Stereo Reconstruction Project

Baran Özer
Yidi Ma
Han Wu
Shiyu Zou

Abstract

This work presents a stereo reconstruction pipeline that extracts accurate 3D structure from stereo image pairs using classical computer vision techniques. The proposed approach encompasses key components such as sparse keypoint matching, fundamental matrix estimation, stereo rectification, disparity map computation, and dense 3D reconstruction. Optimizations, including modifications to the 8-point algorithm and disparity refinement, are investigated to improve reconstruction accuracy. The performance of the method is evaluated on standard benchmark datasets, including Middlebury and ETH3D, using both qualitative and quantitative metrics. Experimental results demonstrate the effectiveness of the proposed approach in generating high-fidelity 3D models from stereo images.

1. Motivation and Idea

3D reconstruction creates virtual 3D representations of objects or scenes, with applications in graphics, animation, medical imaging, and VR. With the availability of affordable 2D imaging devices like smartphone cameras, 2D images offer a cost-effective alternative to specialized 3D scanning technologies. Stereo reconstruction uses two or more 2D images to generate 3D models. In this project, we implemented stereo reconstruction, covering sparse keypoint matching, the 8-point algorithm, rectification, disparity map generation, dense reconstruction, and mesh creation. We also conducted qualitative and quantitative analyses, optimized the 8-point algorithm and dense reconstruction, and compared our results to state-of-the-art methods.

TODO: We will test our implementations on the Middlebury [1] and ETH3D [2] dataset.

2. Related Work

Stereo reconstruction is a well-studied problem in computer vision, with classical methods using epipolar geometry and feature-based matching, while newer deep learning techniques utilize large datasets for disparity estimation.

Early works such as those by Hartley and Zisserman [3] laid the foundation for stereo vision through fundamental matrix estimation and rectification. The development of robust feature descriptors, including SIFT [4] and ORB [5], significantly improved keypoint matching, which is crucial for sparse stereo correspondence.

Disparity estimation has seen significant advances with the introduction of Semi-Global Matching (SGM) [6], which balances computational efficiency and accuracy. More recent approaches leverage convolutional neural networks (CNNs) to estimate disparities, as demonstrated by GC-Net [7] and PSMNet [8]. These methods learn to extract features and predict disparities directly from stereo image pairs, improving robustness to textureless regions and occlusions.

Benchmark datasets such as Middlebury [1] and ETH3D [2] have played a crucial role in evaluating stereo reconstruction methods by providing high-resolution stereo images with ground truth depth maps.

Despite the success of learning-based methods, classical techniques remain relevant for their explainability and lower dependency on large datasets. Recent research explores hybrid methods that combine deep learning with geometric constraints for better results.

3. Method

Our approach for stereo reconstruction includes the following key components:

3.1. Sparse Keypoint Matching

Sparse keypoint matching is a crucial step in stereo reconstruction, as it establishes correspondences between feature points in the left and right images. In this project, Oriented FAST and Rotated BRIEF (ORB) is utilized for efficient keypoint detection and descriptor computation, followed by brute-force matching using the Hamming distance.

3.1.1 Feature Detection using ORB

Keypoints are detected using FAST: pixel I_c is a keypoint if intensity difference with surrounding pixels exceeds threshold T:

$$\max_{p \in \mathcal{N}(I_c)} |I_p - I_c| > T, \qquad \theta = \arctan\left(\frac{\sum y I_{xy}}{\sum x I_{xy}}\right), \tag{1}$$

where I_p is a pixel in neighborhood $\mathcal{N}$, and θ is orientation via intensity centroid for rotation invariance.

3.1.2 Descriptor Computation

For each keypoint (x, y), ORB generates a binary descriptor by comparing n pixel pairs (p_i, q_i) in a predefined pattern:

$$f_i = \begin{cases} 1, & I(p_i) < I(q_i) \\ 0, & \text{otherwise} \end{cases} \quad (i = 1, \ldots, n), \tag{2}$$

resulting in a compact binary string.

3.1.3 Feature Matching

Descriptors are matched via brute-force Hamming distance:

$$d_H(\mathbf{f}_i, \mathbf{f}_j) = \sum\nolimits_k \mathbf{f}_i^k \oplus \mathbf{f}_j^k \quad (\oplus = \text{XOR}). \tag{3}$$

Matches are filtered by distance ranking and Lowe's ratio test:

$$\frac{d_{\text{best}}}{d_{\text{second-best}}} < \tau \ (\tau \approx 0.75). \tag{4}$$

3.2. Fundamental Matrix Calculation

3.2.1 8-Point Algorithm Workflow

The Fundamental Matrix F is computed via two SVD stages:

1. **Least-squares solution:** Solve $Af = 0$ via SVD of A, taking $f = V[:, -1]$ (last column of V), reshaped to 3×3 F.
2. **Rank-2 enforcement:** Recompute $F' = U\Sigma' V^T$ via SVD of F, where $\Sigma' = \text{diag}(\sigma_1, \sigma_2, 0)$.

3.2.2 Core Equations

- **Epipolar constraint:** $\mathbf{x}'^T F \mathbf{x} = 0$ with $\mathbf{x} = (x, y, 1)^T$, $\mathbf{x}' = (x', y', 1)^T$.
- **SVD components:**

$$A = U_A \Sigma_A V_A^T \quad (8 \times 9 \text{ matrix})$$
$$F = U_F \Sigma_F V_F^T \quad \text{(before rank-2)}$$

3.2.3 Key Implementation Details

Stage 1: Least-Squares Solution

- Matrix A constructed from 8+ point correspondences
- Solution f corresponds to $\sigma_{\min}$ of A

Stage 2: Rank Enforcement

- Original SVD: $\Sigma_F = \text{diag}(\sigma_1, \sigma_2, \sigma_3)$
- Modified: $\Sigma' = \text{diag}(\sigma_1, \sigma_2, 0)$
- Final F' ensures $\det(F') = 0$ (rank-2)

3.3. Image Rectification and Disparity Calculation

3.3.1 Stereo Rectification & Disparity Pipeline

Rectification: Align stereo images via homography for improved disparity.
Matching: Apply Semi-Global Matching (SGM) to refine dense disparity maps.

3.3.2 Essential Matrix and Pose Extraction

- **Essential Matrix**: $E = K^\top F K$
- **SVD Decomposition**: $E = U\Sigma V^\top$
- **Rotation & Translation**:

$$W = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$
$$R_{1,2} = UW^{\pm\top}V^\top, \quad t_{1,2} = \pm U[0, 0, 1]^\top$$

  Select (R, t) ensuring positive depth.

3.3.3 Homography Rectification

$$H_L = KR_L K^{-1}, \quad x_{\text{rect}} = H_L x,$$
$$H_R = KR_R K^{-1}, \quad x'_{\text{rect}} = H_R x'$$

3.3.4 Disparity Computation

$$d = x_L - x_R \quad \text{(rectified coordinates)}$$

3.4. Depth Map Generation and 3D Reconstruction

3.4.1 Depth & 3D Reconstruction Pipeline

Depth Estimation:

$$Z = \frac{Bf}{d}, \quad \text{normalized\_depth} = 255 \times \frac{Z - Z_{\min}}{Z_{\max} - Z_{\min}}$$

where $Z_{\min} \to 0$ (black), $Z_{\max} \to 255$ (white).

3.4.2 3D Mesh Generation

- **Point Cloud**: Project depth map via pinhole model:

$$X = (x - c_x)Z/f_x, \quad Y = (y - c_y)Z/f_y,$$
$$Z = Z(x, y)$$

- **Mesh Construction**: Split depth grid quads into triangles:

$$\triangle(p_{00}, p_{10}, p_{01}), \quad \triangle(p_{01}, p_{10}, p_{11}),$$
$$\text{valid if edge length} < 2\,\text{cm}$$

4. Results

We test and evaluate our results on the scenes from the Middlebury Stereo Dataset (2021 and 2014) [1]. It provides high-resolution image pairs, mostly of indoor scenes, with ground truth camera intrinsics and disparity maps.

For our Project, 1 shows the whole pipeline.

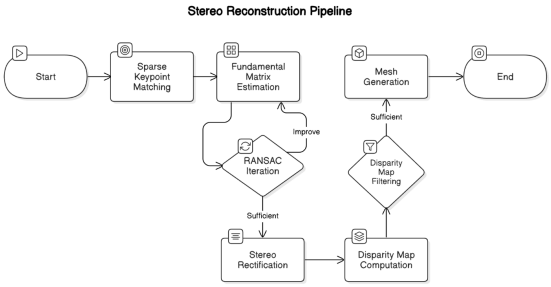


Figure 1. The whole pipeline of our project

4.1. Sparse Keypoint Matching

We applied ORB, SIFT, and BRISK feature detection methods on the Shopvac dataset from Middlebury 2014 and the courtyard dataset from ETH3D. 2 shows the distribution of the top 10 matches detected by these three methods on both datasets.

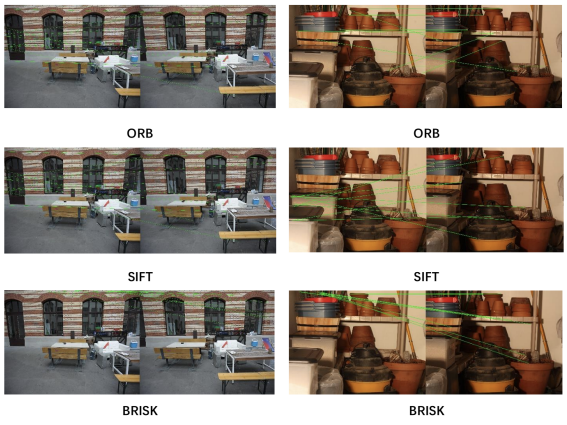
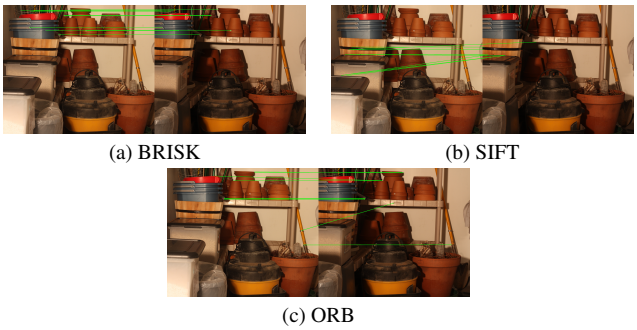


Figure 2. Different sparse keypoint matching methods.

After filtering, the results of the top 10 matches based on BRISK, ORB, and SIFT detection are shown in 3.



(a) BRISK (b) SIFT

(c) ORB

Figure 3. Top 10 Matches After Feature Detection and Filtering.

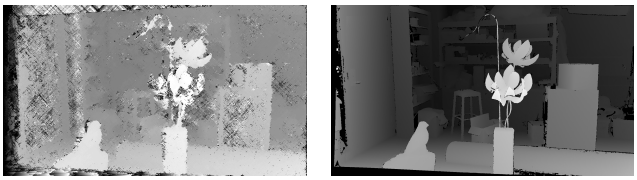
4.2. Rectification

Since the sparse keypoint matching outputs the best results when using BRISK, the following results are all generated with BRISK-based sparse keypoint matching. 4 shows the rectified image pair of the "shopvac imperfect" scene from the 2014 Middlebury dataset.

Figure 4. Rectified image pair based on BRISK of the scene "shopvac imperfect" from Middlebury 2014.

4.3. Disparity Map Computation

We calculated the disparity maps based on the rectified images using BRISK as the feature descriptor. in 5 we can see our calculated disparity map next to the ground truth disparity map from the dataset (provided as pfm file).



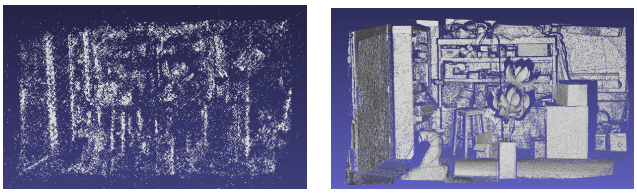
(a) Our own calculated disparity map. (b) Ground truth disparity map.

Figure 5. Ground truth and calculated disparity map of scene: "artroom1" from 2021 Middlebury dataset (with BRISK).

4.4. Depth Estimation and 3D Mesh Generation

Since the dataset contains ground truth disparity maps in the form of pfm files, we used these to test the correctness of our depth estimation and 3D mesh generation methods. As visualized in 5b and 6b our mesh generation method can reconstruct the scene visualized in the disparity map and generate correct 3D meshes.

The generated 3D mesh from our computed disparity map (5a) can be seen in 6a



(a) 3D mesh generated from our own calculated disparity map.

(b) 3D mesh generated from the ground truth disparity map.

Figure 6. 3D meshes generated from the ground truth and own calculated disparity map respectively. BRISK was used for sparse keypoint matching.

5. Analysis

5.1. Sparse Keypoint Matching

According to the data in 1, ORB provides fewer matches (500), but the average distance between matches is smaller (53.958), and its computation time is the shortest (0.117 seconds), demonstrating high efficiency. SIFT, while providing the most matches (2748), has lower match accuracy and a longer computation time (3.146 seconds). BRISK offers a balance, with the number of matches (1996) and average distance (105.13) between ORB and SIFT, and a moderate computation time (0.199 seconds), striking a balance between efficiency and accuracy.

Method	Matches	Avg. Dist.	Time (s)	MAX dist	Min dist
ORB	500	53.958	0.117	95	13
SIFT	2748	230.796	3.146	431.728	18.6815
BRISK	1996	105.13	0.199	179	15

Table 1. Feature Matching Results for Different Methods

In processing the Shopvac dataset from Middlebury 2014, we used two methods to filter the matches: dist threshold and KNN + Lowe's Ratio Test. 1 shows the raw data, with matches representing the number of matches obtained through each detection method, max dist being the maximum distance between matches, min dist being the minimum distance, and avg dist being the average distance between matches.

As shown in 7, for the ORB feature detection method, the KNN + Lowe's Ratio Test filtering method performs better,

Filter_Method	**Dist threshold**			**KNN+Lowe's Ration test**		
Detect_Method	ORB	SIFT	BRISK	ORB	SIFT	BRISK
Matches	143	163	148	59	594	282
Max dist	45	65.3605	52	54	265.902	121
Min dist	13	18.6815	15	13	18.6815	15
Avg dist	33.5385	50.4862	37.5743	29.8644	120.191	56.1915

Figure 7. The Comparision of two filter methods

while for SIFT and BRISK, the dist threshold method yields better results. Overall, the dist threshold method proves to be more effective.

5.2. RANSAC

We selected the top 8 matching points from the dataset to compute the fundamental matrix, and then applied this matrix to all filtered matches to calculate the reprojection error for each match. The RANSAC threshold was determined by calculating the average reprojection error corresponding to the fundamental matrix obtained from all good matches. As shown in2, using RANSAC significantly reduced the reprojection error, with the BRISK method yielding the best results.

Method	ORB	SIFT	BRISK
Threshold	0.03	0.018	0.002
Error of F	35.4793	9123.73	86.8859
Error of F Ransac	33.2376	7.06162	0.507581
Inliers	21	14	6
average error of each match	0.232431	0.0433228	0.0034296

Table 2. Feature Matching Results for Different Methods

5.3. Disparity Computation

After rectification, we apply several key optimizations to enhance disparity estimation:

- **Gaussian Blur**: Reduce noise in both images.
- **Census Transform**: Compute an 8-bit descriptor per pixel for improved robustness.
- **Cost Volume Construction**: Build the cost volume using Hamming distances between Census descriptors.
- **Multi-directional Cost Aggregation**: Aggregate costs in eight directions with smoothness constraints (P1 and P2).
- **Winner-Take-All**: Select the disparity with the minimum cost for each pixel.

Figure 8 shows a comparison between a disparity map computed without these optimizations and our calibrated disparity result using SIFT.

These steps improve both accuracy and robustness for subsequent depth and 3D reconstruction.

(a) Disparity Map without Processing

(b) Our Calibrated Disparity (SIFT)

Figure 8. Comparison of disparity maps: (a) without processing, (b) our calibrated result using SIFT.

5.4. 3D Mesh Generation

Visually inspecting the generated 3D mesh from our disparity map reveals noticeable artifacts and irregularities 6a, indicating a lower overall quality. However, it is important to recognize that the mesh's quality is influenced not only by the mesh generation process itself but also by the accuracy of the input disparity map. Since our computed disparity map contains noise and inaccuracies, we want to isolate the effect of the mesh generation process by applying the same method to the ground truth disparity map provided in the dataset 6b.

As expected, the mesh derived from the ground truth disparity map shows significantly better structure and smoothness, reinforcing that much of the quality degradation in our generated mesh origins from errors in the disparity estimation rather than flaws in the mesh generation process 3.

Disp. Map	Laplacian Smoothness	Mean Edge Length
Our	1.2	0.0037
GT	9.83	0.0094

Table 3. Quantitative Results of the 3D mesh generated based on our and the ground truth disparity map. Scene "artroom1".

A higher Laplacian smoothness value indicates a smoother surface with fewer sharp artifacts. The ground truth-based mesh (9.83) is significantly smoother than our computed disparity-based mesh (1.2), suggesting that errors in disparity estimation introduce undesired roughness and discontinuities. The mean edge length is notably smaller (0.0037) for our disparity-based mesh compared to 0.0094 for the ground truth, indicating excessive short edges likely caused by noisy depth variations.

6. Conclusion

This work presented a complete stereo reconstruction pipeline using classical computer vision techniques. The pipeline encompassed key stages such as sparse keypoint matching, fundamental matrix estimation, stereo rectification, disparity map computation, and dense 3D reconstruction. Several optimizations, including modifications to the 8-point algorithm and disparity refinement, were introduced to enhance reconstruction accuracy.

The proposed method was evaluated on standard benchmark datasets, including Middlebury and ETH3D, using both qualitative and quantitative metrics. Experimental results demonstrated the pipeline's ability to generate 3D models from stereo images while maintaining a balance between computational efficiency and accuracy.

Despite the effectiveness of classical approaches, challenges such as handling textureless regions, occlusions, and computational efficiency in dense reconstruction remain. Future work may explore hybrid approaches that integrate deep learning-based disparity estimation with geometric constraints to further improve robustness and generalization.

References

[1] 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 *Pattern Recognition: 36th German Conference, GCPR 2014, Münster, Germany, September 2-5, 2014, Proceedings 36*, pages 31–42. Springer, 2014. 1, 3

[2] Thomas Schöps, Johannes L. Schönberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1

[3] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003. 1

[4] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.*, 60(2):91–110, 2004. 1

[5] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *Int. Conf. Comput. Vis. (ICCV)*, pages 2564–2571, 2011. 1

[6] H. Hirschmuller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 807–814, 2005. 1

[7] Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Tim Drummond. End-to-end learning of geometry and context for deep stereo regression. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 66–75, 2017. 1

[8] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 5410–5418, 2018. 1