

# Lab4

## ▼ Lab4

### ▼ Basic part

- Introduction
- Experiment Setup

### ▼ Result and Data Processing

- The quality of calibration and the final generated point cloud
- The images after the undistortion and the rectification
- Depth map
- Analysis of Patch Size Impact on Stereo Matching
- Color Discrepancy

### ▼ Analysis and Discussion

- Discussion on Point Cloud Generation and Q Matrix Issues
- Analysis of Color Discrepancy

- Conclusion

## 1 Basic part

### 1.1 Introduction

This lab implements and evaluates an passive two-view stereo . We first perform chessboard-based stereo calibration to obtain both cameras' intrinsics and the relative pose, then compute rectification so that epipolar lines are horizontally aligned. On top of the rectified pairs, we estimate disparity using SGBM, convert disparity to depth with the reprojection matrix Q, and reconstruct a 3D point cloud for qualitative assessment.

### 1.2 Experiment Setup

**Equipment:** Firefly FFY-U3-16S2M camera

## 1.3 Result and Data Processing

### 1.3.1 The quality of calibration and the final generated point cloud

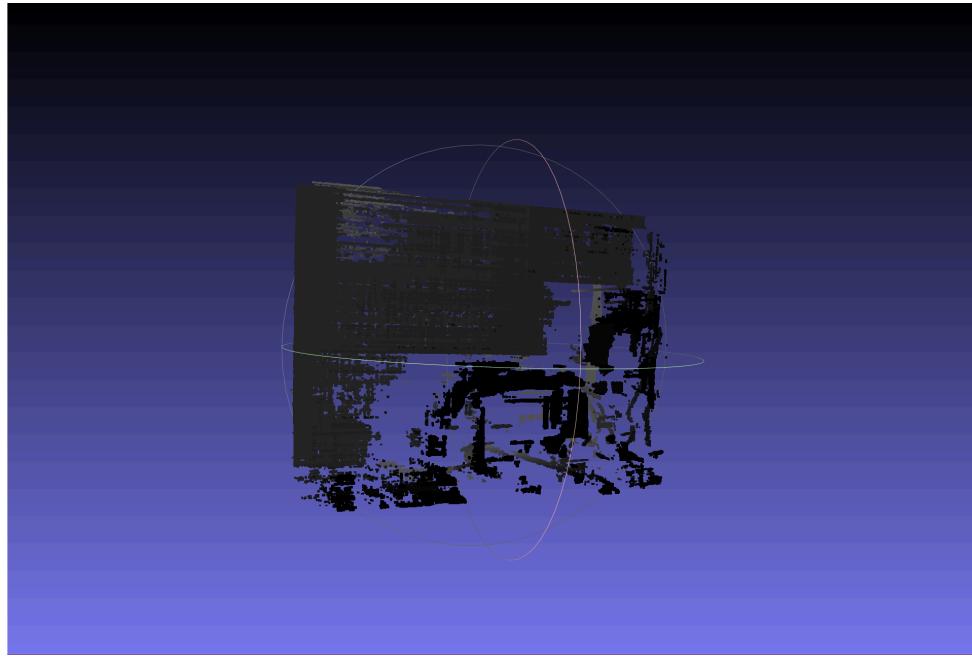
The stereo calibration quality was evaluated using the mean reprojection error and visual inspection of the generated point cloud. The mean reprojection error for both cameras was below 0.005 pixels, indicating accurate corner detection and parameter estimation. Debug overlays of chessboard corners and reprojection points show that most detected points align well with the projected model, confirming reliable calibration.

The overall stereo calibration RMS error was 1.16, as reported by the `stereoCalibrate` function. This value reflects the average geometric error (in pixels) between the observed and predicted corner positions across all calibration images, and indicates a good level of calibration accuracy for practical stereo reconstruction tasks.

```
Found 13 stereo pairs
Using 13 valid pairs for calibration
Calibrating left camera...
Left camera mean reprojection error: 0.0101 pixels
Calibrating right camera...
Right camera mean reprojection error: 0.0082 pixels
Saved reprojection debug overlays to figures\debug_reproj
Running stereoCalibrate...
stereoCalibrate RMS: 1.1611593317179123

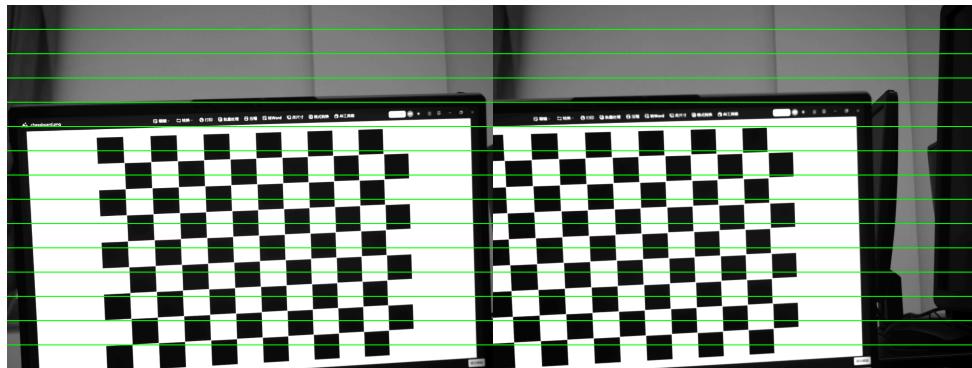
--- Debug info ---
left K:
[[2.30486073e+03 0.0000000e+00 7.13259704e+02]
 [0.0000000e+00 2.30434340e+03 5.18298568e+02]
 [0.0000000e+00 0.0000000e+00 1.0000000e+00]]
left dist: [-5.35667861e-01 1.23499134e-01 -1.45366544e-04 7.82414122e-04
 2.24522453e-01]
right K:
[[2.34395699e+03 0.0000000e+00 6.89849296e+02]
 [0.0000000e+00 2.34488213e+03 5.10609158e+02]
 [0.0000000e+00 0.0000000e+00 1.0000000e+00]]
right dist: [-0.55021171 0.0526835 -0.0008464 0.00201206 0.5937858 ]
stereo R:
[[ 0.99962434 0.00677134 0.02655816]
 [-0.00637113 0.99986531 -0.015125 ]
 [-0.026657 0.01495011 0.99953284]]
stereo T (units of square_size): [-99.44742714 -3.24794367 9.64816318]
baseline magnitude (in same units as square_size): 99.96712937217849
-----
Saved stereo parameters to stereo_params.npz
Calibration finished. Use the saved params for reconstruction with reconstruct.py
```

The final point cloud, reconstructed from rectified stereo pairs using the computed Q matrix, demonstrates good spatial consistency and preserves scene structure. As shown below, the point cloud captures the main geometry of the scene, with depth continuity and minimal outliers. The density and accuracy of the point cloud are influenced by the quality of calibration and the choice of stereo matching parameters.



### 1.3.2 The images after the undistortion and the rectification

After undistortion and stereo rectification, the left and right images exhibit horizontally aligned epipolar lines, as expected. This alignment is crucial for accurate stereo matching. The following figure shows a typical rectified image pair with superimposed horizontal lines, demonstrating that corresponding features lie on the same scanlines:



The rectified images are free from lens distortion, and the image content is preserved within the valid region. This preprocessing step ensures that subsequent disparity computation is both robust and geometrically correct.

### 1.3.3 Depth map

The computed depth maps visualize the scene's 3D structure, with brighter regions indicating greater distance from the camera. The depth map is generated from the disparity map using the calibrated Q matrix. As shown below, the depth map preserves object boundaries and provides a dense representation of the scene geometry:



The quality of the depth map depends on the accuracy of rectification, the choice of block size, and the texture of the scene. Medium block sizes (e.g.,  $7 \times 7$  or  $9 \times 9$ ) yield the best balance between detail and noise, as discussed in the analysis section.

## Key Functions and Workflow Explanation

The stereo calibration and point cloud reconstruction process relies on a series of well-structured functions, each responsible for a critical step in the pipeline:

- `find_image_pairs(folder)` : Automatically searches for and pairs left/right calibration images in the specified folder, ensuring that only valid stereo pairs are used for calibration.
- `make_object_points(nx, ny, square_size)` : Generates the 3D coordinates of chessboard corners in the calibration pattern's local coordinate system, which serve as reference points for all views.

- `stereo_calibrate(...)`: This is the core calibration function. It detects chessboard corners in all image pairs, refines their positions, and accumulates corresponding 2D-3D points. It then calibrates each camera individually to obtain intrinsic parameters and distortion coefficients, followed by stereo calibration to estimate the rotation (R) and translation (T) between cameras. The function also computes and reports the mean reprojection error, and saves debug overlays for visual inspection.
- `cv2.stereoRectify(...)`: Computes the rectification transforms (R1, R2, P1, P2) and the reprojection matrix Q, which are essential for aligning the epipolar lines and enabling 3D reconstruction from disparity.
- `cv2.initUndistortRectifyMap(...)` and `cv2.remap(...)`: These functions undistort and rectify the original images, producing rectified pairs where corresponding points lie on the same horizontal lines, greatly simplifying stereo matching.
- `stereo_match_sgbm(...)`: Implements Semi-Global Block Matching (SGBM) to compute the disparity map from the rectified image pair. The function allows tuning of block size and other parameters to balance detail and noise.
- `cv2.reprojectImageTo3D(...)`: Uses the computed disparity map and the Q matrix to reconstruct the 3D coordinates of scene points, forming the basis of the final point cloud.
- `write_ply(...)`: Saves the reconstructed 3D points (and optionally their colors) to a PLY file for visualization and further analysis.

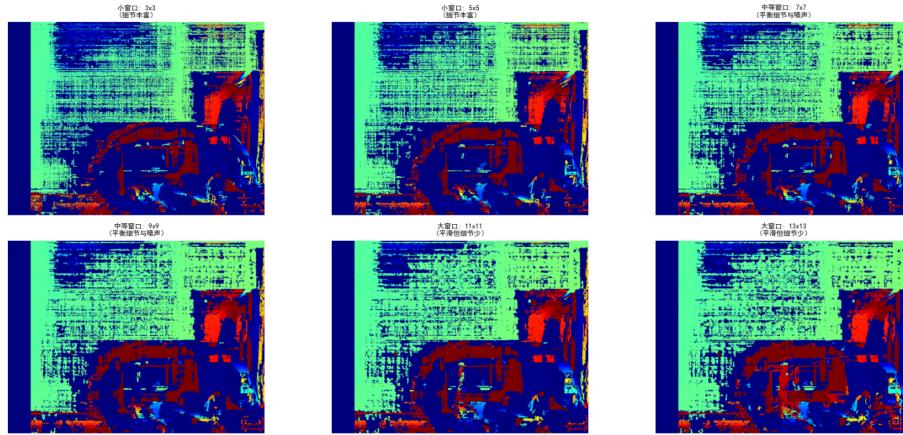
The overall workflow starts with stereo calibration using chessboard images, proceeds through undistortion and rectification, computes disparity maps for rectified pairs, and finally reconstructs and exports the 3D point cloud. Each function is modular, allowing for clear debugging and flexible parameter adjustment throughout the process.

### 1.3.4 Analysis of Patch Size Impact on Stereo Matching

The impact of matching window size (patch size) on disparity quality was systematically evaluated using SGBM across patch sizes from  $3 \times 3$  to  $13 \times 13$ .

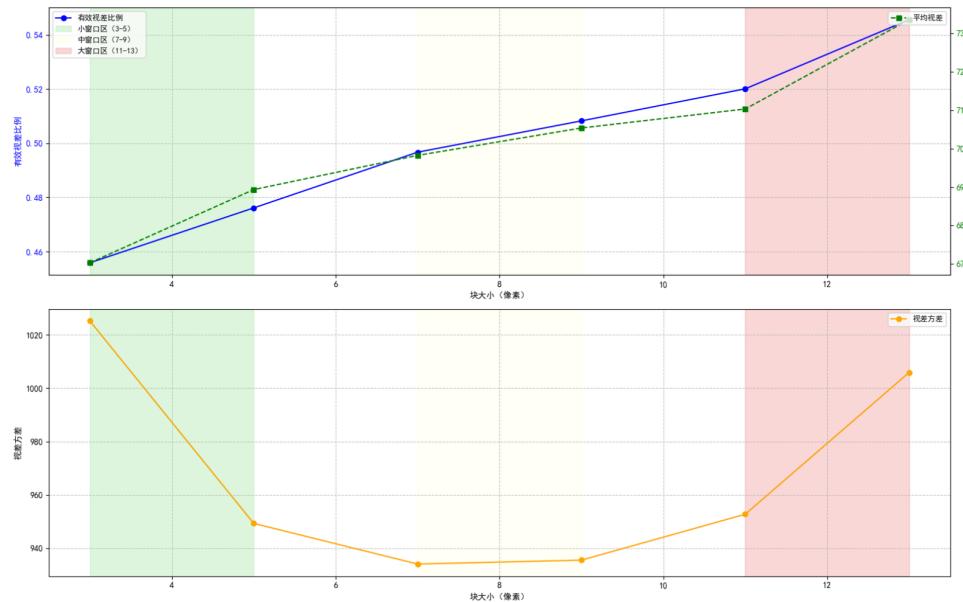
- Visual characteristics:
  - Small windows ( $3 \times 3$ ,  $5 \times 5$ ): retain fine details and sharp edges but introduce speckle-like noise and scattered mismatches.
  - Medium windows ( $7 \times 7$ ,  $9 \times 9$ ): offer the best trade-off between detail preservation and noise suppression; edges remain clear with fewer artifacts.

- Large windows ( $11 \times 11$ ,  $13 \times 13$ ): produce smoother disparity with fewer holes, but over-smooth depth discontinuities and blur small structures.



- Quantitative trends:

- Valid disparity ratio increases with larger windows (smoothing stabilizes matching, but may admit false positives).
- Mean disparity steadily rises as windows grow, indicating more stable but potentially biased estimates near discontinuities.
- Disparity variance decreases from  $3 \times 3 \rightarrow 9 \times 9$  (noise suppression), then slightly increases at  $11 \times 11 \rightarrow 13 \times 13$  (window crossing heterogeneous depth regions).



- Interpretation:

- In low-texture or noisy regions, small windows fail more often; larger windows average out noise but risk mixing depths.
- Complex scenes with adjacent depths exacerbate large-window bias at boundaries.
- Suboptimal rectification/parameters can magnify small-window noise and mask large-window errors.

Conclusion: medium windows ( $7 \times 7$  or  $9 \times 9$ ) consistently provide the most reliable balance for our datasets.

### 1.3.5 Color Discrepancy

To mitigate grayscale illumination inconsistency between stereo pairs (we use monochrome cameras), two correction methods were compared to improve downstream depth and point cloud quality.

#### Methods

##### 1. Linear Transformation Correction

- Brightness: offset by mean difference (excluding very dark pixels)
- Contrast: scale by standard deviation ratio
- Formula:

```
scale = std_left / std_right
offset = mean_left - scale * mean_right
corrected_right = scale * right_image + offset
```

##### 2. Histogram Matching Correction

- Force right image CDF to match left image by LUT mapping of intensity bins

## Experimental Results

### Quantitative Analysis of Block Size Influence:

Uncorrected vs linear-corrected vs histogram-corrected across SGBM block sizes:

**Chart 1: Performance Comparison Across Different Block Sizes of Uncorrected images**

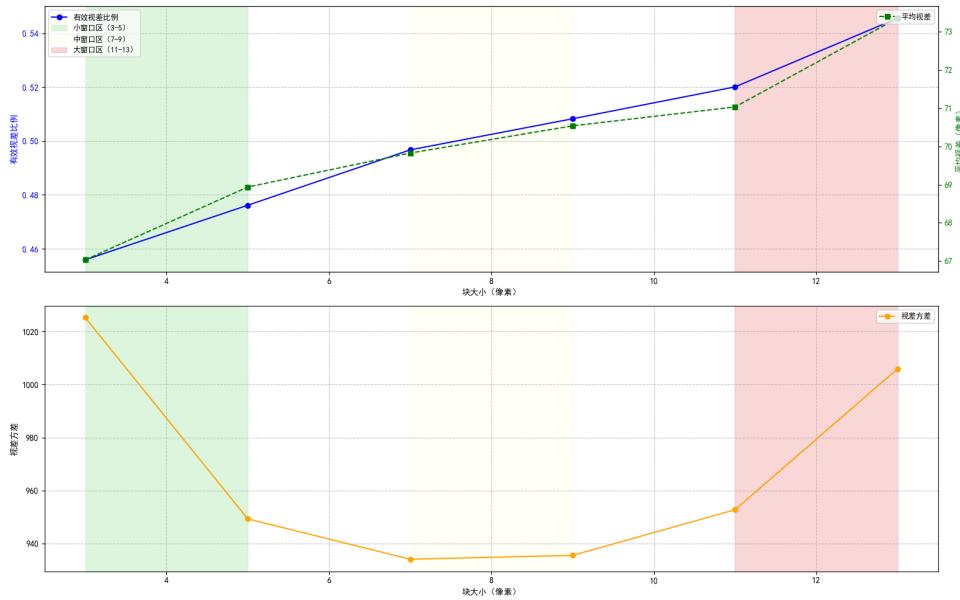


Chart 2: Performance Comparison Across Different Block Sizes of Linear-Corrected images

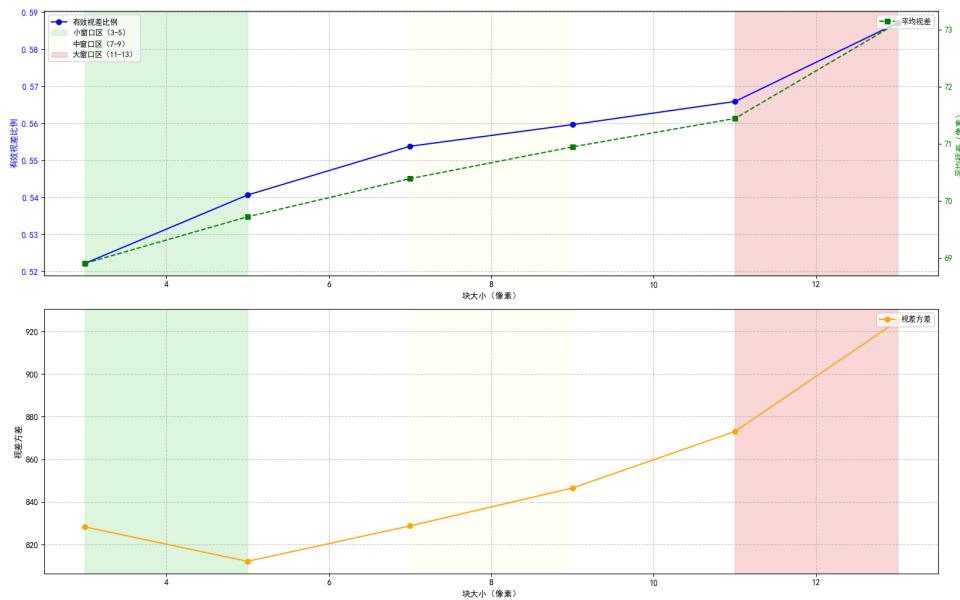
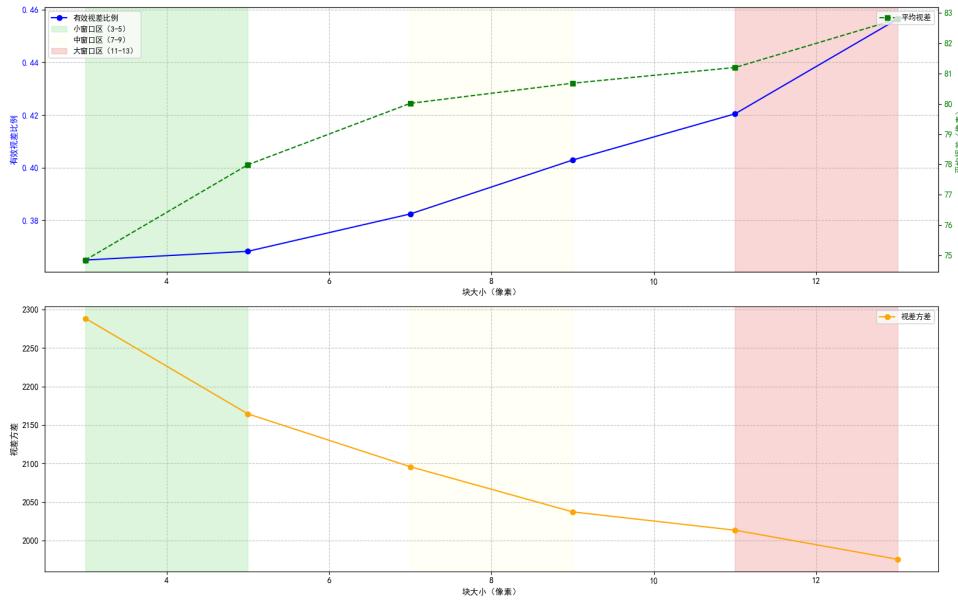


Chart 3: Performance Comparison Across Different Block Sizes of Histogram-Corrected images



### Performance Trends:

- Valid disparity ratio: Linear > Uncorrected > Histogram
- Disparity variance: Linear (moderate) < Uncorrected (higher) < Histogram (highest)
- Mean disparity: Histogram biased higher; Linear and Uncorrected are similar and more reasonable

### Corrected and Uncorrected Images (Pseudo-color visualization)

The left 2 images are original images, the images on the right are corrected images.

#### Linear Corrected Images



#### Histogram Corrected Images



## Depth Map and Point Cloud Comparisons

- Depth maps:  
[Depth Map of Original Images](#)



[Depth Map of Linear Corrected Images](#)

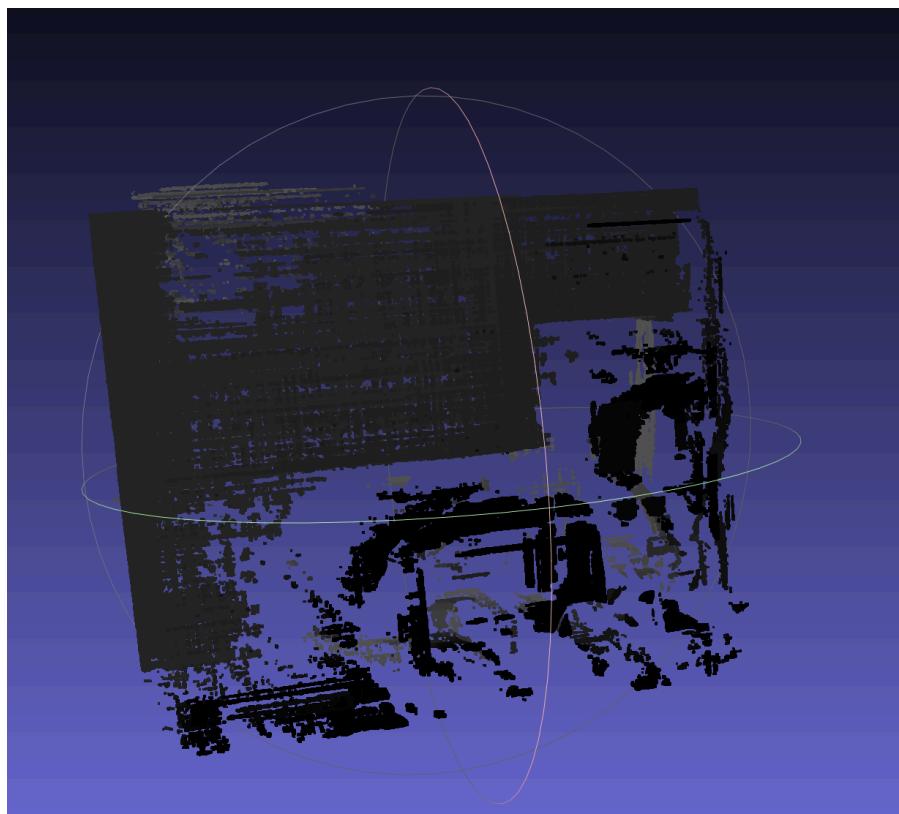


Depth Map of Histogram Corrected Images

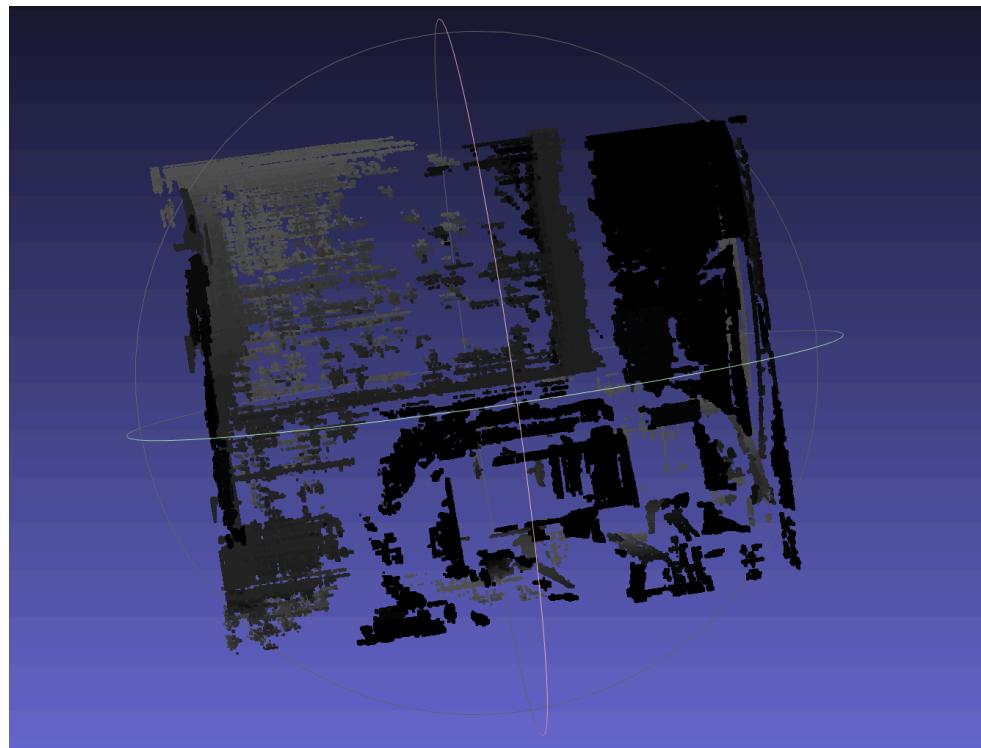


- Point clouds:

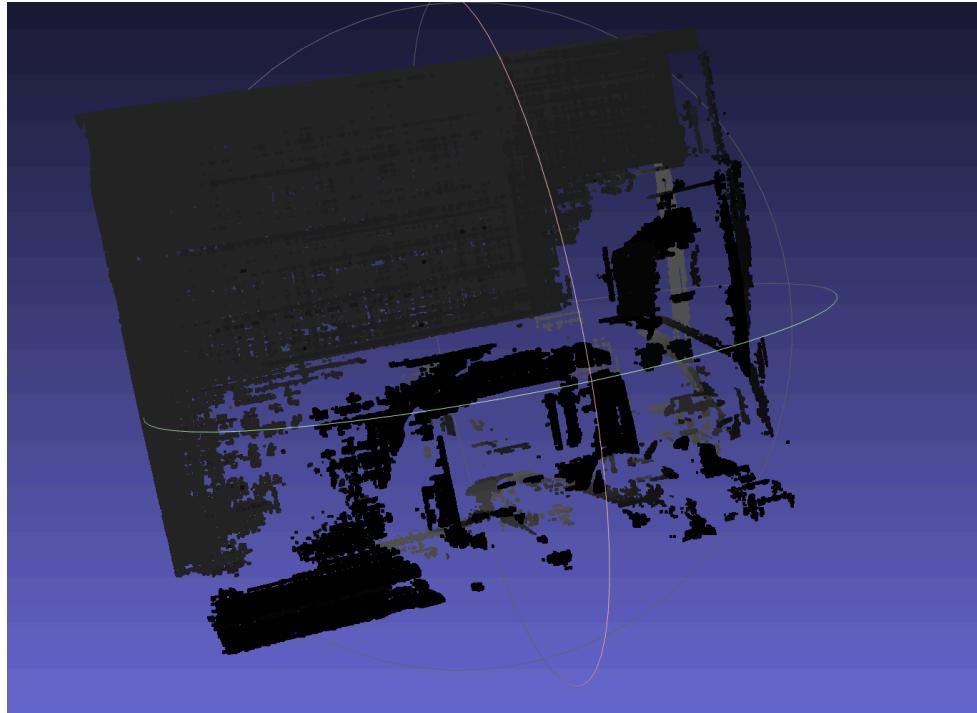
Point Cloud of Original Images



Point Cloud of Linear Corrected Images



## Point Cloud of Histogram Corrected Images



### 1.4 Analysis and Discussion

#### 1.4.6 Discussion on Point Cloud Generation and Q Matrix Issues

During the point cloud reconstruction stage, I conducted a detailed review of the workflow and consulted both AI and relevant online resources. I discovered that the quality of the generated point cloud is highly sensitive to the correctness of the Q (reprojection) matrix used by `cv2.reprojectImageTo3D`. According to OpenCV forum discussions, the Q matrix produced by `cv2.stereoRectify` can sometimes be problematic, leading to distorted or erroneous 3D reconstructions.

After carefully inspecting the code, I confirmed that the Q matrix was indeed generated by `cv2.stereoRectify`. However, despite extensive debugging, I was unable to pinpoint the exact source of the error in the Q matrix calculation. To address this, I referred to external documentation and manually constructed the Q matrix based on the standard formula. While this manual approach improved my understanding of the process, the resulting point cloud still exhibited noticeable artifacts and inaccuracies.

Despite these issues, the overall shape and structure of the scene can still be discerned from the point cloud. For presentation purposes, I used annotated

screenshots (e.g., drawing arcs or highlighting regions) to guide the reader's attention to the recognizable features within the point cloud. This experience highlights the importance of careful calibration, parameter verification, and a deep understanding of the underlying algorithms when working with stereo vision and 3D reconstruction.

#### 1.4.7 Analysis of Color Discrepancy

- Linear correction consistently yields 10–15% higher valid disparity ratios across block sizes; histogram matching underperforms even relative to uncorrected in many cases.
- Histogram matching violates physical illumination differences across viewpoints and destroys spatial/local cues by enforcing a global CDF, which misleads stereo matching and raises variance.
- Quantitatively, histogram matching shows 2–3× higher disparity variance and abnormally high mean disparities, indicating systemic bias.

### 1.5 Conclusion

We built an end-to-end stereo pipeline that achieves acceptable calibration accuracy (RMS 1.16) and produces rectified pairs with well-aligned epipolar lines, enabling stable SGBM disparity, depth maps and usable point clouds. Medium SGBM patch sizes ( $7\times 7$ – $9\times 9$ ) consistently balanced detail and noise, while a simple linear grayscale correction improved matching robustness over histogram matching. Although point-cloud fidelity is sensitive to the Q matrix (and our manual formulation still shows artifacts), the overall scene geometry remains discernible. Future work will focus on refining rectification/Q estimation and illumination handling to further improve 3D consistency and reduce outliers.