Depth Estimation Using a Sliding Camera

Kailin Ge, Han Hu, Jianjiang Feng, *Member, IEEE*, and Jie Zhou, *Senior Member, IEEE*

***Abstract*— Image-based 3D reconstruction technology is widely used in different fields. The conventional algorithms are mainly based on stereo matching between two or more fixed cameras, and high accuracy can only be achieved using a large camera array, which is very expensive and inconvenient in many applications. Another popular choice is utilizing structure-from-motion methods for arbitrarily placed camera(s). However, due to too many degrees of freedom, its computational cost is heavy and its accuracy is rather limited. In this paper, we propose a novel depth estimation algorithm using a sliding camera system. By analyzing the geometric properties of the camera system, we design a camera pose initialization algorithm that can work satisfyingly with only a small number of feature points and is robust to noise. For pixels corresponding to different depths, an adaptive iterative algorithm is proposed to choose optimal frames for stereo matching, which can take advantage of continuously pose-changing imaging and save the time consumption amazingly too. The proposed algorithm can also be easily extended to handle less constrained situations (such as using a camera mounted on a moving robot or vehicle). Experimental results on both synthetic and real-world data have illustrated the effectiveness of the proposed algorithm.**

***Index Terms*— Sliding camera, constrained structure from motion, multi-view stereo, variational depth estimation, pixel-wise frame selection.**

I. Introduction

IMAGE-BASED depth estimation for stereo reconstruction has received worldwide attention over the past decades and is now widely used in augmented reality, 3D modeling and printing, intelligent surveillance, etc.

Conventional depth estimation algorithms are mainly based on image matching between two or more fixed cameras [1], [2]. In this case, the cameras can be calibrated in advance so the step of camera pose estimation can be skipped at runtime. Feature points are extracted from the captured images first and the matches between feature points are utilized to obtain an initial depth estimation. The initial estimation is then refined to a better result using pixel-wise or patch-wise correlation. Some studies have proved that high accuracy can only be produced by using a large camera array and a small array that lacks in redundancy will degrade the precision of stereo reconstruction [3], [4]. But a large camera array is very expensive and inconvenient in many applications.

In recent years, many studies have focused on pose estimation and depth computing by using images captured from arbitrarily placed camera(s) [5], [6]. This kind of method is based on structure-from-motion (SfM) technology [7]. Since the camera(s) are unconstrained, it seems much more convenient in installation. However, due to too many degrees of freedom (DOF) in SfM, its computational cost is heavy and its accuracy is rather limited. In some circumstances (e.g., no plenty number of valid matching features), it may totally fail. To overcome the limitation of SfM, some other works [8]–[11] try to use auxiliary information (e.g., measurements from IMU and/or GPS) to help camera pose estimation and depth computation. But such information is not very accurate and thus not very helpful.

In this paper, we study depth estimation using a sliding camera system, in which a camera is mounted to a controllable straight track (as shown at the top of Fig. 1). This system is cheaper and more portable than a fixed camera array, and can provide dense and continuous views when the camera slides from one end to another (considering the cost of the system and the generalizability of the algorithm, we do not rely on any camera location feedback or any assumption of moving speed). On the other side, since the moving of the camera is constrained by the track, there are much fewer DOF for camera pose estimation. The pipeline of the whole algorithm is shown in Fig. 1. A feature tracking procedure is first applied to the captured image sequence, and the output feature trajectories are further processed by the proposed camera pose initialization algorithm. The camera poses are refined by a constrained optimization algorithm. After compensating camera vibration (an optional procedure), an initial depth map is calculated based on a narrow baseline pair. The depth map is then refined by iteratively adding data from the rest of images, which is controlled by the proposed pixel-wise frame selection framework. Experimental results on both synthetic and real-world data have illustrated the effectiveness of the proposed algorithm.

The main contributions of this paper are listed as below. 1) We propose a model of the sliding-camera system and analyzed its geometric properties. There are three important properties of this system, namely co-linear, concurrent and cross-ratio properties. 2) Based on geometric analysis of the sliding camera system, we propose a parametric model for constrained camera pose estimation as well as a robust

Manuscript received April 15, 2015; revised October 10, 2015 and November 25, 2015; accepted December 3, 2015. Date of publication December 11, 2015; date of current version January 5, 2016. This work was supported in part by the National Natural Science Foundation of China under Grant 61225008, Grant 61572271, Grant 61527808, Grant 61373074, and Grant 61373090, in part by the National Basic Research Program of China under Grant 2014CB349304, in part by the Ministry of Education, China, under Grant 20120002110033, and in part by the Tsinghua University Initiative Scientific Research Program. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Karsten Mueller.

The authors are with the Tsinghua National Laboratory for Information Science and Technology, Department of Automation, Tsinghua University, Beijing 100084, China (e-mail: gkl05@mails.thu.edu.cn; huh04@mails.thu.edu.cn; jfeng@tsinghua.edu.cn; jzhou@tsinghua.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2015.2507984

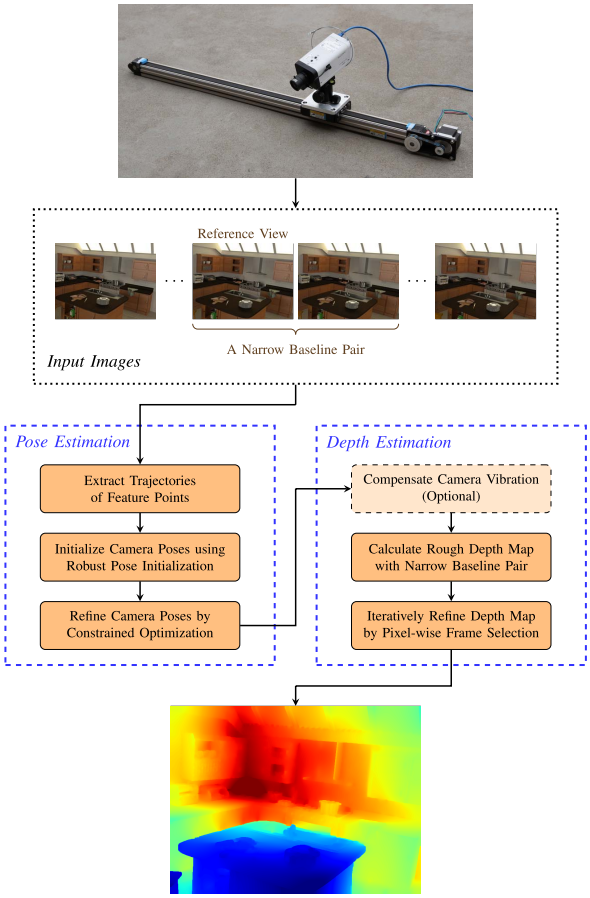


Fig. 1. Pipeline of the whole algorithm. A feature tracking procedure is first applied to the captured image sequence, and the output feature trajectories are further processed by the proposed camera pose initialization algorithm. The camera poses are refined by a constrained optimization algorithm. After compensating camera vibration (an optional procedure), an initial depth map is calculated based on a narrow baseline pair. The depth map is then refined by iteratively adding data from the rest of images, which is controlled by the proposed pixel-wise frame selection framework.

algorithm for camera pose initialization, which can work satisfyingly with only a small number of feature points and is robust to noise. 3) For pixels corresponding to different depth, an adaptive iterative algorithm is proposed to choose optimal frames for stereo matching, which can take advantage of continuously pose-changing imaging and save the time consumption amazingly too. Combining with a variational optimization model, our algorithm can produce results with high precision and low memory consumption. The proposed algorithm can be easily extended to handle less constrained situations (such as using a camera mounted on a moving robot or vehicle). The basic part of this work has been described in our previous conference paper [12], which does not include the robust camera pose initialization, camera vibration compensation and pixel-wise frame selection in the current paper.

The remainder of the paper is organized as follows: In Section II, we model the proposed sliding camera system in mathematics and deduce three important geometric properties of the system. Section III describes the camera pose estimation pipeline, including the proposed robust pose initialization algorithm and the refining optimization. In Section IV, we describe the depth estimation pipeline and the proposed pixel-wise selection algorithm. Section V shows experimental results of the proposed framework on both synthetic and real-world data, and Section VI is the summary and prospect of this paper.

II. System Modeling and Geometry Properties

As described in Section I, in our system we mount the camera to a straight track, which constrains the camera to move straightly and continuously. When moving along the track, the camera captures a series of images $\{\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_n\}$, and our task is to calculate view-dependent depth maps, $\{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_n\}$, from these images. For a static scene, our system is equivalent to a virtual linear array, and all images can be treated as being captured at the same time. In this section, we analyze the geometric properties of our system, and propose a parametric model for constrained camera pose estimation.

A. Geometry Analysis

The fundamental theory of camera geometry has been well studied [13], and the case of camera in pure translational motion (CPT) is also included, which can be treated as a simplified two-view case of our system. Moons *et al.* [14] studied how to do affine reconstruction of a 3D scene when a camera purely translates between two places. Chen *et al.* [15] proposed a method of estimating epipole under CPT. Hu and Tan [16] proposed a method of sparse depth calculation under CPT. We will combine perspective geometry and epipolar geometry of CPT to deduce the model of the sliding-camera system and its geometric properties.

1) Coordinate System: We start with the ideal case: the camera moves straightly along the track without any disturbance (such as shaking). In this case, the camera's orientation (i.e., the $\mathbf{R}$ in projective geometry) keeps constant and the optical centers, $C_1, C_2, \ldots, C_n$, are co-linear. We define the coordinate system as follows:

- *Origin:* The optical center of the first camera of the virtual array (C_1 in Fig. 2). This camera is supposed to be at one end of the virtual array.
- *X-axis:* X-axis is determined by the vector from the optical center of the first camera to that of another in the virtual array ($\overrightarrow{C_1C_2}$ or $\overrightarrow{C_1C_3}$ in Fig. 2). Since $\{C_1, C_2, \ldots, C_n\}$ are co-linear, they are all on X-axis and in this paper their coordinates are denoted by $\{(c_1, 0, 0), (c_2, 0, 0), \ldots, (c_n, 0, 0)\}$ (where $c_1 = 0$). Following this rule, in fact, X-axis is parallel to the track.
- *Y-axis:* Y-axis is determined by the cross product of $\overrightarrow{C_1\underset{\sim}{o}_1}$ (the vector from optical center to the principal point) and X-axis, according to right-hand rule. If $\overrightarrow{C_1\underset{\sim}{o}_1}$ is parallel to X-axis, Y-axis could be any direction orthogonal to X-axis.[1]
- *Z-axis:* Z-axis is determined by the cross product of X-axis and Y-axis, according to right-hand rule. Following these rules, all the principal points, $\{\underset{\sim}{o}_1, \underset{\sim}{o}_2, \ldots, \underset{\sim}{o}_n\}$, are on X-Z-plane.

[1]Since a point on image planes can either be a 3D point or a 2D point (with image coordinates), we denote its 2D form with lower case symbol (e.g., o_1) and 3D form with a under tilde lower case symbol (e.g., $\underset{\sim}{o}_1$).

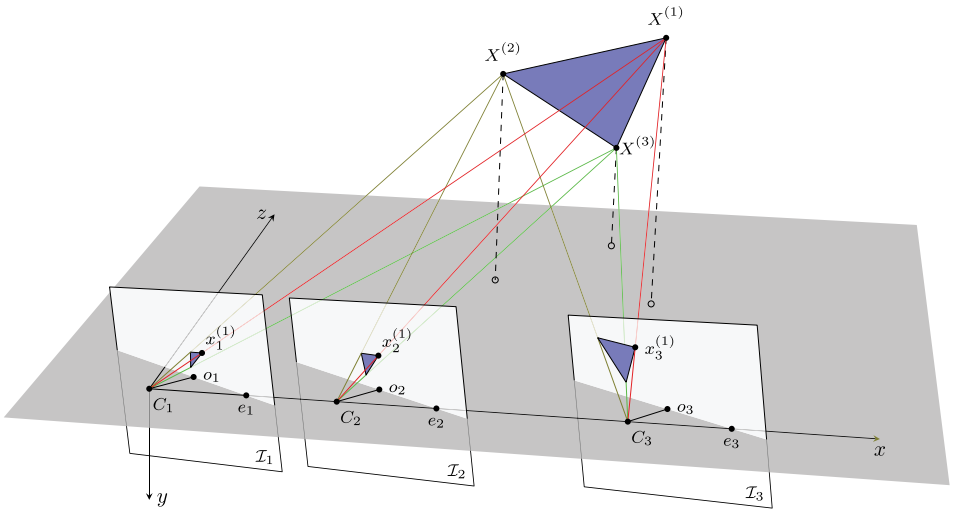


Fig. 2. 3D geometry of the sliding camera system. It can also be treated as many cameras on a track which are parallel and co-linear.

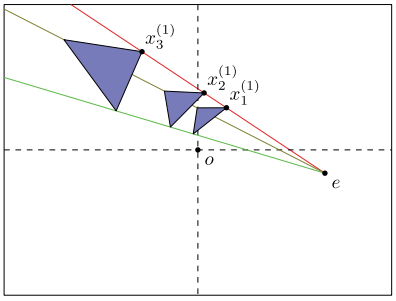


Fig. 3. Among all frames, projections of the same 3D point are co-linear with e, and they also satisfy cross-ratio property.

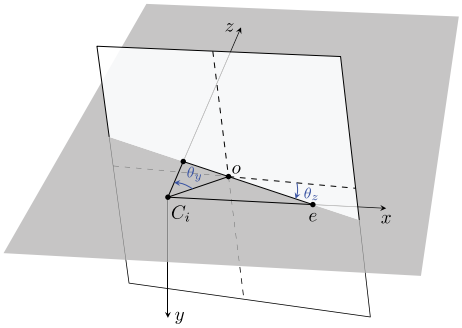


Fig. 4. Geometry relationship between e and $\mathbf{R}$.

From Fig. 2 it can be found that the X-axis intersects all image planes at the same place (with respect to image coordinates of $e_1, e_2, \ldots, e_n$), because all cameras in the virtual array are identical and parallel. Stacking all the images $\{\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_n\}$ together as illustrated in Fig. 3, the principal points $\{o_1, o_2, \ldots, o_n\}$ will coincide at o and $\{e_1, e_2, \ldots, e_n\}$ will coincide at e. Now we prove three important geometric properties of the system, i.e., co-linear, concurrent and cross-ratio properties.

2) Co-Linear and Concurrent Properties:

Proposition 1: Projections, $\{x_1, x_2, \ldots, x_n\}$, of the same 3D point X are co-linear with $e = \mathbf{KR}[1, 0, 0]^{\mathrm{T}}$ with respect to projective geometry (i.e., in perspective space $\mathbb{P}^2$).

Proof: $\forall i \in \{1, 2, \ldots, n\}$, since $C_i = (c_i, 0, 0)$ and $c_1 = 0$, we have

$$\begin{aligned} x_i &= \mathbf{KR}\big(X - C_i\big) \\ &= \mathbf{KR}X - \mathbf{KR}[c_i, 0, 0]^{\mathrm{T}} \\ &= \mathbf{KR}\big(X - C_1\big) - c_i\mathbf{KR}[1, 0, 0]^{\mathrm{T}} \\ &= x_1 - c_i e. \end{aligned}$$

Therefore, homogeneous coordinate x_i can be linearly represented by x_1 and e. That is, x_i, x_1 and e are co-linear according to projective geometry. Then $\{x_1, x_2, \ldots, x_n\}$ are co-linear with e. ■

Proposition 1 shows that the projection trajectories of 3D scene points should be straight lines, and these lines intersect at the same vanishing point of the stacked image plane (Here the image plane is an extended 2D plane $\bar{\mathbb{R}}^2$, where points at infinite are included). This clue will be used for robust camera pose initialization in Section III. In addition, $\mathbf{R}$ can be calculated if the homogeneous coordinate of e is known.

According to the coordinate system defined above, $\mathbf{R}$ can be decomposed using Y-Z Euler angles as

$$\mathbf{R} = \begin{bmatrix} \cos\theta_z & -\sin\theta_z & 0 \\ \sin\theta_z & \cos\theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos\theta_y & 0 & -\sin\theta_y \\ 0 & 1 & 0 \\ \sin\theta_y & 0 & \cos\theta_y \end{bmatrix}, \quad (1)$$

where $-\dfrac{\pi}{2} < \theta_y \leq \dfrac{\pi}{2}$. This relationship is illustrated in Fig. 4. Then we have

$$e = \mathbf{KR}\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \mathbf{K}\begin{bmatrix} \cos\theta_y \cos\theta_z \\ \cos\theta_y \sin\theta_z \\ \sin\theta_y \end{bmatrix}.$$

Now it is easy to find that

$$\sin\theta_y = \frac{e_z}{n_e}, \quad \cos\theta_y = \sqrt{1 - \sin^2\theta_y},$$
$$\cos\theta_z = \frac{e_x}{n_e \cos\theta_y}, \quad \sin\theta_z = \frac{e_y}{n_e \cos\theta_y}, \quad (2)$$

where

$$[e_x, e_y, e_z]^{\mathrm{T}} = \mathbf{K}^{-1}e, \quad n_e = \sqrt{e_x^2 + e_y^2 + e_z^2}.$$

3) Cross-Ratio Property: Another useful geometric property is cross-ratio, which can be used to calculate coordinate correspondences even without normalization of image coordinates. Take Fig. 2 and Fig. 3 for example. Based on the analysis in [14], we have

$$\frac{\left(\overrightarrow{ex_2^{(1)}} \cdot \overrightarrow{d^{(1)}}\right)\left(\overrightarrow{x_1^{(1)}x_3^{(1)}} \cdot \overrightarrow{d^{(1)}}\right)}{\left(\overrightarrow{ex_1^{(1)}} \cdot \overrightarrow{d^{(1)}}\right)\left(\overrightarrow{x_2^{(1)}x_3^{(1)}} \cdot \overrightarrow{d^{(1)}}\right)} = \frac{\overrightarrow{C_1C_3} \cdot \vec{D}}{\overrightarrow{C_2C_3} \cdot \vec{D}},$$

where

$$\overrightarrow{d^{(1)}} = \frac{\overrightarrow{ex_3^{(1)}}}{\left|ex_3^{(1)}\right|}, \quad \vec{D} = \frac{\overrightarrow{C_1C_3}}{\left|C_1C_3\right|}.$$

More generally, if a 3D point X's projections on image $\mathcal{I}_i, \mathcal{I}_j, \mathcal{I}_k$ are x_i, x_j, x_k, respectively, we have

$$\frac{\left(\overrightarrow{ex_j} \cdot \overrightarrow{d(x_k)}\right)\left(\overrightarrow{x_ix_k} \cdot \overrightarrow{d(x_k)}\right)}{\left(\overrightarrow{ex_i} \cdot \overrightarrow{d(x_k)}\right)\left(\overrightarrow{x_jx_k} \cdot \overrightarrow{d(x_k)}\right)} = \frac{\overrightarrow{C_iC_k} \cdot \vec{D}}{\overrightarrow{C_jC_k} \cdot \vec{D}}, \tag{3}$$

where

$$\overrightarrow{d(x_k)} = \frac{\overrightarrow{ex_k}}{\left|ex_k\right|}.$$

Since $\overrightarrow{C_iC_k} \cdot \vec{D} = c_k - c_i$, $\overrightarrow{C_jC_k} \cdot \vec{D} = c_k - c_j$, $\overrightarrow{ex_j} = \overrightarrow{ex_k} + \overrightarrow{x_kx_j}$ and $\overrightarrow{ex_k} \cdot \overrightarrow{d(x_k)} = \left|ex_k\right|$, the above equation can be rewritten as

$$\overrightarrow{x_kx_j} \cdot \overrightarrow{d(x_k)} = \frac{(c_j - c_k)\left|ex_k\right|\left(\overrightarrow{x_kx_i} \cdot \overrightarrow{d(x_k)}\right)}{(c_i - c_k)\left|ex_k\right| + (c_i - c_j)\left(\overrightarrow{x_kx_i} \cdot \overrightarrow{d(x_k)}\right)}. \tag{4}$$

Notice that when e lies at infinity and x_i, x_j, x_k are finite, $\overrightarrow{d(x_k)}$ becomes a constant vector (i.e., all trajectories are parallel), and

$$\lim_{|e|\to\infty} \frac{\overrightarrow{ex_j} \cdot \overrightarrow{d(x_k)}}{\overrightarrow{ex_i} \cdot \overrightarrow{d(x_k)}} = 1.$$

Then Equation (3) and Equation (4) degenerate to

$$\frac{\overrightarrow{x_ix_k} \cdot \overrightarrow{d(x_k)}}{\overrightarrow{x_jx_k} \cdot \overrightarrow{d(x_k)}} = \frac{\overrightarrow{C_iC_k} \cdot \vec{D}}{\overrightarrow{C_jC_k} \cdot \vec{D}}$$

and

$$\overrightarrow{x_kx_j} \cdot \overrightarrow{d(x_k)} = \frac{c_j - c_k}{c_i - c_k}\left(\overrightarrow{x_kx_i} \cdot \overrightarrow{d(x_k)}\right),$$

which are standard disparity transformations.

This cross-ratio property will be used in our camera pose initialization as well as depth estimation algorithm.

B. Parametric Camera Pose Model

Bundle adjustment [17] is a state-of-the-art optimization model of SfM, because it perfectly encodes all information of matched features. However, the matched features are not always sufficient, so a pure bundle adjustment may fail in complicated cases. Therefore, some studies use auxiliary information to enhance the robustness and practicality of bundle adjustment. For example, Lhuillier [10] and Crandall *et al.* [11] use GPS to get a coarse camera location for SfM initialization, [8], [9] use IMU to get relative poses between sequentially captured images, and Nilsson *et al.* [18] combined the motion model of the vehicle. These models do not match our case because none of them reflects a global constraint that the track provides. Therefore, based on the geometric analysis in Section II, we propose the parametric model as

$$\min_{\substack{\{X^{(l)}\},\{c_k\},\\ \mathbf{R},\{\Delta\mathbf{R}_k\}}} \frac{\sum_{k,l} w_k^{(l)} \Psi\left(\left\|x_k^{(l)} - P_k(X^{(l)})\right\|_2^2\right)}{\sum_{k,l} w_k^{(l)}} + \frac{\gamma}{n}\sum_k \Phi\left(\Delta\mathbf{R}_k\right), \tag{5}$$

where $\{X^{(l)}\}$ are the 3D coordinates of matched features, $\{c_k\}$ and $\mathbf{R}$ are described in Section II, and $\{\Delta\mathbf{R}_k\}$ are compensations for possible vibration. The second term in this model handles possible camera vibrations, which has not been addressed in our previous conference paper [12].

The first term of Equation (5) is a variant of conventional bundle adjustment model, where P_k projects a 3D point $X^{(l)}$ onto camera k using the projection matrix as

$$\mathbf{P}_k = \mathbf{K} \cdot \Delta\mathbf{R}_k \cdot \mathbf{R}_k \cdot \left[\begin{array}{c|c} \mathbf{I} & \begin{matrix}-c_k \\ 0 \\ 0\end{matrix}\end{array}\right],$$

and $\Psi(x^2) = \sqrt{x^2 + \varepsilon^2}$ is a robust energy function to tolerate outliers. The second term is for regularization which inhibits large vibration compensation. Here γ is a balance factor and $\Phi\left(\Delta\mathbf{R}_k\right) = \left\|\mathbf{I} - \Delta\mathbf{R}_k\right\|_F^2$ where $\left\|\cdot\right\|_F$ is Frobenius norm.

III. Camera Pose Estimation

Geometric analysis in the previous section provides three important properties of our system, i.e., co-linear, concurrent and cross-ratio properties. In this section, we will propose a pose initialization algorithm based on these properties.

A. Initialization

Feature points are first detected from images as in [19] and then Lucas-Kanade tracker [20] is used to establish the correspondences. Denote the collection of feature points, their visibilities and the corresponding 3D points by $\{x_k^{(l)}\}$, $\{w_k^{(l)}\}$ and $\{X^{(l)}\}$, respectively. The visibilities are defined as:

$$w_k^{(l)} = \begin{cases} 1 & \text{if a matching } x_k^{(l)} \text{ of } X^{(l)} \text{ is found in } \mathcal{I}_k, \\ 0 & \text{otherwise.} \end{cases}$$

The initialization is based on these data, which may contain some incorrect matches introduced by the previous phase.

Algorithm 1 Camera Pose Initialization

Input: $\{x_k^{(l)}\}$ and $\{w_k^{(l)}\}$
Output: $\mathbf{R}$ and $\{c_k\}$

1. Do PCA on all trajectories, and pick out top-m ones with highest straightness scores;
2. Estimate motion vanishing point e using the method of [21] based on the end points of the top-m trajectories;
3. Estimate the sign of e and then calculate initial $\mathbf{R}$ according to Equation (1) and (2);
4. Resort all the trajectories according to both straightness score and co-linearity with e score;
5. Calculate initial $\{c_k\}$ based on resorted top-m trajectories using cross-ratio property.

We first use co-linear property to pick out the top-m (m is set to 10 experimentally in our study) trajectories with highest co-linear scores, which are further used for estimating initial values of e and $\{c_k\}$.[2] The flowchart of our initialization algorithm is shown in Algorithm 1, and the details are described as follows.

We use principal component analysis (PCA) to decompose the points of every trajectory, and the ones with smaller $\lambda_{\min}/\lambda_{\max}$ values are more likely to be co-linear. After penalizing short trajectories, the end points of top-m trajectories are picked out and then used to estimate initial value of the motion vanishing point e using the method in [21]. Using this robust strategy, the estimated initial value of e is proved to be reliable throughout all our experiments.

Notice that in perspective geometry both $\pm e$ (homogeneous coordinates) correspond to the same 2D image coordinate, and the sign does not matter during the phases above. However, when we need to infer $\mathbf{R}$ using Equation (1) and (2), the sign should be known. More intuitively, we need to know whether e is the projection of the positive part of X-axis or the negative part.

We use the following method to determine the sign of e. By projecting e to 2D image coordinate $(e_x, e_y) \in \bar{\mathbb{R}}^2$, if most trajectories are moving towards (e_x, e_y), $\mathbf{R}$ is calculated using $-e$; if backwards, using $+e$.

After e is known, we use a stricter criterion to further check trajectories, i.e., inliners should not only have a high straightness score, but also be co-linear with e. By resorting all the trajectories according to this criterion, we re-choose top-m trajectories for estimating the initial values of $\{c_k\}$. According to Equation (3), we denote

$$r_k = \frac{c_{k+1}-c_k}{c_k-c_{k-1}} = -\frac{\overrightarrow{C_{k+1}C_k}\cdot\vec{D}}{\overrightarrow{C_{k-1}C_k}\cdot\vec{D}} = -\frac{\left(\overrightarrow{ex_{k-1}}\cdot\overrightarrow{d(x_k)}\right)\left(\overrightarrow{x_{k+1}x_k}\cdot\overrightarrow{d(x_k)}\right)}{\left(\overrightarrow{ex_{k+1}}\cdot\overrightarrow{d(x_k)}\right)\left(\overrightarrow{x_{k-1}x_k}\cdot\overrightarrow{d(x_k)}\right)}.$$

[2]We assume that small vibration of camera causes only small shifts of feature points, so the three geometric properties approximately hold true.

Assume $c_1 = 0$ and $c_n = 1$,[3] $c_1, c_2, \ldots, c_{n-1}$ can be calculated as

$$c_k = \frac{1+\sum_{i=2}^{k-1}\prod_{j=2}^{i} r_j}{1+\sum_{i=2}^{n-1}\prod_{j=2}^{i} r_j}.$$

For robustness, r_k is set as the median value of the results of top-m trajectories.

B. Constructed Bundle Adjustment

By clamping $\mathbf{R}$ and $\{c_k\}$ to the initial values calculated above, we use Levenberg-Marquardt method to optimize Equation (5) and get initial values for $\{X^{(l)}\}$ and $\{\Delta\mathbf{R}_k\}$. Then we can find outliers by thresholding reprojection errors: if the reprojection error of $x_k^{(l)}$ is larger than a threshold, set $w_k^{(l)}$ to 0. After this, we fully optimize Equation (5) and get a final camera pose output.

IV. Depth Estimation

In this section, we describe how to embed the geometric properties into a variational model for depth estimation, as well as a pixel-wise frame selection strategy.

Without loss of generality, we choose the k-th view as reference view, and our task is to compute $\mathcal{D}_k$. The depth estimation pipeline consists of two main phases: 1) Choose a view which is adjacent to the k-th view, then estimate a coarse depth map using this narrow-baseline pair. 2) Iteratively refine the depth map using the proposed pixel-wise frame selection strategy. Note that when the vibration of camera is too large to be ignored, a vibration compensation phase should be performed by warping images according to the homography transform, $\mathbf{H}_k = \mathbf{K}\Delta\mathbf{R}_k^{-1}\mathbf{K}^{-1}$ [13].

A. Depth Initialization

There are two mainstream methods for modeling this problem, i.e., Markov random field (MRF) model and variational model:

1) *MRF model*: Many MRF algorithms (e.g., graph cut (GC) [22] and loopy belief propagation (LBP) [23], [24]) have good convergence without relying on an initial solution. But the precision is limited by quantization, and the memory consumption grows linearly as quantization resolution increases.
2) *Variational model*: The result is calculated by continuous optimization (which implies high precision) with low memory consumption. But most algorithms [25]–[28] guarantee only local convergence when the unary term is non-convex, i.e., it is sensitive to initial solution.

Since variational model is sensitive to initial solution and a good initial matching is very difficult to find in wide baseline systems, MRF model is much more widely used in stereo systems. In our sliding camera system, however, because the length of baseline varies continuously, it is easy to get a narrow-baseline pair, in which the captured images

[3]Since there intrinsically exists scale ambiguity in SfM problem, explicit normalization is required.

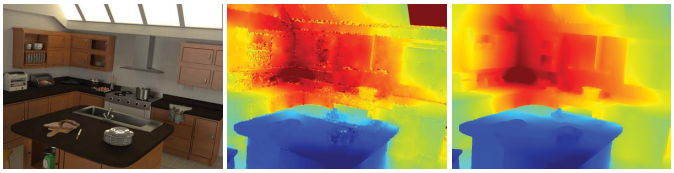


Fig. 5. Depth estimation results. Left: Image of reference view. Middle: Initial estimation from a narrow baseline pair, which is almost correct but has lower accuracy. Right: Result after multi-view depth refinement, which has a much higher accuracy.

differ slightly. Therefore, a good initial solution can be found by narrow triangulation, which compensates for the disadvantage of variational model. On the other hand, variational models require much less memory than MRF models. That is the reason why we choose variational model for depth estimation.

Thank to the continuity of the slider, we can find a view, namely the i-th view, which is near to the reference the k-th view, and is used to initialize the algorithm. It is a two-view stereo problem and we use the method of [27] (a constrained version of [25]) to build the optimization model:

$$\min_{w_{ki}} \; E_D(\omega_{ki}) + \gamma\, E_S(\omega_{ki}). \tag{6}$$

In this optimization, ω_{ki} is a function mapping a coordinate $x_k \in \Omega$ to the tangential disparity along the epipolar line (as illustrated in Fig. 9) between the k-th view and the i-th view. E_D is the data term for measuring pixel variance, which is defined as

$$E_D(\omega_{ki}) = \int_\Omega c(x_k)\Psi\Big(\big\|\mathcal{I}_k^*(x_k) - \mathcal{I}_i^*\big(x_k + w_{ki}(x_k)\cdot\overrightarrow{d(x_k)}\big)\big\|_2^2\Big)\mathrm{d}x_k, \tag{7}$$

where $\mathcal{I}_k^*$ and $\mathcal{I}_i^*$ are 6-channel images containing $(R, G, B, \beta\cdot\nabla R, \beta\cdot\nabla G, \beta\cdot\nabla B)$ for measuring both color values and color gradients, and $\overrightarrow{d(x_k)}$ (as illustrated in Fig. 9) is previously defined in Equation (3). $\Psi(x^2) = \sqrt{x^2+\varepsilon^2}$ is a robust energy function, and a confidence factor is defined as

$$c(x_k) = 1 - \frac{\sigma_c^2}{\big\|\nabla\mathcal{I}_k(x_k)\big\|_2^2 + \sigma_c^2}.$$

E_S is the smoothing term, defined as

$$E_S(\omega_{ki}) = \int_\Omega \xi_k(x_k)\Psi\Big(\big\|\nabla w_i(x_k)\big\|_2^2\Big)\mathrm{d}x_k,$$

where ξ_k is the edge prior to preserve discontinuities as

$$\xi_k(x_k) = \begin{cases} 0.1 & \text{if } \big\|\nabla\mathcal{I}_k(x_k)\big\|_2 > \sigma_c, \\ 1 & \text{otherwise.} \end{cases}$$

In the above formulation, the parameters are experimentally set as: $\beta = 4.0$, $\gamma = 1.6$, $\sigma_c = 8.0$ (when color values are between 0 and 255), $\varepsilon = 0.001$.

Equation (6) can be solved by the methods as described in [25] and [27]. With an all-zero initial solution, the result converges gradually along a fine pyramid of step 0.9, and each step is solved by the fixed point iteration. An example of this step is shown in Fig. 5, from which we can see that the initial solution is roughly correct but lacks accuracy.

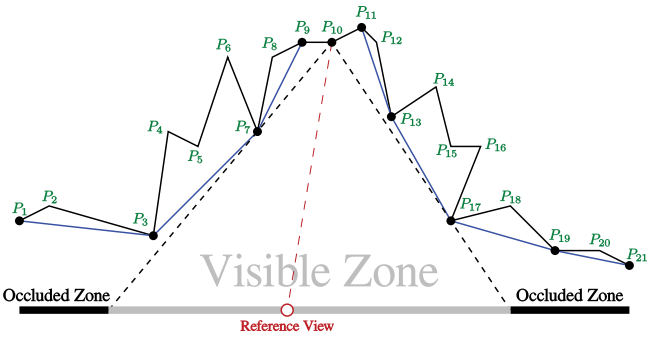


Fig. 6. The estimation of visible zone. Occlusion (as marked by black bold line) is caused by the convex hulls (as marked by blue line) of both sides.

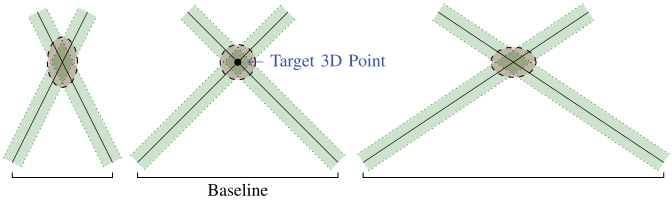


Fig. 7. Errors of different baseline setups. The case in the middle achieves lowest error.

B. Depth Refinement With Pixel-Wise Frame Selection

To achieve higher precision, information from more captured images should be utilized. However, during camera moving along the track, a lot of images are captured which contain too much redundancy and some regions of the reference view may be invisible in certain views due to occlusions or out-of-FOV. If we simply add all images into the optimization, it will consume too much computation resource and the result may be disturbed by occlusions and/or out-of-FOV. Therefore, a proper strategy is required to squeeze useful information from the captured images, which can save computation resource and inhibit those predictable disturbance at the same time.

Conventional methods [5], [6] select several frames from all captured images, and other frames are simply discarded during the later procedure. Besides, studies on "Next Best View Planning" (e.g., [29]–[31]) adopt similar idea, except that they focus on expanding the set of captured frames. In this paper, we want to consider all pixels, and select optimal frames for every pixel, so that useful information could be efficiently extracted. Similar idea has also been studied in [32]–[34]. Farid *et al.* [32] simply select left-hand-side or right-hand-side views according to intensity correlation errors, without considering the length of baseline between views, which is necessary for judging occlusions. Gu *et al.* [33] propose a pseudo selection algorithm by assigning different weights to different frames, but the computation cost of depth estimation can not be reduced because all frames have to be involved in the calculation.

The selection algorithm of [34] is most related to the proposed pixel-wise selection algorithm. It uses depth maps of all the frames to judge occlusions and out-of-FoV. However, when the number of frames n is large (this matches the case of the proposed sliding camera system), calculating depth maps of all

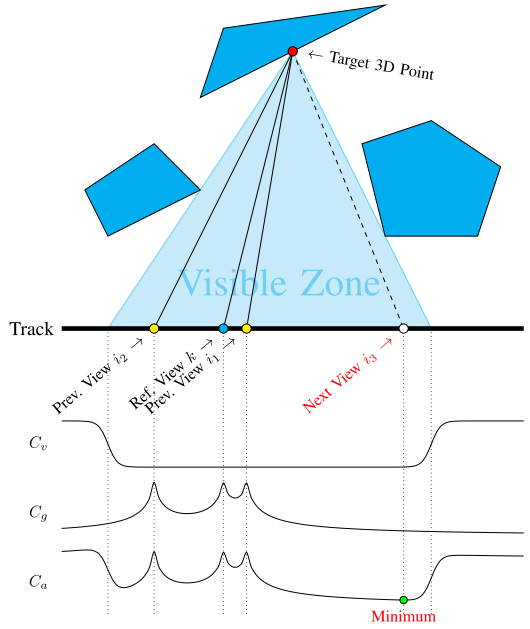


Fig. 8. Illustration for selection score. The visibility cost C_v is derived from the visible zone, and the precision gain cost C_g is calculated from the reference view (the k-th) as well as those already selected views (the i_1-th and the i_2-th). By adding C_v and C_g together, the next selection can be deduced from the minimum of C_a (the i_3-th in this illustration).

frames is extremely expensive. Therefore, we design an efficient selection strategy here to handle lots of available frames.

For every pixel of the reference view, our strategy consists of three sequential steps: 1) visible zone analysis, 2) precision gain estimation, and 3) greedy frame selection according to the information acquired by the previous steps.

1) Visible Zone Analysis: While occlusion reasoning is an essential part for frame selection algorithms, pixel-wise occlusion reasoning usually consumes too much time when there are a large number of frames, so it is obviated in conventional methods. However, in the case of (approximately) linear camera motion, we found this task can be significantly simplified. When moving straightly, the optical center of the camera is constrained on the same spatial straight line, which is the common epipolar axis of all view pairs. This property guarantees that pixel-wise occlusion relationship can be inferred by individually analyzing each epipolar line instead of each pixel.

To be specific, we first choose a group of uniformly distributed epipolar lines which can cover all pixels in the reference view (within a width of 2 pixels in our experiments), then individually analyze occlusions on these epipolar lines. Using the initial depth map calculated in the previous step, the reference view can be projected into 3D space as a mesh grid. For each epipolar line, the intersection of corresponding epipolar plane and the mesh grid is a 3D curve. As illustrated in Fig. 6, an algorithm is required to find out the visible zone of all points, $\{P_i\}$, on the 3D curve.

A straightforward algorithm for this task is, for any point P_i on the curve, traversing all other points, calculating geometric relationships and keeping extremes on both sides, which determines the visible zone of P_i. Take point P_{10} in Fig. 6 for example, this algorithm traverses $\{P_1, P_2, \ldots, P_9\}$ and

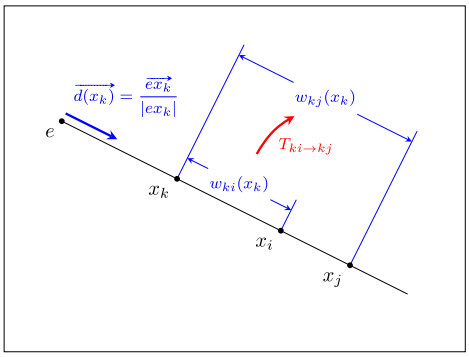


Fig. 9. $T_{ki\to kj}$ encodes the cross-ratio property of our system, which transforms coordinates between different view pairs. $\omega_i(x_k)$ is the disparity of x_k between view k and view i, so as $\omega_j(x_k)$.

$\{P_{11}, P_{12}, \ldots, P_{21}\}$ and finds out the visible zone between P_7 and P_{17}. To calculate visible zones for all points, the temporal complexity of this algorithm is $O(n^2)$. However, by comprehensively analyzing this problem, we found that convex hulls could accelerate visible zone calculation.

Take point P_{10} in Fig. 6 for example again, only the points on the convex hulls $\{P_1, P_3, P_7, P_9\}$ and $\{P_{11}, P_{13}, P_{17}, P_{19}, P_{21}\}$ on both sides should be taken into account. For all points on the curve, only $2n-2$ convex hulls are required, i.e., $f_{ch}\big(\{P_1, P_2, \ldots, P_k\}\big), k \in \{1, 2, \ldots, n-1\}$ and $f_{ch}\big(\{P_k, P_{k+1}, \ldots, P_n\}\big), k \in \{2, 3, \ldots, n\}$, where n is the number of points on the curve, and f_{ch} extracts convex hull from a set of points. Besides, for our task, the f_{ch} has the following recursive relationship as

$$f_{ch}\big(\{P_1, P_2, \ldots, P_k\}\big) = f_{ch}\Big(f_{ch}\big(\{P_1, P_2, \ldots, P_{k-1}\}\big) \cup \{P_k\}\Big),$$

and

$$f_{ch}\big(\{P_k, P_{k+1}, \ldots, P_n\}\big) = f_{ch}\Big(\{P_k\} \cup f_{ch}\big(\{P_{k+1}, P_{k+2}, \ldots, P_n\}\big)\Big).$$

Utilizing this recursive relationship, this task can be done with temporal complexity of $O(n)$. In our experiments, this procedure takes less than one second for an 800×600 image. Finally, these results are intersected with FOV of cameras which produces visibility maps denoted by C_v.

2) Precision Gain Estimation: With the visibility map calculated in the previous step, we are able to pick out frame candidates in which the 3D voxel can be seen. In this step, we try to pick out one frame from these candidates, so that we could minimize the error of depth estimation. A common knowledge on dual-view camera system is that the error of depth measurement varies inversely with the length of baseline, but for a multi-view stereo system a proper error model is required.

Here we use a simple Gaussian model. Suppose the projection on a certain view of a 3D voxel, $X = (x, y, z)$, is a 3D straight line, $l = X + k(X - O)$, with a cylinder uncertainty as in Fig. 7. The fused uncertainty of all observations could be expressed by a quadratic kernel (which is the exponential term of a Gaussian kernel). The uncertainty of a 3D straight

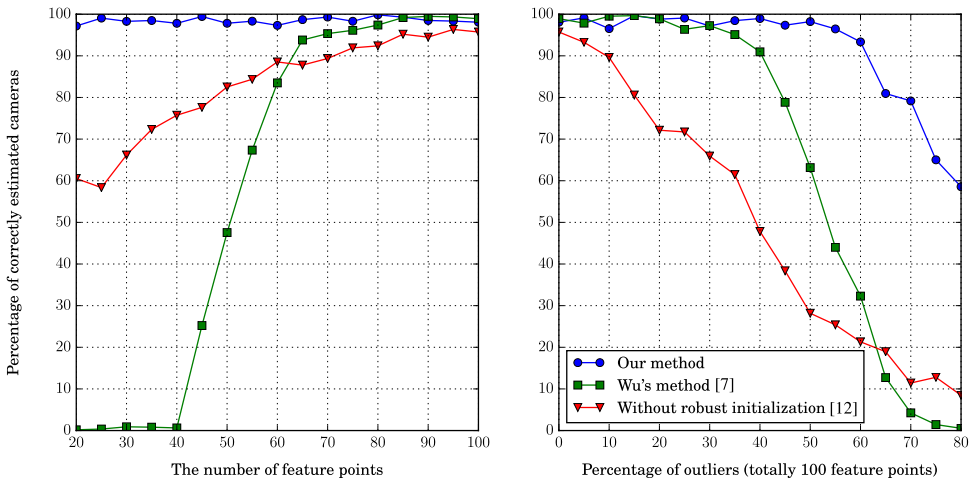


Fig. 10. The performance of camera pose initialization with varying numbers of feature points and outliers on synthetic data.

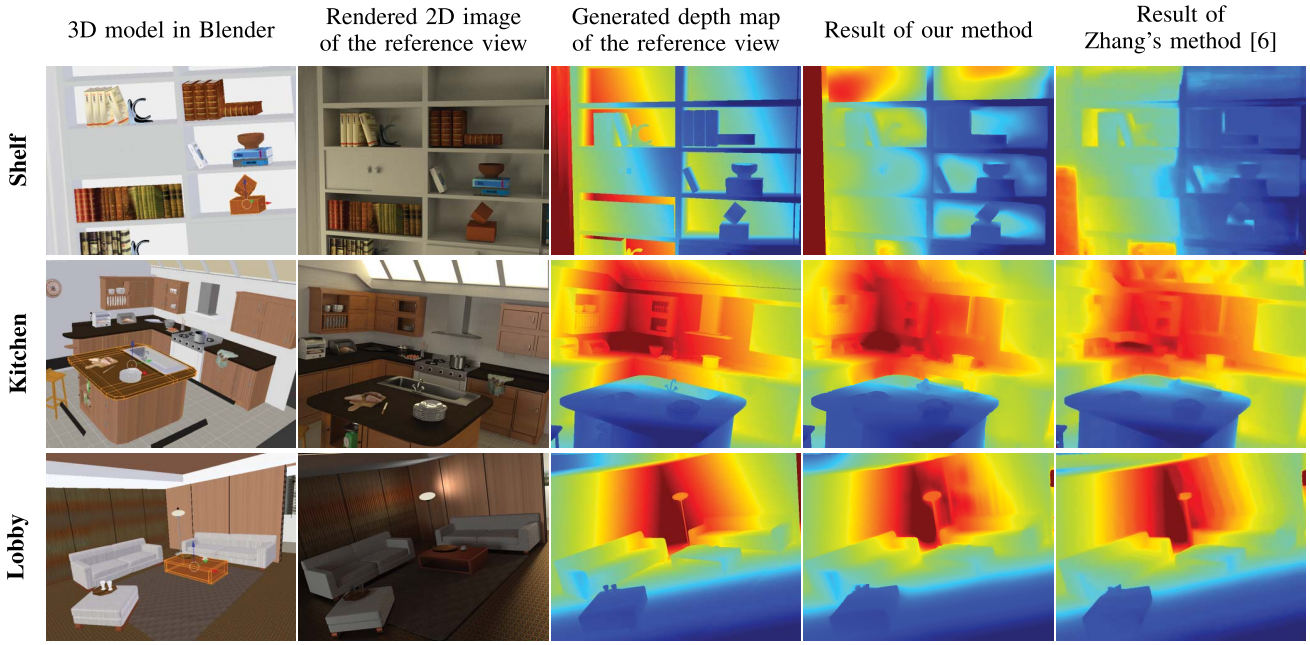


Fig. 11. Our synthetic data set and results. The first column shows 3D models in Blender. The second and third show rendered 2D images and generated ground truth depth maps. The last two show the results of our method and Zhang's method [6].

line, $X + k(X - O)$, can be expressed by a function matrix, $\Sigma_l = \left(\vec{d} \cdot \vec{d}^{\mathrm{T}} + \varepsilon \mathbf{I}\right)^{-1}$, where $\vec{d} = \overrightarrow{OX} / |OX|$. To get the best precision, we try to minimize the following cost as

$$C_g = \left(\sum_l \Sigma_l^{-1}\right)^{-1}.$$

After the calculation of C_v and C_g, we let $C_a = C_v + C_g$, which is illustrated in Fig. 8. By minimizing C_a, we could get the next best selection.

3) Iterative Frame Selection: We have described how we select one frame from the candidates. Now we need to do this iteratively to add multiple frames for depth refinement, as described in Algorithm 2. With specific selections, we can update the depth map $\mathcal{D}$ with the following refining optimization, which is derived from Equation (6), as

$$\min_{w_{ki}} \; E_D(\omega_{ki}) + \gamma \, E_S^{\mathrm{sel}}(\omega_{ki}). \tag{8}$$

where

$$E_D^{\mathrm{sel}}(\omega_{ki}) = \int_\Omega c(x_k) \sum_{j \in S(x_k)} \Psi\Big(\big\| \mathcal{I}_k^*(x_k) - \mathcal{I}_j^*\big(x_k + T_{ki \to kj}(w_{ki})(x_k) \cdot \overrightarrow{d(x_k)}\big)\big\|_2^2\Big) \mathrm{d}x_k. \tag{9}$$

In this equation, $T_{ki \to kj}$ encodes the cross-ratio property of our system, as shown in Fig. 9.

This optimization is fast in practice since: 1) there is already an initial solution calculated by Equation (6), and 2) only the selected data should be involved in the calculation.

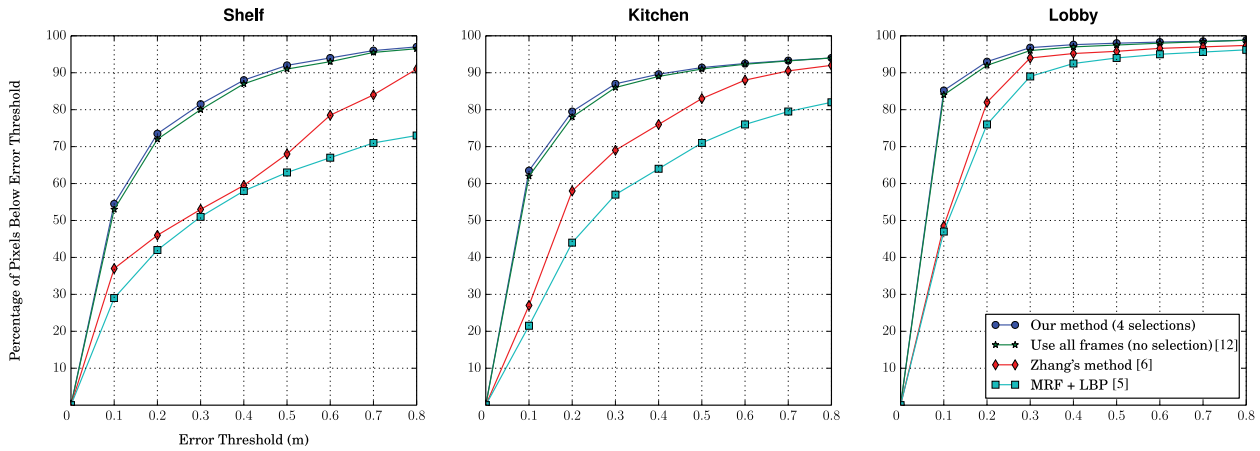


Fig. 12. Statistical comparison between different depth estimation methods on the synthetic data set. Our method outperforms conventional methods especially in the regions where the depth varies continuously. By the way, the proposed pixel-wise frame selection algorithm is slightly better than our previous work [12].

Algorithm 2 Iterative Frame Selection

$\mathcal{D} \leftarrow$ initial depth map;
foreach $x \in \Omega$ **do**
  $\mathcal{S}_x \leftarrow \emptyset$;
end
for i_{iter} from 1 to N_{iter} **do**
  Calculate visibility map C_v based on $\mathcal{D}$;
  Calculate precision gain C_g based on $\mathcal{D}$;
  foreach $x \in \Omega$ **do**
    Find the best frame j according to C_v and C_g;
    $\mathcal{S}_x \leftarrow \mathcal{S}_x \cup \{j\}$;
  end
  Update $\mathcal{D}$ based on current selections;
end

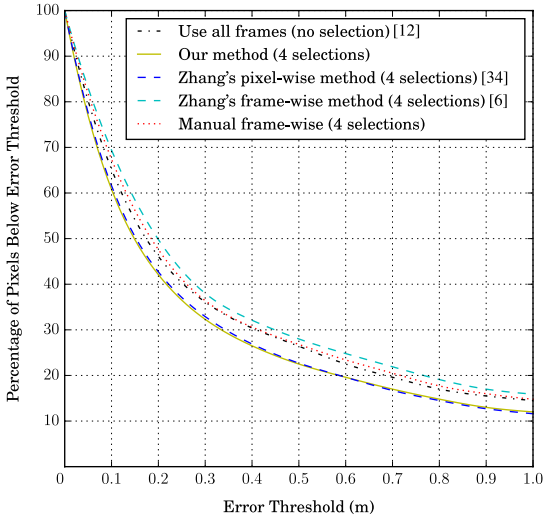


Fig. 13. Comparison on precision of different frame selection strategies. To make the difference prominent, these curves are calculated on occluded regions. These curves show our pixel-wise frame selection algorithm achieves highest precision on occluded regions.

V. Experiments

In this section, we will first carry out experiments on the synthetic data (with ground truth) to evaluate the performance of both pose estimation and depth estimation algorithms. We will also test the practicality of the proposed algorithms on real data.

A. Quantitative Evaluation on Synthetic Data

1) Camera Pose Initialization: We carry out simulation experiments to avoid the interference of other factors. A set of 3D points $\{X^{(l)}\}$ are randomly generated, then these 3D points are projected onto a linearly aligned camera array. With Gaussian noise and FOV cropping, we get a set of 2D projections, $\{x_k^{(l)}\}$, and visibilities, $\{w_k^{(l)}\}$. Based on these synthetic data, we have compared the results of our pose initialization algorithm with the unconstrained one [7], which is a state-of-the-art method for SfM without IMU or GPS. To evaluate the results, we first manually find a global transformation to align the estimated camera structure with ground truth (since image-based SfM algorithms can not retrieve global poses), and then cameras within an error threshold (10% of the total array length for location, 10° for orientation) are considered as correct. In this experiment, outliers are generated by adding salt-and-pepper noise to the 2D projections.

The quantitative results are illustrated in Fig. 10. The left plot shows the performances when the number of feature points varies, and the right plot shows the performance when the percentage of outliers (of totally 100 feature points) varies. These results show that our initialization algorithm is robust to the number of feature points as well as outliers. Wu's method [7] fails when the number of common feature points seen by all the views is less than 20 (50 in total), or when the proportion of outliers is large than 50%. Without this initialization procedure, our previous method [12] fails with bad inputs, especially when there are lots of outliers.

2) Depth Estimation: To quantitatively evaluate the accuracy of depth estimation, we have built a synthetic data set with Blender,[4] a free and open-source software which renders 3D scene models into 2D images with depth maps. Our synthetic data set contains three image sequences, i.e., the "Shelf",

[4]http://www.blender.org/

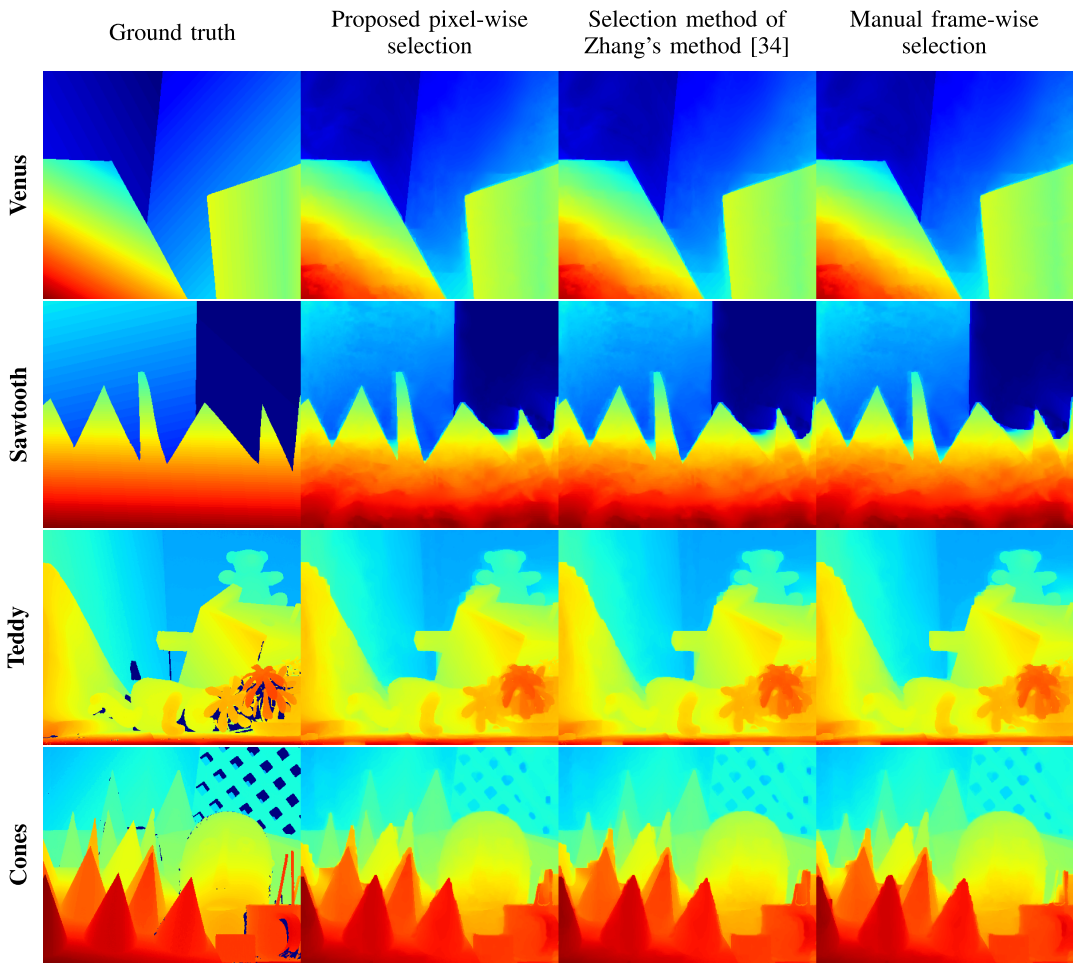


Fig. 14. Visual comparison of different frame selection strategies on the Middlebury data set.

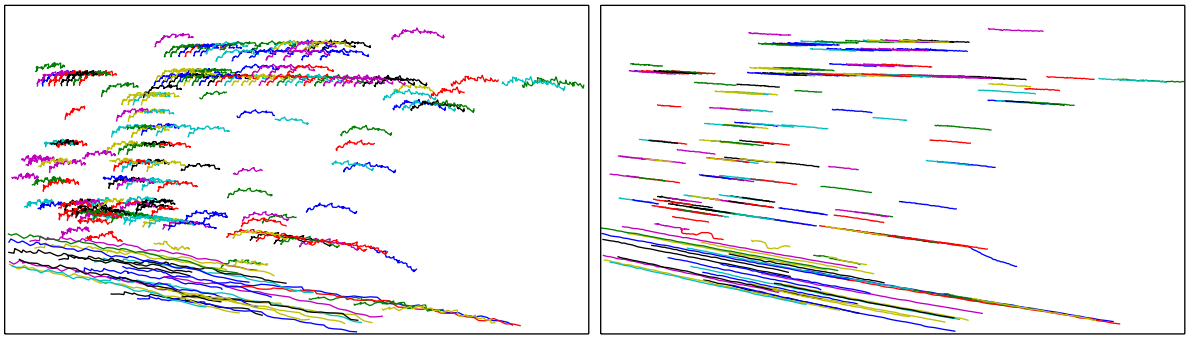
Fig. 15. Samples of images before (the first row) and after (the second row, images are slightly warped so black borders appear) vibration compensation.

the "Kitchen" and the "Lobby". Each sequence consists of 51 images rendered from a virtual linear camera array. The "Shelf" is a degenerate case and the most difficult one. The "Kitchen" contains lots of depth discontinuities. The "Lobby" contains large proportion of plane areas. Some samples of these scenes are shown in Fig. 11, and depth ranges are shown in Table I. Using the ground-truth depth maps generated by Blender, we are able to calculate errors of depth estimations quantitatively.

We first carry out experiments to compare our variational framework with conventional MRF-based methods.[5] Fig. 12 shows the quantitative results of accuracy, where our variational method outperforms state-of-the-art MRF-based methods [5], [6], especially in the regions where depth varies continuously. Besides, the variational framework consumes less memory (up to 300 MB, with 800×600 resolution) than MRF-based methods (up to 4 GB with 100 quantization levels).

Then we carry out quantitative comparison on accuracy for different selection strategies, including our pixel-wise selection using visible zone analysis, pixel-wise selection using depth maps of other frames [34], conventional frame-wise selection (we use Zhang's selection strategy [6], which selects the views with higher overlapping FOV), manual frame-wise selection (we manually choose the best combination within hundreds

[5]The results of Zhang's method [6] reported in this paper are obtained using the ACTS software (http://www.zjucvg.net/acts/acts.html).



(a) (b)

Fig. 16. Trajectories of feature points before (a) and after vibration compensation (b). The figure in (b) shows our vibration compensation algorithm successfully aligns the captured images.

TABLE I
DEPTH RANGES OF OUR SYNTHETIC DATA SET

Name	Minimal Depth (m)	Maximal Depth (m)
Shelf	9.05	11.34
Kitchen	6.52	15.22
Lobby	3.45	9.13

of good candidates) and no selection (when all frames are used [12]). The overall accuracies (including all pixels in the image) of different strategies are close, but when focusing on the regions where pixels may be occluded, the accuracy of our pixel-wise selection strategy is much higher than others. As shown in Fig. 13: 1) Our pixel-wise selection strategy (with only 4 frames selected per pixel) outperforms other frame-wise selection methods, even including the one using all frames. 2) Although the selection algorithm of [34] is as accurate as ours, it is n times slower than our algorithm because it has to calculate depth maps of all available frames (here n is the number of available frames).

Further more, for a sequence consisting of 51 images with 800×600 resolution, camera pose estimation takes less than 10 seconds (including pose initialization and bundle optimization). Depth initialization, iterative frame selection and depth refinement of the reference view take about 50 seconds, 5 seconds and 40 seconds, respectively.[6] All these time values are measured using our Python + Cython code on a 3.0 GHz CPU with single thread. Note that since pixel-wise frame selection can run in parallel, and variational model can also be solved in parallel by GPU [35], it is possible to reduce the total computation time to a few seconds with an optimized implementation.

We also tested the proposed selection algorithm on the Middlebury data set [36], [37]. This data set is widely used as a benchmark for stereo estimation algorithms, and it includes several multi-baseline inputs which are suitable for testing the proposed algorithm. We pick 4 groups of input (Venus, Sawtooth, Teddy and Cones) because these groups contains 9 views each. Fig. 14 and Table II compare the results between the proposed pixel-wise selection algorithm and other strategies. These results tally with the conclusion from our synthetic data set.

[6]For the same resolution, algorithm of [5] takes about 40 seconds for one frame (with 100 quantization levels), and algorithm of [6] takes about 150 seconds (in average) for one frame.

TABLE II
PERCENTAGE OF BAD PIXELS (ERROR THRESHOLD = 1.0) ON THE MIDDLEBURY DATA SET

	Proposed pixel-wise selection	Selection method of Zhang's method [34]	Manual frame-wise selection
Venus	**1.54** %	1.57 %	1.83 %
Sawtooth	4.31 %	**4.27** %	4.98 %
Teddy	10.82 %	**10.67** %	11.89 %
Cones	**12.40** %	12.62 %	13.51 %
Average	**7.27** %	7.28 %	8.05 %

TABLE III
ACCURACY WITH/WITHOUT VIBRATION COMPENSATION

Data set	Algorithm	Percentage of pixels within error threshold of 0.1 m
Without V. C.	Without V. C.	67.73 %
	With V. C.	67.70 %
With V. C.	Without V. C.	35.14 %
	With V. C.	66.89 %

To test the efficiency of our vibration compensation algorithm, we simulated camera vibrations in the synthetic data set, and carried out comparative experiments on data set with/without vibrations by switching on/off the compensation. The quantitative results are shown in Table III, where the accuracy within 0.1 m is used as judgement. When tested on the data set without vibrations, the algorithm with compensation is as accurate as the one without. But when tested on the data set with vibrations, the compensation is necessary, otherwise the accuracy will drop by a large amount.

B. Evaluation on Real-World Data

To validate the practicality of our algorithm, we carry out experiments on real-world scenes. First, we captured several

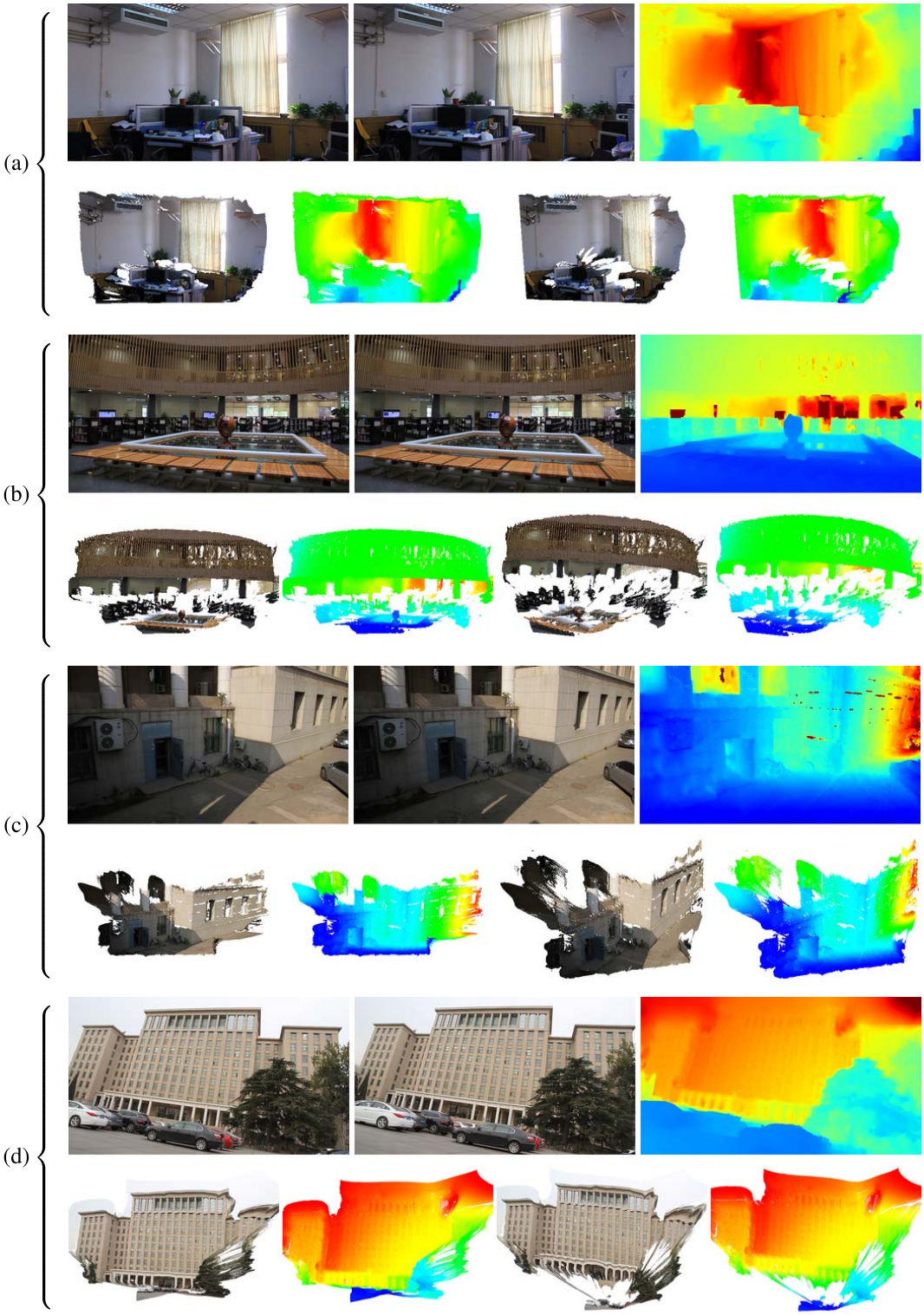


Fig. 17. Depth estimation results for real scenes. Each odd row contains two samples of captured images and the restored depth map. Each even row shows two groups (from two different view angles) of rendered point clouds. The last group is with vibration compensation.

image sequences with the proposed sliding camera system (as in Fig. 1) so that the camera poses are strictly constrained by the track, and then applied our whole pipeline on these data. The first three groups in Fig. 17 (namely (a), (b) and (c)) show samples of these images as well as the final outputs. Group (a) is an indoor scene which contains various kinds of objects. Our algorithm can successfully restore the depth map and the precision of regions with texture is good (e.g., the curtains and things on the desk). The result of textureless areas (e.g., the white wall) is less accurate. Group (b) is also an indoor scene but with a lot of repeated objects (e.g., stools, shelves and fences), which is much more complicated than Group (a). Conventional methods [5], [6] fail to retrieve the depth map under this circumstance, but our algorithm can

still successfully restore the 3D structure of all these repeated objects. Group (c) is an outdoor scene with weak textures on the floor and walls. Although there are no ground-truth, the smoothness of the ground plane and walls indicates good precision of our algorithm.

We also carried out an experiment on an image sequence captured by a camera mounted on a bicycle, which moves under approximately linear translation but with a lot of disturbances. Fig. 16(a) illustrates the outputs (the longest top-100) of Lucas-Kanade tracker [20] on original input images, where the trajectories are all zigzag. In this situation, our previous algorithm [12] could not get a good output. This sequence is also successfully recovered by the proposed algorithm, and Fig. 15 shows image samples before and after applying our vibration compensation algorithm (described in the beginning of Section IV). Fig. 16(b) shows the output trajectories of Lucas-Kanade tracker on compensated images, where the trajectories are almost straight lines. The final outputs are shown in Fig. 17(d), where the 3D structure of the buildings, trees and cars are successfully restored.

VI. Conclusion and Future Work

In this paper, we have presented a novel depth estimation algorithm based on a sliding camera system, which includes a camera pose initialization algorithm and an adaptive iterative optimal frame selection algorithm for stereo matching.

We have analyzed the properties of the sliding camera system. After that, camera pose initialization algorithm is designed to utilize the geometric properties of linearly camera translation, which can work even with only a small number of feature points and is robust to noise. Iterative optimal frame selection algorithm is proposed for pixels corresponding to different depths, which can take advantage of continuously pose-changing imaging and reduce time consumption a lot. The proposed algorithm can also be easily extended to handle less constrained situations (such as using a camera mounted on a moving robot or vehicle). The experiments on synthetic data set show the proposed camera pose initialization algorithm and pixel-wise frame selection algorithm outperform conventional methods. The practicality of the proposed algorithms is also verified on real-world data.

Nevertheless, the proposed depth estimation algorithm is currently unable to handle scenes with moving objects. In the future, we will try to model moving objects according to sparse or low-rank representation, and improve it accordingly.

Acknowledgment

The authors do appreciate Ali Gonzalez,[7] the author of the high quality 3D models they used to build our data set; and they also appreciate the Blend Swap[8] for providing such a good 3D model sharing platform.

References

[1] S. Sakamoto, I. J. Cox, and J. Tajima, "A multiple-baseline stereo for precise human face acquisition," in *Proc. 1st Int. Conf. AVBPA*, 1997, pp. 419–428.

[2] J. Jeon, K. Kim, C. Kim, and Y.-S. Ho, "A robust stereo-matching algorithm using multiple-baseline cameras," in *Proc. IEEE Pacific Rim Conf. Commun., Comput. Signal Process.*, vol. 1. Aug. 2001, pp. 263–266.

[3] M. Okutomi and T. Kanade, "A multiple-baseline stereo," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 4, pp. 353–363, Apr. 1993.

[4] D. Gallup, J.-M. Frahm, P. Mordohai, and M. Pollefeys, "Variable baseline/resolution stereo," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

[5] S. B. Kang and R. Szeliski, "Extracting view-dependent depth maps from a collection of images," *Int. J. Comput. Vis.*, vol. 58, no. 2, pp. 139–163, 2004.

[6] G. Zhang, J. Jia, T.-T. Wong, and H. Bao, "Consistent depth maps recovery from a video sequence," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 6, pp. 974–988, Jun. 2009.

[7] C. Wu, S. Agarwal, B. Curless, and S. M. Seitz, "Multicore bundle adjustment," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 3057–3064.

[8] P. Pinies, T. Lupton, S. Sukkarieh, and J. D. Tardos, "Inertial aiding of inverse depth SLAM using a monocular camera," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 2007, pp. 2797–2802.

[9] S. Weiss and R. Siegwart, "Real-time metric state estimation for modular vision-inertial systems," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 4531–4537.

[10] M. Lhuillier, "Incremental fusion of structure-from-motion and GPS using constrained bundle adjustments," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 12, pp. 2489–2495, Dec. 2012.

[11] D. J. Crandall, A. Owens, N. Snavely, and D. P. Huttenlocher, "SfM with MRFs: Discrete-continuous optimization for large-scale structure from motion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 12, pp. 2841–2853, Dec. 2013.

[12] K. Ge, J. Feng, and J. Zhou, "Dense and continuous depth estimation using a sliding camera," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Apr. 2015, pp. 1181–1185.

[13] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge U.K.: Cambridge Univ. Press, 2003.

[14] T. Moons, L. Van Gool, M. Proesmans, and E. Pauwels, "Affine reconstruction from perspective image pairs with a relative object-camera translation in between," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 1, pp. 77–83, Jan. 1996.

[15] Z. Chen, N. Pears, J. McDermid, and T. Heseltine, "Epipole estimation under pure camera translation," in *Proc. 7th Int. Conf. Digital Image Comput., Techn. Appl.*, 2003, pp. 849–858.

[16] Z. Hu and Z. Tan, "Depth recovery and affine reconstruction under camera pure translation," *Pattern Recognit.*, vol. 40, no. 10, pp. 2826–2836, 2007.

[17] B. Triggs, P. E. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment—A modern synthesis," in *Proc. Int. Workshop Vis. Algorithms*, Corfu, Greece, Sep. 1999, pp. 298–372.

[18] J. Nilsson, J. Fredriksson, and A. C. E. Odblom, "Bundle adjustment using single-track vehicle model," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2013, pp. 2888–2893.

[19] J. Shi and C. Tomasi, "Good features to track," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 1994, pp. 593–600.

[20] J.-Y. Bouguet, "Pyramidal implementation of the affine Lucas Kanade feature tracker description of the algorithm," *Intel Corp.*, vol. 2, no. 1, pp. 1–10, 2001.

[21] J.-P. Tardif, "Non-iterative approach for fast and accurate vanishing point detection," in *Proc. 12th IEEE Int. Conf. Comput. Vis.*, Sep./Oct. 2009, pp. 1250–1257.

[22] V. Kolmogorov and R. Zabih, "Computing visual correspondence with occlusions using graph cuts," in *Proc. 8th IEEE Int. Conf. Comput. Vis.*, vol. 2. Jul. 2001, pp. 508–515.

[23] J. Sun, N.-N. Zheng, and H.-Y. Shum, "Stereo matching using belief propagation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 7, pp. 787–800, Jul. 2003.

[24] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient belief propagation for early vision," *Int. J. Comput. Vis.*, vol. 70, no. 1, pp. 41–54, 2006.

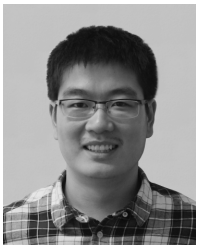
[25] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, "High accuracy optical flow estimation based on a theory for warping," in *Proc. 8th Eur. Conf. Comput. Vis.*, 3024, pp. 25–36.

[26] N. Slesareva, A. Bruhn, and J. Weickert, "Optic flow goes stereo: A variational method for estimating discontinuity-preserving dense disparity maps," in *Proc. 27th DAGM Symp.*, Vienna, Austria, Aug. 2005, pp. 33–40.

[7] http://www.blendswap.com/user/oldtimer

[8] http://www.blendswap.com/

[27] Y. Liu, X. Cao, Q. Dai, and W. Xu, "Continuous depth estimation for multi-view stereo," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 2121–2128.
[28] R. Ranftl, S. Gehrig, T. Pock, and H. Bischof, "Pushing the limits of stereo using variational stereo estimation," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2012, pp. 401–407.
[29] A. Hornung, B. Zeng, and L. Kobbelt, "Image selection for improved multi-view stereo," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.
[30] M. Krainin, B. Curless, and D. Fox, "Autonomous generation of complete 3D object models using next best view manipulation planning," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 5031–5037.
[31] C. Freundlich, P. Mordohai, and M. M. Zavlanos, "A hybrid control approach to the next-best-view problem using stereo vision," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2013, pp. 4493–4498.
[32] H. Farid, S. W. Lee, and R. Bajcsy, "View selection strategies for multi-view, wide-baseline stereo," Dept. Comput. Inf. Sci., Univ. Pennsylvania, Philadelphia, PA, USA, Tech. Rep. MS-CIS-94-18, May 1994.
[33] Z. Gu, X. Su, and J. Yang, "Robust view selection in multiview stereo," *Chin. Opt. Lett.*, vol. 7, no. 3, pp. 198–200, 2009.
[34] G. Zhang, J. Jia, and H. Bao, "Simultaneous multi-body stereo and segmentation," in *Proc. IEEE Int. Conf. Comput. Vis.*, Nov. 2011, pp. 826–833.
[35] A. Wedel, T. Pock, C. Zach, H. Bischof, and D. Cremers, "An improved algorithm for TV-L^1 optical flow," in *Proc. Int. Dagstuhl Seminar*, Dagstuhl Castle, Germany, Jul. 2008, pp. 33–40.
[36] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *Int. J. Comput. Vis.*, vol. 47, nos. 1–3, pp. 7–42, 2002.
[37] D. Scharstein and R. Szeliski, "High-accuracy stereo depth maps using structured light," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 1. Jun. 2003, pp. I-195–I-202.



Kailin Ge received the B.S. degree from the Department of Automation, Tsinghua University, Beijing, China, in 2009, where he is currently pursuing the Ph.D. degree. His research interests include intelligent visual surveillance, stereo acquisition, structure from motion, and multiview stereo.

Han Hu received the B.S. degree from Tsinghua University, China, in 2008, and the Ph.D. degree in computer vision and image processing from the Department of Automation, Tsinghua University, in 2014. He is currently a Researcher with the Institute of Deep Learning, Baidu Research, Beijing, China. His current research interests are mainly on convolutional neural network and optical character recognition.

Jianjiang Feng (M'10) received the B.S. and Ph.D. degrees from the School of Telecommunication Engineering, Beijing University of Posts and Telecommunications, China, in 2000 and 2007, respectively. From 2008 to 2009, he was a Post-Doctoral Researcher with the PRIP Laboratory, Michigan State University. He is currently an Associate Professor with the Department of Automation, Tsinghua University, Beijing. His research interests include fingerprint recognition and computer vision. He is also an Associate Editor of *Image and Vision Computing*.

Jie Zhou (M'01–SM'04) was born in 1968. He received the B.S. and M.S. degrees from the Department of Mathematics, Nankai University, Tianjin, China, in 1990 and 1992, respectively, and the Ph.D. degree from the Institute of Pattern Recognition and Artificial Intelligence, Huazhong University of Science and Technology, Wuhan, China, in 1995. From 1995 to 1997, he served as a Post-Doctoral Fellow with the Department of Automation, Tsinghua University, Beijing, China. Since 2003, he has been a Full Professor with the Department of Automation, Tsinghua University. In recent years, he has authored more than 100 papers in peer-reviewed journals and conferences. Among them, more than 30 papers have been published in top journals and conferences, such as the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, the IEEE TRANSACTIONS ON IMAGE PROCESSING, and *Computer Vision and Pattern Recognition*. His research area includes computer vision, pattern recognition, and image processing. He was a recipient of the National Outstanding Youth Foundation of China. He is also an Associate Editor of the *International Journal of Robotics and Automation*, *Acta Automatica*, and two other journals.