Learning Structure-And-Motion-Aware Rolling Shutter Correction

Bingbing Zhuang[1] Quoc-Huy Tran[2] Pan Ji[2] Loong-Fah Cheong[1] Manmohan Chandraker[2,3]

[1]National University of Singapore [2]NEC Labs America [3]University of California, San Diego

Abstract

An exact method of correcting the rolling shutter (RS) effect requires recovering the underlying geometry, i.e. the scene structures and the camera motions between scanlines or between views. However, the multiple-view geometry for RS cameras is much more complicated than its global shutter (GS) counterpart, with various degeneracies. In this paper, we first make a theoretical contribution by showing that RS two-view geometry is degenerate in the case of pure translational camera motion. In view of the complex RS geometry, we then propose a Convolutional Neural Network (CNN)-based method which learns the underlying geometry (camera motion and scene structure) from just a single RS image and perform RS image correction. We call our method structure-and-motion-aware RS correction because it reasons about the concealed motions between the scanlines as well as the scene structure. Our method learns from a large-scale dataset synthesized in a geometrically meaningful way where the RS effect is generated in a manner consistent with the camera motion and scene structure. In extensive experiments, our method achieves superior performance compared to other state-of-the-art methods for single image RS correction and subsequent Structure from Motion (SfM) applications.

1. Introduction

Many consumer cameras such as webcams or mobile phones employ CMOS sensors due to their cost advantage. However, they come with the limitation of operating on a rolling shutter (RS) mechanism. In contrast to global shutter (GS), which exposes all rows of the sensor array at the same time, RS exposes them on a row-by-row basis from top to bottom with a constant time delay between consecutive rows. In the presence of camera motion during image capture, the delay between the exposure of the first row and last row can cause significant distortions in the captured image, resulting in deviation from the pinhole camera model [16].

Being a pure geometric distortion, the RS effect can be corrected rigorously by recovering the underlying geometry

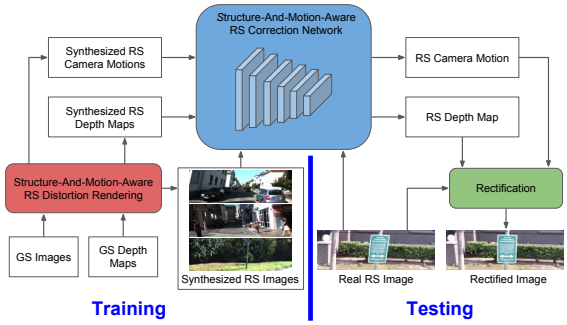

Figure 1. Overview of our approach. From a set of GS images and corresponding GS depth maps, we generate synthetic RS images with corresponding RS camera motions and RS depth maps for training our network. At testing, given a single real RS image, our network predicts an accurate RS camera motion and RS depth map, which are used for correcting RS effects in the input image.

(i.e. camera motion and 3D structure). However, due to the extra unknown parameters arising from the changes in the per-scanline camera poses during the exposure period, the geometric problem for RS cameras is often more complicated than its GS counterpart [3, 8, 18]. In particular, two-view geometry of RS cameras requires 44 2D point correspondences to get a linear solution [8], making it generally intractable in practice. Thus, method that uses two-view geometry to remove the RS effect has to impose special constraints, e.g. assuming differential camera motions and require non-trivial readout calibration [43].

Our first contribution in this work is a geometric degeneracy analysis in RS cameras. Note that despite the widespread deployment of RS cameras in many real-world applications, analyses on potential degeneracies have only emerged recently [4, 21]. In this paper, we show that RS two-view geometry is degenerate in cases of pure translational camera motion. In particular, there are infinitely many combinations of per-scanline camera poses and scene structures that can explain the 2D points in both views exactly in terms of reprojection errors. Such degeneracy poses additional challenges in applying two-view RS geometry for rectification, due to the prevalence of pure translation in practical applications, e.g., driving scenarios [13, 10].

Given such challenges using pure geometric method and

considering the recent successes of deep learning in geometric problems [7, 40, 26, 12], our second contribution is a data-driven approach to RS correction. Specifically, using a CNN-based approach, we learn to predict both the camera scanline velocity and depth from a single RS image. Although single-view depth prediction has been intensively studied [11, 27], the case of camera velocity between scanlines has not been addressed. Despite its ill-posedness at first glance, we show that it is feasible to estimate these camera motions from the distortion in the image appearance. The underlying intuition is that the appearance distortion represents the violation of the rigidity between all the scanlines and scene caused by camera motion during the exposure period. If enough has been learnt about the true geometry of the scene and its objects, this knowledge can be used to recognize the deformation caused by the RS distortion, following which camera motion recovery should be possible. The geometry so obtained can be used for image undistortion.

Our next contribution is a method to synthesize RS images from their GS counterparts in a geometrically faithful way, given the ground truth velocity and depth. This is used to generate large-scale training data. Finally, we also identify a further ambiguity that exists between pitch rotation and the image resizing operation during network training. Fig. 1 provides an overview of our approach. We denote our method as *"SMARSC"* short for *S*tructure-And-*M*otion-*A*ware *R*olling *S*hutter *C*orrection, since it reasons about the scene structures and the latent motions between scanlines.

The first attempt to use CNNs for single-view RS correction is presented in Rengarajan et al. [31]. However, there are significant differences between their work and ours. Their approach only aims at rectified images with visually appealing appearances; thus only per-scanline 2D deformation is modeled. Specifically, they apply a camera motion with only 2 degrees of freedom (DOF). For each pixel (on a scanline), RS effects are restricted to a within-scanline translation and an in-plane rotation. Further, depth information is not included in training. In contrast, our method explicitly takes into account both camera motions and scene structures and hence is able to produce rectified images that are not only visually pleasant but also geometrically consistent.

In summary, our contributions include:

- We identify and establish a detailed proof that RS two-view geometry is degenerate under pure translation. This result is important for understanding the intrinsic and algorithm-independent properties of RS two-view geometry estimation [8, 43].
- For single-view RS correction, we propose a novel CNN-based approach that is strongly governed by the underlying geometry, achieving good performance.
- We propose a geometrically meaningful way to synthesize large-scale training data and identify a geometric ambiguity that arises for training.

2. Related Work

Rolling Shutter Geometry. Many works studying the geometry of RS cameras emerge in recent years [18, 33, 3, 8, 43, 25]. Recently, Albl et al. [4] discuss a degeneracy that pitch rotation and parallel readout direction in two or more views may collapse the reconstructed scene into a plane. Critical motion sequences of RS cameras under a constant angular velocity are discussed in [21]. In this work, we present another degenerate scenario in RS two-view geometry under a pure translational camera motion. A similar result is discussed in [2] for a stereo rig . However, unlike [2] which provides only an intuitive discussion, we offer a formal proof that explicitly delineates the scope and impact of the degeneracy. For example, our proof makes it clear that camera velocity and image/scanline exposure can be different in the two views, while such freedom is not typically assumed for a stereo rig in [2].

Multiple-View RS Correction. Multiple-view methods often explore RS multiple-view geometry (via sparse/dense correspondences between the images) to correct RS effects [15] and simultaneously recover scene structures [43, 37]. They are able to handle different camera motions or scene structures. However, they require two or more input images [15, 43, 37] or non-trivial readout calibration [43, 37].

Single-View RS Correction. Single-view RS correction is inherently an ill-posed problem. To make it tractable, single-view methods assume simplified camera motions, e.g. pure rotation [32, 30, 24] or special scene structures, e.g. Manhattan world [30]. Thus, they cannot work well when the underlying assumptions on camera motions and scene structures do not hold. Further, many of them [32, 30, 24] rely on hand-crafted line/curves features extracted from the input image, thus they cannot handle images with very few or wrongly detected lines/curves. In contrast, our method uses powerful CNN-extracted features and employs a more general 6-DOF camera motion model and depth information to tackle various camera motions and scene structures.

3. Degeneracy in RS Two-View Geometry

RS Camera Modeling and Notations. Let us assume each RS image I has N scanlines in total, denoted as L_i with $i = 1, \dots, N$, and the camera is intrinsically calibrated. Since RS cameras capture each scanline sequentially, we denote the projection matrix for camera pose at the exposure slot of L_i as $\boldsymbol{P}_i = [\boldsymbol{R}_i \;\; \boldsymbol{T}_i]$, with $\boldsymbol{R}_i \in SO(3)$ and $\boldsymbol{T}_i \in \mathbb{R}^3$ being a rotation matrix and a translation vector respectively.

Pure Translation. Suppose that during the exposure period of two images I_1 and I_2, the RS camera undergoes pure translational motion along a constant direction denoted by a unit-norm vector $\boldsymbol{t} = [t_X, t_Y, t_Z]^\top$, and thus $\boldsymbol{P}_i$ of I_1 (respectively $\boldsymbol{P}_j$ of I_2), defined relative to $\boldsymbol{P}_1$ of I_1, can be expressed as $\boldsymbol{P}_i = [\boldsymbol{I} \;\; -p_i\boldsymbol{t}]$ (respectively $\boldsymbol{P}_j = [\boldsymbol{I} \;\; -q_j\boldsymbol{t}]$),

where p_i and q_j are scalars determined by the camera motion magnitude and $\boldsymbol{I}$ represents a 3×3 identity matrix.

Degeneracy Analysis. Here we only discuss for the case of $t_z \neq 0$ (for $t_z = 0$, please see supplementary material). We first formulate RS two-view geometry for pure translational camera motion between a pair of scanlines in the two images in terms of 2D correspondences and depths. Let us consider two scanlines L_i of I_1 and L_j of I_2 with camera poses $\boldsymbol{P}_i$ and $\boldsymbol{P}_j$ respectively, and a 3D point $\boldsymbol{S}$ observed in both scanlines as $\boldsymbol{S}_1 = [X_1, Y_1, Z_1]^\top$ and $\boldsymbol{S}_2 = [X_2, Y_2, Z_2]^\top$ in L_i's and L_j's camera coordinates respectively.

Denoting $\boldsymbol{T}^{ij} = [T_X^{ij}, T_Y^{ij}, T_Z^{ij}]^\top = (q_j - p_i)\boldsymbol{t}$, one can relate $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$ by $\boldsymbol{S}_2 = \boldsymbol{S}_1 - \boldsymbol{T}^{ij}$. Projecting this relationship into the 2D image, one gets $[\frac{X_2}{Z_2}, \frac{Y_2}{Z_2}]^\top = [\frac{X_1 - T_X^{ij}}{Z_1 - T_Z^{ij}}, \frac{Y_1 - T_Y^{ij}}{Z_1 - T_Z^{ij}}]^\top$. Subtracting $[\frac{T_X^{ij}}{T_Z^{ij}}, \frac{T_Y^{ij}}{T_Z^{ij}}]^\top$ on both sides of the above and rearranging, one arrives at

$$\boldsymbol{s}_2 - \boldsymbol{e} = \frac{Z_1}{Z_1 - T_Z^{ij}}(\boldsymbol{s}_1 - \boldsymbol{e}), \tag{1}$$

where $\boldsymbol{e} = [\frac{T_X^{ij}}{T_Z^{ij}}, \frac{T_Y^{ij}}{T_Z^{ij}}]^\top = [\frac{t_X}{t_Z}, \frac{t_Y}{t_Z}]^\top$ denotes the epipole, and $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$ are the 2D projections of $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$ respectively (i.e. $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$ is a 2D correspondence). Since $\boldsymbol{e}$ remains the same for any pair of scanlines, Eq. (1) indicates that all 2D points move along 2D lines radiating from the epipole, as illustrated in Fig. 2(b). This pattern, however, is exactly the same as in a GS camera model, and is the sole cue to recognize a pure translational motion, in which case the epipole is also termed as the *focus of expansion* (FOE) [16]. Therein lies the ambiguity, and in particular, one can explain the observed 2D point displacements by a GS camera model, with the following perturbations to the real T_Z^{ij} and Z_1:

- replacing all T_Z^{ij} with a common T_Z (recall that $\boldsymbol{T}^{ij} = (q_j - p_i)\boldsymbol{t}$ and hence $T_Z^{ij} = (q_j - p_i)t_Z$. One possible value of T_Z is $q_1 t_Z$ achieved by setting $\forall i : p_i = 0$ and $\forall j : q_j = q_1$, as shown in Fig. 2(a)); and,
- distorting the depth Z_1 to become $Z_1' = \frac{T_z}{T_z^{ij}} Z_1$ for each point $\boldsymbol{S}$; this value is obtained by solving $\frac{Z_1'}{Z_1' - T_Z} = \frac{Z_1}{Z_1 - T_Z^{ij}}$ so that Eq. (1) still holds.

Moreover, even if it is known that the observed 2D point movements are captured with a RS camera, the per-scanline camera positions along the translational direction, namely p_i and q_j, cannot be determined. Beyond the global scale ambiguity, there are evidently still infinite number of fake p_i' and q_j' that can produce physically possible (i.e. positive) and yet distorted depth $Z_1' = \frac{T_z^{ij\prime}}{T_z^{ij}} Z_1$ with $T_z^{ij\prime} = (q_j' - p_i')\boldsymbol{t}$.

Intuitively, in the absence of rotation, RS-induced distortion does not affect the direction of 2D point displacements but their motion magnitude. A GS two-view Structure from Motion (SfM) process can still regard it as a pure translational camera motion (with no RS distortions) by compensating the RS distortions with an appropriate corruption in the depths. In other words, no SfM algorithm can extract the RS effects from 2D correspondences only under such motion. Further, even if it is known that a RS camera is being used, the SfM algorithm is still not able to select the correct camera positions and depths due to the infinite number of solutions. Such degeneracy also implies numerical difficulties in SfM when the amount of rotation is small and there are noises in image measurements, though such scenario is theoretically not degenerate.

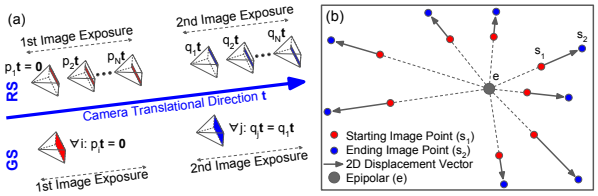

Figure 2. Degeneracy in RS two-view geometry. Both RS and GS pure translation in (a) produce radiating displacement in the 2D points in (b). The red/blue lines in (a) stand for scanlines.

From the above, we arrive at the following proposition:

Proposition. *RS two-view geometry for pure translational camera motion is degenerate, in the sense that one cannot tell if the two images are captured with a RS or GS camera based on 2D correspondences only. Even if the camera is known to be RS a priori, the per-scanline camera positions along the translational direction, namely p_i and q_j, cannot be determined.*

We note that such degeneracy in camera positions along a line also exists in other SfM problems, e.g. translation averaging [22, 38, 42] with collinear camera motions.

4. Structure-And-Motion-Aware Rolling Shutter Correction

In this section, we present the details of our network architecture (Secs. 4.1) and training data generation (4.2) for single-view RS correction. We also identify an ambiguity arising during network training (Sec. 4.3).

RS Image → GS Image (Rectification). Our proposed network takes a single RS image as input and predicts a corresponding RS camera motion and RS depth map, which can be used to perform rectification. In particular, for every pixel in the RS image, we can first back-project it to a 3D point using the estimated depth and then use the estimated per-scanline camera pose to project the 3D point to the GS canvas (the plane defined by $\boldsymbol{P}_1$ of the first scanline), yielding the rectified image. For modeling a RS camera motion, we employ a 6-DOF motion model and assume the camera has a *constant velocity* during the exposure period, which is a reasonable assumption and widely used in many recent works [20, 33, 3, 8, 43, 34]. In particular, we denote the constant per-scanline translational velocity and rotational velocity by $\boldsymbol{v} \in \mathbb{R}^3$ and $\boldsymbol{w} \in so(3)$, and write $\boldsymbol{P}_i = [\exp((i-1)\boldsymbol{w})^\top \; -(i-1)\boldsymbol{v}]$.

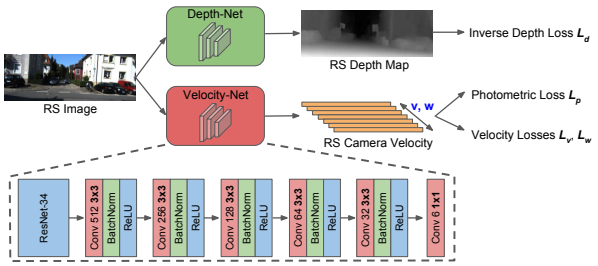

Figure 3. Detailed architecture of our structure-and-motion-aware RS correction network.

4.1. Network Architecture

Our network consists of two sub-networks, namely Depth-Net and Velocity-Net, for learning a RS depth map and RS camera motion respectively from a single image. Following SfMLearner [41], we adopt DispNet [28] as our Depth-Net for single-view RS depth estimation. For our Velocity-Net, we adapt the ResNet-34 architecture [17] by removing the last average pooling layer and adding four 3×3 convolutional layers (each followed by a BatchNorm layer and a ReLU activation layer) for extracting features and one 1×1 convolutional layer (with no bias) for regressing a 6-DOF camera velocity, namely a 3D translational velocity vector $\boldsymbol{v}$ and 3D angular velocity vector $\boldsymbol{w}$. Fig. 3 shows our network architecture in detail. We train our Depth-Net by using a regression loss $\mathcal{L}_d$. We regress inverse depth (instead of depth) to account for increasing uncertainty with increasing depth. For Velocity-Net, the training losses include regression losses $\mathcal{L}_v$ and $\mathcal{L}_w$ for evaluating the estimated translational and angular velocity respectively, and a photometric loss $\mathcal{L}_p$, which minimizes pixel intensity differences between the rectified image (obtained with the predicted camera velocity and the ground truth depth map) and the corresponding ground truth GS image (pixel intensities are scaled to [0,1] before computing $\mathcal{L}_p$). Note that we train the two networks separately, since we rely on synthetic training data which have ground truth for supervision of each network. We use L_1 norm for all the above losses.

4.2. Training Data Generation

Unlike geometry-based methods, our learning-based approach requires a large amount of training data, including RS images with ground truth RS camera velocities and RS depth maps. Since it is difficult to capture real RS images with ground truth velocity and per-pixel depth labels, we put forth a synthetic training data generation pipeline, based on the KITTI Raw dataset [13] (please see supplementary material for the list of sequences used for training and testing).

GS Image $\rightarrow$ RS Image (Distortion). We take the left view of the stereo pair in KITTI Raw as our ground truth GS image. We first compute the dense GS depth map from stereo using the state-of-the-art stereo method of [6]. Next, we generate a 6-DOF camera velocity as our ground truth RS camera motion which gives the per-scanline camera pose as well. We project each pixel in the GS image to the RS canvas, yielding the RS image. In particular, since it is not known which RS scanline the projected pixel will fall on to, we thus project each pixel $\boldsymbol{s}^{GS}$ (with the corresponding depth $Z_{\boldsymbol{s}}^{GS}$) using all RS scanlines L_i (with the corresponding per-scanline camera pose $\boldsymbol{P}_i$) and then select the 2D projection that is nearest to the hypothesized scanline as the corresponding image point in the RS image. This selection of scanline (and hence 2D projection) is made via

$$L_i^* = \arg\min_{L_i} \|[L_i]_y - [\Pi_{\boldsymbol{P}_i}(\boldsymbol{s}^{GS}, Z_{\boldsymbol{s}}^{GS})]_y\|, \qquad (2)$$

where $\Pi_{\boldsymbol{P}_i}$ is the projection function corresponding to the scanline L_i in the RS image and $[.]_y$ returns the row index of a 2D projection or a scanline. Since the above projections produce a set of image points scattered in between the grid intersections of the RS image, we perform interpolation to complete all pixels in the RS image. Note that in the above projections we get the RS per-pixel depth as well, providing us with the ground truth for training our Depth-Net. Also, since the KITTI camera is firmly mounted on the vehicle and hence has little variation in pitch and roll (yaw varies largely when the vehicle turns), we apply a small randomly sampled homography transformation on our ground truth GS image to increase pitch and roll variation before rendering the RS image. Fig. 4 shows example outputs at various steps.

Figure 4. Example outputs at various steps in our training data synthesis: (a) original GS image, (b) GS image transformed by a homography, (c)(d) interpolated RS image and depth map.

We note that since the RS camera often undergoes a small motion during the short exposure period, the rendering is generally able to maintain the sharpness in the original GS image and meanwhile exhibit the desired geometrical distortions in the rendered RS image. Also, due to errors arising from occlusion boundaries and imperfect stereo depth maps, the generated images inevitably contain some small artifacts. However, as we empirically validate in Sec. 5, our network is able to tolerate them and learn useful information.

4.3. Ambiguity between w_x-Induced Distortion and Vertical Image Resizing

The preceding training data generation pipeline returns training images of different sizes, whereas deep learning toolboxes require them to have the same size for efficient computation. To achieve this, one can either crop the images or resize them. Here, we show that the choice between these

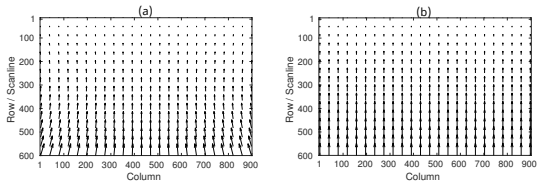

Figure 5. Resemblance between: (a) undistortion flow required to compensate the distortion introduced by RS camera rotation w_x and (b) vertical image resizing, confuses network training.

two options matters and opting for resizing can lead to inferior performance. In particular, we show that the distortion (or the corrseponding undistortion) induced by a small RS camera rotation around the x-axis, namely w_x, is similar to the changes caused by a vertical image resizing. Thus, resizing may undo or aggravate the amount of distortion actually brought about by the w_x in the training data, confusing the network in its learning of the correct camera motion.

Specifically, consider a pixel with coordinates (x, y) in a scanline L_i (corresponding to the pose $\boldsymbol{P}_i = [\exp((i-1)[w_x, 0, 0]^\top)^\top \ \ \mathbf{0}]$). The undistortion flow that will bring the RS image back to the GS image plane can be written as

$$\begin{aligned} u_x^{RS} &= -(i-1)w_x(x-x_0)(y-y_0)/f^2, \\ u_y^{RS} &= -(i-1)w_x(1+(y-y_0)^2/f^2), \end{aligned} \tag{3}$$

where (x_0, y_0) and f represent the principle point and focal length respectively, and we have used the differential expressions by Horn [19] to approximate the displacement induced by small motion. Note that this undistortion flow will be dominated by the linear term $-(i-1)w_x$ in small to medium field of view. When we perform vertical image resizing without properly compensating for its effect in the ground truth camera rotation w_x, the camera motion to be learnt will be confounded. This is because Eq. (3) coincides with the displacement field induced by vertically resizing the image by a factor of $(1+w_x)$ (the first row is taken as the reference row). One example is illustrated in Fig. 5. See supplementary material for more discussion.

While the readers might be reminded by this phenomenon of the well-known Bas-Relief ambiguity [1, 5, 9, 36] in the classical SfM, please note that, unlike Bas-Relief ambiguity, there is no confounding between w_y and horizontal image resizing here, as the distortion induced by the pose $\boldsymbol{P}_i$ only depends on the row index i and not the column index.

5. Experiments

Training Details. We use 42 sequences from KITTI Raw [13] with around 30,000 GS images in total and follow Sec. 4.2 to generate around 30,000 RS images for training our network. In particular, we randomly simulate a 6-DOF camera velocity $\{\boldsymbol{v}, \boldsymbol{w}\}$ such that the total translation magnitude, i.e. $\|(N-1)\boldsymbol{v}\|$, and total rotation magnitude, i.e. $\|(N-1)\boldsymbol{w}\|$, between the first scanline and last scanline are between [0,0.1] meters and $[0, \frac{\pi}{36}]$ radians respectively, and render one RS image from each GS image. Following Sec. 4.3, we crop the rendered images to the same size of 320×960 pixels. Our Depth-Net is trained from scratch, with similar training details as in [41]. For our Velocity-Net, we use the pre-trained weights of ResNet-34 on ImageNet classification to initialize the common layers in our network, whereas the newly added layers are randomly initialized using [14]. We set the weight of the translational velocity loss $\boldsymbol{\mathcal{L}}_v$ to 0.3, and those of the other losses to 1.0. We use ADAM [23] with learning rate 0.001. We set batch size to 40 images and implement our network in pyTorch [29].

Competing Methods. For single-view RS correction, we benchmark our method (SMARSC) against state-of-the-art methods [31, 30]. To deal with the ill-posedness of single-view RS rectification, Purkait et al. [30] assume a pure rotational camera motion and Manhattan world, thus we term their approach as *"MH"*. The work by Rengarajan et al. [31] is the closest to ours, also using CNNs to learn from a large-scale dataset of synthesized RS images. However, their approach uses a limited camera motion (within-scanline translation plus in-plane rotation) and ignores depth, thus we term their method as *"2DCNN"*. For a fair comparison, in synthesizing RS images for training 2DCNN, we rely on the same set of GS images used in our data generation pipeline, but employ their data generation code instead. We tune the motion generation parameters in their code to yield 2D distortions with similar ranges as ours, and render one RS image from each GS image. We also crop their rendered RS images to the same size as ours and adjust their fully-connected layers to fit the new input image size.

5.1. Synthetic Data

5.1.1 Image Resizing vs. Image Cropping

To validate the practical significance of the results in Sec. 4.3, we now demonstrate the advantage of cropping over resizing as a preprocessing step for obtaining training images with the same size. In particular, we first train two versions of our network, one with cropped images and another with resized images. We then generate three image sets with their camera motions restricted respectively to w_x-, w_y-, and w_z-*rotation only*, and evaluate both our trained models on each image set. Fig. 6 shows the cumulative distribution function (CDF) of w_x, w_y, and w_z prediction errors (prediction error is based on the total rotation, i.e. $(N-1)w_x$, $(N-1)w_y$, or $(N-1)w_z$, between the first scanline and last scanline) for both our models. From the results, it is evident that our model trained with resized images has much worse performance in w_x prediction (i.e. left red curve) as compared to its performances in w_y and w_z prediction (i.e. middle and right red curves), which implies that the ambiguity between vertical resizing and w_x-induced distortion has indeed reduced the network's ability to learn w_x-induced distortion. In contrast, our model trained with cropped im-

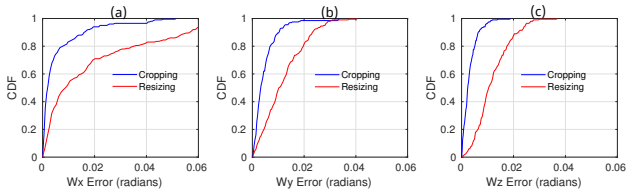

Figure 6. Performances of our networks when trained with cropped images or resized images and tested on image sets with: (a) w_x-rotation only, (b) w_y-rotation only, and (c) w_z-rotation only.

ages achieves similar performances across w_x, w_y, and w_z prediction (i.e. blue curves). Moreover, our model trained with cropped images consistently outperforms our model trained with resized images across all image sets.

5.1.2 Single-view RS Correction

We randomly select 200 GS images from totally over 5,500 frames of the 2 test sequences on KITTI Raw [13] (with no overlap with the sequences used for generating the training data) and follow Sec. 4.2 to synthesize 200 RS images (with 6-DOF camera motions) as our test data. We compare our method (SMARSC) against MH [30] and 2DCNN [31] for single-view RS correction. We first show a few qualitative results to understand the behaviors of the methods intuitively, and then present quantitative comparisons between the estimated and the ground truth undistortion flow (which maps pixels in the input RS image to the ground truth GS image).

Qualitative Comparisons. Fig. 7 plots qualitative results on an input RS image with a typical scene in KITTI Raw. From the results, it is evident that our method produces the best rectified image, with our undistortion flow visually close to the ground truth undistortion flow (e.g. see R2-C3 vs. R1-C3 — "R" and "C" are short for Row and Column respectively). That is partially due to our accurate predicted depth map (e.g. see R2-C4 vs. R1-C4). On the other hand, 2DCNN ignores depth information or at least assumes each scanline has the same depth, and hence it can only rectify some regions in the image while leaving the other regions distorted (e.g. in R4-C1, the blue box is rectified relatively well, whereas the red box still contains notable RS effects). For MH, its performance depends on the quality of line detection and vanishing point estimation (i.e. it assumes Manhattan world) and it does not model camera translation, which exists in this input image. Both MH and 2DCNN do not consider scene depths and hence their undistortion flows do not reflect scene structures (e.g. see R3-C3 and R4-C3). Please see supplementary material for more results.

Quantitative Comparisons. We now quantitatively evaluate the undistortion flow estimated by each method against the ground truth undistortion flow. Fig. 8(a) presents the CDF of undistortion flow errors (flow error is based on the mean end-point error from each image) in the test set of 200 RS images (with 6-DOF camera motions). Although 2DCNN and MH both rely on limited camera motion models, 2DCNN slightly outperforms MH, mostly due to MH's dependence on hand-crafted features and Manhattan world. In contrast, our method, which uses a 6-DOF camera motion model, achieves the best performance on this set.

In some scenarios, the translation of the RS camera can be negligible compared to the overall depth of the scene or the camera undergoes pure rotation [30]. We thus render another test set of 200 RS images with *pure rotation* and plot quantitative results on this set in Fig. 8(b). As expected, MH has slightly better performance than 2DCNN, likely since MH is particularly designed for pure rotation. Nevertheless, MH is still inferior to our method, mostly due to MH's dependence on hand-crafted features and Manhattan world assumption. We note that our method is not specifically trained on RS images with pure rotation but still performs well in this special case.

Sensitivity to Depth Estimation Accuracy. Here we study how accurate depth prediction needs to be so that our method can still benefit from this extra cue as compared to depth-unaware methods, e.g. 2DCNN. We conduct this experiment on another set of 200 synthesized RS images with *pure translation* (distortion induced by rotation is independent of depth and is thus excluded). We use our estimated camera motion (by our Velocity-Net) together with one of the following sources of depth information to perform rectification: 1) our estimated depth (by our Depth-Net), 2) "SMARSC-GT-Depth" — our ground truth depth, and 3) different approximate versions of 2), i.e. we approximate 2) by quantizing its continuous depth range into N_b bins and then for each pixel replacing the continuous depth in 2) by the median depth of the bin it falls into. We evaluate for $N_b = 1, 4, 16$, denoted as "SMARSC-GT-Depth-N_b-Bins" respectively. We also compare against 2DCNN (MH is ignored as it does not model translation). Fig. 8 shows the CDF of undistortion flow errors (again the mean end-point error from each image) of all methods on this image set.

From the results, the accuracy of SMARSC-GT-Depth-N_b-Bins declines with fewer number of bins. However, it still obtains better results than 2DCNN even with $N_b = 1$. This is reasonable, since under pure translation (as illustrated in Fig. 2(b)), a relatively good translation estimate (or FOE estimate) by our Velocity-Net will confine the undistortion flow to its ground truth direction, and hence the accuracy of depth only affects the magnitude of undistortion flow, leading to a small decline in the overall performance. In contrast, such depth-dependent distortion is beyond what the simpler 2D deformation model used in 2DCNN can capture, yielding the above performance gap. Also, our method (SMARSC), which relies on both our estimated camera motion and estimated depth, performs similarly to SMARSC-GT-Depth-4-Bins, which uses our estimated camera motion and merely 4 coarse bins of ground truth depth. This implies that, when combined with our estimated camera motion, our estimated

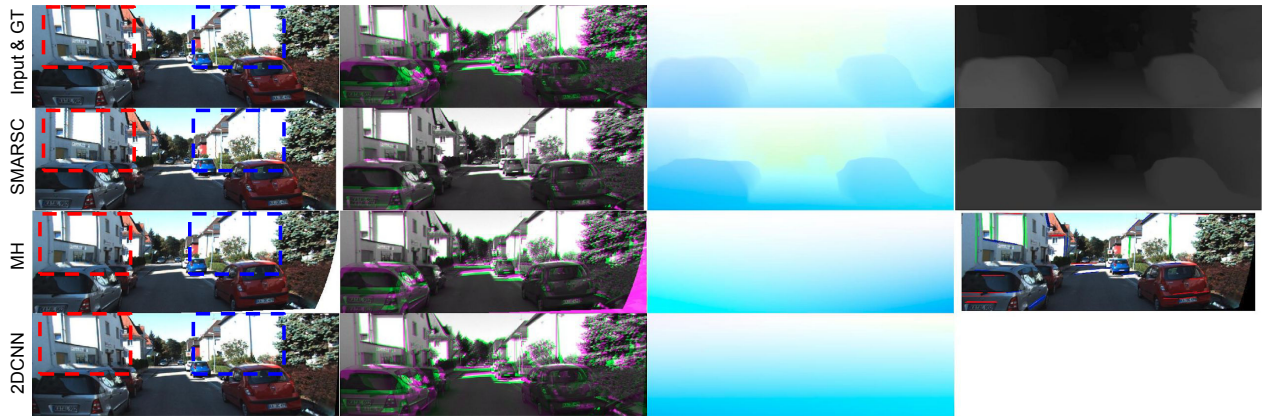

Figure 7. Qualitative comparisons on a synthetic RS image with a typical scene in KITTI Raw. The first row shows the input RS image, input RS image overlaid on ground truth GS image (pink and green colors indicate the intensity differences), ground truth undistortion flow (visualized according to [35]), and ground truth depth map (bright and dark colors indicate small and large depth values respectively). The next three rows plot the results of our method (SMARSC), MH, and 2DCNN respectively with each row showing from left to right the rectified image, rectified image overlaid on ground truth GS image, estimated undistortion flow, and estimated depth map. Note that since MH and 2DCNN do not predict depths, we instead show the line detection result for MH and leave an empty figure for 2DCNN.

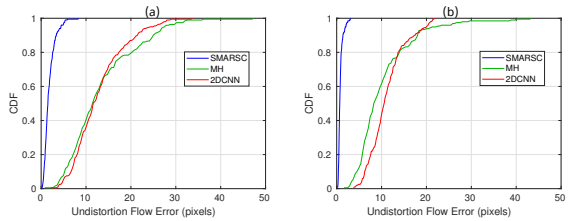

Figure 8. Quantitative comparisons on synthetic RS images with: (a) 6-DOF camera motion and (b) pure rotation.

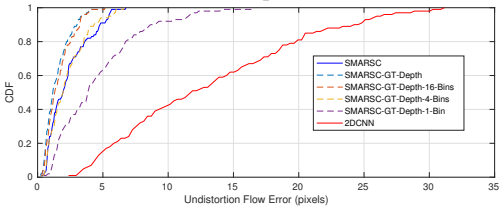

Figure 9. Sensitiveness to depth estimation accuracy.

depth is not very precise but it still reaps benefits as far as a satisfactory rectification is concerned.

5.2. Real Data

We now test on real RS images captured with a Logitech C920 webcam. The images are resized and cropped to match the image size and intrinsic parameters of training data. For the experiments below, we use our model trained entirely on synthetic data and do not have any fine-tuning with real data.

5.2.1 Single-view RS Correction

As it is difficult to capture real RS images with ground truth GS images, we conduct qualitative comparisons in this experiment. Fig. 10 shows some qualitative results of our method (SMARSC), MH, and 2DCNN on real RS images. In general, our method achieves the best overall performance. In particular, we show an evident example of depth-dependent distortions in I_1 with the nearby pole distorted more significantly than the building in the background. For I_1, only our method is able to rectify both the pole and building well. Additionally, we plot the undistortion flow (and depth map) estimated by different methods in Fig. 11, where it can be seen that our undistortion flow (and our depth map) reflects scene structures relatively well. For I_2, all methods are able to recognize those conspicuous distortions, e.g. the distortions of the pole and house highlighted in the red and green box respectively, however, only our method is able to correct the subtle distortions in the blue box. This is also evident in I_3. In particular, a careful inspection will reveal that the car in the red box is actually deformed, and only our method is able to recover a reasonable shape for the car. Similarly, superior performance of our method is observed in I_4. For I_5, we observe that MH returns similar results as ours, due to rich line features in the forms of Manhattan world and probably rotation-dominated camera motion. Please see supplementary material for more results.

5.2.2 SfM with RS Images

We demonstrate another potential application of our RS correction method, which is towards SfM with RS images. For this purpose, we introduce a two-step approach, which first applies our RS correction method (SMARSC) on the input RS images and then employs GS SfM systems (here we use VisualSFM [39]) on the rectified images. We compare our two-step approach against similar two-step approaches which use MH or 2DCNN instead for RS rectification, and a naive approach which applies VisualSFM directly on the original RS images. We use an unordered set of 47 RS images with significant RS distortions collected from the scene I_5 in Fig. 10. Since VisualSFM may return different results at different runs, we run it 10 times for each method.

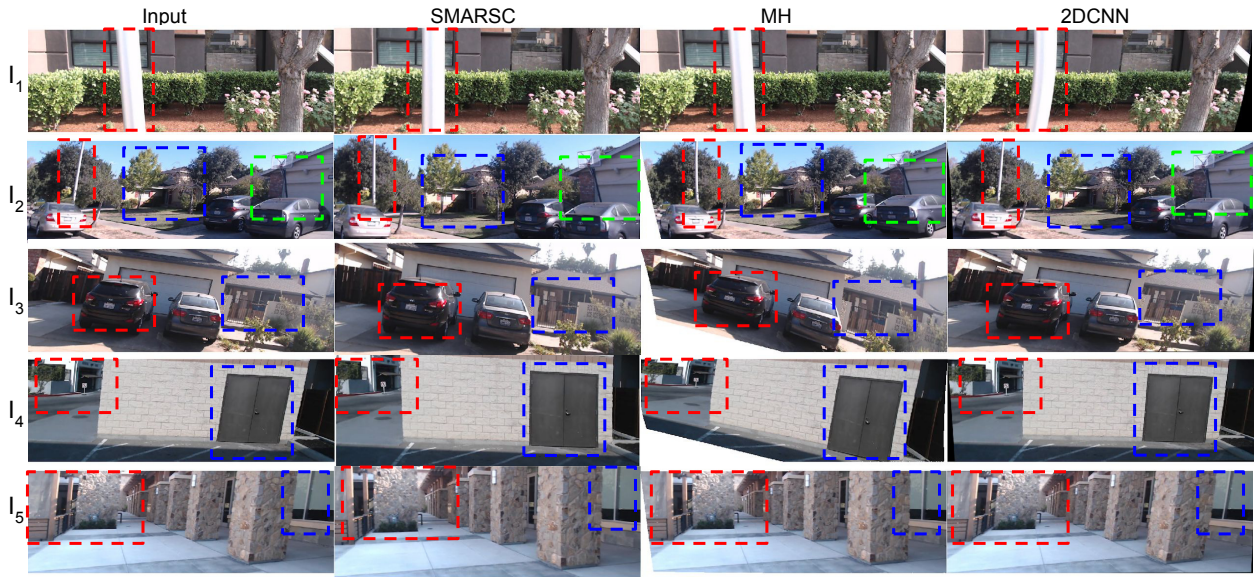

Figure 10. Qualitative comparisons on real RS images. The first column shows different input RS images, while the next three columns plot the results of our method (SMARSC), MH, and 2DCNN respectively.

Figure 11. Undistortion flow (and depth map) estimated by different methods for I_1 in Fig. 10.

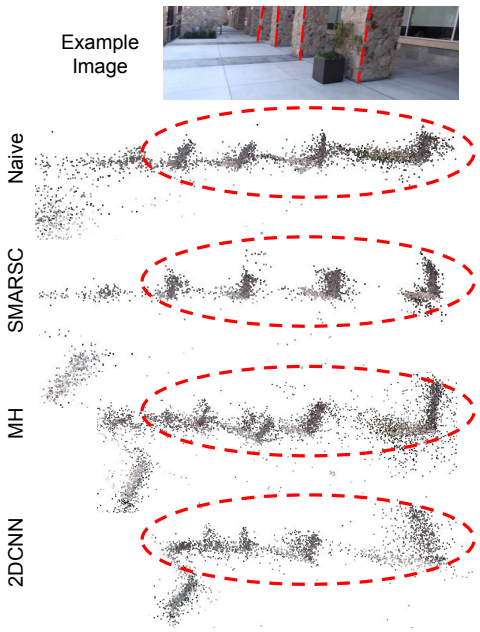

Figure 12. Qualitative results of SfM with RS images in top view. Four equally-spacing pillars are indicated in the top image.

We observe that, among the 10 runs, our method, the naive approach, MH, and 2DCNN fail at **0**, **4**, **7**, and **9** runs respectively. Here, a "failure" means no reasonable structure that reflects the real scene is reconstructed, e.g. the recovered structure is collapsed to a plane. From the results, both MH and 2DCNN break down more often than the naive approach, likely because they introduce more geometrical distortions in their rectified images than those existing in the original RS images. This can mostly be caused by their simplified motion models. In contrast, our method successfully reduces RS effects in the original RS images and is thus more robust than the naive approach. In addition, Fig. 12 shows typical successful reconstructions by different methods in top view. We note that the scene in view consists of equally-spacing pillars, which can be better observed in our result than in those of the other methods. We attribute this to the geometrical faithfulness of our RS correction method.

6. Conclusion

In this paper, we identify and discuss a degenerate case in RS two-view geometry and propose a novel CNN-based approach for single-view RS correction, which is guided by the underlying geometrical properties of the problem. Our method achieves superior performance compared to other state-of-the-art methods for single-view RS correction on both synthetic and real data. Our future work will exploit multiple views and consider other artifacts such as motion blurs that frequently arise in RS images.

Acknowledgments. This work was done during B. Zhuang's internship at NEC Labs America. B. Zhuang and L. F. Cheong also acknowledge the support by the Singapore PSF grant 1521200082.

References

[1] Gilad Adiv. Inherent ambiguities in recovering 3-d motion and structure from a noisy flow field. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (5):477–489, 1989. 5

[2] Omar Ait-Aider and François Berry. Structure and kinematics triangulation with a rolling shutter stereo rig. In *ICCV*, pages 1835–1840. IEEE, 2009. 2

[3] Cenek Albl, Zuzana Kukelova, and Tomas Pajdla. R6p-rolling shutter absolute camera pose. In *CVPR*, pages 2292–2300, 2015. 1, 2, 3

[4] Cenek Albl, Akihiro Sugimoto, and Tomas Pajdla. Degeneracies in rolling shutter sfm. In *ECCV*, pages 36–51. Springer, 2016. 1, 2

[5] Peter N Belhumeur, David J Kriegman, and Alan L Yuille. The bas-relief ambiguity. *International journal of computer vision*, 35(1):33–44, 1999. 5

[6] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. In *CVPR*, pages 5410–5418, 2018. 4

[7] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *ECCV*, pages 628–644. Springer, 2016. 2

[8] Yuchao Dai, Hongdong Li, and Laurent Kneip. Rolling shutter camera relative pose: Generalized epipolar geometry. In *CVPR*, pages 4132–4140, 2016. 1, 2, 3

[9] K Dannilidis and Hans-Hellmut Nagel. The coupling of rotation and translation in motion estimation of planar surfaces. In *CVPR*, pages 188–193. IEEE, 1993. 5

[10] Vikas Dhiman, Quoc-Huy Tran, Jason J Corso, and Manmohan Chandraker. A continuous occlusion model for road scene understanding. In *CVPR*, pages 4331–4339, 2016. 1

[11] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in neural information processing systems*, pages 2366–2374, 2014. 2

[12] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *CVPR*, volume 2, page 6, 2017. 2

[13] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012. 1, 4, 5, 6

[14] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010. 5

[15] Matthias Grundmann, Vivek Kwatra, Daniel Castro, and Irfan Essa. Calibration-free rolling shutter removal. In *ICCP*, pages 1–8. IEEE, 2012. 2

[16] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 1, 3

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 4

[18] Johan Hedborg, Per-Erik Forssén, Michael Felsberg, and Erik Ringaby. Rolling shutter bundle adjustment. In *CVPR*, pages 1434–1441. IEEE, 2012. 1, 2

[19] Berthold KP Horn. Motion fields are hardly ever ambiguous. *International Journal of Computer Vision*, 1(3):259–274, 1988. 5

[20] Sunghoon Im, Hyowon Ha, Gyeongmin Choe, Hae-Gon Jeon, Kyungdon Joo, and In So Kweon. High quality structure from small motion for rolling shutter cameras. In *ICCV*, pages 837–845, 2015. 3

[21] Eisuke Ito and Takayuki Okatani. Self-calibration-based approach to critical motion sequences of rolling-shutter structure from motion. In *CVPR*, pages 4512–4520. IEEE, 2017. 1, 2

[22] Nianjuan Jiang, Zhaopeng Cui, and Ping Tan. A global linear method for camera pose registration. In *ICCV*, 2013. 3

[23] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[24] Yizhen Lao and Omar Ait-Aider. A robust method for strong rolling shutter effects correction using lines with automatic feature selection. In *CVPR*, pages 4795–4803, 2018. 2

[25] Yizhen Lao, Omar Ait-Aider, and Adrien Bartoli. Rolling shutter pose and ego-motion estimation using shape-from-template. In *ECCV*, pages 466–482, 2018. 2

[26] Chi Li, M Zeeshan Zia, Quoc-Huy Tran, Xiang Yu, Gregory D Hager, and Manmohan Chandraker. Deep supervision with shape concepts for occlusion-aware 3d object parsing. In *CVPR*, pages 388–397. IEEE, 2017. 2

[27] Fayao Liu, Chunhua Shen, and Guosheng Lin. Deep convolutional neural fields for depth estimation from a single image. In *CVPR*, pages 5162–5170, 2015. 2

[28] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *CVPR*, pages 4040–4048, 2016. 4

[29] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017. 5

[30] Pulak Purkait, Christopher Zach, and Ales Leonardis. Rolling shutter correction in manhattan world. In *ICCV*, 2017. 2, 5, 6

[31] Vijay Rengarajan, Yogesh Balaji, and AN Rajagopalan. Unrolling the shutter: Cnn to correct motion distortions. In *CVPR*, 2017. 2, 5, 6

[32] Vijay Rengarajan, Ambasamudram N Rajagopalan, and Rangarajan Aravind. From bows to arrows: Rolling shutter rectification of urban scenes. In *CVPR*, pages 2773–2781, 2016. 2

[33] Olivier Saurer, Mare Pollefeys, and Gim Hee Lee. A minimal solution to the rolling shutter pose estimation problem. In *IROS*, pages 1328–1334. IEEE, 2015. 2, 3

[34] David Schubert, Nikolaus Demmel, Vladyslav Usenko, Jörg Stückler, and Daniel Cremers. Direct sparse odometry with rolling shutter. In *ECCV*, pages 699–714, 2018. 3

[35] Deqing Sun, Stefan Roth, and Michael J Black. Secrets of optical flow estimation and their principles. In *CVPR*, 2010. 7

[36] Richard Szeliski and Sing Bing Kang. Shape ambiguities in structure from motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5):506–512, 1997. 5

[37] Subeesh Vasu, Mahesh Mohan MR, and AN Rajagopalan. Occlusion-aware rolling shutter rectification of 3d scenes. In *CVPR*, pages 636–645, 2018. 2

[38] Kyle Wilson and Noah Snavely. Robust global translations with 1dsfm. In *ECCV*, pages 61–75. Springer, 2014. 3

[39] Changchang Wu et al. Visualsfm: A visual structure from motion system. 7

[40] Jiajun Wu, Tianfan Xue, Joseph J Lim, Yuandong Tian, Joshua B Tenenbaum, Antonio Torralba, and William T Freeman. Single image 3d interpreter network. In *ECCV*, pages 365–382. Springer, 2016. 2

[41] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, pages 6612–6619. IEEE, 2017. 4, 5

[42] Bingbing Zhuang, Loong-Fah Cheong, and Gim Hee Lee. Baseline desensitizing in translation averaging. In *CVPR*, pages 4539–4547, 2018. 3

[43] Bingbing Zhuang, Loong-Fah Cheong, and Gim Hee Lee. Rolling-shutteraware differential sfm and image rectification. In *ICCV*, 2017. 1, 2, 3