OUTLIER AND ARTIFACT REMOVAL FILTERS FOR MULTI-VIEW STEREO

Kouya Yodokawa[†], *Koichi Ito*[†], *Takafumi Aoki*[†], *Shuji Sakai*[‡], *Takafumi Watanabe*[‡] *and Tomohito Masuda*[‡]

[†]Graduate School of Information Sciences, Tohoku University, Japan.
[‡] Toppan Printing Co., Ltd., Japan
E-mail: yodokawa@aoki.ecei.tohoku.ac.jp

ABSTRACT

This paper proposes an outlier and artifact removal method for multi-view stereo. The proposed method introduces the three filters, which check (i) consistency among depth maps and their visibility, (ii) left-right consistency and (iii) consistency between the depth map and color intensity, respectively. The proposed method removes outliers and artifacts from depth maps generated by PatchMatch Multi-View Stereo. We demonstrate that the proposed method exhibits the efficient performance on 3D reconstruction compared with conventional methods through a set of experiments using public datasets and under practical situations.

Index Terms— 3D reconstruction, depth map, outlier removal, artifact removal, multi-view stereo

1. INTRODUCTION

Multi-view 3D reconstruction is a technique to reconstruct the 3D shape of a target object from multiple images and has been a topic of interests in the field of computer vision for many years [1, 2]. Multi-view 3D reconstruction generates the 3D shape of a target object through camera parameter estimation by Structure-from-Motion (SfM) and 3D mesh reconstruction by Multi-View Stereo (MVS).

MVS algorithms are categorized into four types: voxel-based, mesh-based, depth-map-based, and patch-based algorithms [1]. Among them, depth-map-based MVS and patch-based MVS can be used for a wide range of applications, since they do not require the initial 3D shape or do not constrain the 3D shape of a target object (e.g., topology). For this reason, many state-of-the-art multi-view 3D reconstruction algorithms employ depth-map-based MVS or patch-based MVS [3, 4, 5, 6, 7, 8, 9, 10]. However, both depth-map-based MVS and patch-based MVS have a drawback, since depth-map-based MVS usually generates artifacts at the object boundaries, patch-based MVS cannot reconstruct 3D shape from low-contrast regions, and both reconstruction results include much outliers with accurate 3D points.

Addressing the above problem, this paper proposes an outlier and artifact removal method for MVS. The proposed method employs PatchMatch Multi-View Stereo (PM-MVS) [11], which is one of the fast depth map generation methods, to reconstruct the depth map of each view from multi-view images. Three types of filters are applied to depth maps so as to remove outliers and artifacts based on multiple-view geometry. The three filters check (i) consistency among depth maps and their visibility, (ii) left-right consistency and (iii) consistency between the depth map and color intensity, respectively. We demonstrate that the proposed method exhibits the efficient performance on 3D reconstruction compared with conventional methods through a set of experiments.

2. PATCHMATCH MULTI-VIEW STEREO (PM-MVS)

This section briefly describes PM-MVS [11], which is used to reconstruct the depth map of each view from multi-view images in this paper. Note that the proposed filtering techniques described in the next section can be used in other MVS algorithms.

PM-MVS generates a depth map from multi-view images by repeating spatial propagation, view propagation and plane refinement. In spatial propagation, a depth is propagated to neighboring pixels and is updated according to Normalized Cross-Correlation (NCC)-based scores. In view propagation, a depth is propagated to neighboring views and is updated according to NCC-based scores. In plane refinement, a depth is refined by comparing the matching scores calculated from the current parameters and the parameters with the addition of minute disturbance.

The accuracy of depth maps generated by PM-MVS is comparable with plane-sweeping approaches, while the computational cost of PM-MVS is much lower than that of plane-sweeping approaches. PM-MVS requires a few dozen-time window matching to calculate the depth of one 3D point, while a simple plane-sweeping approach requires more than a thousand-time window matching to calculate the depth of one 3D point. Refer to [11] for more details of PM-MVS.

3. THREE FILTERING TECHNIQUES

A 3D point cloud is obtained from multiple depth maps and their camera parameters. 3D points of each view are calculated from its depth map and are converted into the world coordinate system so as to merge them into a 3D point cloud. Then, three filtering techniques are applied to remove outliers and artifacts. The three filters check (i) consistency among depth maps and their visibility, (ii) left-right consistency and (iii) consistency of pixel intensity, respectively.

We now consider a set of views $\boldsymbol{V} = \{V_1, V_2, \cdots, V_N\}$. N is the number of views and $\boldsymbol{m} = (u, v)$ is an image coordinate. For each view $V_n \in \boldsymbol{V}$, let $I_n(\boldsymbol{m})$ be an image, $\boldsymbol{A}_n$ be an internal parameter, and $\boldsymbol{R}_n$ and $\boldsymbol{t}_n$ be external parameters consisting of a rotation matrix and a translation vector. The depth map is indicated by $d_n(\boldsymbol{m})$. For each reference view $V_i \in \boldsymbol{V}$, let $\boldsymbol{V}_k \subseteq \boldsymbol{V} \backslash \{V_i\}$ be one of other views. Using the above notation, we describe the detail of the proposed three filters in the following.

3.1. Filter 1: Consistency among Depth Maps and Their Visibility

Filter 1 checks consistency among the multiple depth maps and their visibility. If a 3D point interrupts the visibility of other 3D points or its visibility is interrupted by other 3D points, this point is removed as an outlier.

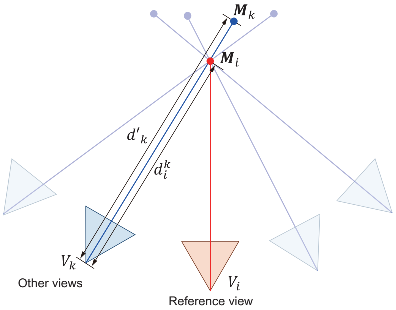

Fig. 1. Geometric relation between the reference view and the other view.

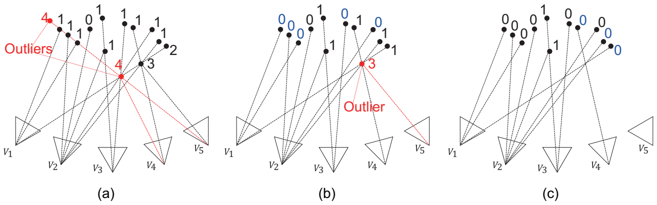

Fig. 2. Example of outlier removal using Filter 1. The red color indicates outliers and the blue color indicates the updated penalty scores.

Let consider the reference view V_i and the other view V_k as shown in Fig. 1. For V_i, the 3D point $\boldsymbol{M}_i = [X_i, Y_i, Z_i]^T$ is calculated from the coordinate $\boldsymbol{m}_i$ on the depth map $d_i(\boldsymbol{m}_i)$ by

$$\boldsymbol{M}_i = \boldsymbol{R}_i^{-1} \left\{ d_i(\boldsymbol{m}_i) \boldsymbol{A}_i^{-1} \tilde{\boldsymbol{m}}_i - \boldsymbol{t}_i \right\}, \quad (1)$$

where $\tilde{\boldsymbol{m}}_i$ indicates the homogeneous coordinate of $\boldsymbol{m}_i$. The corresponding point on the neighboring view V_k is obtained by

$$s\tilde{\boldsymbol{m}}_i^k = \boldsymbol{A}_k \left[\boldsymbol{R}_k | \boldsymbol{t}_k \right] \tilde{\boldsymbol{M}}_i, \quad (2)$$

where s is a scale factor. The depth d_i^k on V_k is calculated from the 3D point $\boldsymbol{M}_i$ as follows

$$d_i^k = R_k^{31} X_i + R_k^{32} Y_i + R_k^{33} Z_i + t_k^3, \quad (3)$$

where R_k^{st} and t^s indicate the (s,t)-th element of $\boldsymbol{R}_k$ and the s-th element of $\boldsymbol{t}_k$, respectively. The depth d'_k is obtained from the depth map of V_k as follows

$$d'_k = d_k(\boldsymbol{m}_i^k). \quad (4)$$

If $(d'_k - d_i^k) > \delta_1$, $\boldsymbol{M}_i$ or $\boldsymbol{M}_k$ can be considered as an outlier, while it cannot be distinguished using the relation between two views. Hence, we introduce a penalty score for $\boldsymbol{m}_i$ and classify $\boldsymbol{m}_i$ into a 3D point or an outlier by majority vote. The penalty scores of $\boldsymbol{m}_i$ and $\boldsymbol{m}_i^k$ are incremented by 1 in the case that $(d'_k - d_i^k) > \delta_1$. The penalty scores of $\boldsymbol{m}_i$ and $\boldsymbol{m}_i^k$ are evaluated for all the other views. After calculating the penalty scores for all the pixels on all the depth maps, we obtain the penalty score map as shown in Fig. 2 (a). $\boldsymbol{m}_i$ having the maximum penalty score can be considered as an outlier and then is removed. In the case of Fig. 2 (a), 3D points with penalty score=4 are removed as outliers. Then, the penalty scores are updated as shown in Fig. 2 (b), since the penalty scores are changed by removing outliers. The above process is repeated until the penalty score of all the remaining 3D points is not greater than 2 as shown in Fig. 2 (c). The threshold δ_1 is set to

$$\delta_1 = 15 \times \frac{d'_k}{f_k} \quad (5)$$

in the experiment, where f_k indicates the focal length, which can be obtained from $\boldsymbol{A}_k$. This threshold means that a 3D point having the depth difference more than 15 pixels is a target to carry penalty.

3.2. Filter 2: Left-Right Consistency

Filter 2 is similar to left-right consistency checking used in binocular stereo matching. We remove a point whose distance from each corresponding point in all other views is longer than threshold, where we use the depth instead of the distance. Therefore, we use the depth $d_i(\boldsymbol{m}_i)$ for the reference view V_i and the depth d'_k for the neighboring view V_k as well as Filter 1. If $|d_i(\boldsymbol{m}_i) - d_k^i| < \delta_2$, we increment the score by 1. If the score is less than 2 after calculating the score for all other views, $\boldsymbol{m}_i$ can be considered as an outlier, since its left-right consistency is collapsed from the viewpoint of stereo matching. The threshold δ_2 is set to

$$\delta_2 = 5 \times \frac{d'_k}{f_k} \quad (6)$$

in the experiments. This threshold means that M_i having the depth difference less than 5 pixels corresponds to M_k.

3.3. Filter 3: Consistency of Pixel Intensity

Filter 3 checks the consistency of pixel intensity among the multiple images to remove artifacts observed around the surface. Let us consider the coordinate $\boldsymbol{m}_i$ on the reference view V_i and its corresponding coordinate $\boldsymbol{m}_i^k$ on other view V_k as well as Filter 1. We do not take care of a 3D point near other points in Filter 1, since it is hard to check the consistency of such a 3D point using only geometric relation. The use of pixel intensity makes it possible to classify such a 3D point into a true 3D point or an outlier. Therefore, if $(d'_k - d_i^k) > 0$, the penalty score of $\boldsymbol{m}_i$ is evaluated by a pixel intensity. The pixel intensity of $\boldsymbol{m}_i$ is given by $I_i(\boldsymbol{m}_i)$ for V_i and $I_k(\boldsymbol{m}_i^k)$ for V_k, respectively, where I_i and I_k indicate the images for each view. If $||I_i(\boldsymbol{m}_i) - I_k(\boldsymbol{m}_i^k)||_2 > \delta_3$, $\boldsymbol{m}_i$ can be considered as an outlier. The penalty scores of $\boldsymbol{m}_i$ and $\boldsymbol{m}_i^k$ are incremented by 1 in this case. The penalty score is evaluated for all the 3D points and all the views as well as Filter 1. A 3D point having the largest penalty score is removed and the score of remaining 3D points is updated. The above process is repeated until the penalty score of all the remaining 3D points is not greater than 2. Note that we design Filter 3 for the purpose of evaluating the effectiveness of artifact removal, although Filter 3 can be combined into Filter 1. The threshold δ_3 is set to 50 in the experiments. This threshold means that the pixel intensity between $\boldsymbol{m}_i$ and $\boldsymbol{m}_i^k$ is different if the Euclidean distance in the RGB-color space is more than 50, where we assumes that each color channel is 8 bits.

4. EXPERIMENTS AND DISCUSSION

This section describes experiments for evaluating the effectiveness of the proposed method.

4.1. Evaluation of the Combination of Filters

The first experiment evaluates the combination of the three filters. We use "Cat" and "Dog" datasets available in ToHoku University Multi-View Stereo (THU-MVS) Datasets[1]. "Cat" and "Dog" are the

[1] http://www.aoki.ecei.tohoku.ac.jp/mvs/

Table 1. Experimental result for evaluating the combination of filters, where RMS errors [mm] are indicated.

Combination	No filter	1	2	3	1, 2	1, 3	2, 1	2, 3
Cat	0.4873	0.3943	0.4158	0.4518	0.3780	0.3982	0.3809	0.4034
Dog	0.4842	0.3595	0.4406	0.4758	0.3470	0.3688	0.3568	0.4411
Combination	3, 1	3, 2	1, 2, 3	1, 3, 2	2, 1, 3	2, 3, 1	3, 1, 2	3, 2, 1
Cat	0.4180	0.3992	0.3863	0.3889	0.3900	0.3924	0.3882	0.3933
Dog	0.3759	0.3989	0.3545	0.3548	0.3656	0.3725	0.3570	0.3600

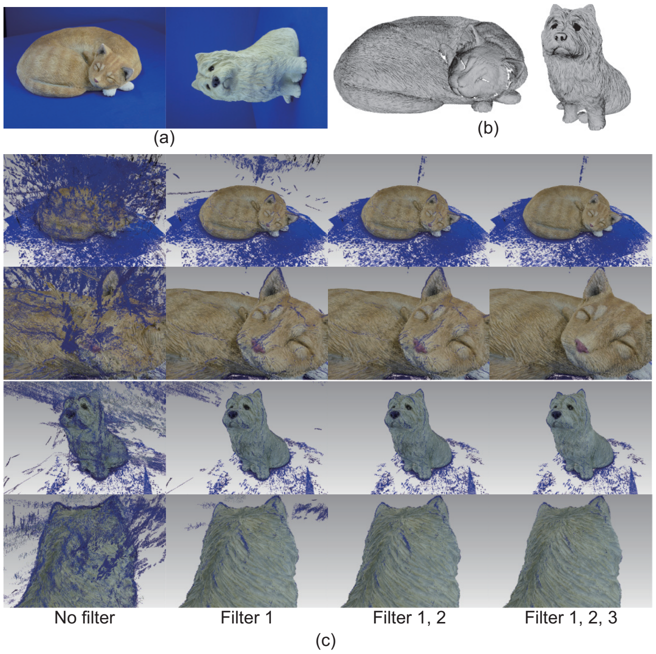

Fig. 3. Experimental results using THU-MVS datasets: (a) images, (b) ground-truth mesh models and (c) reconstructed 3D point clouds for each method.

multi-view image datasets of figurines of a cat and a dog, respectively. We use the RGB-color version of "Cat" and "Dog" to evaluate the effectiveness of Filter 3. The images are captured by a camera (Panasonic Lumix GF6) with $2,272 \times 1,704$ pixels. Cat and Dog datasets consist of 108 and 72 images, respectively, and their ground truth mesh model. 3D point clouds are reconstructed from 36 images in this experiment, where 36 images are selected at intervals of 3 images for Cat and 2 images for Dog. Figure 3 (a) shows the example of an image in the datasets. The ground truth mesh models as shown in Fig. 3 (b) are measured with the 3D digitizing system (Steinbichler, COMET5). The accuracy is evaluated by comparing the reconstructed 3D point clouds and the ground-truth mesh model using the iterative closest point algorithm [12].

Camera parameters are estimated by the sequential SfM pipeline [13]. Correspondence among multi-view images is obtained by Scale-Invariant Feature Transform (SIFT) [14]. Then, camera parameters are calculated using the correspondence and are optimized so as to minimize reprojection error among corresponding points. We use an open-source SfM library, OpenMVG [15] in this paper.

Fig. 3 (c) shows reconstructed 3D point clouds for some combinations. Table 1 shows a summary of Root Mean Square (RMS) errors for all the combinations of the three filters. Filter 1 and filter 2 remove outliers and filter 3 removes artifacts around the object surface as shown in Fig. 3 (c). The combination of Filter 1 and 2 is the best in all the combinations in terms of RMS errors as shown in Table 1. On the other hand, the quality of the 3D point cloud when using the combination of Filter 1, 2, and 3 is higher than that of Filter 1 and 2, since blue-colored points, i.e., artifacts from backgrounds, are removed in Filter 1, 2 and 3, while such points are observed in Filter 1 and 2. In addition, the combination of Filter 1, 2, and 3 has a comparable RMS error with that of Filter 1 and 2. Therefore, the combination of filter 1, 2 and 3 is better than other combinations from the viewpoint of qualitative and quantitative evaluation. We employ the combination of Filter 1, 2 and 3 in the following experiments as the proposed method.

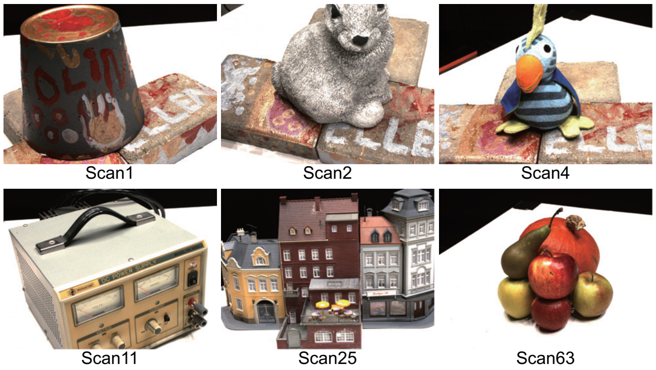

Fig. 4. Images from Jensen's dataset used in the experiment.

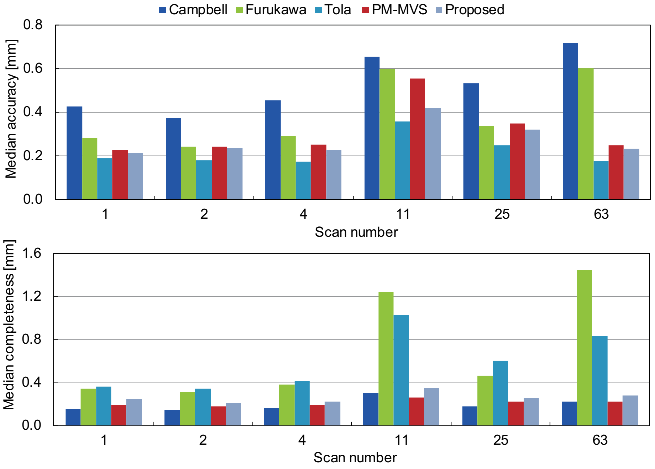

Fig. 5. Experimental results for comparing the proposed method with conventional methods.

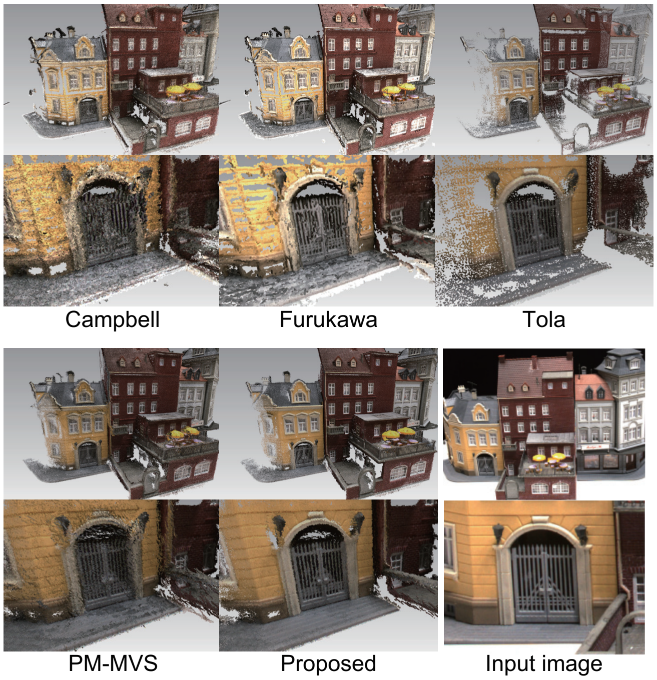

Fig. 6. Reconstruction result of Scan 25 for each method.

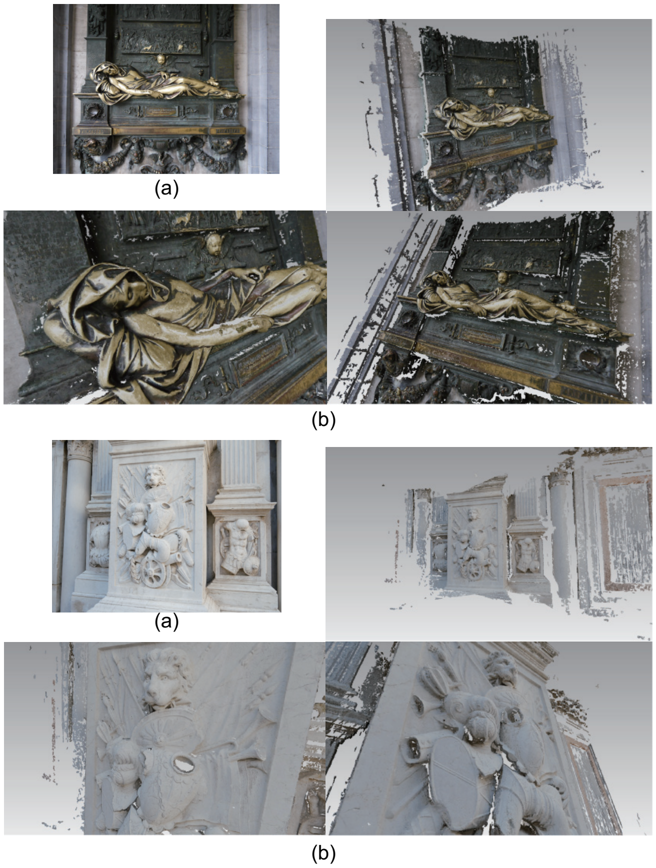

Fig. 7. Reconstruction results using the proposed method under practical situations: (a) input image and (b) reconstructed 3D point clouds.

4.2. Comparison with Conventional Methods

We compare the performance of the proposed method with that of the MVS methods using the public MVS datasets[2] provided Jensen et al. [16]. This dataset consists of a set of MVS images for 128 objects and their camera parameters. The image size is $1,600 \times 1,200$ pixels for all the images. Each object is taken by a camera from 49 or 64 constant viewpoints. For the purpose of performance evaluation, this dataset includes ground-truth 3D point clouds for all the objects. This dataset also includes 3D point clouds for each object reconstructed by the state-of-the-art MVS methods such as Campbell et al. [3], Furukawa et al. [4] and Tola et al. [5]. From this dataset, we selected 6 objects taken from 49 viewpoints. The performance is evaluated by accuracy and completeness which are used in [16]. Accuracy means a degree of correctness of the position of reconstructed points and is calculated as the distance from reconstructed points to the ground truth. Completeness means a degree of coverage of reconstructed points and is calculated as the distance from the ground truth to reconstructed points.

Fig. 5 shows the median of accuracy and median of completeness for each method and Fig. 6 shows an example of reconstructed 3D point clouds of Scan 25 for each method. The proposed method produces fewer outliers than Campbell's and Furukawa's methods and PM-MVS, since outliers can be removed by the three filters. The proposed method reconstructs dense 3D point clouds compared with Furukawa's and Tola's methods, since the median of completeness for the proposed method is lower than both methods. Thus, the proposed method exhibits better accuracy and better completeness among the methods. As observed in Fig. 6, the proposed method can reconstruct the 3D point cloud with fewer outliers and fewer artifacts compared with conventional methods.

4.3. Practical Situation

We reconstruct 3D point clouds from multi-view images taken under practical situations to demonstrate the potential possibilities of the proposed method. Fig. 7 shows some examples of reconstruction results using the proposed method. The upper is the Everard t'Serclaes monument in Brussels and is reconstructed from 11 images. The lower is a carving of Palazzo Ducale in Venice and is reconstructed from 8 images. As observed in Fig. 7 (b), the high-quality 3D point clouds are reconstructed by the proposed method.

5. CONCLUSION

This paper proposed an outlier and artifact removal method for multi-view stereo. We considered the three types of filters, which check (i) consistency among depth maps and their visibility, (ii) left-right consistency and (iii) consistency between the depth map and color intensity, respectively. We demonstrated that the proposed method exhibited efficient performance on 3D reconstruction compared with conventional methods through a set of experiments using public datasets and under practical situations. We will develop a simple and accurate 3D reconstruction system using the proposed method in future work.

6. REFERENCES

[1] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, "A comparison and evaluation of multi-views

[2] http://roboimagedata.compute.dtu.dk/?page_id=36

stereo reconstruction algorithms," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 519–528, 2006.

[2] C. Strecha, W. von Hansen, L. V. Gool, P. Fua, and U. Thoennessen, "On benchmarking camera calibration and multi-view stereo for high resolution imagery," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 1–8, 2008.

[3] N. D. F. Campbell, G. Vogiatzis, C. Hernandez, and R. Cipolla, "Using multiple hypotheses to improve depth-maps for multi-view stereo," *Proc. European Conf. Computer Vision*, pp. 766–779, 2008.

[4] Y. Furukawa and J. Ponce, "Accurate, dense, and robust multiview stereopsis," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 32, no. 8, pp. 1362–1376, 2010.

[5] E. Tola, C. Strecha, and P. Fua, "Efficient large-scale multi-view stereo for ultra high-resolution image sets," *Machine Vision and Applications*, vol. 23, no. 5, pp. 908–920, 2012.

[6] S. Shen, "Accurate multiple view 3D reconstruction using patch-based stereo for large-scale scenes," *IEEE Trans. Image Processing*, vol. 22, no. 5, pp. 1901–1914, 2013.

[7] S. Fuhrmann, F. Langguth, and M. Goesele, "MVE – A multi-view reconstruction environment," *Proc. Eurographics Workshop on Graphics and Cultural Heritage*, pp. 11–18, 2014.

[8] Y. Uh, Y. Matsuhita, and H. Byun, "Efficient multiview stereo by random-search and propagation," *Proc. Int'l Conf. 3D Vision*, pp. 393–400, 2014.

[9] S. Galliani, K. Lasinger, and K. Schindler, "Massibely parallel multiview stereopsis by surface normal diffusion," *Proc. Int'l Conf. Computer Vision*, pp. 873–881, 2015.

[10] A. Locher, M. Perdoch, and L.V. Gool, "Progressive prioritized multi-view stereo," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 3244–3252, 2016.

[11] M. Hiradate, K. Ito, T. Aoki, T. Watanabe, and H. Unten, "An extension of PatchMatch Stereo for 3d reconstruction from multi-view images," *Proc. Asian Conf. Pattern Recognition*, pp. 061–065, 2015.

[12] Z. Zhang, "Iterative point matching for registration of free-form curves and surfaces," *Int'l J. Computer Vision*, vol. 13, no. 2, pp. 119–152, 1994.

[13] P. Moulon, P. Monasse, and R. Marlet, "Adaptive structure from motion with a contrario model estimation," *Proc. Asian Conf. Computer Vision*, pp. 257–270, 2012.

[14] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int'l J. Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[15] P. Moulon, P. Monasse, R. Marlet, and Others, "OpenMVG: An Open Multiple View Geometry library," `https://github.com/openMVG/openMVG`.

[16] R. Jensen, A. Dahl, G. Vogiatzis, E. Tola, and H. Anæs, "Large scale multi-view stereopsis evaluation," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 406–413, 2014.