Final Report

Group 13: Stereo Reconstruction

1 Introduction

Stereo reconstruction is a very active research field in computer vision, which has a wide range of applications in architecture capturing and autonomous driving. In this project, we apply different stereo matching methods to reconstruct 3D scenes and compare their performance. Based on key-point detectors and eight-point algorithm, we recover the camera's extrinsic and rectify the images from left and right camera for the next step. Then we apply three dense matching methods to generate the disparity map respectively and further reconstruct the 3D scene. We evaluate the impact of different detectors and bundle adjustment on the accuracy of the estimated transformation. The experiment shows that SIFT performs better than ORB, and the accuracy of the estimated transform is also improved after using bundle adjustment. For dense matching, with PSMNet, which is the SOTA in disparity prediction, we can get much higher precision than classic methods block matching and semi-global matching. We release the code of our project here: `https://github.com/Dekai21/Stereo_Reconstruction`.

2 Related Work for Stereo Matching

As a classical research topic for decades, traditional stereo matching algorithms can be grouped into three categories: (1) local methods, (2) global methods, and(3) semi-global methods.

Local methods[1][2][3] are done by selecting the disparity with the lowest matching cost. The disparity is conventionally determined by matching a predefined support window of pixels by using different similarity metrics. It runs very fast, but the errors are significant, especially in occlusion areas. Compared with local method, global method[4][5][6] considers the influence of all pixels to eliminate the impact of local areas which can cause errors. It brings better accuracy but also more computational complexity. Semi-global method[7] approximately solves the NP-hard 2D graph partitioning by optimizing a pathwise form of the energy function in many directions. This method achieves a fair trade-off between computation complexity and accuracy.

The implementation of deep learning in stereo vision has been tremendously excellent. MC-CNN[8] successfully substitutes handcrafted matching cost metrics with deep metrics and achieves considerable gain compared to traditional approaches in terms of both accuracy and speed. Chang and Chen[9] employ a spatial pyramid pooling module to extract multiscale representations, compute a cost volume from both image features (encoding) and incorporate a stacked 3D CNN to aggregate contextual features and regress (decoding) disparity maps. This work is used in this project for comparsion.

3 Method

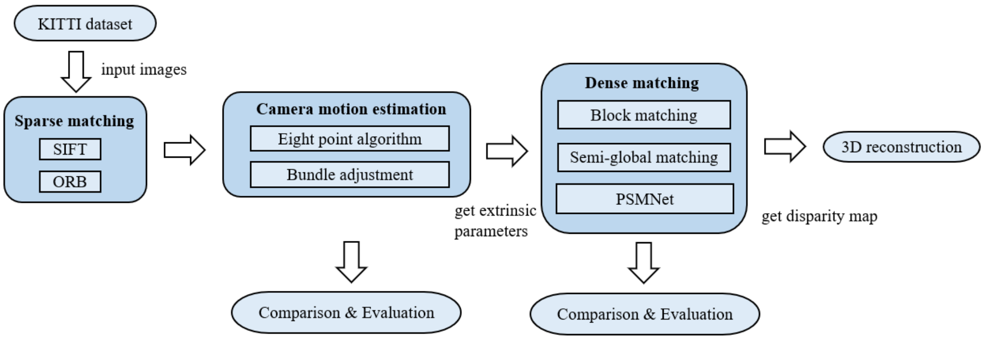


Figure 1: Method overview.

We use KITTI[10] dataset, which provides images of two RGB cameras and corresponding depth information. First, we apply key point detection methods, e.g. SIFT[11] and ORB[12] to extract key points and find best matching pairs from left image and right image. Based on matched point pairs we conduct eight-point algorithm to estimate the external parameters. Then we use bundle adjustment to further refine the result. By computing the error of computed parameters with ground truth, we study the performance of different detectors and camera motion estimation methods.

With extrinsics of the camera, we rectify the left image and right image. Due to the insufficient precision of extrinsics we estimate, we also consider directly using the rectified images provided in the dataset. Then we apply dense matching methods block matching, semi-global matching and PSMnet to construct disparity map.

Finally, we use the disparity map to generate point cloud in 3D space by backprojection. We write a 3D mesh out of it and complete the reconstruction by triangulation. A general workflow is illustrated in Fig. 1.

4 Result

4.1 Comparison of different sparse matching methods

We select 30 images from the KITTI2015 dataset as our test dataset and get their corresponding unrectified images from raw data. We use ORB and SIFT to detect key points in left and right images, which are then used in eight-point algorithm to recover the rotation and translation between left and right images. Then we use bundle adjustment to improve the accuracy. For both SIFT and ORB, we choose 40 keypoint pairs in the image. As for bundle adjustment, we use the result from eight-point algorithm as initialization. For the evaluation of rotation matrix, we convert the rotation matrix into zyx Euler angles and compute the mean squared error. For translation vector, we compute the mean squared error directly. The quantitative results are shown in Table 1.

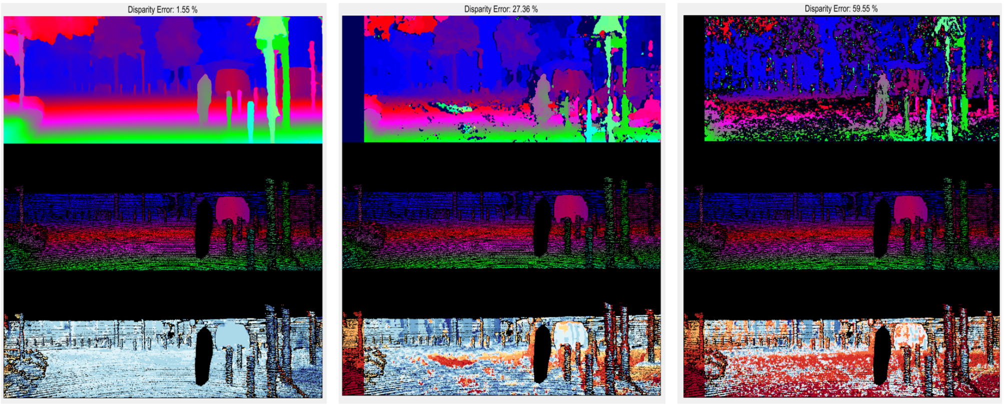
	ORB	SIFT	ORB + BA	SIFT + BA
Euler angle error (degree)	11.575	3.526	7.835	1.380
MSE (m)	0.417	0.316	0.371	0.212

Table 1: Comparison of different detectors, we try four combinations and "BA" represents bundle adjustment.

As the result shows, compared with ORB, the accuracy of the methods using SIFT is greatly improved, but in fact the calculation time of SIFT is slightly longer. In addition, the optimization effect of bundle adjustment is obvious. But overall, the accuracy we obtained is very low, especially the translation vector, given that the baseline is only about 0.54m. Such accuracy is not enough for us to obtain high-quality rectified images. Considering that our focus is to compare the performance of different stereo matching methods, we decide to use the rectified images provided by the dataset to do the follow-up work.

4.2 Comparison of different dense matching methods

We use the rectified images provided in KITTI2015 dataset to get disparity maps with 3 different methods, namely block matching, semi-global matching and PSMNet. For evaluation, we calculate an error rate, that is the percentage of error pixels to all valid pixels. And an error pixel means the difference between calculated disparity value and ground truth is greater than $3dm$ and greater than 5%. The qualitative results are shown in Figure 2, the first row shows our disparity map, the second row shows the ground truth, and the third row shows the difference between our result and ground truth.



PSMnet

Semi-global matching

Block matching

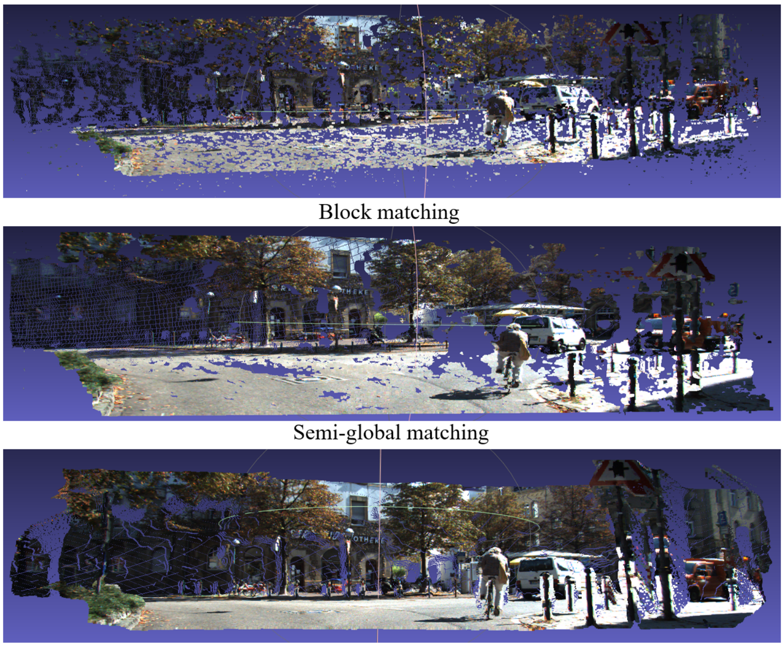
Figure 2: Disparity map evaluation

For comparison, we calculate the average error rate for each method. From Table 2, we know that PSMNet works best, it can ensure that nearly 99% of pixels have an disparity error within 5% or under $3dm$. While block matching has the worst effect, only half of the pixels have lower disparity error.

Method	Block matching	Semi-global matching	PSMNet
Average error rate	53.1%	19.05%	1.07%

Table 2: Comparison of different dense matching methods

After getting disparity map, we can calculate depth and get 3D points, then we can generate 3D meshes by triangulation. Figure 3 gives an example of our mesh reconstruction result.



Block matching

Semi-global matching

PSMnet

Figure 3: Mesh reconstruction

5 Conclusion

In this project, we achieve a pipeline of stereo reconstruction and compare different methods in sparse matching and dense matching. For sparse matching, in general, SIFT runs slower but provides higher accuracy than ORB. With bundle adjustment we can achieve better accuracy than eight point algorithm, but none of these methods we tried can be accurate enough to be applied in subsequent stereo matching, which can be further improved in future work. For dense matching, semi-global method achieves considerable gain compared with block matching in accuracy and speed, but deep learning method outperforms greatly classic methods.

References

[1] P. Anandan. A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision*, 2:283–310, 2004.

[2] Christos Georgoulas and Ioannis Andreadis. Fpga based disparity map computation with vergence control. *Microprocessors and Microsystems*, 34(7):259–273, 2010.

[3] H. Hirschmüller and D. Scharstein. Evaluation of cost functions for stereo matching. *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[4] Eu-Tteum Baek and Yo-Sung Ho. Occlusion and error detection for stereo matching and hole-filling using dynamic programming. *electronic imaging*, 2016:1–6, 2016.

[5] Li Hong and G. Chen. Segment-based stereo matching using graph cuts. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 1, pages I–I, 2004.

[6] Michel Sarkis and Klaus Diepold. Sparse stereo matching using belief propagation. In *2008 15th IEEE International Conference on Image Processing*, pages 1780–1783, 2008.

[7] Heiko Hirschmuller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):328–341, 2008.

[8] Jure Zbontar and Yann LeCun. Computing the stereo matching cost with a convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[9] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[10] Moritz Menze, Christian Heipke, and Andreas Geiger. Joint 3d estimation of vehicles and scene flow. In *ISPRS Workshop on Image Sequence Analysis (ISA)*, 2015.

[11] D.G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157 vol.2, 1999.

[12] Shaharyar Ahmed Khan Tareen and Zahra Saleem. A comparative analysis of sift, surf, kaze, akaze, orb, and brisk. In *2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, pages 1–10, 2018.