Rolling Shutter Motion Deblurring

Shuochen Su[1,2] Wolfgang Heidrich[2,1]
[1]University of British Columbia [2]KAUST

Abstract

Although motion blur and rolling shutter deformations are closely coupled artifacts in images taken with CMOS image sensors, the two phenomena have so far mostly been treated separately, with deblurring algorithms being unable to handle rolling shutter wobble, and rolling shutter algorithms being incapable of dealing with motion blur.

We propose an approach that delivers sharp and undistorted output given a single rolling shutter motion blurred image. The key to achieving this is a global modeling of the camera motion trajectory, which enables each scanline of the image to be deblurred with the corresponding motion segment. We show the results of the proposed framework through experiments on synthetic and real data.

1. Introduction and Related Work

Motion blur (MB) from camera shake is one of the most noticeable degradations in hand-held photography. Without information about the underlying camera motion, an estimate of the latent sharp image can be restored using so-called blind deblurring (BD), which has been studied extensively in the past decade. Some representative work on this problem includes Cho *et al*. [6] in which salient edges and the FFT are used to speed up the uniform deblurring, and the introduction of motion density functions by Gupta *et al*. [7], which specializes in non-uniform cases. Maximum a posteriori (MAP) estimation is a popular way to formulate the objective of blind deblurring – a theoretical analysis on its convergence was conducted in [12] recently.

One common assumption made in almost all previous deblurring methods [6, 7, 8, 9, 12, 15, 16, 17, 18] is the use of a global shutter (GS) sensor. That is, each part of the image is regarded as having been exposed during the exact same interval of time. This assumption, however, does *not* hold for images captured by a CMOS sensor that uses an electronic rolling shutter (RS), which is the case for the majority of image sensors on the market. This is due to the fact that RS exposes each row sequentially as opposed to simultaneously, and that the popularity of RS sensors in mobile devices makes them particularly susceptible to motion blur, especially in low light scenarios. Fig. 1 demonstrates some samples of the point-spread-functions (PSFs) simulated by applying the same camera motion to a global and a rolling shutter camera during exposure. Assuming a static scene and no in-plane rotation, the blur kernel of the global shutter image in Fig. 1(a) is shift invariant, since all pixels integrate over the same motion trajectory. For the rolling shutter image, however, different scanlines integrate over a slightly different segment of the trajectory, resulting in a shift-variant kernel in Fig. 1(b) even in the case of static object and no in-plane rotation. Thus, when applied to the wide range of RSMB images such as that shown in Fig. 1(c), existing methods [6, 7, 8, 9, 12, 15, 16, 17, 18] are destined to fail.

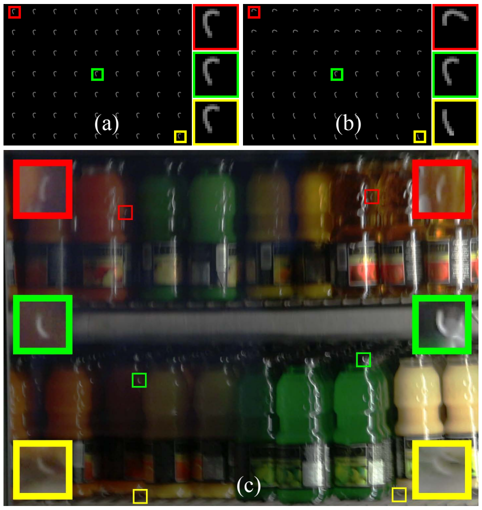


Figure 1: Simulated blur kernels for global (a) and rolling shutter (b) cameras. Blur kernels in global shutter images tend to be spatially invariant (assuming a static scene and negligible in-plane roation) while in a rolling shutter image they are always spatially variant. (c) Spatial variance of the blur kernel in a real RSMB image indicated by light streaks.

The shift variance of the rolling shutter kernel effect can be modeled by capturing the overall camera motion with a gyroscope and computing different kernels for each scanline [11]. Without such specialized hardware, an alternative is to solve different blind deconvolution problems for blocks

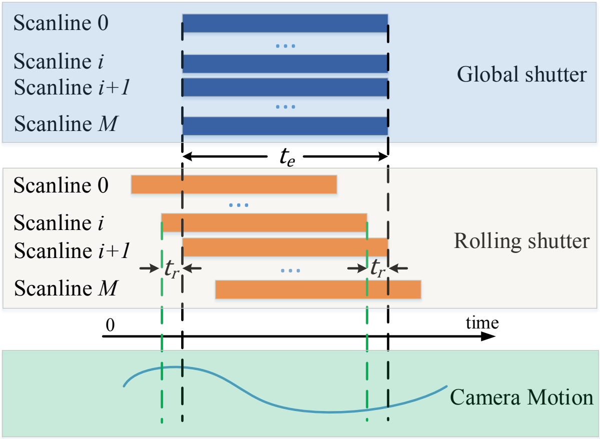


Figure 2: Illustration of the GS (top) and RS (middle) sensor mechanisms. The horizontal bar represents the exposure interval of the i-th scanline in the sensor. $M + 1$ is the number of scanlines in both sensors, t_e denotes the exposure time of each scanline, and t_r is the time the RS sensor takes to readout the i-th scanline before proceeding to the next. When camera motion exists (bottom), all pixels in a GS camera integrate over the same motion trajectory, while different scanlines integrate over a slightly different segment of the trajectory in a RS sensor.

of scanlines, but these solutions would have to be stitched together, which is made more difficult by the rolling shutter wobble.

Geometric distortions in RS images have received more attention, but typically without considering blur. A major motivation for rolling shutter correction is the removal of wobbles in RS videos [1], where inter-row parallax is larger than one pixel. The work by Saurer *et al.* [14] is another example that attempts to make traditional computer vision algorithms (*e.g.* stereo, registration) work on RS cameras. Most of such work relies on sharp input images so that feature detectors can reliably deliver inter-frame homography estimates. Meilland et al. [10] were the first to propose a unified framework for RS and MB, but they rely on a sequence of images that can be registered together. Another approach was presented by Pichaikuppan et al. [13], but it targets change detection and requires a sharp global shutter reference image.

In this paper, we propose a single image approach that deblurs RSMB images by estimating and parametrically modeling each degree of freedom of the camera motion trajectory as polynomial functions of time. The motivation of this parametric representation is based on a recently study of human camera shake from Kohler *et al.* [9], see Sec. 3. To achieve good initial estimates of the motion trajectory coefficients, we adopt a back-projection technique [8] to estimate higher dimensional camera poses from their 2D projections, *i.e.* PSFs. The specific contributions of this work are:

- A blind deblurring technique that handles the characteristic of rolling shutter;
- A method for motion trajectory estimation and refinement from a single RSMB image.

Throughout the paper we assume the scene to be sufficiently far away so that planar homographies are able to describe the transformations. Based on an analysis of the data from Kohler *et al.* [9], we also determine that in-plane rotation is negligible except for very wide angle lenses, allowing us to restrict ourselves to a 2D translational motion model.

2. Motion Blur in Rolling Shutter Cameras

In the global shutter case, motion blur with a spatially varying kernel can be modeled as a blurred image $\mathbf{G} \in \mathbb{R}^{(M+1)\times(N+1)}$ resulting from the integration over all instances of the latent image $\mathbf{L} \in \mathbb{R}^{(M+1)\times(N+1)}$ seen by the camera at poses along its motion path during the exposure period $t \in [0, t_e]$:

$$\mathbf{G} = \frac{1}{t_e}\int_0^{t_e} \mathbf{L}^{\mathbf{p}(t)} dt + \mathbf{N}, \tag{1}$$

where $\mathbf{p}(t) = (p_1(t), p_2(t), p_3(t)) \in \mathbb{R}^3$ corresponds to the camera pose at time t, and $\mathbf{L}^{\mathbf{p}(t)}$ is the latent image transformed to the pose $\mathbf{p}$ at a given time. $\mathbf{N}$ represents a noise image, which is commonly assumed to follow a Gaussian-distribution in each pixel.

The above model proves to be effective for formulating a GSMB image but cannot be directly applied to the RSMB case, as scanlines in RS sensors are exposed sequentially instead of simultaneously as that is assumed in Eq. (1). Specifically, as illustrated in Fig. 2, although each scanline is exposed for the same duration t_e, the exposure window is offset by t_r from scanline to scanline [4]. We thus rewrite Eq. (1) for each row $\mathbf{b}_i$ in a RSMB image $\mathbf{B} = (\mathbf{b}_0^T, \ldots, \mathbf{b}_M^T)^T$ as

$$\mathbf{b}_i = \frac{1}{t_e}\int_{i\cdot t_r}^{i\cdot t_r + t_e} \mathbf{l}_i^{\mathbf{p}(t)} dt + \mathbf{n}_i, \tag{2}$$

where the subscript i also indicates the i-th row in $\mathbf{L}^{\mathbf{p}(t)}$ and $\mathbf{N}$.

Eq. (2) can be expressed in discrete matrix-vector form after assuming a finite number of time samples during the exposure of each row. Assuming that the camera intrinsic matrix $\mathbf{C} \in \mathbb{R}^{3\times3}$ is given, $b_{ij} \in \mathbf{b}_i = (b_{i0}, ..., b_{iN})$ can be exactly determined at any pixel location $\mathbf{x} = (i, j)^T$ in $\mathbf{B}$

$$b_{ij} = \frac{1}{|\mathbb{T}_i|}\sum_{t\in\mathbb{T}_i} \Gamma_{\mathbf{L}}(\mathbf{w}(\mathbf{x}; \mathbf{p}(t))) + n_{ij}, \tag{3}$$

where $\mathbb{T}_i = \{i \cdot t_r + \frac{j}{K} t_e\}_{j=0\ldots K}$ is a set of uniformly spaced time samples in the exposure window of row i, $\Gamma_{\mathbf{L}}(\cdot)$ is the function that bi-linearly interpolates the intensity at some sub-pixel position in $\mathbf{L}$, and $\mathbf{w}(\cdot)$ is a warping function [2, 5] that maps positions $\mathbf{x}$ from the camera frame back to the reference frame of the latent image $\mathbf{L}$ according to the current camera pose $\mathbf{p}$:

$$\mathbf{w}(\mathbf{x};\mathbf{p}) = \frac{\mathbf{D}\mathbf{H}(\mathbf{D}^T\mathbf{x}+\mathbf{e})}{\mathbf{e}^T\mathbf{H}(\mathbf{D}^T\mathbf{x}+\mathbf{e})}. \quad (4)$$

Here $\mathbf{H} = \mathbf{C}\mathbf{R}\mathbf{C}^{-1}$ is a homography matrix, $\mathbf{R} = e^{\mathbf{p}_\times}$ is its rotational component[1], $\mathbf{e} = (0,0,1)^T$, and

$$\mathbf{D} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

From Eq. (3,4) a sparse convolution matrix $\mathbf{K}$ can be created and Eq. (2) can be rewritten as

$$\mathbf{b} = \mathbf{K}\mathbf{l} + \mathbf{n}, \quad (5)$$

where $\mathbf{b}$, $\mathbf{l}$ and $\mathbf{n}$ respectively are the RSMB input, latent image, and noise, all in vector form.

3. Camera Motion Modeling

The sequence of camera poses $\mathbf{p}(t)$ describes a 1D continuous path through the 3D camera pose space [7]. Before explaining how this serves as an important constraint for blind RSMD, we first seek a suitable model for $\mathbf{p}(t)$ from $t = 0$ to $t_e + Mt_r$, and then rewrite the RSMB image formation model based on this modeling.

Exposure time of interest. To discuss the model for $\mathbf{p}(t)$, we need to be more specific about the temporal range of $\cup_{i=0:M}\{\mathbb{T}_i\}$ for exposing the whole image. If the exposure time is large compared to the aggregate readout time for a full frame (i.e. $t_e \gg Mt_r$), then the rolling shutter distortion is dominated by the motion blur and conventional deblurring methods can be used with good results. Unfortunately, this scenario can only occur in still photography, since in video mode the exposure time cannot exceed one over the frame rate. The other extreme is that there is sufficient light for the exposure time to be very short compared to the readout time (i.e. $t_e \ll Mt_r$). In this case, the blur is typically negligible, except for very fast camera motion. Inbetween these two extremes is the common scenario where the exposure time approaches the aggregate readout time. While this scenario is particularly common in small-format cameras and cell-phones, it affects all cameras in lower light conditions. In this case, both rolling shutter distortion and motion blur are present, and existing methods cannot be used.

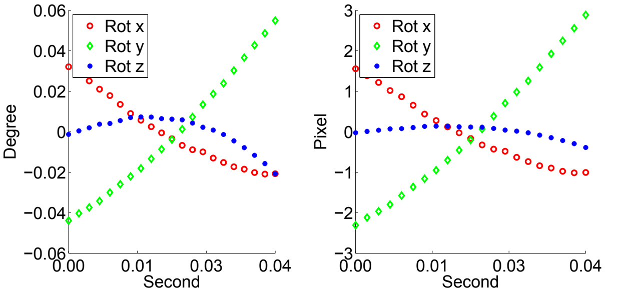


Figure 3: *Left*: a segment of the camera motion trajectory from [9] in 3D rotational pose space; *Right*: the resulting pixel shift at the corner of a full-HD image, assuming the rotation center is the image center. With a moderate angled lens (50mm in this example), the effect of roll (Rot z) reduces to subpixel-level pixel shift in most handheld shake cases. See the supplemental material for more examples.

Pose trajectory model. To get a sense of what the sequence of $\mathbf{p}(t)$ typically looks like for a shaky hand-held camera during the time of interest, we performed an analysis over all the 40 publicly available camera motion trajectories from Kohler *et al*. [9]. The camera pose was recorded at 500Hz by a Vicon system when 6 human objects were asked to hold still a camera in a natural way. Detailed descriptions of the experiment can be referred to the original paper. Here we illustrate in Fig. 3 the three rotational pose trajectories during a randomly selected 1/25s segment (108-128) from the 39th dataset.

As can be seen, even though the blur kernel of a RSMB image varies spatially, the decomposed $p_1(t), p_2(t), p_3(t)$ from the underlying camera motion are in fact parameterizable. This observation generalizes to the other samples from the same dataset. Therefore we decide to fit polynomial functions to the pose trajectories

$$\mathbf{p}(t) = \mathbf{t}\boldsymbol{\theta}, \quad (6)$$

where $\mathbf{t} = (t^P, \ldots t^0)$, $t \in [0, Mt_r + t_e]$, and $\boldsymbol{\theta}$ is a $(P+1) \times 3$ matrix having coefficients of each polynomial function as entries. In this work we set the polynomial degree P to 3 or 4, which achieves a good fit. We note that quartic splines provide a better fit if the trajectory is more complex, *i.e.* the camera shakes at higher frequencies, but this is rarely the case for natural hand-held camera shakes within the exposure time of interest [9].

Finally, we also note that there is an interesting subset of RSMB images captured under medium to long focal lengths, where the contribution of in-plane rotation ($p_3(t)$) is, in fact, so small that it does not result in a noticeable blur – Figure 3 (right) shows an example of the maximal blur caused by in-plane rotation in a full-HD image for an assumed focal length of 50mm. In this example, the blur just due to in-plane rotation is below one pixel even in the

[1] $\mathbf{p}_\times = \begin{bmatrix} 0 & -p_3 & p_2 \\ p_3 & 0 & -p_1 \\ -p_2 & p_1 & 0 \end{bmatrix}$ as the matrix exponential.

Algorithm 1 RSMD algorithm overview.

```
INPUT: RSMB image $\mathbf{b}$, initial weight $\mu_0$
Obtain the trajectory initialization $\boldsymbol{\theta}_0$ (Sec. 4.2.2).
for $k$ = 1 to $n$ do
    Update $\mathbf{l}$ (Eq. 15)
    Update $\boldsymbol{\theta}$ (Eq. 9)
    Decrease $\mu$ by $\tau$
end for
OUTPUT: Deblurred image $\mathbf{l}$, trajectory coefficients $\boldsymbol{\theta}$
```

corner of the image. Similar results are obtained for other motions from Kohler *et al.*'s database. We therefore neglect in-plane rotation in this work, and reduce Eq. (6) to a 2D yaw/pitch space instead of a full 3D rotational pose space for the rest of this paper.

RSMB modeling in trajectory coefficients. With the trajectory model defined in Eq. (6), the convolution matrix $\mathbf{K}$ in Eq. (5) becomes a function of $\boldsymbol{\theta}$

$$\mathbf{b} = \mathbf{K}(\boldsymbol{\theta})\mathbf{l} + \mathbf{n}, \tag{7}$$

where $\mathbf{K}(\boldsymbol{\theta})$ is determined by rewriting Eq. (3,4) in $\boldsymbol{\theta}$ accordingly.

4. Deblurring RSMB Image

Having the forward RSMB model defined in the form of a camera motion model, the latent image $\mathbf{l}$ can be recovered from $\mathbf{b}$ by solving an inverse problem. An overview of our blind RSMD approach is summarized in Algorithm 1 and we will describe it in details in this section.

4.1. Objective

Our objective function for RSMD is given by

$$\min_{\mathbf{l},\boldsymbol{\theta}} \frac{1}{2}\|\mathbf{b} - \mathbf{K}(\boldsymbol{\theta})\mathbf{l}\|_2^2 + \mu\|\nabla\mathbf{l}\|_1, \tag{8}$$

which is composed of a data error term based on Eq. (5) and a sparsity prior on the latent image gradient $\nabla\mathbf{l}$, weighted by a scalar μ. Since the camera motion and kernel normalization [12] are inherently implemented in $\mathbf{K}$, no additional prior for $\mathbf{K}(\boldsymbol{\theta})$ is required in the objective, which differentiates our method from [7, 18].

Similar to conventional blind deblurring algorithms, we update $\mathbf{l}$ and $\boldsymbol{\theta}$ in an alternating fashion. We initialize μ with a relative large value μ_0, thus in the early iterations only the most salient structure in $\mathbf{l}$ will be preserved which will guide the refinement of kernel coefficients $\boldsymbol{\theta}$, given that $\boldsymbol{\theta}$ estimates are not yet accurate. As the optimization progresses, we decrease μ by a factor of τ after each iteration to preserve more details in $\mathbf{l}$. Intermediate outputs are shown in Fig. 5 and the supplemental material.

4.2. Update of Trajectory Coefficients

The objective for updating $\boldsymbol{\theta}$ is given by

$$\boldsymbol{\theta}^{k+1} = \arg\min_{\boldsymbol{\theta}} \sum_{i=0}^{M}\sum_{j=0}^{N} r_{ij}(\boldsymbol{\theta})^2, \tag{9}$$

where $r_{ij}(\boldsymbol{\theta})$ is the residual at $\mathbf{x}$ that depends on $\boldsymbol{\theta}$

$$r_{ij}(\boldsymbol{\theta}) = b_{ij} - \frac{1}{|\mathbb{T}_i|}\sum_{t\in\mathbb{T}_i}\Gamma_{\mathbf{L}^k}(\mathbf{w}(\mathbf{x};\mathbf{t}\boldsymbol{\theta})). \tag{10}$$

Here $\mathbf{L}^k$ is the latent image estimated from the previous iteration k. Solving Eq. (9) is a non-linear optimization task since pixel values in $\mathbf{L}$ are, in general, non-linear in $\boldsymbol{\theta}$.

4.2.1 Gauss-Newton Method

Motivated by image registration, *i.e.* the Lucas-Kanade algorithm [2], we adopt Gauss-Newton method for this non-linear least square problem. In each iteration, Gauss-Newton updates $\boldsymbol{\theta}$ by

$$\boldsymbol{\theta}^{k+1} = \boldsymbol{\theta}^k + \Delta\boldsymbol{\theta}, \tag{11}$$

where $\Delta\boldsymbol{\theta}$ is the solution for the linear system

$$\mathbf{J}_{\mathbf{r}}^T\mathbf{J}_{\mathbf{r}}\,\mathrm{vec}(\Delta\boldsymbol{\theta}) = \mathbf{J}_{\mathbf{r}}^T\mathbf{r}. \tag{12}$$

Here $\mathbf{r}$ is the residual vector, and $\mathbf{J}_{\mathbf{r}}$ is the Jacobian of the residual, both evaluated at $\boldsymbol{\theta}^k$.

The calculation of $\mathbf{J}_{\mathbf{r}}$ is carried out as follows by applying the chain rule on Eq. (10)

$$\nabla r_{ij}(\boldsymbol{\theta})|_{\boldsymbol{\theta}=\boldsymbol{\theta}^k} = -\frac{1}{|\mathbb{T}_i|}\sum_{t} J_{\Gamma_{\mathbf{L}^k}}(\mathbf{w})J_{\mathbf{w}}(\mathbf{p})J_{\mathbf{p}}(\boldsymbol{\theta}), \tag{13}$$

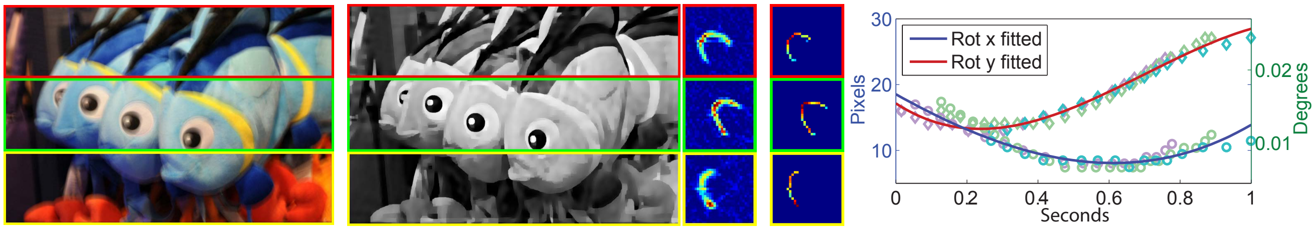
where $J_{\Gamma_{\mathbf{L}^k}}(\mathbf{w})$ is the gradient of image $\mathbf{L}^k$ at $\mathbf{w}(\mathbf{x};\mathbf{t}\boldsymbol{\theta}^k)$. We find Conjugate Gradients efficient for calculating $\Delta\boldsymbol{\theta}$ in Eq. 12.

4.2.2 Initialization of Trajectory Coefficients

A good initial estimate of $\boldsymbol{\theta}$ is essential for the convergence of Gauss-Newton. We approach this problem by solving blind deconvolution problems for blocks of several scanlines over which the PSF is assumed to be approximately constant *for initialization purposes only*. The recovered PSFs for each block are then back-projected into pose space, and initial trajectory estimates are fit. An illustration of this process is also given in Fig. 4.

Local blur kernel estimation. Given a RSMB image (Fig. 4a), we first divide it into several horizontal regions.Inside each region a first estimation of the kernel is recovered with a conventional blind deblurring algorithm

$$\min_{\mathbf{k},\mathbf{l}} \frac{1}{2}\|\mathbf{k} * \mathbf{l} - \mathbf{b}\|_2^2 + \omega_1\|\nabla\mathbf{l}\|_1 + \frac{\omega_2}{2}\|\mathbf{k}\|_2^2, \tag{14}$$



(a) Input. (b) Local blur kernel estimation. (c) (d) Fitted trajectory.

Figure 4: Pipeline of $\boldsymbol{\theta}_0$ initialization. (a) RSMB image; (b) Locally estimated latent patches and blur kernels; (c) Blur kernels after thinning operation; (d) Polynomial fitting of the traced time-stamped data from (c), shown in units of both pixels and degrees.

where $\mathbf{k}$ is the blur kernel and $\mathbf{b}$ is the blurry input for a horizontal region of $\mathbf{B}$. ω_1 and ω_2 are weights that balance the trade-off between the data term and priors on the intrinsic image and blur kernel. This blind deconvolution in Eq. 14 itself is a non-convex optimization problem, and a multi-scale strategy is adopted for avoiding local minima. We show examples of the estimated $\mathbf{l}$ and $\mathbf{k}$ in Fig. 4b.

The blur kernel estimated from the previous step is usually noisy. We extract its skeleton by applying morphological thinning. To preserve the temporal information indicated by intensities, we convolve the initial blur kernel in Fig. 4b with a normalized all-one kernel (*e.g.* 3×3), and then assign the corresponding intensities to its thinned version. Results of this step are shown in Fig. 4c.

Back-projection and tracing. Back-projecting local blur kernels to the camera pose space is trivial for the yaw-pitch only case, but in order to reconstruct the motion trajectory a time stamp needs to be assigned to each camera pose. We achieve this by tracing the curves in Fig. 4c from one end to the other while assigning the time stamps as the accumulated intensity value along the way to each pixel. To avoid outliers when fitting in the next step we also assign a confidence value to the traced data that is proportional to its intensity. Fig. 4d plots the traced result in scattered dots, where the horizontal axis is the time and the vertical axis is the traced pixel locations. The tracing direction is determined by enumerating all possible directions and picking the one with the least fitting residual.

Polynomial fitting. $\boldsymbol{\theta}_0$ can be at last estimated by polynomial fitting the aligned time stamped data from the previous step. We show the fitted curve in Fig. 4d as well.

4.3. Latent Image Update

Fixing $\boldsymbol{\theta}^k$, we solve the latent image update subproblem

$$\mathbf{l}^{k+1} = \arg\min_{\mathbf{l}} \frac{1}{2}\|\mathbf{K}(\boldsymbol{\theta}^k)\mathbf{l} - \mathbf{b}\|_2^2 + \mu\|\nabla\mathbf{l}\|_1. \qquad (15)$$

We use the alternating direction method of multipliers (ADMM) algorithm outlined in Algorithm 2 to address the presence of L1 norm in the objective.

Algorithm 2 ADMM algorithm

```
$\mathbf{l}^{k+1} = \arg\min_{\mathbf{l}} \mathcal{L}_\rho(\mathbf{l}, \mathbf{j}^k, \lambda^k)$
$\mathbf{j}^{k+1} = \arg\min_{\mathbf{j}} \mathcal{L}_\rho(\mathbf{l}^{k+1}, \mathbf{j}, \lambda^k)$
$\lambda^{k+1} = \lambda^k + \rho(\mathbf{D}\mathbf{l}^{k+1} - \mathbf{j}^{k+1})$
```

Here we derive the optimization. We first rewrite the problem as

$$\begin{aligned} \mathbf{l}_{opt} &= \arg\min_{\mathbf{l}} G(\mathbf{l}) + F(\mathbf{j}) \\ &\text{subject to } \mathbf{D}\mathbf{l} = \mathbf{j}, \end{aligned} \qquad (16)$$

where $G(\mathbf{l}) = (1/2\mu)\|\mathbf{K}\mathbf{l} - \mathbf{b}\|_2^2$, $F(\mathbf{j}) = \|\nabla\mathbf{l}\|_1$. This can be expressed as an augmented Lagrangian

$$\begin{aligned} \mathcal{L}_\rho(\mathbf{l}, \mathbf{j}, \lambda) =& G(\mathbf{l}) + F(\mathbf{j}) + \\ & \lambda^T(\mathbf{D}\mathbf{l} - \mathbf{j}) + \frac{\rho}{2}\|\mathbf{D}\mathbf{l} - \mathbf{j}\|_2^2 \end{aligned} \qquad (17)$$

This is now a standard problem that can be solved efficiently with ADMM. Please refer to [3] for more details.

5. Experiments and Results

We perform a series of experiments on both synthetic and real RSMB images, and conduct quantitative and qualitative comparison with conventional blind deblurring work, see Figs. 6-8. To further demonstrate the power of our method, we also compare against a strategy of first rectifying the rolling shutter wobble from videos containing the specific frames before applying conventional blind deblurring algorithms.

5.1. Synthetic Images

The synthetic RSMB images are obtained by simulating a RS sensor with $t_r = 1/50Ms$ and $t_e = 1/50$s, which is one of the standard settings in conventional CMOS sensor cameras when capturing still images or video frames. For synthesizing the camera motion, we selected two segments

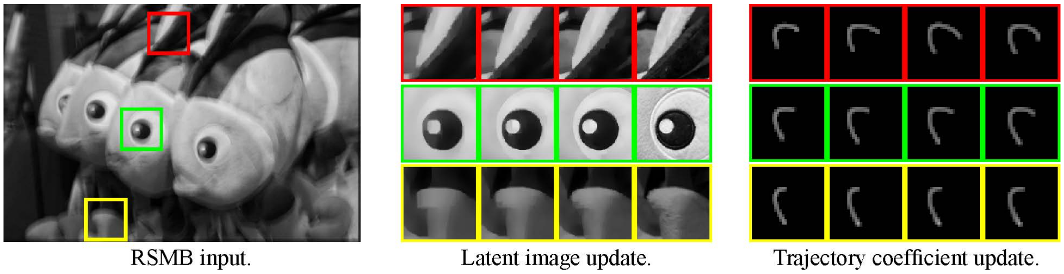


Figure 5: Intermediate results when processing a RSMB image. *Left*: input image; *Middle*: intrinsic image updates; *Right*: camera motion trajectory coefficients $\boldsymbol{\theta}$ updates (visualized as blur kernel). Columns in the intermediate steps represent the first, middle and final iterations, as well as the ground truth values.

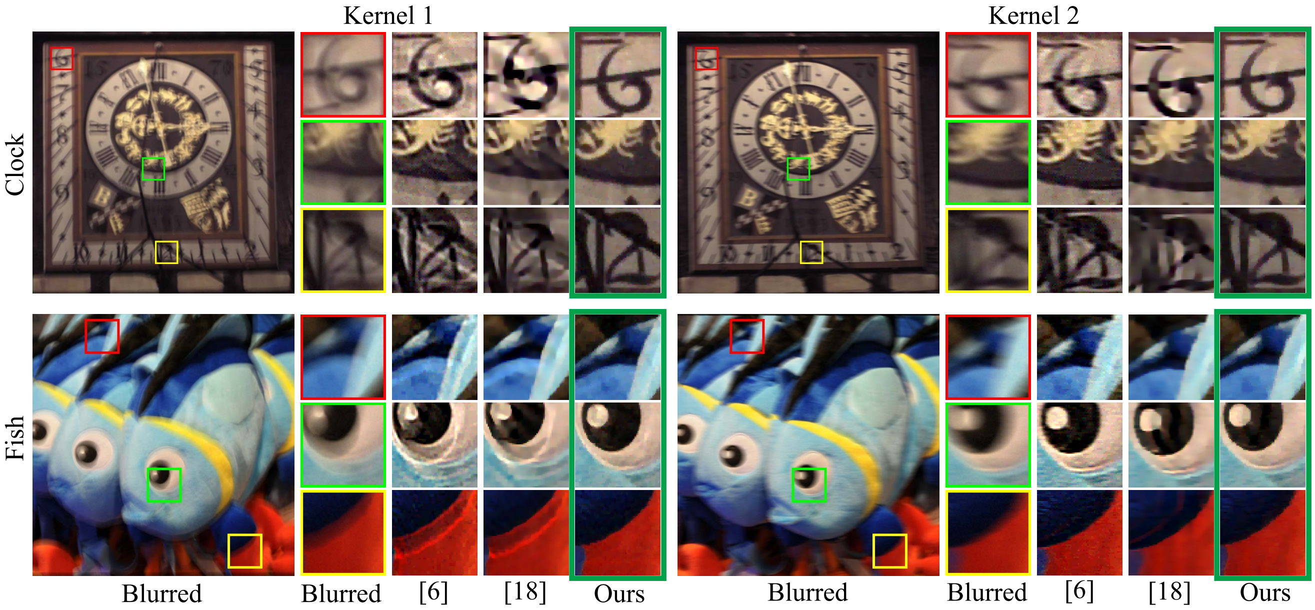


Figure 6: Two sets of test images, *Clock* and *Fish*, synthetically blurred with two typical kinds of camera motion. Zoomed in comparisons shown next to the RSMB images are, from left to right, cropped regions of the blurred image, output from [6] which assumes uniform motion, output from [18] which assumes non-uniform model, and our RSMD output highlighted in green. Please refer to the supplemental material for full-size comparisons and kernel estimations.

of the motion trajectory from Kohler *et al*. [9] that is of the length of our specified t_e and t_r, and applied the motion blur to two images, *Fish* and *Clock*. This gives us 4 sets of RSMB images as shown in Fig. 6 along with the ground truth.

Although both RS and MB deformations exist simultaneously for all kinds of camera motions, the two specific camera motions in the experiment here were selected to highlight specific behaviors. The first motion is highly curved, which means that different regions in the RS image are blurred along different directions, thus emphasizing the spatial variation of the blur kernel. The second blur kernel is predominantly linear, such that the blur kernel is similar for different image regions, although they are displaced by different amounts. This results in geometric distortions known as rolling shutter wobble.

On the first motion, both the uniform [6] and non-uniform [18] methods fail to address the sequential exposure mechanism of rolling shutter. As a result, ringing artifacts are unavoidable because of the incorrect kernel estimation, even though a relatively large weight on the image prior was used to obtain the results. Because our model takes temporal information into account and optimizes the global camera motion trajectory, instead of the discrete camera poses, a better kernel estimation and thus sharper latent image can be estimated.

With the second kind of blur, the resulting distortion/wobble could in the past only be addressed non-blindly

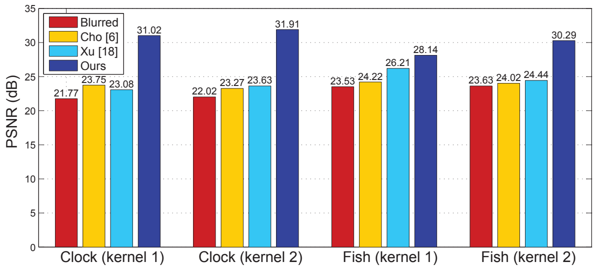


Figure 7: PSNR of the synthetic results.

with multiple sharp frames. Previous work [6, 18] successfully approximates the dominant blur kernel in this case, but leaves geometric distortions unaddressed due to the lack of global motion trajectory modeling within the exposure of the image.

To perform a quantitative evaluation we adopt the metric described in [9], where a minimization problem was first solved to find the optimal scale and translation between the ground truth and the output image from each algorithm. We give the PSNR in Fig. 7, where our method outperformed previous works. Notice that although the geometric misalignment (due to wobble) contributes to part of the PSNR loss in previous methods, their deblurred outputs also contain significant artifacts.

5.2. Real Images

To collect the real RSMB images we captured a short footage in a handheld setting using a Canon 70D DSLR camera mounted with a 50mm lens. A single frame was then extracted from the video which was shot at 24fps and an exposure time of 1/50s for each frame. With a sequence of RSMB images in hand we are able to perform our blind RSMD on each frame to test our joint RS and MB recovery. The footage also provides a necessary input for conventional RS rectification/video stabilization algorithms. We chose a DSLR over a cell-phone because it gives us access to specific values for t_e and t_r, as needed by our method. We adopted the method in [4] to obtain the value of t_r, which is determined by the frame rate and the total number of scanlines per frame.

We show our results in comparison to those of [6, 18] in Fig. 8. The insets clearly demonstrate the details recovered by our method. Please see the supplemental material for full-sized comparisons.

We compare our single image method with those followed by RS-rectification-and-then-blind-deblurring procedure in Fig. 9 (left). The entire video was processed with the rolling shutter correction feature included in Adobe After Effects CS6, before applying the conventional blind deblurring algorithm [17] to the rectified frame. This method still does not delivers comparable results to ours. This is due to the fact that conventional rolling shutter/video stabilization work is incapable of dealing with motion blur. Even after correctly rectifying the image, the blur kernel inside the frame is still difficult to model using traditional GSMD techniques.

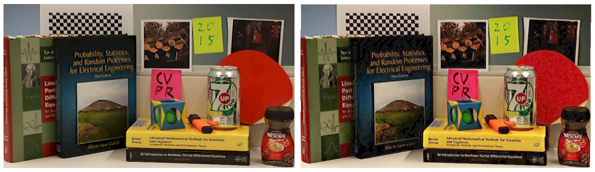


Figure 9: *Left*: Results of applying blind deblurring on the rolling shutter rectified images; *Right*: Results of the deblur-and-stitch strategy.

In Fig. 9 (right) we put the results of the stitched image of multiple deblurred blocks from the RSMB input. With this method, there is a tradeoff in quality between large blocks with potentially non-uniform kernel, and small blocks with potentially insufficient detail. The global trajectory model in our method inherently addresses this problem.

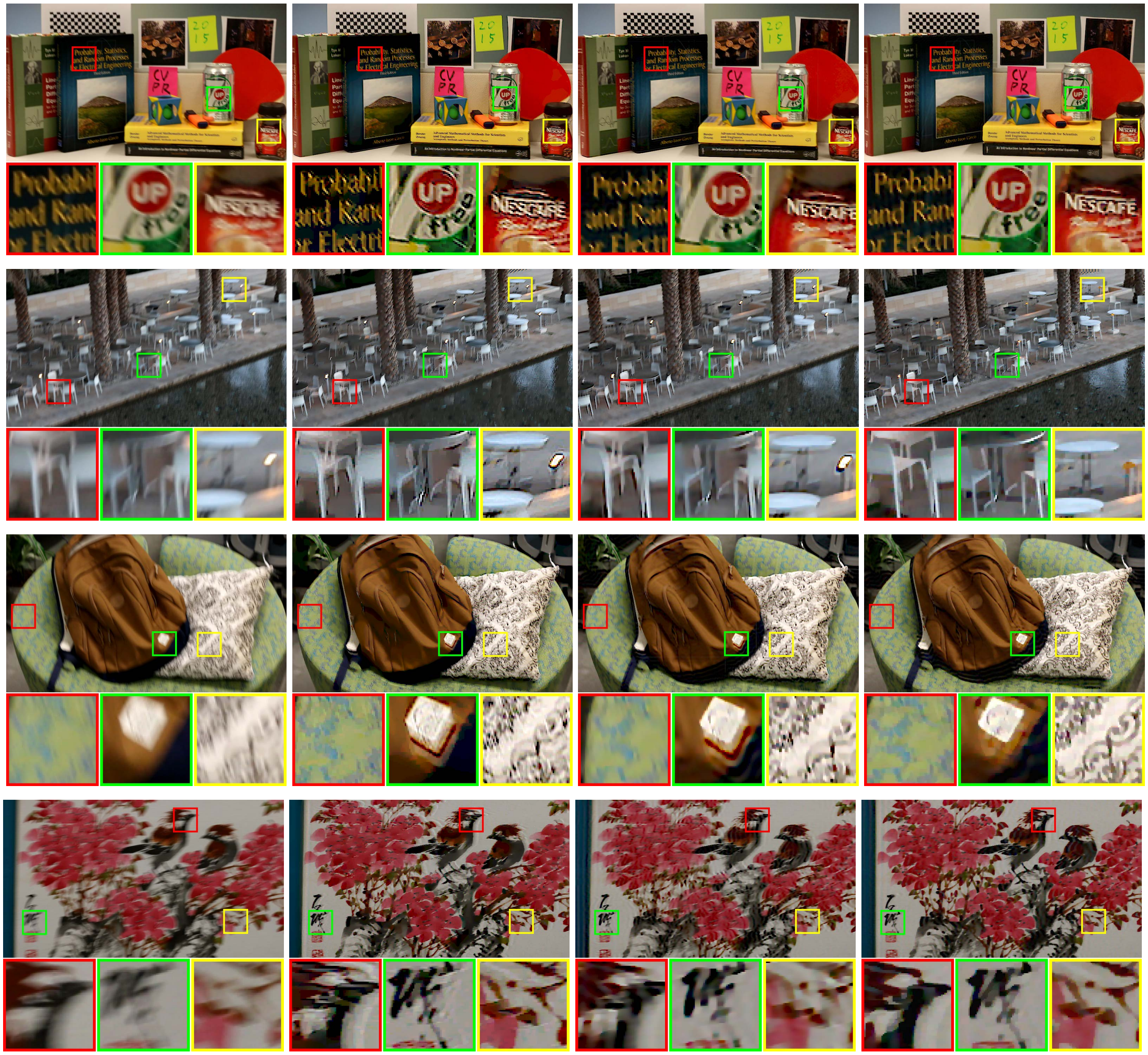
5.3. Parameters and Computational Performance

On an 8GB, 4 core computer our un-optimized MATLAB implementation takes about 1 minute for $\boldsymbol{\theta}$ initialization, 15 minutes for greyscale kernel estimation, and 2 minutes for the final deblurring on each channel of the 800×450 sized color image. The majority of the time is spent on the Jacobian matrix and sparse convolution matrix computations. We note that due to the spatial variation in the blur kernels we are inherently limited to methods that don't solve the deconvolution problem in the Fourier domain. This is an property not just of our solution, but of the RSMD problem in general.

In all of our experiments we set K as 30 which achieves good discretization-efficiency balance. μ is initialized as 1e-2, which decreases by $\tau = 0.7$ after each iteration until it reaches 1e-3 to output the final $\boldsymbol{\theta}$ estimation. The μ for computing the final color image is set as 7e-4. We provide an analysis of the GN and overall alternating optimization algorithms in the supplemental material.

6. Discussion and Future Work

We presented an approach for rolling shutter motion deblurring by alternating between optimizing the intrinsic image and optimizing the camera motion trajectory in pose space. One limitation of our work is that it depends on the imperfect blur kernel estimation from uniform deblurring at multiple regions for trajectory initialization. Another limitation is that non-negligible in-plane rotation commonly exist in wide angle images. In the future, it would be interesting to extend our method to this general case by fitting a full 3D rotational pose trajectory. The key technical chal-



(a) Blurred. (b) Cho *et al*. [6]. (c) Xu *et al*. [18]. (d) Ours.

Figure 8: Comparison of our methods with state-of-art (b) uniform [6] and (c) non-uniform blind deblurring [18] algorithms.

lenge here is to solve the initialization problem (Sec. 4.2.2) for kernels that are no longer shift invariant even within a horizontal region of the image, and how to backproject the resulting kernels into the 3D pose space.

We also realize that even though a polynomial function is sufficient to describe human camera shake within the exposure time of interest, RSMB also widely exists in images captured by drones, street view cars, *etc*., when the camera motion trajectory is likely to be more irregular. In these cases, a non-blind gyroscope assisted method could be a better choice.

Acknowledgements

This work was supported by an NSERC Engage grant. We thank Robin Swanson for the proofreading.

References

[1] S. Baker, E. Bennett, S. B. Kang, and R. Szeliski. Removing rolling shutter wobble. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2392–2399. IEEE, 2010. 2

[2] S. Baker and I. Matthews. Lucas-kanade 20 years on: A unifying framework. *International journal of computer vision*,

56(3):221–255, 2004. 3, 4

[3] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011. 5

[4] D. Bradley, B. Atcheson, I. Ihrke, and W. Heidrich. Synchronization and rolling shutter compensation for consumer video camera arrays. In *Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009. IEEE Computer Society Conference on*, pages 1–8. IEEE, 2009. 2, 7

[5] S. Cho, H. Cho, Y.-W. Tai, Y. S. Moon, J. Cho, S. Lee, and S. Lee. Lucas-kanade image registration using camera parameters. volume 8301, pages 83010V–83010V–7, 2012. 3

[6] S. Cho and S. Lee. Fast motion deblurring. In *ACM Transactions on Graphics (TOG)*, volume 28, page 145. ACM, 2009. 1, 6, 7, 8

[7] A. Gupta, N. Joshi, C. L. Zitnick, M. Cohen, and B. Curless. Single image deblurring using motion density functions. In *Computer Vision–ECCV 2010*, pages 171–184. Springer, 2010. 1, 3, 4

[8] Z. Hu and M.-H. Yang. Fast non-uniform deblurring using constrained camera pose subspace. In *BMVC*, pages 1–11, 2012. 1, 2

[9] R. Köhler, M. Hirsch, B. Mohler, B. Schölkopf, and S. Harmeling. Recording and playback of camera shake: Benchmarking blind deconvolution with a real-world database. In *Computer Vision–ECCV 2012*, pages 27–40. Springer, 2012. 1, 2, 3, 6, 7

[10] M. Meilland, T. Drummond, and A. I. Comport. A unified rolling shutter and motion blur model for 3d visual registration. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 2016–2023. IEEE, 2013. 2

[11] S. H. Park and M. Levoy. Gyro-based multi-image deconvolution for removing handshake blur. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 1

[12] D. Perrone and P. Favaro. Total variation blind deconvolution: The devil is in the details. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 1, 4

[13] V. Pichaikuppan, R. Narayanan, and A. Rangarajan. Change detection in the presence of motion blur and rolling shutter effect. In *Computer Vision ECCV 2014*, pages 123–137. Springer International Publishing, 2014. 2

[14] O. Saurer, K. Koser, J.-Y. Bouguet, and M. Pollefeys. Rolling shutter stereo. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 465–472. IEEE, 2013. 2

[15] Y.-W. Tai, P. Tan, and M. S. Brown. Richardson-lucy deblurring for scenes under a projective motion path. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2011. 1

[16] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce. Non-uniform deblurring for shaken images. *International journal of computer vision*, 98(2):168–186, 2012. 1

[17] L. Xu and J. Jia. Two-phase kernel estimation for robust motion deblurring. In *Computer Vision–ECCV 2010*, pages 157–170. Springer, 2010. 1, 7

[18] L. Xu, S. Zheng, and J. Jia. Unnatural l0 sparse representation for natural image deblurring. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1107–1114. IEEE, 2013. 1, 4, 6, 7, 8