Real time complete dense depth reconstruction for a monocular camera

Xiaoshui Huang[1], Lixin Fan[2], Jian Zhang[1], Qiang Wu[1] and Chun Yuan[3]
[1]University of Technology, Sydney. Sydney,Australia
[2]Nokia Technologies. Tampere, Finland
[3]Graduate school of Shenzhen, Tsinghua University. Shenzhen, China

Abstract

In this paper, we aim to solve the problem of estimating complete dense depth maps from a monocular moving camera. By 'complete', we mean depth information is estimated for every pixel and detailed reconstruction is achieved. Although this problem has previously been attempted, the accuracy of complete dense depth reconstruction is a remaining problem. We propose a novel system which produces accurate complete dense depth map. The new system consists of two subsystems running in separated threads, namely, dense mapping and sparse patch-based tracking. For dense mapping, a new projection error computation method is proposed to enhance the gradient component in estimated depth maps. For tracking, a new sparse patch-based tracking method estimates camera pose by minimizing a normalized error term. The experiments demonstrate that the proposed method obtains improved performance in terms of completeness and accuracy compared to three state-of-the-art dense reconstruction methods VSFM+CMVC, LSD-SLAM and REMODE.

1. Introduction

Complete dense depth reconstruction aims at obtaining the depth of every pixel in the image (e.g. Figure 1) and restoring as much detailed information as possible of the 3D scene. Due to their dense property, high quality depth maps are much needed in many applications, including surface reconstruction and 3D data acquisition. This is an important research topic which has great impacts on multimedia, computer vision and computer graphic applications.

The existing depth reconstruction methods [4, 10, 2, 3, 9, 11, 12, 5, 8, 7] can be categorized from three aspects: capturing camera, completeness and matching approach.

In relation to the capturing camera, the existing depth reconstruction methods can be divided into two types: *stereo* camera and *monocular* camera. Stereo camera methods [7] use two calibrated cameras to capture stereo images in sequence and reconstruct depth. While high quality depth estimations have been demonstrated for this type of approaches, the requirement of stereo cameras prevents it from being used in many real life applications. Monocular camera methods [4, 10, 2, 3, 9, 11] use a single camera to estimate depth for which the mapping thread is used to compute depth and the tracking thread is used to compute camera pose; when conducting mapping, camera pose from tracking step is utilized; and when conducting tracking, the depth map estimated from the last mapping step is utilized. The depth estimation at each pixel is highly parallel and can be implemented in real-time on GPUs. The proposed method belongs to the monocular camera methods.

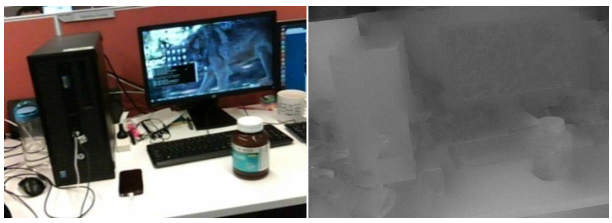


Figure 1. The proposed method builds a complete dense depth map for a monocular camera. Our method combines the robustness and accuracy of dense mapping with efficient sparse patch tracking techniques. Left: image from video. Right: dense depth result.

In relation to the completeness of recovered depth maps, the existing methods can be divided into three types: *sparse, semi-dense* and *complete dense*. Sparse methods usually reconstruct from feature-based methods such as VSFM+CMVS [12][5]. The drawbacks are several: they are time-consuming and use limited image information. Typical examples of semi-dense methods are the works in [2][3], which first reconstruct complete dense maps by stereo matching and then remove depth for low-certainty pixels. Although they can build depth in real-time, only semi-dense depth can be produced (e.g. depth is only produced at the edge regions). Complete dense methods use tracking and mapping to accurately estimate the depth of all image pixels. These methods retain more details than the other methods, which is very helpful for many applications (3D matching and recognition). DTAM [10] and REMODE

[11] are typical examples of complete dense methods, however, the quality of recovered depth map in [10, 11] is tradeoff with completeness and very low when the depth map is complete. Also, they are sensitive to noise depth from initialization or mapping thread. The novel method proposed in this paper demonstrates how to improve the accuracy and the robustness of complete dense depth estimation.

In relation to the matching approach, the existing methods can be divided into three types: *feature point matching* [12, 5], *stereo matching* [2, 3, 8] and *multi-view stereo* [10, 11]. Feature point matching methods find the matched points and estimate depth for sparse keypoints only. The matching stage is often time-consuming since computation for naive near-neighbor searching grows exponentially with the number of keypoints and KD-Tree type of indexing methods do not scale up with high dimensional descriptors such as SIFT. Stereo matching methods restrict the searching along epipolar line between two images with known camera poses. Multi-view matching methods improve depth map by accumulating error terms measured for every (overlapped) pixels in multiple images. While the computational cost seems high, GPUs are often exploited to achieve real-time performances. The method proposed in this paper belongs to the multi-view matching approach.

In this paper, a novel complete dense depth reconstruction method (CD-SLAM) is proposed for monocular camera which improves completeness, accuracy and robustness of depth estimation. Unlike previous dense depth reconstruction methods [11] using the probability method to remove inaccurate depth estimation, we improve the depth accuracy of every pixel by integrating the gradient component into the project error computation, thus making the matching of pixels more discriminative without sacrificing the fast computation speed of original methods. To our best knowledge, this is the first work aiming at improving depth accuracy from the original multi-view matching point computation. Apart from that, due to the normalized projection error in tracking, the proposed method is robust to the depth noise from mapping, hence, it also improves the accuracy from tracking step.

In summary, our contributions are three-fold. Firstly, a new gradient enhanced projection error for mapping optimization is proposed, which leads to improved depth reconstruction accuracy. Secondly, a normalized sparse patch tracking method is proposed, which is robust to the depth noise from mapping. Thirdly, a new complete dense depth map reconstruction pipeline is proposed, which computes complete depth map robustly and accurately.

2. Method

The overall structure of the proposed method consists of two subsystems running in separated threads, namely, dense mapping and sparse patch based tracking(Figure 2).

Our method is partially motivated by recent developments in real-time SLAM technologies [10, 2]. In order to build a highly accurate complete depth map, gradient is added into photometric error computation, which helps to retain structural information in the depth map (e.g. edges). According to [4], sparse patch information in the image is sufficient to get a rough estimate of motion and find the correspondence relations. It achieves high efficiency and accurate camera pose. However, the performance is highly dependent on initialized depth accuracy and sensitive to noise depth. In order to remove the influence of noise depth from the mapping thread, we propose a new sparse patch-based method by considering the depth from the mapping thread as a weight for tracking.

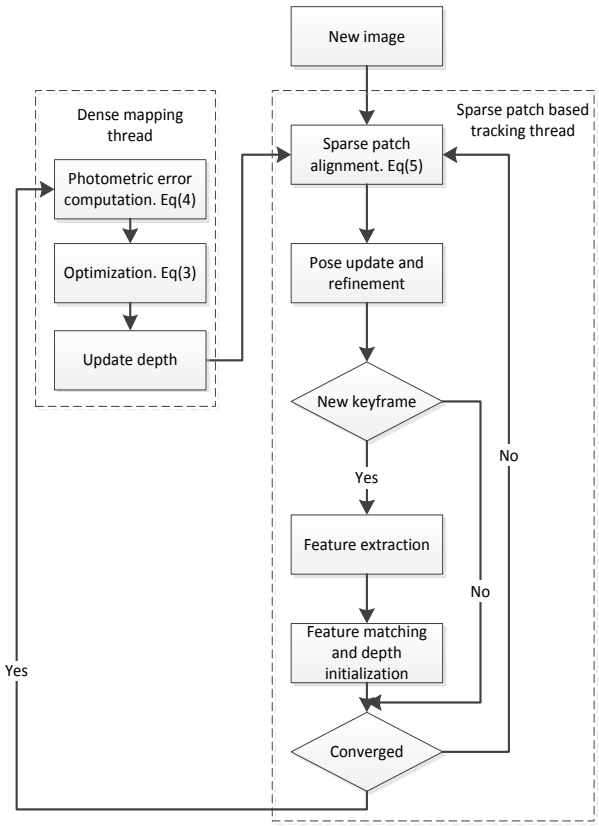


Figure 2. Outline of the proposed complete dense depth reconstruction method.

2.1. Preliminaries.

To obtain projection points of one image in another image, we first review transformations and projections used in this paper.

2.1.1 Basic projection formulations.

Assuming there is a 3D point $\mathbf{M}_w$ in world coordinate frame, it can be transformed to 3D point $\mathbf{M}_c$ in its corre-

sponded camera-centered world coordinate frame by following formulation:

$$\mathbf{M}_c = P_{cw}\mathbf{M}_w \tag{1}$$

where P_{cw} is a 4×4 matrix containing a rotation and a translation component. Note that both $\mathbf{M}_w$ and $\mathbf{M}_c$ are homogeneous coordinates of the form $[x, y, z, 1]^T$.

The camera-centered point $\mathbf{M}_c$ can be projected into image pixels coordinates by a projection function: $\mathbf{m} = \pi(\mathbf{M}_c)$. In contrast, for a pixel $\mathbf{m}$, if the inverse depth value d_u is known, the 3D point $\mathbf{M}$ can be recovered by inverse projection function π^{-1}: $\mathbf{M} = \pi^{-1}(\mathbf{m}, d_u)$.

2.1.2 Photometric error computation.

In the context of multi-view stereo reconstruction, the following project error is minimized to obtain depth estimation at each pixel:

$$C = \Sigma_{p \in D} \| I_p - I_q \| \tag{2}$$

where I_p is the projection of I_q in its visual views. All three RGB channels are used. D is whole image pixel domain that are overlapped in multiple views.

A regularized Huber norm regulariser over gradient of inverse depth map can be introduced to penalize non-smooth surface, hence imposing smoothness constraint [10]:

$$min_{\varepsilon} \Sigma_{u \in D} \{ g(u) \| \bigtriangledown \varepsilon(u) \|_{\epsilon} + \lambda C + \frac{1}{\theta}(\varepsilon(u) - d(u)) \} \tag{3}$$

where $d(u)$ is the depth computed by initial photometric error computation, and $\varepsilon(u)$ is the optimal depth we want to search. The optimization is reached by replacing the weighted Huber regulariser with its conjugate using the Legendre-Fenchel transform [10, 1].

The photometric error reviewed above only uses pixel value, hence, it will be very sensitive to changes in the light (e.g. light changing in different views). As a remedy to ambiguous depth estimation incurred by lighting changes, many existing methods remove uncertain depth.

2.2. Dense mapping

2.2.1 Photometric error enhanced by image gradient.

Unlike the previous projection error computation method, we add gradient into the error computation.

$$C = \Sigma_{p \in D} \{ w \| I_p - I_q \| + (1 - w) \| G_p - G_q \| \} \tag{4}$$

where I_p is the projection of I_q in its visual view, G_p or G_q is the gradient of the corresponding points and w is the relative weight which is set to 0.75. This setting is found to be robust for all experiments illustrated in Section 4.

The constant intensity assumption made in photometric error term (Eq. 2) is often violated in practice, in contrast, the gradient enhanced error term (Eq. 4) is more robust to global lighting changes for edge regions and its advantages is indeed demonstrated in our experimental results. Figure 3 shows that object structures are better preserved near edges with gradient enhance photometric error term.

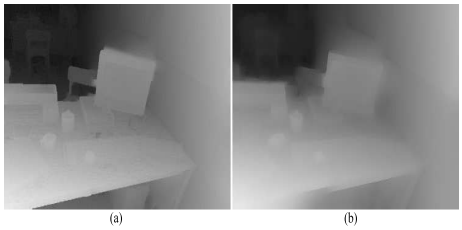


Figure 3. Photometric error computation results (a) with and (b) without gradient.

2.3. Sparse patch based tracking

As demonstrated in [1], it is sufficient to estimate the camera pose of one view by using sparse patches only. We adopt the sparse-patch based tracking in our work. Our camera pose estimation comprises two steps, including keypoint search and pose update. We firstly compute FAST keypoints and construct a 4x4 patch around the positions of the keypoints. Secondly, we project the patches onto their corresponding patches to find the camera pose that minimizes the photometric error of all the patches. As there is a lot of noise in projection matching, we normalize the projection error. This will ensure improved accuracy in camera pose estimation. The projection photometric error of all patches is defined as follows:

$$E_{P_{j,k}} = min_{P_{j,k}} \frac{1}{2} \Sigma_{u_i \in R} \| \frac{\delta I(P_{j,k}, u_i)}{\sigma^2_{\delta I}} \|^2_2 \tag{5}$$

where u_i is the pixel coordinate, R is the image region for which depth is known at j frame and for which the back-projection points are visible in the current image k. To compute the optimal camera pose, we parameterize an incremental update $P_{i,k}(\varepsilon)$ as a twist $\varepsilon \in se(3)$. The details of the update steps are as follows:

$$\delta I(P_{j,k}, u_i) = I_j(p) - I_k(\pi(P_{j,k}(\varepsilon) \cdot U_i)) \tag{6}$$

where U_i is the 3D point of u_i. $U_i = \pi^{-1}(u_i, d_{ui})$. For the weight term, we assume Gaussian image intensity noise σ^2_I, and the incremental weight are related to depth variance V_j. Following [2], the updating term is computed as

$$\sigma^2_{\delta I} = 2\sigma^2_I + (\frac{\partial(\delta I(P_{j,k}, u_i))}{\partial(d_{ui})})^2 V_j(u) \tag{7}$$

To compute the optimal update step $P_{j,k}(\varepsilon)$, we solve it in a similar way to [1][2]. The optimal $P_{j,k}(\varepsilon)$ is found when the solving process convergences.

For pose refinement, we follow the work of [4], which improves the correspondence accuracy into the subpixel level to obtain a better camera pose.

3. Implementation details

The algorithm is bootstrapped to calculate the camera pose of the first two key frames and the initial depth map. We perform FAST keypoint matching for the first two frames and compute the fundamental matrix. The initial depth map is reconstructed using triangulation of these two views.

The system uses a GPU with two threads. When a new image enters, the camera pose is first estimated using the previous depth estimated from the dense mapping or its own depth initialization. Then, we use the dense mapping thread to conduct a complete dense depth map reconstruction.

In the tracking thread, we select the keyframe during the tracking process. As we use a small baseline video, we only compute the keypoints in the keyframe for efficiency. A keyframe is selected when the average change of scene depth of a new frame exceeds 10% relative to previous keyframe. After the keyframe is updated, the farthest frames compared to the new keyframe are removed.

In the mapping thread, an exhaustive search on whole inverse depth range is time-consuming. To address this issue, we add depth-filter to each reference keyframe. When a pixel has a known photometric error, we compute its d_{max} and d_{min}. In the next process of photometric error computation, we search in $[d_{min}, d_{max}]$ instead of searching in the whole inverse depth range.

4. Experimental results

We evaluate the proposed method (CD-SLAM) using the same computer on which LSD-SLAM [2] was running successfully. In all our experiments, we use a computer with I5, 8G memory and NVidia GTX 970 GPU. We compare the completeness and accuracy of our method with LSD-SLAM, which is a state-of-the-art method in monocular camera depth reconstruction. Also, we compare the accuracy of our method with the accuracy of REMODE [11]. In addition, we also compare our method with VSFM+CMVS[12][5], which is a widely used successful system.

For completeness, we use the depth cover percentage at each depth image, which is the rate of the number of depth known pixel (N_{depth}) with respect to all image pixels (N_{total}).

$$R_{completeness} = \frac{N_{depth}}{N_{total}} \tag{8}$$

For accuracy, we compare the estimated depth maps with respect to the ground truth and report performance of different methods. The *Table* dataset, which is used in our work, was presented in [6]. This dataset is constructed by ray-tracing software and the related ground truth depth maps are available.

Output results from different methods are pre-processed so that they can be compared on the same ground. For VSFM+CMVS, we use camera poses computed by SFM to project the 3D points back to the original images. For LSD-SLAM, we edit the code and output the completeness and accuracy of depth map.

4.1. Quantitative comparison in Table dataset

The completeness and accuracy of our method are compared with LSD-SLAM with the Table dataset, using the column entitled *Perfect Ray-Traced Images: Fast Motion*. The sequence has characters of 80 frames per second and ground truth depth after parsing the data. We use 100 images from the sequences to conduct this experiment.

Completeness. Figure 4 illustrates the comparison of the completeness of VSFM+CMVS, SLD-SLAM and CD-SLAM, which demonstrates that CD-SLAM outperforms other methods in terms of completeness. With more completeness depth map, we have more details on the reconstruction depth.

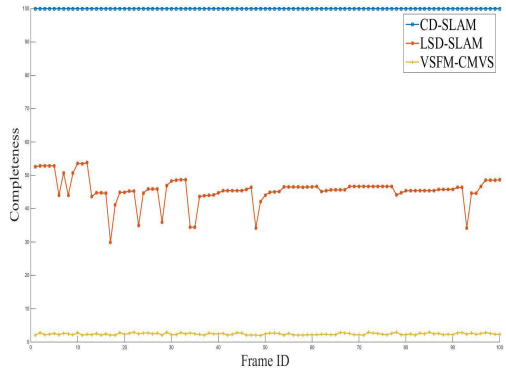


Figure 4. Qualitative comparison of completeness of SFM+CMVS, LSD-SLAM and the proposed method.

An example depth map result for the same original image is shown in Figure 5, which showcases the completeness of depth maps generated by the proposed method.

Threshold accuracy. As the original points of SFM+CMVS are too sparse when projected back into the images, we only compare the accuracy of our method with LSD-SLAM as shown in Figure 6. Accuracy is defined as the percentage of pixels for which the estimated depth error is less than a predefined threshold from the ground truth depth. In our experiments, we set the threshold to be 20, which amounts to about 7.8% of the entire depth range i.e. 256.

It was shown in Figure 6 that the proposed method obtains higher accuracy than LSD-SLAM at most frames with

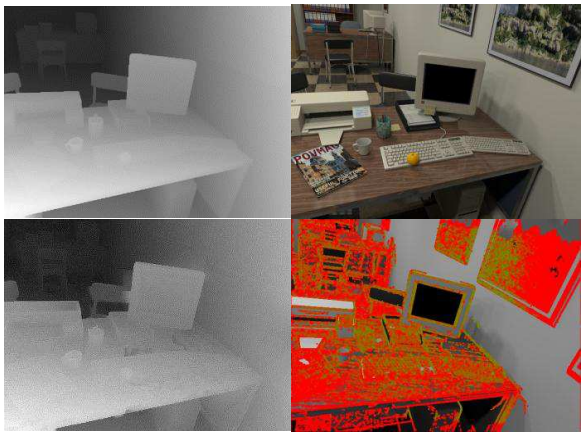


Figure 5. The first row is ground truth(left) and original image(right); the second row is the result of CD-SLAM(left) and LSD-SLAM(right). LSD-SLAM only has depth on yellow and red regions.

mean margin about 10%. LSD-SLAM uses stereo matching to estimate depth, and unfortunately leads to ambiguity matching points even though in edge region. CD-SLAM uses projection minimization and information on more views than LSD-SLAM. Also, our new optimization function integrates image intensity gradient of each point which contributes to higher accuracy by eliminating the ambiguous matcher caused by changes in lighting.

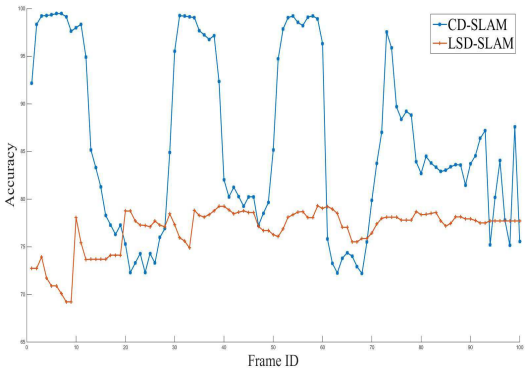


Figure 6. Qualitative comparison of accuracy of LSD-SLAM and CD-SLAM.

Mean accuracy on real depth. We compute the mean accuracy for LSD-SLAM, REMODE and CD-SLAM. Because the depth results are inverse depth, to obtain real depth and know how much difference of the computed depth to ground truth in real life, we compute the inverse of the three methods' depth maps. For LSD-SLAM, only semi-dense depth is taken part in mean computation. As REMODE [6] is the method most similar to the proposed method, complete depth is used for mean computation. The results, shown in Table 1, indicate that our method has a 0.14m mean error, which is comparable with LSD-SLAM and higher than REMODE. However, regarding the depth completeness, our method is complete while LSD-SLAM is semi-dense. In order to obtain comparable accuracy to our method (e.g. 0.1m), REMODE only has about 46% completeness. It is because the proposed method improves accuracy from the original depth computation, which integrates gradient element to improve the matching accuracy at different projection views. However, REMODE only uses the probability method to remove unreliable depth.

Table 1. Mean accuracy comparison results of LSD-SLAM, REMODE and CD-SLAM.

Accuracy	LSD-SLAM	REMODE	CD-SLAM
Mean	0.105	0.35m	0.14m

4.2. Qualitative evaluation with gradient and without gradient.

We evaluate our method with and without gradient. We use the ground truth dataset to test the performance. The threshold is 20 in intensity difference.

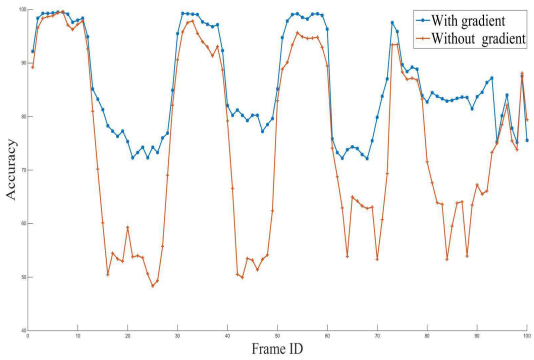
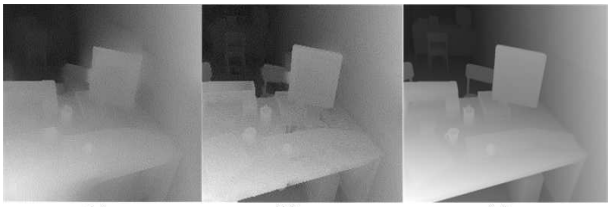


Figure 7. Qualitative evaluation with gradient and without gradient.

Figure 7 demonstrates that the accuracy of the method with gradient outperforms the method without gradient. Figure 8 shows the example data. As can be seen, our method obtains better depth results, particularly in the edge region. Instead of using a regularized term to smooth the depth, our new photometric minimization function computes a better depth result at the original photometric error minimization. Hence, we can obtain highly accurate depth results.

4.3. Runtime

The runtime of the proposed method is a trade-off with accuracy. If we want to pursue highest possible speed, we can gain about 5ms per frame. On the other hand, the results



(a) (b) (c)

Figure 8. Comparison of depth reconstruction result with and without gradient term. (a) is without gradient considering, (b) is with gradient considering, (c) is ground truth depth map.

presented above are obtained at 160ms per frame, which is still a reasonable speed for many real-time applications.

5. Conclusion

A novel complete dense reconstruction method is proposed in this paper. It reconstructs the accurate depth of every pixel in the image. The proposed method introduces a novel projection error computation method which improves the accuracy from the original multi-view stereo. Also, we introduce a normalized error term in pose estimation which is robust enough to deal with noise from the mapping step. The experiments demonstrate that our new pipeline obtains better performance in completeness and accuracy than other state-of-the-art dense depth reconstruction methods.

References

[1] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145, 2011.

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

[3] J. Engel, J. Sturm, and D. Cremers. Semi-dense visual odometry for a monocular camera. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1449–1456. IEEE, 2013.

[4] C. Forster, M. Pizzoli, and D. Scaramuzza. Svo: Fast semi-direct monocular visual odometry. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 15–22. IEEE, 2014.

[5] Y. Furukawa and J. Ponce. Accurate, dense, and robust multiview stereopsis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(8):1362–1376, 2010.

[6] A. Handa, R. A. Newcombe, A. Angeli, and A. J. Davison. Real-time camera tracking: When is high frame-rate best? In *Computer Vision–ECCV 2012*, pages 222–235. Springer, 2012.

[7] S. Heinzle, P. Greisen, D. Gallup, C. Chen, D. Saner, A. Smolic, A. Burg, W. Matusik, and M. Gross. Computational stereo camera system with programmable control loop. In *ACM Transactions on Graphics (TOG)*, volume 30, page 94. ACM, 2011.

[8] X. Huang, C. Yuan, and J. Zhang. Graph cuts stereo matching based on patch-match and ground control points constraint. In *Advances in Multimedia Information Processing–PCM 2015*, pages 14–23. Springer, 2015.

[9] 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.

[10] 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.

[11] M. Pizzoli, C. Forster, and D. Scaramuzza. Remode: Probabilistic, monocular dense reconstruction in real time. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 2609–2616. IEEE, 2014.

[12] C. Wu. Towards linear-time incremental structure from motion. In *3D Vision-3DV 2013, 2013 International Conference on*, pages 127–134. IEEE, 2013.