Change Detection in the Presence of Motion Blur and Rolling Shutter Effect

A.P. Vijay Rengarajan, A.N. Rajagopalan, and R. Aravind

Department of Electrical Engineering, Indian Institute of Technology Madras

Change Detection in the Presence of Motion Blur and Rolling Shutter Effect

Vijay Rengarajan A.P., Rajagopalan A.N., Aravind Rangarajan

Department of Electrical Engineering
Indian Institute of Technology Madras, Chennai 600036, India
{ee11d035,raju,aravind}@ee.iitm.ac.in

Abstract. The coalesced presence of motion blur and rolling shutter effect is unavoidable due to the sequential exposure of sensor rows in CMOS cameras. We address the problem of detecting changes in an image affected by motion blur and rolling shutter artifacts with respect to a reference image. Our framework bundles modelling of motion blur in global shutter and rolling shutter cameras into a single entity. We leverage the sparsity of the camera trajectory in the pose space and the sparsity of occlusion in spatial domain to propose an optimization problem that not only registers the reference image to the observed distorted image but detects occlusions as well, both within a single framework.

1 Introduction

Change detection in images is a highly researched topic in image processing and computer vision due to its ubiquitous use in a wide range of areas including surveillance, tracking, driver assistance systems and remote sensing. The goal of change detection is to identify regions of difference between a pair of images. Seemingly a straightforward problem at first look, there are many challenges due to sensor noise, illumination changes, motion, and atmosphere distortions. A survey of various change detection approaches can be found in Radke et al. [14]. One of the main problems that arises in change detection is the presence of motion blur. It is unavoidable due to camera shake during a long exposure especially when a lowly lit scene is being captured. The same is also true if the capturing mechanism itself is moving, for example in drone surveillance systems.

In the presence of motion blur, traditional feature-based registration and occlusion detection methods cannot be used due to photometric inconsistencies as pointed out by Yuan et al. [23]. It is possible to obtain a sharp image from the blurred observation through many of the available deblurring methods before sending to the change detection pipeline. Non-uniform deblurring works, which employ homography-based blur model, include that of Gupta et al. [6], Whyte et al. [20], Joshi et al. [8], Tai et al. [18] and Hu et al. [7]. Paramanand and Rajagopalan [12] estimate camera motion due to motion blur and the depth map of static scenes using a blurred/unblurred image pair. Cho et al. [3] estimate homographies in the motion blur model posed as a set of image registration problems.

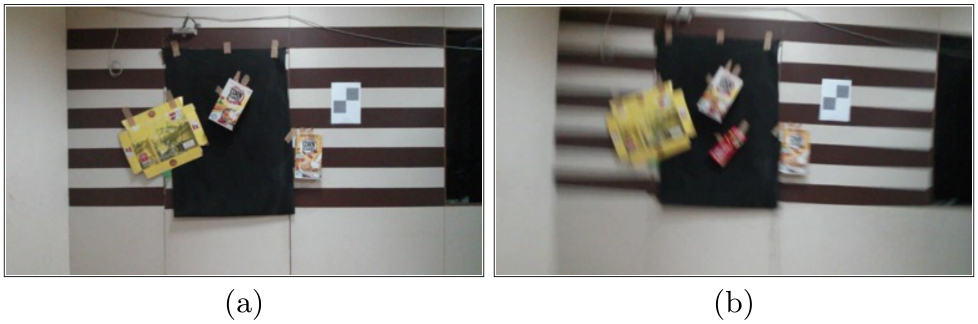


Fig. 1. (a) Reference image with no camera motion, (b) Distorted image with rolling shutter and motion blur artifacts.

A filter flow problem computing a space-variant linear filter that encompasses a wide range of tranformations including blur, radial distortion, stereo and optical flow is developed by Seitz and Baker [16]. Wu et al. [22] develop a sparse approximation framework to solve the target tracking problem in the presence of blur.

Contemporary CMOS sensors employ an electronic rolling shutter (RS) in which the horizontal rows of the sensor array are scanned at different times. This behaviour results in deformations when capturing dynamic scenes and when imaging from moving cameras. One can observe that the horizontal and vertical lines in Fig. 1(a) have become curved in Fig. 1(b). The very study of RS cameras is a growing research area. Ait-Aider et al. [1] compute the instantaneous pose and velocity of an object captured using an RS camera assuming known 2D-3D point correspondences. Liang et al. [9] rectify the RS effect between successive frames in a video by estimating a global motion and then interpolating motion for every row using a Bézier curve. Cho et al. [4] model the motion as an affine change with respect to row index. Baker et al. [2] remove the RS wobble from a video by posing it as a temporal super-resolution problem. Ringaby and Forssén [15] model the 3D rotation of the camera as a continuous curve to rectify and stabilise video from RS cameras. Grundmann et al. [5] have proposed an algorithm based on homography mixtures to remove RS effect from streaming uncalibrated videos. All these papers consider only the presence of RS deformations and the motion blur is assumed to be negligible. They typically follow a feature-based approach to rectify the effect between adjacent frames of a video.

In reality, it is apparent that both rolling shutter and motion blur issues will be present due to non-negligible exposure time. Fig. 1(b) exhibits geometric distortion due to rolling shutter effect and photometric distortion due to motion blur. Hence it is imperative to consider both the effects together in the image formation model. Meilland et al. [11] formulate a unified approach to estimate both rolling shutter and motion blur, but assume uniform velocity of the camera across the image. They follow a dense approach of minimisation of intensity errors to estimate camera motion between two consecutive frames of a video. In this paper, we remove the assumption of uniform camera velocity, and propose a general model that combines rolling shutter and motion blur effects. In the application of change detection, it is customary to rectify the observed image first and then to detect the occluded regions. Instead of following this rectify-difference pipeline, we follow a distort-difference pipeline, in which we first distort the reference image to register it with the observation followed by change detection. In the presence of motion blur, this pipeline has been shown to be simple and effective by Vageeswaran et al. [19] in face recognition and by Punnappurath et al. [13] for the application of image registration in blur. We assume that the reference image is free from blur and rolling-shutter artifacts as is often the case in aerial imagery, where the reference is captured beforehand under conducive conditions. Throughout this paper, we consider the scene to be sufficiently far away from the camera so that planarity can be invoked.

Our main contributions in this paper are:

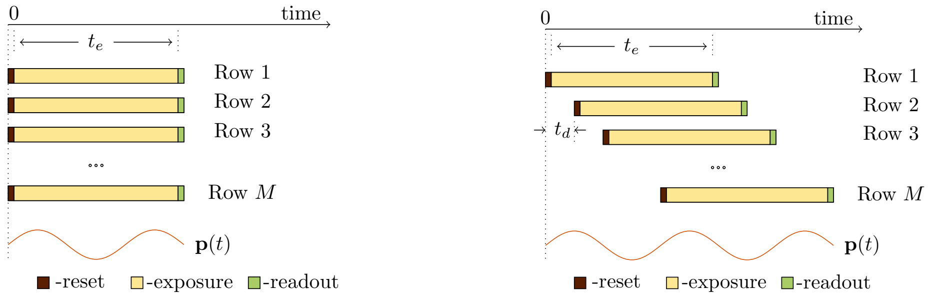
- To the best of our knowledge, the work described in this paper is the first of its kind to perform registration between a reference image and an image captured at a later time but distorted with *both* rolling shutter and motion blur artifacts, and to also simultaneously detect occlusions in the distorted image, all within a *single* framework. We thus efficiently account for both geometric and photometric distortions under one roof.
- Unlike existing works, we do not assume uniform velocity of camera motion during image exposure. Instead, we pose an optimisation problem with sparsity and partial non-negativity constraints to solve simultaneously for camera motion and occlusion for each row in the image.

2 Motion Blur in RS Cameras

In this section, we first explain the working of rolling shutter mechanism followed by a description of our combined motion blur and rolling shutter model.

Fig. 2 shows the mechanism by which sensors are exposed in RS and global shutter (GS) cameras. A GS camera exposes all the pixels at the same time. Fig. 2(a) illustrates this operation by showing same start and end exposure times for each row of the sensor array. The rows of an RS camera sensor array, on the other hand, are not exposed simultaneously. Instead, the exposure of consecutive rows starts sequentially with a delay as shown in Fig. 2(b), where t_e represents the exposure time of a single row and t_d represents the inter-row exposure delay with $t_d < t_e$. Both these values are same for all rows during image capture. The sequential capture causes the vertical line in the left of Fig. 1(a) to get displaced by different amounts in different rows due to camera motion which results in a curved line in Fig. 1(b). We will ignore the reset and read-out times in this discussion.

We now explain our *combined* motion blur and rolling shutter model. Let the number of rows of the image captured be M. Assuming the exposure starts at $t = 0$ for the first row, the ith row of the image is exposed during the time interval $[(i-1)t_d, (i-1)t_d + t_e]$. The total exposure time of the image T_e is $(M-1)t_d + t_e$. Thus the camera path observed by each row in their exposure times is unique. If the camera moves according to $\mathbf{p}(t)$ for $0 \leq t \leq T_e$, then the



(a) Global shutter

(b) Rolling shutter

Fig. 2. Exposure mechanism of global shutter and rolling shutter cameras.

ith row is blinded to the whole time except for $(i-1)t_d \leq t \leq (i-1)t_d + t_e$. Here $\mathbf{p}(t)$ is a vector with six degrees of freedom corresponding to 3D camera translations and 3D camera rotations. Let $\mathbf{f}$ and $\mathbf{g}$ represent respectively, the images captured by the RS camera without and with camera motion. We denote the ith row of any image with a superscript (i). Each row of $\mathbf{g}$ is an averaged version of the corresponding rows in warped versions of $\mathbf{f}$ due to the camera motion in its exposure period. We have

$$\mathbf{g}^{(i)} = \frac{1}{t_e} \int_{(i-1)t_d}^{(i-1)t_d+t_e} \mathbf{f}^{(i)}_{\mathbf{p}(t)}\, dt, \text{ for } i = 1 \text{ to } M, \tag{1}$$

where $\mathbf{f}^{(i)}_{\mathbf{p}(t)}$ is the ith row of the warped version of $\mathbf{f}$ due to the camera pose $\mathbf{p}(t)$ at a particular time t.

We discretise this model of combined rolling shutter and motion blur in (1) with respect to a finite camera pose space $\mathcal{S}$. We assume that the camera can undergo only a finite set of poses during the total exposure time, and this is represented by $\mathcal{S} = \{\boldsymbol{\tau}_k\}_{k=1}^{|\mathcal{S}|}$. Hence we write (1) equivalently as,

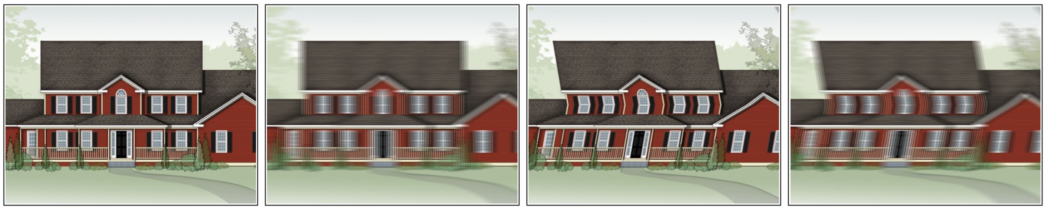
$$\mathbf{g}^{(i)} = \sum_{\boldsymbol{\tau}_k \in \mathcal{S}} \omega^{(i)}_{\boldsymbol{\tau}_k}\, \mathbf{f}^{(i)}_{\boldsymbol{\tau}_k} \tag{2}$$

where $\mathbf{f}^{(i)}_{\boldsymbol{\tau}_k}$ is the ith row of the warped reference image $\mathbf{f}_{\boldsymbol{\tau}_k}$ due to camera pose $\boldsymbol{\tau}_k$. Pose weight $\omega^{(i)}_{\boldsymbol{\tau}_k}$ denotes the fraction of exposure time t_e, that the camera has spent in the pose $\boldsymbol{\tau}_k$ during the exposure of ith row. Since the pose weights represent time, we have $\omega_{\boldsymbol{\tau}_k} \geq 0$ for all $\boldsymbol{\tau}_k$. When the exposure times of $\mathbf{f}^{(i)}$ and $\mathbf{g}^{(i)}$ are same, then by conservation of energy, we have $\sum_{\boldsymbol{\tau}_k \in \mathcal{S}} \omega^{(i)}_{\boldsymbol{\tau}_k} = 1$ for each i. In this paper, we follow a projective homography model for planar scenes [6, 8, 20, 7, 12]. We denote camera translations and rotations by $(\mathbf{T}_k, \mathbf{R}_k)$ and the corresponding motion in the image plane by $(\mathbf{t}_k, \mathbf{r}_k)$.

In fact, our model is general enough that it encompasses both GS and RS camera acquisition mechanisms with and without motion blur (MB) as shown in Table 1. Here $\boldsymbol{\omega}^{(i)}$ is the pose weight vector of the ith row with each of its elements $\omega^{(i)}_{\boldsymbol{\tau}_k}$ representing a number between 0 and 1, which is the weight for the $\boldsymbol{\tau}_k$th pose in the ith row. Fig. 3 showcases images with different types of distortions.

Table 1. Generalised motion blur model for GS and RS cameras

Type	Inter-row delay	Pose weight vector ($1 \leq i \leq M$)
GS	$t_d = 0$	$\omega_{\boldsymbol{\tau}_k}^{(i)} = \begin{cases} 1 & \text{for } k = k_0 \\ 0 & \text{otherwise} \end{cases}$ where k_0 is independent of i
GS+MB	$t_d = 0$	Same $\boldsymbol{\omega}^{(i)}$ for all i
RS	$t_d \neq 0$	$\omega_{\boldsymbol{\tau}_k}^{(i)} = \begin{cases} 1 & \text{for } k = k_i \\ 0 & \text{otherwise} \end{cases}$
RS+MB	$t_d \neq 0$	Different $\boldsymbol{\omega}^{(i)}$ for each i



GS GS+MB RS RS+MB

Fig. 3. Various types of distortions as listed in Table 1.

3 Image Registration and Occlusion Detection

Given the reference image and the distorted image affected by rolling shutter and motion blur (denoted by RSMB from now on) with occlusions, we simultaneously register the reference image with the observed image and detect occlusions present in the distorted image.

Let us first consider the scenario of registering the reference image to the RSMB image without occlusions. We can represent the rows of the RSMB image as linear combinations of elements in a dictionary formed from the reference image. The relationship between them as matrix-vector multiplication from (2) is given by

$$\mathbf{g}^{(i)} = \mathbf{F}^{(i)} \boldsymbol{\omega}^{(i)} \quad i = 1, 2, \ldots, M, \tag{3}$$

where $\mathbf{g}^{(i)} \in \mathbb{R}^{N \times 1}$ is the ith row of the RSMB image stacked as a column vector and N is the width of RSMB and reference images. Each column of $\mathbf{F}^{(i)} \in \mathbb{R}^{N \times |\mathcal{S}|}$ contains the ith row of a warped version of the reference image $\mathbf{f}$, for a pose $\boldsymbol{\tau}_k \in \mathcal{S}$, where $\mathcal{S}$ is the discrete pose space we define, and $|\mathcal{S}|$ is the number of poses in it. Solving for the column vector $\boldsymbol{\omega}^{(i)}$ amounts to registering every row of the reference image with the distorted image.

In the presence of occlusion, the camera observes a distorted image of the clean scene with occluded objects. We model the occlusion as an additive term to the observed image $\mathbf{g}$ (Wright et al. [21]), as $\mathbf{g}_{\text{occ}}^{(i)} = \mathbf{g}^{(i)} + \boldsymbol{\chi}^{(i)}$, where $\mathbf{g}_{\text{occ}}^{(i)}$ is the ith row of the RSMB image with occlusions, $\boldsymbol{\chi}^{(i)}$ is the occlusion vector which contains non-zero values in its elements where there are changes in $\mathbf{g}_{\text{occ}}^{(i)}$ compared to $\mathbf{g}^{(i)}$. Since the occluded pixels can have intensities greater or less than the original intensities, $\boldsymbol{\chi}^{(i)}$ can take both positive and negative values. We compactly write this using a combined dictionary $\mathbf{B}^{(i)}$ as

$$\mathbf{g}_{\text{occ}}^{(i)} = \left[\mathbf{F}^{(i)}\ \mathbf{I}_N\right] \begin{bmatrix} \boldsymbol{\omega}^{(i)} \\ \boldsymbol{\chi}^{(i)} \end{bmatrix} = \mathbf{B}^{(i)}\boldsymbol{\xi}^{(i)}, \quad i = 1, 2, \ldots, M. \tag{4}$$

Here $\mathbf{I}_N$ is an $N \times N$ identity matrix, $\mathbf{B}^{(i)} \in \mathbb{R}^{N\times(|\mathcal{S}|+N)}$ and $\boldsymbol{\xi}^{(i)} \in \mathbb{R}^{(|\mathcal{S}|+N)}$. We can consider the formulation in (4) as a representation of the rows of the RSMB image in a two-part dictionary, the first part being the set of projective transformations to account for the motion blur and the second part accounting for occlusions.

To solve for $\boldsymbol{\omega}^{(i)}$ and $\boldsymbol{\chi}^{(i)}$ is a data separation problem in the spirit of morphological component analysis (Starck et al. [17]). To solve the under-determined system in (4), we impose priors on pose and occlusion weights leveraging their sparseness. We thus formulate and solve the following optimisation problem to arrive at the desired solution.

$$\widetilde{\boldsymbol{\xi}}^{(i)} = \arg\min_{\boldsymbol{\xi}^{(i)}} \left\{ \|\mathbf{g}_{\text{occ}}^{(i)} - \mathbf{B}^{(i)}\boldsymbol{\xi}^{(i)}\|_2^2 + \lambda_1\|\boldsymbol{\omega}^{(i)}\|_1 + \lambda_2\|\boldsymbol{\chi}^{(i)}\|_1 \right\} \tag{5}$$
$$\text{subject to } \boldsymbol{\omega}^{(i)} \succeq 0$$

where λ_1 and λ_2 are non-negative regularisation parameters and $\succeq$ denotes non-negativity of each element of the vector. ℓ_1-constraints impose sparsity on camera trajectory and occlusion vectors by observing that (i) camera can move only so much in the whole space of 6D camera poses, and (ii) occlusion is sparse in all rows in spatial domain. To enforce different sparsity levels on camera motion and occlusion, we use two ℓ_1 regularisation parameters λ_1 and λ_2 with different values. We also enforce non-negativity for the pose weight vector $\boldsymbol{\omega}^{(i)}$. Our formulation elegantly imposes non-negativity only on the pose weight vector.

An equivalent formulation of (5) and its illustration is shown in Fig. 4. We modify the *nnLeastR* function provided in the SLEP package (Liu et al. [10]) to account for the partial non-negativity of $\boldsymbol{\xi}^{(i)}$ and solve (5). Observe that when $\boldsymbol{\xi}^{(i)} = \boldsymbol{\omega}^{(i)}$ and $\mathbf{B}^{(i)} = \mathbf{F}^{(i)}$, (5) reduces to the problem of image registration in the presence of blur.

In our model, the static occluder is elegantly subsumed in the reference image $\mathbf{f}$. It is possible to obtain the exact occlusion mask in $\mathbf{f}$ (instead of the blurred occluder region) as a forward problem, by inferring which pixels in $\mathbf{f}$ contribute to the blurred occlusion mask in $\mathbf{g}$, since the pose space weights $\boldsymbol{\omega}$ of the camera motion are known. Our framework is general, and it can detect occluding objects in the observed image as well as in the reference image (which are missing in the observed image). Yet another important benefit of adding the occlusion vector to the observed image is that it enables detection of even independently moving objects.

3.1 Dynamically Varying Pose Space

Building $\{\mathbf{F}^{(i)}\}_{i=1}^{M}$ in (5) is a crucial step in our algorithm. If the size of the pose space $\mathcal{S}$ is too large, then storing this matrix requires considerable memory

Equivalent formulation of (5):

$$\min_{\boldsymbol{\xi}^{(i)}} \left\{ \lambda_1 \|\widehat{\boldsymbol{\xi}}^{(i)}\|_1 \right\}$$

$$\text{subject to} \begin{cases} \|\mathbf{g}^{(i)}_{\text{occ}} - \widehat{\mathbf{B}}^{(i)}\widehat{\boldsymbol{\xi}}^{(i)}\|_2^2 \le \epsilon, \\ \mathbf{C}\widehat{\boldsymbol{\xi}}^{(i)} \succeq 0, \end{cases}$$

$$\text{where } \widehat{\mathbf{B}}^{(i)} = \begin{bmatrix} \mathbf{F}^{(i)} \frac{\lambda_1}{\lambda_2}\mathbf{I}_N \end{bmatrix},$$

$$\mathbf{C} = \begin{bmatrix} \mathbf{I}_{|\mathcal{S}|} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \text{ and } \widehat{\boldsymbol{\xi}} = \begin{bmatrix} \boldsymbol{\omega} \\ \frac{\lambda_2}{\lambda_1}\boldsymbol{\chi} \end{bmatrix}.$$

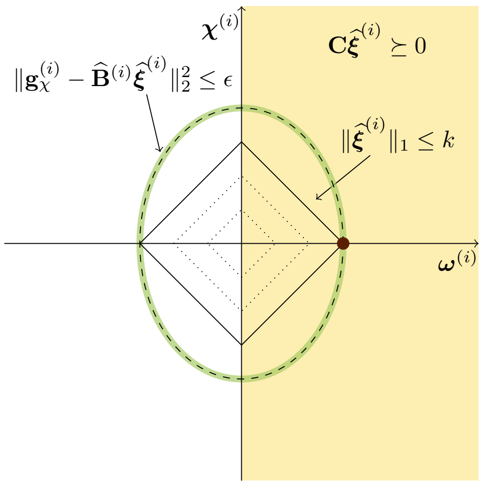

$\boldsymbol{\chi}^{(i)}$
$\mathbf{C}\widehat{\boldsymbol{\xi}}^{(i)} \succeq 0$
$\|\mathbf{g}^{(i)}_{\chi} - \widehat{\mathbf{B}}^{(i)}\widehat{\boldsymbol{\xi}}^{(i)}\|_2^2 \le \epsilon$
$\|\widehat{\boldsymbol{\xi}}^{(i)}\|_1 \le k$
$\boldsymbol{\omega}^{(i)}$

Fig. 4. Illustration of the constraints in our optimisation framework in two dimensions.

and solving the optimisation problem becomes computationally expensive. We also leverage the continuity of camera motion in the pose space. We note the fact that the camera poses that a row observes during its exposure time will be in the neighbourhood of that of its previous row, and so we dynamically vary the search space for every row. While solving (5) for the ith row, we build $\mathbf{F}^{(i)}$ on-the-fly for the restricted pose space which is exclusive to each row.

Let $N(\boldsymbol{\tau}, \mathbf{b}, \mathbf{s}) = \{\boldsymbol{\tau} + q\mathbf{s} : \boldsymbol{\tau} - \mathbf{b} \preceq \boldsymbol{\tau} + q\mathbf{s} \preceq \boldsymbol{\tau} + \mathbf{b}, q \in \mathbb{Z}\}$ denote the neighbourhood of poses around a particular 6D pose vector $\boldsymbol{\tau}$, where $\mathbf{b}$ is the bound around the pose vector and $\mathbf{s}$ is the step-size vector. We start by solving (5) for the middle row $M/2$. Since there is no prior information about the camera poses during the time of exposure of the middle row, we assume a large pose space around the origin (zero translations and rotations), i.e. $\mathcal{S}^{(M/2)} = N(\mathbf{0}, \mathbf{b}_0, \mathbf{s}_0)$ where $\mathbf{b}_0$ and $\mathbf{s}_0$ are the bound and the step-size for the middle row, respectively. We build the matrix $\mathbf{F}^{(M/2)}$ based on this pose space. We start with the middle row since there is a possibility that the first and last rows of the RSMB image may contain new information and may result in a wrong estimate of the weight vector. Then we proceed as follows: for any row $i < M/2-1$, we build the matrix $\mathbf{F}^{(i)}$ only for the neighbourhood $N(\boldsymbol{\tau}_c^{(i+1)}, \mathbf{b}, \mathbf{s})$, and for any row $i > M/2+1$, we use only the neighbourhood $N(\boldsymbol{\tau}_c^{(i-1)}, \mathbf{b}, \mathbf{s})$ where $\boldsymbol{\tau}_c^{(i)}$ is the centroid pose of the ith row, which is given by

$$\boldsymbol{\tau}_c^{(i)} = \frac{\sum_{\boldsymbol{\tau}_k} \omega^{(i)}_{\boldsymbol{\tau}_k} \boldsymbol{\tau}_k}{\sum_{\boldsymbol{\tau}_k} \omega^{(i)}_{\boldsymbol{\tau}_k}}. \tag{6}$$

4 Experimental Results

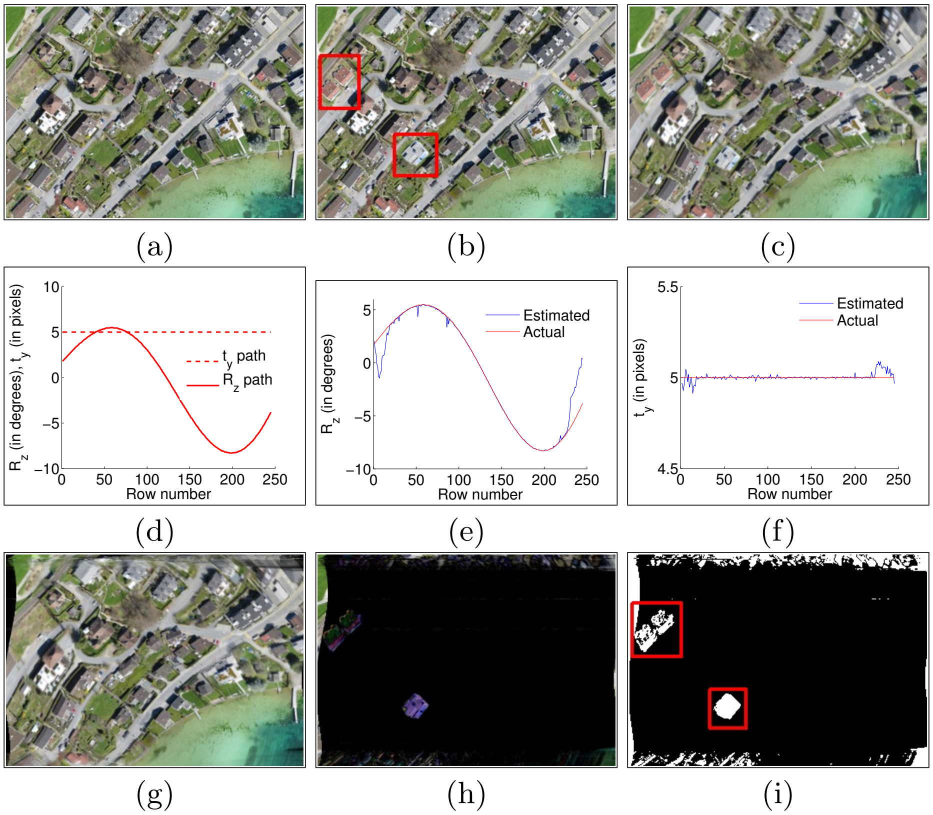
To evaluate the performance of our technique, we show results for both synthetic and real experiments. For synthetic experiments, we simulate the effect of RS and MB for a given camera path. We estimate the pose weight vector and occlusion vector for each row using the reference and RSMB images. We also compare the estimated camera motion trajectory with the actual one. Due to the unavailability of a standard database for images with both RS and MB effects, and in particular, for the application of change detection, we capture our own images for the real experiments. We use a hand-held Google Nexus 4 mobile phone camera to capture the desired images. The RS and MB effects are caused by intentional hand-shake.

4.1 Synthetic Experiments

The effect of RS and MB is simulated in the following manner. We generate a discrete path of camera poses of length $(M-1)\beta+\alpha$. To introduce motion blur in each row, we assign α consecutive poses in this discrete path. We generate the motion blurred row of the RSMB image by warping and averaging the row of the reference image according to these poses. Since the row index is synonymous with time, a generated camera path with continuously changing slope corresponds to non-uniform velocity of the camera. The RS effect is arrived by using different sets of α poses for each row along the camera path. For the ith row, we assign α consecutive poses with index from $(i-1)\beta+1$ to $(i-1)\beta+\alpha$ in the generated discrete camera path. Thus each row would see a unique set of α poses with β index delay with respect to the previous row. The centroid of poses corresponding to each row will act as the actual camera path against which our estimates are compared.

In the first experiment, we simulate a scenario where RS and MB degradations happen while imaging from an aerial vehicle. We first add occluders to the reference image (Compare Figs. 5(a) and (b)). The images have 245 rows and 345 columns. While imaging a geographical region from a drone, RS effect is unavoidable due to the motion of the vehicle itself. Especially it is difficult to maintain a straight path while controlling the vehicle. Any drift in the flying direction results in in-plane rotations in the image. We introduce different sets of in-plane rotation angles to each row of the image to emulate flight drifts. We generate a camera motion path with non-uniform camera velocity for in-plane rotation R_z. We use $\alpha=20$ and $\beta=3$ while assigning multiple poses to each row as discussed earlier. The centroid of R_z poses for each row is shown as a continuous red line in Fig. 5(d) which is the actual camera path. Geometrical misalignment between the reference and RSMB images in the flying direction (vertical axis) is added as a global t_y shift which is shown as a dotted red line in Fig. 5(d). The RSMB image thus generated is shown in Fig. 5(c). Though we generate a sinusoidal camera path in the experiment, its functional form is not used in our algorithm.

We need to solve (5) to arrive at the registered reference and occlusion images. Since there is no prior information about possible camera poses, we assume a large initial pose space around the origin while solving for the middle row: x-translation $t_x = N(0,10,1)$ pixels, y-translation $t_y = N(0,10,1)$ pixels, scale $t_z = N(1,0.1,0.1)$, rotations $R_x = N(0,2,1)^\circ$, $R_y = N(0,2,1)^\circ$ and $R_z = N(0,8,1)^\circ$. The columns of $\mathbf{F}^{(M/2)}$ contain the middle rows of the



(a) (b) (c)

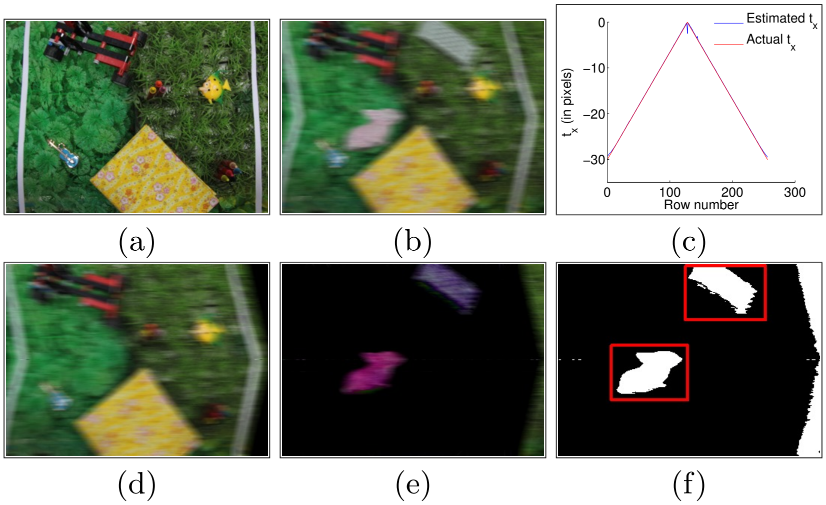
(d) (e) (f)

(g) (h) (i)

Fig. 5. (a) Reference image with no camera motion, (b) Reference image with added occlusions, (c) RSMB image, (d) Simulated camera path, (e) Estimated R_z camera path (blue) overlaid on simulated camera path (red), (f) Estimated t_y camera path (blue) overlaid on simulated camera path (red), (g) Registered reference image, (h) Occlusion image, and (i) Thresholded occlusion image.

warps of the reference image $\mathbf{f}$ for all these pose combinations. For the remaining rows, the search neighbourhood is chosen around the centroid pose of its neighbouring row. Since the camera would move only so much between successive rows, we choose a relatively smaller neighbourhood: $N(t_{cx}, 3, 1)$ pixels, $N(t_{cy}, 3, 1)$ pixels, $N(t_{cz}, 0.1, 0.1)$, $N(R_{cx}, 2, 1)^\circ$, $N(R_{cy}, 2, 1)^\circ$ and $N(R_{cz}, 2, 1)^\circ$. Here $[t_{cx}, t_{cy}, t_{cz}, R_{cx}, R_{cy}, R_{cz}]$ is the centroid pose vector of the neighbouring row as discussed in Section 3.1. Since we work in [0–255] intensity space, we use $255 \times \mathbf{I}_N$ in place of $\mathbf{I}_N$ in (4). The camera trajectory experienced by each row is very sparse in the whole pose space and hence we set a large λ_1 value of 5×10^3. We set $\lambda_2 = 10^3$ since the occlusion will be comparatively less sparse in each row, if present. We empirically found out that these values work very well for most images and different camera motions as well.

On solving (5) for each $1 \leq i \leq M$, we get the estimated pose weight vectors $\{\widetilde{\boldsymbol{\omega}}^{(i)}\}_{i=1}^{M}$ and occlusion vectors $\{\widetilde{\boldsymbol{\chi}}^{(i)}\}_{i=1}^{M}$. We form the registered reference image using $\{\mathbf{F}^{(i)}\widetilde{\boldsymbol{\omega}}^{(i)}\}_{i=1}^{M}$ and the occlusion image using $\{255\ \mathbf{I}_N\widetilde{\boldsymbol{\chi}}^{(i)}\}_{i=1}^{M}$. These are shown in Figs. 5(g) and (h), respectively. Fig. 5(i) shows the thresholded binary image with occlusion regions marked in red. The estimated camera trajectories for R_z and t_y are shown in Figs. 5(e) and (f). Note that the trajectories are



(a) (b) (c)

(d) (e) (f)

Fig. 6. (a) Reference image with no camera motion, (b) RSMB image, (c) Estimated t_x camera path (blue) overlaid on simulated camera path (red), (d) Registered reference image, (e) Occlusion image, and (f) Thresholded occlusion image.

correctly estimated by our algorithm. The presence of boundary regions in the occluded image is because of the new information, which are not in the reference image, coming in due to camera motion.

In the next experiment, we consider a scenario where there is heavy motion blur along with the RS effect. An image of a synthetic grass-cover with objects is shown in Fig. 6(a). After adding occluders, we distort the reference image to create an image which is heavily blurred with zig-zag horizontal translatory RS effect. The RSMB image is shown in Fig. 6(b). The camera path simulated is shown in Fig. 6(c) in red. The algorithm parameters are the same as that for the previous experiment. The two output components of our algorithm, the registered and occlusion images, are shown respectively in Figs. 6(d) and (e). Boxed regions in the thresholded image in Fig. 6(f) show the effectiveness of our framework. The estimated camera trajectory is shown in blue in Fig. 6(c). More synthetic examples are available at `http://www.ee.iitm.ac.in/ipcvlab/research/changersmb`.

4.2 Real Experiments

In the first scenario, the reference image is a scene with horizontal and vertical lines, and static objects as shown in Fig. 7(a). This is captured with a static camera. We then added an occluder to the scene. With the camera at approximately the same position, we recorded a video of the scene with free-hand camera motion. The purpose of capturing a video (instead of an image) is to enable comparisons with the state-of-the-art as will become evident subsequently. From the video, we extracted a frame with high RS and MB artifacts and this is shown in Fig. 7(b). Our algorithm takes only these *two images* as input. We perform geometric and photometric registration, and change detection simultaneously by solving (5). To register the middle row, we start with a large pose

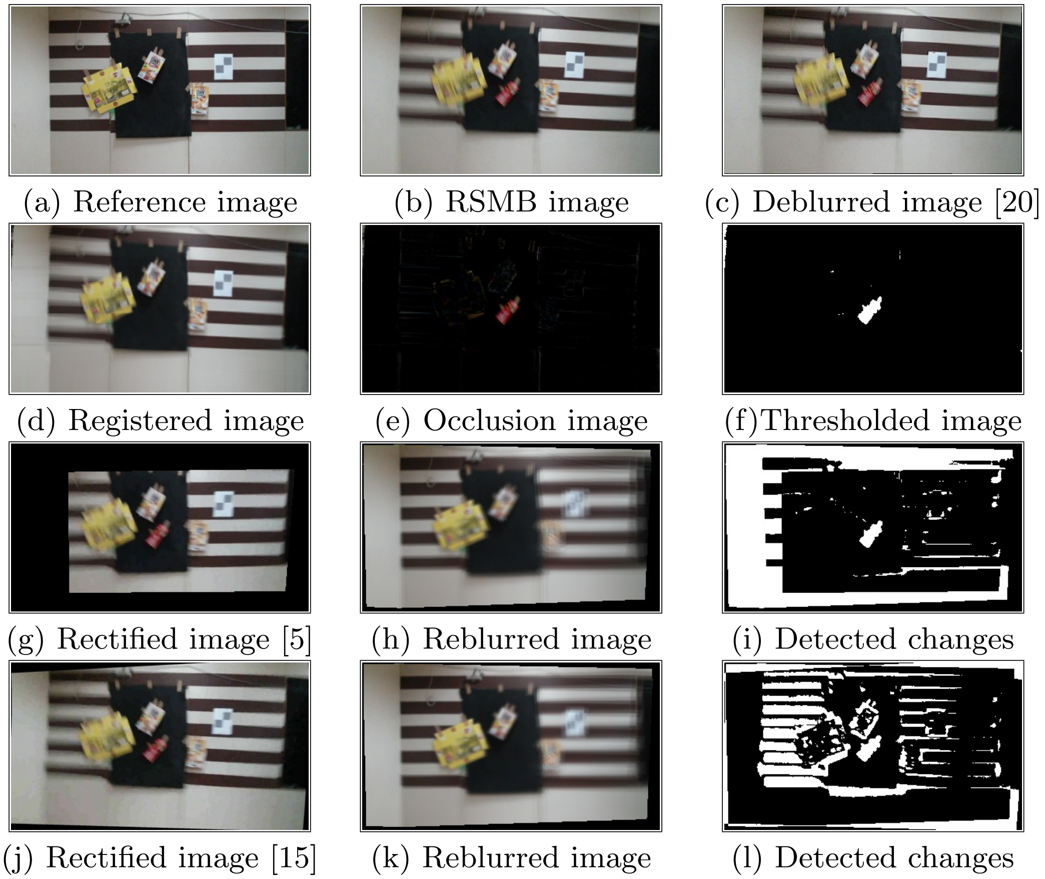

(a) Reference image
(b) RSMB image
(c) Deblurred image [20]
(d) Registered image
(e) Occlusion image
(f)Thresholded image
(g) Rectified image [5]
(h) Reblurred image
(i) Detected changes
(j) Rectified image [15]
(k) Reblurred image
(l) Detected changes

Fig. 7. (a)-(b): Reference and RSMB images (inputs to our algorithm), (c): RSMB image deblurred using Whyte et al. [20], (d)-(f): Proposed method using combined RS and MB model, (g)-(i): Rectify RS effect from video using Grundmann et al. [5], then estimate the kernel [20] and reblur the reference image, and detect changes, (j)-(l) Rectify-blur estimation pipeline using Ringaby and Forssén [15].

space: $t_x, t_y = N(0, 8, 1)$ pixels, $t_z = N(1, 0.1, 0.1)$, $R_x, R_y = N(0, 6, 1)^\circ$, and $R_z = N(0, 10, 1)^\circ$. The regularization parameters are kept the same as used for synthetic experiments. The relatively smaller pose space adaptively chosen for other rows is: $N(t_{cx}, 3, 1)$ pixels, $N(t_{cy}, 3, 1)$ pixels, $N(t_{cz}, 0.1, 0.1)$, $N(R_{cx}, 1, 1)^\circ$, $N(R_{cy}, 1, 1)^\circ$ and $N(R_{cz}, 1, 1)^\circ$. The registered reference image is shown in Fig. 7(d). The straight lines of the reference image are correctly registered as curved lines since we are forward warping the reference image by incorporating RS. The presence of motion blur is also to be noted. This elegantly accounts for both geometric and photometric distortions during registration. Figs. 7(e) and (f) show the occlusion image and its thresholded version respectively.

We compare our algorithm with a serial framework which will rectify the RS effect and account for MB independently. We use the state-of-the-art method of Whyte et al. [20] for non-uniform motion blur estimation, and recent works of Grundmann et al. [5] and Ringaby and Forssén [15] for RS rectification. Since the code of the combined RS and MB approach by Meilland et al. [11] hasn't been shared with us, we are unable to compare our algorithm with their method.

The RSMB image is first deblurred using the method of Whyte et al. The resulting deblurred image is shown in Fig. 7(c). We can clearly observe that the deblurring effort itself has been unsuccessful. This is because the traditional motion blur model considers a single global camera motion trajectory for all the pixels. But in our case, each row of the RSMB image experiences a different camera trajectory, and hence there is no surprise that deblurring does not work.

Due to the failure of non-uniform deblurring on the RSMB image, we consider the task of first rectifying the RS effect followed by MB kernel estimation. Since the RS rectification methods of Grundmann et al. and Ringaby and Forssén are meant for videos, to let the comparison be fair, we provide the captured video with occlusion as input to their algorithms. We thus have in hand now, an RS rectified version of the video. The rectified frames using these two algorithms corresponding to the RSMB image we had used in our algorithm are shown in Figs. 7(g) and (j).

We now estimate the global camera motion of the rectified images using the non-uniform deblurring method. While performing change detection, to be consistent with our algorithm, we follow the reblur-difference pipeline instead of the deblur-difference pipeline. We apply the estimated camera motion from the rectified frame on the reference image, and detect the changes with respect to the rectified frame. These reblurred images are shown in Figs. 7(h) and (k). Note that from Figs. 7(i) and (l), the performance of occlusion detection is much worse than our algorithm. The number of false positives is high as can be observed near the horizontal edges in Fig. 7(i). Though the RS rectification of Grundmann et al. works reasonably well to stabilise the video, the rectified video is not equivalent to a global shutter video especially in the presence of motion blur. The camera motion with non-uniform velocity renders invalid the notion of having a global non-uniform blur kernel. The RS rectification of Ringaby et al. is worse than that of Grundmann et al., and hence the change detection suffers heavily as shown in Fig. 7(l). Hence it is amply evident that the state-of-the-art algorithms cannot handle these two effects together, and that an integrated approach is indispensable. To further confirm the efficacy of our method, we show more results.

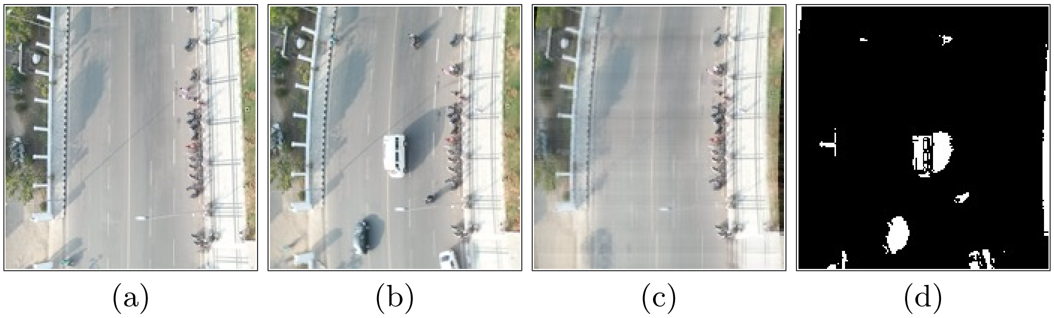


Fig. 8. (a) Reference image with no camera motion, (b) RSMB image with prominent curves due to y-axis camera rotation, (c) Reference image registered to RSMB image, and (d) Thresholded occlusion image.

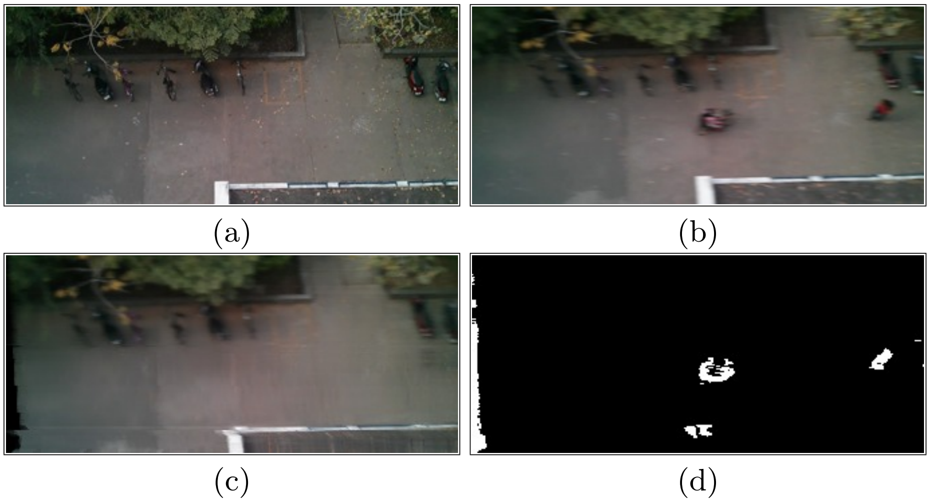


Fig. 9. (a) Reference image, (b) RSMB image, (c) Registered image, and (d) Thresholded occlusion image.

In the next example, we capture an image from atop a tall building looking down at the road below. The reference image in Fig. 8(a) shows straight painted lines and straight borders of the road. The RSMB image is captured by rotating the mobile phone camera prominently around the y-axis (vertical axis). This renders the straight lines curved as shown in Fig. 8(b). Our algorithm works quite well to register the reference image with the RSMB image as shown in Fig. 8(c). The occluding objects, both the big vehicles and smaller ones, have been detected correctly as shown in Fig. 8(d). We do note here that one of the small white columns along the left edge of the road in the row where the big van runs, is detected as a false occlusion.

Figs. 9(a) and (b) show respectively, the reference image and the distorted image with prominent horizontal RS and MB effects. Figs. 9(c) and (d) show our registered and thresholded occlusion images, respectively. We can observe that the shear effect due to RS mechanism is duly taken care of in registration and the occluding objects are also correctly detected. The parapet in the bottom right of the image violates our planar assumption and hence its corner shows up wrongly as an occlusion.

4.3 Algorithm complexity and run-time

We use a gradient projection based approach to solve the ℓ_1-minimisation problem (5) using SLEP [10]. It requires a sparse matrix-vector multiplication with order less than $O(N(|\mathcal{S}| + N))$ and a projection onto a subspace with order $O(|\mathcal{S}| + N)$ in each iteration with convergence rate of $O(1/k^2)$ for the kth iteration. Here N is the number of columns and $|\mathcal{S}|$ is the cardinality of the pose space (which is higher for the middle row). Run-times for our algorithm using an unoptimised MATLAB code without any parallel programming on a 3.4GHz PC with 16GB RAM are shown in Table 2. We do note here that, since the motion

Table 2. Run-times of our algorithm for Figs. 5 to 9, with $t_{total}, t_{mid}, t_{other}$ representing total time, time for middle row, and average time for other rows respectively. All time values are in seconds.

Fig.	Rows × Cols	t_{total}	t_{mid}	t_{other}
5	245 × 345	712	28	2.8
6	256 × 350	746	30	2.8
7	216 × 384	644	29	2.9
8	167 × 175	404	34.5	2.2
9	147 × 337	317	29	2.0

blur estimation of rows in the top-half and bottom-half are independent, they can even be run in parallel.

The bounds of the camera pose space and the step sizes of rotations and translations used here, work well on various real images that we have tested. Step sizes are chosen such that the displacement of a point light source between two different warps is at least one pixel. Decreasing the step sizes further increases the complexity, but provides little improvement for practical scenarios. The large bounding values for the middle row used suffice for most real cases. However, for extreme viewpoint changes, those values can be increased further, if necessary. We have observed that the given regularisation values (λ_1 and λ_2) work uniformly well in all our experiments.

5 Conclusions

Increased usage of CMOS cameras forks an important branch of image formation model, namely the rolling shutter effect. The research challenge is escalated when the RS effect entwines with the traditional motion blur artifacts that have been extensively studied in the literature for GS cameras. The combined effect is thus an important issue to consider in change detection. We proposed an algorithm to perform change detection between a reference image and an image affected by rolling shutter as well as motion blur. Our model advances the state-of-the-art by elegantly subsuming both the effects within a single framework. We proposed a sparsity-based optimisation framework to arrive at the registered reference image and the occlusion image simultaneously. The utility of our method was adequately demonstrated on both synthetic and real data.

As future work, it would be interesting to consider the removal of both motion blur and rolling shutter artifacts given a single distorted image, along the lines of classical single image non-uniform motion deblurring algorithms.

References

1. Ait-Aider, O., Andreff, N., Lavest, J.M., Martinet, P.: Simultaneous object pose and velocity computation using a single view from a rolling shutter camera. In: Proc. ECCV. pp. 56–68 (2006)

2. Baker, S., Bennett, E., Kang, S.B., Szeliski, R.: Removing rolling shutter wobble. In: Proc. CVPR. pp. 2392–2399. IEEE (2010)
3. Cho, S., Cho, H., Tai, Y.W., Lee, S.: Registration based non-uniform motion deblurring. In: Computer Graphics Forum. vol. 31, pp. 2183–2192. Wiley Online Library (2012)
4. Cho, W.h., Kim, D.W., Hong, K.S.: CMOS digital image stabilization. IEEE Trans. Consumer Electronics 53(3), 979–986 (2007)
5. Grundmann, M., Kwatra, V., Castro, D., Essa, I.: Calibration-free rolling shutter removal. In: Proc. ICCP. pp. 1–8. IEEE (2012)
6. Gupta, A., Joshi, N., Zitnick, C.L., Cohen, M., Curless, B.: Single image deblurring using motion density functions. In: Proc. ECCV. pp. 171–184 (2010)
7. Hu, Z., Yang, M.H.: Fast non-uniform deblurring using constrained camera pose subspace. In: Proc. BMVC. pp. 1–11 (2012)
8. Joshi, N., Kang, S.B., Zitnick, C.L., Szeliski, R.: Image deblurring using inertial measurement sensors. ACM Trans. Graphics 29(4), 30 (2010)
9. Liang, C.K., Chang, L.W., Chen, H.H.: Analysis and compensation of rolling shutter effect. IEEE Trans. Image Proc. 17(8), 1323–1330 (2008)
10. Liu, J., Ji, S., Ye, J.: SLEP: Sparse Learning with Efficient Projections. Arizona State University (2009), http://www.public.asu.edu/∼jye02/Software/SLEP
11. Meilland, M., Drummond, T., Comport, A.I.: A unified rolling shutter and motion blur model for 3D visual registration. In: Proc. ICCV (2013)
12. Paramanand, C., Rajagopalan, A.: Shape from sharp and motion-blurred image pair. Intl. Jrnl. of Comp. Vis. 107(3), 272–292 (2014)
13. Punnappurath, A., Rajagopalan, A., Seetharaman, G.: Registration and occlusion detection in motion blur. In: Proc. ICIP (2013)
14. Radke, R.J., Andra, S., Al-Kofahi, O., Roysam, B.: Image change detection algorithms: A systematic survey. IEEE Trans. Image Proc. 14(3), 294–307 (2005)
15. Ringaby, E., Forssén, P.E.: Efficient video rectification and stabilisation for cellphones. Intl. Jrnl. Comp. Vis. 96(3), 335–352 (2012)
16. Seitz, S.M., Baker, S.: Filter flow. In: Proc. ICCV. pp. 143–150. IEEE (2009)
17. Starck, J.L., Moudden, Y., Bobin, J., Elad, M., Donoho, D.: Morphological component analysis. In: Optics & Photonics 2005. pp. 59140Q–59140Q. International Society for Optics and Photonics (2005)
18. Tai, Y.W., Tan, P., Brown, M.S.: Richardson-lucy deblurring for scenes under a projective motion path. IEEE Trans. Patt. Anal. Mach. Intell. 33(8), 1603–1618 (2011)
19. Vageeswaran, P., Mitra, K., Chellappa, R.: Blur and illumination robust face recognition via set-theoretic characterization. IEEE Trans. Image Proc. 22(4), 1362–1372 (2013)
20. Whyte, O., Sivic, J., Zisserman, A., Ponce, J.: Non-uniform deblurring for shaken images. Intl. Jrnl. Comp. Vis. 98(2), 168–186 (2012)
21. Wright, J., Yang, A.Y., Ganesh, A., Sastry, S.S., Ma, Y.: Robust face recognition via sparse representation. IEEE Trans. Patt. Anal. Mach. Intell. 31(2), 210–227 (2009)
22. Wu, Y., Ling, H., Yu, J., Li, F., Mei, X., Cheng, E.: Blurred target tracking by blur-driven tracker. In: Proc. ICCV. pp. 1100–1107. IEEE (2011)
23. Yuan, L., Sun, J., Quan, L., Shum, H.Y.: Blurred/non-blurred image alignment using sparseness prior. In: Proc. ICCV. pp. 1–8. IEEE (2007)