3D imaging stereometry with various interaxial distances on 2D images of a Rubik's cube and Tsai grid

David D. Daffon,* Justine A. Elvena, Jei Aldrindael B. Gamboa, and Walter S. Go

National Institute of Physics, UP Diliman
*Corresponding author: dddaffon@up.edu.ph

Abstract

This paper explored 3D imaging using stereometry for calculating depth of points in an image. Using an iPhone camera, multiple images of a Tsai grid and a Rubik's cube are captured at different camera transverse distances and geometric formulas are applied to estimate the depth. Additionally, we investigated the effect of varying the interaxial distances on the depth calculation and 3D surface reconstruction. The general shape and positioning of the objects were recovered from the 3D reconstructions, but the calculated depths varied from actual distances. Possible sources of error include the sensitivity of depth estimation to pixel disparities of points and small miasalignments in capturing images. Despite these limitations, the experiment demonstrated the ability to recover 3D information from the object's 2D images.

Keywords: stereometry, 3D reconstruction

1 Introduction

Images are often used to capture moments and store information. However, the three-dimensional nature of objects are lost when taking photographs. Imaging techniques may be used to recover the depth and shape of the recorded data [1]. In this paper, we use 3D imaging stereometry to reconstruct depth and 3D perspectives from flat 2D images.

3D imaging stereometry works as follows. Consider an object point located at an unknown axial distance Z from the camera lens. In the image plane, the point appears at a transverse distance x_1 from the optical axis of the camera. After translating the camera transversely by a distance b, the same point will be recorded at a new position x_2. Using geometry of similar triangles, the following ratios are obtained [1]

$$\frac{x_1}{f} = \frac{L}{Z}, \qquad \frac{x_2}{f} = \frac{L-b}{Z}. \tag{1}$$

Where L is the horizontal distance of the object from the axial axis of the camera's initial position and f is the camera's focal length. Using (1) we solve for Z such that

$$Z = \frac{bf}{x_1 - x_2}, \tag{2}$$

which if done for various points on the object allows us to recover the 3D information and reconstruct the 3D surface [2].

For this experiment, we implement 3D imaging stereometry to estimate depth from images of a Tsai grid, a box, and a Rubik's cube. We focus on how varying the camera position b affects the estimation of depth and quality of 3D reconstructions. Post-processing methods are also explored to better profile and characterize the reconstructed objects, especially the distinct patterns of the Rubik's cube and Tsai grid.

2 Methodology

2.1 Experimental Setup

We set up the experiment by creating a "T-square" with the available rulers so that we may obtain measurements for both the z distance and the interaxial distance, b. The rulers were positioned such that the zero for the 24-inch ruler is on the right-hand side of the figure, while the zero for the 12-inch ruler is on the top side of Figure 1. The objects used are a Tsai grid that is constructed using a chess mat and Funko POP box, and a Rubik's cube with one layer turned at an angle to create more depth. For the Tsai Grid setup, we placed the corner of the Funko POP box on the end of the 24-inch ruler opposite the

12-inch ruler as shown in Fig. 1. Meanwhile, in the Rubik's cube setup the corner of the Rubik's cube is placed at the 12-inch mark on the ruler. These gave us a reference z distance for both objects.

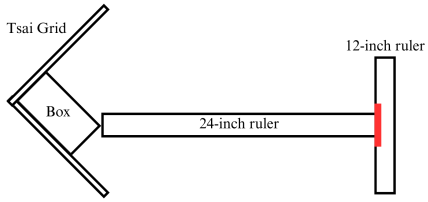


Figure 1: Picture-taking setup. The red box represents the position of the camera phone.

For imaging, we used an iPhone 13 mini set across the 12-inch ruler with various distances from its top (according to the diagram) side acting as our reference for b. To fix the camera in an upright position, another box is used for vertical alignment.

2.2 Programmatic Analysis with Python

We used Python and the open-source image processing library OpenCV for the programmatic analysis and 3D reconstruction of the images. OpenCV was used to pinpoint corners on the objects, which were converted to 2D Numpy arrays. We used equation (2) to create a z-finding function, wherein we fed the 2D Numpy arrays, resulting in a distance z *per corner*. Lastly, using Matplotlib 3D plots, we used complete spatial information x, y, z to reconstruct our objects. (View the code here.)

3 Results and Discussion

3.1 Tsai Grid Setup

We successfully calculated the depth of the Tsai grid setup and reconstructed it using a 3D plot as seen in the Figure 2. This result is for the interaxial distance $b = 3$ in. Multiple views of the reconstruction are shown in the figures, displayed beside the actual setup for comparison. The top of the reconstructed chess mat shows a triangular plane that fills in the gap between its sides. This feature does not exist in the actual setup, and it appears in the plots only because the points are interpolated. The red dots correspond to the same points in the plot and the actual setup. In the plots, key elements of the actual setup, such as the chess mat being opened at some angle and the 3D geometry of a box sandwiched between it, are discernible. Similar plots were obtained for other various values of b in the stereometry reconstruction. The overall scaling and shape is preserved across all the plots made. However, the relative positioning of points varied. In the actual setup, the corners of the chess mat were at equal distances from the plane of the camera, but in the reconstructions one side often appeared closer. Despite this discrepancy, the relative placement of the box remained consistent, positioned between the extreme z values, while the farthest point from the camera was consistently the folded midsection part of the chess mat.

Table 1: Measured z-positions compared with actual values at different b settings (part 1).

Point	Actual z (in.)	b = 1 in. (err)	b = 2 in. (err)	b = 3 in. (err)
Box nearest edge	24	14.1 (41%)	15.39 (36%)	16.33 (32%)
Chess mat midsection	28.75	15.39 (46%)	17.21 (40%)	17.38 (40%)
Chess mat corners	22.13	L – 13.15 (41%) R – 13.00 (41%)	L – 14.63 (34%) R – 13.68 (38%)	L – 15.95 (28%) R – 14.33 (35%)
Rubik's cube vertex at bottom	12	7.65 (36%)	8.07 (33%)	7.58 (37%)

Table 1: Measured z-positions compared with actual values at different b settings (part 2).

Point	Actual z (in.)	b = 4 in. (err)	b = 5 in. (err)	b = 6 in. (err)
Box nearest edge	24	17.14 (29%)	18.40 (23%)	16.83 (30%)
Chess mat midsection	28.75	18.80 (35%)	21.27 (26%)	19.02 (34%)
Chess mat corners	22.13	L – 16.42 (26%) R – 15.19 (31%)	L – 17.46 (21%) R – 16.53 (25%)	L – 15.81 (29%) R – 15.13 (32%)
Rubik's cube vertex at bottom	12	7.88 (34%)	7.72 (36%)	8.04 (33%)

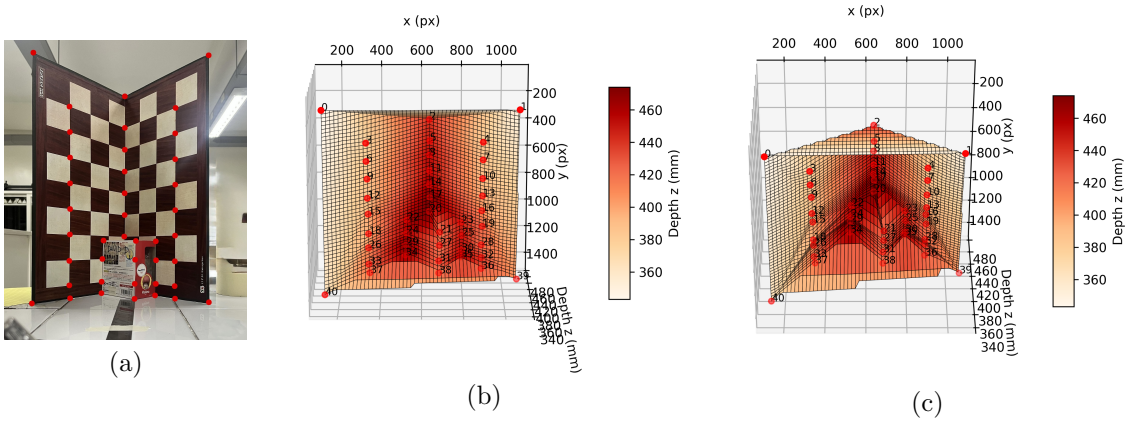


Figure 2: (a) Reference image of Tsai Grid setup and (b, c) Two slightly different viewing angles of its 3D surface reconstruction for $b = 3$ inches. The red dots indicate the same points between the setup and the reconstruction.

Although the plots looked the same for various values of b, we calculated different values for the depth z for the same points. The calculated z for selected points, along with the actual distance from the camera, are summarized in Table 1. Possible sources of error and deviation of calculated z for different b values are slight zooming or distortion when taking photos from different angles. This became apparent when the recorded images were aligned vertically, and not everything matched, meaning that more than horizontal shift happened between the captures. The largest contributor to error, however, is the resizing of images during pre-processing. Because the original photos have different dimensions, they were rescaled for consistency and the applicability of stereometry. Resizing the photos directly affected the results because it altered the pixel to millimeter ratio of the sensor width to the image width in the calculation of z in Eq. (2), which is essential in relating the pixel sizes to real-world lengths. It is not sufficient to directly relate the object distances in the photos and the pixel sizes because real-world lengths and pixel sized depend fundamentally on the intrinsics of the camera. Moreover, z is very sensitive to errors in the disparity $d = x_1 - x_2$, especially for objects far from the camera wherein d is small, because of their inverse relationship. This explains why the reconstructions were consistent in relative shape, but differ significantly in absolute depth.

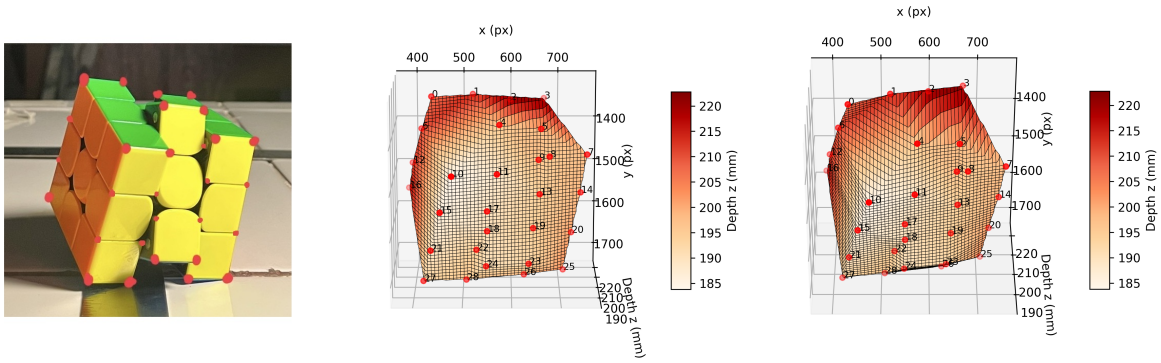
3.2 Rubik's Cube Setup

The Rubik's cube, with its layers rotated, was more challenging to reconstruct since the object was small and the sample points were close. This meant that interpolation of these sample points often connected the layers that were supposed to be separated. A greater number of corresponding points would have been necessary for a more accurate surface. Nevertheless, we see in Figure 3 that a recognizable resemblance of the Rubik's cube is captured by the reconstruction, although the rotations on the cube's layers were not resolved. The leftmost face can be easily identified in the plots since, interpolating between the sample points for this face is straightforward. In contrast, the interpolation of the points in the middle part of the Rubik's cube did not accurately give the effect of the layers being rotated. It is important to note that the top view does not resemble a cube, as depth is only computed for the visible points in the front that are seen when taking images. Other plots for different b values were similar to the ones showed in the figure. The relative scaling was preserved, but the absolute values for z were different for the same reasons outlined in the last section.

The last row in Table 1 shows the calculated and actual z of the vertex at the bottom face of the cube closest to the camera for different b values. As in the chess mat setup, large errors were observed in the calculation of z, for the same reasons discussed earlier (resizing scaling and disparity sensitivity). Additionally, the smaller physical size of the Rubik's cube compared to the chess mat likely magnified errors, as each pixel now corresponds to larger fractions of length of the object. This led to smaller values for d in calculating z, which as mentioned before results to z becoming highly sensitive to small errors in d.

3.3 Effect of b on stereometry accuracy

Figure 4 shows the average error of the calculated depth as a function of b for the two setups. It is seen that increasing b leads to smaller errors for the chess mat, reaching a minimum until $b = 5$ in. before rising again for $b = 6$ in. This suggests that an optimal interaxial distance for depth recovery can be obtained



(a) (b) (c)

Figure 3: (a) Reference image of Rubik's cube setup and (b and c) Two slightly different viewing angles of its 3D surface reconstruction for $b = 3$ inches.

up to a certain point, where further increase beyond that point could introduce more error possibly from distortion, The chess mat setup, being larger and being a flat plane having a more straightforward orientation than the Rubik's cube, is more sensitive to this effect, where pronounced improvements in the error are seen. In contrast, it is seen in the plot that the error for the Rubik's cube fluctuates, decreasing and increasing from point to point. This can be attributed to the mentioned higher fractional length of the object contained in one pixel, which makes the calculation of z highly sensitive. Thus, object size and configuration of planes play a role in how b can affect the accuracy of z.

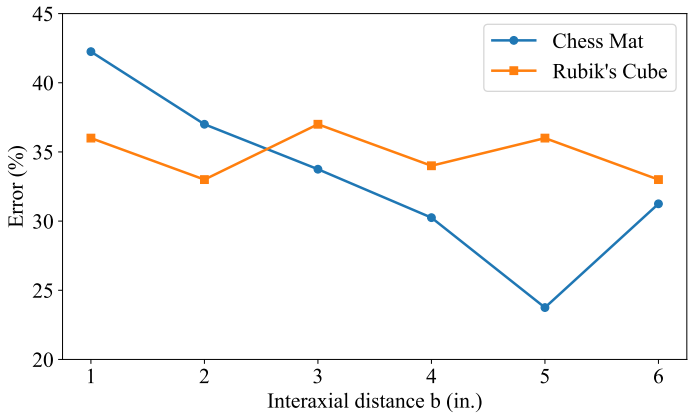


Figure 4: Average error in the depth z for various interaxial distances b

4 Conclusions

In this experiment, we applied 3D imaging stereometry to reconstruct the shape and depth of a Tsai grid setup and a Rubik's cube. The reconstructions successfully captured the overall shape and relative positioning of the objects. This showed that 3D information can be retrieved from flat 2D images. However, the calculated depth values varied significantly with small changes in interaxial distance b. Possible sources of error include small misalignments when capturing the photos and the sensitivity of depth calculations to small pixel shifts.

Overall, the experiment demonstrated the potential of stereometry in recovering 3D information from 2D images but also highlighted the importance of careful setup and calibration for accurate depth measurements. For future work, we recommend using cameras with fixed resolution , capturing more sample points for better interpolation, and performing camera calibration to obtain correct focal length.

References

[1] S. R., *Computer Vision: Algorithms and Applications* (Springer, Switzerland AG, 2022).

[2] F. Li, J. Johnson, and S. Yeung, Technical Report, Stanford University (2020), https://web.stanford.edu/class/cs231a/course_notes/01-camera-models.pdf.