Generating Sharp Panoramas from Motion-blurred Videos

Yunpeng Li[1]   Sing Bing Kang[2]   Neel Joshi[2]   Steve M. Seitz[3]   Daniel P. Huttenlocher[1]

[1] Cornell University, Ithaca, NY 14853, USA {yuli,dph}@cs.cornell.edu
[2] Microsoft Research, Redmond, WA 98052, USA {sbkang,neel}@microsoft.com
[3] University of Washington, Seattle, WA 98195, USA seitz@cs.washington.edu

Abstract

In this paper, we show how to generate a sharp panorama from a set of motion-blurred video frames. Our technique is based on joint global motion estimation and multi-frame deblurring. It also automatically computes the duty cycle of the video, namely the percentage of time between frames that is actually exposure time. The duty cycle is necessary for allowing the blur kernels to be accurately extracted and then removed. We demonstrate our technique on a number of videos.

1. Introduction

A convenient way to generate a panorama is to take a video while panning and then stitch the frames using a commercial tool such as AutoStitch, Hugin, Autodesk Stitcher, or Microsoft Image Composite Editor. However if there is significant camera motion, the frames in the video can be very blurry. Stitching these frames will result in a blurry panorama, as shown in Figure 1 (b). In this paper, we describe a new technique that is capable of generate sharp panoramas such as that shown in (c).

Our framework assumes that the scene is static and adequately far away from the camera. Hence the apparent motion and motion blur in the video are mainly due to camera rotation. This allows us to parameterize the image motion as a homography [18]. Moreover, we assume that the camera motion is piecewise linear (i.e., the velocity is constant between successive frames). This is a reasonable approximation for videos due to their high capture rate.

We pose the problem of generating a sharp panorama from a sequence of blurry input photos as that of estimating the camera motion, its duty cycle, and the sharpened images, where the motion and the duty cycle give us the blur kernel for sharpening. In our approach, all these are estimated jointly by minimizing an energy function in a multi-image deconvolution framework, which we shall describe
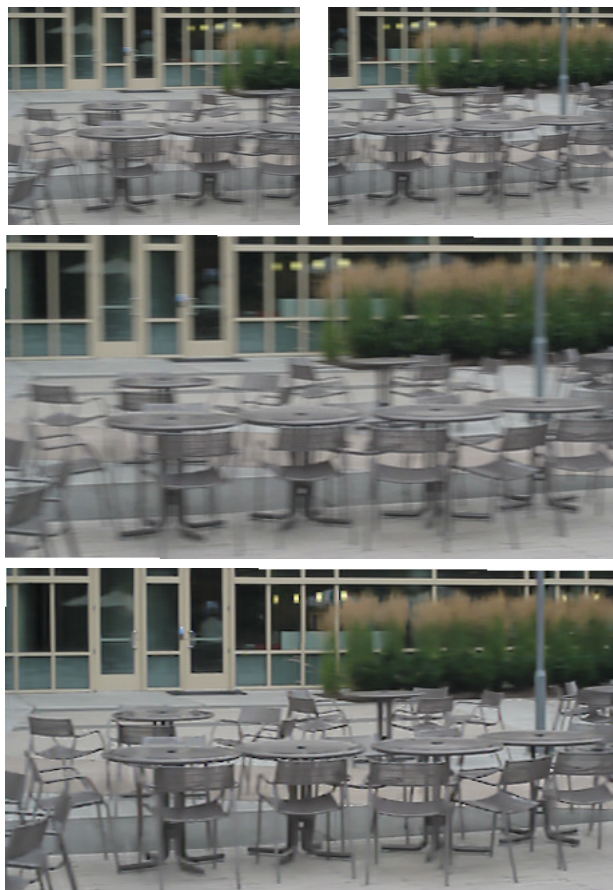
Figure 1. Stitching example. First row: (a) Input frames (only first and last frames shown). Second row: (b) Result of directly stitching the input frames. Third row: (c) Result of our technique.

in detail in later sections. Note that the blur kernel in our model, though parameterized by global motion, is in fact spatially varying, which is a necessary consequence of the modeling of camera rotation.

The main contributions of this paper are: (1) the ability to estimate camera duty cycles from blurred videos, and (2) the formulation as a single joint optimization of duty cy-

Blur kernel# images	Single-image	Multi-image
Spatially constant	[17]	[2]
Multi. piecewise-const.	[8]	[7], [5], [15]
Spatially varying	[6]	[4], Ours

Table 1. Categorization of deblurring techniques, illustrated with some of the representative works. Not a complete taxonomy

cle, motion and latent deblurred images in a manner that is computationally tractable.

2. Related Work

Image and video deblurring has been studied extensively in the context of computer vision, graphics, and signal processing. Table 1 lists representative deblurring techniques in our brief survey, which is not meant to be exhaustive.

Multi-image techniques. Jin *et al.* [7] address the problem of simultaneous tracking and deblurring in motion blurred image sequences. They use the observation that motion blur operations are commutative to match blur across frames for tracking. However, the blur kernels estimated in this way are only relative, namely they satisfy the proposed blur constraint but are not necessarily the actual kernels that produced the input images. Moreover, the work assumes that the blur kernel can be modeled as a 1D Gaussian, which is not typical of real-world motion blur.

Cho *et al.*'s work [5] uses segmentation masks to simultaneously estimate multiple (parametric) motion blurs over the whole image as well as handle occlusion. It can be regarded as an extension of the previous technique [7], since it uses a similar blur constraint and also assumes that the blur kernel is a 1D Gaussian. While capable of estimating multiple motions and blurs, it shares most of the assumptions of [7] and hence has similar limitations.

Bascle *et al.* [4] jointly perform deblurring and super-resolution on image sequences using a generative back-projection technique. The technique assumes that the 1D motion that produces blur is completely determined by affine motion between images. This, however, is only true when the camera duty cycle, i.e., the relative exposure time, is known *a priori*, and their work does not address the issue of estimating duty cycles.

The work that has the most similar goal to ours is that of Onogi and Saito [15]; they also deblur video frames for panorama generation. In their case, they perform feature tracking with the assumption that there are only foreground and background global motions. Like Bascle *et al.*, they derive the blur directly from the estimated motions, implicitly assuming known duty cycle, and invert each frame *independently* using the Wiener filter. In contrast, our approach is jointly estimate motion and duty cycle, and performs multi-

image deblurring.

Single-image techniques. Many single-image deblurring techniques have been proposed, and here, we mention only a small number of representative methods. One of the more recent techniques is that of Shan *et al.* [17], where the blur is due to camera shake and assumed to be spatially invariant. While impressive results were reported, it appears that their system parameters are highly data-dependent, which limits its practicality.

Levin [9] removes motion blur from a single image by predicting derivative distributions as a function of the blur kernel width, assuming the motion blur is caused by constant velocity motion. Joshi *et al.* [8] model spatially-varying motion blur by dividing the image into non-overlapping sub-windows and estimating a blur kernel for each window. The method uses hypotheses of sharp edges instead of motion cues for blur estimation.

Dai and Wu [6] derive a local motion blur constraint similar to the optical flow constraint. Their constraint is based on matte extraction and is capable of recovering spatially-variant motion blur. However, it is sensitive to matting errors.

Techniques with specialized capture process. There are also techniques for deblurring that either use specialized hardware or assume a highly controlled capture process. For instance, Tai *et al.* [19] use simultaneously captured high-resolution video at low frame rate and low-resolution video at high frame rate. Yuan *et al.* [21] use a blurred long-exposure image and a noisy short-exposure image of the same scene. Tico and Vehvilainen [20] address the problem of motion blur by taking many short exposure (and hence high noise) images so that they are relatively free from blur. However, it does not actually model blur and hence is not applicable to inputs that has a contain substantial amount of blur.

Agrawal *et al.* [2] uses multi-image deconvolution to address the problem of zeros in the frequency domain of blur kernels. This is done by jointly deblurring a sequence of images that are taken with different and known exposure times. The method also requires that the scene consists of a single moving object at constant velocity and static background. While such requirements are not uncommon in the deblurring literature [16, 5], they are not typically met in most real-world videos.

In contrast, our inputs are videos from readily available off-the-shelf digital equipment.
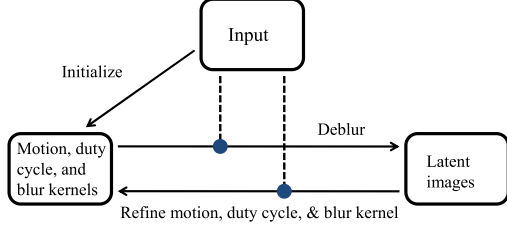
Figure 2. Overview of our method

3. Overview

We solve the deblurring problem by jointly estimating motion, duty cycles, and latent (sharp) images. [1] This is performed by minimizing an energy function corresponding to multi-image deconvolution. We will describe the energy function in the following section.

The energy minimization is carried out using gradient descent. This is viable under our model since we can compute the derivative of the energy function with respect to motion, duty cycles, as well as latent images. We initialize motion by computing global parametric optical flow between successive video frames using the Lucas-Kanade method [13]. Duty cycles are set to an initial guess, which needs not be accurate. Subsequently, we alternate between updating latent images while holding motion and duty cycles constant (i.e., performing deconvolution) and updating motion and duty cycles while holding the latent images constant. See Figure 2 for illustration. The updates are described in Section 4 and 5 respectively. Although in theory all three sets of variables can be optimized simultaneously, the alternating scheme has a much smaller and hence manageable memory footprint and is effective in practice.

4. Multi-image Deconvolution

Let $I_1, \cdots, I_n$ be the sequence of observed images with motion blur and $L_1, \cdots, L_n$ be the corresponding underlying latent images without motion blur. [2] If we regard each image as a m-dimensional vector, where m is the number of pixels, then the spatially varying motion blur can be represented as sparse m-by-m matrices $B_1, \cdots, B_n$, namely

$$I_i = B_i L_i + N_i \tag{1}$$

for each image $i \in \{1, \cdots, n\}$, where N_i is the noise. Recall that B_i is parameterized by motion (i.e., homographies) and duty cycles under our model. Similarly, let $A_{i,j}$ denote the warping to frame i from frame j, which is determined also by the relative motion, i.e.,

$$L_i = A_{i,j} L_j. \tag{2}$$

---

[1] In our framework, motion and duty cycles parameterize blur kernels (which will be described in the next section).

[2] Assume images have been converted into linear (luminance) space.

Hence

$$I_i = B_i A_{i,j} L_j + N_i. \tag{3}$$

Assuming Gaussian noise, the maximum-likelihood estimate for frame j is then obtained by minimizing the energy function

$$E_{\mathrm{ML}}(L_j) = \sum_{i=j_-}^{j_+} \left|\left| D_i^{-1}(B_i A_{i,j} L_j - I_i) \right|\right|^2, \tag{4}$$

where $j_- = \max(j - r, 1)$, $j_+ = \min(j + r, n)$, D_i is a diagonal matrix whose entries are the standard deviations of noise at each pixel in the i-th image, and r is the number of nearby observations to include in each temporal direction. In our work, r is typically in the range of 1 to 3. Note that if $r = 0$, the problem reduces to single image deconvolution. Because of noise in the observation I_i as well as empirical errors in the recovered warping $A_{i,j}$ and blur B_i, a common approach is to introducing an image prior on L that typically regularizes its gradients (e.g. [10]). Hence the maximum *a posteriori* estimate corresponds to the minimum of the energy function

$$E_{\mathrm{MAP}}(L_j) = \sum_{i=j_-}^{j_+} \left|\left| D_i^{-1}(B_i A_{i,j} L_j - I_i) \right|\right|^2 + \rho(L_j), \tag{5}$$

where $\rho(\cdot)$ is the functional form of the prior. The overall energy function can be minimized with respect the latent images L_j using gradient-based MAP-deconvolution techniques (e.g. [10]).

5. Motion and Duty Cycle Estimation

In this section, we describe how to refine motion and duty cycles given the latent images. Again, let $\mathbf{I} = (I_1, \cdots, I_n)$ be the blurred video frames and $\mathbf{L} = (L_1, \cdots, L_n)$ be the underlying sharp frames that we want to recover. Let $\mathbf{H} = (H_1, \cdots, H_n)$ be the warps to each frame from some reference frame. Let $\boldsymbol{\tau} = (\tau_1, \cdots, \tau_n)$ denote the duty cycles of each frame. We denote $\boldsymbol{\theta} = (\mathbf{H}, \boldsymbol{\tau})$ for notational convenience. Hence both $A_{i,j}$ and B_i (defined in the previous section) are functions of $\mathbf{H}$ and $\boldsymbol{\tau}$, and we will subsequently write them as $A_{i,j}^{\boldsymbol{\theta}}$ and $B_i^{\boldsymbol{\theta}}$ to reflect this. Since the correct warps and duty cycles should result in a deblurred output with lower energy than incorrect ones, it is desirable to minimize Equation (5) over the whole sequence with respect to these variables as well. Hence we aim to minimize the following energy function

$$E(\mathbf{L}, \boldsymbol{\theta}) = \sum_{j=1}^{n} \sum_{i=j_-}^{j_+} \left|\left| D_i^{-1}(B_i^{\boldsymbol{\theta}} A_{i,j}^{\boldsymbol{\theta}} L_j - I_i) \right|\right|^2 + \rho(L_j). \tag{6}$$

The minimization of Equation (6) with respect to $\mathbf{L}$ amounts to MAP-deconvolution, which we already addressed in the

previous section; therefore the rest of this section will describe how to minimize it with respect to $\boldsymbol{\theta}$.

5.1. Pure Translation

We start with pure translation for simplicity of presentation. In this case, the warps $\mathbf{H}$ can be represented by the corresponding 2-component vectors $\mathbf{h} = (\mathbf{h_1}, \cdots, \mathbf{h_n})$. Thus we let $\boldsymbol{\theta} = (\mathbf{h}, \boldsymbol{\tau})$ for pure translation. Since the image prior $\rho(L_j)$ does not depend on $\boldsymbol{\theta}$, it can be ignored as far as the optimization of $\boldsymbol{\theta}$ is concerned. Also for conciseness we omit the noise matrices D^{-1} from the subsequent notation, since it is simply a weighting factor on each pixel. Furthermore, we will write $E(\mathbf{L}, \boldsymbol{\theta})$ as simply $E(\boldsymbol{\theta})$ since minimization is with respect to $\boldsymbol{\theta}$. Hence

$$E(\boldsymbol{\theta}) = \sum_{j=1}^{n} \sum_{i=j_-}^{j_+} \left\| B_i^{\boldsymbol{\theta}} A_{i,j}^{\boldsymbol{\theta}} L_j - I_i \right\|^2 \qquad (7)$$

Let

$$L_{i,j}^{\boldsymbol{\theta}} = A_{i,j}^{\boldsymbol{\theta}} L_j \qquad (8)$$

and

$$I_{i,j}^{\boldsymbol{\theta}} = B_i^{\boldsymbol{\theta}} L_{i,j}^{\boldsymbol{\theta}} = B_i^{\boldsymbol{\theta}} A_{i,j}^{\boldsymbol{\theta}} L_j, \qquad (9)$$

i.e., $L_{i,j}^{\boldsymbol{\theta}}$ is the sharp frame i obtained by warping L_j, and $I_{i,j}^{\boldsymbol{\theta}}$ is the blurred version of $L_{i,j}^{\boldsymbol{\theta}}$. Therefore

$$E(\boldsymbol{\theta}) = \sum_{j=1}^{n} \sum_{i=j_-}^{j_+} \sum_{p \in \text{pixels of } I_i} \delta I_{i,j,\mathbf{p}}(\boldsymbol{\theta}) \qquad (10)$$

where

$$\delta I_{i,j,\mathbf{p}}(\boldsymbol{\theta}) = (I_{i,j}^{\boldsymbol{\theta}}(\mathbf{p}) - I_i(\mathbf{p}))^2. \qquad (11)$$

Here we use $\mathbf{p}$ to denote the 2-component vector representing a pixel's location $(x_p, y_p)^T$ in the image. Thus it suffices to find the derivative of $\delta I_{i,j,\mathbf{p}}(\boldsymbol{\theta})$, which in turn depends on the derivative of $I_{i,j}^{\boldsymbol{\theta}}(\mathbf{p})$ (with respect to $\boldsymbol{\theta}$). Recall that in our model the blur kernel is determined by the relative warps to the two adjacent frames due the assumption of piecewise linear motion, i.e.,

$$I_{i,j}^{\boldsymbol{\theta}}(\mathbf{p}) = \int_{t=-\tau_i}^{\tau_i} L_{i,j}^{\boldsymbol{\theta}}(\mathbf{p} + t(\mathbf{h_{i+\text{sign}(t)}} - \mathbf{h_i}))dt, \qquad (12)$$

where $\text{sign}(t)$ is 1 if t is positive and -1 otherwise. Thus it can be approximated by a sequence of sampled points on the motion path for each point,

$$I_{i,j}^{\boldsymbol{\theta}}(\mathbf{p}) = \frac{1}{2s+1} \sum_{k=-s}^{s} L_{i,j}^{\boldsymbol{\theta}}(\mathbf{p} + \qquad (13)$$
$$\frac{|k|}{2s+1}\tau_i(\mathbf{h_{i+\text{sign}(k)}} - \mathbf{h_i}))$$
$$= \frac{1}{2s+1} \sum_{k=-s}^{s} L_j(\mathbf{p} +$$
$$\frac{|k|}{2s+1}\tau_i(\mathbf{h_{i+\text{sign}(k)}} - \mathbf{h_i}) + (\mathbf{h_i} - \mathbf{h_j}))$$

where constant s is the number of samples in each temporal direction (independent of the duty cycles) used to approximate the blur kernel. In our case $s = 50$. The derivative of $I_{i,j}^{\boldsymbol{\theta}}(\mathbf{p})$ is

$$\frac{d}{d\boldsymbol{\theta}} I_{i,j}^{\boldsymbol{\theta}}(\mathbf{p}) = \nabla L_j(\mathbf{p_{i,j,k}^{\boldsymbol{\theta}}}) \cdot \frac{1}{2s+1} \sum_{k=-s}^{s} \frac{d}{d\boldsymbol{\theta}} \mathbf{p_{i,j,k}^{\boldsymbol{\theta}}}, \quad (14)$$

where

$$\mathbf{p_{i,j,k}^{\boldsymbol{\theta}}} = \mathbf{p} + \frac{|k|}{2s+1}\tau_i(\mathbf{h_{i+\text{sign}(k)}} - \mathbf{h_i}) + (\mathbf{h_i} - \mathbf{h_j}) \quad (15)$$

and ∇L_j is the image gradient of L_j. In the case of $i = n$ and $k > 0$, $\mathbf{h_i} - \mathbf{h_{i+1}}$ is replaced with $\mathbf{h_{i-1}} - \mathbf{h_i}$ as an approximation (since $\mathbf{h_{n+1}}$ does not exist). The case of $i = 1$ and $k < 0$ is handled similarly.

5.2. Full Homography

The case for a homography is analogous. Recall that $\mathbf{H} = (H_1, \cdots, H_n)$ defines the warps to each frame from some reference frame, and the relative warp to frame i from frame j is thus $H_{i,j} = H_j^{-1}H_i$. Let $\mathbf{p}$ now denote the homogeneous coordinates of a pixel, i.e., $\mathbf{p} = (x_p, y_p, 1)^T$. Thus to extend the relationship between the energy function and $\boldsymbol{\theta}$ from translation to general homography, we only need to rewrite Eqn (14) and (15), so that

$$I_{i,j}^{\boldsymbol{\theta}}(\mathbf{p}) = \frac{1}{2s+1} \sum_{k=-s}^{s} L_{i,j}^{\boldsymbol{\theta}}(\mathbf{p_{i,j,k}^{\boldsymbol{\theta}}}) \qquad (16)$$

with

$$\mathbf{p_{i,j,k}^{\boldsymbol{\theta}}} = \phi(H_{i,j}\left[\mathbf{p} + \frac{|k|}{2s+1}\tau_i\left(\phi(H_{i+\text{sign}(k),i} \cdot \mathbf{p}) - \mathbf{p}\right)\right]) \qquad (17)$$

where $\phi(\cdot)$ is the projection of points in homogeneous coordinates onto the image plane $z = 1$, i.e.,

$$\phi((x, y, z)^T) = (\frac{x}{z}, \frac{y}{z}, 1)^T. \qquad (18)$$

Equation 17 uses the approximation that every pixel is moving at constant velocity in between successive frames. The approximation is reasonable for videos since one can expect the perspective change between successive frames to be small. We make the further approximation that $\phi(H_{i+\text{sign}(k),i} \cdot \mathbf{p}) \approx H_{i+\text{sign}(k),i} \cdot \mathbf{p}$, which is valid since the perspective change in $H_{i+\text{sign}(k),i}$ is small. Hence Equation (17) simplifies to

$$\mathbf{p_{i,j,k}^{\boldsymbol{\theta}}} = \phi(H_j^{-1}\left[H_i + \frac{|k|}{2s+1}\tau_i\left(H_{i+\text{sign}(k)} - H_i\right)\right]\mathbf{p}). \qquad (19)$$

The derivatives with respect to warps and duty cycles can be obtained using standard matrix calculus with the chain rule. Hence, the energy function can be minimized via gradient-based optimization methods (L-BFGS [14] in our case).

Sparsity prior + IRLS   SLRF prior + L-BFGS

Figure 3. Sample output from deblurring using the sparsity prior
[10] optimized by IRLS and the SLRF prior [11] optimized by
L-BFGS, with otherwise identical parameters. The latter is notice-
ably sharper and yet has less ringing artifacts.

6. Experiments

We evaluate our model on both synthetic and real
videos. For energy minimization, we use limited-memory
BFGS (L-BFGS) of [14]. Compared with iterative re-
weighted least square (IRLS), L-BFGS does not have the
re-weighting step and requires only the value and the deriva-
tive of the energy function. In our experiments we found it
to be converge slightly faster than IRLS and reach compara-
ble energy levels. As is outlined in Section 4, an image prior
is used to regularize the output. In particular we use the
recently proposed sparse long-range random field (SLRF)
[11], which we found to produce noticeably sharper images
and fewer ringing artifacts than the sparsity prior [10] (Fig-
ure 3). [3] The noise levels of the input videos are automati-
cally estimated using the method of Liu *et al.* [12].

6.1. Synthetic Videos

For quantitative evaluation, we generated synthetic
blurred videos with ground truth information. This is
done by moving a virtual video camera in front of a high-
resolution image, which serves as the scene to be recorded.
Motion blur is generated by temporally averaging consec-
utive frames. Since the virtual camera can have an arbi-
trarily high frame rate, we have dense enough samples to
accurately approximate the motion blur kernel. We output
the middle frame of the samples (of successive frames) that
are used to produce the blurred frame as the corresponding
"true" sharp frame.

Figure 4 shows a sample frame from deblurring a se-
quence with synthesized motion blur. The goal is to com-
pare multi-image deconvolution against single-image de-
convolution. In this particular experiment, motion from an
initial motion estimation and the known duty cycle value
were used. Hence no iterative refinement was performed.
We also did not use any image prior in this case since
the purpose here is to compare multi-image deconvolution
against the single-image counterpart, the two of which dif-

---

[3]All the figures are best viewed electronically with zooming-in.

Ground truth   Blurred

Single-image deconvolution   Multi-image deconvolution

Figure 4. Sample output frame from synthetic video sequence. The
camera undergoes translational motion. The result of multi-image
deconvolution is sharper and has fewer ringing artifacts than that
of single-image deconvolution.

fer only in the data term (i.e., the part of the energy func-
tion excluding the image prior). For this experiment, 7
blurred frames are used for reconstructing each deblurred
frame for multi-image deconvolution (the temporal radius
$r = 3$). The results show that the image restored using
multi-image deconvolution exhibits noticeably fewer ring-
ing artifacts, demonstrating the advantage of using multiple
observations.

In Figure 5, the top-left image shows a blurred frame of a
synthetic video where the camera undergoes in-plane rota-
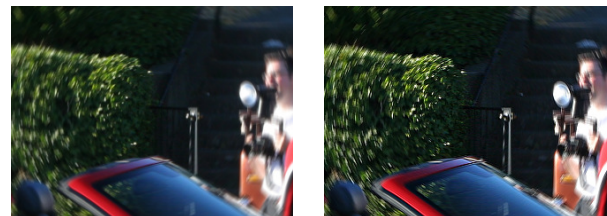
Blurry frame   Result of Shan *et al.* [17]
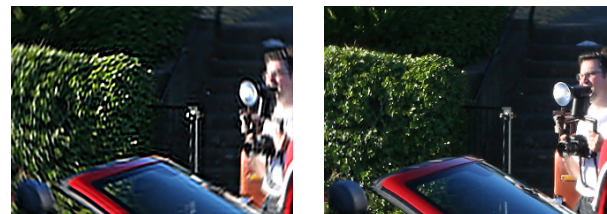
Result of Joshi *et al.* [8]   Our result

Figure 5. Sample output frame from synthetic video sequence. The
amera undergoes in-plane rotation. Our model is capable of han-
dling such challenging scenarios by modeling spatially varying
blur and utilizing motion motion cues from the video.

tion about its center. As can be seen from the image, the blur is highly non-uniform. The amount of blur is lower near the center and higher near the border of the image, and the direction of blur also changes with location. The bottom-right image shows the corresponding frame of the deblurred output using our method, which is sharp and clean. In contrast, models using piece-wise constant blur kernels can not produce satisfactory results on such sequences (e.g. top-right, the result of Shan *et al.* [17], and bottom-left, the result of Joshi *et al.* [8]). This demonstrates the importance of being able to model spatially varying motion blur.

Figure 6 shows the errors of motion, duty cycles, and restored frames, over the whole 11-frame video sequence, plotted against the number of iterations completed. The plots correspond to four different sequences with combinations of two types of camera motions (translation/perspective rotation) and two duty cycle values values (approximately half/full). A temporal radius of 1 is used in these experiments. Motion errors are expressed as average end-point errors [3] of warps between successive frames; restoration errors are in the range between 0 and 255. In all these experiments, the duty cycles are initialized to 1 for the entire sequence. As can be observed from the figures, the duty cycles converge to the correct values even when the initial value is quite far away from the true values. In the case where the initialization is close to the ground truth, the estimated values remain close to the true values over the iterations. The experiments show that the estimation of duty cycles is quite robust. Finding the correct duty cycles is essential for determining the blur kernel size and subsequently achieving good deblur performance. This is reflected in the plots, where the restoration error is highly correlated with the duty cycle error. Figure 7 shows a sample frame of the deblur output using the initial incorrect duty cycles and one using the refined duty cycles (after the last iterations). The difference is emphatic.

One may notice that the decrease in motion error is far less than the decrease in duty cycle and restoration errors, and that the sequences with perspective rotation have higher motion error than the translation sequences. This is likely due to the approximations in handling homographies, as described in Section 5. This, however, does not pose a problem, since the motion estimation is still adequately accurate for reliable restoration.

6.2. Real Videos

Real-world video sequences are simply collected using the video mode of a point-and-shoot camera. We used a Canon SD1100, a Canon A540, and a FlipHD for this purpose. For the SD1100 and the A540, we performed gamma correction using the camera response functions (CRF) from the web [1] that were obtained by calibrating on a set of benchmark scenes and taking the average. We were not
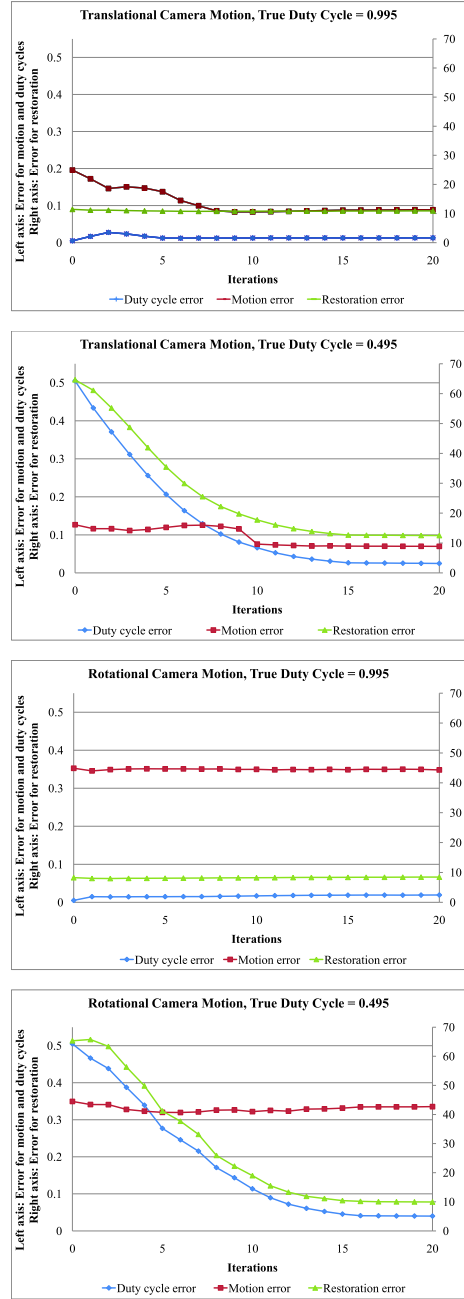

Figure 6. Errors of motion, duty cycles, and restoration (i.e., deblur output) on synthetic video sequences, plotted against the number iterations completed. For the last two, the camera rotation is out-of-plane, hence perspective change

able to find the CRF for the FlipHD, and thus simply used a gamma value of 1.8. The step is necessary for real videos since the CRF can be highly non-linear [1]. For all experiments, initial duty cycle was set to one of $1/4$, $1/2$, or 1, depending on if the sequence appears to have a mild, moderate, or high degree of blur, and a temporal radius of $r = 1$ is used.
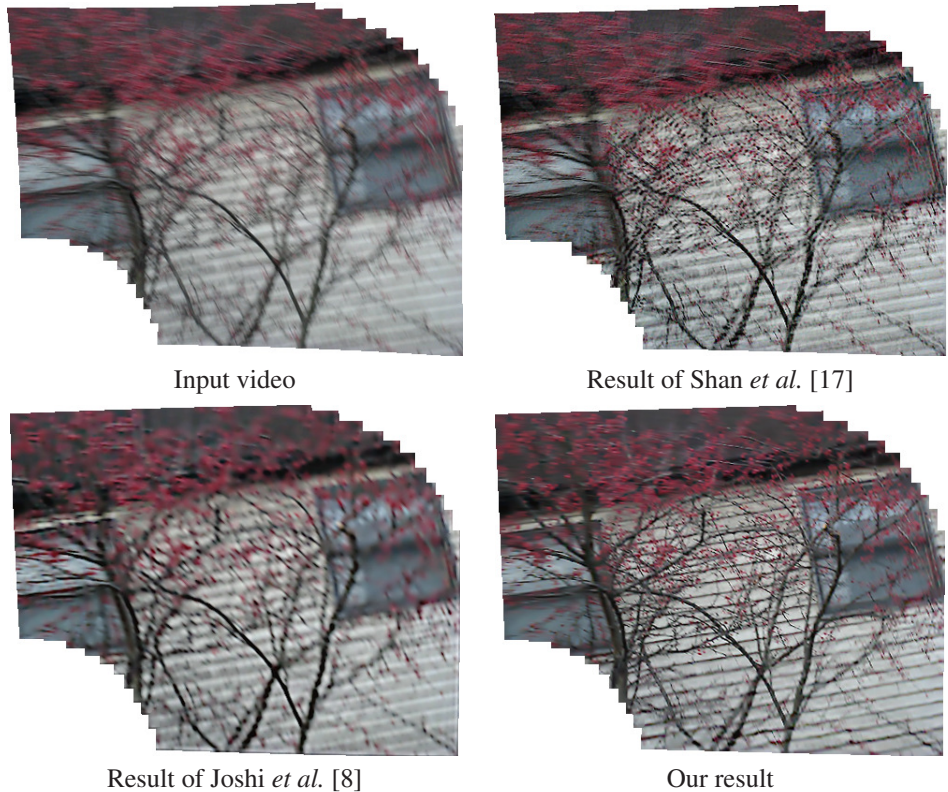
Input video

Result of Shan *et al.* [17]

Result of Joshi *et al.* [8]

Our result

Figure 8. Stitched Panoramas. The input video was taken by a Canon SD1100. Gamma correction is performed for all three methods.

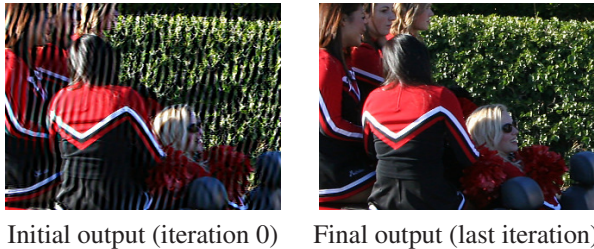Initial output (iteration 0)   Final output (last iteration)

Figure 7. Comparison of sample frames from the initial and the final deblurred output on the synthetic sequence with rotational camera motion and a true duty cycle value of 0.495 (the last plot in Figure 6). The initial duty cycle value is 1.

Figure 8 shows the panorama stitched from a blurry video and the those stitched from the deblurred outputs of Shan *et al.* [17], Joshi *et al.* [8], and our method respectively, using Microsoft Composite Image Editor (ICE). Our method produces far more pleasant result than the competing methods. Figure 9 displays addition sample frames of the deblurred outputs on real videos, showing the enhancement over the inputs.

Our method may fail to produce satisfactory results in certain scenarios, which typically involve hash lighting or moving objects (Figure 10). Nonetheless, such scenes are not suited for producing panoramas in the first place.

7. Discussion and Future Work

In this paper, we have shown how to generate sharp, high-quality panoramas from blury input videos. Our technique is based on joint global motion estimation and multi-frame deblurring. It also automatically computes the duty cycle of the video (i.e., percentage of time between frames that is actually exposure time). This is a strong step forward in increasing the ease with which users can capture panorama. Video cameras and video modes on still cameras and phones have become more prevalent, thus there will be increased need for users to generate panoramas from videos; however, many of these video devices have limited capture quality due to the need for lightweight, low cost components. As a result, dealing with blurred videos will also become quite necessary.

Our results suggest several areas for future work. While we have assumed the scene is static, a logical next step is to allow for some amount of local scene motion. In this case, we would down-weight areas that are not consistent with the global motion – this would also us to potentially remove moving objects or, alternatively, keep a moving object intact by reinserting a view of it at one selected time.

Another area of future work is to use our method for video stabilization. If moving objects are successfully handled, i.e., don't introduce artifacts, then we could resample
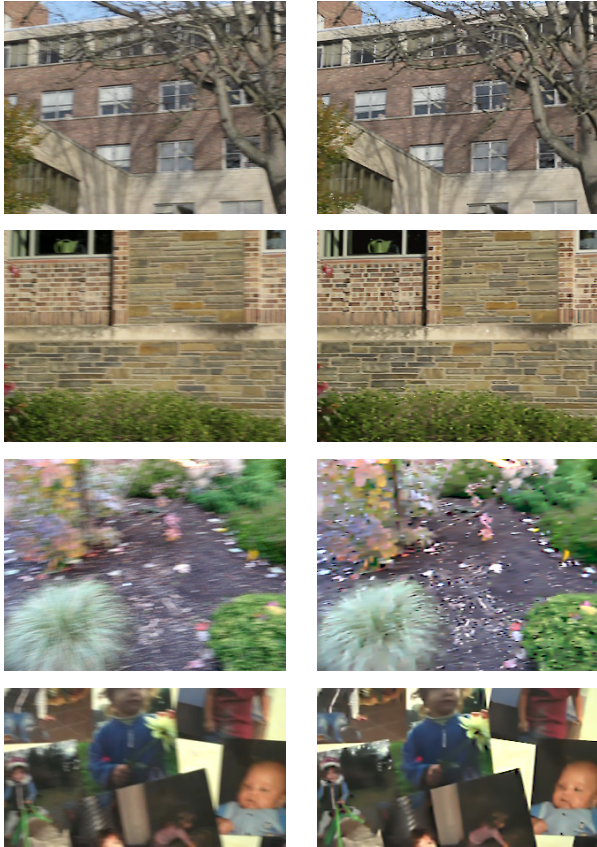
Figure 9. Sample results on real video sequences. Left column: frames of input video; right column: frames of deblurred output. Row 1 and 2: Canon SD1100; Row 3: Canon A540; Row 4: FlipHD.

Figure 10. Failure cases on real videos. Left image: high saturation and contrast. Right image: moving object, where the person near the low-left corner was crossing in front of the camera.

a smooth version of our recovered input path to generate a stabilized video, where the static areas are deblurred.

Lastly, there are situations where our method may fail to produce satisfactory result. Such scenarios usually involve high degrees of saturation and contrast, which are typically also difficult for all other deblurring methods. Many cameras perform some post-processing that may be dependent on the scene radiance distribution. While using the average CRF over a few benchmark scenes provides a good approximation in most cases, it can break down in harsh lighting

conditions. How to adaptively estimate the CRF in these situations is interesting direction for future work.

References

[1] `luxal.dachary.org/webhdr/cameras.shtml`.

[2] A. Agrawal, Y. Xu, and R. Raskar. Invertible motion blur in video. In *SIGGRAPH*, 2009.

[3] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Szeliski. A database and evaluation methodology for optical flow. In *ICCV*, 2007.

[4] B. Bascle, A. Blake, and A. Zisserman. Motion deblurring and super-resolution from an image sequence. In *ECCV*, 1996.

[5] S. Cho, Y. Matsushita, and S. Lee. Removing non-uniform motion blur from images. In *ICCV*, 2007.

[6] S. Dai and Y. Wu. Motion from blur. In *CVPR*, 2008.

[7] H. Jin, P. Favaro, and R. Cipolla. Visual tracking in the presence of motion blur. In *CVPR*, 2005.

[8] N. Joshi, R. Szeliski, and D. Kriegman. PSF estimation using sharp edge prediction. In *CVPR*, 2008.

[9] A. Levin. Blind motion deblurring using image statistics. In *NIPS*, 2006.

[10] A. Levin, R. Fergus, F. Durand, and W. T. Freeman. Image and depth from a conventional camera with a coded aperture. In *SIGGRAPH*, 2007.

[11] Y. Li and D. P. Huttenlocher. Sparse long-range random field and its application to image denoising. In *ECCV*, 2008.

[12] C. Liu, R. Szeliski, S. B. Kang, C. L. Zitnick, and W. T. Freeman. Automatic estimation and removal of noise from a single image. *PAMI*, 30(2):299–314, 2008.

[13] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI*, 1981.

[14] J. Nocedal. Updating quasi-newton matrices with limited storage. *Mathematics of Computation*, 35:773–782, 1980.

[15] M. Onogi and H. Saito. Mosaicing and restoration from blurred image sequence taken with moving camera. In *ICAPR*, 2005.

[16] A. Rav-Acha and S. Peleg. Two motion-blurred images are better than one. *Pattern Recognition Letters*, 26(3):311–317, 2005.

[17] Q. Shan, J. Jia, and A. Agarwala. High-quality motion deblurring from a single image. In *SIGGRAPH*, 2008.

[18] Y. Tai, P.Tan, L.Gao, and M. Brown. Richardson-lucy deblurring for scenes under projective motion path. Technical report, KAIST, 2009.

[19] Y. W. Tai, D. Hao, M. S. Brown, and S. Lin. Image/video deblurring using a hybrid camera. In *CVPR*, 2008.

[20] M. Tico and M. Vehvilainen. Robust image fusion for image stabilization. In *ICASSP*, 2007.

[21] L. Yuan, J. Sun, L. Quan, and H.-Y. Shum. Image deblurring with blurred/noisy image pairs. In *SIGGRAPH*, 2007.