GPU based multi-scale depth map calculation for 3D reconstruction

Tiansheng Wu, Hui Wang, Yanling Wang, Min Liang, Jie Li*

School of information, Shanxi University of Finance and Economics, Taiyuan, China

*Corresponding author’s e-mail: lijiescu@aliyun.com

Abstract. When using multi-view stereo (MVS) method to calculate depth map for reconstructing dense 3D model, the model precision is often influenced by low-texture areas. To solve this problem, in this paper, we present an efficient multi-scale depth map based MVS method. First, the coarsest scale depth map is calculated which can easily capture the rich texture information in low-texture areas. Then we upsample this depth map to higher scale, and four forward-backward reprojections are used to reduce the error of depth difference between the last scale and the current scale. In addition, a GPU and CPU based cooperative optimization architecture is built to optimize and accelerate depth map computation in PatchMatch (PM) stage. Experimental results show that, the proposed method can reconstruct quite accurate and dense point clouds compared to state-of-the-art methods.

1. INTRODUCTION

MVS has been a hot research in 3D reconstruction for decades which is a key method used for generating dense 3D model, i.e., reconstructing 3D dense model for object and scene from multiple different views which is calibrated by structure-from-motion (SFM) system [1,2]. In the past few years, much great efforts have been put into improving the quality of dense 3D reconstructions, and some works have achieved remarkable results. However, performing efficient and accurate multi-view stereo with large data, repetitive patterns, low texture, occlusion problems is still a challenging problem.

Recently, many effective methods have been presented to solve above problems especially PM stereo which is a powerful tool to compute depth map. There are many methods based on this method having an impressive result whatever in accuracy and efficiency. For example, Xu and Tao [4] change the propagation scheme and propose a new framework that great reconstructs low-textured areas which is called multi-scale geometric consistency guidance (ACMM). Xu et al. [5] use differential geometry property of image surface to choose a different matching window size for every pixel, it is different from the most methods that use fixed matching window size. Shen [6] proposes a new depth-map merging based multiple view stereo method for large-scale scenes. Wu et al. [7] present a method that improves the completeness of 3D reconstruction. However, there are still many problems in MVS due to large-scale datasets, untextured regions and computational efficiency, and so on.

To tackle these problems, in this paper, we present a multi-scale depth map computation method that for large-scale datasets and low-texture areas. MVS can be divided into four different classes [3]. Our method is based on depth maps merging method in that it adapts to large-scale scenes with high resolution and can be easily parallelized at image level. In terms of large-scale scenes, a new framework that uses CPU multi-threaded and GPU accelerate the speed of depth map computation is presented. For low-texture regions, a multi-scale depth map merging method is used by ourselves. This
idea is based on a simple principle, for the same low-textured regions when patch window size is fixed, the smaller image scale is, the richer the texture information is. Then, we will make the best of this information at coarser scales to settle this problem that the untextured areas is hard to match at finer scales.

2. METHOD

Our proposed method shares the similar pipeline that the procedure of depth map-based method, and we mainly focus on computing depth map. All details will be presented as follows.

2.1. View Selection and Point Matching

The input of the MVS is the results of SFM which is called calibrated images, and SFM is handled by [2]. To put it simply, the calibrated images are so called images with camera parameters and 3D point cloud sets. First step is choosing a set of neighboring images for every input image. Theoretically, using every image as reference image is good for the final result, while we just select some great images in practice. The great images that we define are following two principles. 1) A neighbor image and a reference image are commonly sharing some features, in other words, they have enough overlapping area. 2) The distance of baseline that between the camera center of neighbor image and reference image has enough length, not big or small.

The next step is to match points. Firstly, the parameters initiations are performing for every pixel with the depth and the normal. Note that our initiations will utilize the multi-threaded to improve the efficiency. Then, for every reference image $I_r$, given its candidate image $I_v$, and the camera parameters is corresponding to $\{K_r, R_r, C_r\}$ and $\{K_v, R_v, C_v\}$. Given a random pixel $p$ with the normal $n$ and the depth $d$ in $I_r$, we need to find its corresponding point in $I_v$. This relevancy is defined as follows:

$$x' = Hx$$

where $x'$ and $x$ are in $I_v$ and $I_r$ respectively, and $H$ is a homography which is defined as:

$$H = K_v \left( R_v R_r^{-1} + \frac{R_v (C_r - C_v)n^T}{n^T X} \right) K_r^{-1}$$

where $X = dK_r^{-1}(x \ y \ 1)^T$ which is a 3D point, and putting a feature point to three-dimensional space we can get $X$.

2.2. Multi-Scale Depth Map Scheme

After multiple correspondences between the reference image and its candidate image sets are calculated, we need calculate the photometric consistency, i.e., the matching score will be computed for every pixel used to PM. The matching score is computed as the Normalized Cross Correlation (NCC), namely, computing the color similarity, and the maximum value will be chose for every reference image.

Before plane propagation, our multi-scale depth map scheme is proposed. Note that our method is different from [8]. Specifically, all input images are downsampled $l$-scale ($l > 0$), among that $0$-scale denotes the original image and the corresponding camera external parameters will change at the same time, while the camera internal parameters will not change. Then, using Section A point matching method to calculate the corresponding relationship. We use the smallest scale image called coarsest scale to perform PM. In order to improve the speed of computation, we use GPU to calculate depth map. In detail, sequential propagation is used to the propagation of the depth and the normal. When the matching score of the current pixel is less than its neighboring pixels, then the depth and normal of the current pixel are replaced by neighboring pixels, and this process is repeated until all pixels are compared. Note that we use GPU to accelerate plane propagation, and all reference images are parallel computed. In this paper, our plane propagation executes four times.

When the depth maps of coarsest reference images are completed, the next step is upsampling all depth maps. We employ joint bilateral upsampler [9] which upsamples the $l$-scale depth maps to $(l-1)$-scale. Then we get all $(l-1)$-scale depth maps. After that multi-view geometry relation will be
applied to reduce the upsampling error which refers to the difference between $l$-scale pixels and $(l-1)$-scale pixels that are uncertain. Inspired by [4,10], we adopt the forward-backward reprojection scheme to reduce this error and put the depth and the normal of these pixels to true values. Given a pixel $p$ with the known depth and normal, the upsampling error is defined as:

$$\tau = ||p_2 - p||$$

(3)

where $p_2$ is handled by two backward reprojections and two forward reprojections. When $\tau < k$ (in this paper we set $k = 0.3$), we keep the original values, otherwise, the values of the current pixel are replaced.

Repeating above processes until the scale of depth map is 0 and we can get the final depth maps. The following procedure is reprojecting all depth maps to 3D space and merging them, and we refer to the method in [11]. To this end, a whole point cloud is reconstructed.

3. EXPERIMENTAL RESULTS

All of our experiments are implemented on PC with Intel Core i7-8700 CPU, NVIDIA GeForce RTX 2060 GPU and 32G RAM. The method is implemented in C++ and ran on CPU, except that PM method is implemented in CUDA and ran on GPU. We evaluate respectively the effectiveness of our method on public datasets (such as Fountain_P11 from [3] etc.), a series of images in GitHub taken by others, and lion dataset taken by ourselves. We will describe our results in detail from three parts.

3.1. Depth Map Evaluation

In this paper, we set $l = 2$, namely, the scales of images are three. Different scales of the two views are depicted in figure 1 and we can clearly see the results of depth maps become better.

Recently, COLMAP [10] is a mature software for SFM and MVS which is used to our sparse reconstruction for acquiring camera pose. Our proposed method is used to reconstruct dense model and our results will compare to without using multi-scale depth map method. Apparently, our approach has significant results for its good depth map estimation as is shown in figure 2. The number of plane propagation that we set 4, and the normal maps of different scales are shown in figure 3.

(a) Reference image    (b) 2-scale depth map   (c) 1-scale depth map    (d) 0-scale depth map

Figure 1. Different scales of the two views from Fountain P11 dataset. Obviously, 0-scale depth map is better than the others.
Figure 2. The comparison between depth map computed with and without our proposed method. Obviously, (a) is better than (b). More importantly, the low-texture areas can be computed greatly, such as green region in (a).

3.2. Point Cloud Evaluation

Three datasets are used to evaluate the final point cloud. As is depicted in figure 4, in terms of accuracy, our experimental results demonstrate competing performance against the state of the art.

|                     | fountain-P11 | stone-P34 | lion-P28 |
|---------------------|--------------|-----------|----------|
| CPU                 | 70min        | 150min    | 115min   |
| CPU+GPU             | 25min        | 45min     | 36min    |
| Ratio               | 2.80         | 3.33      | 3.19     |

3.3. Speed

The speed of the proposed method is validated on three datasets, and results are shown in table 1. Note that we merely test one scale. From the table we can see that the ratio of acceleration achieves a speed from 2.80 to 3.33.

Figure 3. The normal maps of different scales. The scales are 2, 1 and 0 in turn from top row to bottom row. Every column denotes one propagation.
CONCLUSION

In this work, we have presented an efficient multi-scale depth map based MVS method for depth map estimation which takes accuracy and efficiency into account. The key of our method is multi-scale depth map computation. Based on large-scale scenes, multiple images could be easily parallelized in GPU. In terms of low-texture areas, these areas would be well reconstructed. In future works, we plan to improve the quality of the reconstructed model by light field information.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 61801279, in part by the Shanxi Province under Grant 201801D221160 and Grant 2019L0471 and in part by Postgraduate Innovation Project of Shanxi Province under Grant 2020SY174 and 2020SY177.

References

[1] Brown, M., & Lowe, D. G. (2005). Unsupervised 3D object recognition and reconstruction in unordered datasets. In: Fifth International Conference on 3-D Digital Imaging and Modeling (3DIM'05). Ottawa, ON, Canada. pp. 56-63.

[2] Schonberger, J. L., & Frahm, J. M. (2016). Structure-from-motion revisited. In: Proceedings of the IEEE conference on computer vision and pattern recognition. Las Vegas, Nevada. pp. 4104-4113.

[3] Strecha, C., Von Hansen, W., Van Gool, L., Fua, P., & Thoennessen, U. (2008, June). On benchmarking camera calibration and multi-view stereo for high resolution imagery. In: 2008 IEEE conference on computer vision and pattern recognition. pp. 1-8.

[4] Xu, Q., & Tao, W. (2019). Multi-scale geometric consistency guided multi-view stereo. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Long Beach, CA. pp. 5483-5492.
[5] Xu, Z., Liu, Y., Shi, X., Wang, Y., & Zheng, Y. (2020) MARMVS: Matching Ambiguity Reduced Multiple View Stereo for Efficient Large Scale Scene Reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Online. pp. 5981-5990.

[6] Shen, S. (2013). Accurate multiple view 3d reconstruction using patch-based stereo for large-scale scenes. IEEE transactions on image processing., 22(5): 1901-1914.

[7] Wu, P., Liu, Y., Ye, M., Li, J., & Du, S. (2016). Fast and adaptive 3D reconstruction with extensively high completeness. IEEE Transactions on Multimedia., 19(2): 266-278.

[8] Wu, P., Liu, Y., Ye, M., Xu, Z., & Zheng, Y. (2017). Geometry guided multi-scale depth map fusion via graph optimization. IEEE transactions on image processing., 26(3): 1315-1329.

[9] Kopf, J., Cohen, M. F., Lischinski, D., & Uyttendaele, M. (2007). Joint bilateral upsampling. ACM Transactions on Graphics (ToG)., 26(3): 96-es.

[10] Schönberger, J. L., Zheng, E., Frahm, J. M., & Pollefeys, M. (2016, October). Pixelwise view selection for unstructured multi-view stereo. In: European Conference on Computer Vision. Amsterdam, The Netherlands. pp. 501-518.

[11] Goesele, M., Snavely, N., Curless, B., Hoppe, H., & Seitz, S. M. (2007, October). Multi-view stereo for community photo collections. In: 2007 IEEE 11th International Conference on Computer Vision. Rio de Janeiro, Brazil. pp. 1-8.