Computer Vision

Project title: Stereo Camera System

University Of Burgundy
Master's in Computer Vision and Robotics

Submitted by: Muhammad Usama Javaid
Ahmed Khalil
Favour Eberechi Wobidi

Supervised by: Prof. Yohan Fougerolle

Date: April 30, 2025

Abstract

This project builds a stereo vision system using simple USB cameras to create 3D models. It captures image pairs, aligns them, and reconstructs a 3D point cloud through triangulation. The system follows nine key steps, from camera calibration to point cloud refinement. Tests show that good 3D reconstruction is possible even in less-than-ideal conditions, with room for future improvements..

1. Introduction

Stereo vision allows us to estimate depth by comparing two images taken from different viewpoints. This technique is widely used in areas like robotics, 3D mapping, and object detection. In this project, we develop a stereo camera system using low-cost USB cameras and a regular computer. The aim is to create a setup that is simple, portable, and easy to adapt for different uses.

The system follows a standard stereo vision pipeline, which includes steps like camera calibration, image alignment, and 3D point generation. Each step is chosen to balance performance and simplicity.
The system follows a typical stereo vision pipeline, with the following main steps:

1. Camera Calibration
2. Image Rectification
3. Disparity Map
4. Disparity Map with WLS filter
5. 3D dense reconstruction
6. Feature Matching
7. Epipolar Geometry Estimation
8. Triangulation
9. Point Cloud Post-Processing or 3D Sparse Reconstruction

The implementation is done in Python using the OpenCV and Open3D libraries. Each stage is chosen to keep the system efficient and functional.

2. Methodology

2.1 Camera Calibration

Camera calibration is the first and most important step in stereo vision. It helps us find out the camera's internal settings (called intrinsic parameters) and its position and orientation in space (called extrinsic parameters). The intrinsic parameters include things like the focal length, the center of the image (principal point), and lens distortion, which affects how straight lines may appear curved in photos. The extrinsic parameters tell us how the camera is placed and rotated in the real world.

In this project, we use **Zhang's method**, which is one of the most common calibration techniques. It works by taking several pictures of a flat pattern, like a checkerboard, from different angles. The method then compares the actual corner points in the image with where they are supposed to be and reduces the difference (called reprojection error). This way, it accurately finds the camera's settings and corrects any distortion, allowing us to make precise 3D measurements.

2.1.1 Challenges and Calibration Strategy

- Our first challenge during the system setup was the selection of an appropriate camera baseline. When the baseline was too narrow (3 cm), the resulting disparity between stereo images was minimal, yielding insufficient depth perception.
- Contrarily, wider baselines in the range of 17–24 cm introduced geometric inconsistency and increased susceptibility to epipolar alignment errors due to imperfect synchronization and lens distortion.
- Additionally, some of the tested USB cameras exhibited suboptimal intrinsic characteristics, including unstable focal lengths and uncalibrated digital zoom levels, leading to unreliable calibration outputs.
- After multiple trials with alternative cameras available in the laboratory, we achieved more stable calibration results by selecting a stereo pair with improved optical consistency.
- A baseline of approximately 12–15 cm was adopted, as it provided a practical trade-off between measurable depth range and feature correspondence robustness, consistent with typical human interpupillary distance standards used in stereo vision setups.
- Additionally, during the calibration process, we observed that the reprojection error was initially too large, indicating inaccuracies in parameter estimation.
- This was partly due to the lack of synchronization between the two cameras during image acquisition, which led to temporal misalignment in corresponding image pairs.
- To overcome this, we simultaneously captured images by manually triggering both cameras in coordination and numbering the image pairs systematically.
- A custom Python script was developed to open both cameras at the same time, capture synchronized images, and save them to a specific directory using a consistent naming convention.
- We took approximately 12-15 datasets for calibration.
- This approach yielded a more temporally consistent dataset of approximately 100 stereo image pairs, significantly improving the reliability of the calibration results.
- Another major issue that significantly hampered our progress was the poor image quality caused by inadequate ambient lighting conditions in the laboratory.
- The overhead fluorescent lights introduced severe glare and unpredictable shadows, while also adding noise that drastically reduced image contrast.

- As a result, the feature detection and matching process became extremely unreliable, forcing repeated recalibration attempts and wasting valuable time.
- At one point, the lighting conditions became the single biggest obstacle to moving forward with the system pipeline.
- To address this, we had to carefully redesign the imaging environment.
- All interfering ambient light sources were eliminated, and a controlled lighting setup was installed using uniform, diffuse illumination to ensure even exposure across the entire field of view.
- This adjustment made a remarkable difference — not only did it improve the sharpness and contrast of the captured images, but it also restored the reliability of feature detection and matching, allowing the project to progress smoothly beyond this critical stage.

2.2 Stereo Rectification

Stereo rectification is the process of transforming a pair of images so that the epipolar lines become aligned and parallel, simplifying the stereo correspondence problem to a one-dimensional search. Given the intrinsic and extrinsic parameters from stereo calibration, rectification computes a pair of homographies that reproject both images onto a common image plane where corresponding points lie on the same horizontal line.

This step is crucial for accurate disparity computation and subsequent 3D reconstruction. It relies heavily on the accuracy of the estimated camera parameters, particularly the rotation and translation between the stereo pair, as well as effective distortion correction.

2.2.1 Challenges and Rectification Strategy

During the stereo calibration and rectification process, we encountered significant image distortion, as well as rotation and translational misalignments between the stereo camera views. These errors stemmed from imprecise extrinsic parameter estimation, further aggravated by minor mechanical inconsistencies in the physical camera mounts and initial calibration errors.

To address these issues, we applied lens undistortion using the distortion coefficients obtained during calibration. This preprocessing step effectively corrected radial and tangential distortions in both images, leading to a more accurate rectification outcome.

However, due to persistent instability in the estimated rotation matrix—likely due to subpixel-level noise and image quality limitations—we assumed an identity matrix for rotation during the rectification stage. While this assumption simplifies the geometry and imposes an idealized alignment of the image planes, it was empirically found to improve the consistency of epipolar alignment in our rectified image pairs.

Overall, the combination of image undistortion and constrained rectification provided sufficient alignment to enable reliable feature matching and disparity estimation in subsequent stages.

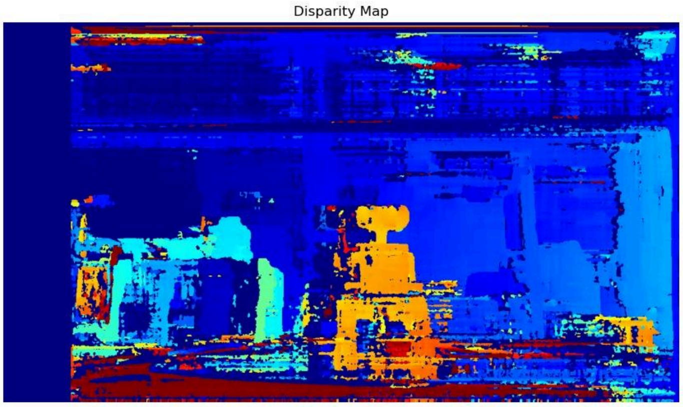
2.3 Disparity Depth Mapping

Disparity mapping is a critical step in stereo vision that involves computing the pixel-wise horizontal displacement between corresponding points in a rectified stereo image pair. The disparity ddd is inversely proportional to the depth Z of a scene point, given by the relation:

$$Z = \frac{f \cdot B}{d}$$

where *f* is the focal length of the camera and *B* is the baseline (distance between the two cameras). Accurate disparity estimation enables recovery of the 3D structure of the scene via triangulation.

Semi-Global Block Matching (SGBM) is a robust and widely used algorithm for dense disparity estimation. It aggregates matching costs over multiple paths to improve accuracy and suppress noise while preserving object boundaries. However, the quality of the disparity map is highly dependent on prior steps such as rectification and image quality.



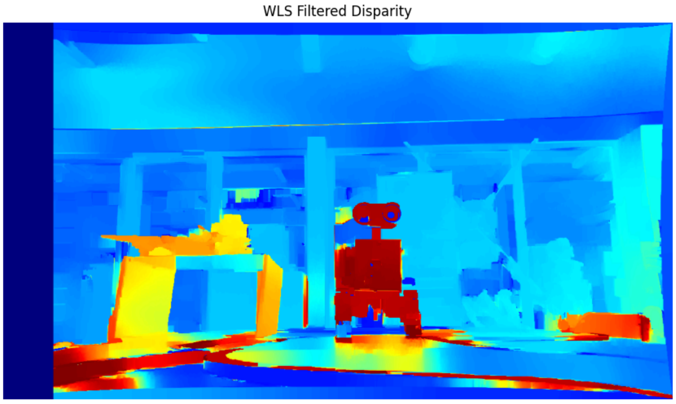
2.3.1 Challenges and Refinement Strategy

Our initial attempts at disparity computation were hindered by the poor quality of stereo rectification. Misaligned epipolar lines introduced false matches and artifacts in the disparity map, resulting in depth inconsistencies and invalid reconstructions.

To mitigate these issues, we carefully tuned the parameters of the **StereoSGBM** algorithm, including `minDisparity`, `numDisparities`, `blockSize`, and regularization terms such as `P1` and `P2`. These adjustments helped balance the trade-off between smoothness and edge preservation in the disparity output.

After generating the raw disparity map, we applied **post-processing** and **normalization** to enhance the dynamic range and improve interpretability. To further refine the results, we utilized a **Weighted Least Squares (WLS) filter**, which effectively suppressed noise and preserved sharp transitions at object boundaries. The WLS filter uses the left image as guidance, promoting spatial coherence in homogeneous regions while maintaining discontinuities.

The integration of WLS filtering significantly improved the quality of the disparity visualization and directly enhanced the accuracy and realism of the 3D point cloud generated in the reconstruction stage.

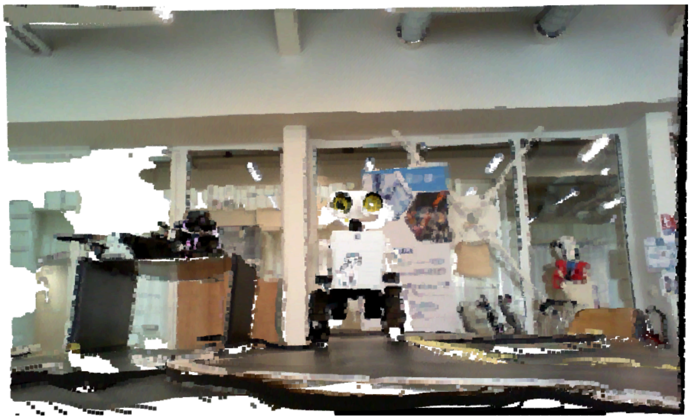


2.4 3D reconstruction

3D reconstruction is the process of creating a three-dimensional model of a scene or object from two or more 2D images taken from different viewpoints. In stereo vision, this involves computing depth from disparity between image pairs and generating a 3D point cloud representing the scene's geometry.

2.4.1 Challenges

- Incorrect Q matrix caused distorted 3D output.
- Open3D crashed without proper downsampling.
- Misaligned color data led to odd point cloud visuals.
- Noisy disparity introduced outliers in 3D reconstruction.



2.5 Feature Matching

Feature detection and matching are fundamental in stereo vision for establishing correspondences between two images. These correspondences are essential for estimating stereo geometry and performing accurate triangulation for 3D reconstruction.

In this project, we leveraged a variety of feature detectors and descriptors—including ORB, SIFT, AKAZE, and BRIEF—in combination with matching algorithms like Brute-Force (BF) matcher and FLANN-based matcher. All matching was conducted on rectified image pairs to ensure alignment along epipolar lines, which reduces the search space and improves matching reliability.

2.5.1 Experimental Evaluation of Feature Matchers

We evaluated multiple combinations of feature detectors and matchers, comparing their performance based on the number of keypoints, the ratio of valid matches, and their robustness to noise and image quality. Below are summarized results for each configuration:

ORB + Brute-Force Matcher

ORB (Oriented FAST and Rotated BRIEF) is a binary descriptor known for its speed and rotation invariance.

The Brute-Force matcher computes the Hamming distance between binary descriptors and returns the best matches.

Initial matches: 2082

Inlier matches after RANSAC: 1557

Valid 3D points: 1377 (88.4%)

This combination proved effective in balancing speed and accuracy, making it suitable for real-time applications.

SIFT + FLANN

SIFT (Scale-Invariant Feature Transform) is robust to scale and illumination changes, while FLANN (Fast Library for Approximate Nearest Neighbors) accelerates matching for high-dimensional descriptors.

Keypoints detected – Left: 2421, Right: 2646

Total matches (KNN pairs): 2421

Good matches: 744

Bad matches: 1677

Despite its robustness, the large number of outliers made this method less efficient without further filtering.

AKAZE + Brute-Force Matcher

AKAZE is a nonlinear scale-space detector optimized for speed.

Keypoints detected – Left: 2276, Right: 2269

Good matches: 814

Bad matches: 146

AKAZE offered a strong trade-off between robustness and computational cost.

BRIEF + Brute-Force Matcher

BRIEF (Binary Robust Independent Elementary Features) is a simple yet fast descriptor, but it is not invariant to rotation or scale.

Keypoints detected – Left: 3262, Right: 3506

Good matches: 1202

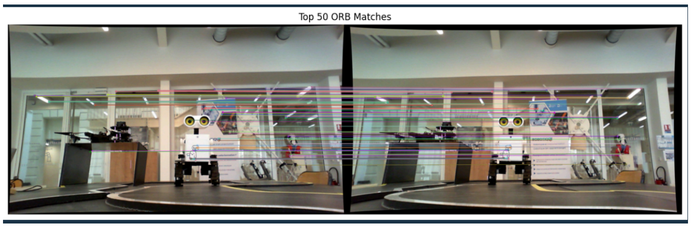
Bad matches: 2060

While BRIEF generated many keypoints, its lack of geometric invariance led to a high number of poor matches.

Final Choice and Rationale

After empirical comparison, we selected **ORB + Brute-Force matcher** as our primary feature matching strategy. ORB provided sufficient robustness and efficiency, while the Brute-Force matcher with Hamming distance allowed us to accurately match binary descriptors.

Post-processing with **RANSAC (Random Sample Consensus)** was used to eliminate outliers and ensure geometric consistency in the matches. This step refined the matches by enforcing epipolar constraints, which is crucial for stereo geometry estimation and 3D triangulation.



Epipolar Geometry

Understanding epipolar geometry is critical for reconstructing 3D scenes from stereo images. It describes the intrinsic projective geometry between two views and is governed by the fundamental matrix (F), which relates corresponding points between two images via the epipolar constraint:

$$x'^T F x = 0$$

Where x and x' are homogeneous coordinates of corresponding points in the left and right images, respectively.

2.5.2 Fundamental Matrix Estimation

To compute the fundamental matrix, we first extracted and matched features between rectified stereo images. We then applied the **eight-point algorithm** and **RANSAC** (Random Sample Consensus) to robustly estimate the fundamental matrix while eliminating outlier correspondences. RANSAC ensures that only geometrically consistent matches contribute to the estimation, which is crucial for accuracy.

The estimated fundamental matrix encodes both the epipolar lines and the epipoles, allowing us to constrain search lines for correspondence to 1D. This step significantly reduces the complexity of matching and prepares the input for essential matrix computation and triangulation.

2.5.3 Visual Validation

We verified the correctness of the fundamental matrix by overlaying the computed epipolar lines on the image pairs. If the matrix is estimated accurately, the corresponding point in one image should lie on the epipolar line in the other. In our case, we observed consistent alignment, especially after improving feature matching and calibration, confirming that the fundamental geometry was correctly recovered.

2.6 Triangulation

Triangulation is the process of determining the 3D location of a point by intersecting two rays cast from calibrated camera centers through corresponding image points. Given accurate stereo calibration and matched feature points, triangulation mathematically estimates the 3D coordinates of those points in space.

We used the projection matrices P and P' obtained from the calibration step, corresponding to the left and right cameras, respectively. Each pair of matched points x and x' was then used to solve for the 3D point X that projects back to the image points:

$$x = PX, \quad x' = P'X$$

To solve this, we employed the **linear least squares method** via OpenCV's `cv2.triangulatePoints()` function. The function constructs a system of equations based on the projection constraints and solves for the homogeneous coordinates of the 3D point.

2.6.1 Challenges Encountered

- Inaccurate rectification and calibration in earlier stages introduced projection inconsistencies, which initially affected triangulation reliability.
- Synchronization between the stereo pair was essential. We ensured simultaneous image capture and sequential labeling, resulting in a dataset of 100 well-aligned stereo pairs.
- Errors from incorrect matches were mitigated using RANSAC, ensuring only inlier correspondences were used for triangulation.

This step served as a foundation for building the sparse 3D reconstruction, where each triangulated point corresponds to a valid image pair match.

2.7 3D Point Cloud Post-Processing and Sparse Reconstruction

After obtaining 3D coordinates through triangulation, we proceeded to build a sparse 3D point cloud, a spatial representation of the scene composed of reconstructed feature points. This step aimed to verify whether the triangulated features could spatially replicate the original scene structure.

2.7.1 Methodology

We used the matched feature points from the ORB detector and BF matcher, filtered by RANSAC, as input for triangulation. The resulting 3D points were projected into a 3D coordinate system using the known stereo camera projection matrices. The visualization of these points was intended to yield a sparse reconstruction (or point cloud) of the observed scene.

2.7.2 Challenges

The initial point cloud appeared blank and lacked meaningful structure. This was primarily due to:

- Poor disparity estimation from earlier rectification errors.
- Inconsistent depth values derived from feature matches, which led to invalid or zero-valued 3D points.

2.7.3 Resolution Strategy

Upon reviewing the pipeline, we recognized that the matched image points came from the rectified images but lacked corresponding calibrated depth information. To resolve this:

- We reprojected the key points from the **original, calibrated stereo image pair** (before rectification).
- We ensured that these images were geometrically aligned with their camera parameters and used them to re-estimate depth accurately.

By integrating depth perception from the calibrated image pair with the previously matched features, we successfully generated a sparse yet valid point cloud that preserved the spatial configuration of the original scene.

3. Conclusion and Future Improvements

This project demonstrated how to create a simple stereo vision system using two USB cameras to reconstruct 3D scenes. The process included several key steps: camera calibration, image rectification, feature detection and matching, estimating the relationship between images (epipolar geometry), triangulation, generating depth maps, and building a 3D point cloud.

We faced several challenges, such as low-quality images, poor lighting, and difficulty with camera alignment. However, through repeated testing and careful adjustments, we improved the setup and achieved meaningful results. By adjusting parameters and applying filtering techniques like WLS, we improved the quality of our disparity maps, which led to better 3D reconstructions.

We followed reliable methods such as Zhang's camera calibration technique, Semi-Global Block Matching (SGBM) for depth estimation, and RANSAC for filtering out bad matches. These helped us get more accurate results. Although the final point cloud was sparse, it still showed the general shape of the scene.

Future Improvements

- Automatically capturing synchronized images.
- Using dense 3D reconstruction methods.
- Better handling of lighting conditions.

Making the system work on other platforms like Linux or macOS.

Overall, the system we built is flexible, educational, and useful for learning how stereo vision works in practice.

References

1. Zhang, Z. (2000). A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11), 1330–1334.

2. Hartley, R., & Zisserman, A. (2004). *Multiple View Geometry in Computer Vision*. Cambridge University Press.

3. Bradski, G. (2000). The OpenCV Library. *Dr. Dobb's Journal of Software Tools.*

4. Hirschmüller, H. (2008). Stereo processing by semi-global matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2), 328–341.

5. Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF. In *Proc. IEEE ICCV*.

6. Fischler, M. A., & Bolles, R. C. (1981). Random Sample Consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6), 381–395.