Occlusion-Aware Rolling Shutter Rectification of 3D Scenes

Subeesh Vasu[1], Mahesh Mohan M. R.[2], A. N. Rajagopalan[3]
Indian Institute of Technology Madras
subeeshvasu@gmail.com[1], ee14d023@ee.iitm.ac.in[2] raju@ee.iitm.ac.in[3]

Abstract

A vast majority of contemporary cameras employ rolling shutter (RS) mechanism to capture images. Due to the sequential mechanism, images acquired with a moving camera are subjected to rolling shutter effect which manifests as geometric distortions. In this work, we consider the specific scenario of a fast moving camera wherein the rolling shutter distortions not only are predominant but also become depth-dependent which in turn results in intra-frame occlusions. To this end, we develop a first-of-its-kind pipeline to recover the latent image of a 3D scene from a set of such RS distorted images. The proposed approach sequentially recovers both the camera motion and scene structure while accounting for RS and occlusion effects. Subsequently, we perform depth and occlusion-aware rectification of RS images to yield the desired latent image. Our experiments on synthetic and real image sequences reveal that the proposed approach achieves state-of-the-art results.

1. Introduction

The world of consumer photography is experiencing a proliferation of low-budget commercial cameras, which are generally built upon CMOS sensors. Most CMOS cameras employ a rolling shutter (RS) mechanism in which the pixels on the sensor plane are exposed in a row-wise manner from top to bottom with a constant inter-row delay. Because of this, images and videos captured using a moving RS camera are often affected by image distortions which are a hindrance to scene understanding. Often, even a small camera motion causes visible geometric distortions in the captured image, since the camera motion experienced by each row can be different from other rows, leading to the so-called rolling shutter effect [15, 31, 30, 29] which is increasingly becoming a common nuisance factor in photography.

Geometric distortions that stem from RS effects violate the properties of the perspective camera model. Removing these distortions from the captured images (also called as RS rectification) will not only create a pleasing visualization, but will also allow us to use the rectified images for applications such as geometric analysis [17] by employing algorithms that are designed to work with images obtained from global shutter cameras.

Several works have attempted problem of RS rectification but under different assumptions. The works in [25, 10, 39] try to remove RS distortions assuming translational motion of the camera. While [9] uses a zooming and translational motion model, [8] employs an affine motion model. An affine RS rectification model for videos shot by cameras attached to moving vehicles is proposed in [3]. While [15] uses a homography mixture model, [13, 31] assumes pure rotational motion of the camera. Works also exist [30, 29] that have proposed single image RS rectification. While [30] attempts to exploit the presence of straight lines in urban scenes, [29] uses features learned through a deep neural network to perform RS rectification.

It must be noted that, none of these works are designed to handle depth-dependent RS distortions induced by 3D scenes. However, there exist works which have attempted to address related problems assuming depth-dependent RS distortions. Examples relevant to our scenario include stereo [32], sparse and dense 3D reconstruction [22, 33] and absolute pose estimation [2, 27] using RS images. However, most of these works assume the availability of camera motion from GPS/INS readings and are not aimed at addressing the problem of image rectification in 3D scenes.

In this paper, we attempt to solve the challenging problem of latent image restoration (*i.e.*, RS rectification) from a set of RS frames, which are affected by depth-dependent geometric distortions as well as occlusion effects. The problem of interest is very ill-posed, since many inter-dependent components such as the camera motion causing the distortions and scene geometry are also unknown. To break the underlying inter-dependencies among these unknowns, we solve for each unknowns in a sequential manner, while ensuring that the individual sub-problems are robust enough to handle the associated dependencies. Following the literature in multi-view stereo and multi-image restoration schemes, we adopt a layered model to represent the underlying 3D scene, wherein we express the 3D scene as a combination of a number of scene planes.

The work that comes close to ours is [45], where a modified differential SfM algorithm to estimate relative pose from two consecutive RS frames was proposed. It was also the first attempt of its kind to propose a depth-aware rectification method that is based on full 3D reconstruction. However, they rely on strong assumptions of small camera motion involving constant acceleration motion and also ignore occlusion effects. In contrast, we specifically address the case of a fast moving camera involving significant occlusion effects in the captured images while not assuming any strong constraints on the nature of underlying camera motion. While SfM algorithms can deliver dense depth maps, the recovered depth map will be erroneous in cluttered environments. Therefore, use of such estimates for RS rectification can lead to visually unpleasing artifacts and local deformations implying that a direct use of RS SfM methods for RS rectification is not the best choice. Whereas, by making use of the piece-wise planar structure of scenes we are able to handle occlusion effects as well as scene clutter to deliver accurate rectification results.

An example illustrating our approach for RS rectification of a two-layer scene is shown in Fig. 1. The input to our algorithm is a set of images captured using a continuously moving RS camera. As is evident from Fig. 1, the vertical post standing close to the camera appears to be more slanted as compared to the background (BG) layer because of the depth-dependent RS effect. To rectify the input image, we not only need to infer the scene geometry, but also must combine pixel intensities from multiple input images to fill in the holes that get generated due to the *intra-frame* occlusion effects (a property exhibited solely by RS cameras). As revealed in Fig. 1, our approach consists of occlusion robust optimization frameworks to sequentially recover the scene structure and occluded layer intensities followed by the final recovery of the desired latent image. Our proposed approach also delivers useful by-products such as camera motion, BG layer intensities, and the scene depth-map. Conventional methods on occlusion-aware image stitching [44], multi-image based occlusion removal [43, 11, 19] etc. relies on GS image formation models, and can leads to failure in the presence of RS distortions. The by-products from our approach can be used as the inputs to these methods thereby extending their scope to RS images as well.

Our main contributions are summarized below.

• This is the first attempt to formally address the problem of recovering the latent image of a 3D scene given a set of observations that are affected by RS distortions and occlusion effects arising from camera motion.

• We advocate a layered image formation model to explain the depth-dependent RS distortions and occlusion effects associated with 3D scenes, and propose an occlusion-aware rectification approach based on this model to recover the underlying latent image.

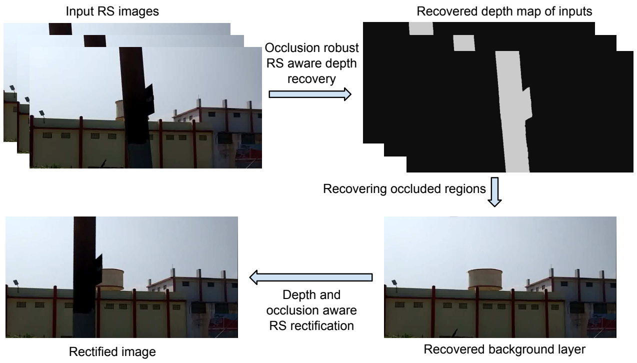

Input RS images
Recovered depth map of inputs
Occlusion robust RS aware depth recovery
Recovering occluded regions
Depth and occlusion aware RS rectification
Rectified image
Recovered background layer

Figure 1. Our approach for occlusion-robust RS-aware scene inference and occlusion-aware RS image rectification.

• Our proposed approach achieves state-of-the-art restoration results, and also delivers potential by-products which can be beneficial for other applications.

2. Image formation model

In this section, we begin with an RS image formation model for the case of planar scenes.

2.1. Rolling shutter imaging for planar scenes

Due to the sequential acquisition in an RS camera, each row in the image gets exposed to the scene during a slightly different interval of time. Therefore, in the presence of camera motion, the camera pose seen by different rows will be different. Consider a sequence of ν number of RS distorted images $R^i|_{i=1}^{\nu}$ of a planar scene captured under continuous camera motion, where each image has N_r rows and N_c columns. If there was no camera motion during the exposure time, then the captured image would have been an undistorted image (*i.e.*, the latent image) F of the scene. Fig. 2 illustrates the acquisition model for images captured using a moving RS camera. As we discussed earlier, if the camera is undergoing motion, camera poses experienced by different rows in R^i will not be the same. Let $\mathbf{p}$ denote the set of 6D camera poses representing the camera motion during the capture of all input images. The camera pose corresponding to each row of R^i will have an associated element in $\mathbf{p}$. The pose vector $\mathbf{p}$ can be viewed as the concatenation of ν smaller pose vectors each corresponding to the individual RS frames. We denote the small pose vector corresponding to R^i as $\mathbf{p}^i$.

Since image acquisition is done in a continuous fashion, to associate the actual camera motion with the rows in an image, we also need to consider the camera motion during the time delay t_b (aka inter-frame time delay) between any two successive frames. To account for the camera motion during t_b, we define n_b number of blank rows [31] and treat the motion associated with an entire RS frame as a vector

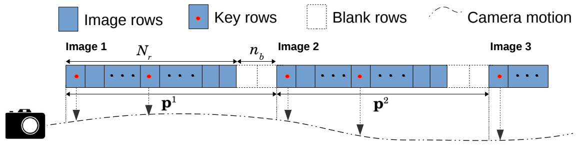


Figure 2. Acquisition model for multiple images captured using an RS camera under continuous camera motion.

of length $N_r + n_b$ (as revealed in Fig. 2). Thus, $\mathbf{p}$ contains the discretized camera poses for a total of $\nu(N_r + n_b)$ rows.

We can now express the j^{th} row of R^i in terms of the camera poses in $\mathbf{p}$ as [13, 30]

$$R^i_j = \{\mathbf{H}(\mathbf{p}^i_j) \circ F\}_j \tag{1}$$

where $\mathbf{p}^i_j$ is the j^{th} pose vector in $\mathbf{p}^i$, $\mathbf{H}(\mathbf{p}^i_j)$ is the homography corresponding to $\mathbf{p}^i_j$, and $\circ$ refers to the geometric warping operation. The above equation can be read as follows: j^{th} row of the RS image R^i can be obtained by taking the j^{th} row of an image obtained by warping the latent image F using the homography $\mathbf{H}(\mathbf{p}^i_j)$. We can relate $\mathbf{H}(\mathbf{p}^i_j)$ to $\mathbf{p}^i_j$ as follows [16]

$$\mathbf{H}(\mathbf{p}^i_j) = \mathbf{K}_q\big(\mathbf{\Theta}^i_j + \frac{1}{d}\mathbf{T}^i_j\mathbf{n}^t\big)\mathbf{K}_q^{-1} \tag{2}$$

where $\mathbf{K}_q$ is the camera intrinsic matrix formed of the focal length q, $\mathbf{\Theta}^i_j$ is the rotation matrix (built from 3D rotation components in $\mathbf{p}^i_j$), $\mathbf{T}^i_j$ is a column-vector representing the 3D translational motion of the camera corresponding to $\mathbf{p}^i_j$, t denotes the transpose operation, and $\mathbf{n}$ and d refers to the normal and distance of the scene plane.

From Eqs. 1,2, a sparse matrix $\mathbf{W}^i_r$ can be created to relate the images R^i and F in a matrix-vector multiplication form as follows [35].

$$\mathbf{r}^i = \mathbf{W}^i_r\mathbf{f} \tag{3}$$

where $\mathbf{r}^i$, $\mathbf{f}$ are the lexicographically arranged column vector forms of R^i and F. $\mathbf{W}^i_r$ is a warping matrix which embeds the pixel motion between R^i and F. Next, we will extend these formulations for the case of layered 3D scenes.

2.2. Rolling shutter imaging for layered 3D scenes

Following the works in [20, 21, 36, 37], we adopt a multilayer model to represent the underlying 3D scene. Let us assume that the scene of our interest comprise L layers, indexed by $l \in \{0, .., L-1\}$, where $l = 0$ represents the BG layer. All the layers follow a depth ordering constraint of the form $d_l > d_{l+1}$, where d_l refers to the distance of scene plane corresponding to l^{th} layer. Using multilayer model, the latent reference image $\mathbf{f}$ can be expressed as [20]

$$\mathbf{f} = \sum_{l=0}^{L-1}\left(\prod_{m=l+1}^{L-1}(1-\boldsymbol{\alpha}_m)\right)\odot\boldsymbol{\alpha}_l\odot\mathbf{f}_l \tag{4}$$

where $\prod$ and $\odot$ refers to element-wise multiplication operation, $\boldsymbol{\alpha}_l$ is an alpha blending mask with value 1 over the spatial support region corresponding to l^{th} layer and 0 otherwise, and $\mathbf{f}_l$ is the intensity image corresponding to l^{th} layer. Note that, in the captured image, pixels from all the layers need not be fully visible since the foreground layers can occlude the layers behind them. In Eq. 4, while α_l denotes the entire mask corresponding to l^{th} layer, the portions of l^{th} layer that are visible in the reference image is actually determined by $\left(\prod_{m=l+1}^{L-1}(1-\boldsymbol{\alpha}_m)\right)\odot\boldsymbol{\alpha}_l$, which accounts for the occlusion effects from other layers too.

Assuming that the appearance and shape of each layer is constant ([4, 12, 42, 1]), the latent image corresponding to a new view point ϑ of the camera can be expressed as

$$\mathbf{g} = \sum_{l=0}^{L-1}\left(\prod_{m=l+1}^{L-1}(1-\mathbf{W}_{\vartheta,m}\boldsymbol{\alpha}_m)\right)\odot\mathbf{W}_{\vartheta,l}(\boldsymbol{\alpha}_l\odot\mathbf{f}_l) \tag{5}$$

where $\mathbf{W}_{\vartheta,l}$ is the warping matrix which embeds the per pixel motion of l^{th} layer between $\mathbf{f}$ and $\mathbf{g}$. Eq. 5 can be treated as the image formation model of a GS camera.

Now, consider the case of a set of images $\mathbf{r}^i|_{i=1}^{\nu}$ captured by a moving RS camera. We can relate $\mathbf{r}^i$ in terms of the latent layer specific intensity ($\mathbf{f}_l$) and mask ($\boldsymbol{\alpha}_l$) images as

$$\mathbf{r}^i = \sum_{l=0}^{L-1}\left(\prod_{m=l+1}^{L-1}(1-\mathbf{W}^i_{r,m}\boldsymbol{\alpha}_m)\right)\odot\mathbf{W}^i_{r,l}(\boldsymbol{\alpha}_l\odot\mathbf{f}_l) \tag{6}$$

where $\mathbf{W}^i_{r,l}$ is the warping matrix which embeds the RS distortions associated with the l^{th} layer in $\mathbf{r}^i$. $\mathbf{W}^i_{r,l}$ can be constructed using a set of homographies $\mathbf{H}_l(\mathbf{p}^i_j)|_{j=1}^{N_r}$ which represents the row-wise varying camera motion involved in the formation of $\mathbf{r}^i$. $\mathbf{H}_l(\mathbf{p}^i_j)$ which can be expressed as a function of $\mathbf{p}^i_j$ and the scene plane parameters (normal vector $\mathbf{n}_l$ and depth value d_l) as

$$\mathbf{H}_l(\mathbf{p}^i_j) = \mathbf{K}_q\big(\mathbf{\Theta}^i_j + \frac{1}{d_l}\mathbf{T}^i_j\mathbf{n}^t_l\big)\mathbf{K}_q^{-1} \tag{7}$$

Since foreground (FG) layers can occlude the layers behind them, only a part of each layer will be visible in a captured RS frame $\mathbf{r}^i$. Apart from Eq. 6, we can model such occlusion effects in terms of *occluded layer mask* $\tilde{\boldsymbol{\alpha}}^i_{r,l}$ as follows.

$$\mathbf{r}^i = \sum_{l=0}^{L-1}\tilde{\boldsymbol{\alpha}}^i_{r,l}\odot\mathbf{W}^i_{r,l}\mathbf{f}_l \tag{8}$$

where

$$\tilde{\boldsymbol{\alpha}}^i_{r,l} = \mathbf{W}^i_{r,l}\boldsymbol{\alpha}_l\odot\prod_{m=l+1}^{L-1}(1-\mathbf{W}^i_{r,m}\boldsymbol{\alpha}_m) \tag{9}$$

The elementary difference between Eq. 6 and Eq. 8 is that, the former uses the same mask to relate a single layer in all RS frames whereas the latter uses different masks for different frames. An example with detailed illustration of different components in our image formation model is provided in the supplementary material. Now, we will discuss our proposed approach for RS rectification which is built on the image formation model discussed thus far.

3. Rolling shutter rectification

Our objective is to recover the latent image $\mathbf{f}$, from a given set of RS distorted frames $\mathbf{r}^i|_{i=1}^{\nu}$ (where ν denotes the number of RS distorted frames) captured under continuous camera motion. This problem is heavily ill-posed since the only known information is the RS distorted frames, and to arrive at the latent image, we also need to solve for the camera motion ($\mathbf{p}$) along with layer-specific plane parameters (d_l, $\mathbf{n}_l$), mask ($\boldsymbol{\alpha}_l$), and intensities ($\mathbf{f}_l$). Therefore, we divide the original problem into a number of sub-problems and sequentially solve for each unknown.

3.1. Camera motion estimation and depth recovery

Our first task is to recover the camera motion involved in all the input frames ($\mathbf{p}$), along with the number of depth layers (L), and the associated plane parameters ($d_l, \mathbf{n}_l$). We estimate all the above-mentioned unknowns by making use of the optical flow correspondences [26] computed across consecutive frames. Let the point correspondences between two consecutive frames be $\mathbf{x} \leftrightarrow \mathbf{x}'$. Let Ψ denote the set of all point correspondences, Ψ_l be a *subset* from the set of all points correspondences that belongs to l^{th} layer, and Ψ_o denote the set of outlier/erroneous correspondences in the optical flow estimate. In practice, the outlier correspondences get generated either because of local errors in optical flow estimate or because of the non-existence of corresponding points (due to occlusion effects) across the image pair.

We will now seek the camera trajectory $\mathbf{p}$ spanning the whole exposure period of all the input frames, assuming that sufficient number of layer correspondences can be obtained from the optical flow. Since the number of unknown poses in p is too high, we use a key-row interpolation approach [31] that treats only the camera poses at certain rows (called as key-rows) as unknowns. We use a total of n_k equally spaced key-rows over the entire camera trajectory, thereby reducing the number of unknown poses from $Z(M + n_b)$ to n_k. The camera pose corresponding to an arbitrary row is obtained by interpolating the poses corresponding to the two adjacent key-rows. We apply linear interpolation for translations and spherical linear interpolation for rotations [34]. The following form of optimization is used to solve for the desired camera trajectory $\mathbf{p}$, and plane parameters ($\mathbf{n}_l$, d_l) corresponding to all the layers.

$$\arg\min_{\mathbf{p},\mathbf{n}_l,d_l} \sum_{i=1}^{\nu-1} \left(\sum_{l=0}^{L-1} \sum_{<\mathbf{x},\mathbf{x}'>\in\Psi_l} c(\mathbf{x},\mathbf{x}') \right) + \left(\sum_{<\mathbf{x},\mathbf{x}'>\in\Psi\setminus\Psi_l} \min\left(\min_l c(\mathbf{x},\mathbf{x}'), \gamma_o \right) \right) \quad (10)$$

where

$$c(\mathbf{x},\mathbf{x}') = ||\mathbf{H}_l(\mathbf{p}^i(\mathbf{x}))^{-1}\mathbf{x} - \mathbf{H}_l(\mathbf{p}^{i+1}(\mathbf{x}'))^{-1}\mathbf{x}'||_2 \quad (11)$$

where $\mathbf{p}^i(\mathbf{x})$ is the pose vector from the camera trajectory $\mathbf{p}^i$ corresponding to the row index of $\mathbf{x}$, γ_o is a scalar constant determining the cost for an outlier correspondence, and $c(\mathbf{x},\mathbf{x}')$ is the cost for a true correspondence. It is straightforward to see that the contribution from c is minimum for a specific camera trajectory which will take each pair from true correspondences to a common point in the latent reference frame, which is in fact the camera trajectory which we seek for.

To form the above cost function, we divide the available correspondences into two mutually exclusive classes, wherein the class Ψ_l comprises layer correspondences which are guaranteed to contain no outliers whereas $\Psi\setminus\Psi_l$ contains all doubtful correspondences which can either be a true correspondence or an outlier. The second term in Eq. 10 is used to assign an outlier robust cost to all doubtful correspondences. The cost for each pixel in $\Psi\setminus\Psi_l$ is determined as the minimum error value incurred by assigning it to any one of the layers or an outlier. Note that, the second term in Eq. 10 itself is sufficient enough to estimate the unknowns, since it contains the cost associated with both the true correspondences and outliers. However, the introduction of the first term allows us to regularize the entire optimization based on the knowledge of true correspondences ensuring faster convergence.

Optimization details: To begin with, the only known information to solve Eq. 10 is the set of all point correspondences Ψ (obtained from optical flow). Since the number of layers is also unknown, we first solve Eq. 10 assuming that there exists only a single layer, and no true layer correspondences are known. By doing so, the optimization in Eq. 10 will give the plane parameters corresponding to the most dominant layer (*i.e.*, the layer with most true correspondences across all input images) and the underlying camera motion. At the end of this optimization, the correspondences in Ψ with cost lower than γ_o is classified as the true correspondences for the dominant layer. Next, we will find the plane parameters and true correspondences of the next dominant layer by solving Eq. 10 in a similar fashion, but using the current estimate of camera motion and the correspondences obtained after removing the true correspondences of the first dominant layer. This process is repeated until it is impossible to identify sufficient number of true correspondences for a unique layer. This kind of an iterative optimization will also help us to identify the number of layers in the scene. Finally, a joint optimization for all the unknowns is done by solving Eq. 10, but with the initial estimates obtained from previous stages. For the final joint optimization, the true layer correspondences obtained earlier is used to compute the cost for the first term in Eq. 10 and the remaining correspondences are used for the second term. A special case of our motion estimation approach is when the input images contain no depth-dependent RS distortions, wherein the algorithm will return the number of layers as one.

The estimates of $\widehat{\mathbf{p}}, \widehat{\mathbf{n}}_l, \widehat{d}_l$ obtained from the joint optimization using Eq. 10 can be directly used to construct the motion matrices $\mathbf{W}^i_{r,l}$ mentioned in Eq. 6. Also, the depth values obtained for each plane in the scene can be used to determine the ordering of the depth layers. Next, we will discuss our approach for recovering the layer masks based on the knowledge of motion matrices and layer ordering.

3.2. Occlusion-robust recovery of layer masks

In this section, we will discuss our approach to estimate the layer masks by making use of the motion matrices obtained from Section 3.1. We will first solve for the occluded masks $\tilde{\boldsymbol{\alpha}}^i_{r,l}$ corresponding to each input image, and are then combined to form final estimates of full layer masks $\boldsymbol{\alpha}_l$.

3.2.1 Occluded layer mask estimation

We employ a multi-label optimization approach to solve for the occluded layer masks $\tilde{\boldsymbol{\alpha}}^i_{r,l}$. Note that $\tilde{\boldsymbol{\alpha}}^i_{r,l}$ is defined with respect to a RS frame $\mathbf{r}^i$, and it represents the spatial support of l^{th} layer that is visible in $\mathbf{r}^i$. We assign one label for each layer in $\mathbf{r}^i$ and solve for the label assignment for every pixels in $\mathbf{r}^i$ using graph cuts [7, 23, 6]. We use the following form of cost function to express desired properties of the labeling.

$$C(\beta_u) = D_u(\beta_u) + \lambda_s \sum_{u' \in \mathcal{N}_u} S_{u,u'}(\beta_u, \beta_{u'}) \tag{12}$$

where $D_u(\beta_u)$ is the data cost to assign the label β_u to pixel u, $\mathcal{N}_u$ is a neighborhood of pixels around u, $S_{u,u'}(\beta_u, \beta_{u'})$ is the spatially varying smoothness cost to assign the labels (β_u, $\beta_{u'}$) to the adjacent pixels (u, u') and λ_s is the scalar weight on the smoothness term.

The smoothness cost $S_{u,u'}(\beta_u, \beta_{u'})$ that we employ has the following form.

$$S_{u,u'}(\beta_u, \beta_{u'}) = \mu_u(1 - k^{|\beta_u - \beta_{u'}|}) \tag{13}$$

where μ_u is an edge aware weight defined at pixel u as

$$\mu_u = \begin{cases} 0, & \text{if } |\nabla \mathbf{r}^i(u)| > \mu_t \\ 1, & \text{otherwise} \end{cases} \tag{14}$$

where $\nabla \mathbf{r}^i(u)$ is the gradient of the image $\mathbf{r}^i$ at u and μ_t is the threshold used to remove the weak gradients that are unlikely to be a part of object boundary. Thus the term μ_u helps to relax the smoothness constraint along the boundaries of different layers in $\mathbf{r}^i$.

The data cost that we use for l^{th} layer of $\mathbf{r}^i$ is given by

$$\sum_{\substack{i_1=1 \\ i_1 \neq i}}^{\nu} \mathbf{v}(i_1, i, l, u) || \{ \mathbf{W}^i_{r,l} (\mathbf{W}^{i_1}_{r,l} \circ^{-1} \mathbf{r}^{i_1}) - \mathbf{r}^i \}(u) ||_1 \tag{15}$$

where $\circ^{-1}$ refers to the inverse warping operation and $\mathbf{v}(i_1, i, l, u)$ is the visibility map defined on the coordinates of $\mathbf{r}^i$. It is straight forward to see that, for layer l, the above equation assigns the sum of L_1 norm between the pixel values from $\mathbf{r}^i$ and its neighboring frames. For layer l, the warping across frames are done by assuming that all the pixels belongs to layer l. Hence the data cost ensures that the label assignment obtained by minimizing Eq. 12 respect the relation in Eq. 8. To avoid errors in data cost that arises from layer occlusions we use the visibility map $\mathbf{v}(i_1, i, l, u)$ [38, 21, 14] defined on the coordinates of $\mathbf{r}^i$. In the binary valued visibility map $\mathbf{v}(i_1, i, l, :)$, a value of 0 is assigned to the pixels in the l^{th} layer of $\mathbf{r}^i$ that get occluded when we warp them to the coordinates of $\mathbf{r}^{i_1}$. Formally, $\mathbf{v}(i_1, i, l, u)$ can be computed as

$$\mathbf{v}(i_1, i, l, u) = \prod_{m=l+1}^{L-1} (1 - \boldsymbol{\varphi}(i_1, i, m, l, u)) \tag{16}$$

where

$$\boldsymbol{\varphi}(i_1, i, m, l, :) = \mathbf{W}^i_{r,l} \mathbf{W}^{i_1}_{r,l} \circ^{-1} \mathbf{W}^{i_1}_{r,m} \mathbf{W}^i_{r,m} \circ^{-1} \tilde{\boldsymbol{\alpha}}^i_{r,m} \tag{17}$$

Interestingly, the visibility map estimation requires $\tilde{\boldsymbol{\alpha}}^i_{r,m}$, whereas to perform an occlusion-robust estimation of $\tilde{\boldsymbol{\alpha}}^i_{r,m}$ we need visibility map, thus creating a chicken-and-egg dependency [14]. To resolve this issue, we will first introduce an additional label (say o) while solving Eq. 12. The label o (typically assigned with a constant value for all pixels) corresponds to the pixels in $\mathbf{r}^i$ for which the labeling cannot be done with high confidence. We will then iteratively solve Eq. 12 and Eq. 16 to find the optimal solution. To solve this alternating estimation effectively, we follow an approach similar to [21]. We progressively freeze the high-confidence label assignments (by assigning their data cost for other labels high) and apply graph cut only on the remaining pixels.[1] The iterative refinement using Eq. 12 and 16 is continued by raising the label cost for o in each step, until there are no pixels in r^i that get assigned to o. By performing such an iterative refinement of label assignment, we can arrive at the desired solution for the binary valued occluded layer mask $\tilde{\boldsymbol{\alpha}}^i_{r,l}$.

[1] Refer supplementary for more details.

3.2.2 Recovering fractional occluded layer mask

In practice, regions around the boundaries of different layers contain a mixture of intensities from both layers resulting in a gradual variation in intensity from one layer to another rather than an abrupt transition. This effect becomes more prevalent when the images are captured using a moving camera, since camera motion can induce motion blur. To accurately model such effects one need to use masks with fractional values along the object boundaries. Hence, to obtain a refined estimate of the occluded layer mask, we solve a regularized form of closed-form matting [24] to yield a fractional layer mask. Note that unlike the way we have defined in the image formation model (Eq. 4), the layer masks will no longer be binary valued and it can have fractional values between 0 and 1. To avoid confusions, we use $\tilde{\boldsymbol{\alpha}}^i_{r,l}$ and $\tilde{\boldsymbol{\varrho}}^i_{r,l}$ to denote the binary and fractional occluded layer mask, respectively.

For the estimation of $\tilde{\boldsymbol{\varrho}}^i_{r,l}$, we first generate a trimap [24] based on the estimate of $\tilde{\boldsymbol{\alpha}}^i_{r,l}$ obtained from Eq. 12. We apply dilation and erosion on $\tilde{\boldsymbol{\alpha}}^i_{r,l}$ to form the uncertain region (*i.e.*, the region in trimap with values between 0 and 1) and replace the values at uncertain regions in $\tilde{\boldsymbol{\alpha}}^i_{r,l}$ with 0.5 to form the trimap $\boldsymbol{\varrho}'$. The final estimate of fractional occluded layer mask $\tilde{\boldsymbol{\varrho}}^i_{r,l}$ is obtained by solving the following form of regularized optimization problem.

$$\arg\min_{\boldsymbol{\varrho}} \boldsymbol{\varrho}^T(\mathbf{L}+\lambda_1\mathbf{A})\boldsymbol{\varrho} + \lambda_2(\boldsymbol{\varrho}-\boldsymbol{\varrho}')^T\mathbf{C}(\boldsymbol{\varrho}-\boldsymbol{\varrho}') \quad (18)$$

where $\mathbf{L}$ is the matting Laplacian matrix [24] derived from $\mathbf{r}^i$. $\mathbf{C}$ is a diagonal matrix whose diagonal elements are zero for all pixels from uncertain regions and one otherwise. The constraint (scaled by a high valued weighting parameter λ_2) enforced through $\mathbf{C}$ is used to ensure that the regions for which the labels are known are classified as it is in the final matte estimate. The diagonal matrix $\mathbf{A}$ (scaled by λ_1) is used to regularize the matte estimation problem based on the prior knowledge about the BG layer intensities.

Since there exists significant translational camera motion across input images, portions of the occluded regions in one input image will be visible in neighboring frames. This allows us to estimate the true BG layer intensity values over the uncertain regions in trimap, despite of the presence of boundary errors in current estimate of $\tilde{\boldsymbol{\alpha}}^i_{r,l}$. We exploit this possibility to borrow the visible sure BG pixels from other images to yield the BG layer intensity value $\xi(u)$ at u in $\mathbf{r}^i$.[2] The diagonal entries of $\mathbf{A}$ (in Eq. 18) is assigned with a nonzero value of $\exp(-\lambda_3||\xi(u)-\mathbf{r}^i(u)||_1)$ at the uncertain regions, and other locations are assigned to 0. Thus, while solving the optimization problem in Eq. 18, at places where the known intensity values in ξ is near to $\mathbf{r}^i$, the values of alpha matte estimate are encouraged to go to 0 through $\mathbf{A}$.

3.3. Estimation of latent layer intensities

Estimation of latent layer intensity $\mathbf{f}_l|_{l=0}^{L-1}$ is done by posing it as a linear least-square problem. To do so, we will express Eq. 8 as a linear relation between the unknowns and known quantities. In $\mathbf{r}^i$, the contribution from l^{th} layer is given by

$$\tilde{\boldsymbol{\varrho}}^i_{r,l} \odot \mathbf{W}^i_{r,l}\mathbf{f}_l = \mathbf{D}_{\tilde{\boldsymbol{\varrho}}^i_{r,l}}\mathbf{W}^i_{r,l}\mathbf{f}_l = \mathbf{B}^i_{r,l}\mathbf{f}_l \quad (19)$$

where $\mathbf{D}_{\tilde{\boldsymbol{\varrho}}^i_{r,l}}$ is a diagonal matrix formed with diagonal entries being the elements in $\tilde{\boldsymbol{\varrho}}^i_{r,l}$, and the motion matrix $\mathbf{B}^i_{r,l}$ embeds both the spatial location and motion of l^{th} layer in the i^{th} RS distorted frame. Let $\tilde{\mathbf{f}} = \left[\ \mathbf{f}^t_0 \ .. \ \mathbf{f}^t_{L-1}\ \right]^t$ and $\mathbf{B}^i = \left[\mathbf{B}^i_{r,0} \ .. \ \mathbf{B}^i_{r,L-1}\right]$, where t denotes the matrix transpose operation. Now we can express Eq. 8 in the form of matrix-vector multiplication as

$$\mathbf{r}^i = \mathbf{B}^i\tilde{\mathbf{f}} \quad (20)$$

We solve the following optimization problem to obtain the layer estimate $\tilde{\mathbf{f}}$.

$$\arg\min_{\tilde{\mathbf{f}}} \lambda_{\mathbf{f}}||\nabla\tilde{\mathbf{f}}||_1 + \sum_{i\in\Gamma}||\mathbf{r}^i - \mathbf{B}^i\tilde{\mathbf{f}}||_2 \quad (21)$$

where Γ denotes the indexes of input images which we use for estimating $\widehat{\mathbf{f}}$ and $\lambda_{\mathbf{f}}$ is a scale factor. In Eq. 21, the second term is the data cost which ensures that the estimate of layer intensities agree with the observed RS images, and the first term is used to regularize the optimization by enforcing natural sparsity on image gradients [40].

Latent layer mask estimation and final image restoration: To obtain the latent layer masks $\boldsymbol{\alpha}_l$, the estimates of $\tilde{\boldsymbol{\varrho}}^i_{r,l}$ from different frames are warped to the reference latent image coordinate and then combined as follows.

$$\boldsymbol{\alpha}_l = \max_{i\in\Gamma} \mathbf{W}^i_{r,l} \circ^{-1} \tilde{\boldsymbol{\varrho}}^i_{r,l} \quad (22)$$

From the available estimates of $\mathbf{f}_l|_{l=0}^{L-1}$ and $\boldsymbol{\alpha}_l|_{l=0}^{L-1}$, we can directly solve for the latent image using Eq. 4. In Eq. 21, since we are solving for a single image (for each layer) which can explain image layers from multiple RS frames, the resulting estimate will contain information from occluding regions in the reference RS frame too ensuring that the recovered latent image will contain no holes.

4. Experiments

In our experiments, the middle frame (among all the input frames) is usually considered as the reference image. If the input RS image does not contain any depth-dependent distortions, then we will no longer require the layered model, and hence our camera motion estimation in Eq. 10 can be directly followed by the image rectification

[2] Refer to supplementary material for more details.

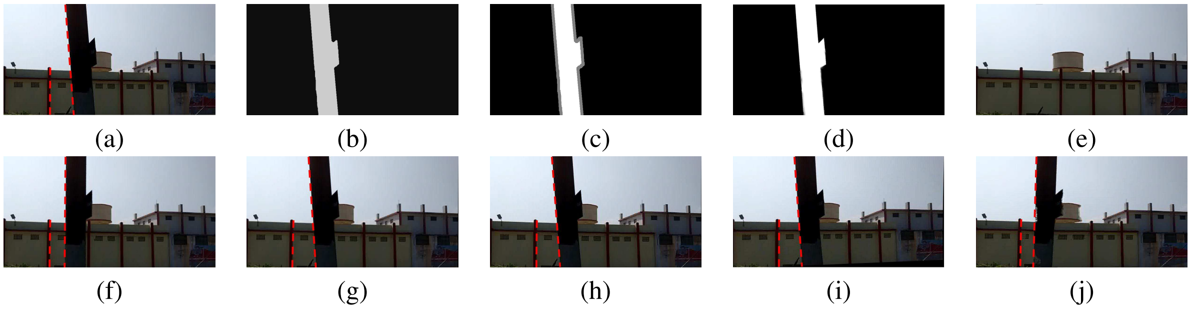


Figure 3. Real experiment: RS rectification of a two layer scene. (a) Reference input image, and (b) recovered scene depth. (c) Trimap used for fractional layer mask recovery, (d) estimated factional layer mask. (e) Recovered latent background layer. Rectified image using (f) proposed method, (g) [31], (h) [30], (i) [29], and (j) [45].

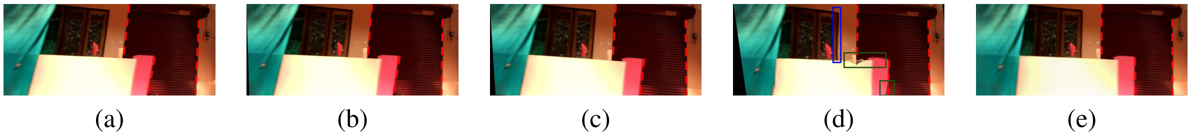


Figure 4. Real experiment, 3 layer scene: (a) Reference image input, Rectified image using (b) [15], (c) [29], (d) [45], and (e) proposed method.

in Eq. 21. We use [41] to obtain the optical flow correspondences, and to solve Eq. 21, we employ alternating direction method of multipliers (ADMM) algorithm [5, 35].

Next, we show the experimental results of our proposed algorithm on both synthetic and real image data. For performance evaluation we compare our method with the state-of-the-art RS rectification methods based on multiple images [31, 15, 45] as well as single image [30, 29].

4.1. Quantitative evaluation

For quantitative evaluation, since there exist no publicly available datasets with GT information, we have generated two different kinds of image datasets (using sample images from [18]) each containing 10 RS image sequences. The first dataset (S_1) contains images with RS distortions involving no depth effects (which simulates the practical scenarios involving RS distortions induced by fronto-parallel planar scenes or camera motions involving pure rotations). The second dataset (S_2) is more relevant to our problem where the images are affected by depth-dependent RS distortions as well as occlusions. We perform the quantitative evaluation using two different metrics; 1. PSNR (in dB) of the rectified image, and 2. average pixel motion error (APME) which is the root mean squared error between the GT motion and the estimated motion (measured in pixel shifts) for all the pixels that are visible in both the GT image and the reference RS image. While PSNR measures the accuracy in terms of the pixel intensities, APME is a measure of the accuracy in camera motion estimation as well as scene geometry. In Table 1, we list the average values of PSNR and APME obtained on our synthetic datasets. As is evident from Table 1, in S_1, since there exist no depth-dependent distortions, the competing multi-image based methods and our approach perform equally well, whereas the ill-posedness of single image based methods lead to severe rectification errors for many examples. In contrast, for S_2, all the competing methods fail in motion recovery as well as rectification, whereas the proposed method shows significant improvement.

Table 1. Performance comparison on synthetic datasets S_1 and S_2

	Dataset S_1		Dataset S_2	
Method	PSNR	APME	PSNR	APME
[15]	29.56	0.93	24.21	7.29
[31]	29.80	0.71	23.89	6.36
[30]	25.33	3.94	23.10	10.17
[29]	26.17	3.25	23.51	8.64
Ours	30.21	0.51	28.90	0.62

4.2. Real data

Here we show the rectification results of the proposed method when applied on real image sequences from existing methods as well as our own dataset. The real examples in our dataset are from a XIAOMI Mi5 mobile phone camera and captured either with a handshake or from a moving vehicle. For our real images, we have used the calibration procedure in [28] to find the number of blank rows (n_b). Due to space constraints, here we provide visual comparisons only for a few representative real examples.

Real examples from our dataset for scenes containing two and three layers are shown in Fig. 3 and Fig. 4, respectively. The scene depth map shown in Fig. 3(b) is formed by combining the depth values obtained from Eq. 10 and the occluded layer mask estimate obtained through iterative

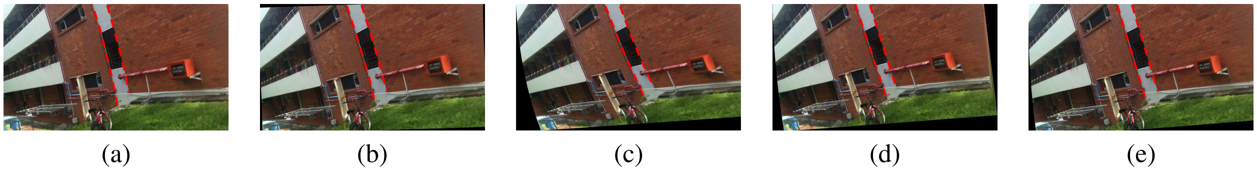


Figure 5. Real example from [45] with RS distortions induced mainly by camera rotations (*i.e.*, no depth-dependent distortions). This is a special case of our approach where our rectification algorithm will boil down to that of a single layer model: (a) Reference input image. Rectified image using (b) [30], (c) [29], (d) [45], and (e) our approach.

graph cut. A trimap (Fig. 3(c)) is generated using the occluded layer mask to recover the fractional occluded mask shown in Fig. 3(d). The fractional layer mask is then used to combine information from multiple images to yield the recovered latent background layer in Fig. 3(e). Note that, this is another potential by-product of our approach, which can aid applications wherein one would like to remove the occluding objects from multiple RS affected images. Comparison for the example in Fig. 3 reveals that, the background dominance leads to a complete failure of existing methods in recovering the motion as well as latent image, whereas the proposed approach estimates the motion accurately and elegantly subsumes information from multiple images to yield occlusion-aware restoration. A similar observation can be made from the example in Fig. 4 too. As is evident from the rectification results in Fig. 3 and Fig. 4, estimation methods in [15, 31, 30, 29] capture the motion closer to that of the background layer which leads to partial rectification of background layers alone. Among the competing methods, the best rectification performance is achieved by [45], which is the only work to attempt estimation of depth-aware motion. The depth estimation and rectification in [45] depends on the optical flow estimates obtained at every pixel. In the real examples of Figs. 3 and 4, significant occlusion effects and lack of sufficient details result in many outliers in the optical flow estimates. These errors along with possible deviation from constant acceleration camera motion assumption leads to partial failure of [45] in terms of the camera motion estimation (as indicated by the slants retained in each scene planes in Figs. 3(j), 4(d)), although [45] is the best among the competing methods in capturing motion from multiple layers. In terms of rectification, at pixel positions where the algorithm in [45] resulted in serious errors in pixel correspondences, the rectified image contains visually unpleasing artifacts/holes (refer to the regions marked with green color in Fig. 4(d)), and at places of small errors the rectified image contains undesired local deformations (refer to the region marked with blue color in Fig. 4(d)). However, our approach is well-designed to handle outliers in optical flow estimates ensuring reliable performance even in cluttered environments. Furthermore, while recovering the intensity from occluding regions is impossible for [45], our occlusion-aware restoration fill-in the occluding regions using information from neighboring frames to achieve significant improvement over all the existing methods.

Finally, in Fig. 5, we show a real example from [45], where RS distortions in input images are induced by dominant camera rotations. For this case the RS distortions are no longer depth-dependent and can be treated as equivalent to our case for a single layer scene. Unlike the case of a fast-moving camera, the rotational motion often leads to curvature distortions in the captured images, wherein the straight lines in the input images will appear as curved in the captured images. As is evident from the comparisons, both our approach as well as [45] is able to rectify the input images satisfactorily, since both these methods are equally well designed to handle this scenario. Due to the ill-posed nature, the single image-based methods in [30, 29] retain minor visible distortions in their rectification results.

In the supplementary material, we provide additional details on our theoretical findings and algorithm implementation, a comprehensive listing of the main steps involved in our image rectification algorithm, additional quantitative evaluations, and visual comparisons on both synthetic as well as real examples.

5. Conclusions

In this paper, we proposed an algorithm to remove RS distortions from images of a 3D scene with special focus on the case of image capture using a fast moving camera where both depth-dependent RS distortions as well as occlusion effects are prevalent. Our proposed approach attempts to sequentially recover the camera motion and scene structure which is then used to perform an occlusion-aware RS rectification of RS images. Quantitative and qualitative experimental results on both synthetic and real RS images reveal that our approach achieves state-of-the-art results on RS rectification of 3D scenes. As a by-product, we could also recover the scene geometry in the presence of RS and occlusion effects, which is the first attempt of its kind. Potential exists to make use of the by-products from our approach for applications such as depth-aware image stitching, multi-image based occlusion removal etc.

Acknowledgements We are grateful to Bingbing Zhuang for sharing the results of [45] for some of the experimental comparisons. The first author thanks Vijay Rengarajan for giving the codes of [30, 29] for the comparisons. We gratefully acknowledge the travel grant support from Google Research India.

References

[1] B. Ahn, T. H. Kim, W. Kim, and K. M. Lee. Occlusion-aware video deblurring with a new layered blur model. *arXiv preprint arXiv:1611.09572*, 2016. 3

[2] C. Albl, Z. Kukelova, and T. Pajdla. R6p-rolling shutter absolute camera pose. In *CVPR*, pages 2292–2300, 2015. 1

[3] S. Baker, E. Bennett, S. B. Kang, and R. Szeliski. Removing rolling shutter wobble. In *CVPR*, pages 2392–2399. IEEE, 2010. 1

[4] L. Bar, B. Berkels, M. Rumpf, and G. Sapiro. A variational framework for simultaneous motion estimation and restoration of motion-blurred video. In *ICCV*, pages 1–8. IEEE, 2007. 3

[5] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011. 7

[6] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE transactions on pattern analysis and machine intelligence*, 26(9):1124–1137, 2004. 5

[7] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on pattern analysis and machine intelligence*, 23(11):1222–1239, 2001. 5

[8] W.-h. Cho and K.-S. Hong. Affine motion based cmos distortion analysis and cmos digital image stabilization. *IEEE Transactions on Consumer Electronics*, 53(3):833–841, 2007. 1

[9] W.-h. Cho, D.-W. Kim, and K.-S. Hong. Cmos digital image stabilization. *IEEE Transactions on Consumer Electronics*, 53(3):979–986, 2007. 1

[10] J.-B. Chun, H. Jung, and C.-M. Kyung. Suppressing rolling-shutter distortion of cmos image sensors by motion vector detection. *IEEE Transactions on Consumer Electronics*, 54(4):1479–1487, 2008. 1

[11] A. Colombari and A. Fusiello. Patch-based background initialization in heavily cluttered video. *IEEE Transactions on Image Processing*, 19(4):926–933, 2010. 2

[12] S. Dai and Y. Wu. Removing partial blur in a single image. In *CVPR*, pages 2544–2551. IEEE, 2009. 3

[13] P.-E. Forssén and E. Ringaby. Rectifying rolling shutter video from hand-held devices. In *CVPR*, pages 507–514. IEEE, 2010. 1, 3

[14] Y. Furukawa, C. Hernández, et al. Multi-view stereo: A tutorial. *Foundations and Trends® in Computer Graphics and Vision*, 9(1-2):1–148, 2015. 5

[15] M. Grundmann, V. Kwatra, D. Castro, and I. Essa. Calibration-free rolling shutter removal. In *Computational Photography (ICCP), 2012 IEEE International Conference on*, pages 1–8. IEEE, 2012. 1, 7, 8

[16] A. Gupta, N. Joshi, C. Lawrence Zitnick, M. Cohen, and B. Curless. Single image deblurring using motion density functions. *ECCV*, pages 171–184, 2010. 3

[17] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 1

[18] H. Hirschmuller and D. Scharstein. Evaluation of cost functions for stereo matching. In *CVPR*, pages 1–8. IEEE, 2007. 7

[19] P.-M. Jodoin, L. Maddalena, A. Petrosino, and Y. Wang. Extensive benchmark and survey of modeling methods for scene background initialization. *IEEE Transactions on Image Processing*, 26(11):5244–5256, 2017. 2

[20] N. Jojic and B. J. Frey. Learning flexible sprites in video layers. In *CVPR*, volume 1, pages I–I. IEEE, 2001. 3

[21] S. B. Kang, R. Szeliski, and J. Chai. Handling occlusions in dense multi-view stereo. In *CVPR*, volume 1, pages I–I. IEEE, 2001. 3, 5

[22] B. Klingner, D. Martin, and J. Roseborough. Street view motion-from-structure-from-motion. In *ICCV*, pages 953–960, 2013. 1

[23] V. Kolmogorov and R. Zabin. What energy functions can be minimized via graph cuts? *IEEE transactions on pattern analysis and machine intelligence*, 26(2):147–159, 2004. 5

[24] A. Levin, D. Lischinski, and Y. Weiss. A closed-form solution to natural image matting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):228–242, 2008. 6

[25] C.-K. Liang, Y.-C. Peng, and H. Chen. Rolling shutter distortion correction. In *Visual Communications and Image Processing 2005*, pages 59603V–59603V. International Society for Optics and Photonics, 2005. 1

[26] C. Liu et al. *Beyond pixels: exploring new representations and applications for motion analysis*. PhD thesis, Massachusetts Institute of Technology, 2009. 4

[27] L. Magerand, A. Bartoli, O. Ait-Aider, and D. Pizarro. Global optimization of object pose and motion from a single rolling shutter image with automatic 2d-3d matching. In *ECCV*, pages 456–469. Springer, 2012. 1

[28] M. Meingast, C. Geyer, and S. Sastry. Geometric models of rolling-shutter cameras. *arXiv preprint cs/0503076*, 2005. 7

[29] V. Rengarajan, Y. Balaji, and A. N. Rajagopalan. Unrolling the shutter: Cnn to correct motion distortions. In *CVPR*, July 2017. 1, 7, 8

[30] V. Rengarajan, A. N. Rajagopalan, and R. Aravind. From bows to arrows: Rolling shutter rectification of urban scenes. In *CVPR*, June 2016. 1, 3, 7, 8

[31] E. Ringaby and P.-E. Forssén. Efficient video rectification and stabilisation for cell-phones. *International Journal of Computer Vision*, 96(3):335–352, 2012. 1, 2, 4, 7, 8

[32] O. Saurer, K. Koser, J.-Y. Bouguet, and M. Pollefeys. Rolling shutter stereo. In *ICCV*, pages 465–472, 2013. 1

[33] O. Saurer, M. Pollefeys, and G. Hee Lee. Sparse to dense 3d reconstruction from rolling shutter images. In *CVPR*, June 2016. 1

[34] K. Shoemake. Animating rotation with quaternion curves. In *ACM SIGGRAPH computer graphics*, volume 19, pages 245–254. ACM, 1985. 4

[35] S. Su and W. Heidrich. Rolling shutter motion deblurring. In *CVPR*, pages 1529–1537. IEEE, 2015. 3, 7

[36] D. Sun, E. B. Sudderth, and M. J. Black. Layered image motion with explicit occlusions, temporal consistency, and depth ordering. In *Advances in Neural Information Processing Systems*, pages 2226–2234, 2010. 3

[37] D. Sun, E. B. Sudderth, and M. J. Black. Layered segmentation and optical flow estimation over time. In *CVPR*, pages 1768–1775. IEEE, 2012. 3

[38] R. Szeliski and P. Golland. Stereo matching with transparency and matting. In *ICCV*, pages 517–524. IEEE, 1998. 5

[39] D. T. Vo, S. Lertrattanapanich, and Y.-T. Kim. Automatic video deshearing for skew sequences captured by rolling shutter cameras. In *2011 18th IEEE International Conference on Image Processing*, pages 625–628. IEEE, 2011. 1

[40] Y. Wang, J. Yang, W. Yin, and Y. Zhang. A new alternating minimization algorithm for total variation image reconstruction. *SIAM Journal on Imaging Sciences*, 1(3):248–272, 2008. 6

[41] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid. DeepFlow: Large displacement optical flow with deep matching. In *ICCV*, Sydney, Australia, Dec. 2013. 7

[42] J. Wulff and M. J. Black. Modeling blurred video with layers. In *ECCV*, pages 236–252. Springer, 2014. 3

[43] X. Xu and T. S. Huang. A loopy belief propagation approach for robust background estimation. In *CVPR*, pages 1–7. IEEE, 2008. 2

[44] F. Zhang and F. Liu. Parallax-tolerant image stitching. In *CVPR*, pages 3262–3269, 2014. 2

[45] B. Zhuang, L.-F. Cheong, and G. Hee Lee. Rolling-shutter-aware differential sfm and image rectification. In *ICCV*, Oct 2017. 2, 7, 8

Occlusion-Aware Rolling Shutter Rectification of 3D Scenes SUPPLEMENTARY MATERIAL

Subeesh Vasu[1], Mahesh Mohan M. R.[2], A. N. Rajagopalan[3]
Indian Institute of Technology Madras
subeeshvasu@gmail.com[1], ee14d023@ee.iitm.ac.in[2] raju@ee.iitm.ac.in[3]

The contents of this supplementary material are arranged as follows. (i) illustration of the main components involved in our image formation model, (ii) additional details on our theoretical findings and algorithm implementation, (iii) a comprehensive listing of the main steps involved in our rectification algorithm, and (iv) additional visual comparisons on both synthetic and real examples. In this supplementary material, numbers of equations and sections without 'S' are with respect to our main paper.

S1. Image formation model revisited

In Fig. S1, the main components of our proposed image formation model are illustrated with the help of a toy example. Here, we have simulated the image formation process of a three layer-scene and explain the differences between the captured images using an RS and a GS camera while undergoing translational motion. The first row contains the elementary units (latent intensities and masks for all the layers) which are used to represent the latent image of a scene. Second and third rows show the layer-wise motion involved in the image formation, if we were to capture the scene from a moving GS camera (*i.e.*, the image formation in Eq. 5). Since all the sensors in a GS camera get exposed to the scene at the same time, an image captured by a moving GS camera can be treated as equivalent to that of an image captured using a single camera pose. In order to generate the image for a given camera pose, we need to warp each layer using a layer-specific homography (Eq. 7) which is again a function of the plane parameters as well as the new camera pose ϑ. It can be observed in Fig. S1 that the pixels from each layer that are visible in the captured image (*i.e.*, pixels that are not occluded FG layers) are decided by the occluded layer masks.

For the case of an RS camera, different rows in the captured image can correspond to different camera poses. Therefore, the homography experienced by a row can potentially be different as compared to another row even if both rows come from the same layer. As shown in the fourth and fifth row of Fig. S1, the RS effect induces depth-dependent geometric distortions (*e.g.*, the vertical lines in the scene get slanted more for the layers closer to the camera) and self-occlusions (since the pixels that are visible in the image captured by a GS camera are not fully visible in the RS image) in the captured images.

S2. Algorithmic details

S2.1. Camera motion estimation and depth recovery

Optimization in Eq. 10 performs an outlier-robust estimation of camera motion and plane parameters, wherein the second term is an outlier robust cost. In each iteration, the outlier robust cost ensures that the optimization is not negatively influenced by correspondences from non-dominant layers. In each step, this optimization by default will classify the correspondences to two classes. The first class will correspond to the most dominant layer while all other correspondences goes to the second class. This leads to a robust estimation of plane parameters in each step, irrespective of the proportion of correspondences in each layer. Also, in our formulation, depth variations within a plane are already factored in. More specifically, the depth-variations induced by inclined planes are subsumed by the plane parameters (normal (**n**) and perpendicular distance (d)).

S2.2. Iterative graph cut and visibility map

Due to space constraints, we provided only a brief discussion on the iterative refinement of occluded layer mask in Section 3.2.1. We will now provide a detailed discussion of our approach for iterative refinement of pixel labeling and visibility map. In the first iteration, to solve Eq. 12, we assign visibility map as 1 for all the pixels, and assign a small scalar value as the cost to o. Thus the solution of Eq. 12 automatically maps the low-confident candidate pixels to label o (which represents the pixels whose label cannot be assigned with high confidence). Since the occluded pixels will have a high cost for all the labels, they get classified under o. The estimate for $\tilde{\alpha}^i_{r,m}$ returned by Eq. 12 is then used to compute a refined visibility map using Eq. 16. The

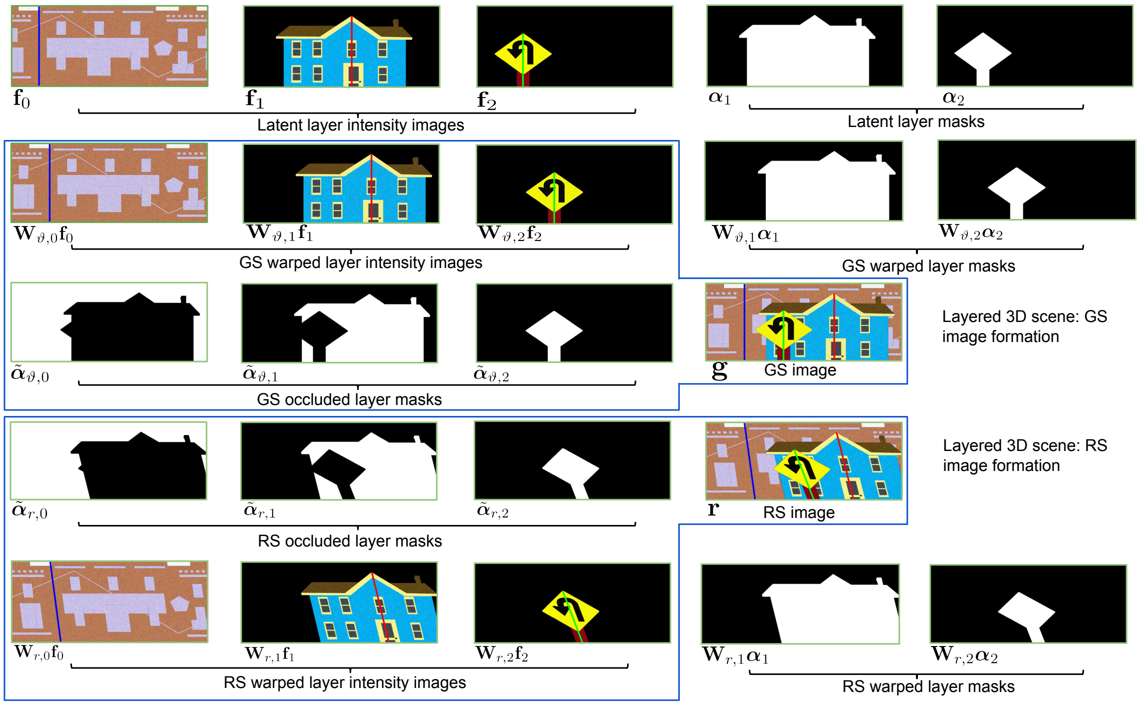

$\mathbf{f}_0$
$\mathbf{f}_1$
$\mathbf{f}_2$
Latent layer intensity images
$\boldsymbol{\alpha}_1$
$\boldsymbol{\alpha}_2$
Latent layer masks
$\mathbf{W}_{\vartheta,0}\mathbf{f}_0$
$\mathbf{W}_{\vartheta,1}\mathbf{f}_1$
$\mathbf{W}_{\vartheta,2}\mathbf{f}_2$
GS warped layer intensity images
$\mathbf{W}_{\vartheta,1}\boldsymbol{\alpha}_1$
$\mathbf{W}_{\vartheta,2}\boldsymbol{\alpha}_2$
GS warped layer masks
$\tilde{\boldsymbol{\alpha}}_{\vartheta,0}$
$\tilde{\boldsymbol{\alpha}}_{\vartheta,1}$
$\tilde{\boldsymbol{\alpha}}_{\vartheta,2}$
GS occluded layer masks
$\mathbf{g}$ GS image
Layered 3D scene: GS image formation
$\tilde{\boldsymbol{\alpha}}_{r,0}$
$\tilde{\boldsymbol{\alpha}}_{r,1}$
$\tilde{\boldsymbol{\alpha}}_{r,2}$
RS occluded layer masks
$\mathbf{r}$ RS image
Layered 3D scene: RS image formation
$\mathbf{W}_{r,0}\mathbf{f}_0$
$\mathbf{W}_{r,1}\mathbf{f}_1$
$\mathbf{W}_{r,2}\mathbf{f}_2$
RS warped layer intensity images
$\mathbf{W}_{r,1}\boldsymbol{\alpha}_1$
$\mathbf{W}_{r,2}\boldsymbol{\alpha}_2$
RS warped layer masks

Figure S1. Layered model of 3D scenes. An example for a scene with 3 layers (with depth ordering $d_0 < d_1 < d_2$) illustrating the image formation in an RS camera. The first row represents the latent layer intensities and masks that we have used to synthesize the RS image $\mathbf{r}$ and latent (or GS) image $\mathbf{g}$. Second and third row simulate the warp effects on the layer intensities and masks in a GS camera. Fourth and fifth row illustrate the warp effects associated with an RS camera. For each case, we show the warped forms of layer intensities, masks, and finally the occluded layer masks. The occluded layer masks decide the visible pixels in the image plane, which when combined with the RS distorted layer intensities generates the captured GS/RS image. It can be observed that, the RS image contains depth-dependent geometrical distortions where the distortion experienced by the layer close to the camera ($\mathbf{f}_2$, $\boldsymbol{\alpha}_2$) is more severe as compared to BG layers.

refined visibility map helps to reduce the effect of outliers (generated due to occlusion effects) in the cost assignment. The visibility map obtained from the first iteration is then used to update the data cost used in Eq. 12. Also, since the pixels that have labels other than o were assigned with high confidence, we do not want to make any changes to these labels. Hence we freeze these pixel labels and then re-do the label assignment for the remaining pixels with the updated data cost. The updating process of both the visibility mask and data cost along with progressive freezing of high-confidence assignments is continued until convergence. In each iteration, we will also increase the occlusion cost to ensure efficient convergence of the entire optimization. In our implementation, we begin by assigning the occlusion cost as 5 and then in each iteration we increase it by a factor of 1.2.

Figs. S3(d-g) show the improvement in the label assignments (for an image sequence from our dataset S_2) as iteration progresses. In each image shown in Figs. S3(d-g), the white pixels correspond to the label o, and other labels are displayed with lower gray values, with the pixel value for BG label being zero. It can be observed that the iterative refinement allows us to handle the negative influences of occlusion effects in label assignment and the mask estimation converges to the desired solution after few iterations.

S2.3. Regularization for fractional mask recovery

As we explained in Section 3.2.2, when there exists significant translational motion of the camera, portions of the occluded regions in one input image will be visible in some other images. We have exploited this possibility to find the background layer intensity values $\xi(u)$ for the pixels in the uncertain region of the trimap. To find the values of $\xi(u)$ we use the following procedure. For each uncertain pixel (say u) in $\mathbf{r}^i$, we identify the lower most layer (say l') among the two layers it can belong to. Now, layer l' represents the background layer and we are looking for the pixel intensities from layer l' at u. For each layer (say l) and input image $\mathbf{r}^i$, we generate a binary mask $\Psi^i_{r,l}$ with ones over the region which will surely enclose all the pixels that belong to l. $\Psi^i_{r,l}$ is generated by assigning the uncertain regions in trimap ϱ' to 1. The visible pixels from l' in other images

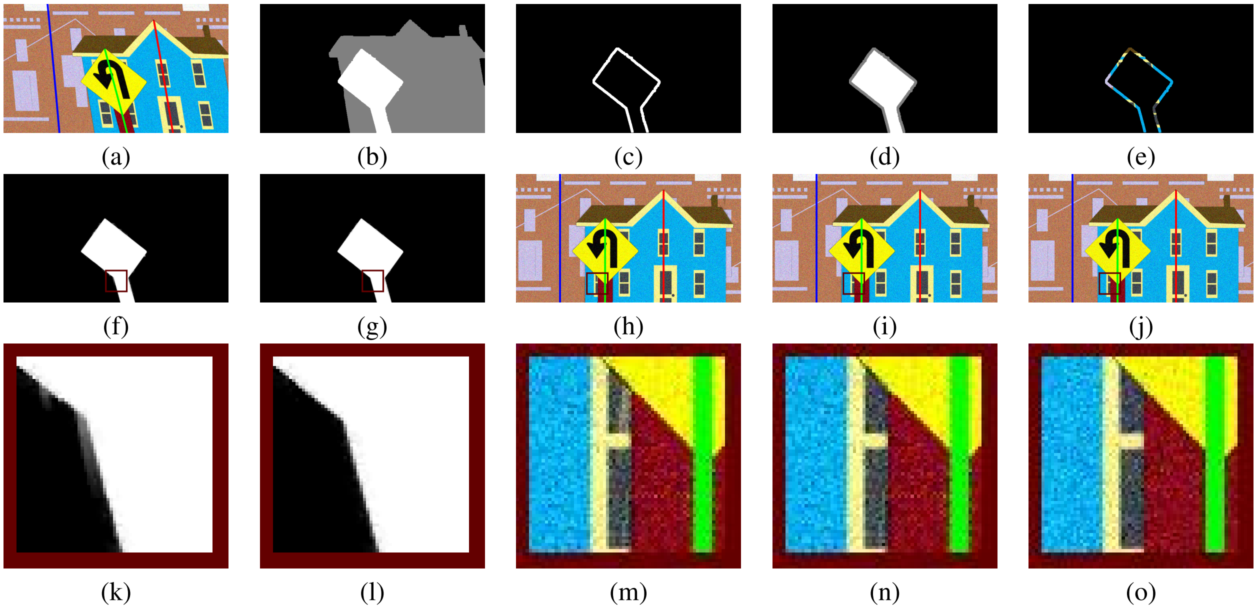


Figure S2. Synthetic experiment: Importance of the proposed regularization for the recovery of fractional layer mask. (a) Reference input image. (b) Occluded layer mask of reference frame obtained through iterative graph cut. For the foremost layer, (c) selected confusion region and (d) generated trimap for mask refinement through alpha matting. (e) Background layer estimate (obtained from neighboring frames) for the regularization of alpha matting. Fractional layer mask obtained through (f) direct alpha matting in [2] and (g) our regularized (using background layer estimate in (e)) alpha matting. Rectified image with mask estimate from (h) direct alpha matting ([2]) and (i) regularized alpha matting. (j) Ground truth latent image. (k-o) Zoomed in patches from (f-j).

are then combined to yield the true intensity value $\xi(u)$ at u. Visible pixels from l' can be found by inspecting the regions in other images that get excluded by the occlusion maps (obtained by applying Eq. 17 on $\Psi^i_{r,l}$) of all the layers l which can potentially occlude the layer l' (*i.e.*, $\forall\ l > l'$). This process is repeated to identify the true intensity values corresponding to all uncertain pixels in $\mathbf{r}^i$.

An example to reveal the importance of our proposed regularization scheme is shown in Fig. S2. Fig. S2(c) shows the uncertain region corresponding to the foremost layer over which we intend to refine the mask values. For the fractional layer mask recovery, we use the trimap shown in Fig. S2(d), and the background layer intensity (Fig. S2(e)) obtained from neighboring frames. To highlight the importance of regularization, we show the fractional masks retrieved with and without regularization in Figs. S2(f-g). It can be observed from the zoomed-in portions shown in Figs. S2(k-l) that our proposed regularization helps to mitigate the errors that generated by the direct use of the approach in [2]. Figs. S2(h-i) show the restored images obtained by using the alpha matte obtained from [2] and our proposed approach, respectively. By comparing with the ground truth image in Fig. S2(j) it can be seen that the error present in the alpha matte obtained from [2] creates visible artifacts along the layer boundaries, whereas the alpha matte obtained using our regularization scheme results in visually pleasing outputs devoid of such artifacts.

S2.4. Additional implementation details

In our experiments, among all the input images $\mathbf{r}^i|_{i=1}^{\nu}$, we choose the middle frame (say e) as the reference RS image. While estimating the camera motion using Eq. 10 we define the camera pose ($\mathbf{p}^e_1$) corresponding to the first row of $\mathbf{r}^e$ as the origin of camera trajectory. The camera motion obtained by enforcing such a condition results in a latent image corresponding to the camera pose $\mathbf{p}^e_1$. For the optimization in Eq. 10, the first row of all the input images are always considered as key-rows. For the cases with the camera motion involving significant translations, we apply RANSAC on all the point correspondences to find the most dominant translational shift and use it to initialize the translational pose vectors for the first row (all other dimensions of these pose vectors are initialized to 0) of every frame except the reference frame. The camera pose at every other key-row is initialized by values obtained through interpolation with respect to the initial pose vectors at the first rows of subsequent frames.

The optimization problems in Eq. 10 and Eq. 18 are solved using MATLAB functions *lsqnonlin* and *mldivide* respectively. We have used empirically found values of n_k, γ_o, k, μ_t, λ_s, λ_1, λ_2, λ_3, $\lambda_{\mathbf{f}}$ in our experiments. For real experiments, the value of focal length (q) was set to 2929.64 pixels (obtained via camera pre-calibration). To build W^i_r, the camera pose of each row is obtained by interpolating from adjacent key-rows (as discussed in Section. 3.1). The

Algorithm 1 Occlusion-aware RS image rectification

```
Input: Set of RS distorted frames $\{R^i\}_{i=1}^{\nu}$.
Output: Camera pose trajectory ($\mathbf{p}$), plane parameters ($\mathbf{n}_l$, $d_l|_{l=1}^{L-1}$), layer masks ($\boldsymbol{\alpha}_l|_{l=1}^{L-1}$), layer intensity images ($\mathbf{f}_l|_{l=0}^{L-1}$), and latent image ($\mathbf{f}$).
Compute optical flow across all consecutive frames. Find the camera motion and plane parameters (and true layer correspondences) of the dominant layer by setting number of layers in Eq. 10 to 1.
Using the estimated camera motion and remaining optical flow correspondences, solve Eq. 10 in an iterative fashion to find the number of layers as well as the plane parameters and the set of true layer correspondences for other layers.
Use estimates from previous steps as initialization to refine the estimates on $\mathbf{p}$, $\mathbf{n}_l$, and $d_l|_{l=1}^{L-1}$ by solving the joint-optimization in Eq. 10.
Estimate occluded binary layer masks $\tilde{\boldsymbol{\alpha}}_{r,l}^i|_{l=1}^{L-1}$ by solving Eq. 12.
Estimate occluded factional layer masks $\tilde{\boldsymbol{\varrho}}_{r,l}^i|_{l=1}^{L-1}$ by solving Eq. 18 .
Estimate latent layer intensity images $\mathbf{f}_l|_{l=0}^{L-1}$ by solving Eq. 21.
Combine warped instances of $\tilde{\boldsymbol{\varrho}}_{r,l}^i|_{l=1}^{L-1}$ according to Eq. 22 to yield latent layer masks $\boldsymbol{\alpha}_l|_{l=1}^{L-1}$.
Find the rectified image $\mathbf{f}$ using Eq. 4.
```

key-rows were spaced equal distance apart and we used 4 key-rows/frame in our experiments. As we discussed in Section 3.1, the iterative identification of plane parameters is repeated until it is impossible to identify sufficient number of true correspondences for a unique layer, wherein the sufficient number was set to 5% of all available correspondences in our experiments. While solving Eq. 10 we use the outlier correspondence cost (*i.e.*, γ_o) as a fixed scalar value of 2. While solving Eq. 12, we use k = 0.4, μ_t as a threshold derived based on the statistics of the overall gradients in the image, and the labels β were assigned with values in the range [1, L+1]. Erosion and dilation operations used in fractional layer mask recovery are done typically by 5 pixels each. For the implementation of various optimization problems in our work we set the parameter values as $\lambda_s = 3$, $\lambda_1 = 2$, $\lambda_2 = 100$, $\lambda_3 = 5$, and $\lambda_{\mathbf{f}} = 0.1$. We have used Sobel operator to find the gradients in Eq. 14 and Eq. 21.

To compare the performance with [1, 5] we have used our own implementation of their algorithm (since the code was not available). For [4, 3] we have used the code provided by the authors, and for [6] we used the results sent by the authors upon request. For the generation of the dataset corresponding to synthetic experiments, we have manually created masks of different shapes to use as spatial support for FG layers. To simulate a scene with L layers, we randomly select L Images from [18] and $L-1$ masks to form latent layer intensity images and latent layer masks (an example is shown in the first row of Fig. S1). A random number generator was used to find the depth, and normal corresponding to each layer. Random third-degree polynomials were used to generate synthetic camera trajectories. We then followed our layered-image formation (as pointed out in Section S1) to generate the synthetic images.

Efficiency: Our unoptimized implementation in MATLAB on an Intel Core i5 CPU takes around 215 seconds to run the entire algorithm for a 3 layer scene with input images of resolution 700 × 400, wherein the individual units for camera motion estimation, occluded layer mask estimation using graph cut, fractional layer mask recovery using alpha matting, estimation of latent layer intensities, and final rectification consumes about 126, 42, 35, 11, and 0.6 seconds respectively.

S3. Overall Algorithm

The main steps involved in our rectification pipeline are listed in Algorithm 1. In Fig. S3 we show few representative intermediate results obtained during rectification of a synthetic example, which provides more clarity on the role of each step in our algorithm. The three middle frames among all the input images used in this experiment are shown in Figs. S3(a-c). The iterative refinement of the occluded layer mask corresponding to the reference frame in Fig. S3(b) is illustrated through Figs. S3(d-g). We repeat this process to find the occluded layer masks for all three input images. At the end of occluded layer mask estimation we have the binary mask associated with the visible pixels for all the layers in all three input images. These binary masks are used to generate the trimap, which is then provided as input to the fractional occluded layer mask estimation unit. The trimap corresponding to the foremost layer of the reference frame is shown in Fig. S3(i), Fig. S3(j) shows the background layer intensities at the uncertain regions of the trimap which we use for regularization of fractional mask estimation. Fig. S3(j) shows the recovered fractional layer mask.

The processing steps involved in the aggregation of masks and intensities from multiple images are shown in Figs. S3(l-r). Figs. S3(l-n) include the fractional mask estimates corresponding to the middle layer of the input images in Figs. S3(a-c), which are then re-warped to the latent image coordinates and combined to yield the full layer mask estimate of the middle layer (Fig. S3(o)). The latent layer intensities of all the layers obtained by solving Eq. 21 are shown in Fig. S3(i). It should be noted that, we can recover the background layer intensities only for those positions for which the pixels are visible in atleast one of the input images. Recovered latent image and the ground truth image are shown in Fig. S3(s) and Fig. S3(t), respectively. A careful inspection around the layer boundaries of the recovered latent image in Fig. S3(t) reveals the presence of holes at some locations. These holes corresponds to the pixels in the latent coordinates that are not visible in *any* of the input images.

S4. Additional quantitative comparisons

To quantitatively compare the performance in terms of the camera motion parameters, we report (in Table S1) the values of root mean squared error of translations (E_t, in pixels), and average geodesic distance of rotations (E_r in degrees) obtained over the synthetic datasets S_1 and S_2.

Table S1. Motion error on datasets S_1 and S_2

	Dataset S_1		Dataset S_2	
Method	E_t	E_r	E_t	E_r
[1]	0.72	0.17	6.31	1.20
[5]	0.55	0.14	5.95	1.31
[4]	2.70	0.41	8.05	2.15
[3]	2.34	0.36	6.89	1.92
Ours	0.33	0.14	0.47	0.16

For quantitative comparison with [6], we have used 5 images from our synthetic data-set S_2 for which the the authors of [6] provide us with the *rectified images* and *motion estimates*. Comparisons for those 5 images are provided in Table S2.

Table S2. Quantitative evaluation on 5 images from dataset S_2

Method	Et	Er	PSNR
[6]	2.41	0.92	25.70
Ours	0.51	0.18	29.53

S5. Additional qualitative comparisons

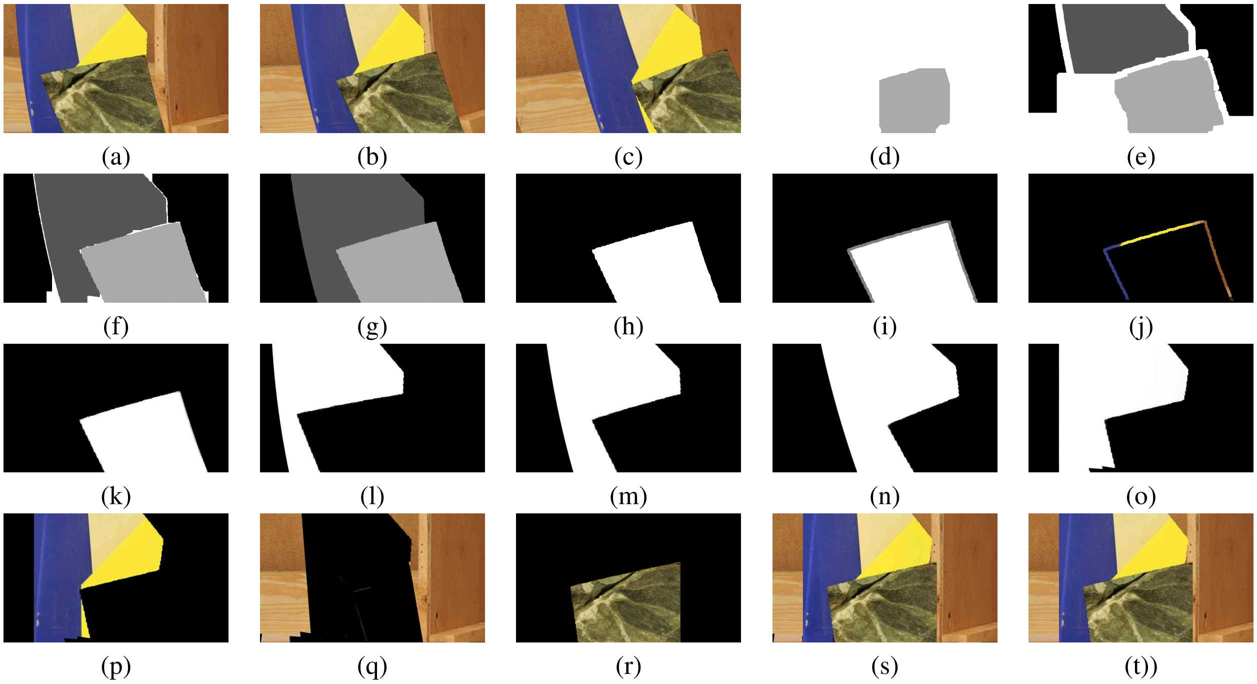
In this section, we provide many more visual comparisons of our rectification algorithm using both synthetic and real examples. Comparisons for three synthetic examples are shown in Fig. S4, where the input RS images are generated to simulate scenes containing two (third row) and three layers (first and second row). While the camera motion for the example in the first row contains dominant translations, the second and third rows contain both camera translations and rotations. Our method integrates information from multiple images to fill in the holes in those regions where the information in the latent image is not visible in the reference input image. In contrast, existing methods do not account for the occlusion effects, and hence attempt to recover the latent image solely from a single RS image, leading to significant errors in terms of latent image restoration. Also, since existing methods do not account for depth and occlusion dependencies, the camera motion estimation itself goes wrong. Consequently, the RS distortions are prevalent in all the layers of the rectified images too. It can be observed that, in few cases, since the background layer is dominant, existing methods end up giving a camera motion estimate close to that of the background layer, resulting in partial rectification of background layer alone.

Similar observations can be found from our real examples in Fig. S5 too, wherein the first two columns correspond to two layer scenes and the third column comprises a three-layer scene. Interestingly, while attempting to rectify the examples in the first and third column of Fig. S5, the single image based methods ([4],[3]) introduces curvature distortions in the rectified images.

In Fig. S6, we have shown results of real examples corresponding to the same scene, but with different kinds of camera motion. As can be observed, the first two columns contain images containing RS distortions induced by dominant-translations, whereas the last two columns contain images captured using dominant in-plane rotations (resulting in curvature distortion in the captured images). For the cases of camera-motion involving dominant translations, existing multi-image based methods ([1, 5]) have partially captured the background motion, whereas the single-image based method in [4] induces undesired curvature distortions. In the absence of depth-dependent distortions (third column of Fig. S6), multi-image based methods deliver accurate rectification results. While the single image based method in [4] fails to rectify, [3] retains minor distortion in the rectified image. In all the examples, our proposed approach is able to deliver accurate rectification results.

Visual comparisons for the case of camera motion involving dominant rotations are provided in Fig. S7. All input images used in Fig. S7 were taken from the dataset provided by [6]. Examples in Fig. S7 indicate that when the depth-dependent RS distortions are absent, multi-image based methods (including ours) are able to deliver accurate results, whereas the performance of single image based methods depends on the nature of the scene. While both single image based methods ([4],[3]) perform well when there exist sufficient number of straight lines in the scene (Fig. S5, first and third row), their rectification fails completely for other scenarios.

To conclude, the performance of existing single image based methods depends on the nature of the scene, since they are designed to work well only for specific classes such as images of urban scenes containing many straight lines ([4, 3]), or face images ([3]) etc. Existing multi-image based methods ([1, 5]) are quite successful in doing rectification for the cases involving no depth-dependent distortions. While none of these methods is successful in removing depth-dependent distortions, our proposed approach can handle the cases of both depth-dependent RS distortions as well as occlusion effects.

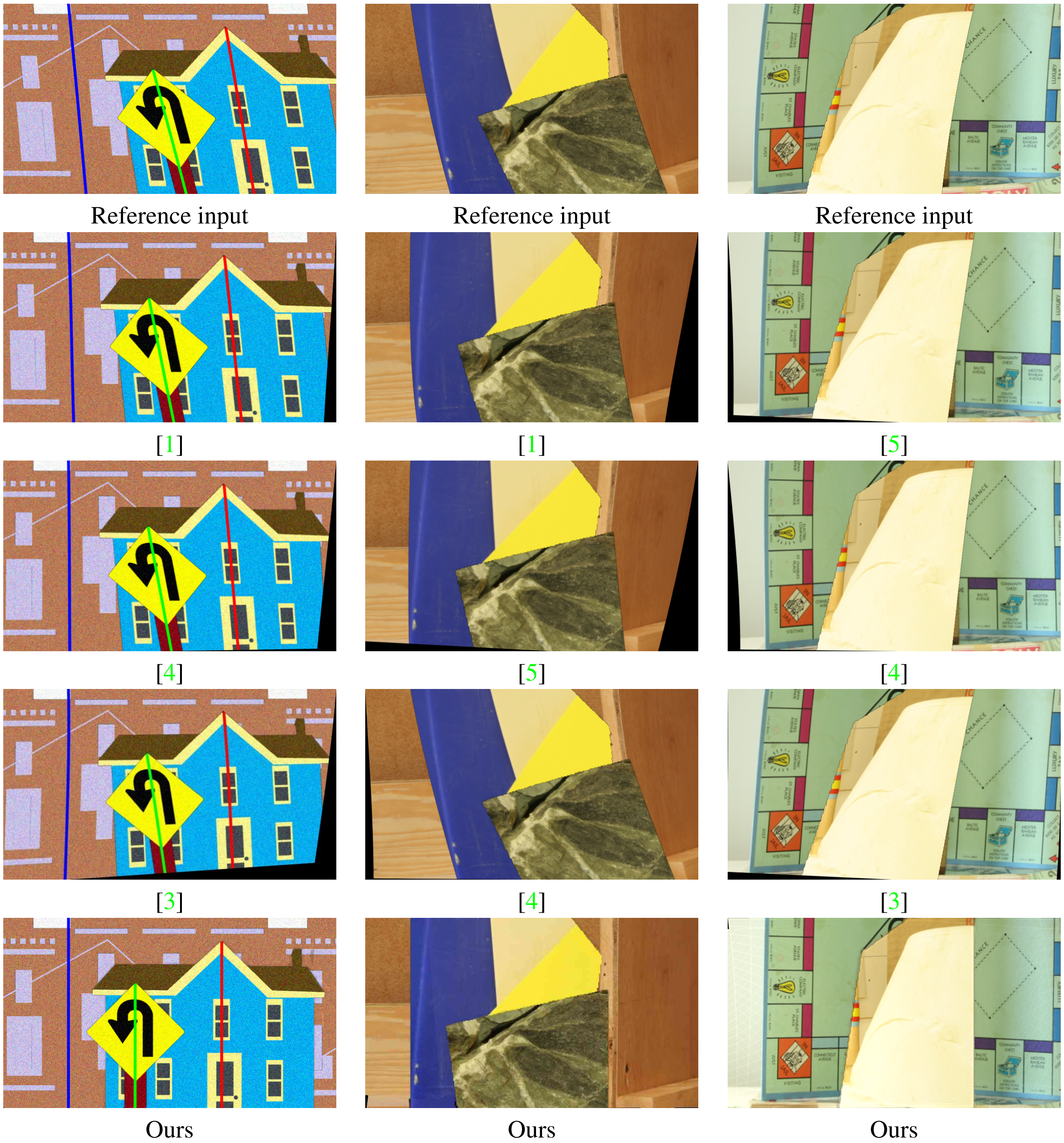

(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

(k) (l) (m) (n) (o)

(p) (q) (r) (s) (t))

Figure S3. Detailed illustration of all the main steps involved in our RS rectification method. Synthetic example of a three layer scene. (a-c) Input images. Occluded layer mask recovery (for the reference image in (b)) through iterative graph cut: Label assignment after (d) 1^{st} iteration, (e) 3^{rd} iteration, (f) 6^{th} iteration, and (g) 11^{th} iteration. Recovering fractional layer mask: (h) Binary layer mask (of the fore-most layer) obtained through graph cut, (i) trimap with non-binary value at confusion region, (j) background layer intensity obtained from neighboring frames for regularization of alpha-matte estimation, and (k) recovered fractional layer mask. Results on the full layer mask recovery of the middle layer: (l-n) Estimated occluded fractional layer masks for RS images in (a-c) and (o) recovered latent full layer mask. Latent layer intensity recovery results for all the layers: (p) middle layer, (q) background layer, and (r) foreground layer. (s) Final rectified image. (t) Ground truth image.



Reference input Reference input Reference input

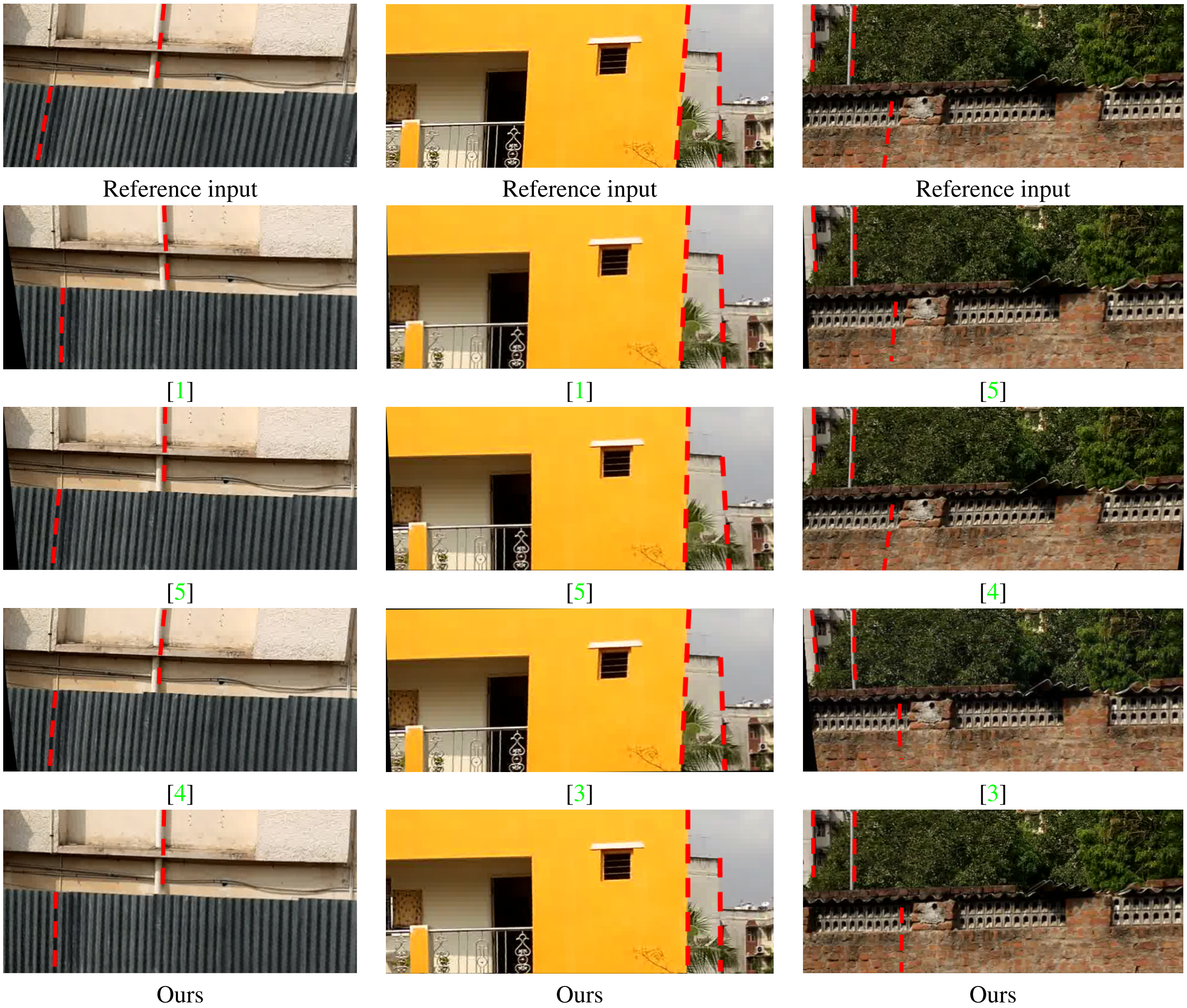
[1] [1] [5]

[4] [5] [4]

[3] [4] [3]

Ours Ours Ours

Figure S4. Synthetic examples: scenes with three (first and second column) and two layers (third column).



Reference input Reference input Reference input

[1] [1] [5]

[5] [5] [4]

[4] [3] [3]

Ours Ours Ours

Figure S5. Real examples for scenes containing two (first and second column) and three layers (third column).

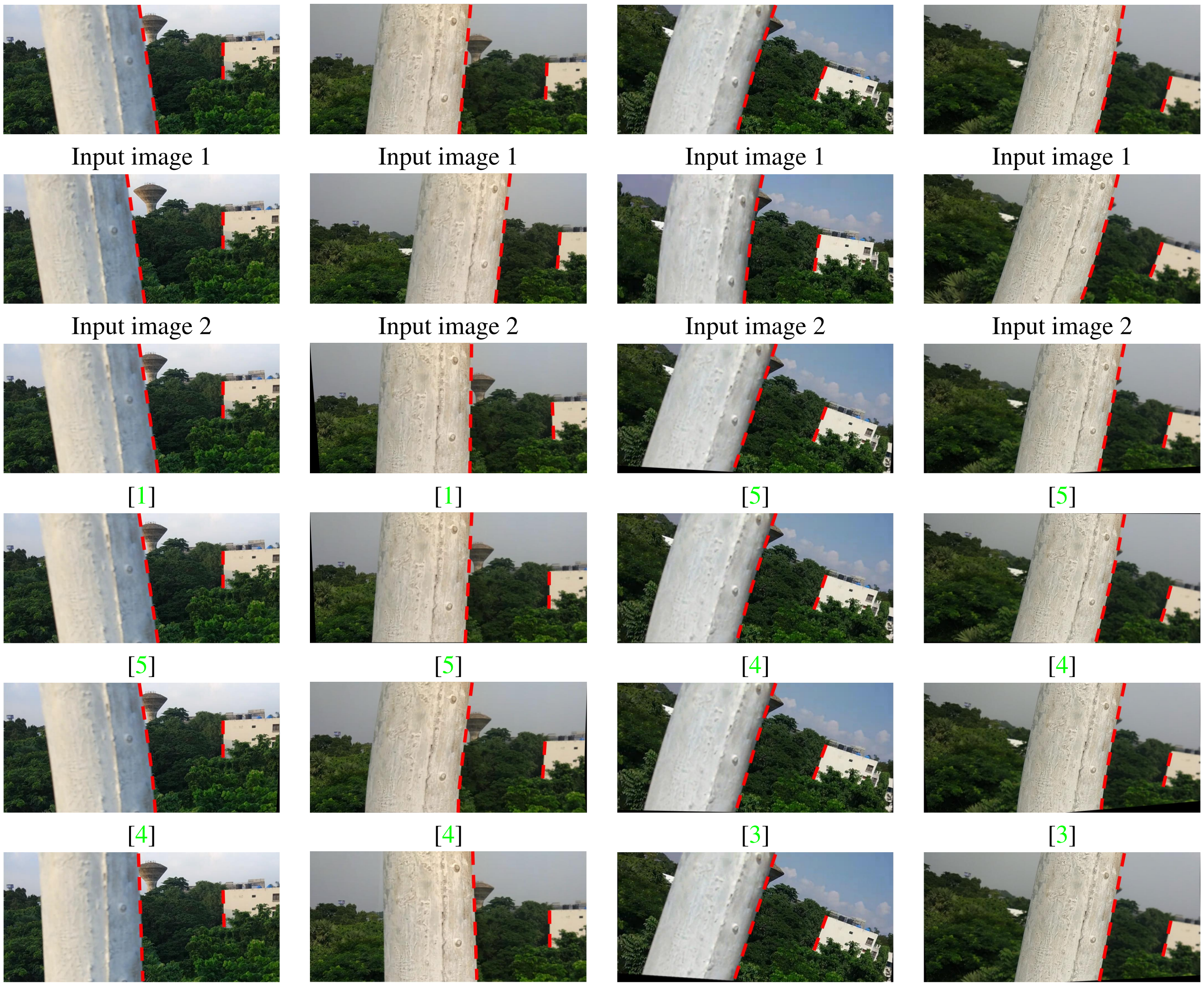
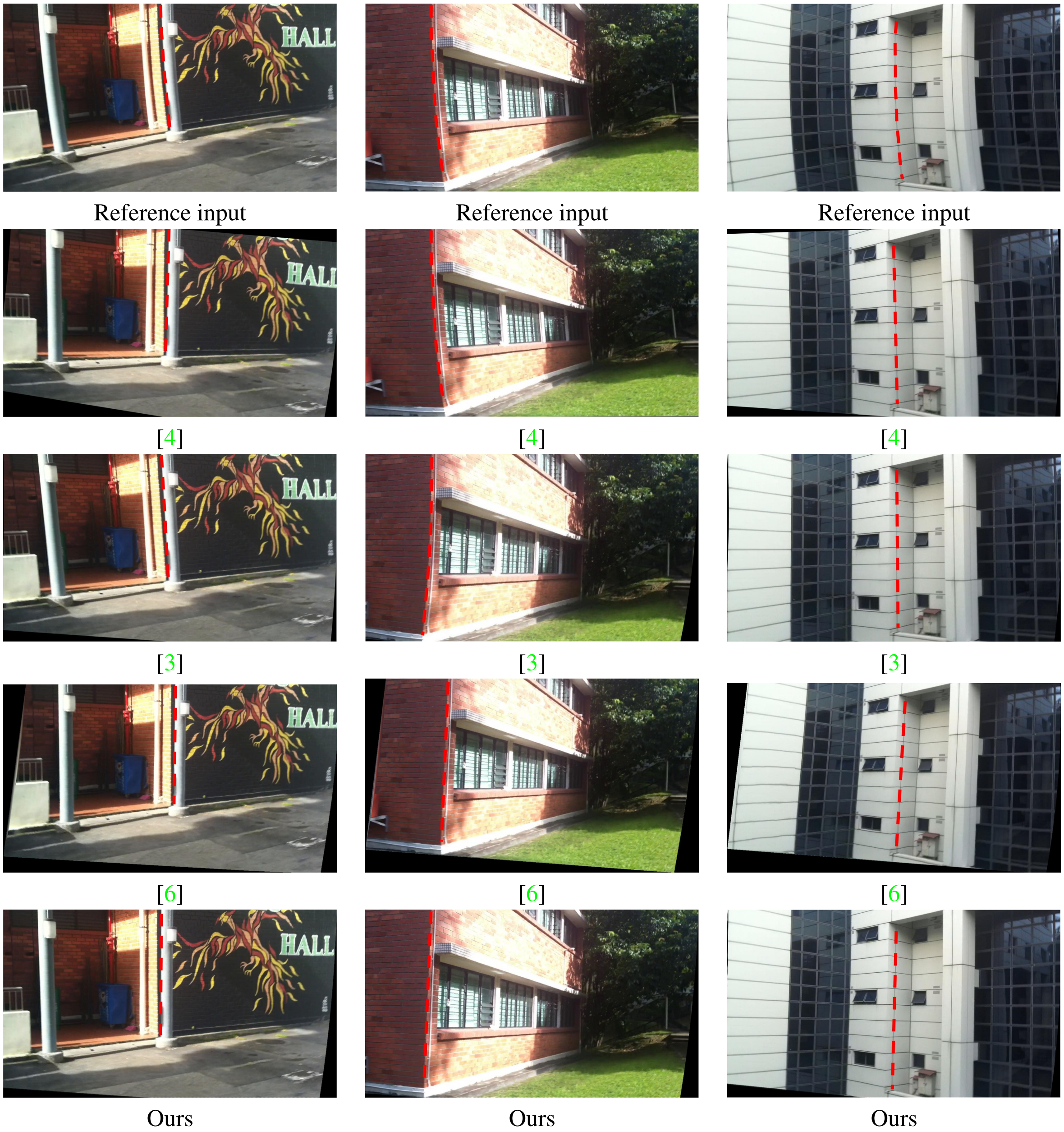


Figure S6. Real examples of a scene containing two layers, for different kinds of camera motion. Camera motion involving dominant translations from left to right (first column), dominant translations from right to left (second column), dominant in-plane rotations (*i.e*. no depth-dependent distortions) in anti-clockwise direction (third column), and dominant in-plane rotations (*i.e*. no depth-dependent distortions) in the clockwise direction (fourth column). For all the examples shown above, the input image shown in the first row is used as the reference frame.



Reference input Reference input Reference input

[4] [4] [4]

[3] [3] [3]

[6] [6] [6]

Ours Ours Ours

Figure S7. Real examples from [6] with camera motion involving dominant rotations (*i.e*. no depth dependent distortions).

References

[1] M. Grundmann, V. Kwatra, D. Castro, and I. Essa. Calibration-free rolling shutter removal. In *Computational Photography (ICCP), 2012 IEEE International Conference on*, pages 1–8. IEEE, 2012. 4, 5, 7, 8, 9

[2] A. Levin, D. Lischinski, and Y. Weiss. A closed-form solution to natural image matting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):228–242, 2008. 3

[3] V. Rengarajan, Y. Balaji, and A. N. Rajagopalan. Unrolling the shutter: Cnn to correct motion distortions. In *CVPR*, July 2017. 4, 5, 7, 8, 9, 10

[4] V. Rengarajan, A. N. Rajagopalan, and R. Aravind. From bows to arrows: Rolling shutter rectification of urban scenes. In *CVPR*, June 2016. 4, 5, 7, 8, 9, 10

[5] E. Ringaby and P.-E. Forssén. Efficient video rectification and stabilisation for cell-phones. *International Journal of Computer Vision*, 96(3):335–352, 2012. 4, 5, 7, 8, 9

[6] B. Zhuang, L.-F. Cheong, and G. Hee Lee. Rolling-shutter-aware differential sfm and image rectification. In *ICCV*, Oct 2017. 4, 5, 10