Correspondence

Removing Motion Blur With Space–Time Processing

Hiroyuki Takeda, *Member, IEEE*, and Peyman Milanfar, *Fellow, IEEE*

***Abstract*—Although spatial deblurring is relatively well understood by assuming that the blur kernel is shift invariant, motion blur is not so when we attempt to deconvolve on a frame-by-frame basis: this is because, in general, videos include complex, multilayer transitions. Indeed, we face an exceedingly difficult problem in motion deblurring of a single frame when the scene contains motion occlusions. Instead of deblurring video frames individually, a fully 3-D deblurring method is proposed in this paper to reduce motion blur from a single motion-blurred video to produce a high-resolution video in both space and time. Unlike other existing approaches, the proposed deblurring kernel is free from knowledge of the local motions. Most importantly, due to its inherent locally adaptive nature, the 3-D deblurring is capable of automatically deblurring the portions of the sequence, which are motion blurred, without segmentation and without adversely affecting the rest of the spatiotemporal domain, where such blur is not present. Our method is a two-step approach; first we upscale the input video in space and time without explicit estimates of local motions, and then perform 3-D deblurring to obtain the restored sequence.**

***Index Terms*—Inverse filtering, sharpening and deblurring.**

I. Introduction

EARLIER in [1], we proposed a space–time data-adaptive video upscaling method, which does not require explicit subpixel estimates of motions. We named this method *3-D steering kernel regression* (3-D SKR). Unlike other video upscaling methods, e.g., [2], it is capable of finding an unknown pixel at an arbitrary position in not only space, but also time domains by filtering the neighboring pixels along the local 3-D orientations, which comprise spatial orientations and motion trajectories. After upscaling the input video, one usually performs a frame-by-frame deblurring process with a shift-invariant spatial (2-D) *point spread function* (PSF) in order to recover high-frequency components. However, typically, since any motion blurs present are often shift variant due to motion occlusions or nonuniform motions in the scene, they remain untreated, and hence, motion deblurring is a challenging problem. The main focus of this paper is to illustrate one important but so far unnoticed fact: any successful space–time interpolators enable us to remove the motion blur effects by deblurring with *shift-invariant* space–time (3-D) PSF and without any object segmentation or motion information. In this presentation, we use our 3-D SKR method as such a space–time interpolator.

Manuscript received August 13, 2010; revised December 28, 2010, March 03, 2011; accepted March 04, 2011. Date of publication March 24, 2011; date of current version September 16, 2011. This work was supported in part by the U.S. Air Force under Grant FA9550-07-1-0365 and the National Science Foundation under Grant CCF-1016018. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. James E. Fowler.

H. Takeda was with the Electrical Engineering Department, University of California, Santa Cruz, CA 95064 USA. He is now with the University of Michigan, Ann Arbor, MI 48109 USA (e-mail: htakeda@umich.edu).

P. Milanfar is with the Electrical Engineering Department, University of California, Santa Cruz, CA 95064 USA (e-mail: milanfar@soe.ucsc.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2011.2131666

The practical solutions to blind motion deblurring available so far largely only treat the case, where the blur is a result of global motions due to the camera displacements [3], [4], rather than motion of the objects in the scene. When the motion blur is not global, then it would seem that segmentation information is needed in order to identify what part of the image suffers from motion blur (typically due to fast-moving objects). Consequently, the problem of deblurring moving objects in the scene is quite complex because it requires 1) segmentation of moving objects from the background, 2) estimation of a spatial motion PSF for each moving object, 3) deconvolution of the moving objects one by one with the corresponding PSFs, and finally 4) putting the deblurred objects back together into a coherent and artifact-free image or sequence [5]–[8]. In order to perform the first two steps (segmentation and PSF estimation), one would need to carry out global/local motion estimation [9]–[12]. Thus, the deblurring performance strongly depends on the accuracy of motion estimation and segmentation of moving objects. However, the errors in both are in general unavoidable, particularly, in the presence of multiple motions, occlusion, or nonrigid motions, i.e., when there are any motions that violate parametric models or the standard optical flow brightness constancy constraint.

In this paper, we present a motion deblurring approach for videos that is free of both explicit motion estimation and segmentation. Briefly speaking, we point out and exploit what in hindsight seems obvious, though apparently not exploited so far in the literature: that motion blur is by nature a temporal blur, which is caused by relative displacements of the camera and the objects in the scene while the camera shutter is opened. Therefore, a temporal blur degradation model is more appropriate and physically meaningful for the general motion deblurring problem than the usual spatial blur model. An important advantage of the use of the temporal blur model is that regardless of whether the motion blur is global (camera induced) or local (object induced) in nature, the temporal PSF stays shift invariant,[1] whereas the spatial PSF must be considered shift variant in essentially all state-of-the-art frame-by-frame (or 2-D, spatial) motion deblurring approaches [5]–[8].

The examples in Figs. 1 and 2 illustrate the advantage of our space–time (3-D) approach as compared to the blind motion deblurring methods in the spatial domain proposed by Fergus *et al.* [3] and Shan *et al.* [4]. For the first example, the ground truth, a motion-blurred frame, and the restored images by Fergus' method, Shan's method, and our approach are shown in Fig. 1(a)–(e) including some detailed regions in Fig. 1(f)–(j), respectively. As can be seen from this example, their methods [3], [4] deblur the background, while, in fact, we wish to restore the details of the mug. This is because both blind methods are designed for the removal of the global blur effect caused by translational displacements of the camera (i.e., ego-motion). Segmentation of the moving objects is necessary to deblur segments one by one with different motion PSFs. On the other hand, the second example in Fig. 2 is a case, where spatial segmentation of motion regions is simply not practical. The pepper image shown in Fig. 2(b) is blurred by another type of motion, namely, the rotation of the camera. When the camera rotates about its optical axis while capturing an image, the middle portion of the image is less blurry than the outer regions because the pixels in the middle move relatively little. Similar to the previous example, the restored images by Fergus', Shan's, and our approaches are shown in Fig. 2(d) and (e). We will discuss this example in more

[1]We assume that the exposure time stays constant.

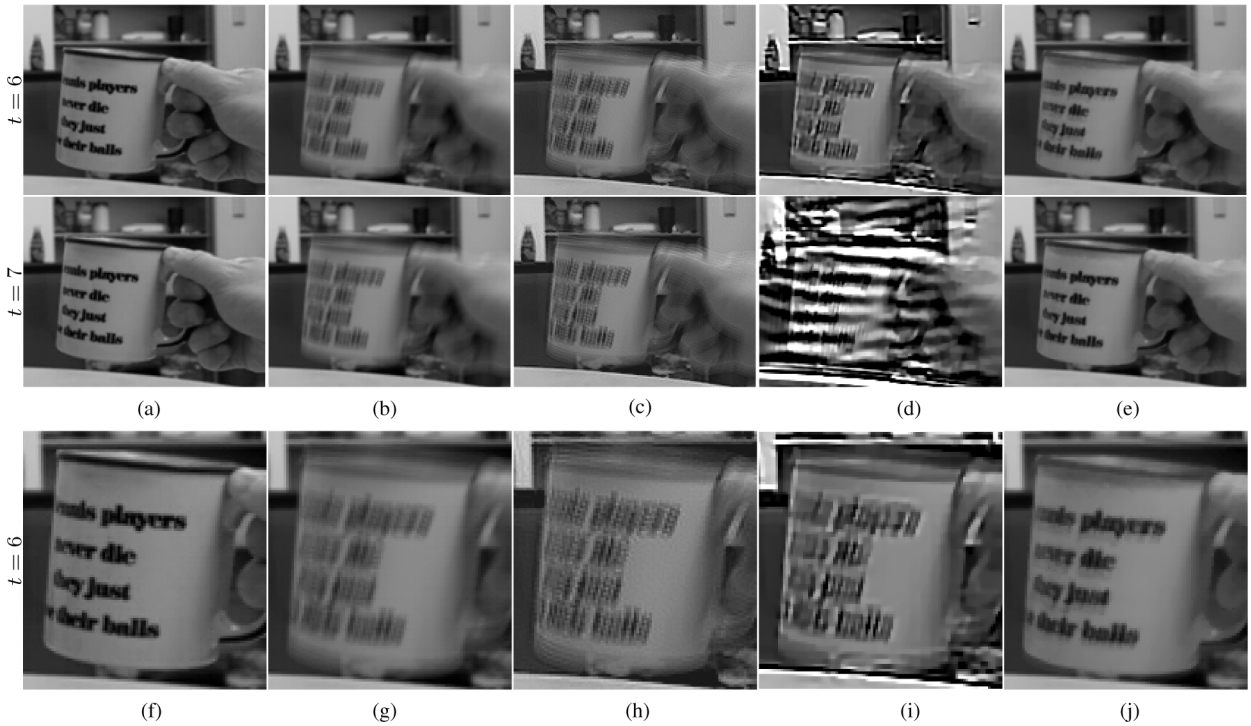

Fig. 1. Motion (temporal) deblurring example of the Cup sequence (130 × 165, 16 frames) in which a cup moves upward. (a) Two frames of the ground truth at times $t = 6$ to 7. (b) Blurred video frames generated by taking the average of five consecutive frames (the corresponding PSF is $1 \times 1 \times 5$ uniform) [PSNR: 23.76 dB (top), 23.68 dB (bottom), and structure similarity (SSIM): 0.76 (top), 0.75 (bottom)]. (c)–(e) Deblurred frames by Fergus's method [3] [PSNR: 22.58 dB (top), 22.44 dB (bottom), and SSIM: 0.69 (top), 0.68 (bottom)], Shan's method [4] [PSNR: 18.51 dB (top), 10.75 dB (bottom), and SSIM: 0.57 (top), 0.16 (bottom)], and the proposed 3-D total variation (TV) method (13) [PSNR: 32.57 dB (top), 31.55 dB (bottom), and SSIM: 0.98 (top), 0.97 (bottom)], respectively. The figures (f)–(j) are the selected regions of the video frames (a)–(e) at time $t = 6$, respectively. (a) Ground truth. (b) Blurred frames. (c) Fergus *et al.* [3]. (d) Shan *et al.* [4]. (e) Proposed method (13). (f) Ground truth. (g) Blurred frames. (h) Fergus *et al.*, [3]. (i) Shan *et al.* [4]. (j) Proposed method (13).

detail in Section III and a few more examples are also available at our website.[2] Although the blind methods are capable of estimating complex blur kernels, when the blur is spatially nonuniform, they no longer work. We briefly summarize some existing methods for the motion deblurring problem in the next section.

II. Motion Deblurring in 2-D and 3-D

A. Existing Methods

Ben-Ezra and Nayar [5], Tai *et al.* [6], and Cho *et al.* [7] proposed deblurring methods, where the spatial motion PSF is obtained from the estimated motions. Ben-Ezra and Nayar [5] and Tai *et al.* [6] used two different cameras: a low-speed high-resolution camera and a high-speed low-resolution camera, and capture two videos of the same scene at the same time. Then, they estimate motions using the high-speed low-resolution video so that detailed local motion trajectories can be estimated, and the estimated local motions yield a spatial motion PSF for each moving object. On the other hand, Cho *et al.* [7] took a pair of images by a camera with some time delay or by two cameras with no time delay but some spatial displacement. The image pair enables the separation of the moving objects and the foreground from the background. Each part of the images is often blurred with a different PSF. The separation is helpful in estimating the different PSFs individually, and the estimation process of the PSFs becomes more stable.

Whereas the deblurring methods in [5]–[7] obtain the spatial motion PSF based on the global/local motion information, Fergus *et al.* proposed a blind motion deblurring method using a relationship between the distribution of gradients and the degree of blur [3]. With this in hand, the method estimates a spatial motion PSF for each segmented object.

Later, inspired by Fergus' blind motion deblurring method, Levin [8] and Shan *et al.* [4] proposed blind deblurring methods for a single blurred image caused by a shaking camera. Although their methods are limited to global motion blur, using the relationship between the distribution of derivatives and the degree of blur proposed by Fergus *et al.*, they estimated a shift-invariant PSF without parametrization.

Ji and Liu [13] and Dai and Wu [14] also proposed derivative-based methods. Ji and Liu estimated the spatial motion PSF by a spectral analysis of the image gradients, and Dai and Wu obtained the PSF by studying how blurry the local edges are, as indicated by local gradients. Recently, another blind motion deblurring method was proposed by Chen *et al.* [15] for the reduction of global motion blur. They claimed that the PSF estimation is more stable with two images of the same scene degraded by different PSFs, and also used a robust estimation technique to stabilize the PSF estimation process further.

With the advancement of computational algorithms, as mentioned earlier, the data-acquisition process has been also studied. Using multiple cameras [5]–[7] is one simple way to make the identification of the underlying motion-blur kernel easier. Another technique called *coded exposure* improves the estimation of both blur kernels and images [16]. The idea of the coded exposure is to preserve some high-frequency components by repeatedly opening and closing the shutter while the camera is capturing a single image. Although it makes the SNR ratio worse, the high-frequency components are helpful in not only finding the blur kernel, but also estimating the underlying image with higher quality. When the blur is spatially variant, then scene segmentation is necessary [17].

[2]http://users.soe.ucsc.edu/~htakeda/VideoDeblurring/VideoDeblurring.htm

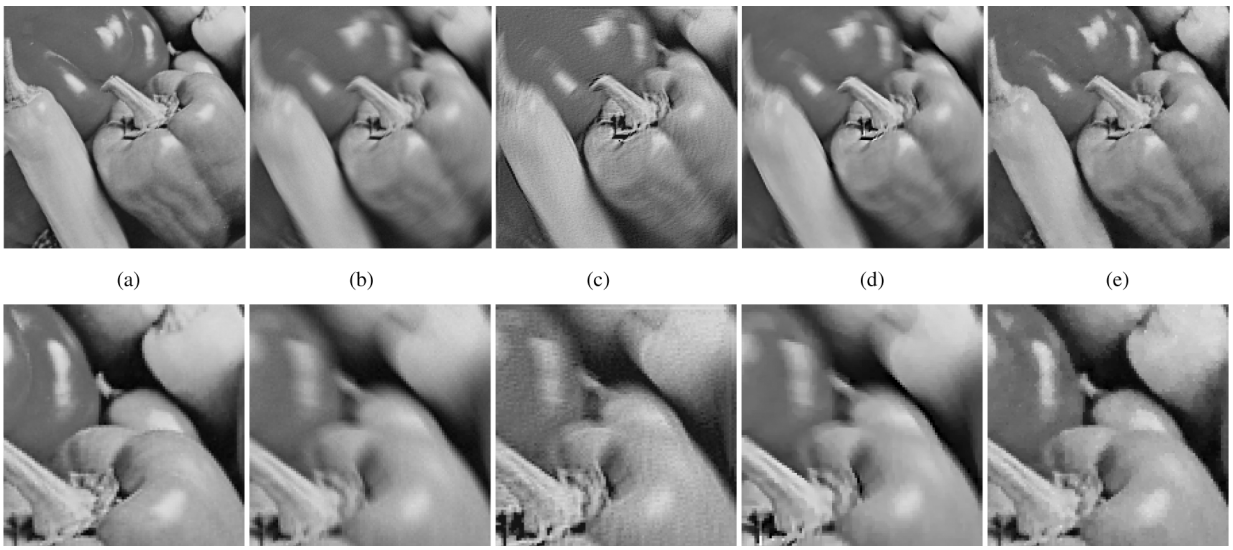

Fig. 2. Motion deblurring example of a rotating pepper sequence (179×179, 90 frames). (a) One of the frames from a simulated sequence, which we generate by rotating the pepper image counterclockwise 1° per frame. (b) Blurred frame generated by taking the average of eight consecutive frames (the corresponding PSF is a $1 \times 1 \times 8$ shift-invariant uniform PSF) and adding white Gaussian noise with standard deviation $\sigma = 2$ ($\mathrm{PSNR} = 27.10$ dB, $\mathrm{SSIM} = 0.82$). (**c**) and (d) Deblurred frames by Fergus' method [3] ($\mathrm{PSNR} = 23.23$ dB, $\mathrm{SSIM} = 0.61$), and Shan's method [4] ($\mathrm{PSNR} = 25.12$ dB, $\mathrm{SSIM} = 0.81$), respectively. (**e**) Deblurred frame by the proposed method ($\mathrm{PSNR} = 33.12$ dB, $\mathrm{SSIM} = 0.90$). The images in the second column show the magnifications of the upper right portions of the images in the first column. (a) Ground truth. (b) Blurred frame. (c) Fergus *et al.* [3]. (d) Shan *et al.* [4]. (e) Proposed method (13).

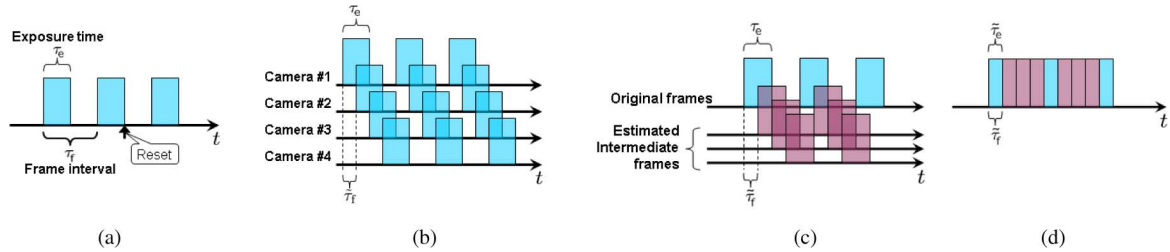

Fig. 3. Schematic representation of the exposure time τ_e and the frame interval τ_f. (a) Standard camera. (b) Multiple videos taken by multiple cameras with slight time delay is fused to produce a high frame rate video. (c) Original frames with estimated intermediate frames, Frame rate upconversion. (d) Temporally deblurred output frames.

B. Path Ahead

All the methods mentioned earlier are similar in that they aim at removing motion blur by spatial (2-D) processing. In the presence of multiple motions, the existing methods would have to estimate shift-variant PSF and segment the blurred images by local motions (or depth maps). However, occlusions make the deblurring problem more difficult because pixel values around motion occlusions are a mixture of multiple objects moving in independent directions. In this paper, we reduce the motion blur effect from videos by introducing the space–time (3-D) deblurring model. Since the data model is more reflective of the actual data-acquisition process, even in the presence of motion occlusions, deblurring with 3-D blur kernel can effectively remove both global and local motion blur without segmentation or reliance on explicit motion information.

Practically speaking, for videos, it is not always preferable to remove all the motion blur effect from video frames. Particularly, for videos with relatively low frame rate (e.g., 10–20 frames per second), in order to show smooth trajectory of moving objects, motion blur (temporal blur) is often intentionally added. Thus, when removing (or more precisely "reducing") the motion blur from videos, we would need to increase the temporal resolution of the video. This operation can be thought of as the familiar frame rate up-conversion, with the following caveat: in our context, the intermediate frames are not the end results of interest, but as we will explain shortly, rather a means to obtain a deblurred sequence, at possibly the original frame rate. It is worth noting that the temporal blur reduction is equivalent to shortening the exposure time of video frames. Typically, the exposure time τ_e is less than the time interval between the frames τ_f (i.e., $\tau_e \leq \tau_f$), as shown in Fig. 3(a). Many commercial cameras set τ_e to less than $0.5\tau_f$ (see for instance [18]). Borissoff in [18] pointed out that τ_e should ideally depend on the speed of moving objects. Specifically, the exposure time should be half of the time it takes for a moving object to run through the scene width, or else temporal aliasing would be visible. In [19], Shechtman *et al.* presented a space–time super resolution (SR) algorithm, where *multiple* cameras capture the same scene at once with slight spatial and temporal displacements. Then, multiple low-resolution videos in space and time are fused to obtain a spatiotemporally super-resolved sequence. As a postprocessing step, they spatiotemporally deblur the super-resolved video so that the exposure time τ_e nearly equals to the frame interval τ_f. Recently, Agrawal *et al.* proposed a temporal coded sampling technique for temporal video SR in [20], where multiple cameras simultaneously capture the same scene with different frame rates, exposure times, and temporal sampling positions. Their proposed method carefully optimizes those frame sampling conditions so that the space–time SR can achieve higher quality results. By contrast, in this paper, we demonstrate that the problem of motion blur restoration can be solved using a *single*, possibly low frame rate, video sequence.

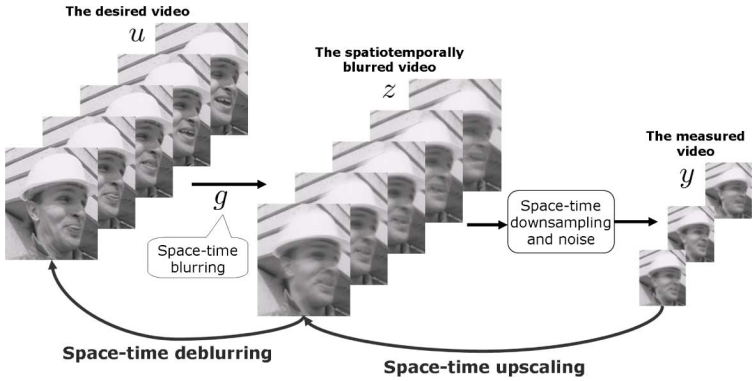

Fig. 4. Forward model addressed in this paper. We estimate the desired video $\mathbf{u}$ by two-step approach: 1) space–time upscaling, and 2) space–time deblurring.

To summarize, frame-rate up-conversion is necessary in order to avoid temporal aliasing. Furthermore, unlike motion deblurring algorithms which address the problem purely in the spatial domain [3]–[7], [13]–[15], we deblur with a shift-invariant 3-D PSF, which is effective for any type of motion blur. Examples were illustrated in Figs. 1 and 2, and more will be shown later in Section III. The following are the assumptions and the limitations of our 3-D deblurring approach.

Assumptions

1) *The camera settings are fixed*:
The aperture size, the focus length, the exposure time, and the frame interval are all fixed. The photosensitivity of the image sensor array is uniform and unchanged.
2) *One camera captures one frame at a time*:
In our approach, only one video is available, and the video is shot by a single camera, which captures one frame at a time. Also, all the pixels of one frame are sampled at the same time (without time delay).
3) *The aperture size is small*:
We currently assume that the aperture size is so small that the out-of-focus blur is almost homogeneous.
4) *The spatial and temporal PSFs are known*:
In the current presentation, our primary focus is to show that a simple deblurring with the space–time (3-D) shift-invariant PSF can effectively reduce the complicated, nonuniform motion blur effects of a sequence of images.

Limitations

1) *The performance of our motion deblurring depends on the performance of the space–time interpolator*:
The space–time interpolator needs to generate the missing intermediate *blurry* frames, while preserving spatial and temporal blur effects.
2) *The temporal upscaling factor affects our motion deblurring:*
To remove the motion blur completely, the temporal upscaling factor of the space–time interpolator must be set to so large that the motion speed slows down to less than 1 pixel per frame. For instance, when the temporal upscaling factor is not large enough and an object in the upscaled video moves 3 pixels per frame, the moving object would be still blurry along its motion trajectory in a 3-pixel-wide window even after we deblur. However, as discussed in this section, the motion blur is sometimes necessary for very fast moving objects in order to preserve a smooth motion trajectory.

C. Video Deblurring in 3-D

Next, we extend the single image (2-D) deblurring technique with total variation (TV) regularization to space–time (3-D) motion deblurring for videos. Ringing suppression is of importance because the ringing effect in time creates significant visual distortion for the output videos.

1) Data Model: The exposure time τ_e of videos taken with a standard camera is always shorter than the frame interval τ_f, as illustrated in Fig. 3(a). It is generally not possible to reduce motion blur by temporal deblurring when $\tau_e < \tau_f$ (i.e., the temporal support of the PSF is shorter than the frame interval τ_f). This is because the standard camera captures one frame at a time. The camera reads a frame out of the photosensitive array, and the array is reset to capture the next frame.[3] Unlike the spatial sampling rate, the temporal sampling rate is always below the Nyquist rate. This is an electromechanical limitation of the standard video camera. One way to have a high-speed video with $\tau_e > \tau_f$ is to fuse multiple videos captured by multiple cameras at the same time with slight time delay, as shown in Fig. 4(b). As we mentioned earlier, the technique is referred to as *space–time SR* [19] or *high-speed videography* [21]. After the fusion of multiple videos into a high-speed video, the frame interval becomes shorter than the exposure time and we can carry out the temporal deblurring to reduce the motion blur effect.

An alternative to using multiple cameras is to generate intermediate frames, which may be obtained by frame interpolation (e.g., [22] and [1]), so that the new frame interval $\tilde{\tau}_f$ is now smaller than τ_e, as illustrated in Fig. 3(c). Once we have the video sequence with $\tau_e > \tilde{\tau}_f$, the temporal deblurring reduces τ_e to be nearly equally to $\tilde{\tau}_f$, and the video shown in Fig. 3(d) is our desired output. It is worth noting that, in the most general setting, generation/interpolation of temporally intermediate frames is indeed a very challenging problem. However, since our interest lies mainly in the removal of motion blur, the temporal interpolation problem is not quite as complex as the general setting. In the most general case, the space–time SR method [19] employing multiple cameras may be the only practical solution. Of course, it is possible to apply the frame interpolation for the space–time super-resolved video to generate an even higher speed video. However, in this paper, we focus on the case, where only a single video is available and show that our frame interpolation method (3-D SKR [1]) enables motion deblurring. We note that the performance of the motion deblurring, therefore, depends on how well we interpolate intermediate frames. As long as the interpolator successfully generates intermediate (upscaled) frames, the 3-D deblurring can reduce the motion blur effects. Since, typically, the exposure time of the frames is relatively short even at low frame rate (10–20 frame per second), we assume that local motion trajectories between frames are smooth enough that the 3-D SKR method interpolates

[3]Most commercial charge-coupled device (CCD) cameras nowadays use the interline CCD technique, where the charged electrons of the frame are first transferred from the photosensitive sensor array to the temporal storage array and the photosensitive array is reset. Then, the camera reads the frame out of the temporal storage array while the photosensitive array is capturing the next frame.

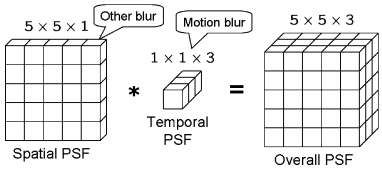

Fig. 5. Overall PSF kernel in video (3-D) is given by the convolution of the spatial and temporal PSF kernels.

the trajectories. When multilayered large (fast) motions are present, it is hard to generate intermediate frames using only a single video input due to severe occlusions. Consequently, a video with higher frame rate is necessary.

Fig. 4 illustrates an idealized forward model, which we adopt in this paper. Specifically, the camera captures the first frame by temporally integrating the first few frames (say the first, second, and third frames) of the desired video $\mathbf{u}$, and the second frame by integrating, for example, the fifth frame and the following two frames.[4] Next, the frames are spatially downsampled due to the limited number of pixels on the image sensor. We can regard spatial and temporal sampling mechanisms of the camera altogether as space–time downsampling effect, as shown in Fig. 4.

In our paper, we assume that all the frames in a video are taken by a camera with the same setting (focus, zoom, aperture size, exposure time, frame rate, etc.). Under such conditions, the spatial PSF, caused by the physical size of one pixel on the image sensor, and the temporal PSF, whose support size is given by the exposure time, also remain unchanged, no matter how the camera moves and no matter what scene we shoot. Therefore, the 3-D PSF, given by the convolution of the 2-D spatial PSF and the 1-D temporal PSF, as depicted in Fig. 5, is shift invariant.

Under these assumptions, we estimate the desired output u by a two-step approach: 1) space–time upscaling, and 2) space–time deblurring. In our earlier study, we proposed a space–time upscaling method in [1], where we left the motion (temporal) blur effect untreated, and removed only the spatial blur with a shift-invariant (2-D) PSF with TV regularization. In this paper, we study the reduction of the spatial *and* temporal blur effects simultaneously with a shift-invariant (3-D) PSF. A 3-D PSF is effective because the spatial blur and the temporal blur (frame accumulation) are both shift invariant. PSF becomes shift variant when we convert the 3-D PSF into 2-D temporal slices, which yield the spatial PSF due to the moving objects for frame-by-frame deblurring. Again, unlike the existing methods [3]–[7], [13]–[15], after the space–time upscaling, no motion estimation or scene segmentation is required for the space–time deblurring.

Having graphically introduced our data model in Fig. 4, we define the mathematical model between the blurred data denoted y and the desired signal u with a 3-D PSF g as follows:

$$y(\mathbf{x}) = z(\mathbf{x}) + \varepsilon = (g * u)(\mathbf{x}) + \varepsilon \tag{1}$$

where ε is the independent and identically distributed zero mean noise value (with otherwise no particular statistical distribution assumed), $\mathbf{x} = [x_1, x_2, t]$ is the 3-D (space–time) coordinate in vector form, $*$ is the convolution operator, and g is the combination of spatial blur g_{s} and the temporal blur g_τ

$$g(\mathbf{x}) = g_{\mathrm{s}}(x_1, x_2) * g_\tau(t). \tag{2}$$

[4]Perhaps, a more concise description is that motion blur effect can always be modeled as a single 1-D shift-invariant PSF in the direction of the time axis. This is simply because the blur results from multiple exposure of the same fast-moving object in space during the exposure time. The skips of the temporal sampling positions can be regarded as temporal downsampling.

If the sizes of the spatial and temporal PSF kernels are $N \times N \times 1$ and $1 \times 1 \times \tau$, respectively, then the overall PSF kernel has size $N \times N \times \tau$, as illustrated in Fig. 5. We will discuss how to select the 3-D PSF for deblurring later in Section II-C. While the data model (1) resembles the one introduced by Irani and Peleg [23], we note that ours is a 3-D data model. More specifically, we consider an image sequence (a video) as one data set and consider the case where only a single video is available. The PSF and the downsampling operations are also all in 3-D.

In this paper, we split the data model (1) into

$$\mathrm{Spatiotemporal\,(3-D)\,upsampling\,problem:}$$
$$y_i = z(\mathbf{x}_i) + \varepsilon_i \tag{3}$$
$$\mathrm{Spatiotemporal\,(3-D)\,deblurring\,problem:}$$
$$z(\mathbf{x}_j) = (g * u)(\mathbf{x}_j) \tag{4}$$

where $\mathbf{x}_i = [x_{1i}, x_{2i}, t_i]^T$ is the pixel sampling position of the low-resolution video with index i, $\mathbf{x}_j = [x_{1j}, x_{2j}, t_j]^T$ is the pixel sampling position of the high-resolution video with index j, and y_i is the ith sample of the low-resolution video ($y_i = y(\mathbf{x}_i)$). We estimate $u(\mathbf{x}_j)$ for all j by a two-step approach:

Step 1. upscaling of y_i to have the motion-blurred high-resolution video $z(\mathbf{x}_j)$;

Step 2. deblurring of $z(\mathbf{x}_j)$ to have the motion-deblurred high-resolution video $u(\mathbf{x}_j)$.

For the upscaling problem, we first upsample the low-resolution video (y_i) and register it onto the grid of the desired high-resolution video, as illustrated in Fig. 6. Since the sampling density of the low-resolution video $(\mathbf{x}_i)$ is lower than the density of the high-resolution video $(\mathbf{x}_j)$, there are missing pixels. In Fig. 6, the blank pixel lattice indicates that a pixel value is missing, and we need to fill those missing pixels. We use our 3-D SKR [1] (reviewed in Section II-C2) to estimate the missing pixels. For the deblurring problem, since each blurry pixel $(z(\mathbf{x}_j))$ is coupled with its space–time neighbors due to the space–time blurring operation, it is preferable that we rewrite the data model (4) in matrix form as follows:

$$\mathrm{Spatiotemporal\,(3D)\,deblurring\,problem}: \underline{\mathbf{z}} = \mathbf{G}\underline{\mathbf{u}} \tag{5}$$

where $\underline{\mathbf{z}} = [\ldots, z(\mathbf{x}_j), \ldots]^T$ and $\underline{\mathbf{u}} = [\ldots, u(\mathbf{x}_j), \ldots]^T$. For example, let us say that the low-resolution video (y_i) is of size $(L/r_s) \times (M/r_s)$ and (T/r_t) frames, where r_s and r_t are the spatial and temporal upsampling factors, respectively. Then, the blurred version of the high-resolution video $\mathbf{z}$, which is available after the space–time upscaling, and the video of interest $\mathbf{u}$ are of size $L \times M \times T$, and the blurring operator $\mathbf{G}$ is of dimension $LTM \times LTM$. The matrices with underscore represent that they are lexicographically ordered into column-stacked vector form (e.g., $\underline{\mathbf{z}} \in \mathcal{R}^{LMT \times 1}$). Using (5), we present our 3-D deblurring in Section II-C3. But first, we describe the upscaling method.

2) Space–Time (3-D) Upscaling: The first step of our two-step approach is upscaling. Given the spatial and temporal upsampling factors r_s and r_t, we spatiotemporally upsample the low-resolution video and then register all the pixels (y_i) of the low-resolution video onto a grid of the high-resolution video, where the pixel positions in the high-resolution grid are labeled by $\mathbf{x}_j$, as illustrated in Fig. 6. Due to the lower sampling density of the low-resolution video, there are missing pixels in the high-resolution grid, and our task is to estimate the samples $z(\mathbf{x}_j)$ for all j from the measured samples y_i for $i = 1, \ldots, (LMT/r_s^2 r_t)$. Assuming that the underlying blurred function $z(\mathbf{x})$ is locally smooth and it is N-times differentiable, we can write the relationship between

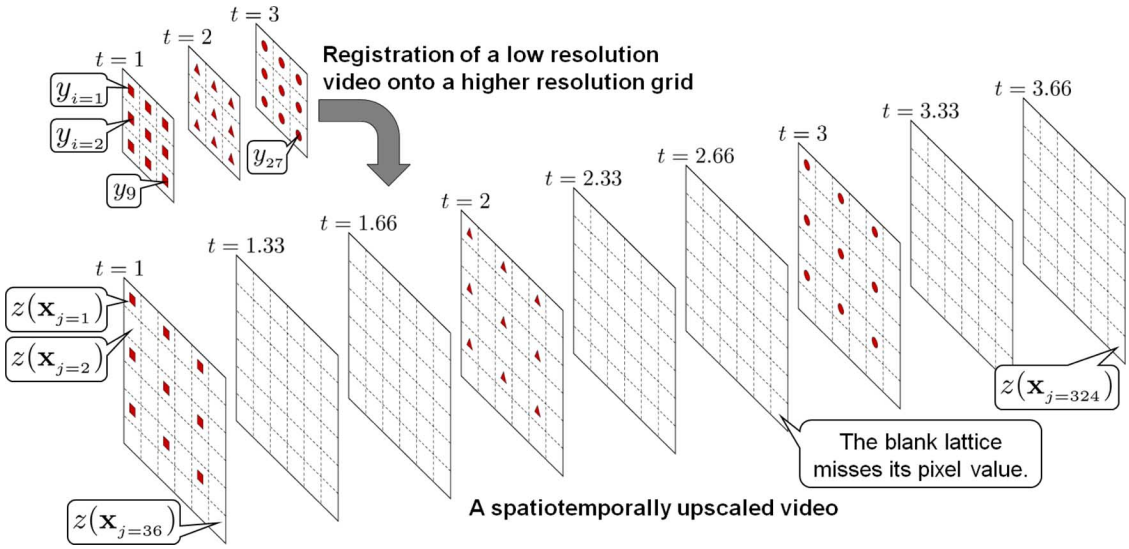

Fig. 6. Schematic representation of the registration of the low-resolution video onto a high-resolution grid. In the illustration, a low-resolution video (3 × 3, 3 frames) is upsampled with the spatial upsampling factor $r_s = 2$ and the temporal upsampling factor $r_t = 3$.

the unknown pixel value $z(\mathbf{x}_j)$ and its neighboring sample y_i by Taylor series as follows:

$$\begin{aligned} y_i &= z(\mathbf{x}_i) + \varepsilon_i \\ &= z(\mathbf{x}_j) + \{\nabla z(\mathbf{x}_i)\}^T(\mathbf{x}_i - \mathbf{x}_j) \\ &\quad + (\mathbf{x}_i - \mathbf{x}_j)^T\{\mathcal{H}z(\mathbf{x}_j)\}(\mathbf{x}_i - \mathbf{x}_j) + \cdots + \varepsilon_i \\ &= \beta_0 + \beta_1(\mathbf{x}_i - \mathbf{x}_j) \\ &\quad + \beta_2^T \mathrm{vech}\{(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T\} + \cdots + \varepsilon_i \end{aligned} \tag{6}$$

where ∇ and $\mathcal{H}$ are the gradient (3 × 1) and Hessian (3 × 3) operators, respectively, and $\mathrm{vech}\{\cdot\}$ is the half-vectorization operator that lexicographically orders the lower triangular potion of a symmetric matrix into a column-stacked vector. Furthermore, β_0 is $z(\mathbf{x}_j)$, which is the signal (or pixel) value of interest, and the vectors β_1 and β_2 are

$$\begin{aligned} \beta_1 &= \left[\frac{\partial z(\mathbf{x})}{\partial x_1}, \quad \frac{\partial z(\mathbf{x})}{\partial x_2}, \quad \frac{\partial z(\mathbf{x})}{\partial t}\right]^T\Bigg|_{\mathbf{x}=\mathbf{x}_j} \\ \beta_2 &= \frac{1}{2}\left[\frac{\partial^2 z(\mathbf{x})}{\partial x_1^2}, \quad 2\frac{\partial^2 z(\mathbf{x})}{\partial x_1 \partial x_2}, \quad 2\frac{\partial^2 z(\mathbf{x})}{\partial x_1 \partial t}\right. \\ &\quad \left.\frac{\partial^2 z(\mathbf{x})}{\partial x_2^2}, \quad 2\frac{\partial^2 z(\mathbf{x})}{\partial x_2 \partial t}, \quad \frac{\partial^2 z(\mathbf{x})}{\partial t^2}\right]^T\Bigg|_{\mathbf{x}=\mathbf{x}_j}. \end{aligned} \tag{7}$$

Since this approach is based on *local* signal representations, a logical step to take is to estimate the parameters $\{\beta_n\}_{n=0}^N$ using the neighboring samples (y_i) in a local analysis cubicle ω_j around the position of interest $\mathbf{x}_j$ while giving the nearby samples higher weights than samples farther away. A weighted least-square formulation of the fitting problem capturing this idea is

$$\begin{aligned} \min_{\{\beta_n\}_{n=0}^N} \sum_{i\in\omega_j} \Big[& y_i - \beta_0 - \beta_1^T(\mathbf{x}_i - \mathbf{x}_j) \\ & - \beta_2^T \mathrm{vech}\{(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T\} - \cdots \Big]^2 K(\mathbf{x}_i - \mathbf{x}_j) \end{aligned} \tag{8}$$

with the Gaussian kernel (weight) function

$$K(\mathbf{x}_i - \mathbf{x}_j) = \sqrt{|\mathbf{C}_i|}\exp\left\{-\frac{(\mathbf{x}_i - \mathbf{x}_j)^T\mathbf{C}_i(\mathbf{x}_i - \mathbf{x}_j)}{2h^2}\right\} \tag{9}$$

where h is the global smoothing parameter. This is the formulation of the kernel regression [24] in 3-D. We set $h = 0.7$ for all the experiments, and $\mathbf{C}_i$ is the smoothing (3 × 3) matrix for the sample y_i, which dictates the "footprint" of the kernel function and we will explain how we obtain it shortly. The minimization (8) yields a pointwise estimator of the blurry signal $z(\mathbf{x}_j)$ with the order of local signal representation (N)

$$\hat{z}(\mathbf{x}_j) = \hat{\beta}_0 = \sum_{i\in\omega_j} W_i(K(\mathbf{x}_i - \mathbf{x}_j), N)y_i \tag{10}$$

where W_i is the weights given by the choice of $\mathbf{C}_i$ and N. For example, choosing $N = 0$ (i.e., we keep only β_0 in (8) and ignore all the higher order terms), the estimator (10) becomes

$$\hat{z}(\mathbf{x}_j) = \sum_i \frac{K(\mathbf{x}_i - \mathbf{x}_j)\, y_i}{\sum_i K(\mathbf{x}_i - \mathbf{x}_j)}. \tag{11}$$

We set $N = 2$ as in [24] and the size of the cubicle ω_i is 5 × 5 × 5 in the grid of the low-resolution video in this paper. Since the pixel value of interest $z(\mathbf{x}_j)$ is a local combination of the neighboring samples, the performance of the estimator strongly depends on the choice of the kernel function, or more specifically the choice of the smoothing matrix $\mathbf{C}_i$. In our previous study [1], we obtain $\mathbf{C}_i$ from the local gradient vectors in a local analysis cubicle ξ_i, whose center is located at the position of y_i

$$\mathbf{C}_i = \mathbf{J}_i^T\mathbf{J}_i$$

and

$$\mathbf{J}_i = \begin{bmatrix} \vdots & \vdots & \vdots \\ z_{x_1}(\mathbf{x}_p) & z_{x_2}(\mathbf{x}_p) & z_t(\mathbf{x}_p) \\ \vdots & \vdots & \vdots \end{bmatrix}, \qquad p \in \xi_i \tag{12}$$

where p is the index of the sample positions around the ith sample (y_i) in the local analysis cubicle ξ_i, $z_{x_1}(\mathbf{x}_j)$, $z_{x_2}(\mathbf{x}_j)$, and $z_t(\mathbf{x}_j)$ are the gradients along the vertical (x_1), horizontal (x_2), and time (t) axes, respectively. In this paper, we first estimate the gradients

$(\beta_1 = [z_{x_1}(\mathbf{x}_p), z_{x_2}(\mathbf{x}_p), z_t(\mathbf{x}_p)])$ using (8) with $\mathbf{C}_i = \mathbf{I}$ and set ξ_i a $5 \times 5 \times 5$ cubicle in the grid of the low-resolution video y, and then, plugging in the estimated gradients into (12), we obtain the locally adaptive smoothing matrix $\mathbf{C}_i$ for each y_i. With $\mathbf{C}_i$ given by (12), the kernel function faithfully reflects the local signal structure in space–time (we call it the *steering* kernel function), i.e., when we estimate a pixel on an edge, the kernel function gives larger weights for the samples (y_i) located on the same edge. On the other hand, if there is no local structure, all the nearby samples have similar weights. Hence, the estimator (10) preserves local object structures while suppressing the noise effects in flat regions. We refer the interested reader to [24] for further details. Once all the pixels of interest have been estimated using (10), we fill them in the matrix $\mathbf{z}$ (5) and deblur the resulting 3-D data set at once, as explained in the following section.

3) Space–Time (3-D) Deblurring: Assuming that, at the space–time upscaling stage, noise is effectively suppressed [1], the important issue that we need to carefully treat in the deblurring stage is the suppression of the ringing artifacts, particularly, across time. The ringing effect in time may cause undesirable flicker when we play the output video. Therefore, the deblurring approach should smooth the output pixel across not only space, but also time. To this end, using the data model (5), we propose a 3-D deblurring method with the 3-D version of TV to recover the pixels across space and time

$$\hat{\underline{\mathbf{u}}} = \arg\min_{\underline{\mathbf{u}}} \left\{ \|\underline{\mathbf{z}} - \mathbf{G}\underline{\mathbf{u}}\|_2^2 + \lambda \|\mathbf{\Gamma}\underline{\mathbf{u}}\|_1 \right\} \tag{13}$$

where λ is the regularization parameter, and $\mathbf{\Gamma}$ is a high-pass filter. The joint use of L_2-, L_1-norms is fairly standard [25]–[27], where the first term (L_2-norm) is used to enforce the fidelity of the reconstruction to the data (in a mean-squared sense), and the second term (L_1-norm) is used to promote sparsity in the gradient domain, leading to sharp edges in space and time and avoid ringing artifacts. Specifically, we implement the TV regularization as follows:

$$\|\mathbf{\Gamma}\underline{\mathbf{u}}\|_1 \Rightarrow \sum_{l=-1}^{1}\sum_{m=-1}^{1}\sum_{t=-1}^{1} \left\| \underline{\mathbf{u}} - \mathbf{S}_{x_1}^{l}\mathbf{S}_{x_2}^{m}\mathbf{S}_{t}^{t}\underline{\mathbf{u}} \right\|_1 \tag{14}$$

where $\mathbf{S}_{x_1}^{l}$, $\mathbf{S}_{x_2}^{m}$, and $\mathbf{S}_{t}^{t}$ are the shift operators that shift the video $\underline{\mathbf{u}}$ toward x_1, x_2, and t-directions with l, m, and t-pixels, respectively. We iteratively minimize the cost $C(\underline{\mathbf{u}}) = \|\underline{\mathbf{z}} - \mathbf{G}\underline{\mathbf{u}}\|_2^2 + \lambda\|\mathbf{\Gamma}\underline{\mathbf{u}}\|_1$ in (13) with (14) to find the deblurred sequence $\hat{\underline{\mathbf{u}}}$ using the steepest descent method

$$\hat{\underline{\mathbf{u}}}^{(\ell+1)} = \hat{\underline{\mathbf{u}}}^{(\ell)} + \mu \left. \frac{\partial C(\underline{\mathbf{u}})}{\partial \underline{\mathbf{u}}} \right|_{\underline{\mathbf{u}}=\hat{\underline{\mathbf{u}}}^{(\ell)}} \tag{15}$$

where μ is the step size, and

$$\begin{aligned} \frac{\partial C(\underline{\mathbf{u}})}{\partial \underline{\mathbf{u}}} &= -\mathbf{G}^T(\underline{\mathbf{z}} - \mathbf{G}\underline{\mathbf{u}}) \\ &\quad + \lambda \sum_{l=-1}^{1}\sum_{m=-1}^{1}\sum_{t=-1}^{1} \left(\mathbf{I} - \mathbf{S}_{x_1}^{-l}\mathbf{S}_{x_2}^{-m}\mathbf{S}_{t}^{-t} \right) \\ &\quad \times \operatorname{sign}\left(\underline{\mathbf{u}} - \mathbf{S}_{x_1}^{l}\mathbf{S}_{x_2}^{m}\mathbf{S}_{t}^{t}\underline{\mathbf{u}} \right). \end{aligned} \tag{16}$$

We initialize $\hat{\underline{\mathbf{u}}}^{(\ell)}$ with the output of the space–time upscaling (i.e., $\hat{\underline{\mathbf{u}}}^{(0)} = \underline{\mathbf{z}}$), and manually select a reasonable 3-D PSF $(\mathbf{G})$ for the experiments with real blurry sequences.

In this paper, we select a 3-D PSF based on the exposure time τ_e and the frame interval τ_f of the input videos (which are generally available from the camera setting), and the user-defined spatial and temporal upscaling factors r_s and r_t. Specifically, we select the spatial PSF an $r_s \times r_s$ uniform PSF. Currently, we ignore the out-of-focus blur, and we obtain the temporal support size τ of the temporal PSF by

$$\tau = \frac{\tau_e}{\tau_f} \times r_t \tag{17}$$

where r_t is the user-defined temporal upscaling factor. Convolving the spatial PSF and the temporal PSF as shown in Fig. 5, we have a 3-D $(r_s \times r_s \times \tau)$ PSF for the deblurring (13). Our deblurring method with the $r_s \times r_s \times \tau$ PSF reduces the effective exposure time of the upscaled video. Specifically, after the deblurring, the effective exposure time of the output video is given by

$$\tilde{\tau}_e = \frac{\tau_e}{\tau} = \frac{\tau_f}{r_t}. \tag{18}$$

Therefore, when the temporal upscaling factor r_t is not high, the exposure time $\tilde{\tau}_e$ is not shortened by very much, and some motion blur effects may be seen in the output video. For example, if an object moves 3 pixels per frame in the spatiotemporally upscaled video, the moving object would be still blurry along its motion trajectory in a 3-pixel-wide window even after we deblur.

III. Experiments

We illustrate the performance of our proposed technique on both real and simulated sequences. To begin, we first illustrate motion deblurring performance on the Cup sequence, with simulated motion blur.[5] The Cup example is the one we briefly showed Section I. This sequence contains relatively simple transitions, i.e., the cup moves upward. Fig. 1(a) shows the ground-truth frames, and Fig. 1(b) shows the motion-blurred frames generated by taking the average of five consecutive frames, i.e., the corresponding PSF in 3-D is $1 \times 1 \times 5$ uniform. The deblurred images of the Cup sequence by Fergus' method [3], Shan's method[6] [4], and our approach (13) with $(\mu, \lambda) = (0.75, 0.04)$ are shown in Fig. 1(c)–(e), respectively. Fig. 1(f)–(j) shows the selected regions of the video frames Fig. 1(a)–(e) at time $t = 6$, respectively. The corresponding PSNR[7] and SSIM[8] values are indicated in the figure captions. It is worth noting here again that, although motion occlusions are present in the sequence, the proposed 3-D deblurring requires neither segmentation nor motion estimation. We also note that, in a sense, one could regard a $1 \times 1 \times \tau$ PSF as a 1-D PSF. However, in our paper, a $1 \times N \times 1$ PSF and a $1 \times 1 \times N$ are, for example, completely different. The $1 \times N \times 1$ PSF blurs along the horizontal (x_2) axis, while on the other hand, the $1 \times 1 \times N$ PSF blurs along the time axis.

The second example in Fig. 2 is also a simulated motion deblurring. In this example, the motion blur is caused by the camera rotation about its optical axis. We generated a video by rotating the pepper image counterclockwise 1° per frame for 90 frames. This is equivalent to rotating the camera clockwise 1° per frame. The sequence of the rotated pepper image is the ground-truth video in this example. Then, we blurred the video by blurring with a $1 \times 1 \times 8$ uniform PSF (this is equivalent to taking the average of eight consecutive frames), and added white Gaussian noise (standard deviation $= 2$). Fig. 2(a) and (b) shows one frame from the ground-truth video and the noisy blurred video. When the camera rotates, the pixels rotate at different speeds in proportion to the distance from the center of the rotation. Consequently,

[5]In order to examine how well the motion blur will be removed, we do not take the spatial blur into account for the experiments.

[6]The software is available at http://w1.cse.cuhk.edu.hk/~leojia/programs/deblurring/deblurring.htm. We set the parameter "`noiseStr`" to 0.05 and used the default setting for the other parameters for all the examples.

[7]PSNR ratio $= 10\log_{10}(255^2/\text{mean square error})$ (in decibels).

[8]The software for Structure SIMilarity index is available at http://www.ece.uwaterloo.ca/~z70wang/research/ssim/.

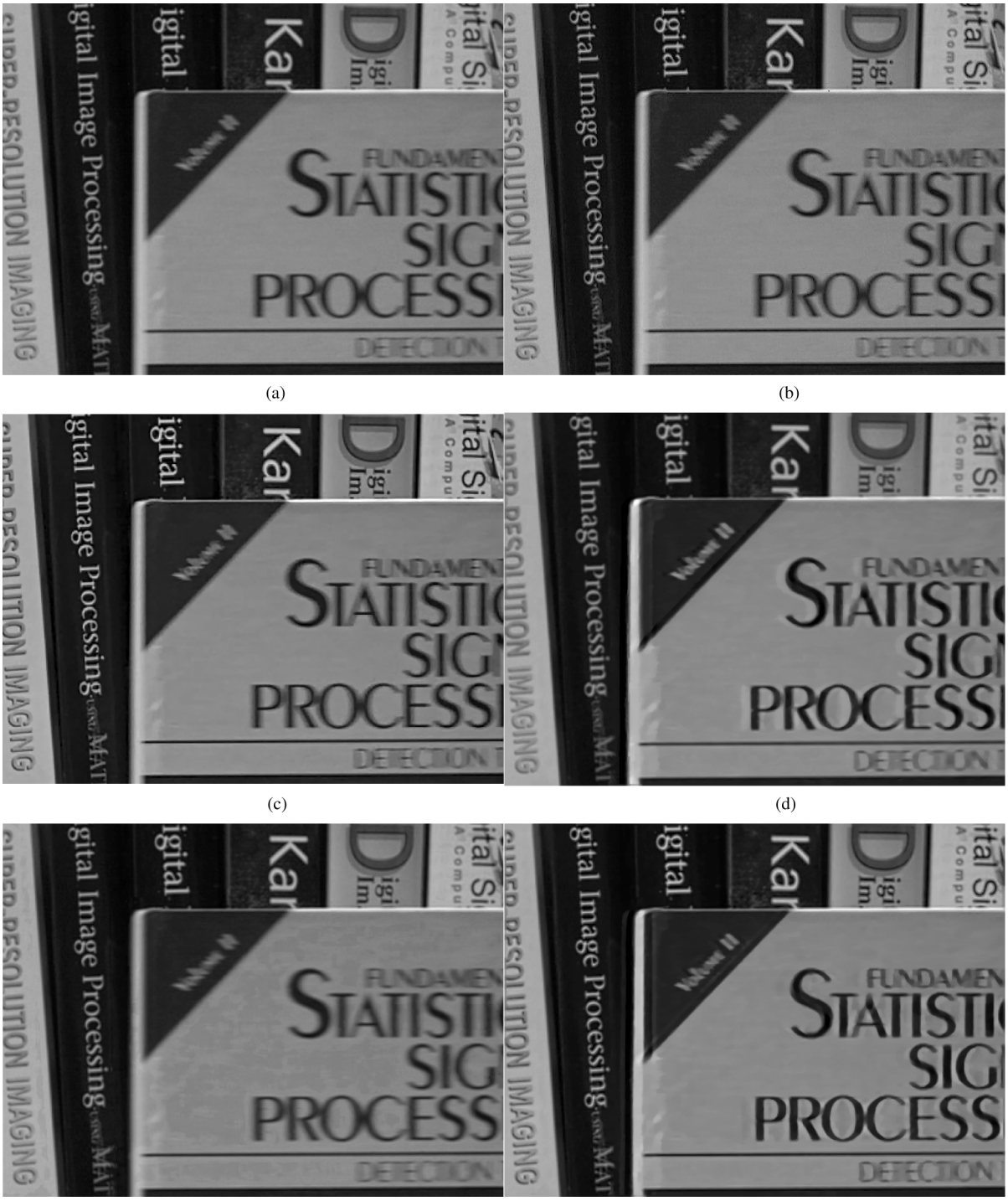

Fig. 7. Motion (temporal) deblurring example of the Book sequence (380×510, 10 frames) with real motion blur. (a) Frame of the ground truth at time $t = 6$. (b) and (c) Deblurred frames by Fergus's [3] and Shan's methods [4]. (d) and (f) Deblurred frames at $t = 6$ and 6.5 by the proposed 3-D TV method (13) using a $1 \times 1 \times 8$ uniform PSF. (e) One of the estimated intermediate frame at $t = 6.5$ by the 3-D SKR (10).

the motion blur is spatially variant. Even though the (temporal) PSF is independent of the scene contents or the camera motion, the shift-invariant 3-D PSF causes spatially variant motion blur effects. Using the blurred video as the output of a space–time interpolator, we deblurred the blurred video by Fergus' and Shan's blind methods. One deblurred frame by each blind method is shown in Fig. 2(c) and (d), respectively. Our deblurring result is shown in Fig. 2(e). We used the $1 \times 1 \times 8$ shift-invariant PSF for our deblurring (13) with $(\mu, \lambda) = (0.5, 0.15)$.

The next experiment shown in Fig. 7 is a realistic example, where we deblur a low temporal resolution sequence degraded by real motion blur. The cropped sequence consists of ten frames, and the sixth frame (at time $t = 6$) is shown in Fig. 7(a). Motion blur can be seen in the foreground (i.e., the book in front moves toward right about 8 pixels per frame). Similar to the previous experiment, we first deblurred those frames individually by Fergus' and Shan's methods [3], [4]. Their deblurred results are in Fig. 7(b) and (c), respectively. For our method,

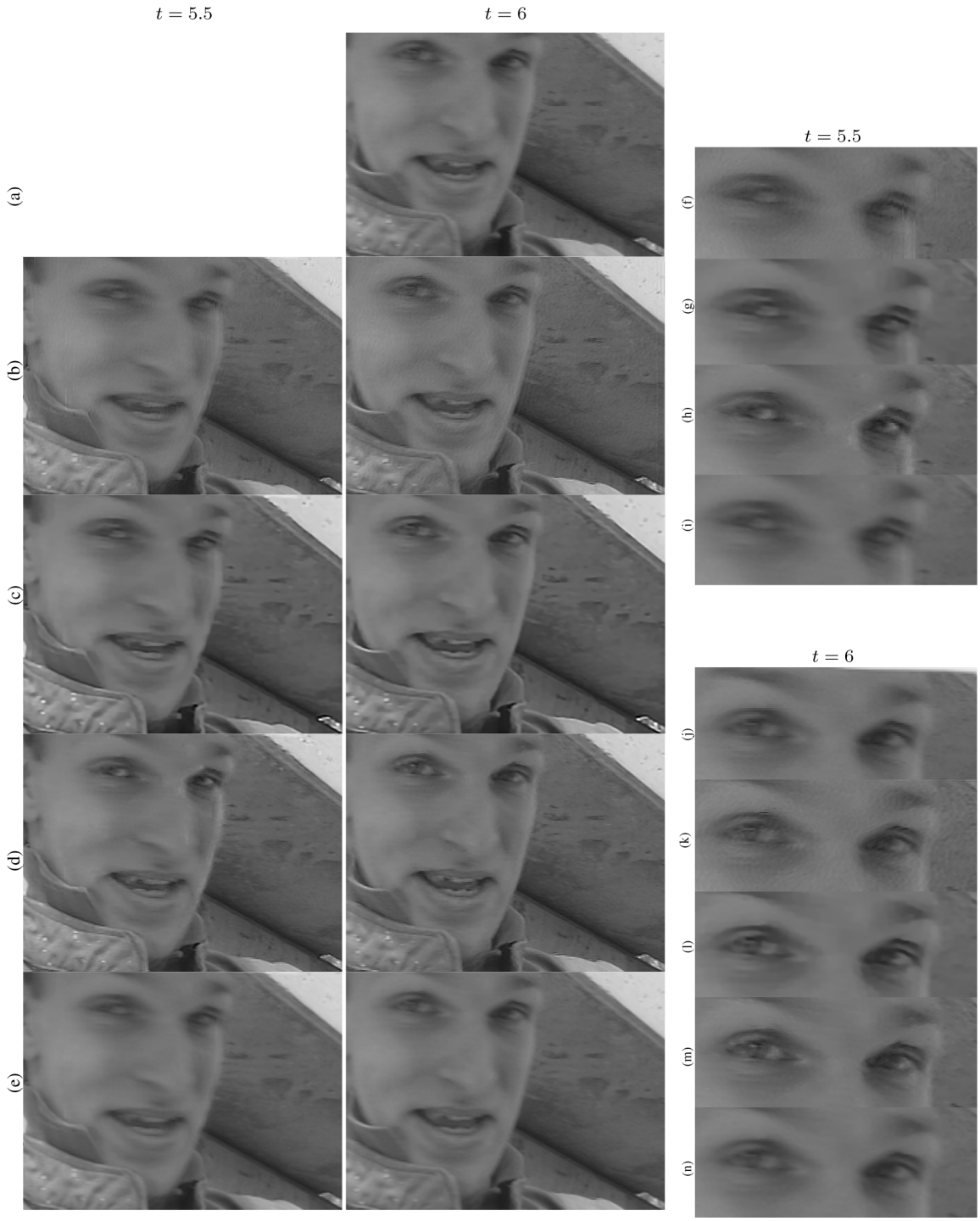

Fig. 8. 3-D (spatiotemporal) deblurring example of the Foreman sequence in CIF format. (a) Cropped frame at time $t = 6$. (b) and (c) Deblurred results of the upscaled frame shown in (e) by Fergus' [3] and Shan's methods [4] (d) Deblurred frames by the proposed 3-D TV method (13) using a $2 \times 2 \times 2$ uniform PSF. (e) Upscaled frames by 3-D SKR [1] at time $t = 6$ and 6.5 in both space and time with the spatial and temporal upscaling factors of $r_s = 2$ and $r_t = 8$, respectively. The figures (f)–(i) and (j)–(n) are the selected regions of the frames shown in (a)–(e) at $t = 6$ and 6.5.

temporal upscaling is necessary before deblurring. Here, it is indeed the case that exposure time is shorter than the frame interval ($\tau_e < \tau_f$), as shown in Fig. 3(a). Using the 3-D SKR method (10), we upscaled the sequence with the upscaling factors $r_s = 1$ and $r_t = 8$ in order to generate intermediate frames to have the sequence, as illustrated in Fig. 3(c). We chose $r_t = 8$ to slow the motion speed of the book down to about 1 pixel per frame so that the motion blur of the book will be almost completely removed. One of the estimated intermediate frames at $t = 6.5$ is shown in Fig. 7(e). Then, we deblurred the upscaled video with a $1 \times 1 \times 8$ uniform PSF by the proposed method (13) with $(\mu, \lambda) = (0.75, 0.06)$. We took the book video in dim light, and the exposure time is nearly equal to the frame interval. Selected deblurred frames[9] are shown in Fig. 7(d) and (f).

The last example is another real example. This time we used the Foreman sequence in CIF format. Fig. 8(a) shows one frame of the

[9]We must note that, in case severe occlusions are present in the scene, the blurred results for the interpolated frames contain most of the errors/artifacts, and this issue is one of or important future works.

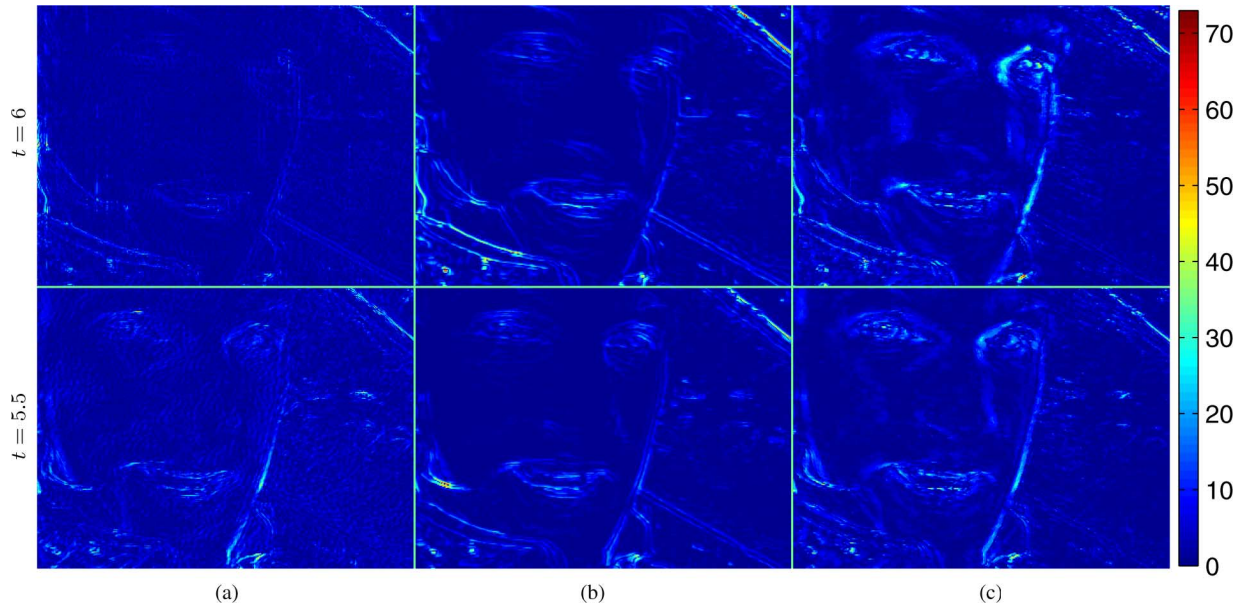

Fig. 9. Deblurring performance comparisons using absolute residuals (the absolute difference between the deblurred frames shown in Fig. 8(b)–(d) and the estimated frames shofwn in Fig. 8(e)). (a) Fergus' method [3]. (b) Shan's method [4]. (c) Our proposed method (13).

cropped input sequence (170×230, 10 frames) at time $t = 6$. In this example, we upscaled the Foreman sequence using 3-D SKR (10) with spatial and temporal upscaling factor of $r_s = 2$ and $r_t = 8$, respectively, and Fig. 8(e) show the estimated intermediate frame at time $t = 5.5$ and the estimated frame at $t = 6$. We note that these frames are the intermediate results of our two-step deblurring approach. We also note that our 3-D SKR successfully estimated the blurred intermediate frames, as seen in the figures, and the motion blur is spatially variant; the man's face is blurred as a result of the out-of-plane rotation of his head. In this time, we deblur the upscaled frames using Fergus' and Shan's methods [3], [4], and the proposed 3-D deblurring method using a $2 \times 2 \times 2$ uniform PSF. The exposure time of the Foreman sequence is unavailable, and we manually chose the temporal support size of the PSF to produce reasonable deblurred results. The deblurred frames are in Fig. 8(b)–(d), respectively, and Fig. 8(f)–(i) and (j)–(n) are the selected regions of the frames shown in (a)–(e) at $t = 5.5$ and 6, respectively. In addition, in order to compare the performance of our proposed method to Fergus' and Shan's methods, in Fig. 9, we compute the absolute residuals (the absolute difference between the deblurred frames shown in Fig. 8(b)–(d) and the estimated frames shown in Fig. 8(e) in this case). The results illustrate that our 3-D deblurring approach successfully recovers more details of the scene, such as the man's eye pupils, and the outlines of the face and nose even without scene segmentation.

IV. Conclusion and Future Works

In this paper, instead of removing the motion blur as spatial blur, we proposed deblurring with a 3-D space–time invariant PSF. The results showed that we could avoid segmenting video frames based on the local motions, and that temporal deblurring effectively removed motion blur even in the presence of motion occlusions.

For all the experiments in Section III, we assumed that exposure time was known. In our future work, we plan on extending the proposed method to the case, where the exposure time is also unknown.

References

[1] H. Takeda, P. Milanfar, M. Protter, and E. Elad, "Superresolution without explicit subpixel motion estimation," *IEEE Trans. Image Process.*, vol. 18, no. 9, pp. 1958–1975, Sep. 2009.

[2] Q. Shan, Z. Li, J. Jia, and C. Tang, "Fast image/video upsampling," presented at the ACM Trans. Graph. (SIGGRAPH ASIA), Singapore, 2008.

[3] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. Freeman, "Removing camera shake from a single photograph," *ACM Trans. Graph.*, vol. 25, pp. 787–794, 2006.

[4] Q. Shan, J. Jia, and A. Agarwala, "High-quality motion deblurring from a single image," *ACM Trans. Graph.*, vol. 27, pp. 73:1–73:10, 2008.

[5] M. Ben-Ezra and S. K. Nayar, "Motion-based motion deblurring," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 6, pp. 689–698, Jun. 2004.

[6] Y. Tai, H. Du, M. S. Brown, and S. Lin, "Image/video deblurring using a hybrid camera," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Anchorage, AK, Jun. 2008, pp. 1–8.

[7] S. Cho, Y. Matsushita, and S. Lee, "Removing non-uniform motion blur from images," in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, Rio de Janeiro, Brazil, Oct. 2007, pp. 1–8.

[8] A. Levin, "Blind motion deblurring using image statistics," presented at the Conf. Neural Inf. Process. Syst., Vancouver, BC, 2006.

[9] P. Milanfar, "Projection-based, frequency-domain estimation of superimposed translational motions," *J. Opt. Soc. Amer.: A, Opt. Image Sci.*, vol. 13, no. 11, pp. 2151–2162, Nov. 1996.

[10] P. Milanfar, "Two dimensional matched filtering for motion estimation," *IEEE Trans. Image Process.*, vol. 8, no. 3, pp. 438–444, Mar. 1999.

[11] D. Robinson and P. Milanfar, "Fast local and global projection-based methods for affine motion estimation," *J. Math. Imag. Vis. (Invited Paper)*, vol. 18, pp. 35–54, Jan. 2003.

[12] D. Robinson and P. Milanfar, "Fundamental performance limits in image registration," *IEEE Trans. Image Process.*, vol. 13, no. 9, pp. 1185–1199, Sep. 2004.

[13] H. Ji and C. Liu, "Motion blur identification from image gradients," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Anchorage, AK, Jun. 2008, pp. 1–8.

[14] S. Dai and Y. Wu, "Motion from blur," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Anchorage, AK, Jun. 2008, pp. 1–8.

[15] J. Chen, L. Yuan, C. Tang, and L. Quan, "Robust dual motion deblurring," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Anchorage, AK, Jun. 2008, pp. 1–8.

[16] A. Agrawal and R. Raskar, "Resolving objects at higher resolution from a single motion-blurred image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Minneapolis, MN, Jun. 2007, pp. 1–8.

[17] Y. Tai, N. Kong, S. Lin, and S. Shin, "Coded exposure imaging for projective motion deblurring," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, San Francisco, CA, Jun. 2010, pp. 2408–2415.
[18] E. Borissoff, "Optimal temporal sampling aperture for HDTV varispeed acquisition," *SMPTE Motion Imag. J.*, vol. 113, no. 4, pp. 104–109, 2004.
[19] E. Shechtman, Y. Caspi, and M. Irani, "Space-time super-resolution," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 4, pp. 531–545, Apr. 2005.
[20] A. Agrawal, M. Gupta, A. Veeraraghavan, and S. G. Narasimhan, "Optimal coded sampling for temporal super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, San Francisco, CA, 2010, pp. 599–606.
[21] B. Wilburn, N. Joshi, V. Vaish, M. Levoy, and M. Horowitz, "High-speed videography using a dense camera array," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Washington, DC, 2004, pp. 294–301.
[22] A. Huang and T. Nguyen, "Correlation-based motion vector processing with adaptive interpolation scheme for motion-compensated frame interpolation," *IEEE Trans. Image Process.*, vol. 18, no. 4, pp. 740–752, Apr. 2009.
[23] M. Irani and S. Peleg, "Improving resolution by image registration," *CVGIP: Graph. Models Image Process.*, vol. 53, no. 3, pp. 231–239, May 1991.
[24] H. Takeda, S. Farsiu, and P. Milanfar, "Kernel regression for image processing and reconstruction," *IEEE Trans. Image Process.*, vol. 16, no. 2, pp. 349–366, Feb. 2007.
[25] L. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D*, vol. 60, pp. 259–268, Nov. 1992.
[26] C. Vogel and M. Oman, "Iterative methods for total variation denoising," *SIAM J. Sci. Comput.*, vol. 17, pp. 227–238, 1996.
[27] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin, "An iterative regularization method for total variation-based image restoration," *SIAM J. Multiscale Model. Simul.*, vol. 4, pp. 460–489, 2005.