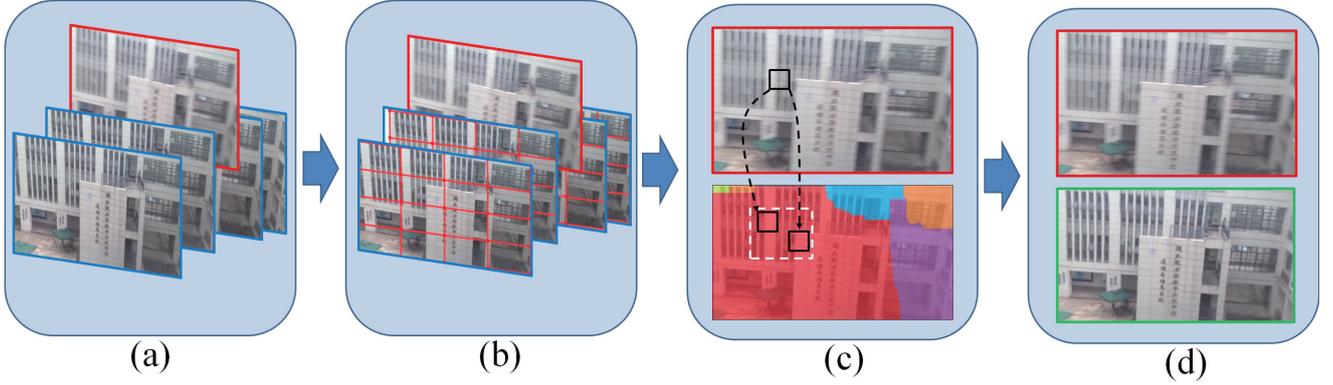


Fig. 3. The pipeline of deblurring one frame in our method. (a) A blurry frame (red border) and its nearby sharp frames (blue border). (b) Sharp frames are aligned with the blurry frame by mesh warping. (c) A sharp map, generated by sharp regions from sharp frames, with different colors representing different frames. (d) The deblurred result.

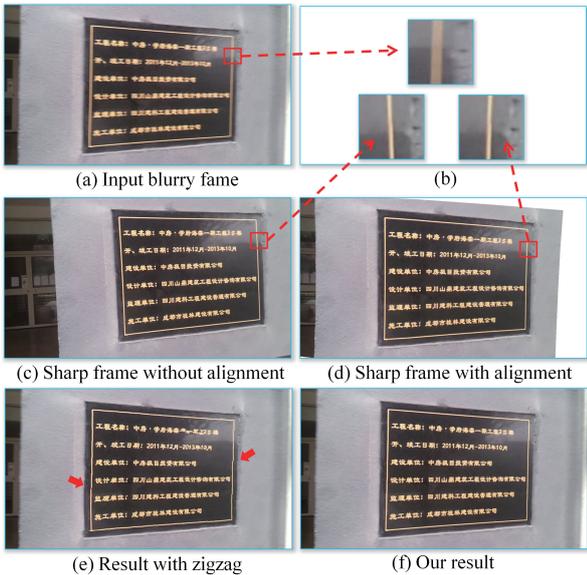


Fig. 4. Deblurred results with and without pre-alignment.

3.1. Pre-alignment

We detect features on the sharp frame [24] and track them to the blurry frame [25]. The mesh-based warping [26, 27] is adopted to warp the sharp frame based on the tracked features. More advanced approaches can be considered [28]. Though the alignment quality is limited due to the inaccuracy of tracking in blur, it can successfully compensate the global camera motion which is similar to searching across translations, rotations and scales [21]. Without it, we only search the translational space. Fig. 4 shows a comparison of deblurring one frame with and without pre-alignment. As shown in Fig. 4 (b), without the alignment, the best matched sharp patch is slightly leaned, accumulating to zigzag artifacts (Fig. 4 (e)).

3.2. Region Selection

To improve the efficiency, we only search for a sparse regular grid of pixels (every 6 pixels), instead of all pixels. This sparse sampling can avoid over smoothing where a pixel is covered by many patches if sampled densely. Each sharp frame provides a search region for a blur patch. In general, we can search all of them. For speed, we only search from one sharpest region among all sharp regions.

Suppose that the grid is a graph $\zeta = \langle \nu, \varepsilon \rangle$, where ν is the set of all nodes and ε is the set of all arcs connecting adjacent nodes. The nodes are blur patch centers and the arcs denote four connected neighboring system. Different sharp frames have different labels $\{x_i\}$. We want to assign a unique label x_i for each node $i \in \nu$. The solution $X = \{x_i\}$ can be obtained by minimizing the energy function:

$$E(X) = \sum_{i \in \nu} E_1(x_i) + \lambda \sum_{(i,j) \in \varepsilon} E_2(x_i, x_j). \quad (1)$$

Here, $E_1(x_i)$ evaluates the sharpness of a region and is calculated as the averaged gradient magnitude within a patch, and E_2 encourages the spatial coherence. If a blur patch searches in the sharp frame t , we want its neighboring blur patches to search in the same sharp frame t . $E_2(x_i, x_j)$ is defined as:

$$\begin{cases} x_i = x_j, & E_2 = 0. \\ x_i \neq x_j, & E_2 = 500. \end{cases} \quad (2)$$

In our implementation, we set $\lambda = 1$ and the number of labels as 8. Specifically, for a blurry frame, we find 4 sharp frames in the future and another 4 in the past. The energy can be minimized efficiently via graph cut [29], which produces a sharp map for the subsequent DPM search.

3.3. Synthesis

We apply the DPM search according to the sharp map, after which each blur patch can find one sharp patch S_k , which is used to replace the blur patch. The sharp patches may have



Fig. 5. Our deblurred results on various casually captured videos.

overlaps so that a pixel receives multiple values. The final value of a pixel is calculated as:

$$p(i, j) = \frac{1}{Z} \sum_{p(i, j) \in S_k} w_k S_k(i', j'), \quad (3)$$

where k indexes sharp patches and (i, j) indexes pixels, (i', j') refers to the same pixel (i, j) in the patch local coordinates, w_k is a weight that is calculated for each patch to be proportional to the gradient magnitude of a patch, and Z is a normalization factor, computed as $Z = \sum_{p(i, j) \in S_k} w_k$. Here, we want to assign more weights to a sharper patch. More advanced approaches can be adopted [30, 31].

3.4. Details and Speed

Without pre-alignment, we have to search from a large region as the camera is shaky. Now, we can only search from a small region (Fig. 3 (c), white rectangle), centered at the same location of a blur patch in the sharp frame. During DPM, we set the patch size as 21×21 pixels. At synthesis, we choose a relatively smaller patch size 12×12 to reduce pixel overlaps. The search region has size 15×15 . We run our method on an Intel i7 2.4GHz CPU. We deblur a frame with resolution 1280×720 in 10 seconds (1 second for MRF and the rest for DPM). In [3], with an Intel i7 CPU, a frame with the same resolution is deblurred around one minute. Our method can be further accelerated by parallel processing (e.g., GPU).

3.5. Iterative Refinement

Deblurring every blurry frame consists of a pass. In most situations, one pass is enough. However, for situations where no sharp frames are detected around a blurry frame, we need to take multiple passes. Specifically, to propagate frame details across a longer range, we adopt iterative scheme where a deblurred frame can be considered as sharp frame to deblur the remaining blurry frames in the next iteration.

4. EXPERIMENTS AND DISCUSSIONS

We compare our method with the state-of-art method [3] in Fig. 6. Clearly, our method produces a sharper result (some

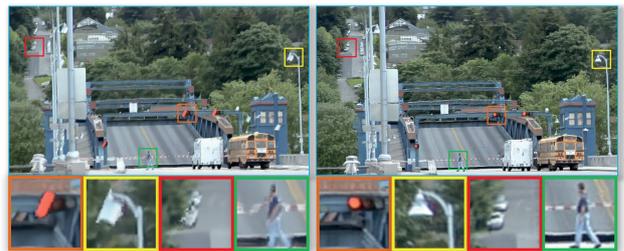


Fig. 6. Comparison with the state-of-art method [3]. The left shows the result of [3] obtained from their project page. The right shows our result.

of the regions are highlighted). In the method of [3], it cannot handle excessive shaky clips. With the pre-alignment, we can deblur those footages properly. If blur kernels are estimated correctly, both DPM and CPM have similar effects in deblurring. However, if the kernel estimation goes wrong, CPM can introduce some side-effects (e.g., zigzag, ghosting). In a sense, DPM is similar to “Ostrich algorithm” by ignoring the difficulties. However, such neglecting does not introduce any substantial harms to the results, which becomes the most interesting part of this paper. More results are shown in Fig. 5.

5. CONCLUSION

We have presented a synthesis-based video deblurring framework that restores blurry frames from nearby sharp frames. We found that our proposed DPM can successfully approximate CPM and works well in practice. Without forward convolution or deconvolution, our method is simple yet effective. We use the pre-alignment and the sharp map to reduce the search space, which not only increase the efficiency but also improve the accuracy of DPM. Moreover, the proposed method is scalable for parallel computing. Its robustness has been tested over various challenging videos.

6. ACKNOWLEDGE

This work has been supported by National Natural Science Foundation of China (61502079 and 61370148).

References

- [1] Y. Matsushita, E. Ofek, W. Ge, X. Tang, and H. Shum, "Full-frame video stabilization with motion inpainting," *IEEE PAMI*, vol. 28, pp. 1150–1163, 2006.
- [2] Y. Huang and N. Fan, "Inter-frame information transfer via projection onto convex set for video deblurring," *IEEE J. of Selected Topics in Signal Processing*, vol. 2, no. 5, pp. 275–284, 2011.
- [3] S. Cho, J. Wang, and S. Lee, "Video deblurring for handheld cameras using patch-based synthesis," *ACM TOG*, vol. 31, no. 4, pp. 64:1–64:9, 2012.
- [4] S. Cho and S. Lee, "Fast motion deblurring," in *ACM TOG*, 2009, vol. 28, p. 145.
- [5] R. Fergus, B. Singh, A. Hertzmann, S. Roweis, and W. Freeman, "Removing camera shake from a single photograph," *ACM TOG*, vol. 25, no. 3, pp. 787–794, 2006.
- [6] Q. Shan, J. Jia, and A. Agarwala, "High-quality motion deblurring from a single image," *ACM TOG*, vol. 27, no. 3, pp. 73, 2008.
- [7] L. Xu and J. Jia, "Two-phase kernel estimation for robust motion deblurring," in *ECCV*, pp. 157–170, 2010.
- [8] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce, "Non-uniform deblurring for shaken images," *IJCV*, vol. 98, no. 2, pp. 168–186, 2012.
- [9] M. Hirsch, C. J. Schuler, S. Harmeling, and B. Schölkopf, "Fast removal of non-uniform camera shake," in *ICCV*, 2011, pp. 463–470.
- [10] S. Cho, Y. Matsushita, and S. Lee, "Removing non-uniform motion blur from images," in *ICCV*, 2007, pp. 1–8.
- [11] Z. Hu and M.-H. Yang, "Fast non-uniform deblurring using constrained camera pose subspace," in *BMVC*, 2012, pp. 1–11.
- [12] L. Xu, S. Zheng, and J. Jia, "Unnatural l0 sparse representation for natural image deblurring," in *CVPR*, 2013, pp. 1107–1114.
- [13] H. Zhang and D. Wipf, "Non-uniform camera shake removal using a spatially-adaptive sparse penalty," in *NIPS*, 2013, pp. 1556–1564.
- [14] S. Su and W. Heidrich, "Rolling shutter motion deblurring," in *CVPR*, 2015, pp. 1529–1537.
- [15] A. Levin, Y. Weiss, F. Durand, and W. Freeman, "Efficient marginal likelihood optimization in blind deconvolution," in *CVPR. IEEE*, 2011, pp. 2657–2664.
- [16] L. Yuan, J. Sun, L. Quan, and H. Shum, "Progressive inter-scale and intra-scale non-blind image deconvolution," *ACM TOG*, vol. 27, no. 3, pp. 74, 2008.
- [17] K. Sunkavalli, N. Joshi, S.B. Kang, M. Cohen, and H. Pfister, "Video snapshots: Creating high-quality images from video clips," *IEEE TVCG*, vol. 18, no. 11, pp. 1868–1879, 2012.
- [18] H. Zhang and J. Yang, "Intra-frame deblurring by leveraging inter-frame camera motion," in *CVPR*, 2015, pp. 4036–4044.
- [19] T. Kim and K. Lee, "Generalized video deblurring for dynamic scenes," in *CVPR*, 2015, pp. 5426–5434.
- [20] C. Barnes, E. Shechtman, A. Finkelstein, and D. Goldman, "Patchmatch: A randomized correspondence algorithm for structural image editing," *ACM TOG*, vol. 28, no. 3, pp. 24, 2009.
- [21] C. Barnes, E. Shechtman, D. Goldman, and A. Finkelstein, "The generalized patchmatch correspondence algorithm," in *ECCV*, pp. 29–43, 2010.
- [22] C. Barnes, F. Zhang, L. Lou, X. Wu, and S. Hu, "Patchtable: efficient patch queries for large datasets and applications," *ACM TOG*, vol. 34, no. 4, pp. 97, 2015.
- [23] Z. Su, J. Pan, Z. Hu, and M.-H. Yang, "Deblurring text images via l0-regularized intensity and gradient prior," in *CVPR*, 2014.
- [24] E. Rosten, R. Porter, and T. Drummond, "Faster and better: A machine learning approach to corner detection," *IEEE PAMI*, vol. 32, no. 1, pp. 105–119, 2010.
- [25] J. Shi and C. Tomasi, "Good features to track," in *CVPR*, 1994, pp. 593–600.
- [26] S. Liu, L. Yuan, P. Tan, and J. Sun, "Bundled camera paths for video stabilization," *ACM TOG*, vol. 32, no. 4, pp. 78, 2013.
- [27] T. Igarashi, T. Moscovich, and J. Hughes, "As-rigid-as-possible shape manipulation," *ACM TOG*, vol. 24, no. 3, pp. 1134–1141, 2005.
- [28] L. Yuan, J. Sun, L. Quan, and H.-Y. Shum, "Blurred/non-blurred image alignment using sparseness prior," in *ICCV*, 2007, pp. 1–8.
- [29] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE PAMI*, vol. 26, no. 9, pp. 1124–1137, 2004.
- [30] V. Kwatra, I. Essa, A. Bobick, and N. Kwatra, "Texture optimization for example-based synthesis," *ACM TOG*, vol. 24, no. 3, pp. 795–802, 2005.
- [31] W. Freeman, E. Pasztor, and O. Carmichael, "Learning low-level vision," *IJCV*, vol. 40, no. 1, pp. 25–47, 2000.