Relative Camera Refinement for Accurate Dense Reconstruction

Yao Yao
yyaoag@cse.ust.hk

Shiwei Li
slibc@cse.ust.hk

Siyu Zhu
szhu@cse.ust.hk

Hanyu Deng
hdeng@cse.ust.hk

Tian Fang*
tianft@cse.ust.hk

Long Quan
quan@cse.ust.hk

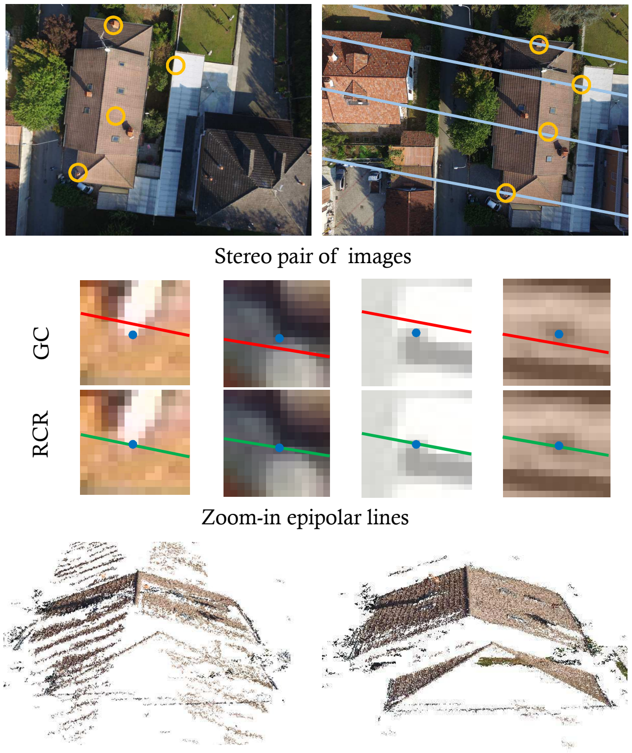
The Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong

Abstract

Multi-view stereo (MVS) depends on the pre-determined camera geometry, often from structure from motion (SfM) or simultaneous localization and mapping (SLAM). However, cameras may not be locally optimal for dense stereo matching, especially when it comes from the large scale SfM or the SLAM with multiple sensor fusion. In this paper, we propose a local camera refinement approach for accurate dense reconstruction. Firstly, we refines the relative geometry of independent camera pair using a tailored bundle adjustment. The refinement is also extended to a multi-view version for general MVS reconstructions. Then, the non-rigid dense alignment is formulated as an inverse-distortion problem to transfer point clouds from each local coordinate system to a global coordinate system. The proposed framework has been intensively validated in both SfM and SLAM based dense reconstructions. Results on different datasets show that our method can significantly improve the dense reconstruction quality.

1. Introduction

Modern methods of image-based 3D reconstruction separate the problem into camera geometry reconstruction and multi-view stereo (MVS), where the former one computes camera parameters of each image capture, passing to the latter to reconstruct dense representation. Research on MVS assumes camera parameters are confidently given, and only focuses on the dense reconstruction part. In fact, the accuracy of camera parameter itself is critical to the reconstruction quality, as the core of stereo is the pixelwise matching problem with epipolar line constraint. In another word, the slight inaccuracy of camera geometry would significantly spoil the quality of the dense reconstruction.



Stereo pair of images

Zoom-in epipolar lines

Dense results using global cameras and refined cameras

Figure 1. GC: Global Cameras of standard SfM. RCR: our Relative Camera Refinement method. Top: image pair with 4 matching points and epipolar lines. Middle: zoom-in point locations and epipolar lines. Bottom: reconstructed point cloud using global cameras (left) and our refined cameras (right).

While different benchmarks [36, 40, 17] provide ground truth camera parameters, in real-world reconstructions the camera geometry could hardly be perfect for stereo matching. Firstly, for image-based 3D modeling, as images are usually taken under different settings, scales and illuminations, mismatches of image features are inevitable, and er-

*Tian Fang is with Shenzhen Zhuke Innovation Technology since 2017.

rors would accumulate and propagate to the whole SfM or SLAM reconstruction. Also, when it comes to a large-scale SfM, optimizing thousands of hundreds of pose parameters is computationally difficult, and the posibility of stucking in local minima increases as well. Secondly, to make the camera geometry reconstruction more stable, measurements from other sensors like inertial measurement unit (IMU) [22, 37, 23], global positioning system (GPS) [46] and ground control points (GCP) are fused with the image-based visual measurements. The fusion of different measurements is widely applied in SLAM research, aiming for a stable and smooth camera trajectory reconstruction. However, due to its real-time nature, the accuracy of its camera pose is often not eligible for high-quality dense reconstruction. This long exsisting problem, however, attracts only a few attention from previous researchers [15, 48].

In our work, we highly consider the importance of the accurate camera geometry for dense reconstruction. Instead of directly using the camera pose yielded from SfM or SLAM for dense reconstruction, we add one more step in between, *i.e.*, the proposed *Relative Camera Refinement* (RCR). RCR fine-tunes the relative geometry of selected camera pairs, in order to increase the camera pose accuracy for high-quality dense reconstruction. More specifically, we apply a tailor-made bundle adjustment to iteratively refine each camera pair with respect to camera intrinsic, radial distortion and relative camera pose. Fine-scale features are progressively introduced to the camera refinement with careful mismatches filtering. After the pairwise refinement, we extend the RCR to the multi-view for general MVS reconstructions. Reported by the statistics in our experiments, the proposed method is able to robustly reduce the average reprojection error from around 2 pixels to around 0.3 pixel, which is crucial to the high-quality dense reconstruction.

One challenge brought by the local camera refinement is how to align dense results from different local coordinate systems back to a globally consistent coordinate system, since each camera group is optimized independently. This is not a concern for previous dense reconstructions where only one global camera system is used. But here, the transformations between point clouds from different camera groups are non-rigid and difficult to solve. To tackle this problem, we develop a novel dense alignment strategy to registers all dense results back to one global camera system. The key idea is to convert pixel matches in the local coordinate systems to a global coordinate system via inverse distortion, and then use the global cameras to triangulate the matches to get 3D points. With the proposed non-rigid alignment, local dense reconstructions are able to be consistently merged together for later processes.

The main contributions of this paper can be summarized as twofold:

- Proposing the RCR method to produce the precise relative camera geometry for stereo matching, which is then extended to a multi-view version that is crucial to the high quality dense reconstruction.
- Solving the non-rigid point cloud alignment problem between the local and the global camera systems by converting it into an inverse distortion problem.

2. Related Work

Global Camera Geometry Reconstruction Reconstructing accurate camera geometry for MVS is a widely studied topic in computer vision. Two important categories of techniques for camera geometry recovery are the SfM methods and the SLAM methods. The SfM methods usually run in offline and have achieved great success in the past decade, making large scale 3D reconstructions possible [38, 18, 2, 14, 9, 27, 35]. Bundle adjustment (BA) [42] is applied in the last step of SfM to minimize the reprojection error and refine camera parameters. Apart from the offline SfM methods, in robotic areas researchers apply the SLAM methods to reconstruct the camera odometry in real-time [10, 21, 30, 32, 12]. To eliminate the drift problem in odometry reconstructions, different sensors are fused together to get a robust trajectory. For example, in the context of visual-inertial odometry, filtering based techniques [29, 7, 19] or keyframe-based non-linear optimizations [23, 37, 20, 31] are applied to fuse the visual-inertial measurements, producing very stable trajectories over long periods of time. However, it also leads to oversmoothing of the individual poses, which in turn leads to problems during stereo matching.

Camera Refinement for MVS Other methods have been proposed to provide an even better geometry for MVS reconstruction. Lhuillier and Quan [24] propose a quasi-dense approach to reconstruct the geometry in an iteratively growing manner. Furukawa *et al.* [15] rescales images to appropriate resolutions so as to minimize the reprojection error, and utilizes selected dense points to help refine the camera geometry for PMVS [16]. Delaunoy *et al.* [11] propose to minimize photometric reprojection error between a generated model and the observed images to jointly refine cameras and model shape. These works, while refining camera parameters, maintain a globally consistent camera system for later MVS processes.

Instead of optimizing everything globally, the divide-and-conquer strategy is commonly used in the large-scale 3D reconstructions. The authors of [39, 33] partition the large BA problem into smaller and better conditioned sub-problems. To handle the scalability problem of MVS reconstruction, researches divide cameras into clusters [5, 14, 18, 8, 47] to partition the large-scale reconstruction problem into several smaller ones. The most related work to ours is the method proposed by Zhu *et al.* [48], which partitions large scale SfM to several local clusters, and applies the local camera refinement within each cluster for better dense reconstruction. However, in this paper the alignment step is ignored after the individual local camera refinement, resulting the potential dense stratification problem in the final reconstruction. In this paper, we again stress the importance of the local camera geometry, and proposed an efficient algorithm to solve the global dense alignment problem.

3. Local Camera Refinement

The key component of our local camera refinement is the robust two-view relative camera refinement (RCR) algorithm described in Section 3.1. Based on the RCR method, the multi-view camera refinement extensions are discussed in 3.2.

3.1. Relative Camera Refinement

Pairwise BA A tailor-made pairwise BA is designed as the basic tool used in our camera refinement. Different from the general method, given a camera pair $\{\mathbf{C}_l, \mathbf{C}_r\}$, the pairwise BA fixes the following parameters in the optimization: (1) the left camera rotation $\mathbf{R}_l$, (2) left camera center $\mathbf{X}_{c_l}$, and (3) the baseline length l_{bl} between the camera pair. These are to avoid slow convergence and even shrinking the whole scene to a singularity, since the scene is up to rotation, translation and scale during the optimization. It is noteworthy that the particular choice of fixing baseline length l_{bl} (instead of the right camera center $\mathbf{X}_{c_r}$) is because optimization of relative camera pose would be restricted if $\mathbf{R}_l$, $\mathbf{X}_{c_l}$ and $\mathbf{X}_{c_r}$ are all fixed. In our parameterization, the camera center $\mathbf{X}_{c_r}$ of the right camera can be expressed by two rotation angles θ (yaw and pitch) of the baseline. Let $\mathbf{P} = \{\mathbf{P}_i\}$ denotes the 3D point positions and $\mathbf{p} = \{\mathbf{p}_{ij}\}$ its corresponding feature position in image I_j, $\mathbf{K} = \{\mathbf{x}_c, f\}$ represents the camera intrinsic where $\mathbf{x}_c$ is the principle point and f the focal length, $\kappa = \{k_1, k_2, k_3\}$ denotes the camera distortion, then our BA can be formulated as:

$$\arg \min_{\mathbf{P},\mathbf{R}_r,\theta,\mathbf{K},\kappa} \sum_i \sum_{j=l,r} \rho\big(\pi_{\mathbf{C}_j}(\mathbf{P}_i) - \mathbf{p}_{ij}\big)^2 \tag{1}$$

Where $\rho(\cdot)$ is the robust penalty and:

$$\pi_{\mathbf{C}_j}(\mathbf{P}_i) = f_j \, \mathcal{F}_{\kappa_j}\big(\langle \mathbf{R}_j(\mathbf{P}_i - \mathbf{X}_{c_j})\rangle\big) + \mathbf{x}_{c_j} \tag{2}$$

$$\mathcal{F}_\kappa(\mathbf{u}) = \mathbf{u}(1 + k_1\|\mathbf{u}\|^2 + k_2\|\mathbf{u}\|^4 + k_3\|\mathbf{u}\|^6) \tag{3}$$

The operation $\langle [x, y, z] \rangle = [x/z, y/z]$ in equation 2 converts the homogeneous coordinate to the image coordinate. $\mathcal{F}_\kappa(\cdot)$ is the distortion equation and equation 2 projects a 3D point to its image coordinate. In solving the minimization problem, 3D point positions $\{\mathbf{P}_i\}$ are initialized by triangulating the feature matches between the camera pair. As fixing $\mathbf{R}_l$, $\mathbf{X}_{c_l}$, l_{bl} already reduces the DoF of the system by 7, and the input geometry provides a good initialization, our pairwise BA is able to converge in a small number of iterations (≤ 10) in most cases.

Progressive Refinement One factor affecting the BA accuracy is the uncertainty of the feature center coordinate. Features with larger scales are usually more robust to be matched between images with large pose variances, but provide less accurate measurements for the feature center coordinates. In our camera refinement, we prefer to select the fine-scale features to achieve utmost relative camera accuracy.

```
Algorithm 1 Relative Camera Refinement
Input: Global coarse geometry from SfM.
Output: Optimal relative geometry.
  Divide putative matches into groups $\{\mathbf{M}_1, \mathbf{M}_2, \mathbf{M}_3\}$ by
  match scales. Compute the average scales $\{\bar{s_i}\}$ for each
  group of matches.
  for $\mathbf{M}_i$ from coarse to fine scale do
    1st BA with Huber loss:
      Filter matches in $\mathbf{M}_i$ whose Sampson error $e > \bar{s_i}$.
      Refine cameras using $\mathbf{M}_i$ and BA with Huber loss.
    2nd standard L2 BA:
      Recompute Sampson error and filter $\mathbf{M}_i$ again.
      Perform standard BA with new matches.
  end for
```

For SfM based reconstruction, firstly SIFT features are extracted in the full scale-space, and then the putative matches are computed using the approximate nearest matching method [26] (this two steps should have been done in the SfM). An observation is when feature scale goes down, the match can provide a more accurate measurement but is more vulnerable to mismatch. According to this, we cluster matches into k groups $\{\mathbf{M}_0, \mathbf{M}_1, ..., \mathbf{M}_{k-1}\}$ by the maximum scale of its two features $s_i = \max(s_1, s_2)$. The camera pair will then be refined using each match group from coarse to fine scale. In our experiments, we set $k = 3$ and $\mathbf{M}_0 = \{m_i \mid s_i > 8\}$, $\mathbf{M}_1 = \{m_i \mid 4 < s_i \leq 8\}$, $\mathbf{M}_2 = \{m_i \mid s_i \leq 4\}$. In between refinements using matches with different scales, the refined geometry from coarse-scale matches will be used to filter the finer scale matches for the next round refinement. The Sampson error e with distortion consideration is applied to filter the mismatches:

$$e = \frac{\|(\mathcal{F}_\kappa^{-1}(\mathbf{p}_l))^T \mathbf{F} (\mathcal{F}_\kappa^{-1}(\mathbf{p}_r))\|^2}{\|(\mathcal{F}_\kappa^{-1}(\mathbf{p}_l))^T \mathbf{F}\|^2 + \|\mathbf{F}(\mathcal{F}_\kappa^{-1}(\mathbf{p}_r))\|^2}, \tag{4}$$

Where $\mathbf{F}$ is the fundamental matrix between two cameras.

$\{\mathbf{p}_l, \mathbf{p}_r\}$ are feature coordinates in the original distorted images, and $\{\mathcal{F}_{\kappa}^{-1}(\mathbf{p}_l), \mathcal{F}_{\kappa}^{-1}(\mathbf{p}_r)\}$ the corresponding feature coordinates in the undistorted images. The inverse distortion $\mathcal{F}_{\kappa}^{-1}(\cdot)$ is solved using method described in 4.2. Matches with Sampson error larger than the average scale $\bar{s_i}$ of the group will be filtered. Moreover, inside each round of refinement, we apply the BA twice. The first BA is carried out with the Huber Loss penalty as to tolerate mismatches that satisfy the epipolar geometry. The matches will then be filtered again with the new Sampson errors, and the second standard BA is applied with the newly filtered matches. The overall description of our RCR is given in Algorithm 1.

3.2. Multi-view Extension

For N-view ($N > 2$) stereo algorithms where refined cameras from RCR can not be directly applied, we extend our camera refinement to a general multi-view version for existing MVS methods. As accurate feature matches can be recovered using the RCR method, for each image pair in the cluster, we apply RCR to establish the reliable two-view feature matches. Multi-view matches are then reconstructed by merging two-views if the same feature appears in different image pairs. We then perform a standard local BA within the cluster. For the initialization of the BA, we initialize the camera parameters using global SfM cameras, and 3D point positions by triangulating the above two-view and multi-view matches.

The critical point to our local camera refinement is the sufficient fine-scale feature matches introduced in pairwise RCR, which have been usually overshadowed by the global larger-scale features in SfM reconstruction. Though mismatches might exists, they have already satisfied the epipolar line constrain in the RCR step, and would not affect the quality of camera geometry refinement. Compared to the formal local refinement method [48], our method consistently keeps a smaller reprojection error and guarantees the better refinement quality. It is also important to note that ours is an efficient algorithm, and the total running time of the camera refinement is neglectable to the dense reconstruction process itself.

4. Globally Consistent Dense Reconstruction

4.1. Local Dense Reconstruction

The refined cameras are first used to undistort images within the image pair/cluster. Then the MVS algorithms could be applied to generate the local dense reconstructions. We choose plane sweeping stereo [44] to generate dense point cloud in our pipeline. The purposes of choosing plane sweep stereo are twofold: (1) its two-view version is the simplest form stereo matching, whose dense results can directly reveal the local geometry quality; (2) plane sweeping is very suitable for real-time applications thanks to its easy parallelism, which is commonly used in SLAM based stereo reconstructions [34, 41, 22].

Inspired by [45, 6], we applied the weighted zero mean normalized cross correlation (weighted ZNCC) for matching cost computation. Let $\mathcal{N}_p$ and $\mathcal{N}_q$ be two patch windows centered at pixels $\mathbf{p}_c$ and $\mathbf{q}_c$. The matching cost between these two patches are defined as:

$$C = \frac{\sum_{\mathbf{p}\in\mathcal{N}_p,\mathbf{q}\in\mathcal{N}_q} w(\mathbf{p},\mathbf{p}_c)(I_p - \bar{I}_p)(I_q - \bar{I}_q)}{\sqrt{\sum_{\mathbf{p}\in\mathcal{N}_p}(I_p - \bar{I}_p)^2}\sqrt{\sum_{\mathbf{q}\in\mathcal{N}_q}(I_q - \bar{I}_q)^2}} \tag{5}$$

Where $\mathbf{p}$, $\mathbf{q}$ are the corresponding pixels in two windows, I_p, I_q the intensities of these two pixels, and $\bar{I}_q$, $\bar{I}_q$ the average intensities of windows $\mathcal{N}_p$ and $\mathcal{N}_q$. The weight $w(\mathbf{p},\mathbf{p_c}) = \exp(\frac{\|I_p - I_{p_c}\|}{\gamma})$ with a regularization factor $\gamma = 12$ is used to penalize pixels that differ a lot from the left center pixel $\mathbf{p}_c$, which could be seen as a soft segmentation of the matching windows. The final matching cost of between $\mathbf{p}_c$ and $\mathbf{q}_c$ is aggregated from patches from four different scales as recommended by the original paper [44]: $C_{pq} = \sum_{\sigma} C_{\sigma}$. To filter out some obvious outliers, a KD tree is built upon the point cloud. We calculate the mean distance l from one point to its k nearest neighborhoods where $k = 15$. Points with $l > 4\bar{l}$ will be filtered, and here $\bar{l}$ is the mean of l among all dense points. For SLAM based dense reconstruction, we also use images in the sliding key-frame window to validate and filter the point cloud as proposed in [41].

4.2. Global Dense Alignment

Point clouds from local camera geometries are mutually inconsistent. As the camera parameters has been altered during the refinement, aligning locally reconstructed point clouds is non-rigid. In our paper, we register points from different coordinate system back the one globally consistent camera system, which is reformulated into a matching registration problem and finally solved via inverse distortion.

Matching Registration We first review the process of how a 3D point can be produced in a given camera system: (1) finding the matching pixels that refer to the same 3D point in different images; (2) triangulating these pixels to generate a 3D point with the given camera parameters. An observation from this process is once the pixel matches in global camera system are determined, a globally consistent point can be reconstructed. So the dense alignment problem can be converted to the registration of pixel matches in the relative camera system to a global camera system.

Let us introduce three images $\{I_0, I_g, I_r\}$ that actually refer to the same image: (1) the original distorted image I_0; (2) the **global image** I_g, which is undistorted from

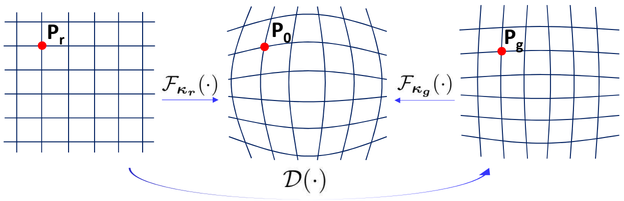

$\mathbf{P_r}$
$\mathbf{P_0}$
$\mathbf{P_g}$
$\mathcal{F}_{\boldsymbol{\kappa_r}}(\cdot)$
$\mathcal{F}_{\boldsymbol{\kappa_g}}(\cdot)$
$\mathcal{D}(\cdot)$

Figure 2. Illustration of the process of matching registration. From left to right: the **refined local image** I_r undistorted using the refined local cameras, the original distorted image I_0, and the **global image** I_g undistorted using the global cameras. The coordinates mappings are shown as the arrows.

I_0 using the global camera $\mathbf{C}_g$; (3) the **refined local image** I_r, which is undistorted from I_0 using the refined local camera $\mathbf{C}_r$. For a pixel $\mathbf{p}$, we define $\{\mathbf{p}_0, \mathbf{p}_g, \mathbf{p}_r\}$ as its corresponding pixel coordinates in these three images, and $\{\mathbf{u}_0, \mathbf{u}_g, \mathbf{u}_r\}$ the corresponding image coordinates. According to the distortion equation we have $\mathbf{u}_0 = \mathcal{F}_{\kappa_g}(\mathbf{u}_g)$, $\mathbf{u}_0 = \mathcal{F}_{\kappa_r}(\mathbf{u}_r)$. Combining the pixel-image coordinate relation that $\mathbf{u} = (\mathbf{p} - \mathbf{x}_c)/f$, the conversion between $\mathbf{p}_g$ and $\mathbf{p}_r$ can be expressed as:

$$\mathbf{p}_g = \mathcal{D}(\mathbf{p}_r) = f_g \cdot \mathcal{F}_{\kappa_g}^{-1}\big(\mathcal{F}_{\kappa_r}(\frac{\mathbf{p}_r - \mathbf{x}_{c_r}}{f_r})\big) + \mathbf{x}_{c_g} \tag{6}$$

Where κ, $\mathbf{x_c}$, f are camera distortion, camera center and focal length respectively. Figure 2 illustrate the distortion relationship within these three images.

Based on the matching registration, our dense alignment strategy can be summarized as: (1) using function $\mathcal{D}(\cdot)$ to convert pixel matches from refined local images to global images; (2) using the global cameras to triangulate the matches to get the corresponding 3D points. In this way, the dense results from the different local camera systems can be registered together in the global SfM camera system. An example of our global dense alignment is shown in Figure 3.

Inverse distortion The key of applying equation 6 is how to solve the inverse distortion $\mathcal{F}_{\kappa}^{-1}(\cdot)$. As there is no closed-form solution for $\mathcal{F}_{\kappa}^{-1}(\cdot)$, our method is to formulate it into an optimization problem with proper initialization. Given the distortion relation $\mathbf{u}_0 = \mathcal{F}_{\kappa}(\mathbf{u}_g)$, we define the minimization problem as:

$$\arg\min_{\mathbf{u}_g} \|\mathbf{u}_0 - \mathcal{F}_{\kappa_g}(\mathbf{u}_g)\|^2 \tag{7}$$

This non-linear least square problem could have multiple local minimums as the distortion equation is a 7-order polynomial. In this case, the initialization of $\mathbf{u}_g$ is very important to the final solution. Luckily the distortion of the local refined camera is closed to that of the global camera, so the image coordinate $\mathbf{u}_r$ in the relative image I_r can be used to initialize $\mathbf{u}_g$. With this good initialization, the minimization is able to quickly converge to the correct local minimum. In all our experiments, the optimization converges within 3 iterations using Levenberg-Marquardt algorithm, which makes our dense alignment an efficient process.

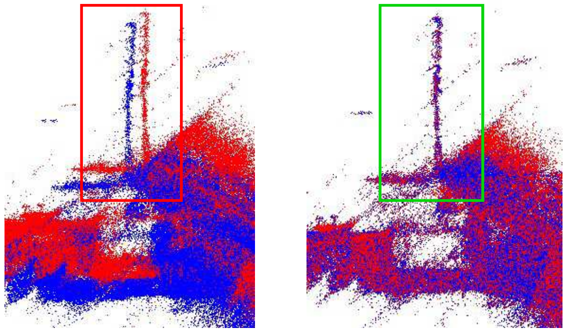


Figure 3. The point clouds from two stereo pairs before and after dense alignment. Left: the red and the blue point clouds are inconsistent before the alignment. Right: two point clouds are well aligned after employing the proposed method.

5. Experiment

We test our method on both SfM based dense reconstruction and the SLAM based dense reconstruction. For SfM based reconstruction, we use the open source SfM software openMVG [28] to recover the global camera geometry for all real-world datasets. For SLAM based reconstruction, we apply the visual-inertial odometry provided by the Apple ARKit [1] to get the camera parameters. Our bundle adjustment is implemented using Ceres Solver [3] and the inverse distortion is solved by levmar [25]. The image pairs used in our reconstructions are selected silimar to method [18]. To better illustrate the dense reconstruction quality, we apply the mesh reconstruction algorithm [43] to reconstruct the mesh surface. The proposed method is implemented and evaluated on a PC with 8-Core Intel i7-4770K processor and 32GB memory. A single NVIDIA GTX980Ti graphic card is used to accelerate the dense and mesh reconstructions.

5.1. SfM Based Dense Reconstruction

Synthetic Datasets. Firstly, we demonstrate the importance of accurate camera geometry for dense reconstruction using synthetic data. Given the ground truth cameras of *fountain-P11* and *Herz-Jesu-P8* datasets [40], we slightly perturb the camera parameters to simulate inaccurate SfM results. We use the term *noise level* to measure the inaccuracy degree. Let $\boldsymbol{\beta}$ be Eular angles of the camera rotation, $\boldsymbol{X_c}$ the camera center, $\bar{\boldsymbol{X}}_c$ the average center of all cameras, f the focal length and $\boldsymbol{\kappa}$ the camera distortion. At noise level i ($i = 0, 1, \ldots, 10$), camera parameters are perturbed

dataset	#images	resolution	#pairs	reprojection error		#matches / pair		#points / pair		running time / pair		
				GC	RCR	GC	RCR	GC	RCR	RCR	plane sweeping	alignment
fountain-P11	11	6.3M	10	-	0.121	-	3809	0.187M	0.459M	4.32s	1.56s	1.37s
Herz-jesu-P8	8	6.3M	7	-	0.132	-	3629	0.159M	0.383M	3.41s	1.13s	1.28s
House	34	12M	35	0.770	0.239	150	504	0.204M	0.226M	1.20s	2.31s	0.518s
Factory	103	12M	103	0.904	0.181	514	2074	0.426M	0.508M	2.43s	1.39s	1.91s
City A	48285	24M	43521	1.527	0.343	123	1483	0.831M	0.941M	3.24s	31.7s	3.63s
City B	98113	50.3M	91581	2.153	0.398	105	3205	2.41M	2.85M	3.58s	19.4s	9.27s

Table 1. Statistics of all experiment datasets. The cameras of fountain-P11 and Herz-jesu-P8 are perturbed from ground truth cameras at noise level 8. GC represents global cameras, and RCR represents the relative cameras from our RCR methods. GC method and RCR method are compared on the number of features used for camera estimation, reprojection errors after BA, average dense point numbers and algorithm running time.

as: $\boldsymbol{\beta} = \boldsymbol{\beta} + i \cdot [\frac{1}{2}\delta, \frac{1}{2}\delta, \frac{1}{2}\delta]^T$, $\boldsymbol{X_c} = \boldsymbol{X_c} + i \cdot \frac{1}{2}\delta(\boldsymbol{X_c} - \bar{\boldsymbol{X}}_c)$, $f = f \cdot (1 + i \cdot \delta)$ and $\boldsymbol{\kappa} = \boldsymbol{\kappa} \cdot (1 + i \cdot \delta)$, where δ is a small random number that uniformly distributed in range $[-10^{-3}, 10^{-3}]$. At each noise level, we reconstruct the dense point clouds 10 times with independently perturbed cameras. To quantify the point cloud accuracy, we employ the Hausdorff distance [4] to measure the difference between the dense point cloud and the given ground truth mesh, and the average Hausdorff distance among all 10 reconstructions is regarded as the dense reconstruction error. Also, the average point number at each noise level is chosen as the indicator for the local dense reconstruction quality.

The quantitative comparison is demonstrated in Figure 4. As SfM inaccuracy (noise level) increases, the number of dense points dramatically decreases if the perturbed cameras are used. However, after applying our RCR, the number of reconstructed dense points remains unchanged in whichever noise level. This indicates that our refinement consistently produces the locally precise geometry for the local dense reconstruction. Nevertheless, as the noise level goes up, the inaccurate global camera system serves as a less precise camera system for dense alignment, which inevitably results in higher Hausdorff error. Overall, our RCR consistently keeps a lower Hausdorff error comparing to the standard approach without RCR.

Real-world Datasets. We test our RCR method in four real-world datasets, namely, *House*, *Factory*, *City A* and *City B* datasets. *House* and *Factory* datasets contain 34 and 103 images at 4K resolution respectively. One difficulty of reconstructing these two scenes is to recover the repetitive structures in building roofs. Features detected in repetitive areas are vulnerable to mismatch thus accurate local camera geometries are hard to achieve, which will again affect the dense reconstruction. In the selected image pair with global SfM cameras, the epipolar lines deviate about 2 pixels from the matching points. However, with our RCR, epipolar lines become precise for all points, and as a result, the dense point cloud is clean and complete even in the highly repetitive region. Detailed results on dense and mesh reconstructions of these two datasets can also be found in Figure 8.

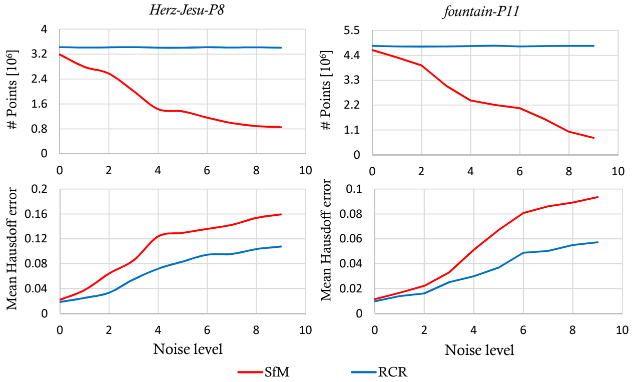


Figure 4. Quantitative evaluation on point number and Hausdorff distance changes with increasing SfM noise level.

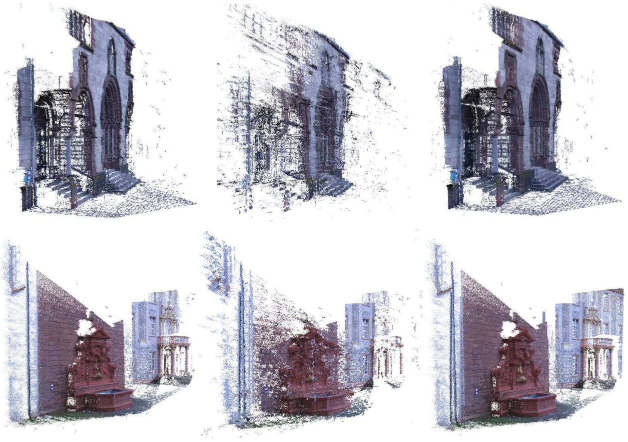


Figure 5. Comparison on dense reconstructions using RCR cameras and global cameras. The left result are reconstructed using the ground truth cameras, while the middle one is using the perturbed cameras at noise level 8, and the right one is reconstructed using the perturbed cameras with our RCR method.

City A and *City B* are two very large datasets that contain 48285 images at 6000 x 4000 resolution and 98113 images at 8688 x 5792 resolution respectively. As we have mentioned, in large scale SfM reconstructions, features with larger scales are preferred. Also, other compromised approaches like feature resampleing [13] will also affect the SfM quality. Starting from a small region in each dataset,

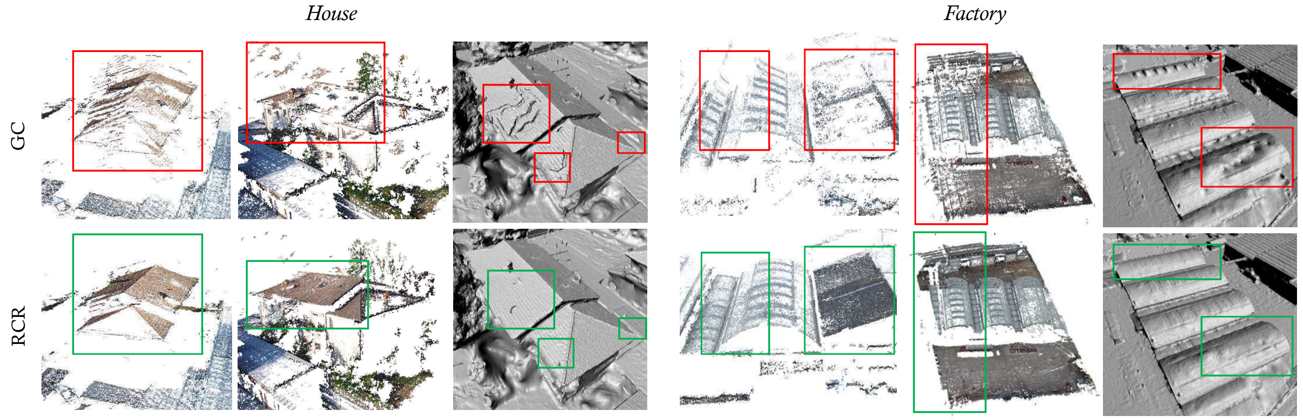


Figure 8. Dense and mesh reconstructions of *House* and *Factory* datasets with plane sweeping stereo. From top to bottom are reconstructions using SfM and our RCR method. Our method significantly improves the dense and mesh reconstructions for these two dataset. The orange box shows the misalignment problem remains in paper [48].

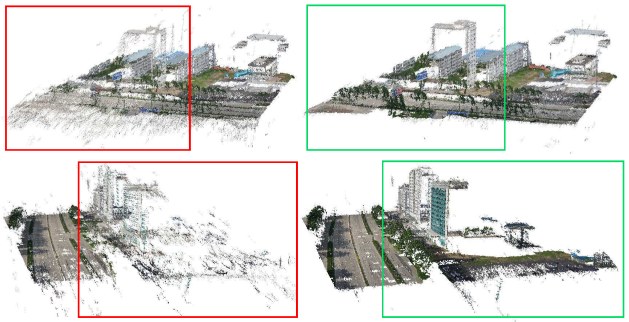


Figure 6. Dense reconstructions of *City A* dataset. The comparison between SfM method (left) and ours (right) clearly demonstrates the effectiveness of our RCR method.

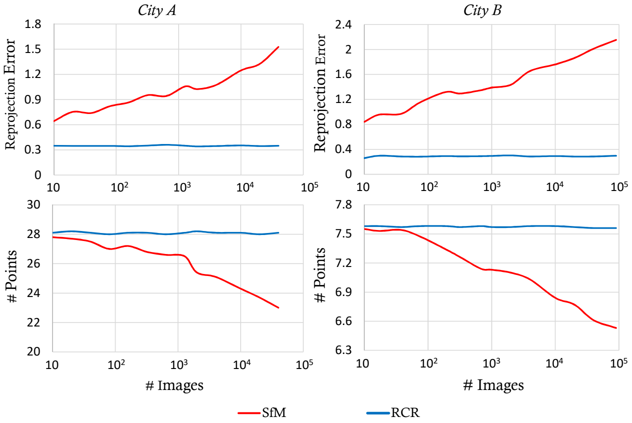


Figure 7. Reprojection errors of cameras and reconstructed point numbers with increasingly larger subset of images using SfM (red) and our RCR method (blue).

we perform the SfM algorithm to reconstruct the camera geometries with progressively larger subset of images. For SfM at different scales, we compare the dense reconstructions using SfM cameras with using our RCR method. Figure 7 shows that the reprojection errors are getting larger and the reconstructed points are becoming lesser within the selected regions if the SfM scale goes up, indicating that it is more difficult to ensure the local geometry quality in large scale SfM reconstruction. With our RCR method, the local reprojection error is kept to a stable small value. The dense reconstruction results of *CityA* from SfM with all images are shown in Figure 9, where we can see roads and buildings are severely deteriorated. In contrast, with our RCR, the reconstructed roads and buildings become complete and clean. More results on epipolar lines and dense reconstructions of the two cities can be found in the supplementary material. And statistics of the reconstructions of all datasets can be found in Table 1.

	#clusters	#images /cluster	reprojection error			Running time/cluster			
			SfM	LR	Ours	LR	Ours	PMVS	Aligment
House	4	9	0.77	0.613	0.315	3.87	6.87	134	1.89
Factory	10	10	0.904	0.851	0.328	5.64	28.5	305	5.13
Campus	164	9	1.32	0.939	0.408	6.21	24.6	239	7.31

Table 2. Statistics of experiment on the multi-view extension. Our local refinement keeps a lower reprojection error compared to the SfM and LR methods.

Multi-view evaluation We evaluate the multi-view extension of our method using the patch-based multi-view stereo (PMVS [16]). The cameras are divided into small groups by the camera clustering algorithm CMVS [14]. We compared the proposed method with the SfM method as well as the local refinement method (LR method [48]), which uses a simple local BA to refine the cameras and ignores the dense alignment step. Table 2 shows the reconstruction statistics of *House*, *Factory* datasets, where we can find that our multi-view extension keeps a lower reprojection error than the SfM and LR methods. Visual comparisons on the dense and mesh results can be found in Figure 9, which clearly shows the importance of both the local camera refinement and the global dense alignment steps.

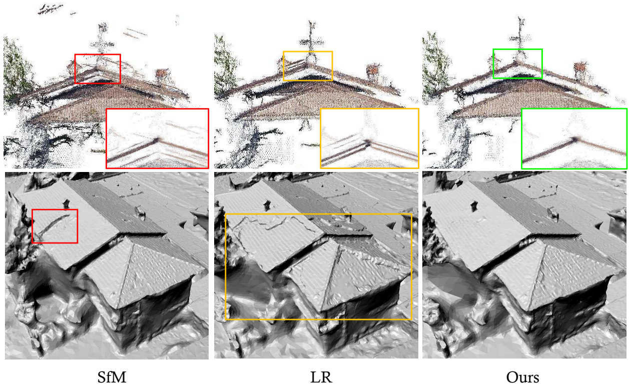


Figure 9. PMVS results using three different methods. The orange areas clearly shows the importance of global dense alignment.

5.2. SLAM Based Dense Reconstruction

We also test our method on the SLAM based dense reconstruction. The ARKit [1] is a new commercial IMU-SLAM system that runs on IOS equipment. As we have mentioned, the fusion of the IMU and visual measurements increase the stability and smoothness of camera trajectory, however it decreases the accuracy of camera poses, which will cause trouble to the stereo matching. We export camera poses and video frames from ARKit and follow the dense algorithm of method [41] to reconstruct the point cloud. Images are resized to 640 x 360 resolution and we perform plane sweeping for the new key-frame to generate the depth map. The past key-frames in the sliding window are used to validate each pixel of the newly generated depth map.

To mimic the real-time performance of SLAM system, we detect the ORB feature proposed in [30] rather than the SIFT feature for camera refinement. Considering the camera intrinsic and the distortion in the SLAM system are known beforehand, our pairwise BA can be carried out by also fixing the camera intrinsic and the distortion. Also, we perform the pairwise BA with all features at once rather than the progressive refinement as mentioned in section 3.1. In the dense alignment step, because the intrinsic and the distortion are fixed, the matching registration step can be skipped ($\mathbf{p}_g = \mathbf{p}_r$), and we only need to triangulate the local pixel matches using the global SLAM cameras. The camera refinement and dense alignment in the SLAM system are much more efficient compared to the proposed one for SfM system, and will not affect the real-time performance of the SLAM system.

The reconstruction results are shown in Figure 10. We can see the direct reconstructions with SLAM cameras are with very low quality (totally $190,884$ points). With our camera refinement and dense alignment, the number of reconstructed points is dramatically increased to $749,119$. We also show the reconstruction result with camera refinement only, and we can obviously observe the dense stratification in the dense reconstruction.

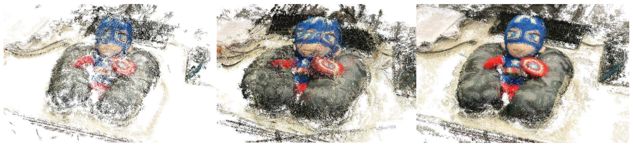


Figure 10. SLAM based dense reconstructions. Left: reconstruction from direct SLAM cameras; Middle: reconstruction with only camera refinement. Right: the proposed method with both camera refinement and dense alignment.

5.3. Discussion

Running Time. Table 1 and 2 report the running time of each step in the reconstruction pipeline. For the two-view method, the running time of our RCR and alignment are at the same level with the GPU-based plane sweeping stereo. For the multi-view extension, although performing the RCR for every image pair within the cluster takes a longer time than the simple LR method, the total running time of the additional refinement and alignment is still an order of magnitude faster than the dense reconstruction algorithm itself, and is negligible to the whole reconstruction process. For the SLAM based reconstruction, the refinement can be carried out extremely fast as mentioned in section 5.2, and can be run in real-time for the key-frame based SLAM system.

Limitations. The dense alignment merges local dense reconstructions to the global camera system, which is based on the assumption that the SfM or SLAM still maintains a globally consistent camera system. One potential problem is that, if the SfM or SLAM result is of extremely low quality, point cloud stratification will still occur after the dense alignment. However, in our experiments, both OpenMVG and Apple ARKit are able to provide qualified global cameras for the dense alignment.

6. Conclusion

We have presented a dense reconstruction framework that collaborates the local fine cameras geometry and the global cameras geometry for accurate dense reconstructions. Two novel components, the local camera refinement and the global dense alignment, have been introduced as the additional steps to the traditional dense reconstruction pipeline. Intensive experiments on both SfM and SLAM based stereo reconstructions have demonstrated that our method is able to significantly improve the dense reconstruction quality.

Acknowledgement. This work is supported by Hong Kong RGC 16208614, T22-603/15N, Hong Kong ITC PSKL12EG02, and China 973 program, 2012CB316300

References

[1] Augmented reality for ios. https://developer.apple.com/arkit.

[2] S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. M. Seitz, and R. Szeliski. Building rome in a day. *Communications of the ACM*, 54:105–112, 2011.

[3] S. Agarwal, K. Mierle, and Others. Ceres solver. http://ceres-solver.org.

[4] N. Aspert, D. Santa Cruz, and T. Ebrahimi. Mesh: measuring errors between surfaces using the hausdorff distance. In *ICME (1)*, pages 705–708, 2002.

[5] C. Bailer, M. Finckh, and H. P. Lensch. Scale robust multi view stereo. In *European Conference on Computer Vision*, pages 398–411. Springer, 2012.

[6] M. Bleyer, C. Rhemann, and C. Rother. Patchmatch stereo-stereo matching with slanted support windows. In *Bmvc*, volume 11, pages 1–11, 2011.

[7] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart. Robust visual inertial odometry using a direct ekf-based approach. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 298–304. IEEE, 2015.

[8] A. Brahmachari and S. Sarkar. View clustering of wide-baseline n-views for photo tourism. In *2011 24th SIBGRAPI Conference on Graphics, Patterns and Images*, pages 157–164. IEEE, 2011.

[9] D. Crandall, A. Owens, N. Snavely, and D. Huttenlocher. Discrete-continuous optimization for large-scale structure from motion. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3001–3008. IEEE, 2011.

[10] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. Monoslam: Real-time single camera slam. *IEEE transactions on pattern analysis and machine intelligence*, 29(6):1052–1067, 2007.

[11] A. Delaunoy and M. Pollefeys. Photometric bundle adjustment for dense multi-view 3d modeling. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1486–1493. IEEE, 2014.

[12] J. Engel, T. Schöps, and D. Cremers. Lsd-slam: Large-scale direct monocular slam. In *European Conference on Computer Vision*, pages 834–849. Springer, 2014.

[13] T. Fang and L. Quan. Resampling structure from motion. In *European Conference on Computer Vision*, pages 1–14. Springer, 2010.

[14] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski. Towards internet-scale multi-view stereo. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1434–1441. IEEE, 2010.

[15] Y. Furukawa and J. Ponce. Accurate camera calibration from multi-view stereo and bundle adjustment. *International Journal of Computer Vision*, 84(3):257–268, 2009.

[16] Y. Furukawa and J. Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE transactions on pattern analysis and machine intelligence*, 32(8):1362–1376, 2010.

[17] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3354–3361. IEEE, 2012.

[18] M. Goesele, N. Snavely, B. Curless, H. Hoppe, and S. M. Seitz. Multi-view stereo for community photo collections. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE, 2007.

[19] J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis. Consistency analysis and improvement of vision-aided inertial navigation. *IEEE Transactions on Robotics*, 30(1):158–176, 2014.

[20] V. Indelman, S. Williams, M. Kaess, and F. Dellaert. Information fusion in navigation systems via factor graph based incremental smoothing. *Robotics and Autonomous Systems*, 61(8):721–738, 2013.

[21] G. Klein and D. Murray. Parallel tracking and mapping for small ar workspaces. In *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, pages 225–234. IEEE, 2007.

[22] K. Kolev, P. Tanskanen, P. Speciale, and M. Pollefeys. Turning mobile phones into 3d scanners. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3946–3953, 2014.

[23] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale. Keyframe-based visual–inertial odometry using nonlinear optimization. *The International Journal of Robotics Research*, 34(3):314–334, 2015.

[24] M. Lhuillier and L. Quan. A quasi-dense approach to surface reconstruction from uncalibrated images. *IEEE transactions on pattern analysis and machine intelligence*, 27(3):418–433, 2005.

[25] M. Lourakis. levmar: Levenberg-marquardt nonlinear least squares algorithms in C/C++. http://www.ics.forth.gr/~lourakis/levmar, Jul. 2004. [Accessed on 31 Jan. 2005.].

[26] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[27] P. Moulon, P. Monasse, and R. Marlet. Global fusion of relative motions for robust, accurate and scalable structure from motion. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3248–3255, 2013.

[28] P. Moulon, P. Monasse, R. Marlet, and Others. Openmvg. an open multiple view geometry library. https://github.com/openMVG/openMVG.

[29] A. I. Mourikis and S. I. Roumeliotis. A multi-state constraint kalman filter for vision-aided inertial navigation. In *Robotics and automation, 2007 IEEE international conference on*, pages 3565–3572. IEEE, 2007.

[30] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.

[31] R. Mur-Artal and J. D. Tardós. Visual-inertial monocular slam with map reuse. *IEEE Robotics and Automation Letters*, 2(2):796–803, 2017.

[32] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison. Dtam: Dense tracking and mapping in real-time. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2320–2327. IEEE, 2011.

[33] K. Ni, D. Steedly, and F. Dellaert. Out-of-core bundle adjustment for large-scale 3d reconstruction. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE, 2007.

[34] M. Pollefeys, D. Nistér, J.-M. Frahm, A. Akbarzadeh, P. Mordohai, B. Clipp, C. Engels, D. Gallup, S.-J. Kim, P. Merrell, et al. Detailed real-time urban 3d reconstruction from video. *International Journal of Computer Vision*, 78(2-3):143–167, 2008.

[35] J. L. Schonberger and J.-M. Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4104–4113, 2016.

[36] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 519–528. IEEE, 2006.

[37] S. Shen, N. Michael, and V. Kumar. Tightly-coupled monocular visual-inertial fusion for autonomous flight of rotorcraft mavs. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 5303–5310. IEEE, 2015.

[38] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3d. In *ACM transactions on graphics (TOG)*, volume 25, pages 835–846. ACM, 2006.

[39] D. Steedly, I. Essa, and F. Dellaert. Spectral partitioning for structure from motion. In *Proceedings of the 2003 9th IEEE International Conference on Computer Vision (ICCV)*, volume 2, pages 996–1003, 2003.

[40] C. Strecha, W. von Hansen, L. Van Gool, P. Fua, and U. Thoennessen. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. Ieee, 2008.

[41] P. Tanskanen, K. Kolev, L. Meier, F. Camposeco, O. Saurer, and M. Pollefeys. Live metric 3d reconstruction on mobile phones. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 65–72, 2013.

[42] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon. Bundle adjustmenta modern synthesis. In *International workshop on vision algorithms*, pages 298–372. Springer, 1999.

[43] H.-H. Vu, P. Labatut, J.-P. Pons, and R. Keriven. High accuracy and visibility-consistent dense multiview stereo. *IEEE transactions on pattern analysis and machine intelligence*, 34(5):889–901, 2012.

[44] R. Yang and M. Pollefeys. Multi-resolution real-time stereo on commodity graphics hardware. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 1, pages I–211. IEEE, 2003.

[45] K.-J. Yoon and I. S. Kweon. Adaptive support-weight approach for correspondence search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):650–656, 2006.

[46] R. Zhang, T. Fang, S. Zhu, and L. Quan. Multi-scale tetrahedral fusion of a similarity reconstruction and noisy positional measurements. In *Asian Conference on Computer Vision*, pages 30–44. Springer, 2014.

[47] R. Zhang, S. Li, T. Fang, S. Zhu, and L. Quan. Joint camera clustering and surface segmentation for large-scale multi-view stereo. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2084–2092, 2015.

[48] S. Zhu, T. Fang, J. Xiao, and L. Quan. Local readjustment for high-resolution 3d reconstruction. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3938–3945. IEEE, 2014.