Depth Sensing Beyond LiDAR Range

Kai Zhang[1] Jiaxin Xie[2] Noah Snavely[1] Qifeng Chen[2]
[1]Cornell Tech, Cornell University [2]HKUST

Abstract

Depth sensing is a critical component of autonomous driving technologies, but today's LiDAR- or stereo camera–based solutions have limited range. We seek to increase the maximum range of self-driving vehicles' depth perception modules for the sake of better safety. To that end, we propose a novel three-camera system that utilizes small field of view cameras. Our system, along with our novel algorithm for computing metric depth, does not require full pre-calibration and can output dense depth maps with practically acceptable accuracy for scenes and objects at long distances not well covered by most commercial LiDARs.

1. Introduction

Depth perception is crucial in autonomous driving for obstacle avoidance, route planning, etc. Existing depth-sensing solutions typically rely on LiDAR, stereo cameras, or time-of-flight sensors [2, 7, 9]. To the best of our knowledge, these systems have significant limits on their maximum ranges. For instance, the recent Waymo [2] and nuScenes [7] self-driving car datasets both feature LiDAR ranges of $\sim$80 meters, while the stereo cameras in KITTI [9] cannot see distant objects in detail because of their wide field of view (FOV) (about 80°).[1] It takes just 3 seconds for a vehicle to travel 80 meters at a speed of 60 miles/h, which is too short a time window in unforeseen emergency situations. While some high-end LiDARs claim to reach a maximum of 300 meters, e.g., Velodyne's Alpha Puck™ [1], these are not only expensive but also produce very sparse point clouds for distant objects due to power and cost constraints. Such limited range can become a critical issue when a self-driving vehicle is a heavily weighted truck, or moving at high speed. The earlier an autonomous vehicle perceives the depth of obstacles on its driving route, the safer the technology is, as an early defensive response can be made in case of emergency.

Hence, there is a need for dense, accurate depth perception beyond the LiDAR range. In this work, we seek longer-range *dense* depth sensing beyond the 200 meter range, which is not well covered by most existing commercial LiDARs for autonomous driving. To that end, we propose a cost-effective solution that utilizes three cameras with small fields of view. Equipped with telephoto lenses, these cameras can perceive faraway scenes or objects. [2] Our novel three-camera setup can resolve geometric ambiguities that arise in stereo systems based on only two small-FOV cameras. For small-FOV stereo cameras, such ambiguities are caused by (1) a small baseline/depth ratio, (2) difficulty in calibrating small-FOV cameras, and (3) maintaining the calibration during usage. Surprisingly, we can solve these problems by adding a specific third camera, without requiring a fully accurate calibration of camera parameters, by using a novel depth disambiguation algorithm. Our proposed three-camera system, along with our depth estimation algorithm, can produce dense depth maps without the need to fully pre-calibrate camera intrinsics and extrinsics. Moreover, it is robust to small vibrations in camera orientations that are inevitable for cameras attached to moving vehicles. We demonstrate the effectiveness of our approach with both synthetic and real-world data. Experiments show that our method can achieve a 3% relative error at a distance of 300 meters in terms of depth estimation accuracy.

In summary, our contributions are three-fold. First, to our knowledge, our approach is the first to address the problem of *dense* depth map acquisition at a range beyond that of most LiDARs in the domain of autonomous driving. Second, we propose a novel camera setup and depth estimation algorithm that requires only partial camera calibration. Third, we validate the effectiveness of our long-range depth-sensing system on both synthetic and real-world data.

2. Problem setup and related work

In this section, we formulate our problem setup, review prior work, and analyze the applicability of relevant existing algorithms to our problem.

For depth estimation at a distance over 200 meters, one seemingly straightforward solution is to construct a stereo

[1]The cameras in the Waymo dataset have a 50-degree horizontal FOV.

[2]Note that our system is not aimed to replace existing short-range LiDAR, but instead to complement it in a cost-effective way because long-range LiDAR sensors are expensive, power-inefficient and only capture sparse depth measurements for distant objects.

camera system with two small-FOV cameras and attach it to the vehicle. However, there are several challenges with such a setup that distinguish it from typical stereo camera setups:

- Because the baseline is restricted by the vehicle width (e.g., 2 meters), the baseline/depth ratio is very small in our problem setup, leading to a narrow triangulation angle for estimating 3D points from image correspondences. Hence the geometric setting of this problem is particularly ill-conditioned.
- Unlike cameras with standard FOVs, small-FOV cameras are near-orthographic when a scene's depth variation is much smaller than its average depth. The absence of strong perspective effects can lead to problems when using standard checkerboard-based calibration of intrinsic and extrinsic stereo camera parameters [27].
- The reduced FOV increases the system's sensitivity to vibrations. A small perturbation in orientation will lead to a noticeable change in image content. Such vibrations are difficult to avoid in real-world moving-vehicle scenarios, even if the stereo camera is rigidly mounted.

A pratical solution to long-range depth estimation with small-FOV cameras must address these challenges. Note that our problem setup also shares similarities with those explored in areas including structure from motion (SfM), structure from small motion (SfSM), and uncalibrated stereo rectification and calibration of cameras with telephoto lenses.

SfM. SfM algorithms aim to automatically recover camera poses from image collections [21, 22, 3, 19]. The minimal case is two-view SfM, which is also an important component of multi-view SfM [5]. Two-view SfM often works through the decomposition of an essential matrix obtained from intrinsically-calibrated images [17], followed by bundle adjustment [24]. However, essential matrix estimation is challenging in our case due to its ill-conditioned geometric setup. Another key issue is the *bas-relief ambiguity* [23, 6] present in SfM when the baseline/depth ratio is small and the camera is near-orthographic. The bas-relief ambiguity can cause unwanted distortions of the reconstructed scene, leading to large depth estimation errors for distant objects.

SfSM. Structure from small motion refers to the SfM problem under small camera motion. Previous work [26, 16, 10, 15] reconstructs scene geometry from video clips with accidental motion caused by handshake. These methods exploit multi-view redundancies in video clips to overcome the high depth uncertainty arising from the small baseline/depth ratio. However, SfSM requires a video clip as input for the sake of abundant redundant observations, which is not suitable for autonomous driving due to the real-time constraint, as well as the presence of moving objects such as vehicles and pedestrians. Moreover, Ha et al. [11] observe that SfSM is also vulnerable to the bas-relief ambiguity, and try to reduce this ambiguity with a separate rotation estimation step. But they assume pre-calibration of camera intrinsics, which is not trivial for small-FOV cameras. In addition, the errors in their estimated rotations are relatively large, considering the tiny triangulation angles involved, yielding inaccurate depth estimates for distant scenes.

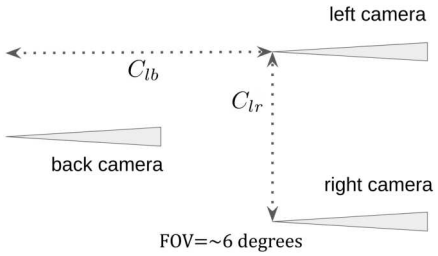


Figure 1: Top-down view of our proposed camera setup. The back camera can be positioned slightly higher than the left and right ones so that its view is not obscured. $C_{lr}, C_{lb} \approx 2$m.

Uncalibrated stereo rectification. Stereo cameras are often pre-calibrated before deployment, allowing for online rectification using calibrated intrinsics and poses. The case when such calibration is unavailable has also been studied, e.g., by Loop and Zhang [18], and Hartley [13]. However, their methods assume that the fundamental matrix is known. As with two-view SfM, such methods will be brittle in the face of the ill-conditioned fundamental matrix estimation problem and the inherent bas-relief ambiguity.

Calibration of cameras with telephoto lenses. Huang et al. [14] equip a pan-tilt camera with a telephoto lens to capture biometric features over a long range. They demonstrate the degeneracy of calibrating a long-focal-length camera with the 2D-2D correspondences from a checkerboard because the perspective effect is weak. They also show that 2D-3D correspondences are essential for calibrating such a camera. Our proposed approach, however, does not require full calibration of camera intrinsics and is more practically convenient. We only need to know the focal length, which can be read out from the lens's specification sheet.

3. Method

Our approach has two major components: the camera setup and the accompanying depth estimation algorithm. Details of both components are provided below.

Our camera setup requires three small-FOV cameras, placed according to Fig. 1. Two of the cameras form a left-right stereo pair, while a third is placed in the back of these two. The left and right cameras are mounted to a vehicle's front, with the back camera on the vehicle's tail. Each camera faces forward along the driving direction. We assume that the three cameras' focal lengths f are the same and known. Furthermore, the distance between the optical centers of the

Algorithm 1: Depth Estimation

Input : left, right, back images; f, C_{lr}, C_{lb}
Output : depth map for left view

1 Pseudo-rectify left, right images
2 Estimate disparity
3 Remove ambiguity in the estimated disparity map
4 Convert disparity to depth, and return

left and right cameras, denoted C_{lr}, is assumed to be known, as well as the distance between left and back cameras C_{lb} along the z-axis. No additional information is required. Like the baseline C_{lr}, the distance C_{lb} should also be as large as possible to benefit subsequent processing. In practice, C_{lr} and C_{lb} are limited by vehicle size.

Our depth estimation algorithm takes the three images captured by our camera system as input and outputs a dense depth map for the left view. Our algorithm, detailed in Algo. 1, is comprised of three modules: pseudo-rectification, stereo matching, and ambiguity removal. The pseudo-rectification step transforms the left and right images with affine warps so that they form a *pseudo*-stereo pair. After pseudo-rectification, a standard stereo matching algorithm can be used to compute a disparity map. This disparity map, however, has an unknown constant offset as a result of pseudo-rectification. This ambiguity is resolved with the help of the back image, and the ambiguity-free disparity map is finally converted to a depth map.

Step 1: Pseudo-rectification. Standard stereo rectification utilizes the intrinsics and relative poses of two cameras to warp the left and right images with homographies such that the epipolar lines are aligned with image x-axis. Pure 2D methods assuming a known fundamental matrix have also been proposed, e.g., by Loop and Zhang [18]. However, in our problem setup, neither the full intrinsics and relative pose are known accurately, nor can the fundamental matrix be reliably estimated in the case of a small baseline/depth ratio. We propose a pseudo-rectification procedure based upon the observation stated in Prop. 1. Our algorithm approximately rectifies the two images with estimated affine transformations, and only depends on the left and right images.

Proposition 1. *When a small-FOV camera is rotated by a small amount, the homography warping the original image to the rotated view is approximately an affine transformation.*

Proof. Let the 3D rotation be $\mathbf{R} = \mathbf{R}_z(\alpha)\mathbf{R}_y(\beta)\mathbf{R}_x(\gamma)$, where α, β, γ are Euler angles. Our proof strategy is to show that the three component-wise rotations all result in approximately affine transformations. First, the homography representing $\mathbf{R}_z(\alpha)$ is always affine. Now let's consider $\mathbf{R}_x(\gamma)$. Let (x, y, z) be a 3D point in the camera coordinate frame, and denote $\frac{y}{z} = \tan\theta$, where θ is the angle between the z-axis and the vector $(0, y, z)$. Suppose (x, y, z) becomes (x', y', z') after $\mathbf{R}_x(\gamma)$ is applied. We then have

$$x' = x, \frac{y'}{z'} = \tan(\theta - \gamma), z' = z\frac{\cos(\theta - \gamma)}{\cos\theta}. \quad (1)$$

Because both the camera FOV and the rotation is small, both $|\theta|$ and $|\theta - \gamma|$ are roughly bounded by $\frac{FOV}{2}$ and hence small. We then have the approximations:

$$\frac{x'}{z'} \approx \frac{x}{z}, \frac{y}{z} \approx \theta, \frac{y'}{z'} \approx \theta - \gamma. \quad (2)$$

We then project the 3D point into image space via $u = f\frac{x}{z} + c_x, v = f\frac{y}{z} + c_y$, where f is the focal length, (c_x, c_y) is the principal point, and (u, v) are pixel coordinates. This gives us

$$u' \approx u, v' \approx v - f\gamma. \quad (3)$$

This indicates that $\mathbf{R}_x(\gamma)$ approximately translates the image along the row axis. By similar logic, one can show that the homography representing $\mathbf{R}_y(\beta)$ is also approximately affine, which completes our proof. □

Algorithm 2: Pseudo-rectification

Input : left and right images
Output : pseudo-rectified left and right images

1 Initialize $\mathbf{H}^{(l)} = \mathbf{H}^{(r)} = [\mathbf{I}_{2\times 2}, \mathbf{0}_{2\times 1}]$
2 Detect matches between left and right images
3 **for** $t = 1 : T$ **do**
4 Randomly sample M matches
5 Solve for candidate $\mathbf{H}^{(l)}_{21}, \mathbf{H}^{(l)}_{22}, \mathbf{H}^{(r)}_{21}, \mathbf{H}^{(r)}_{22}, \mathbf{H}^{(r)}_{23}$
6 If # inliers increases, update the matrix entries
7 **end**
8 Update $\mathbf{H}^{(l)}_{11}, \mathbf{H}^{(l)}_{12}, \mathbf{H}^{(r)}_{11}, \mathbf{H}^{(r)}_{12}$ by enforcing the norm and determinant constraints
9 Update $\mathbf{H}^{(r)}_{13}$ by imposing the disparity constraint
10 Warp left and right images by $\mathbf{H}^{(l)}, \mathbf{H}^{(r)}$, respectively

Algorithmic details of our pseudo-rectification are specified in Algo. 2. We use RANSAC [8] to find a pair of rectifying affine transformations, $\mathbf{H}^{(l)}, \mathbf{H}^{(r)} \in \mathcal{R}^{2\times 3}$, that map corresponding pixels to the same y-coordinates. This y-coordinate constraint only fixes some of the parameters in $\mathbf{H}^{(l)}, \mathbf{H}^{(r)}$. To determine the rest, we need additional constraints. For instance, we choose $\mathbf{H}^{(l)}$ to be rigid, which preserves inter-pixel distances and is important for our disparity disambiguation. Other constraints are also imposed as needed.

In steps 1-2, we initialize two identity affine transformations, and detect sparse feature matches between the left and right images using SURF keypoints [4]. Let the N detected matches be $\{(\mathbf{x}^{(l)}_i, \mathbf{x}^{(r)}_i), i = 1, \ldots, N\}$, where

$\mathbf{x}_i^{(l)}, \mathbf{x}_i^{(r)}$ are homogeneous pixel coordinates in the left and right views, respectively. In steps 3-7, we solve for $\mathbf{H}_{21}^{(l)}, \mathbf{H}_{22}^{(l)}, \mathbf{H}_{21}^{(r)}, \mathbf{H}_{22}^{(r)}, \mathbf{H}_{23}^{(r)}$ with RANSAC, by enforcing that corresponding pixels should have the same y-coordinate in the rectified views. At each RANSAC trial, a subset of M matches is randomly sampled; then we construct a homogeneous linear system of M equations, with each sampled match resulting in one equation,

$$\langle \mathbf{H}_{2,1:3}^{(l)}, \mathbf{x}^{(l)} \rangle - \langle \mathbf{H}_{2,1:3}^{(r)}, \mathbf{x}^{(r)} \rangle = 0.^{3} \tag{4}$$

Additionally, because it is the difference between $\mathbf{H}_{23}^{(l)}$ and $\mathbf{H}_{23}^{(r)}$ that matters, rather than their absolute values, in Eq. 4, we manually set $\mathbf{H}_{23}^{(l)} = 0$. The SVD solution to the homogeneous linear system is also scaled such that $\mathbf{H}_{22}^{(l)} > 0$ and $||\mathbf{H}_{2,1:2}^{(l)}|| = 1$.[4] In step 6, the number of inliers is defined as

$$\sum_{i=1}^{N} \mathbb{1}\big\{ \big| \langle \mathbf{H}_{2,1:3}^{(l)}, \mathbf{x}_i^{(l)} \rangle - \langle \mathbf{H}_{2,1:3}^{(r)}, \mathbf{x}_i^{(r)} \rangle \big| < \epsilon \big\}, \tag{5}$$

where $\mathbb{1}\{\cdot\}$ is the indicator function, and ϵ is a threshold on the residual epipolar errors after rectification. In step 8, we solve for $\mathbf{H}_{11}^{(l)}, \mathbf{H}_{12}^{(l)}$ by further imposing the norm constraint $||\mathbf{H}_{1,1:2}^{(l)}|| = ||\mathbf{H}_{2,1:2}^{(l)}||$ and the determinant constraint $\det(\mathbf{H}_{1:2,1:2}^{(l)}) > 0$. Similar constraints are also imposed on $\mathbf{H}^{(r)}$ to get $\mathbf{H}_{11}^{(r)}, \mathbf{H}_{12}^{(r)}$. Finally, most existing stereo matching algorithms assume the disparity values to be all negative; thus in step 9, we set $\mathbf{H}_{13}^{(r)}$ to the 1-percentile of the set

$$\big\{ \langle \mathbf{H}_{1,1:3}^{(l)}, \mathbf{x}_i^{(l)} \rangle - \langle \mathbf{H}_{1,1:3}^{(r)}, \mathbf{x}_i^{(r)} \rangle - \phi, i = 1, \ldots, N \big\}, \tag{6}$$

where $\phi = 50$ pixels is a protective margin. We then warp the left and right images with $\mathbf{H}^{(l)}$ and $\mathbf{H}^{(r)}$ respectively to obtain the pseudo-rectified stereo pair.

Step 2: Disparity estimation. In our work, we adopt the state-of-the-art learning-based stereo matching method of Yang *et al*. using their provided pretrained model [25]. Other stereo matching algorithms can also be substituted into our pipeline.

Step 3: Ambiguity removal. Because our pseudo-rectification method does not require accurate camera poses, the estimated disparity map is subject to an unknown global shift compared with that from *true* stereo rectification. The unknown shift is physically linked to the unknown y-axis orientations of the left and right cameras (see the proof of Prop. 1), and mathematically reflected by the freedom to arbitrarily set $\mathbf{H}_{13}^{(l)}$ and $\mathbf{H}_{23}^{(r)}$ in our pseudo-rectification algorithm. This ambiguity prevents us from recovering absolute depth from disparity. To resolve it, one needs to know the ambiguity-free disparity value for at least one pixel of the rectified left view. This is equivalent to inferring one or more pixels' depths, because of the depth-to-disparity formula

$$d = f \cdot \frac{C_{lr}}{z}, \tag{7}$$

where d is a pixel's disparity and z is its depth. Our ambiguity removal method utilizes the back view in our camera setup and is based on Prop. 2 for inference of pixel depths.

Proposition 2. *For two pixels in the left image with the same depth, if they are m_l pixels apart, while their corresponding pixels in the back image are m_b pixels apart, then the depth of these two pixels in the left camera's coordinate frame is*

$$z = \frac{C_{lb}}{\frac{m_l}{m_b} - 1}. \tag{8}$$

Proof. Denote the two same-depth pixels as $\mathbf{x}_1^{(l)}$ and $\mathbf{x}_2^{(l)}$ in the left image, and their corresponding 3D points as $\mathbf{X}_1^{(l)}$ and $\mathbf{X}_2^{(l)}$ in the camera coordinate frame. Then one can show,

$$m_l = ||\mathbf{x}_1^{(l)} - \mathbf{x}_2^{(l)}|| = \frac{f}{z^{(l)}} \cdot ||\mathbf{X}_1^{(l)} - \mathbf{X}_2^{(l)}||, \tag{9}$$

where $z^{(l)}$ is the common depth of $\mathbf{x}_1^{(l)}$ and $\mathbf{x}_2^{(l)}$. By similar logic and notation, for the back view, we have

$$m_b = \frac{f}{z^{(b)}} \cdot ||\mathbf{X}_1^{(b)} - \mathbf{X}_2^{(b)}||. \tag{10}$$

Because of our special camera setup, we have

$$\mathbf{X}_1^{(l)} - \mathbf{X}_2^{(l)} = \mathbf{X}_1^{(b)} - \mathbf{X}_2^{(b)}, z^{(b)} = z^{(l)} + C_{lb}. \tag{11}$$

Hence,

$$\frac{m_l}{m_b} = \frac{z^{(l)} + C_{lb}}{z^{(l)}}. \tag{12}$$

Rewriting the equation leads to $z = z^{(l)} = \frac{C_{lb}}{\frac{m_l}{m_b} - 1}$. □

Details of our ambiguity removal algorithm can be found in Algo. 3. We first detect sparse matches between the left and back images with SURF [4]. Then, in steps 2-7, we estimate the unknown disparity offset for a number of times in order to reduce uncertainty of single measurement, each time with two matches randomly sampled from all the matches. Suppose the sampled two matches are $(\mathbf{x}_1^{(l)}, \mathbf{x}_1^{(b)})$ and $(\mathbf{x}_2^{(l)}, \mathbf{x}_2^{(b)})$ at each time, in which $\mathbf{x}_1^{(l)} = (u_1^{(l)}, v_1^{(l)})$ is a pixel in the left view and similar rule applies to $\mathbf{x}_2^{(l)}, \mathbf{x}_1^{(b)}, \mathbf{x}_2^{(b)}$. Let the inter-pixel distances be denoted as $m_l = ||\mathbf{x}_1^{(l)} - \mathbf{x}_2^{(l)}||$ in the left view, $m_b = ||\mathbf{x}_1^{(b)} - \mathbf{x}_2^{(b)}||$ in the back view, and the values of the input disparity map at pixel locations $\mathbf{x}_1^{(l)}, \mathbf{x}_2^{(l)}$ are d_1, d_2, respectively. With the

[3] We use $\langle \cdot, \cdot \rangle$ to represent the inner product of two vectors. We also follow MATLAB notation of slicing matrices.

[4] $\|\cdot\|$ denotes the L_2-norm of a vector unless otherwise noted.

Algorithm 3: Ambiguity Removal

Input : rectified left image, back image, f, C_{lr}, C_{lb}, and estimated disparity map

Output : ambiguity-free disparity map

1 Detect matches between left and back images
2 **for** $t = 1 : T$ **do**
3 Randomly sample two matches
4 **if** *The two matches are far from each other in the left image, and have similar disparity values in the input disparity map* **then**
5 Estimate the disparity offset, and cache it
6 **end**
7 **end**
8 Shift the input disparity map by the median of all the cached disparity offset estimates

help of Prop. 2 and Eq. 7, the offset q resolving the ambiguity in the estimated disparity map can then be calculated as,

$$q = f \cdot \frac{C_{lr}}{C_{lb}} \cdot \left(\frac{m_l}{m_b} - 1 \right) - \frac{d_1 + d_2}{2}. \tag{13}$$

To suppress uncertainties in Eq. 13, disparity offset estimation is only performed when the conditions (1) $m_l > m_b$, (2) $m_l > \delta$, and (3) $|d_1 - d_2| < \eta$ are all satisfied, where δ and η are preset thresholds. Condition (1) serves as a sanity check on whether the two sampled matches are physically valid. Condition (2) ensures that there is a sufficient difference between m_l and m_b, while Condition (3) aims to guarantee that the two pixels have approximately equal depths. We take the median value of all the candidate disparity offset estimates, then shift the input disparity map to produce the ambiguity-free disparity map in step 8.

4. Experiments

In this section, we first illustrate what role the bas-relief ambiguity plays in our problem with a toy simulation example, and then show the effectiveness of our approach on both synthetic and real-world data.

4.1. Bas-relief ambiguity

To give readers a more concrete understanding, we demonstrate the influence of the bas-relief ambiguity on our problem mentioned in Sec. 2 through a simulation. In our simulation, we first generate a Gaussian surface $z = a + b \cdot \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$, in which $a = b = 300$ and $\sigma = 10$. We place the left camera at the origin and the right camera at $(2, 0, 0)$, and set both cameras' orientation to the identity. The image dimensions are set to 4608×3456, and the horizontal FOV to $6°$, with centered principal points.

We randomly sample 1,500 points from the Gaussian surface, and project them to the left and right images using the ground-truth poses. This yields 1,500 noise-free correspondences. To mimic real-world feature matching, we corrupt the projected pixel locations (u, v) in the right image with random noise (n_u, n_v) according to a 2D Gaussian distribution $\mathcal{N}(0, \mathrm{diag}(1/\sqrt{2}, 1/\sqrt{2}))$. From these noisy matches, together with ground-truth camera intrinsics, we estimate the essential matrix and perform two-view SfM to recover the right camera's relative pose with respect to the left one. Since SfM has a scale ambiguity, we scale the recovered translation vector such that the estimated distance between the two camera centers is the same as in the ground-truth. Figures 2 and 3 show the reconstructed 3D points and the corresponding recovered relative pose in one of our multiple runs. Despite the translation and x, z-axis rotations are almost perfectly recovered, the rotation about y-axis has a $0.207°$ error due to the bas-relief ambiguity, leading to severe distortions in the reconstruction.

4.2. Synthetic data

Setup. We generate synthetic images, along with ground-truth depth maps, for a set of scenes.[5] The horizontal camera FOV is set to $6°$, with image dimensions 4608×3456. The corresponding focal length is 43,963 pixels. For each scene, the left, right, and back images are rendered according to our camera setup in Fig. 1. To determine the cameras' positions, we first create a bounding box for the scene; then the left camera's pose is manually chosen such that the distance between its camera center and the bounding box centroid equals $S \big/ \tan(\frac{FOV}{2})$, with S being the bounding box's diagonal length. Both C_{lr} and C_{lb} are set to $1/150 \cdot S \big/ \tan(\frac{FOV}{2})$. Hence the baseline/depth ratio is as small as $\sim 1/150$; in other words, the intersection angle for the corresponding rays in the left and right views is just $\sim 0.382°$. The setup is equivalent to that of sensing depth for objects $\sim$300m away with a 2m baseline. The relative orientations of the right and back cameras with respect to the left one are generated by randomly sampling their x, y Euler angles from $[-1°, 1°]$, and z Euler angle from $[-5°, 5°]$.

Pseudo-rectification. We test our proposed pseudo-rectification method on this synthetic data. We set the number of sampled matches M at each RANSAC trial to 10, and the inlier epipolar error threshold ϵ to 2 pixels. In Fig. 4, we show an example for which both the true rectification with ground-truth poses and our purely image-based pseudo-rectification are performed. To facilitate visual inspection, we show two 160×120 crops of the pseudo-rectified views. Their locations are marked by the red boxes in the uncropped images. In addition to the rectification quality, the horizontal disparity is also visible from the crops. We then process the pseudo-rectified stereo pair with a stereo matching method [25]. In Fig. 5, we show the estimated disparity map, along with the

[5] The 3D models for rendering might not be in their real-world scale.

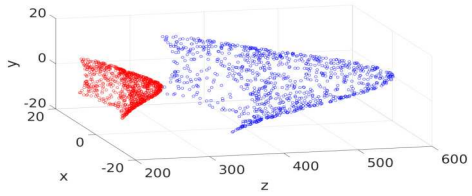


Figure 2: Ground-truth (blue) and the reconstructed (red) scene points. The unit for x, y, z axes is meter.

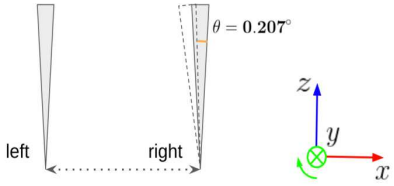


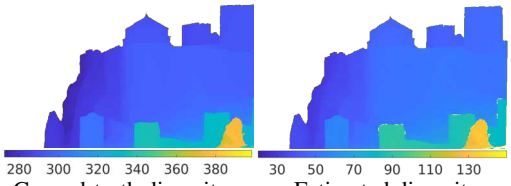
Figure 3: Top-down view of ground-truth relative pose (solid) and the recovered one (dashed). θ is exaggerated for illustration.

Raw left and right views

True-rectified left and right views

Pseudo-rectified left and right views

Figure 4: Pseudo-rectification on synthetic images.



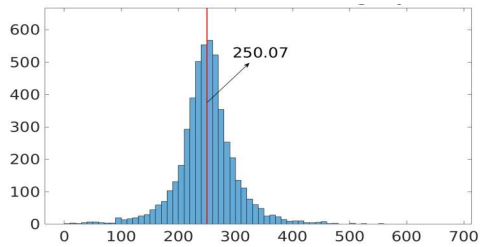


Figure 5: Histogram of 5,000 cached disparity offset estimates for the example in Fig. 4. The red line marks the final value we take.

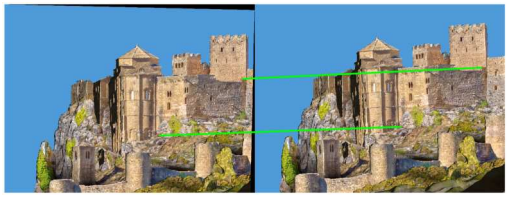


Figure 6: An example pair of matches used to resolve ambiguity for the example in Fig. 4. The two points are 1849.2px apart in the left image, and 1836.7px apart in the back one. Their original estimated disparities are 49.0px and 50.5px, respectively. According to Eq. 13, this yields a disparity offset estimate of 249.4px.

ground-truth, to illustrate the existence of an unknown global shift (∼250px).

Ambiguity removal. Next, we evaluate our ambiguity removal algorithm. We set the inter-pixel distance threshold δ to 300 pixels, and the disparity difference threshold η to 3 pixels in our ambiguity removal step. For the synthetic example in Fig. 4, Fig. 5 shows the histogram of all the cached disparity offset estimates produced by step 2-7 of Algo. 3, while Fig. 6 visualizes an example pair of matches that meet our criterion in step 4. The final value taken in step 8 is marked by the red line in the histogram plot; we can see that it is aligned with the mode of the histogram, and also in agreement with the two disparity maps. We finally convert the ambiguity-free disparity map to a depth map via Eq. 7.

The estimated depth map, compared with the ground-truth, are presented in the first row of Fig. 7. One can see that our proposed method outputs a depth map with relative errors below 3% at the majority (95.4%) of pixel locations. Another two synthetic examples can be seen in Fig. 9.

Loop and Zhang's rectification. As a comparison, we replace our pseudo-rectification with Loop and Zhang's rectification scheme [18], while keeping the other parts of our pipeline unchanged. The fundamental matrix required by their approach is estimated with a RANSAC-based normalized 8-point algorithm [12] from the same set of matches as that in our pseudo-rectification. Their results are shown in the second row of Fig. 7. The relative error map indi-

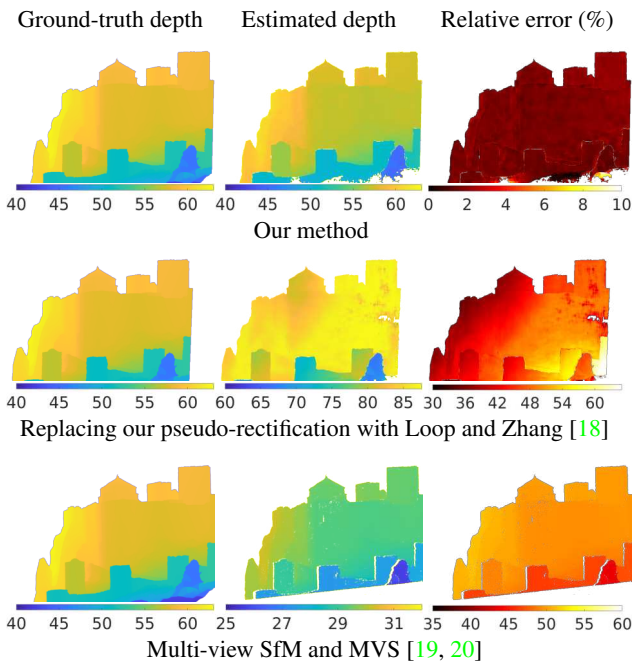


Figure 7: Comparison among different algorithms. For rectification-based methods, the ground-truth depth map has been warped to align with the rectified view. For SfM, we have used the full ground-truth intrinsic matrix.

cates that the depth map is strongly distorted when their method is used, for a similar reason to two-view SfM that we demonstrate in Sec. 4.1. In experiments, we also find that fundamental matrix estimation is quite unstable due to the tiny baseline/depth ratio, which causes Loop and Zhang's method to produce inconsistent results in different runs.

SfM+MVS. One might hypothesize that the back view can also help fix the bas-relief ambiguity in SfM. To test it, we feed the three views into COLMAP [19, 20] to run multi-view SfM and MVS. We use the ground-truth camera intrinsics, and initialize the three cameras' orientations to the identity; the left, right, and back camera centers are initialized to their ground-truth locations. The recovered pose and reconstruction are finally scaled such that the distance between the left and right camera centers is the same as in the ground-truth. Fig. 7 shows that even with the additional back view, SfM still suffers from the bas-relief ambiguity as in the case of stereo views. One of the key factors distinguishing SfM/SfSM from our approach is that, their bundle adjustment objective, i.e., average reprojection error, treats all the image-space observations equally, most of which are actually not informative for fixing the ambiguity, while our method only exploits a small carefully-chosen subset of all the observations, i.e., same-depth pixel pairs. Moreover, in practice, the principal points in camera intrinsics are unknown and can be tens of pixels away from the image center; this has no effect on our method, but can further hurt SfM/SfSM.

	Failure	<1%	<2%	<3%
Ours	0	**45.3%**	**80.1%**	**96.9%**
Loop and Zhang [18]	0	1.14%	2.73%	5.99%
SfM+MVS [19, 20]	15	6.71%	12.7%	19.1%

Table 1: Quantitative results on 40 synthetic scenes for methods in Fig. 7. "Failure" means the number of scenes for which a method fails to output a depth map. The metric is the portion of pixels with relative depth error below certain threshold, i.e., 1%, 2%, 3%, averaged over the successful scenes.

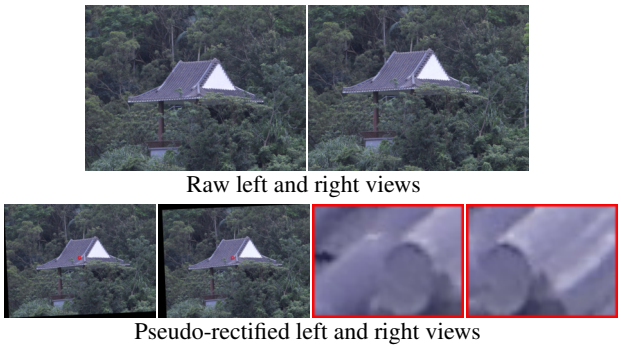


Figure 8: Pseudo-rectification on real-world images.

Tab. 1 quantitatively compares the aforementioned different algorithms on 40 synthetic scenes. Unlike other methods, our approach is not affected by the bas-relief ambiguity and outputs much more accurate depth estimates.

4.3. Real-world data

We capture real-world data with a Nikon P1000 super-zoom camera; the camera is mounted on a tripod and manually moved to three positions in line with our proposed camera setup in order to acquire the left, right, and back images. The captured images are of the same size, 4608 $\times$ 3456, as in the synthetic case. The 35mm equivalent focal length is 400mm, which corresponds to a camera FOV of 5.16° horizontally and 3.44° vertically. Because ground-truth dense depth maps are difficult to obtain for distant real-world scenes without special equipment, we use a laser rangefinder to acquire a point-wise depth measurement for a point of interest in the captured scene. We can then check if the estimated depth agrees with the measured one. The hyper-parameters, i.e., $M, \epsilon, \delta, \eta$, remain unchanged compared to the synthetic case.

We first show qualitative results of pseudo-rectification on real-world images in Fig. 8; like before, a 60 $\times$ 45 sub-area cropped out of the rectified views is presented to ease visual inspection. One can see that our pseudo-rectification generalizes very well to real-world images. In Fig. 10, we show the estimated depth maps; the measured depth from the laser rangefinder and the corresponding estimated value are marked inside the red boxes on the pseudo-rectified left

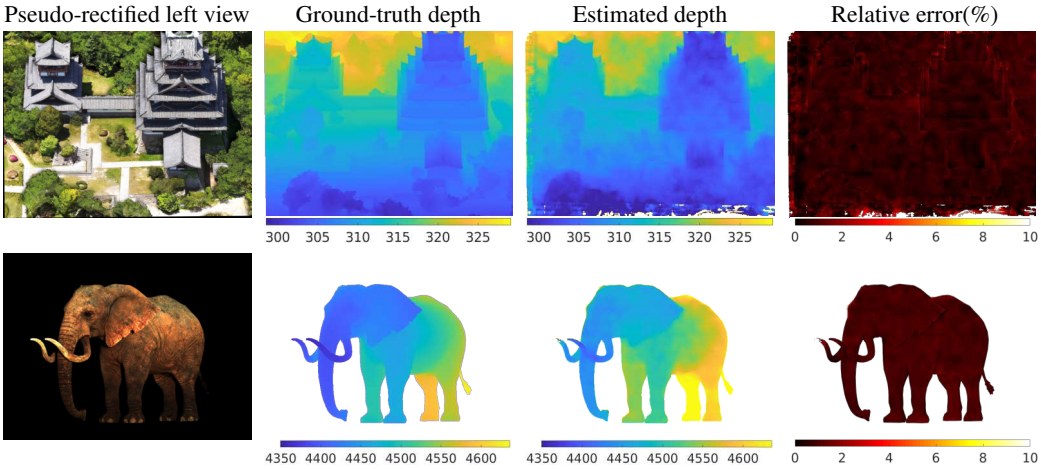


Figure 9: Results on synthetic data. Portion of pixel locations with <3% relative error (top to bottom): 98.5%, 99.7%.

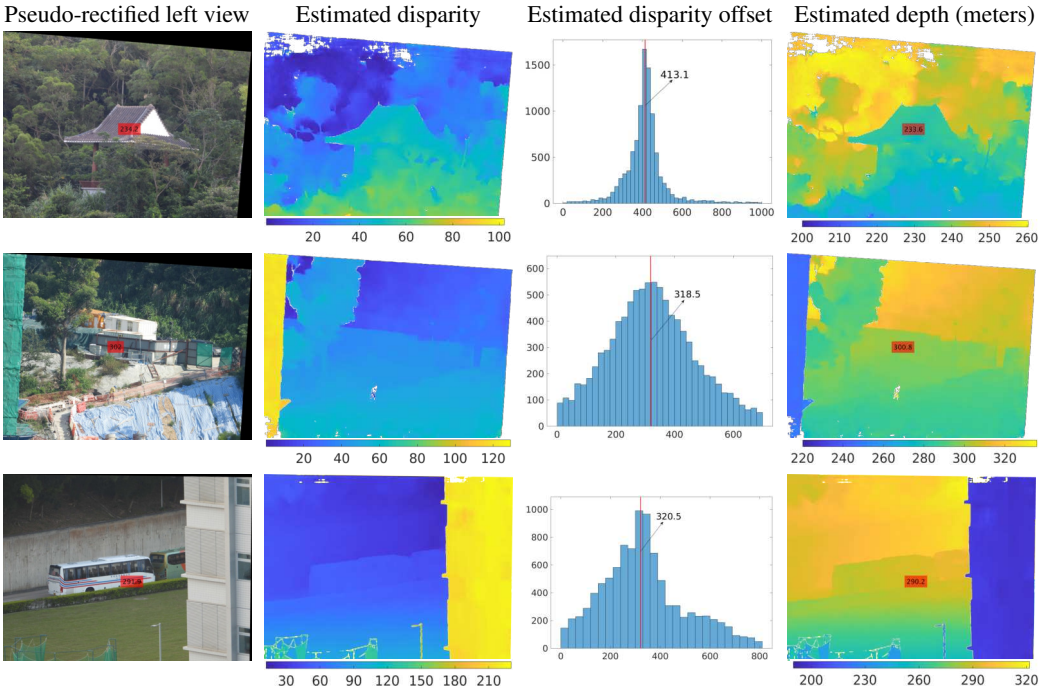


Figure 10: Results on real-world data. The depth measurements from a laser rangefinder are marked in the red box on the pseudo-rectified left images; the corresponding estimated values by our method are marked on the estimated depth maps. Laser-measured values (top to bottom): 234.2m, 302m, 291.9m; our estimated values (top to bottom): 233.6m, 300.8m, 290.2m.

view and the estimated depth map, respectively. From a practical perspective, the accuracy of our estimated depths is quite acceptable for applications in autonomous driving, considering the large distances to the scenes.

5. Discussion

In this work, we propose a novel vision-based solution to the long-range depth sensing problem in autonomous driving. We propose a three-camera system consisting of small-FOV cameras and a corresponding processing pipeline. Our end-to-end solution is very practical in that it does not assume full calibration of the camera system, and is robust to small system vibrations. Experiments show that our system enables *dense* depth acquisition of faraway objects (>200m) that are beyond the range of most commercial LiDARs for self-driving vehicles. This can be particularly helpful for heavily-weighted autonomous trucks moving at high speed.

As future work, we plan to conduct thorough experiments in real-world driving scenarios by building and testing a road-deployable hardware system.

References

[1] Velodyne Alpha Puck LiDAR, 2019. Available at https://velodynelidar.com/vls-128.html. Accessed: Oct. 18, 2019. 1

[2] Waymo Open Dataset: An autonomous driving dataset, 2019. Available at https://www.waymo.com/open. Accessed: Oct. 18, 2019. 1

[3] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M Seitz, and Richard Szeliski. Building Rome in a Day. *Communications of the ACM*, 54(10):105–112, 2011. 2

[4] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded Up Robust Features. In *Proc. European Conf. on Computer Vision (ECCV)*, pages 404–417. Springer, 2006. 3, 4

[5] Christian Beder and Richard Steffen. Determining an Initial Image Pair for Fixing the Scale of a 3D Reconstruction from an Image Sequence. In *Joint Pattern Recognition Symposium*, pages 657–666. Springer, 2006. 2

[6] Peter N Belhumeur, David J Kriegman, and Alan L Yuille. The Bas-Relief Ambiguity. *Int. J. of Computer Vision*, 35(1):33–44, 1999. 2

[7] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A multimodal dataset for autonomous driving. *arXiv preprint arXiv:1903.11027*, 2019. 1

[8] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 3

[9] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are We Ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2012. 1

[10] Hyowon Ha, Sunghoon Im, Jaesik Park, Hae-Gon Jeon, and In So Kweon. High-quality Depth from Uncalibrated Small Motion Clip. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, pages 5413–5421, 2016. 2

[11] Hyowon Ha, Tae-Hyun Oh, and In So Kweon. A closed-form solution to rotation estimation for structure from small motion. *IEEE Signal Processing Letters*, 25(3):393–397, 2017. 2

[12] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge university press, 2003. 6

[13] Richard I Hartley. Theory and Practice of Projective Rectification. *Int. J. of Computer Vision*, 35(2):115–127, 1999. 2

[14] Xinyu Huang, Jizhou Gao, and Ruigang Yang. Calibrating Pan-Tilt Cameras with Telephoto Lenses. In *Proc. Asian Conf. on Computer Vision (ACCV)*, pages 127–137. Springer, 2007. 2

[15] Sunghoon Im, Hyowon Ha, Gyeongmin Choe, Hae-Gon Jeon, Kyungdon Joo, and In So Kweon. Accurate 3D Reconstruction from Small Motion Clip for Rolling Shutter Cameras. *Trans. Pattern Analysis and Machine Intelligence*, 41:775–787, 2019. 2

[16] 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 *Proc. Int. Conf. on Computer Vision (ICCV)*, pages 837–845, 2015. 2

[17] H Christopher Longuet-Higgins. A Computer Algorithm for Reconstructing a Scene from Two Projections. *Nature*, 293(5828):133, 1981. 2

[18] Charles Loop and Zhengyou Zhang. Computing Rectifying Homographies for Stereo Vision. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 125–131. IEEE, 1999. 2, 3, 6, 7

[19] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-Motion Revisited. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, pages 4104–4113, 2016. 2, 7

[20] Johannes L. Schonberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise View Selection for Unstructured Multi-View Stereo. In *Proc. European Conf. on Computer Vision (ECCV)*, 2016. 7

[21] Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo Tourism: Exploring Photo Collections in 3D. In *ACM Trans. Graphics*, volume 25, pages 835–846. ACM, 2006. 2

[22] Noah Snavely, Steven M Seitz, and Richard Szeliski. Modeling the world from internet photo collections. *International journal of computer vision*, 80(2):189–210, 2008. 2

[23] Richard Szeliski and Sing Bing Kang. Shape Ambiguities in Structure From Motion. In *Proc. European Conf. on Computer Vision (ECCV)*, pages 709–721. Springer, 1996. 2

[24] Bill Triggs, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon. Bundle AdjustmentA Modern Synthesis. In *International Workshop on Vision Algorithms*, pages 298–372. Springer, 1999. 2

[25] Gengshan Yang, Joshua Manela, Michael Happold, and Deva Ramanan. Hierarchical Deep Stereo Matching on High-Resolution Images. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, June 2019. 4, 5

[26] Fisher Yu and David Gallup. 3D Reconstruction from Accidental Motion. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2014. 2

[27] Zhengyou Zhang. A Flexible New Technique for Camera Calibration. *Trans. Pattern Analysis and Machine Intelligence*, 22, 2000. 2