Stereo Reconstruction

Li Zhou
li.zhou@tum.de

Danya Liu
danya.liu@tum.de

Ang Li
ang30.li@tum.de

Abstract

Stereo reconstruction is the process of creating a 3D representation of a scene or an object using multiple(in this project just two) 2D images captured from different viewpoints. This report proposed a whole pipeline of stereo reconstruction that utilized several feature detectors such as SIFT, SURF and so forth at the beginning. Then feature matching algorithms including Brute-Forth matcher and FLANN were performed. After that, the eight-point or five-point algorithm was implemented to obtain the fundamental matrix and recover the pose based on the calculated matrix. Afterwards, the bundle adjustment algorithm was utilized to optimize the recovered pose mentioned above. To get desired disparity maps, image rectification and dense matching methods such as SGBM were implemented as well. Eventually, reconstructing 3D models and generating corresponding meshes were done. Experiments on several benchmark dataset showed that the proposed method performed well in terms of accuracy, speed, and robustness. The results demonstrated the effectiveness of using such algorithms for stereo reconstruction.

1. Introduction

As mentioned above, stereo reconstruction technology is utilized in different fields. And in the past few years, there has been great progress in stereo reconstruction because of advanced computer vision algorithms and available large datasets. However, the challenge of obtaining disparity maps from images based on traditional methods is still here, which is the motivation of this report as well. And this report provides a concise overview of the proposed pipeline for stereo reconstruction.

2. Related Work

2.1. Feature Detectors

What should be mention is that feature detectors play an important role in computer vision with the use of identifying distinctive points namely key points in images. And this section will introduce various feature detectors utilized in this project.

- SIFT (Scale-invariant Feature Transform)
  SIFT [1] is a popular feature detection algorithms which can detect local features in images which are invariant to scale, orientation and affine distortion. Its features can be presented by a set of key points, each of which can be described by a 128-dimension vector.
- SURF (Speeded Up Robust Features)
  SURF [2] is a feature detection algorithm, which has the same principles as SIFT. However, SURF utilizes a different approach to detect key points. Besides, its features can be presented by a set of key points as well, each of which can be described by a 64-dimension vector instead of a 128-dimension vector. Therefore, SURF is faster than SIFT because of the smaller size of feature vectors. And it is more robust to noise and occlusions, but less invariant to scale and orientation.
- ORB (Orientation FAST and Rotated BRIEF)
  ORB [3] is an efficient feature detection algorithm with the combination of the FAST corner detector with the BRIEF descriptor. And its features can be presented by a set of key points, each of which can be described by a 256-dimension vector. ORB is faster than SIFT and SURF and it is more robust to noise and occlusions, however, it is less invariant to scale and orientation.
- FREAK (Fast Retina Keypoint)
  FREAK [4] is a binary descriptor, created to be fast and robust to affine transformations. It utilizes a binary string to represent the gradient orientation of a feature. What should be mentioned is that FREAK is just a descriptor; therefore, it should be used together with a detector such as FAST. However, FREAK is sensitive to rotations which might affect detection results in some situations.
- BRISK (Binary Robust Invariant Scalable Keypoints)
  BRISK [5] is a binary feature detector and descriptor, which is designed to be fast and perform well in low-light condition. And it utilizes a binary string to represent the gradient orientation of a feature. However,

BRISK is sensitive to scale changes which might affect detection results in some situations.

- KAZE (KAnade-AEgaki-SIngh-Kosecka)
  KAZE [6] is a non-binary feature detector and descriptor, which is designed to detect features in non-flat regions. KAZE utilizes a Gaussian scale space to find features. However, KAZE is slower than other feature detectors, making it not available for real-time detection.

2.2. Matching Methods

- BF Matching (Brute-Forth)
  Brute-Forth Matching is a straightforward algorithm which compares every feature in one image to every feature in another image. The algorithm calculates the distance between two features and selects the closest match. And this process is repeated for all the features in the first image, while the result is a set of matches between two images.

- FLANN Matching (Fast Library for Approximate Nearest Neighbors)
  FLANN Matching [7] is an efficient algorithm which utilizes an index-based method to find correspondences between features. This algorithm creates an index of the features in one image firstly and then searches for the closest match in the index for each feature in another image. It is faster than Brute-Forth matching, therefore, it is usually utilized for large datasets.

- RANSAC (Random Sample Consensus)
  RANSAC [8] is a robust parameter estimation algorithm widely used in computer vision for object detection and recognition. RANSAC works by randomly sampling a minimal subset of data points from the input, and then fitting a model to these points. This model is evaluated against the remaining data, and points that are consistent with the model are retained. This process should be repeated multiple times, and the final model is selected as the one that fits the largest number of inliers.

2.3. Epipolar Geometry

- Eight-Point Algorithm
  The Eight-Point algorithm [9] is one of the simplest algorithms for obtaining the fundamental matrix. It utilizes eight pairwise points between given images to calculate the fundamental matrix. However, it is sensitive to noise and outliers.

- Five-Point Algorithm
  The Five-Point algorithm [10] utilizes five pairwise points between given images to calculate the fundamental matrix. And it implements a non-linear optimization approach, which makes it more robust compare to the Eight-Point Algorithm.

2.4. Bundle Adjustment

Bundle Adjustment [11] aims at minimizing the reprojection error between the observed 2D coordinates and the projected 3D coordinate. And the formula is shown in figure 1.

$$E(R, T, \boldsymbol{X}_1, \ldots, \boldsymbol{X}_N) = \sum_{j=1}^{N} \left|\tilde{\boldsymbol{x}}_1^j - \pi(\boldsymbol{X}_j)\right|^2 + \left|\tilde{\boldsymbol{x}}_2^j - \pi(R, T, \boldsymbol{X}_j)\right|^2$$

Figure 1. Formula of bundle adjustment

2.5. Dense Matching

- SGBM (Semi-Global Block Matching)
  SGBM [12] is the combination of local and global algorithms, which uses a cost aggregation method to optimize the matching result for each pixel. It can generate a smoother and more accurate depth map compared to Block Matching which will be discuss later.

- BM (Block Matching)
  BM [13] is a local algorithm, which is fast and efficient for real-time applications. The matching cost is calculated based on the intensity difference between pixels in two images. However, the generated depth map by using BM algorithm is less accurate compared to the SGBM algorithm.

3. Method

The overall method(pipeline) of this project is shown below in Figure 2. The dataset we use in the project is the middlebury dataset(2014) [14]. It's a dataset that contains multiple high quality stereo image pairs together with their ground truth information. In this pipeline, it's worth mentioning the mathematical principle behind eight-point algorithm. It makes use of epipolar constraint and solve the equation shown in Figure 3 to get the fundamental matrix where x_1, x_2 are coordinates of the points and F is the fundamental matrix. After that we can recover the pose from the fundamental matrix. Another point worth emphasizing is the conversion from disparity map to depth map. When converting the disparity map to depth map, we followed the formula shown in Figure 4, where B is the baseline, f is the focal length, and Z is the disparity.

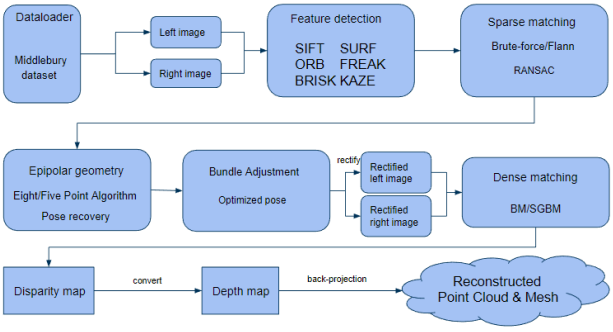

Figure 2. Pipeline of this project

$$\boldsymbol{x}_2'^{\top} F \boldsymbol{x}_1' = 0$$

Figure 3. Formula of eight-point algorithm

$$disparity = x - x' = \frac{Bf}{Z}$$

Figure 4. Conversion from disparity to depth

Figure 5. matches using SIFT

4. Results

We conducted several experiments among feature detection and dense matching by incorporating different experimental settings. The corresponding results are listed below in sections.

4.1. Feature Detection and Matching

Table 1 displays the number of keypoints detected by six different detectors. Using the detected keypoints, we further obtained the paired matches for each detector as shown in Table 2. This table also includes a comparison of the results when using RANSAC and different sparse matching techniques, such as BF Matcher and FLANN Matcher. With the exception of FREAK, the number of matches is generally in line with the number of detected keypoints, indicating the validity of our matching results. After the application of RANSAC, only the inliers were kept for further analysis. For visualization, the matches detected by SIFT on playtable-perfect and piano-perfect from middlebury dataset can be seen in Figure 5.

detector	keypoints
SIFT	17380
SURF	18256
ORB	500
FREAK	51824
BRISK	3524
KAZE	8287

Table 1. **Number of keypoints** are recognized by six detectors on playtable-perfect.

	without RANSAC		with RANSAC	
Detector	BF	FLANN	BF	FLANN
SIFT	769	717	159	275
SURF	1065	1057	40	290
ORB	41	2	23	2
FREAK	7	115	7	37
BRISK	209	70	50	32
KAZE	311	1200	90	444

Table 2. **Number of matches** for sparse matching are compared on playtable-perfect for six detectors respectively.

4.2. Pose Evaluation

The outcome of the transformation matrix can be observed in Table 3 and Table 4 after obtaining the matched points. It is evident that the initial results from the five-point algorithm are significantly better than those from the eight-point algorithm due to its robustness against random selections of matched points. On the other hand, bundle adjustment suffer greatly from noisy points, resulting in a deviation from the optimal direction. However, when combined with RANSAC, bundle adjustment is still crucial to optimize the poses.

4.3. Block Matching

To assess the results of dense matching, we employ four metrics - BAD0.5, BAD2.0, BAD4.0, and RMS which can be seen in Table 5. For BAD0.5, BAD2.0 and so on, their meaning can be revealed by the formula shown in Figure 9. Briefly, it computes the proportion of the differences between the estimated value and the true value that are greater

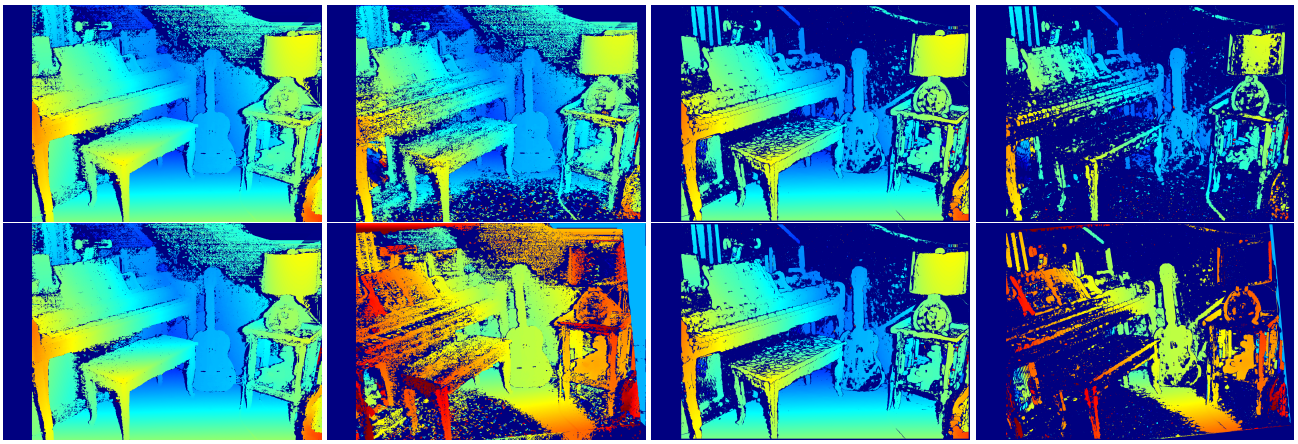

Figure 6. **Color disparity map** on piano-perfect. From left to right and top to bottom, **1:** 8-point+SGBM+BA, **2:** 8-point+SGBM, **3:** 8-point+BM+BA, **4:** 8-point+BM, **5:** 5-point+SGBM+BA, **6:** 5-point+SGBM, **7:** 5-point+BM+BA, **8:** 5-point+BM.

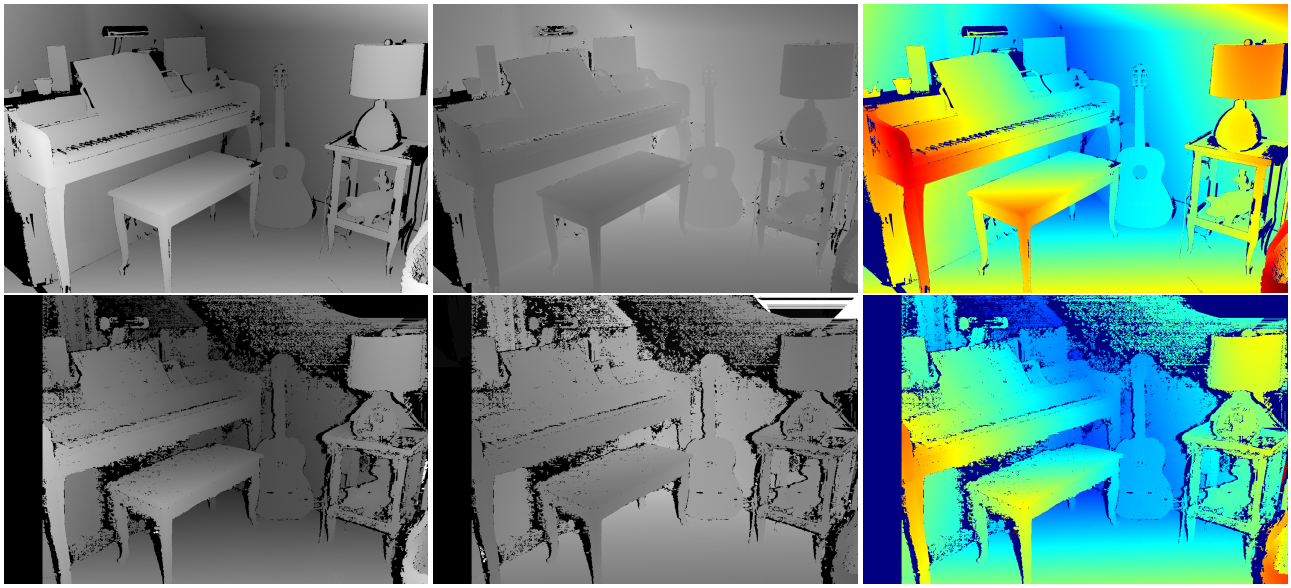

Figure 7. **Disparity map and depth map**. The first line shows the ground truth while the second line shows the rectified image.

	Eight-Point	Five-Point
initial	0.06073	0.00136
BA	0.11895	0.00466
RANSAC	0.00051	0.03602
BA & RANSAC	0.00044	0.00044

Table 3. **L2-distance for Rotation**$\{\downarrow\}$ in different settings considering existence of bundle adjustment and RANSAC. Initial is the direct result after five/eight-point algorithm.

	Eight-Point	Five-Point
initial	1.91453	0.11951
BA	0.30961	1.99959
RANSAC	0.02135	0.11126
BA & RANSAC	0.00319	0.00319

Table 4. **L2-distance for Translation**$\{\downarrow\}$ in different settings considering existence of bundle adjustment and RANSAC. Initial is the direct result after five/eight-point algorithm.

than a certain value. And for RMS, we can refer to the formula in Figure 10, and it computes the root mean square of the differences between the estimated value and the true value. For these four metrics, the lower value means better result. The importance of bundle adjustment is again emphasized, as it significantly improves the outcome of dense matching for both BM and SGBM. Overall, SGBM outperforms BM in all metrics, making it the more suitable dense

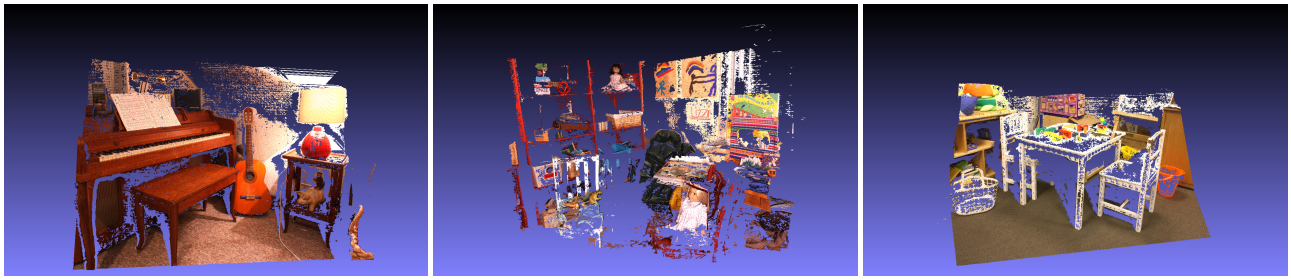

Figure 8. **3D meshes** on piano-perfect, playroom-perfect, playtable-perfect.

matching technique for this dataset.

Additionally, we present our results visually in the form of color disparity maps, as seen in Figure 6. When using bundle adjustment, the disparity map accurately portrays the relative position and potential 3D structure. Conversely, without bundle adjustment, the disparity map is not reliable due to the highly deviated transformation matrix, leading to noise or meaningless smooth holes. A comparison between SGBM and BM reveals that SGBM provides more solid depth information for objects close in proximity, such as the piano and piano stool, while BM offers more consistent depth information when image pixels are significantly impacted by lighting. Finally, we also verify the refined results of maps **1** in Figure 6 against the ground truth. As SGBM surpasses BM in all evaluation metrics, we will generate meshes based on its results.

$$\frac{1}{N}\sum_{(x,y)\in N}\{|d_{est}(x,y)-d_{gt}(x,y)|>\delta_D\}$$

Figure 9. Formula of BAD δ_D metric

$$\sqrt{\frac{1}{N}\sum_{(x,y)\in N}|d_{est}(x,y)-d_{gt}(x,y)|^2}$$

Figure 10. Formula of RMS metric

	BAD0.5↓	BAD2.0↓	BAD4.0↓	RMS↓
BM	86.61	86.58	86.56	100.36
BM & BA	54.44	39.42	38.53	63.29
SGBM	86.98	86.93	86.85	92.19
SGBM & BA	50.19	27.04	24.92	45.13

Table 5. **Evaluation of dense matching.** We select BM and SGBM as our dense matching method, which is here based on the result of five-point algorithm. BAD0.5{↓}, BAD2.0{↓}, BAD4.0{↓} and RMS{↓} are used as the evaluation metrics.

4.4. Mesh Generation

The final reconstructed meshes are displayed in Figure 8. It is evident that the relative positioning is correct and the overall reconstruction performance is satisfactory. However, there are some missing parts in shadows and some flat slices orthogonal to the camera's capture direction, which still reveals the limitation of detectors and matching methods towards the stereo reconstruction problem.

5. Conclusion

In summary, SIFT is considered the most reliable feature detector as it can perform well in various experimental conditions, as shown in Table 2. To determine the pose, either the eight-point algorithm or the five-point algorithm can only provide rough estimates. On the other hand, RANSAC is able to identify and eliminate outliers, which provides a solid foundation for bundle adjustment to optimize the pose. Only after these two crucial steps, the resulting transformation matrix is sufficiently accurate compared to the ground truth based on L2-distance. Regarding dense matching methods, SGBM is superior to BM in all evaluation metrics, as indicated in Table 5. Further research can involve exploring other block matching methods since SGBM is not robust to changes in lighting conditions.

References

[1] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 1

[2] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006. 1

[3] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *2011 International conference on computer vision*, pages 2564–2571. Ieee, 2011. 1

[4] Alexandre Alahi, Raphael Ortiz, and Pierre Vandergheynst. Freak: Fast retina keypoint. In *2012 IEEE conference on computer vision and pattern recognition*, pages 510–517. Ieee, 2012. 1

[5] Stefan Leutenegger, Margarita Chli, and Roland Y Siegwart. Brisk: Binary robust invariant scalable keypoints. In *2011 International conference on computer vision*, pages 2548–2555. Ieee, 2011. 1

[6] Pablo Fernández Alcantarilla, Adrien Bartoli, and Andrew J Davison. Kaze features. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part VI 12*, pages 214–227. Springer, 2012. 2

[7] Marius Muja and David Lowe. Flann-fast library for approximate nearest neighbors user manual. *Computer Science Department, University of British Columbia, Vancouver, BC, Canada*, 5, 2009. 2

[8] Konstantinos G Derpanis. Overview of the ransac algorithm. *Image Rochester NY*, 4(1):2–3, 2010. 2

[9] Richard I Hartley. In defense of the eight-point algorithm. *IEEE Transactions on pattern analysis and machine intelligence*, 19(6):580–593, 1997. 2

[10] David Nistér. An efficient solution to the five-point relative pose problem. *IEEE transactions on pattern analysis and machine intelligence*, 26(6):756–770, 2004. 2

[11] Bill Triggs, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon. Bundle adjustment—a modern synthesis. In *Vision Algorithms: Theory and Practice: International Workshop on Vision Algorithms Corfu, Greece, September 21–22, 1999 Proceedings*, pages 298–372. Springer, 2000. 2

[12] Christian Banz, Holger Blume, and Peter Pirsch. Real-time semi-global matching disparity estimation on the gpu. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 514–521. IEEE, 2011. 2

[13] Aroh Barjatya. Block matching algorithms for motion estimation. *IEEE Transactions Evolution Computation*, 8(3):225–239, 2004. 2

[14] Daniel Scharstein, Heiko Hirschmüller, York Kitajima, Greg Krathwohl, Nera Nešić, Xi Wang, and Porter Westling. High-resolution stereo datasets with subpixel-accurate ground truth. In *German conference on pattern recognition*, pages 31–42. Springer, 2014. 2