URS-NeRF: Unordered Rolling Shutter Bundle Adjustment for Neural Radiance Fields

Bo Xu*[1,2], Ziao Liu[2], Mengqi Guo[1], Jiancheng Li[2], and Gim Hee Lee[1]

[1] National University of Singapore
[2] Wuhan University
https://boxulibrary.github.io/projects/URS-NeRF/

Abstract. In this paper, we propose a novel rolling shutter bundle adjustment method for neural radiance fields (NeRF), which utilizes the unordered rolling shutter (RS) images to obtain the implicit 3D representation. Existing NeRF methods suffer from low-quality images and inaccurate initial camera poses due to the RS effect in the image. Furthermore, the previous method that incorporates RS images into NeRF requires strict sequential data input, thus limiting its widespread applicability. In contrast, our method recovers the physical formation of RS images by estimating camera poses and velocities, thereby removing the input constraints on sequential data. Moreover, we adopt a coarse-to-fine training strategy, in which the RS epipolar constraints of the pairwise frames in the scene graph are used to detect the camera poses that fall into local minima. The poses detected as outliers are corrected by the interpolation method with neighboring poses. The experimental results validate the effectiveness of our method over state-of-the-art works and demonstrate that the reconstruction of 3D representations is not constrained by the requirement of video sequence input

Keywords: Rolling Shutter Camera · Bundle Adjustment · Neural Radiance Fields

1 Introduction

NeRF [27] has recently emerged as a ground-breaking implicit 3D representation that provides new perspectives for computer vision and graphics. The prerequisites to learn good representation of a 3D scene with NeRF are high-quality images and accurate camera poses. However, it can be challenging to acquire such high-quality images and accurate camera poses from the commonly used RS cameras due to rolling shutter distortions caused by sequential scanning time of each row or column of the images taken from a moving camera [25]. Neglecting the distortions in images and inaccurate pose estimations due to the rolling shutter effect is detrimental to learning 3D representations with NeRF. Nonetheless, since NeRF has become a de-facto approach for learning 3D representations and RS cameras are widely used in many consumer products such as

[*] The work was done while Bo Xu is a visiting student at the National University of Singapore.

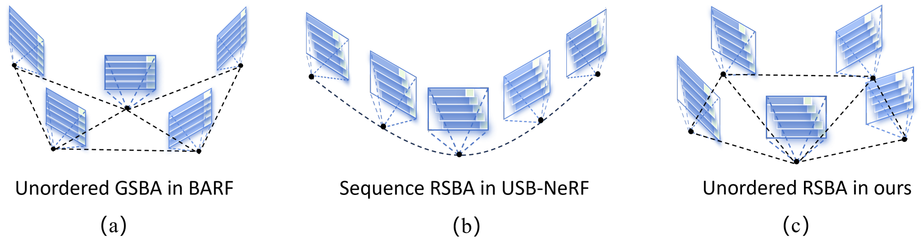


Fig. 1: Illustration of different NeRF BA settings. (a) BARF [24] works on unordered global shutter (GS) images, but is unsuitable for images with RS distortions. (b) USB-NeRF [22] requires the input to be strictly sequential video frames, which lacks the generality in BA. (c) Our method works on unordered images with RS distortions.

mobile phones due to inexpensive cost, low energy consumption and high-frame rates, it is therefore imperative to propose a framework for NeRF with rolling shutter images.

Many works [14, 20, 23, 30, 33, 40] propose the adaptation of bundle adjustment (BA) with the RS camera model to improve RS camera pose estimation and remove rolling shutter distortions. The inputs to these methods are either sequential video frames or unordered images. Methods based on sequential video frames [14, 30] can leverage the connection between consecutive frames to improve motion estimation results. For example, the use of smooth motion assumptions between consecutive frames and the spline-based trajectory model provides an effective way of reconstructing the 3D scenes with rolling shutter images. However, these constraints do not work for the more general unordered image setting and are susceptible to loop-closure errors when revisiting previously visited scenes. To circumvent the lack of consecutive frame constraints, the linear and angular velocities of the unordered camera images can be estimated under the assumption of constant motion during the exposure time. Although this weaker constraint would result in decreased accuracy, it can be easily mitigated with bundle adjustment.

NeRF demonstrates the impressive capability of encoding the implicit scene representation and rendering high-quality images at novel views with only a small set of coordinate-based MLPs. This leads to USB-NeRF [22] in introducing rolling shutter model into BARF [24] to learn the 3D representation and recover the camera motion trajectory simultaneously. However, the spline-based trajectory model used in USB-NeRF requires the input to be strictly from sequential video frames, and therefore lacks the generality of learning the 3D scene from unordered images. In view of this limitation, we propose rolling shutter bundle adjustment for neural radiance fields using unordered images in this paper. The differences between the various methods are demonstrated in Fig. 1. A rolling shutter model that estimates the camera pose, linear, and angular velocities to recover the camera motion corresponding to each row of the rolling shutter image is introduced into our method. The 3D representation and the camera motion are then trained by maximizing the photometric consistency between the rendered and captured rolling shutter images. To achieve stable optimization, we adopt a coarse-to-fine optimization strategy and utilize rolling shutter epipolar geometry constraints of the pairwise frames in the scene graph to detect the camera poses that are sub-optimal. The poses detected as outliers are corrected by the interpolation method with neighboring poses. Our method allows for the synthesis of high-quality GS images from novel view and the generation of datasets with varying degrees of rolling shutter effects, which is significant for the rolling shutter research community, *e.g.* rolling shutter bundle adjustment and simultaneous localization and mapping (SLAM).

Our **main contributions** are summarized as follows:

- We propose URS-NeRF which puts the RS camera model together with bundle adjustment and Tri-MipRF [15] for rolling shutter effect removal, novel view image synthesis, and RS camera pose and velocity estimations from unordered rolling shutter images.
- We introduce a coarse-to-fine strategy to prevent the bundle adjustment with rolling shutter motion parameters and NeRF from getting trapped in local minima. We further suggest a strategy to check for the erroneous pose in the bundle adjustment using the rolling shutter epipolar constraints of the pairwise frames in the scene graph.
- Extensive experimental evaluations are conducted on both synthetic and real datasets to evaluate the performance of our method. The experimental results demonstrate that our method achieves superior performance in terms of rolling shutter effect removal, novel view image synthesis, and camera motion estimation.

2 Related Work

Rolling Shutter Effect Correction. The 3D reconstruction with RS images has been widely studied. Depending on the type of input data (e.g., video sequence or unordered images) [23], the RSBA methods employ different models to formulate the camera motion corresponding to different scanlines within the exposure time of the image. Im et al. [17] make use of an RS video to solve RSSfM and present a small motion interpolation-based RSBA algorithm applicable to compensate for the rolling shutter effect. To model more complex motion of the camera, Patron et al. [30] propose a spline-based trajectory model to better reformulate the RS camera motion between consecutive frames. Zhuang et al. [40] develop a 9-point algorithm to estimate the relative pose from two consecutive RS images. The high-quality GS images can also be recovered with the relative poses. Despite the promising results achieved by using video-based RS methods, the overly restrictive constraints on input images still affect the applications, especially in case of the classical SfM pipeline. Albl et al. [1] address the unordered RSSfM problem and point out the planar degeneracy configuration of RSSfM. Liao et al. [23] proposes a normalization and covariance standardization weighting RSBA method that can be used to recover the camera poses with independent RS inputs. Unlike the introduction of velocity parameters or imposition of continuous time and motion through pose interpolation, a local differential RS constraint is proposed by Lao et al. [21] to deal with RS effects in SfM. With the development of deep learning, There are many methods proposed for RS effect correction with the network. Rengarajan et al. [31] propose a convolutional neural network (CNN) to estimate the row-wise camera motion from a single RS image. Fan et al. [11] recover the global shutter image from two consecutive images with unrolling shutter networks. Furthermore, Fan et al. [12] present a refined scheme under which the bilateral motion field recovered from two RS frames is used to produce high-fidelity GS video frames at arbitrary times. However, these methods usually require the use of a large dataset to complete training, and the generalization performance is constrained on the images of different characteristics, as verified in [22]. On the contrary, our approach does not require any pre-trained models, thus demonstrating superior generalization capabilities.

Neural Radiance Fields. Recently, NeRF [27] has attracted widespread attention due to its impressive capability to represent 3D scenes. Plenty of extensions are proposed for better performance in practice. To complete the training of NeRF with inaccurate or unknown cameras, a series of improvement algorithms [7, 13, 24, 35] have been proposed to optimize the network and camera poses simultaneously. Meanwhile, there are a lot of works focusing on how to improve the rendering quality [2–4] and training speed [6, 28, 37]. [29] address these concerns with multi-resolution hash encoding, which achieves instant reconstruction in around five minutes and rendering in real-time. [15] propose a Tri-Mip encoding into NeRF, which achieves high-fidelity anti-aliased renderings and efficient reconstruction. These works need both high-quality GS images and corresponding accurate camera poses, which is not suitable for the RS task. Li et al. [22] propose unrolling shutter bundle adjusted neural radiance fields, in which the motion trajectory of the RS video sequence is parameterized with the cubic B-Spline interpolation method. However, the method based on BARF requires a lengthy training time. Moreover, there is a limit on its applicability due to the highly complex constraints on the input. In contrast, our method does not have these limitations and works on the unordered rolling shutter images.

3 Notations and Preliminaries

3.1 Bundle Adjusting Neural Radiance Fields

BARF [24] is the first work to present bundle adjusting NeRF. Given the camera view with pose $\mathbf{T}_c^w = (\mathbf{R}_c^w, \mathbf{t}_c^w)$, a simple neural network such as MLPs is used to output the color $\mathbf{c} = (r, g, b)$ and volume density σ for each 3D location $\mathbf{x}^w$ and camera view direction $\mathbf{d}^w$ in the world coordinate frame. Using N 3D points $\mathbf{x}^c$ sampled along a ray in the camera frame $\mathbf{x}(t) = \mathbf{o} + t\mathbf{d}$, $\mathbf{x}^w$ can be computed by: $\mathbf{x}^w = \mathbf{T}_c^w \mathbf{x}^c$, and $\mathbf{d}^w$ can be computed by: $\mathbf{d}^w = \mathbf{R}_c^w \mathbf{d}^c$. Then the color and volume density of the sampled point are obtained as: $(\mathbf{c}, \sigma) = F(\mathbf{x}^w, \mathbf{d}^w)$. After getting the point color $\mathbf{c}_n$ and volume density σ_n of all the N points, the per-pixel RGB $\mathbf{c}\left(\mathbf{r}\right)$ value can be computed, *i.e.*:

$$\mathbf{c}\left(\mathbf{r}\right) = \sum_{i=1}^{N} T_i \left(1 - \exp\left(-\sigma_i \delta_i\right)\right) \mathbf{c}_i, \quad T_i = \exp\left(-\sum_{j=1}^{i-1} \sigma_j \delta_j\right), \tag{1}$$

where δ_i indicates the distance between i^{th} sample and $\left(i+1\right)^{th}$ sample, and T_i is the accumulated transmittance along the ray $\mathbf{r}$ from camera center to i^{th} 3D point. Finally, the image synthesis process is given by:

$$\hat{\mathbf{I}} = \mathcal{C}\left(\mathcal{F}\left(\omega\left(\mathbf{x}^{c_1}, \mathbf{T}\right); \mathbf{\Theta}\right), \ldots, \mathcal{F}\left(\omega\left(\mathbf{x}^{c_k}, \mathbf{T}\right); \mathbf{\Theta}\right)\right), \tag{2}$$

where $\mathcal{C}\left(\cdot\right)$ denotes the ray composition function, $\mathcal{F}\left(\cdot\right)$ indicates the NeRF network, $\omega\left(\cdot\right)$ is the rigid transformation which projects the point $\mathbf{x}$ from the camera frame to the world frame by the camera pose $\mathbf{T}$, and Θ indicates the network parameters.

Since the whole pipeline is differentiable, camera poses $\mathbf{T}$ and MLP network can be jointly optimized by supervising the rendering output and RGB image with the L2 distance:

$$\mathcal{L}_{rgb} = \sum_{i}^{K} \sum_{\mathbf{d}} ||\hat{\mathbf{I}}_i - \mathbf{I}_i\left(\mathbf{d}\right)||, \tag{3}$$

where K is the total number of images in the training dataset.

3.2 Tri-Mip encoding

To achieve both high-fidelity anti-aliased renderings and efficient reconstruction, Tri-Mip encoding is introduced into the implicit neural radiance field [15]. Instead of performing ray casting that ignores the area of the pixel in NeRF, the rendered pixels are formulated as a disc on the image plane. The radius of the disc can be computed by $\dot{r} = \sqrt{\Delta x \cdot \Delta y / \pi}$, where Δx and Δy are the width and height of the pixel in world coordinates. For each pixel, a cone casting is performed from the camera projection center $\mathbf{o}$ along the camera view direction $\mathbf{d}$. The cone are then sampled with a set of spheres $S\left(\mathbf{x}, \mathbf{r}\right)$ that are inscribed with the cone, where the center $\mathbf{x}$ and radius $\mathbf{r}$ of the sphere can be written as:

$$\mathbf{x} = \mathbf{o} + t\mathbf{d}, \quad \mathbf{r} = \frac{||\mathbf{x} - \mathbf{o}||_2 \cdot f\dot{r}}{||\mathbf{d}||_2 \cdot \sqrt{\left(\sqrt{||\mathbf{d}||_2^2 - f^2} - \dot{r}\right)^2 + f^2}}, \tag{4}$$

where f is the focal length. Furthermore, the spheres $S\left(\mathbf{x}, \mathbf{r}\right)$ are represented as feature vectors $\mathbf{f}$ by the Tri-Mip encoding that is parameterized by three trainable mipmaps $\mathcal{M}$:

$$\mathbf{f} = \text{Tri-Mip}\left(\mathbf{x}, \mathbf{r}; \mathcal{M}\right), \quad \mathcal{M} = \{\mathcal{M}_{XY}, \mathcal{M}_{XZ}, \mathcal{M}_{YZ}\}. \tag{5}$$

To make reconstructed scene coherent at different distance, the base level $\mathcal{M}^{L_0}$ is randomly initialized and other levels $\left(\mathcal{M}^{L_i}, i = 1, 2, \cdots, M\right)$ are derived from the previous level $\mathcal{M}^{L_{i-1}}$ by downscaling 2× along the height and width for each mipmap.

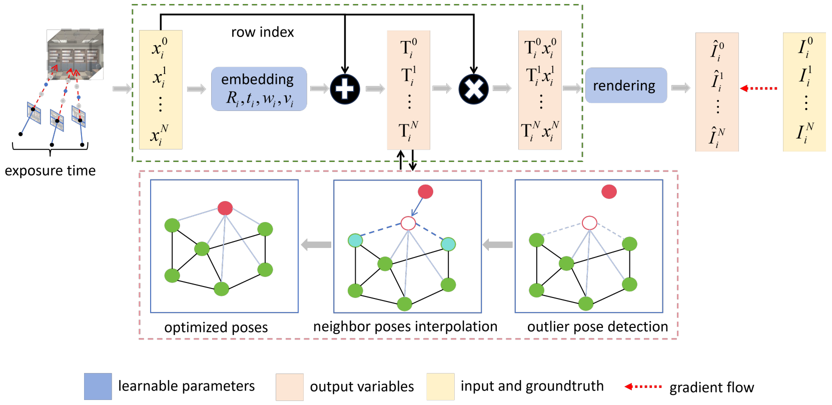

exposure time
row index
x_i^0 x_i^1 ⋮ x_i^N
embedding
R_i, t_i, w_i, v_i
T_i^0 T_i^1 ⋮ T_i^N
$T_i^0 x_i^0$ $T_i^1 x_i^1$ ⋮ $T_i^N x_i^N$
rendering
$\hat{I}_i^0$ $\hat{I}_i^1$ ⋮ $\hat{I}_i^N$
I_i^0 I_i^1 ⋮ I_i^N
optimized poses
neighbor poses interpolation
outlier pose detection
learnable parameters
output variables
input and groundtruth
gradient flow

Fig. 2: Overall pipeline of our proposed framework. We adopt a coarse-to-fine strategy to train rolling shutter images, and the scene graph is used to detect and correct the estimated poses that belong to outliers. Refer to the text for more details.

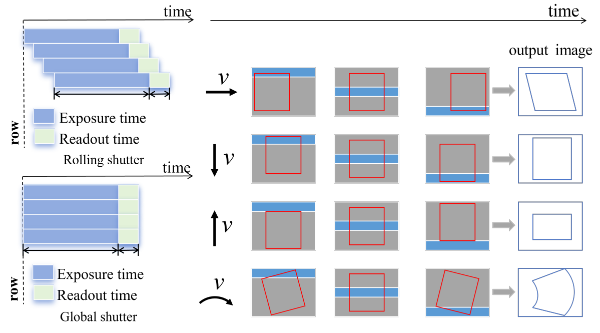
4 Our Method

Overview. Figure 2 shows the illustration of our framework. Since the image captured with a RS camera is exposed row-by-row, the pose corresponding to each scanline is different. Based on the query coordinates $\{\mathbf{x}^j\}_{j=0}^{L}$ in each rolling shutter image, our model constructs line-wise transformations $\{\mathbf{T}_i^j\}_{i=0,j=0}^{K,L}$ with the frame-dependent parameters $\mathbf{R}_i, \mathbf{t}_i, \boldsymbol{\omega}_i, \mathbf{v}_i$ and the row index of the sampled ray (*cf*. Sec. 4.1). Subsequently, the sampled rays are transformed from the query coordinates into the global coordinates. Finally, the color of each pixel can be rendered to get the rendered image $\{\hat{\mathbf{I}}^j\}_{j=0}^{L}$, which we use to minimize the photometric error with the given image $\{\mathbf{I}^j\}_{j=0}^{L}$ in our bundle adjustment formulation (*cf*. Sec. 4.2). To detect sub-optimal camera poses, a scene graph is constructed according to the number of the matched keypoints. The poses detected as outliers are corrected by the interpolation method with neighboring poses (*cf*. Sec.4.3).

4.1 Rolling Shutter Camera Model

In contrast to GS cameras where the whole image is captured simultaneously, each scanline of the RS camera is captured at different timestamps. Consequently, significant rolling shutter distortions appear in the image when the camera undergoes large motion, as can be seen in Fig 3. It is, therefore, difficult to obtain accurate prior RS camera poses using COLMAP [34] for implicit neural radiance fields. Since the rows of an image are not taken at the same time, it is necessary to find the camera pose $\mathbf{T}_i^j(t)$ as a function of time t to model the RS camera. We assume that the time t when a pixel is read out is linearly related to the vertical y-coordinate of the image. Furthermore, the camera motion is assumed to be constant during frame capture, which usually is well-satisfied for RS cameras. We follow [1,8,21] to model the instantaneous-motion as:

$$\mathbf{R}_i^j(u_j) = (\mathbf{I} + [\boldsymbol{\omega}_i]_\times u_j)\,\mathbf{R}_i^0, \quad \mathbf{t}_i^j(u_j) = \mathbf{t}_i^0 + \mathbf{v}_i u_j, \tag{6}$$



(a) RS and GS model (b) Image distortion in different motion modes

Fig. 3: (a) Image formation models of a RS camera (top) and a GS camera (bottom). (b) Final image shapes of different motion modes for RS camera. It demonstrates that each row of a rolling shutter image is captured at different timestamps, and would thus lead to different image distortions when the image is captured by a moving camera.

where $\mathbf{R}_i^j(u_j) \in \mathbf{SO}(3)$ and $\mathbf{t}_i^j(u_j) \in \mathbb{R}^3$ define the camera rotation and translation when the row index u_j is acquired, respectively. $[\boldsymbol{\omega}_i]_\times$ represents the skew-symmetric cross-product matrix of vector $\boldsymbol{\omega}_i$. $\mathbf{R}_i^0$ and $\mathbf{t}_i^0$ are the rotation and translation matrix when the first row is observed. $\mathbf{v}_i = [v_{ix}, v_{iy}, v_{iz}]^\top$ is the linear velocity vector and $\boldsymbol{\omega}_i = [\omega_{ix}, \omega_{iy}, \omega_{iz}]^\top$ is the angular velocity vector. Therefore, the rolling shutter image synthesis process can be written as:

$$\hat{\mathbf{I}}^r = \mathcal{C}\left(\mathcal{F}\left(\mathbf{f}\left(\omega\left(\mathbf{x}^{c_1}, \mathbf{T}, \boldsymbol{\omega}, \mathbf{v}\right)\right); \boldsymbol{\Theta}\right), \ldots, \mathcal{F}\left(\mathbf{f}\left(\omega\left(\mathbf{x}^{c_k}, \mathbf{T}, \boldsymbol{\omega}, \mathbf{v}\right)\right); \boldsymbol{\Theta}\right)\right) \tag{7}$$

where $\mathbf{f}(\cdot)$ is the Tri-Mip encoding. By supervising the synthesis output and RGB image with Eq. 3, we can train the MLP $\boldsymbol{\Theta}$ and mipmaps $\mathcal{M}$ representing the scene obtained by a GS camera with rolling shutter images, and then synthesize novel view global shutter images. It is worth noting that when $\boldsymbol{\omega}$ and $\mathbf{v}$ are set to zero, the training of the network degenerates to using global shutter images.

4.2 Coarse-to-fine Bundle Adjustment

To train the network with Tri-Mip encoding that accounts for the rolling shutter effect, we add $\mathcal{K} \times 6$ learnable pose embedding, $\mathcal{K} \times 3$ learnable linear velocity embedding and $\mathcal{K} \times 3$ learnable angular velocity embedding in the Tri-MipRF [15]. Subsequently, the gradients of $\mathcal{L}_{rgb}$ with respect to camera pose $\mathbf{T}_i$, linear velocity vector $\mathbf{v}_i$ and angular velocity vector $\boldsymbol{\omega}_i$ are derived from Eq. 7 as:

$$\frac{\partial \mathcal{L}_{rgb}}{\partial \mathbf{T}_i} = \sum_i^K \sum_{\mathbf{d}} \sum_n^N \frac{\partial \mathcal{C}}{\partial \mathcal{F}_n} \cdot \frac{\partial \mathcal{F}_n}{\partial \mathbf{f}_i^n} \cdot \frac{\partial \mathbf{f}_i^n}{\partial \mathbf{T}_i}, \tag{8a}$$

$$\frac{\partial \mathcal{L}_{rgb}}{\partial \mathbf{v}_i} = \sum_i^K \sum_{\mathbf{d}} \sum_n^N \frac{\partial \mathcal{C}}{\partial \mathcal{F}_n} \cdot \frac{\partial \mathcal{F}_n}{\partial \mathbf{f}_i^n} \cdot \frac{\partial \mathbf{f}_i^n}{\partial \mathbf{v}_i}, \tag{8b}$$

$$\frac{\partial \mathcal{L}_{rgb}}{\partial \boldsymbol{\omega}_i} = \sum_i^K \sum_{\mathbf{d}} \sum_n^N \frac{\partial \mathcal{C}}{\partial \mathcal{F}_n} \cdot \frac{\partial \mathcal{F}_n}{\partial \mathbf{f}_i^n} \cdot \frac{\partial \mathbf{f}_i^n}{\partial \boldsymbol{\omega}_i}. \tag{8c}$$

From the perspective of chain rule differentiation, introducing the estimated parameters $\mathbf{v}$ and $\boldsymbol{\omega}$ which are independent of the camera poses increases the degree-of-freedom of the model compared to the interpolation methods. However, this makes the bundle adjustment prone to getting trapped in local minima due to the lack of additional constraints, especially when the prior camera pose is inaccurate. Inspired by the multi-stage methods [10,13,18,38] that use the iterative refinement approach to enhance the performance of optimization, we introduce a multi-stage strategy that performs coarse-to-fine optimization in series. As shown in Fig. 2, the camera poses estimated from the coarse-stage are utilized as the initialization for the fine-stage. In the coarse-stage, the input images are downsampled to increase the receptive field of the sampled points, thereby accelerating the convergence of the optimization. The features encoded through Tri-Mip encoding are then fed into an MLP network to obtain the color and volume density. The initial inaccurate camera motion parameters and network parameters are optimized by supervising the rendering output and the corresponding downsampled RGB image. Finally, the erroneously estimated camera poses falling into local optima are identified and replaced to ensure the effectiveness of the coarse optimization (*cf*. Sec. 4.3). In the fine-stage, origin resolution images and the estimated camera poses from the coarse-stage are used for the final training. To learn the fine details as well as the rolling shutter effect in the image, we reinitialize the learning rates of parameters of MLP and mipmaps. The final optimized network model and camera parameters are obtained by supervising the rendering output and the original resolution RGB image.

4.3 Erroneous Pose Detection

In the optimization of the coarse-stage, estimated camera poses that are grossly wrong lead to an erroneous optimization thus preventing the model from self-correction in the following fine-stage. Unfortunately, existing methods [16, 36, 39] that assess the quality of the pose estimation by inferior rendering quality cannot be used due to the rolling shutter effect in the image. Consequently, we introduce the rolling shutter epipolar geometry constraints [8] as evaluation metrics to detect erroneously estimated poses. Given a pair of matched points $\mathbf{m}^i = \left[u^i, v^i, 1\right]$ and $\mathbf{m}^j = \left[u^j, v^j, 1\right]$ in the source frame $\mathbf{T}_s$ and target frame $\mathbf{T}_t$, respectively, the rolling shutter epipolar error can be written as [8]:

$$e^{ij} = \left[u^i, v^i, 1\right] \mathbf{K}^{-\top} \left[\mathbf{t}_{st}^0 + u^i \mathbf{v}_s - u^j \mathbf{R}_{st} \mathbf{v}_t\right]_\times \mathbf{R}_{st} \mathbf{K}^{-1} \left[u^j, v^j, 1\right]^\top , \tag{9}$$

where $\mathbf{R}_{st} = \left(\mathbf{I} + [\boldsymbol{\omega}_s]_\times u^i\right) \mathbf{R}_{st}^0 \left(\mathbf{I} - [\boldsymbol{\omega}_t]_\times u^j\right)$. $\mathbf{R}_{st}^0$ and $\mathbf{t}_{st}^0$ represent the relative rotation and translation of camera poses, respectively. $\mathbf{K}$ is the calibrated camera intrinsic matrix and assumed to be known. To construct the epipolar constraint, we extract keypoints for each rolling shutter image using SuperPoint [9], and obtain feature matches for each candidate image pair using SuperGlue [32]. Only the matching point pairs from nearby views are utilized to reduce the number of the wrong matches. We construct a scene graph to obtain the nearby views. Two images become neighbors when they share enough image keypoint matches. We simply select nearby views by sorting their neighbors according to the number of matches in descending order. The matched points are identified as outliers when e^{ij} exceeds δ_{th}. Moreover, the camera poses are regarded as low-quality if the number of outliers exceeds a certain ratio and are replaced by the neighboring poses with the interpolation method [22]. The scene graph construction only needs to be executed once for each scene in a preprocessing step.

5 Experiments

We evaluate the effectiveness of our method on synthetic and real datasets. The performance of rolling shutter effect removal and novel view image synthesis are benchmark with the commonly used metrics: PSNR, SSIM and LPIPS between the recovered global shutter images and the ground truth global shutter images. In addition, we also conduct a quantitative assessment of newly generated images with various rolling shutter effects. For camera pose estimation, we perform a SIM(3) alignment against the ground truth trajectory to get the absolute trajectory error (ATE). The Root Mean Square Error (RMSE) of the translation and rotation part is used for evaluation.

Baseline. We compare our method against the learning-free method DiffSfM [40] which constructs SfM similar to our rolling shutter modeling to remove the rolling shutter effect, and several learning-based methods: NeRF [27], BARF [24], Tri-MipRF [15] and USB-NeRF [22]. For NeRF, BARF and Tri-MipRF, we assume the inputs are global shutter images to train the implicit radiance fields. We re-implemented the interpolation method introduced in USB-NeRF based on Tri-MipRF to maintain consistency of the backbone. We denote our re-implementation of USB-NeRF as USB-NeRF-RE.

Datasets. We use the synthetic dataset WHU-RS [5], and real datasets ZJU-RS [19] to verify the effectiveness of our method. The WHU-RS dataset contains rolling shutter images and time-synchronized global shutter images and accurate ground truth collected in an ordinary room. To compare the impact of different rolling shutter effects, the datasets are divided into two trajectories with three sequences of different motion speeds (i.e. slow, medium, and fast corresponding to different rolling shutter effects). The scanline readout time is approximately 69.44μs. ZJU-RS datasets are collected with two mobile phones. Since the datasets lacks corresponding GS images, we choose two sequences and only evaluate the accuracy of the recovered camera trajectories compared with the groundtruth trajectories. The scanline readout time is approximately 20.83μs. We also conducted a qualitative analysis of rendering quality in a LivingRoom scene [22] and new RS dataset generation by setting different camera motions on the LLFF dataset [26].

Experimental Settings. We parameterize the camera poses $\mathbf{T}$ with SE(3) and initialize all the camera poses with the ground truth. To simulate inaccurate camera poses, the Gaussian noises with standard deviation $\delta_{trans} = 0.10$m and $\delta_{rot} = 1.15°$ are added to the translation part and rotation part of the initial

Table 1: Ablation experiments for Tri-MipRF, Tri-MipRF-BA, URS-NeRF-wo and URS-NeRF on Traj1-medium scene of WHU-RS dataset. Traj1-medium-large and Traj1-medium-slow are subjected to Gaussian noise with standard deviation $\delta_{trans} = 0.30$ m, $\delta_{rot} = 1.15°$ and $\delta_{trans} = 0.10$ m $\delta_{rot} = 1.15°$ rad, respectively. For each metric, the best in **bold**.

	Traj1-medium-large			Traj1-medium-small		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Tri-MipRF	15.24	0.46	0.70	16.04	0.45	0.64
Tri-MipRF-BA	26.35	0.84	0.13	27.13	0.86	0.11
URS-NeRF-wo	27.16	0.84	0.11	29.01	0.90	0.07
URS-NeRF	**29.37**	**0.90**	**0.05**	**29.34**	**0.91**	**0.06**

poses, which is similar to [24]. As the whole sequence of the WHU-RS and ZJU-RS datasets are too long for NeRF to process, we choose a subset of frames for each sequence. The initial linear velocity $\mathbf{v}$ and angular velocity $\boldsymbol{\omega}$ of each pose are set to zero. The camera intrinsic and scanline readout time of the RS camera are assumed known (provided by the dataset). To test the performance of different methods on unordered frames, we also randomly shuffled the selected sequences to generate the unordered datasets.

5.1 Ablation Experiments

To illustrate the effect of the introduced bundle adjustment in Tri-MipRF, the superiority of our rolling shutter modeling method, and the behavior of the coarse-to-fine strategy, we conduct ablation experiments to evaluate the performance of rolling shutter removal for the training view. Methods used for comparison include: Tri-MipRF is the baseline, Tri-MipRF-BA introduces the bundle adjustment in the Tri-MipRF, URS-NeRF-wo removes the coarse-to-fine strategy used in the URS-NeRF and URS-NeRF. As shown in Tab.1, our URS-NeRF exhibits superior performance on both small perturbation settings and large perturbation settings. This means our method formulates the physical image formation process of the RS camera, and it also verifies the effectiveness of our coarse-to-fine strategy. By comparing the experimental results of Tri-MipRF and Tri-MipRF-BA, we find that the introduction of bundle adjustment into Tri-MipRF can significantly improve the quality of 3D scene reconstruction and the accuracy of the pose estimation. This is also shown in Fig. 4. From the experiment results of Tri-MipRF-BA and URS-NeRF-wo, we can see that the quality of rendering is further improved after modeling the RS effect in the image. The comparison of URS-NeRF-wo and URS-NeRF shows that the coarse-to-fine approach used in our URS-NeRF enables the detection and correction of poses trapped in local minima and leads to more precise poses and enhances the rendering quality. In the large and small settings, the PSNR of our URS-NeRF improves by 8.13% and 1.13%, respectively, compared to URS-NeRF-wo.

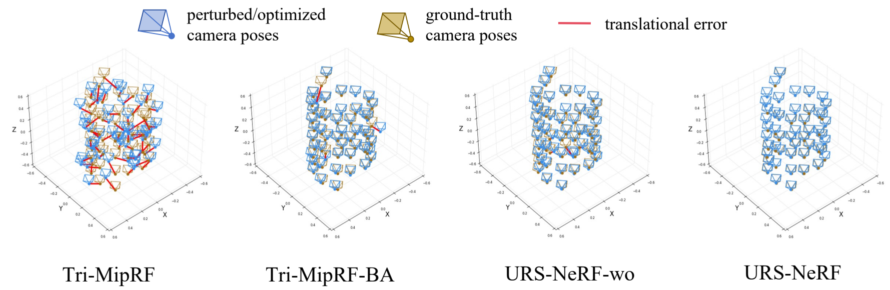
5.2 Quantitative Experiments

We evaluate the performance of our method against state-of-the-art methods in terms of rolling shutter removal and the accuracy of trajectory estimation.

Table 2: Quantitative comparisons on the synthetic datasets in terms of rolling shutter effect removal for training view on WHU-RS dataset. For the fast, medium, and slow modes of the WHU-RS dataset, the average values of each metric are computed from two scenes. For each metric, the best in **bold** for the unordered datasets and blue for the sequence video datasets.

		WHU-RS-Fast			WHU-RS-Medium			WHU-RS-Slow		
		PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Un-view	NeRF	21.66	0.57	0.67	20.89	0.56	0.69	21.05	0.56	0.67
	BARF	16.27	0.49	0.64	17.65	0.53	0.58	15.70	0.47	0.63
	DiffSfM	24.86	0.81	0.18	26.75	0.86	0.12	27.71	0.87	0.10
	Tri-MipRF	16.34	0.47	0.61	16.80	0.50	0.57	16.91	0.49	0.57
	Tri-MipRF-BA	26.17	0.86	0.13	26.67	0.83	0.22	28.01	0.82	0.17
	USB-NeRF-RE	16.91	0.50	0.61	19.18	0.60	0.50	21.07	0.65	0.42
	URS-NeRF	**27.55**	**0.85**	**0.10**	**27.90**	**0.88**	**0.08**	**29.42**	**0.89**	**0.08**
Seq-view	USB-NeRF-RE	29.07	0.87	0.12	29.54	0.88	0.11	30.16	0.89	0.10
	URS-NeRF	27.70	0.85	0.14	28.97	0.87	0.11	29.41	0.88	0.11

Tab. 2 presents comparisons between the groundtruth global shutter images and the synthetic images without RS effect from the training view. Our URS-NeRF achieves the best performance on the fast, medium, and slow modes of the unordered images. This is because our method effectively models the RS effect in the image, resulting in the implicit 3D scene representation that removes the RS effect. Since NeRF and Tri-MipRF do not incorporate bundle adjustment, the inaccurate poses and rolling shutter effects in the images result in poor rendering quality. Furthermore, comparing NeRF with BARF, and Tri-MipRF with Tri-MipRF-BA, we can observe that the rendering quality of Tri-MipRF-BA is improved compared to Tri-MipRF. However, the rendering quality of BARF is worse than that of NeRF. This is because the poses estimated by BARF fall into a local minimum when training the images for trajectory 2. Our URS-NeRF also outperforms DiffSfM, which employs the same rolling shutter modeling method as ours. However, DiffSfM only utilizes two frames to remove the RS effect, while our method uses images from multi-views, resulting in better performance. Fi-



Tri-MipRF Tri-MipRF-BA URS-NeRF-wo URS-NeRF

Fig. 4: Visual comparison of the initial and optimized camera poses for Tri-MipRF, Tri-MipRF-BA, URS-NeRF-wo, and URS-NeRF on Traj1-medium-large setting. URS-NeRF successfully realigns all the camera frames, while some estimated poses of Tri-MipRF-BA and URS-NeRF-wo get stuck at suboptimal solutions. The original Trimip-RF lacks the bundle adjustment, resulting in the poorest performance.

nally, by comparing the results of USB-NeRF-RE and URS-NeRF, our method works on both unordered and sequential data, while USB-NeRF-RE fails to reconstruct the 3D scene on unordered data. The difference in RS effect removal between our method and USB-NeRF on sequential data is marginal. This also demonstrates the generalization of our method. It is worth noting that our method performs best in the low-speed mode. As the camera movement speed increases, the rendering accuracy deteriorates. This is because slower speeds align more accurately with the assumption of constant velocity, resulting in better modeling performance.

Table 3 presents the camera motion trajectory estimation results with both synthetic and real datasets. The results demonstrate that both BARF and Tri-MipRF-BA suffer from the rolling shutter effect. The introduced distortions would affect the estimation of the poses and may even make the estimated pose worse. On the contrary, the accuracy of the pose estimation is improved since DiffSfM and URS-NeRF formulate the physical image formation process of RS camera when training the implicit neural radiance field. It is also evident from the pose estimation results that the cubic interpolation method used by USB-NeRF-RE imposes strict limitations on the input data, leading to the inability to handle unordered images. However, our method is not affected by the data order, making it more flexible for practical applications.

5.3 Qualitative Experiments

We also evaluate the qualitative performance of our method against the other baseline methods. Fig. 5 presents the comparisons with WHU-RS dataset. From both trajectory 1 and trajectory 2 scenarios, it can be seen that our method can obtain a 3D representation that is not affected by the rolling shutter effect under both fast and medium-speed camera movements, thereby synthesizing global shutter images. NeRF, BARF, and Tri-MipRF fail to learn the underlying undisturbed 3D scene representation, which proves the necessity to properly model the physical image formation process of RS camera when training with rolling shutter images. By comparing the rendering results of USB-NERF-RE and IRS-NERF, we can see that USB-NeRF is unable to handle unordered input data and the rendering quality deteriorates as the camera movement speed increases. More rendering results can be found in Fig. 6.

Table 3: Camera pose estimation on unordered view dataset and sequence view dataset. We evaluate the average values of the translation error (m) and rotation error (°). For each metric, the best in **bold** for unordered datasets and blue for the sequence video datasets.

		WHU-RS-Fast		WHU-RS-Medium		WHU-RS-Slow		ZJU-RS	
		Trans	Rot	Trans	Rot	Trans	Rot	Trans	Rot
Un-view	BARF	0.041	3.534	0.150	8.373	0.092	1.967	0.055	5.653
	DiffSfM	0.020	0.962	0.011	0.839	**0.009**	**0.386**	0.013	**1.237**
	Tri-MipRF-BA	0.019	1.710	0.016	1.431	0.023	1.677	0.011	2.009
	USB-NeRF-RE	0.495	10.787	0.375	7.743	0.304	8.511	0.124	15.912
	URS-NeRF	**0.014**	**0.697**	**0.009**	**0.381**	0.013	0.472	**0.007**	2.871
Seq-view	USB-NeRF-RE	0.008	0.649	0.009	0.324	0.011	0.296	0.009	2.462
	URS-NeRF	0.016	0.600	0.006	0.310	0.009	0.326	0.007	1.732

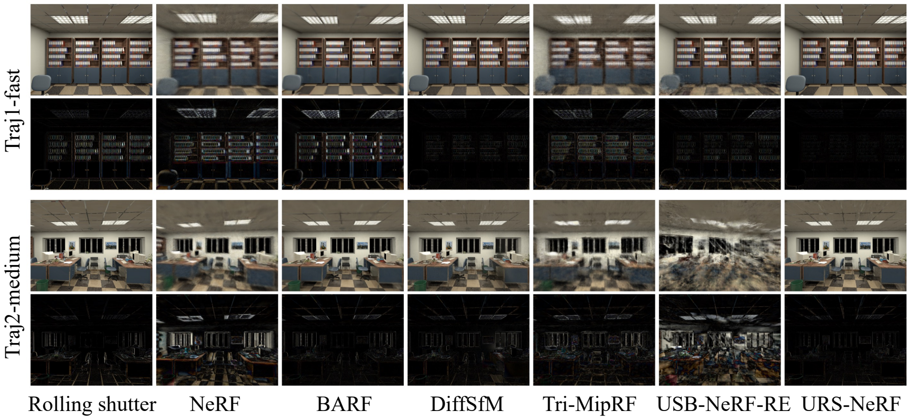


Fig. 5: Qualitative comparisons with the unordered images on the WHU-RS datasets. The second row consists of disparity maps between rendered images and ground truth, where darker areas indicate better performance. The experiments demonstrate a significant improvement in both rendering quality and reduction of rolling shutter effects with our method.

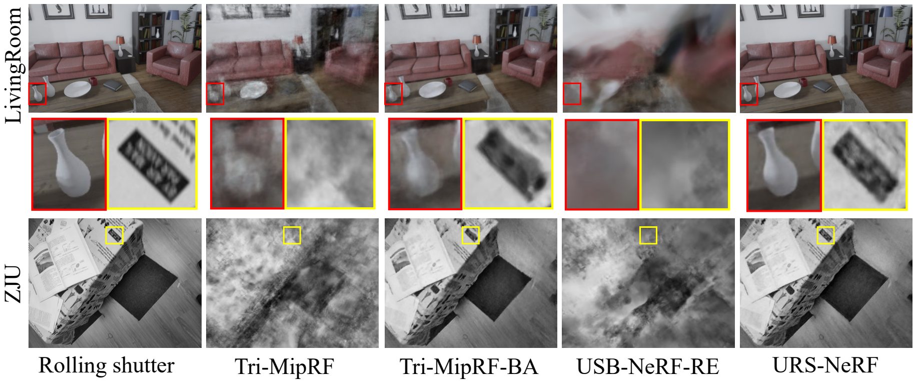


Fig. 6: Qualitative comparisons on ZJU and LivingRoom datasets. The detailed and overall images demonstrate that our method achieves better performance compared to other works on unordered images.

Since our method models the physical image formation process of a RS camera by estimating the linear and angular velocities of the camera, we are not only able to train a 3D scene representation unaffected by RS effect but also synthesize new images with various levels of RS effects. As shown in Fig. 7, our URS-NeRF is capable of restoring the GS images and generating images with different degrees of RS effects using the estimated linear and angular velocities. Furthermore, we can also train an implicit neural field using GS images and synthesize images with varying RS effects by setting different camera movement speeds. More novel view image synthesis and RS dataset generation results can

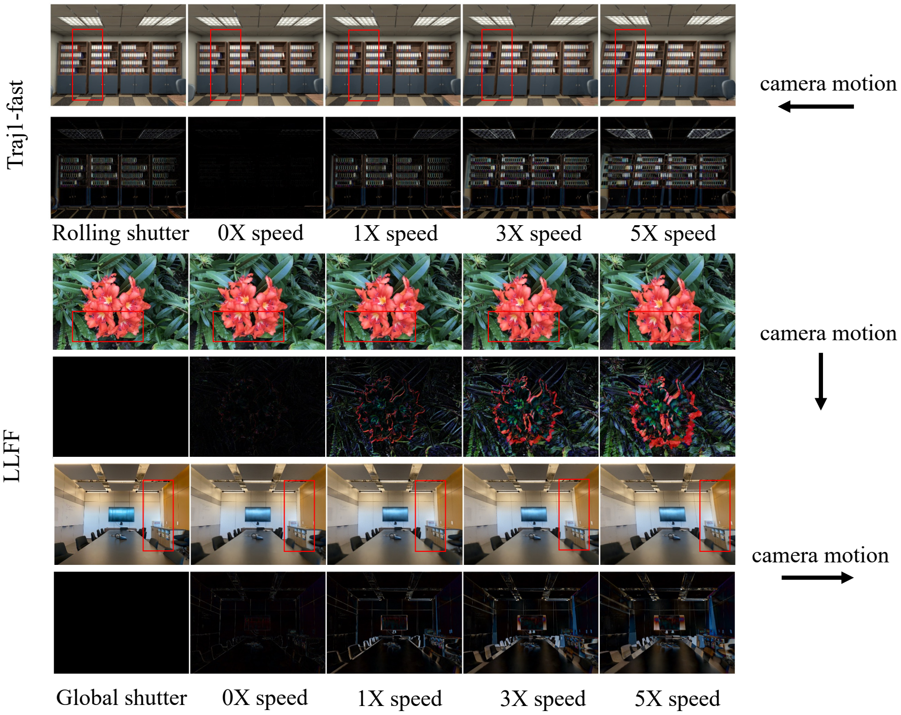


Fig. 7: New rolling shutter datasets generation with WHU-RS datasets. Our method can not only remove the rolling shutter effect (2^{th} cols) from the rolling shutter images but also synthesize datasets with varying degrees of rolling shutter effects (the camera motion speeds are 1x, 3x, and 5x, respectively).

also be found in the Appendix and supplementary video. They also demonstrate the superior performance of our method over prior works.

6 Conclusion

In this paper, we propose URS-NeRF for unordered rolling shutter bundle adjustment for neural radiance fields. The method introduces bundle adjustment into Tri-MipRF to estimate the RS camera pose, velocity and implicit 3D representation. To prevent the bundle adjustment with the rolling shutter model from getting stuck in local minima, we adopt a coarse-to-fine strategy and erroneous pose detection. Experimental results demonstrate that our URS-NeRF can successfully learn the true underlying 3D representations and recover the motion trajectory from a set of given unordered input RS images. Our method offers greater flexibility compared to interpolation-based approaches, providing a novel solution for implicit reconstruction with RS images.

Acknowledgement. This research / project is supported by the National Research Foundation, Singapore, under its NRF-Investigatorship Programme (Award ID. NRF-NRFI09-0008), the Tier 2 grant MOE-T2EP20120-0011 from the Singapore Ministry of Education, and the National Key Research and Development Program of China (2021YFB2501100).

References

1. Albl, C., Sugimoto, A., Pajdla, T.: Degeneracies in rolling shutter sfm. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V 14. pp. 36–51. Springer (2016)
2. Barron, J.T., Mildenhall, B., Tancik, M., Hedman, P., Martin-Brualla, R., Srinivasan, P.P.: Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5855–5864 (2021)
3. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5470–5479 (2022)
4. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Zip-nerf: Anti-aliased grid-based neural radiance fields. arXiv preprint arXiv:2304.06706 (2023)
5. Cao, L., Ling, J., Xiao, X.: The whu rolling shutter visual-inertial dataset. IEEE Access **8**, 50771–50779 (2020)
6. Chen, A., Xu, Z., Geiger, A., Yu, J., Su, H.: Tensorf: Tensorial radiance fields. In: European Conference on Computer Vision. pp. 333–350. Springer (2022)
7. Chen, Y., Lee, G.H.: Dbarf: Deep bundle-adjusting generalizable neural radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 24–34 (2023)
8. Dai, Y., Li, H., Kneip, L.: Rolling shutter camera relative pose: Generalized epipolar geometry. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4132–4140 (2016)
9. DeTone, D., Malisiewicz, T., Rabinovich, A.: Superpoint: Self-supervised interest point detection and description. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops. pp. 224–236 (2018)
10. Dollár, P., Welinder, P., Perona, P.: Cascaded pose regression. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp. 1078–1085. IEEE (2010)
11. Fan, B., Dai, Y., He, M.: Sunet: symmetric undistortion network for rolling shutter correction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4541–4550 (2021)
12. Fan, B., Dai, Y., Zhang, Z., Liu, Q., He, M.: Context-aware video reconstruction for rolling shutter cameras. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17572–17582 (2022)
13. Fu, H., Yu, X., Li, L., Zhang, L.: Cbarf: Cascaded bundle-adjusting neural radiance fields from imperfect camera poses. arXiv preprint arXiv:2310.09776 (2023)
14. Hedborg, J., Forssén, P.E., Felsberg, M., Ringaby, E.: Rolling shutter bundle adjustment. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. pp. 1434–1441. IEEE (2012)

15. Hu, W., Wang, Y., Ma, L., Yang, B., Gao, L., Liu, X., Ma, Y.: Tri-miprf: Tri-mip representation for efficient anti-aliasing neural radiance fields. In: ICCV (2023)
16. Huynh-Thu, Q., Ghanbari, M.: Scope of validity of psnr in image/video quality assessment. Electronics letters **44**(13), 800–801 (2008)
17. Im, S., Ha, H., Choe, G., Jeon, H.G., Joo, K., Kweon, I.S.: Accurate 3d reconstruction from small motion clip for rolling shutter cameras. IEEE transactions on pattern analysis and machine intelligence **41**(4), 775–787 (2018)
18. Jarrett, K., Kavukcuoglu, K., Ranzato, M., LeCun, Y.: What is the best multi-stage architecture for object recognition? In: 2009 IEEE 12th international conference on computer vision. pp. 2146–2153. IEEE (2009)
19. Jinyu, L., Bangbang, Y., Danpeng, C., Nan, W., Guofeng, Z., Hujun, B.: Survey and evaluation of monocular visual-inertial slam algorithms for augmented reality. Virtual Reality & Intelligent Hardware **1**(4), 386–410 (2019)
20. Lao, Y., Ait-Aider, O., Araujo, H.: Robustified structure from motion with rolling-shutter camera using straightness constraint. Pattern Recognition Letters **111**, 1–8 (2018)
21. Lao, Y., Ait-Aider, O., Bartoli, A.: Solving rolling shutter 3d vision problems using analogies with non-rigidity. International Journal of Computer Vision **129**, 100–122 (2021)
22. Li, M., Wang, P., Zhao, L., Liao, B., Liu, P.: Usb-nerf: Unrolling shutter bundle adjusted neural radiance fields. arXiv preprint arXiv:2310.02687 (2023)
23. Liao, B., Qu, D., Xue, Y., Zhang, H., Lao, Y.: Revisiting rolling shutter bundle adjustment: Toward accurate and fast solution. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4863–4871 (2023)
24. Lin, C.H., Ma, W.C., Torralba, A., Lucey, S.: Barf: Bundle-adjusting neural radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5741–5751 (2021)
25. Meingast, M., Geyer, C., Sastry, S.: Geometric models of rolling-shutter cameras. arXiv preprint cs/0503076 (2005)
26. Mildenhall, B., Srinivasan, P.P., Ortiz-Cayon, R., Kalantari, N.K., Ramamoorthi, R., Ng, R., Kar, A.: Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. ACM Transactions on Graphics (TOG) **38**(4), 1–14 (2019)
27. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM **65**(1), 99–106 (2021)
28. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. ACM Trans. Graph. **41**(4), 102:1–102:15 (Jul 2022). https://doi.org/10.1145/3528223.3530127, https://doi.org/10.1145/3528223.3530127
29. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. ACM Transactions on Graphics (ToG) **41**(4), 1–15 (2022)
30. Patron-Perez, A., Lovegrove, S., Sibley, G.: A spline-based trajectory representation for sensor fusion and rolling shutter cameras. International Journal of Computer Vision **113**(3), 208–219 (2015)
31. Rengarajan, V., Balaji, Y., Rajagopalan, A.: Unrolling the shutter: Cnn to correct motion distortions. In: Proceedings of the IEEE Conference on computer Vision and Pattern Recognition. pp. 2291–2299 (2017)

32. Sarlin, P.E., DeTone, D., Malisiewicz, T., Rabinovich, A.: Superglue: Learning feature matching with graph neural networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4938–4947 (2020)
33. Saurer, O., Pollefeys, M., Lee, G.H.: Sparse to dense 3d reconstruction from rolling shutter images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3337–3345 (2016)
34. Schönberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
35. Song, L., Wang, G., Liu, J., Fu, Z., Miao, Y., et al.: Sc-nerf: Self-correcting neural radiance field with sparse views. arXiv preprint arXiv:2309.05028 (2023)
36. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing **13**(4), 600–612 (2004)
37. Yu, A., Li, R., Tancik, M., Li, H., Ng, R., Kanazawa, A.: Plenoctrees for real-time rendering of neural radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5752–5761 (2021)
38. Zamir, S.W., Arora, A., Khan, S., Hayat, M., Khan, F.S., Yang, M.H., Shao, L.: Multi-stage progressive image restoration. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 14821–14831 (2021)
39. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 586–595 (2018)
40. Zhuang, B., Cheong, L.F., Hee Lee, G.: Rolling-shutter-aware differential sfm and image rectification. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 948–956 (2017)