A Unified Rolling Shutter and Motion Blur Model for 3D Visual Registration

Maxime Meilland
CNRS-I3S, University of
Nice Sophia Antipolis, France
meilland@i3s.unice.fr

Tom Drummond
Monash University, Australia
tom.drummond@monash.edu

Andrew I. Comport
CNRS-I3S, University of
Nice Sophia Antipolis, France
Andrew.Comport@cnrs.fr

Abstract

Motion blur and rolling shutter deformations both inhibit visual motion registration, whether it be due to a moving sensor or a moving target. Whilst both deformations exist simultaneously, no models have been proposed to handle them together. Furthermore, neither deformation has been considered previously in the context of monocular full-image 6 degrees of freedom registration or RGB-D structure and motion. As will be shown, rolling shutter deformation is observed when a camera moves faster than a single pixel in parallax between subsequent scan-lines. Blur is a function of the pixel exposure time and the motion vector. In this paper a complete dense 3D registration model will be derived to account for both motion blur and rolling shutter deformations simultaneously. Various approaches will be compared with respect to ground truth and live real-time performance will be demonstrated for complex scenarios where both blur and shutter deformations are dominant.

1. Introduction

Electronic rolling shutter (RS) cameras have been becoming increasingly present in a wide number of applications and devices due to their low cost, low power consumption and continual read-out properties. In particular they are able to acquire much higher frequency scene dynamics via their intrinsic time-varying intra-image measurements whereas GS sensors acquire the entire image at the same time instant. This comes, however, at the cost of a more complex camera projection model. More specifically, each horizontal scan-line in a RS sensor is acquired at a different time instant and the data can be read-out in parallel. Unfortunately RS cameras capture deformed images if the camera is in motion or objects move in the scene.

On the other hand, image motion blur (MB) affects a large range of algorithms that deal with moving sensors (i.e. registration, SFM, camera shake, video analysis,etc.). MB depends directly on each pixel's exposure period (electronic shutter interval) and even with small motions some amount of blur is present. Subsequently if there is enough motion to produce RS deformations then there is imperatively enough motion to create MB effects also. In [3] on RS deformations, the authors note the problem of MB but choose to not address it. Other papers choose to minimise the effect by reducing the exposure to a minimum and using artificially bright lighting or easy to detect markers.

1.1. Rolling Shutter Motion Deformation

Early work that specifically modelled RS deformations was published in [26]. In this work the authors used an array of CMOS cameras to create undistorted images by selecting the scan-lines from different cameras but which were acquired at the same time instant. Prior to that there was some study made on X-slit, crossed-slit, or two-slit non-central projection cameras [6] and these models are closely related to the RS model, however, they do not consider motion between projections. A first study on estimating structure-and-motion from a RS video sequence is given in [17]. Here the authors correct image distortion using temporal optic flow correspondences and the assumption of a constant fronto-parallel camera velocity. The authors only validated their model on simulations but in practice the lateral rotational movements, which were assumed zero, are the most significant image deformation components.

A prominent model for estimating RS deformations is based on 12 parameters (6 for pose and 6 for velocity) using a known 3D model. In the seminal publication [1] it was necessary to initialize the pose and also correspondences between a target 3D model and the image. Later extensions involved considering 3D line models and regions of interest (ROI), using a high-end RS camera, to increase tracking frame-rate [5]. Structure of the scene was also estimated using stereo in [2]. In [14] $6 + N \times 6$ degrees of freedom (dof) are estimated by tracking groups of scan-lines independently to model non-uniform motion and more recently the same authors proposed a polynomial projection model. In all of these papers [1, 5, 2, 14], restrictive black and white markers were used to simplify feature extraction and avoid modelling low level feature deformations. In [7]

1550-5499/13 $31.00 © 2013 IEEE
DOI 10.1109/ICCV.2013.252

the approach is very similar to [1] except they model only rotation or planar scenes and use Harris features with a KLT tracker but only synthetic results are shown.

Rectifying RS images is another approach whereby motion estimates are used to re-render the images as though all the pixels were imaged by a GS. In [3], RS rectification is modelled via a translational model (an affine model was also considered). The deformations were treated as an underlying high-frequency jitter of the camera and this high-frequency motion is estimated using optic-flow point feature correspondences. The authors also calibrate the time coefficient between the capture of subsequent rows in the camera. In [8], the rectification of RS deformations of videos is achieved based on planar homographies. Lastly, in [9] RS rectification is simplified to rotation only and the authors use this to perform structure from motion estimation and Bundle Adjustment respectively.

1.2. Motion Blur Deformation

Surprisingly all these previous works have only considered RS deformation but none have handled motion blur. MB, in its general form, varies with respect to the full 6 dof motion of the camera. Ideally one would like to perform image-deblurring so as to obtain a deconvolved image of the scene. Some recent work on de-blurring includes [11] who performs 6dof de-blurring using an intertial sensor or [20] which models non-blind deblurring with over-exposed pixels. An overview of spatially invariant de-blurring is given in [24]. Previous authors have noticed that real-time 6dof pose estimation is much more efficient if blurred images are directly aligned rather than attempting the computationally expensive task of de-blurring. In [12], an inertial sensor was used to estimate motion and perform pose tracking in the presence of MB. Later rotations and MB were estimated using only a single image [13]. In [10] a generative translational MB model was proposed for a KLT tracker. Additionally they show that de-blurring the current estimate can be performed in an off-line reconstruction process. In [15], a homography based MB estimation approach is proposed. In this case 8 parameters of the $\mathbb{SL}(3)$ Lie group are estimated using 8 dof estimation for the homography parameters plus either 8 dof for the MB direction or 1 additional dof for the MB magnitude (limited to high frame-rate). In [21] a very similar approach to [15] is proposed but an Efficient Second-order Minimization (ESM) is used. In that paper only 8 parameters are estimated and the MB velocities are directly computed between subsequent estimates of the homography. Recently [18] gave a complete state-of-the-art on MB rendering for computer graphics. These approaches are important since they provide rendering techniques to allow generating blurred images in real-time.

1.3. Rolling Shutter and Motion Blur

None of the previously cited papers on RS and MB deformations have, however, attempted to simultaneously correct for both rolling shutter and blur distortion. The dual problem that should be considered is to both:

- correct for rolling shutter distortions that are induced by sensor motion or moving objects,
- correct for image blur induced by integrating moving light rays during the sensor exposure period.

It is clear that in a RS sensor these two issues are implicitly coupled. The RS deformation is observable when there is parallax between two successive scan-lines due to motion. Underneath this threshold only a small amount of blur will be observable and it will depend on the pixel exposure time and the motion observed in the image. In [23] different analog and digital imaging shutter mechanics are presented and the coupled effect of motion-blur and rolling shutter deformations is discussed. The paper, however, does not look at removing distortion or estimating unknown parameters.

1.4. Dense vs Feature-based

Another drawback of previous approaches is that they are mostly ”feature-based”. For RS models, [17, 3] use 2D optic flow to obtain geometric point correspondences. In [1, 5, 2, 14] markers are used for matching and in [7, 8, 9] KLT features are used. For MB, [12, 13] used edge features (edgels). Feature-based approaches inherently use rigid low-level operators to extract and match features. They are therefore prone to modelling error and do not work as intended on distorted images unless they are re-designed. On the other hand dense *direct* approaches are much more robust, especially in the case of MB. This can be attested by the fact that direct approaches still work in the presence of MB and that most MB approaches [10, 15, 18] consider direct region tracking. Recent approaches in dense localization and mapping [4, 19, 25] have shown that dense 3D registration can be performed in real-time using the full-image. To our knowledge neither RS nor MB deformation models have been considered for dense real-time 3D registration.

1.5. Overview

In this paper a unified model is proposed for monocular direct 6 dof pose tracking from a dense 3D model in the presence of both RS and MB deformations. The same model is also used to perform real-time structure and motion estimation for an RGB-D sensor. The main contributions are:

- A unified approach for both rolling shutter and motion blur estimation, via a 6 dof state model that improves on [1] for RS and [15] for MB.
- Dense minimization of intensity errors across the entire image as [4, 19, 25, 16] instead of using features.

- Motion blur model is also valid for global-shutter cameras by simply setting the camera readout time to zero.
- Real-time implementation on GPU, with color (RGB) MB estimation and RS correction.

For the MB model, the proposed approach follows [15, 18], however, direct estimation is performed on the entire image rather than a single patch and the 6 parameters of the $\mathbb{SE}(3)$ Lie group are estimated rather than the 8 parameters of $\mathbb{SL}(3)$. For the RS component, it will be shown that only 6 velocity parameters are sufficient instead of 12 as in [5, 2, 14]. In essence, the additional 6 dof corresponds to estimating the 3D model pose, however, as we show it can be calibrated only once in the first image. In fact, direct image-based approaches do not require any initialization if the first image is taken as the world frame. Both 6 and 12 dof models will be compared and detailed further in the article and the proposed approach will be shown to be valid for monocular model based registration.

2. Dense image observation model

Live direct 3D model-based tracking will be defined for monocular cameras using a dense large-scale world model that has been acquired in real-time by an automatic mapping process. The paper will also consider real-time 3D model acquisition by performing dense structure and motion (SaM) estimation with RGB-D sensors. The approach is based on real-time dense tracking and mapping as in [4, 19, 16, 25]. In the present paper a graph of RGB-D key-frames is stored to represent the 3D model within which 6 dof poses are the edges in the graph. All local key-frames can be transformed into a global world frame to obtain an equivalent 3D model. In this context consider a calibrated camera sensor with a colour brightness function $\mathbf{I} : \Omega \times \mathbb{R}^+ \rightarrow \mathbb{R}^+; (\mathbf{p}) \mapsto \mathbf{I}(\mathbf{p}, t)$, where $\Omega = [1, n] \times [1, m] \subset \mathbb{R}^2$, $\mathbf{P} = (\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_{nm})^\top \in \mathbb{R}^{mathrmmn\times 2} \subset \Omega$ are pixel locations within the image acquired at time t, and $n \times m$ is the dimension of the sensor image. It is convenient to consider the set of measurements in vector form such that $\mathbf{I}(\mathbf{P}, t) \in \mathbb{R}^{+\mathrm{nm}\times 1}$.

Now consider a key-frame that has been predicted from the 3D model $\boldsymbol{\mathcal{I}}^* = \{\mathbf{I}^*, \mathbf{D}^*\}$ as done in [16], or equally an image of an RGB-D sensor, $\boldsymbol{\mathcal{I}} = \{\mathbf{I}(t), \mathbf{D}(t)\}$, to be the set containing both intensities and depth measurements. $\mathbf{D} : \Omega \times \mathbb{R}^+ \rightarrow \mathbb{R}^+; (\mathbf{p}, t) \mapsto \mathbf{D}(\mathbf{p}, t)$ is the depth function associated to each pixel of the image. Note that t and $\mathbf{P}$ may be omitted in these functions for clarity.

Consequently $\mathbf{V} = (\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_{nm})^\top \in \mathbb{R}^{\mathrm{mn}\times 3}$ is defined as the matrix of 3D vertices related to the image pixels according to the following point-depth back-projection:

$$\mathbf{v}_i = \mathbf{K}^{-1}\overline{\mathbf{p}}_i\mathbf{D}(\mathbf{p}_i), \tag{1}$$

where $\mathbf{K} \in \mathbb{R}^{3\times 3}$ is the intrinsic matrix of the camera and $\overline{\mathbf{p}}_i$ are the homogeneous pixels coordinates.

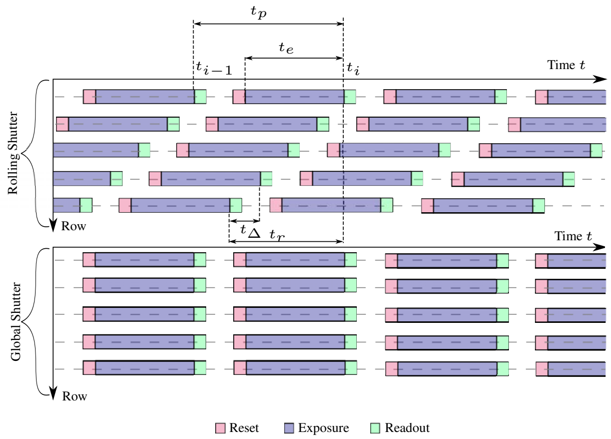


Figure 1. Top, RS camera model. Each raw of the sensor is sequentially exposed during a fixed exposure time t_e. The total readout time t_r is the delay between the readout of the first and the last row. The frame period t_p is the time delay between the readout of the same raw of the image. Bottom, global shutter mode. All rows of the image are exposed simultaneously during a fixed exposure time t_e.

The objective here is to register a *current* image $\mathbf{I}$ with a *reference* image $\mathbf{I}^*$ predicted from the 3D model (*e.g.* a graph of key-frames), where $\mathbf{I}$ is undergoing a full 3D transformation $\mathbf{T} = (\mathbf{R}, \mathbf{t}) \in \mathbb{SE}(3)$ defined between $\mathbf{I}$ and $\mathbf{I}^*$. Throughout, $\mathbf{R} \in \mathbb{SO}(3)$ is a rotation matrix and $\mathbf{t} \in \mathbb{R}(3)$ a translation vector. A superscript $*$ will be used throughout to designate the predicted reference view variables.

2.1. Global shutter

With a global shutter camera, all the pixels of the sensor are simultaneously exposed during the acquisition period t_p (see Figure 1). Under the assumption of brightness consistency and assuming that the exposure time of the sensor t_e is infinitesimally small, if the true pose $\widetilde{\mathbf{T}}$ is known ($\sim$ will denote true values throughout) then the warped image intensity at pixel $\mathbf{p}^*$ is equal to the reference image intensity:

$$\mathbf{I}^*(\mathbf{p}^*) = \mathbf{I}\left(w(\widetilde{\mathbf{T}}; \mathbf{K}, \mathbf{v}^*)\right), \tag{2}$$

where the warping function $w(\widetilde{\mathbf{T}}, \mathbf{K}, \mathbf{v}^*)$ warps a vertex $\mathbf{v}^*$, associated with the back-projected pixel $\mathbf{p}^*$ from (1), with the rigid transformation $\widetilde{\mathbf{T}}$ onto the normalized image plane:

$$\overline{\mathbf{p}}^w = \mathbf{K}\mathbf{\Pi}\widetilde{\mathbf{T}}\overline{\mathbf{v}}^*, \tag{3}$$

where the matrix $\mathbf{\Pi} = [\boldsymbol{I}_{3\times 3}, \mathbf{0}] \in \mathbb{R}^{3\times 4}$ projects 4 vectors onto 3 space. An overline will be used to indicate homogeneous coordinates normalized w.r.t. the last component. Since the projected pixel $\overline{\mathbf{p}}^w$ may not correspond to integer coordinates, a bilinear interpolation is used to obtain the corresponding intensities. Note that the intrinsic matrix $\mathbf{K}$ is assumed constant over time and may be omitted in the warping functions for clarity.

2.2. Rolling shutter

Now considering that the current image $\mathbf{I}$ has been acquired with a RS camera, under constant linear and angular velocities $\mathbf{x}_v = (\boldsymbol{\upsilon}, \boldsymbol{\omega}) \in \mathbb{R}^6$. As depicted in Figure 1, each row of a RS sensor is exposed sequentially with a time delay $t_\Delta = \frac{t_r}{n}$, where t_r is the total readout time and n is the number of rows in the image. The value of t_r is assumed constant for the camera and can be obtained from a calibration procedure as described in [22]. $\widetilde{\mathbf{T}}$ will be the pose of the last exposed row of the image at time t_i with respect to the last exposed row of the image at time t_{i-1}.

The warping function that transfers a current image intensity onto the reference frame is then defined such that:

$$\mathbf{I}^*(\mathbf{p}^*) = \mathbf{I}\Big(w_2\left(\mathbf{T}(\tau\widetilde{\mathbf{x}}_v)^{-1}, w_1(\widetilde{\mathbf{T}}, \mathbf{v}^*)\right)\Big), \tag{4}$$

where the first warping $w_1(\cdot)$ is the standard global shutter warping of equation (3). The second warping $w_2(\cdot)$ transfers the warped pixel from the GS space to the RS space. The scalar value τ is the time constant for a particular scan-line but since the reference pixels have been warped with a 6dof transformation, their coordinates no longer have a integer correspondence with a scan-line in the current image and they are scattered. It is therefore necessary to compute the scan-line constant for each pixel by:

$$\tau = t_\Delta \mathbf{e}_2^\top \mathbf{p}^w, \tag{5}$$

where $\mathbf{p}^w$ is the warped pixel resulting from the first warping $w_1(\cdot)$, which applies the motion, and $\mathbf{e}_2 = (0, 1)^\top$ extracts only the vertical coordinate of each pixel.

The matrix $\mathbf{T}(\cdot) = e^{[\cdot]_\wedge}$ is the integral of a constant velocity over τ, obtained by the exponential matrix of $\widetilde{\mathbf{x}}_v$,

$$\mathbf{T}(\tau\widetilde{\mathbf{x}}_v) = e^{\tau[\widetilde{\mathbf{x}}_v]_\wedge} = \int_0^\tau \widetilde{\mathbf{x}}_v \mathrm{d}t \in \mathbb{SE}(3), \tag{6}$$

with the operator $[.]_\wedge$ as:

$$[\widetilde{\mathbf{x}}_v]_\wedge = \begin{bmatrix} [\boldsymbol{\omega}]_\times & \boldsymbol{\upsilon} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \in \mathfrak{se}(3),$$

where $[.]_\times$ represents the skew symmetric matrix operator.

Due to the associative properties of the warping functions, the rolling shutter projection can be denoted as

$$\mathbf{I}^*(\mathbf{p}^*) = \mathbf{I}\left(w_{rs}(\mathbf{T}(\tau\widetilde{\mathbf{x}}_v)^{-1}\widetilde{\mathbf{T}}, \mathbf{v}^*)\right). \tag{7}$$

2.3. Motion blur

The blurring model detailed in this section is based on [15] for planar homography patches parametrized on $\mathbb{SL}(3)$. Here this idea is extended to use the entire image and to base transformations on $\mathbb{SE}(3)$.

Reconsidering the case of a global shutter camera and focusing on an image $\mathbf{I}$ corrupted by motion blur. Let $\mathbf{I}^u$ be the un-blurred version of that image (which is usually not available). Given the true 6 dof velocity $\widetilde{\mathbf{x}}_v$ and an exposure time of t_e, the blurred intensity can be generated at pixel $\mathbf{p}$ from the un-blurred image by the following model:

$$\mathbf{I}(\mathbf{p}) = \frac{1}{t_e}\int_{t_i - t_e}^{t_i} \mathbf{I}^u\left(w(\mathbf{T}(-t\widetilde{\mathbf{x}}_v), \mathbf{v})\right)\mathrm{d}t, \tag{8}$$

where $\mathbf{v}$ is the vertex corresponding to the pixel $\mathbf{p}$.

In image-based tracking the reference frame is usually maintained untouched to avoid corrupting the measurements and the aim is to transform and de-blur the current image such that it is equal to the reference as in equation (2):

$$\mathbf{I}^*(\mathbf{p}^*) = \mathbf{I}^u\left(w(\widetilde{\mathbf{T}}, \mathbf{v}^*)\right), \tag{9}$$

however, as has been shown in [10], de-blurring the current image $\mathbf{I}$ to obtain $\mathbf{I}^u$ from (8) is expensive and ill-conditioned.

It is therefore more efficient to introduce motion blur into the reference image so as to maintain this equality in the presence of blur. To create the same blur as observed in the current image, it is necessary to first transform the reference image to the current image (using the 3D model), then integrate the blurred set of intensities according to the motion vector and finally re-transform the new image back to the reference image (again using the 3D model). The current blurred image must still be warped to the reference according to equation (2). Finally, the equality can be written as:

$$\mathbf{I}\left(w(\widetilde{\mathbf{T}}, \mathbf{v}^*)\right) = \frac{1}{t_e}\int_{t_i - t_e}^{t_i} \mathbf{I}^*\left(w(\widetilde{\mathbf{T}}^{-1}\mathbf{T}(-t\widetilde{\mathbf{x}}_v)\widetilde{\mathbf{T}}, \mathbf{v}^*)\right)\mathrm{d}t. \tag{10}$$

In practice, the integral term of equation (10) is approximated with a discrete sum over M samples. This blur generation technique correspond to warping M images and averaging their values into a single image and is valid for constant velocity and under brightness consistency assumption.

2.4. Unified model

Now considering that the current image $\mathbf{I}$ is acquired with a RS camera, under the exposure period t_e, the following equality is obtained by combining equations (4) and (10):

$$\mathbf{I}\Big(w_{rs}(\mathbf{T}(\tau\widetilde{\mathbf{x}}_v)^{-1}\widetilde{\mathbf{T}}, \mathbf{v}^*)\Big) = \int_{t_i - t_e}^{t_i} \mathbf{I}^*\Big(w(\widetilde{\mathbf{T}}^{-1}\mathbf{T}(-t\widetilde{\mathbf{x}}_v)\widetilde{\mathbf{T}}, \mathbf{v}^*)\Big)\mathrm{d}t$$

This models consists in simultaneously warping the current image with RS distortions to a virtually blurred reference frame. Global shutter sensors are also handled, by simply setting the total readout time $t_r = 0$, as well as non-blurred images by setting the exposure time to an infinitesimally small value $t_e = \epsilon$.

3. Non-linear pose estimation

Now supposing that only close approximations $\widehat{\mathbf{T}}$ and $\widehat{\mathbf{x}}_v$ of the true pose $\widetilde{\mathbf{T}}$ and the true velocity $\widetilde{\mathbf{x}}_v$ are available. The aim is to estimate respectively the pose and velocity incremental transformations $\{\mathbf{x}_p, \mathbf{x}_v\}$ of the true values $\{\widetilde{\mathbf{x}}_p, \widetilde{\mathbf{x}}_v\}$ which satisfy

$$\widetilde{\mathbf{T}} = \widehat{\mathbf{T}}\mathbf{T}(\mathbf{x}_p) \quad \text{and} \quad \widetilde{\mathbf{x}}_v = \widehat{\mathbf{x}}_v + \mathbf{x}_v \ . \tag{11}$$

The 12 dof state vector is therefore $\mathbf{x} = \{\mathbf{x}_p, \mathbf{x}_v\}$ and it can be estimated by minimizing the following objective function in a non-linear least-squares procedure:

$$\widehat{\mathbf{x}} = \arg\min_{\mathbf{x}} \sum_{\mathbf{p}^* \in \mathbf{P}^*} \rho\Big(\mathbf{I}_w(\mathbf{x}, \mathbf{p}^*) - \mathbf{I}_b^*(\mathbf{x}, t_e, \mathbf{p}^*)\Big), \tag{12}$$

where $\mathbf{I}_w$ is the current (naturally blurred) warped image with RS distortions given by (7) and $\mathbf{I}_b^*$ is the reference (virtually blurred) image of equation (10). ρ is a robust M-estimator based on Huber's influence function which rejects un-modelled data such as self occlusions and local illumination changes.

The derivation of the 12 dof state RS model as it was first proposed by [1] assumes that the pose and the velocity are not coupled. This generic model allows to estimate the initial pose between a 3D model and the image along with the velocity. In a live tracking framework, the pose increment $\mathbf{T}(\mathbf{x})$ at time t_i is usually initialized with the last estimated pose at time t_{i-1}. If the time constant between t_{i-1} and t_i is known, then the true velocity can be obtained from the instantaneous velocity twist that parametrizes the pose. Therefore the state vector can be reduced to only 6 dof by assuming a constant velocity during the frame period t_p, leading to $\widetilde{\mathbf{x}}_v = \frac{1}{t_p}\widetilde{\mathbf{x}}_p$.

The unknown $\mathbf{x}$ is then obtained using a standard reweighted Gauss-Newton approach:

$$\mathbf{x} = -(\mathbf{J}^T\mathbf{W}\mathbf{J})^{-1}\mathbf{J}^T\mathbf{W}(\mathbf{I}_w - \mathbf{I}_b^*), \tag{13}$$

where the $nm \times 6$ Jacobian matrix $\mathbf{J}$ is evaluated at $\mathbf{x} = \mathbf{0}$, and where $\mathbf{W}$ is a diagonal weighting matrix of dimensions $nm \times nm$ obtained by M-estimation.

The pose estimate $\widehat{\mathbf{T}}$ is finally homogeneously updated by

$$\widehat{\mathbf{T}} \leftarrow \widehat{\mathbf{T}}\mathbf{T}(\mathbf{x}), \tag{14}$$

and the minimization is iterated until the increment $\mathbf{x}$ is sufficiently small: $\|\mathbf{x}\| < \epsilon$.

4. Structure and Motion

Whilst most classic RS or MB approaches perform model-based pose estimation using a monocular camera, it is also possible to consider the RS and MB model proposed in Section 2.4 for real-time structure and motion estimation using an RGB-D sensor (projective light, stereo or other). It is assumed that both the colour image and the depth image have synchronised rolling shutter cameras so that the same 6 velocity parameters can be used to rectify both images. The colour and 3D structure is estimated by fusing corrected RGB-D image over time as published in [16]. In that case the RS function of (4) is used to correct for distortion before fusion. As the real-time MB model of (10) is generative, this results in integrating blurred images into the 3D model. For the moment it is possible to perform computationally expensive de-blurring of the key-frames as a post process as for example in [20]. Future research will look at optimising these approaches for real-time de-blurring.

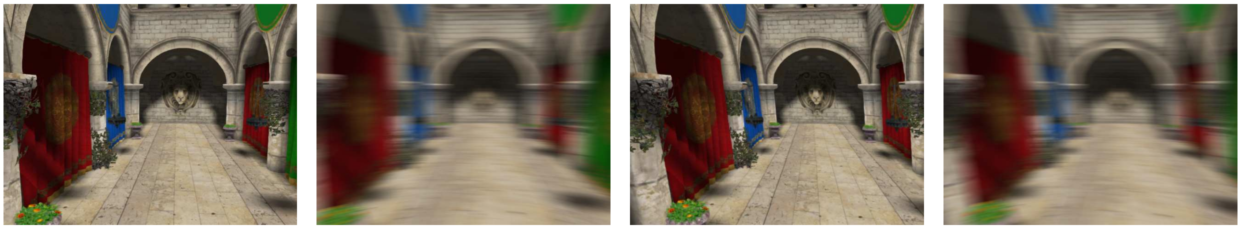
5. Experimental Results

A real-time implementation of the proposed approach was developed on the GPU using OpenCL. The SaM algorithm runs at 30 Hz with input images of size 640 × 480 pixels, on a Nvidia GTX 670 GPU. For more details on the real-time optimisation please refer to [16]. In the following experiments, the motion blur generation of equation (10) is performed with $M = 20$ samples, which appears to be sufficient to minimize aliasing and allows real-time computation. A more efficient strategy would be to adjust the number of samples with the camera velocity and exposure.

5.1. Simulated results

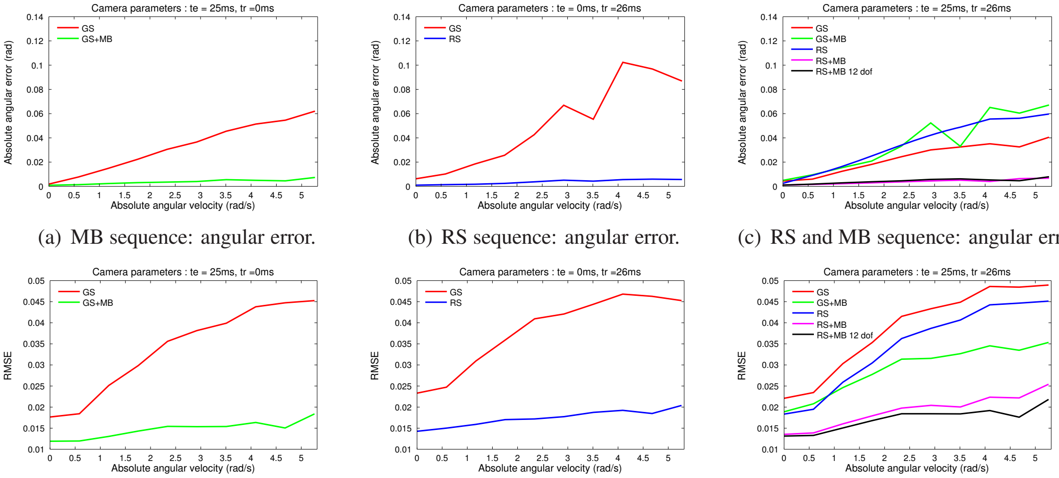
The algorithm has been tested on synthetic sequences of images with ground truth, generated from the Sponza atrium model (http://www.crytek.com). The rendering engine was designed using OpenGL with ambient illumination. Motion blur is obtained by invoking the rendering pipeline M times during the exposure time t_e, and the resulting images are averaged into a single image. In order to generate realistic motion blur and to avoid aliasing effects, 100 samples are used. RS effects are generated using equation (4) by re-projecting the rendered image into a new frame.

Three sequences of 445 images were generated using the same input trajectory computed from 6 dof velocity increments integrated over the frame period $t_e = 0.033s$. The first sequence simulates a global shutter camera with motion blur ($t_e = 0.025s, t_r = 0.0s$). The second sequence simulates a non-blurred rolling shutter camera ($t_e = 0.0s, t_r = 0.026s$), and the third sequence simulates a rolling shutter camera with motion blur ($t_e = 0.025s, t_r = 0.026s$). Figure 2 shows the image n^o369 of each sequence and illustrates the distortions induced by each camera model.



(a) Global shutter, without blur (b) Global shutter with blur (c) Rolling shutter without blur (d) Rolling shutter with blur

Figure 2. Synthetic scene observed with a constant velocity from the same viewpoint using different camera models. (a) is a perfect global shutter camera, (b) is a global shutter camera with motion blur, (c) is a rolling shutter camera without motion blur and (d) is a rolling shutter camera with motion blur.



(a) MB sequence: angular error. (b) RS sequence: angular error. (c) RS and MB sequence: angular error.

(d) MB sequence: RMSE. (e) RS sequence: RMSE. (f) RS and MB sequence: RMSE.

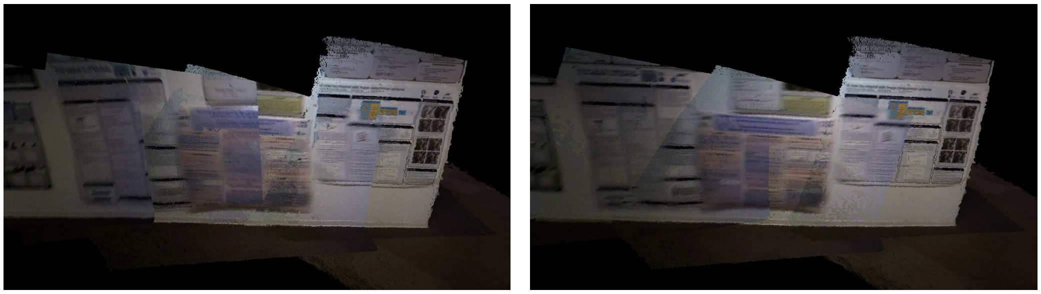
Figure 3. Simulation results for the 3 sequences. First row: Angular pose error with respect to the angular velocity. Second row: Reprojection error (RMSE) with respect to the angular velocity.

For each scenario, a single static reference frame $\mathcal{I}^* = \{\mathbf{I}^*, \mathbf{D}^*\}$ has been taken and is used as a reference for all trials. Figures 3(a),(b),(c) report the absolute angular error with respect to the absolute angular velocity of the camera. Figures 3(d),(e),(f) report the root mean squared error of the objective function (12) with respect to the absolute angular velocity of the camera. For each sequence, several registration models where considered: global shutter without motion blur (GS), global shutter with motion blur (GS+MB), rolling shutter without motion blur (RS), rolling shutter with motion blur (RS+MB) and rolling shutter with motion blur using 12 parameters (RS+MB 12 dof).

For the first sequence (only corrupted by motion blur) (a),(d), it can be seen that modelling motion blur (GS+MB) considerably improves the accuracy of the pose estimation compared to the standard model (GS). In the second sequence (only corrupted by rolling shutter perturbation) (b),(e) the same analysis can be made, modelling rolling shutter (RS) also improves the accuracy of the pose estimation compared to the standard model (GS). In the third sequence containing both motion blur and rolling shutter effects (c),(f), it appears that only modelling motion blur (GS+MB) or only modelling rolling shutter effects (RS) do not improve the accuracy even if the image re-projection error (RMSE) is smaller than the standard model (GS). This emphasizes the correlation between rolling shutter and motion blur effects in the image projection subsequently creating a false minimum. When blur and rolling shutter effects are simultaneously estimated, pose estimation remains accurate even with high velocities. The 12 dof model gives similar results to 6 dof but requires inverting a larger Jacobian and takes longer to converge. In Figure 3, only the rotational error w.r.t. ground truth has been provided since both translational and rotational components behave similarly and rotational movements produce much larger image velocities unless a high speed vehicle is used.

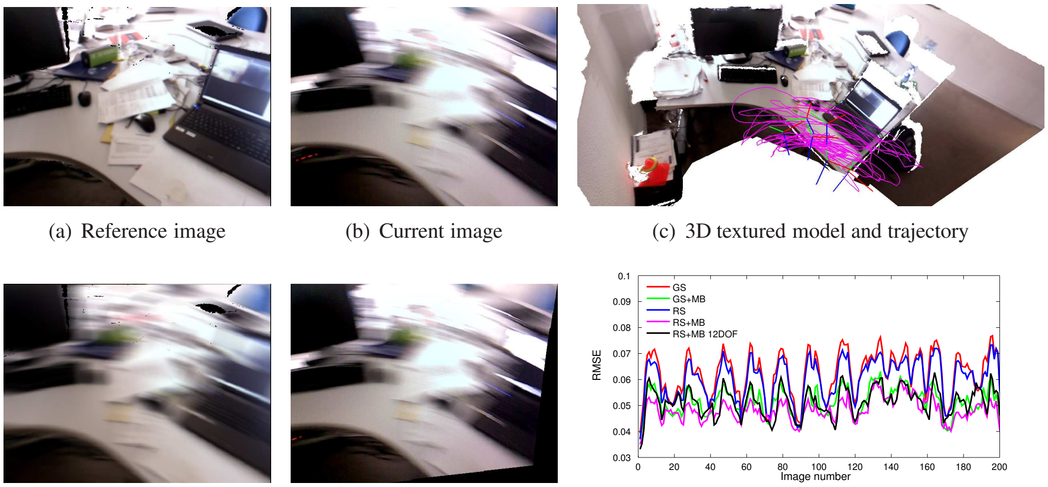
5.2. Real data

For the experiments, a calibrated Asus Xtion Pro Live RGB-D camera was used as RGB-D for SFM and as a monocular camera for model-based registration. The readout time t_r of the rolling shutter which was calibrated in [22] was used for the purposes of the following experiments.



(a) SaM with GS (b) SaM with RS

Figure 4. Structure and motion results. 3D point clouds obtained with the GS model (a) and with the RS model (b). The RS model allows to correctly handle image deformations.



(a) Reference image (b) Current image (c) 3D textured model and trajectory

(d) Reference with virtual blur (e) Current image undistorted (f) RMSE

Figure 5. Images extracted from a real sequence. (a) is the virtual reference frame used for registration generated from the dense 3D model shown in (c). (b) is the current image undergoing rolling shutter and motion blur distortions. (d) is the virtual image after applying the estimated blur. (e) is the current image after rolling shutter effect removal. The completely dark regions in the images correspond to regions where no information is available. (f) shows the RMSE for 200 images of the sequence.

In order to build a dense 3D model the real-time SaM approach proposed in [16] was used with the proposed rolling shutter and motion blur deformation model as introduced in Section 4. A first sequence was acquired by a user running down a corridor with a hand-held RGB-D camera. The images of figure 4 show a portion of the point clouds obtained using the GS model and the proposed RS model. For the GS point cloud 4(a), it can be seen that the posters on the wall are not correctly aligned due to the RS deformation. One the other hand, the RS model was able to correct the distortions 4(b), since the posters are well aligned.

A second set of monocular registration experiments were carried out in an office containing a desk, books and clutter using the 3D model acquired from the SaM step. The reconstructed model is composed of 6 key-frames shown in Figure 5(c). The camera was waved around the environment with very fast movements in each of the 6 dof and the estimated trajectory is also shown. In Figure 5 several images of the real sequence are shown. Due to computational constraints, it is not possible to iteratively generate a dense key-frame from the 3D model during registration. As such, some small occlusions are considered as outliers in the registration process (see the contour of the screen in Figure 5(d)). Figure 5(f) shows the root mean square image re-projection error. For visualization purposes only 200 images of a 1100 images sequence are given. The five different techniques are again compared. It can be seen that the RS+MB model performs the best and maintains a low RMSE across the entire sequence. The worst case is the standard GS model. It can be seen that GS+MB maintains an RMSE error which is also quite low but still slightly worse than the RS+MB. As was observed in the simulations with ground truth this model minimizes the error well but the pose estimate is not accurate. In the RS case the RMSE is quite poor most likely due to its inability to handle the motion blur. In practice for the same camera velocity, exposure value and read-out constants, the motion blur for this setup gives a much larger deformation than the rolling shutter effects. Finally, the 12 dof RS+MB only model gives average performance and high noise sensitivity characteristics are present. This could be explained by the fact that the model is over parametrized and so the estimate varies with noise in the image.

Many more results are provided in the associated video including large scale SaM and robust tracking for both simulated and real results.

6. Conclusions

This paper has addressed the problem of model-based 6 dof motion estimation using a consumer-level rolling shutter camera undergoing fast movements within large scenes. A unified solution for simultaneously estimating both motion blur and rolling shutter deformations was proposed within a direct dense registration framework that does not require feature extraction and matching. The same model was also used for live SaM using an RGB-D sensor. Results have shown the superior performance of the approach with respect to competing approaches using both sequences with ground truth and also via a live demonstrator that runs in real-time. It has been shown that it is only necessary to estimate the velocity twist of the camera motion to estimate rolling shutter, motion blur and camera pose information. This is an improvement over previous rolling shutter approaches because none handle motion blur nor do they parametrise the system with 6 dof therefore improving precision, robustness and computational efficiency.

References

[1] O. Ait-Aider, N. Andreff, J. Lavest, and P. Martinet. Exploiting rolling shutter distortions for simultaneous object pose and velocity computation using a single view. In *IEEE Int. Conf. on Computer Vision Systems*, 2006. 1, 2, 5

[2] O. Ait-Aider and F. Berry. Structure and kinematics triangulation with a rolling shutter stereo rig. In *Int. Conf. on Computer Vision*, 2009. 1, 2, 3

[3] S. Baker, E. P. Bennett, S. B. Kang, and R. Szeliski. Removing rolling shutter wobble. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2010. 1, 2

[4] A. I. Comport, E. Malis, and P. Rives. Accurate quadrifocal tracking for robust 3d visual odometry. In *IEEE Int. Conf. on Robotics and Automation*, 2007. 2, 3

[5] R. Dahmouche, O. Ait-Aider, N. Andreff, and Y. Mezouar. High-speed pose and velocity measurement from vision. In *IEEE Int. Conf. on Robotics and Automation*, 2008. 1, 2, 3

[6] D. D.Feldman, T. Pajdla. On the epipolar geometry of the crossed-slits projection. In *IEEE Int. Conf. on Computer Vision*, 2003. 1

[7] P. Forsse and E. Ringaby. Rectifying rolling shutter video from hand-held devices. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2010. 1, 2

[8] M. Grundmann, V. Kwatra, D. Castro, and I. Essa. Calibration-free rolling shutter removal. In *Int. Conf. on Computational Photography*, 2012. 2

[9] J. Hedborg, P.-E. Forssen, M. Felsberg, and E. Ringaby. Rolling shutter bundle adjustment. In *IEEE Computer Vision and Pattern Recognition*, 2012. 2

[10] H. Jin, P. Favaro, and R. Cipolla. Visual tracking in the presence of motion blur. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2005. 2, 4

[11] N. Joshi, S. Kang, L. Zitnick, and R. Szeliski. Image deblurring with inertial measurement sensors. In *ACM SIGGRAPH*, 2010. 2

[12] G. Klein and T. Drummond. Tightly integrated sensor fusion for robust visual tracking. In *British Machine Vision Conf.*, 2002. 2

[13] G. Klein and T. Drummond. A single-frame visual gyroscope. In *British Machine Vision Conf.*, 2005. 2

[14] L. Magerand and A. Bartoli. A generic rolling shutter camera model and its application to dynamical pose estimation. In *Int'l Symp. on 3D Data Processing, Visualization and Transmission*, 2010. 1, 2, 3

[15] C. Mei and I. Reid. Modeling and generating complex motion blur for real-time tracking. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2008. 2, 3, 4

[16] M. Meilland and A. I. Comport. On unifying key-frame and voxel-based dense visual slam at large scales. In *IEEE Int. Conf. on Intelligent Robots and Systems.*, 2013. 2, 3, 5, 7

[17] M. Meingast, C. Geyer, and S. Sastry. Geometric models of rolling-shutter cameras. *CoRR*, 2005. 1, 2

[18] F. Navarro, F. J. Serón, and D. Gutierrez. Motion blur rendering: State of the art. *Comp. Graph. Forum*, 2011. 2, 3

[19] R. A. Newcombe, S. Lovegrove, and A. J. Davison. Dtam: Dense tracking and mapping in real-time. In *IEEE Int. Conf. on Computer Vision*, 2011. 2, 3

[20] J. S. O. Whyte and A. Zisserman. Deblurring shaken and partially saturated images. In *IEEE Color and Photometry in Computer Vision Workshop*, 2011. 2, 5

[21] Y. Park, V. Lepetit, and W. Woo. Esm-blur: Handling & rendering blur in 3d tracking and augmentation. In *IEEE Int. Symp. on Mixed and Augmented Reality*, 2009. 2

[22] E. Ringaby and P.-E. Forssén. Scan rectification for structured light range sensors with rolling shutters. In *IEEE Int. Conf. on Computer Vision*, 2011. 4, 6

[23] M. Schoberl, S. Fossel, H. Bloss, and A. Kaup. Modeling of image shutters and motion blur in analog and digital camera systems. In *IEEE international conference on Image processing*, 2009. 2

[24] Šorel Michal and Šroubek Filip. *Restoration in the presence of unknown spatially varying blur, Image Restoration: Fundamentals and Advances*. CRC Press, 2012. 2

[25] T. Whelan, H. Johannsson, M. Kaess, J. Leonard, and J. McDonald. Robust real-time visual odometry for dense RGB-D mapping. In *IEEE Int. Conf. on Robotics and Automation*, 2013. 2, 3

[26] B. Wilburn, N. Joshi, V. Vaish, M. Levoy, and M. Horowitz. High-speed videography using a dense camera array. In *IEEE Computer Vision and Pattern Recognition*, 2004. 1