GEO-1016 Assignment 1: Calibration

Group 9: Ming-Chieh Hu, Neelabh Singh & Daan Schlosser

March 27, 2025

1 Methodology

For each pair of points in the 3D coordinate system and the 2D image plane, we know their relationship is given by $\mathbf{p}_i = K[R \;\; \mathbf{t}] = M\mathbf{P}_i$. With n pairs of these corresponding points in a scene, the entire system becomes:

$$\begin{bmatrix} \mathbf{P}_1^T & 0^T & -u_1\mathbf{P}_1^T \\ 0^T & \mathbf{P}_1^T & -v_1\mathbf{P}_1^T \\ \vdots & \vdots & \vdots \\ \mathbf{P}_n^T & 0^T & -u_n\mathbf{P}_n^T \\ 0^T & \mathbf{P}_n^T & -v_n\mathbf{P}_n^T \end{bmatrix} \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{bmatrix} = P\mathbf{m} = 0, \text{ as we denote } M = \begin{bmatrix} \mathbf{m}_1^T \\ \mathbf{m}_2^T \\ \mathbf{m}_3^T \end{bmatrix} \tag{1}$$

We then followed the instructions in the example codebase (`calibration_method.cpp`) and decomposed the whole process as the following major steps.

Constructing P Matrix P has a size of $2n \times 12$, where n is the number of 3D-2D correspondences. Then for each 3D-2D correspondence, 3D point $\mathbf{P}_i$ is transformed into homogeneous coordinates, a 4D vector (X_i, Y_i, Z_i and 1) . The pixel coordinates $\mathbf{p}_i$ components u_i and v_i are also retrieved and used to construct P. All these values are combined according to the following matrix structure:

$$\begin{bmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -u_1X_1 & -u_1Y_1 & -u_1Z_1 & -u_1 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -v_1X_1 & -v_1Y_1 & -v_1Z_1 & -v_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ X_n & Y_n & Z_n & 1 & 0 & 0 & 0 & 0 & -u_nX_n & -u_nY_n & -u_nZ_n & -u_n \\ 0 & 0 & 0 & 0 & X_n & Y_n & Z_n & 1 & -v_nX_n & -v_nY_n & -v_nZ_n & -v_n \end{bmatrix} \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{2}$$

Solving $\mathbf{m}$ If in the equation $P\mathbf{m} = \mathbf{0}$, P has a pivot position in every column (i.e., $rank = 12$, which is very likely to happen given $2n \geq 12$), then $P\mathbf{m} = \mathbf{0}$ has only the trivial solution $\mathbf{m} = \mathbf{0}$. Therefore, we reformulate this as a minimization problem and use singular value decomposition (SVD) to find the closest solution. If we let $P = UDV^T$, then the solution to the above minimization problem is to set $\mathbf{m}$ equal to the last column of V.

Verify inputs and intermediate results To ensure the function operates correctly, we implemented a series of value checks. First, we verify that the number of correspondences is at least six. We also ensure that the sizes of the 2D and 3D point sets match. Additionally, we validate the singular value decomposition (SVD) results by checking that $P = U\Sigma V^T$ holds and confirming that $UU^T = I$ and $VV^T = I$. Lastly, we check that K^{-1} exists. To further ensure accuracy, the value of $||P\mathbf{m}||^2$ is displayed to verify that it is close to zero.

Extract parameters After reformatting the vector $\mathbf{m}$ into the matrix M, we now want to solve for the extrinsic and intrinsic parameters explicitly, which are denoted in the following :

$$R = \begin{bmatrix} \mathbf{r}_1^T \\ \mathbf{r}_2^T \\ \mathbf{r}_3^T \end{bmatrix}, \mathbf{t} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \; and \; K = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \alpha & -\alpha\cot\theta & c_x \\ 0 & \frac{\beta}{\sin\theta} & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

Since our SVD-solved projection matrix (denoted by $\mathcal{M}$) is known up to scale, the true projection matrix M is some scalar multiple of $\mathcal{M}$, i.e., $\rho\mathcal{M} = M$. Dividing by the scaling parameter gives $\mathcal{M} = \frac{1}{\rho}M$.

Because we have obtained all the entries of $\mathcal{M}$, denote $\mathcal{M} = \begin{bmatrix} A & \mathbf{b} \end{bmatrix} = \begin{bmatrix} \mathbf{a}_1^T & b_1 \\ \mathbf{a}_2^T & b_2 \\ \mathbf{a}_3^T & b_3 \end{bmatrix}$, we solve for the intrinsics:

$$
\begin{aligned}
\rho &= \pm\frac{1}{\|\mathbf{a}_3\|} \\
c_x &= \rho^2(\mathbf{a}_1 \cdot \mathbf{a}_3) \\
c_y &= \rho^2(\mathbf{a}_2 \cdot \mathbf{a}_3) \\
\cos\theta &= -\frac{(\mathbf{a}_1 \times \mathbf{a}_3) \cdot (\mathbf{a}_2 \times \mathbf{a}_3)}{\|\mathbf{a}_1 \times \mathbf{a}_3\| \cdot \|\mathbf{a}_2 \times \mathbf{a}_3\|} \\
\alpha &= \rho^2\|\mathbf{a}_1 \times \mathbf{a}_3\| \sin\theta \\
\beta &= \rho^2\|\mathbf{a}_2 \times \mathbf{a}_3\| \sin\theta \\
f_x &= \alpha \\
f_y &= \beta \sin\theta \\
s &= -\alpha \cot\theta
\end{aligned}
\tag{4}
$$

The extrinsics are:

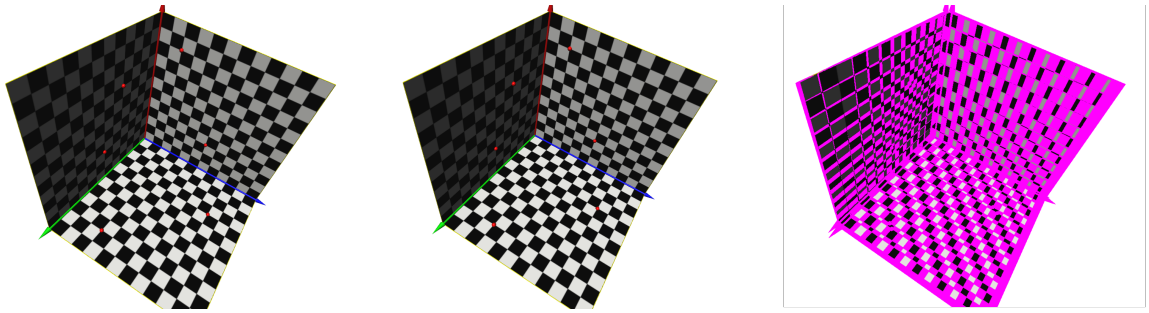
$$
\begin{aligned}
\mathbf{r}_1 &= \frac{\mathbf{a}_2 \times \mathbf{a}_3}{\|\mathbf{a}_2 \times \mathbf{a}_3\|} \\
\mathbf{r}_3 &= \rho\mathbf{a}_3 \\
\mathbf{r}_2 &= \mathbf{r}_3 \times \mathbf{r}_1 \\
\mathbf{t} &= \rho K^{-1}\mathbf{b}
\end{aligned}
\tag{5}
$$

2 Calibration Result

2.1 Data collection

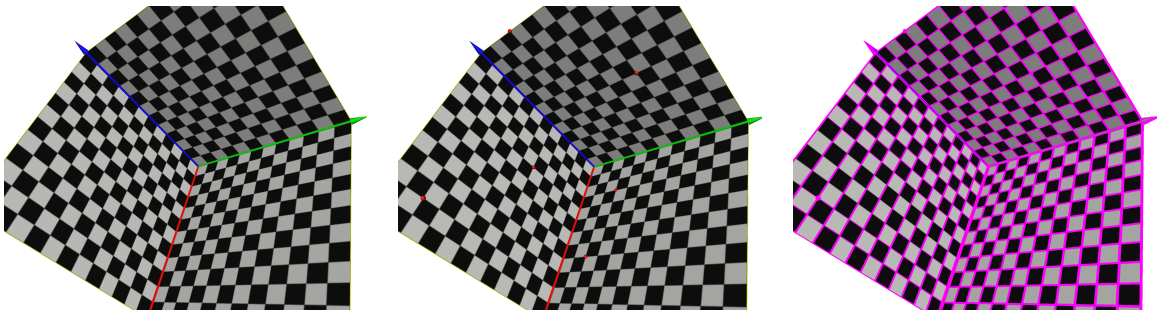
We collected 2 sets of test data with 6 pairs of points each, in total 12 pairs of points (*collected_data_1-2*). These 2 test datasets were collected using the same tool pixspy.com. Combined with the provided 3 sets of test data (*test_data_1-3*), we present analyses of 5 results. Results for our collected data are displayed in Figure 1, 2 and Table 1, 2; Results for given test data are displayed in Figure 3, 4, 5 and Table 3, 4, 5.

2.2 Result visualization



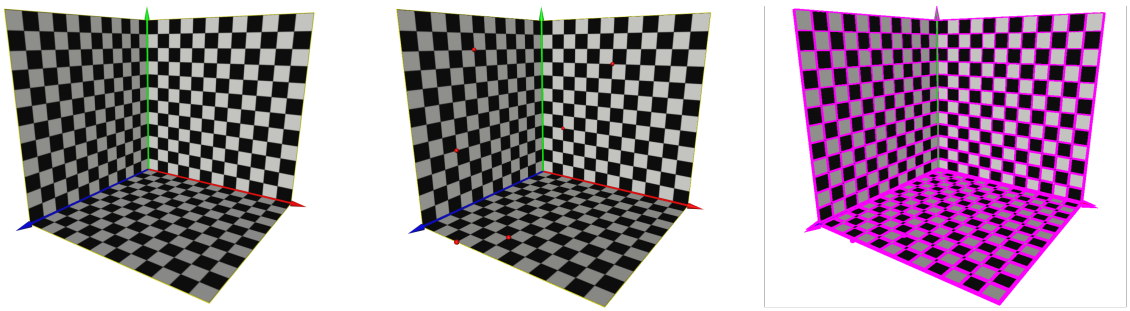
(a) Ground truth (b) Our result (c) Difference

Figure 1: Camera calibration results for *collected_data_1*



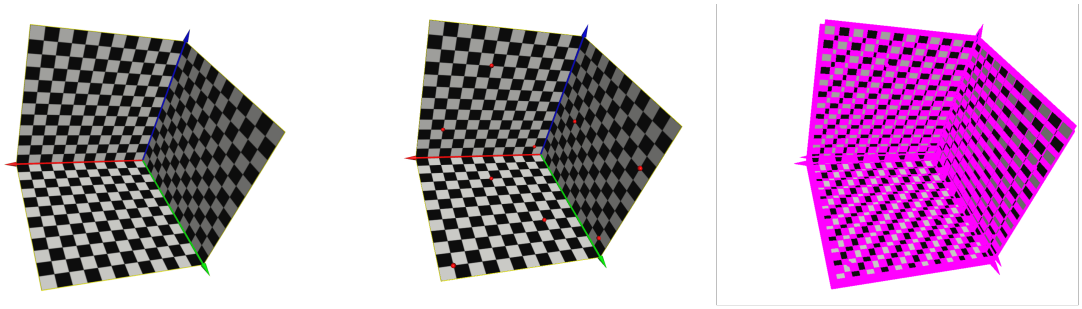
(a) Ground truth (b) Our result (c) Difference

Figure 2: Camera calibration results for *collected_data_2*



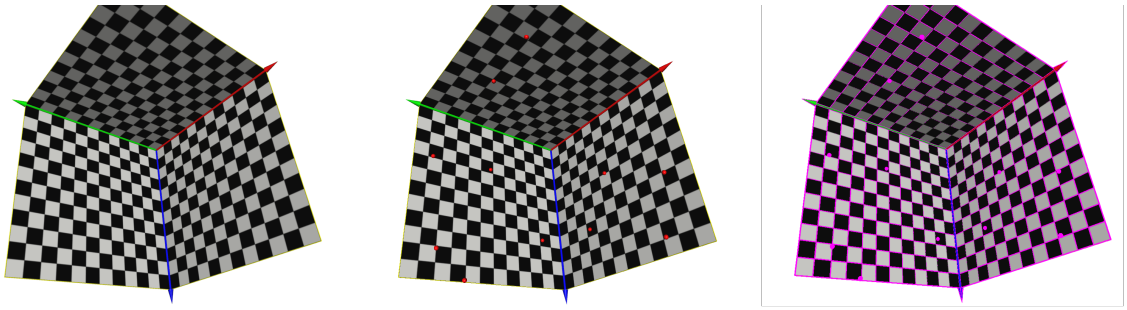
(a) Ground truth (b) Our result (c) Difference

Figure 3: Camera calibration results for *test_data_1*



(a) Ground truth (b) Our result (c) Difference

Figure 4: Camera calibration results for *test_data_2*



(a) Ground truth (b) Our result (c) Difference

Figure 5: Camera calibration results for *test_data_3*

2.3 The accuracy of the results

Error sources There are many factors that account for the noise in the data set, leading to a higher error in the calibration. Some errors can be attributed to human error due to imperfect measurements. One of the reasons could also be the low pixel resolution of the viewer, and the captured images. This makes it difficult, if not impossible, to determine the right pixel for the corresponding 3D coordinate. Additionally, even when the point which is visually closest one is given, it is not accurate, since the true point often lies between pixel values.

Accuracy Generally, the accuracy of the camera calibration improves as the number of correspondence points increases. This is evidently visible in the Figure 5 compared to the other test cases, see Figure 3 and Figure 4. Additionally, the Table 5 also reflects that the reprojection error values are less when there are 11 correspondence points, compared to the reprojection error values when there are only 6 correspondence points, see Table 3. However, in the presence of noise in the test data, increasing the number of correspondence points does not necessarily lead to better calibration. We can see in Table 4 that the reprojection error is significantly higher, even though it has 9 correspondence points, compared to the case which has only 6 correspondence points see Table 3. Additionally, in Figure 4 it is clearly visible that the camera is not calibrated well after reprojection and has error, due to noise in the test data.

Effect of zooming Among the two results of our collected data, both the value of $||P\mathbf{m}||^2$ (Table 1 and 2) and the visualized difference from ImageDiff (Figure 1 and 2) indicate that the *collected_data_2* is less noisy compare to *collected_data_1*. However, the reprojection error for the former is oddly comparable to the latter. We believe this difference is due to zooming. As the camera zooms in, it captures the same area but displays it larger and with more detail, effectively increasing the focal length. To account for this, we calculate a scaling factor based on the ratio of their focal lengths. As shown in Equation 6, we propose multiplying by a scalar to enable a fair comparison on the same basis. This assumption is reasonable because the obtained value (3.69) is close to the ratio of the two $||P\mathbf{m}||^2$ values, $\frac{2.5952610^8}{5.8365810^{-9}} \approx 4.45$.

$$\begin{aligned} &\text{focal length for our collected data 1 } f_{1,x} = 1430.10,\ f_{1,y} = 1434.22 \\ &\text{focal length for our collected data 2 } f_{2,x} = 1944.22,\ f_{2,y} = 1944.88 \\ &\text{scale multiplier } s_x = \frac{f_{2,x}}{f_{1,x}},\ s_y = \frac{f_{2,y}}{f_{1,y}} \\ &\text{total scale multiplier } s = s_x^2 + s_y^2 \approx 3.69 \end{aligned} \tag{6}$$

$\|\|P\mathbf{m}\|\|^2 = 2.59526 \times 10^{-8}$				
Pair No.	**3D Coordinates (world)**	**Our Projection**	**Ground Truth**	**Pairwise L2 Loss**
1	(3 6 0)	(702.254 979.868)	(702 980)	0.0820609
2	(8 4 0)	(753.877 657.115)	(753 657)	0.78252
3	(10 0 2)	(983.965 516.915)	(983 517)	0.939059
4	(3 0 6)	(1243.46 888.125)	(1243 888)	0.230755
5	(0 10 3)	(612.926 1282.08)	(613 1282)	0.0121833
6	(0 4 9)	(1316.15 1159.89)	(1316 1160)	0.0345103
Reprojection Error (L2 Loss)				**2.08109**

Table 1: Result and projection error of *collected_data_1*

$\|\|P\mathbf{m}\|\|^2 = 5.83658 \times 10^{-9}$				
Pair No.	**3D Coordinates (world)**	**Our Projection**	**Ground Truth**	**Pairwise L2 Loss**
1	(4 0 5)	(708.922 844.867)	(709 845)	0.0236861
2	(10 0 10)	(129.158 1007.1)	(129 1007)	0.035878
3	(0 8 6)	(1249.3 351.938)	(1250 352)	0.495223
4	(0 3 12)	(584.131 132.019)	(585 132)	0.756302
5	(3 3 0)	(1141.17 966.188)	(1141 966)	0.0632351
6	(8 2 0)	(982.638 1314.89)	(982 1315)	0.419565
Reprojection Error (L2 Loss)				**1.79389**

Table 2: Result and projection error of *collected_data_2*

$\|\|P\mathbf{m}\|\|^2 = 4.42946 \times 10^{-9}$				
Pair No.	**3D Coordinates (world)**	**Our Projection**	**Ground Truth**	**Pairwise L2 Loss**
1	(2 4 0)	(1021.23 648.012)	(1021 648)	0.0530015
2	(6 9 0)	(1282.59 307.999)	(1282 308)	0.345008
3	(5 0 10)	(733.648 1225.95)	(734 1226)	0.126055
4	(3 0 12)	(458.725 1250.07)	(459 1250)	0.0813551
5	(0 4 9)	(458.129 765.949)	(458 766)	0.0191686
6	(0 10 7)	(550.65 233.016)	(550 233)	0.423114
Reprojection Error (L2 Loss)				**1.0477**

Table 3: Result and projection error of *test_data_1*

$\|\|P\mathbf{m}\|\|^2 = 5.79503 \times 10^{-6}$				
Pair No.	**3D Coordinates (world)**	**Our Projection**	**Ground Truth**	**Pairwise L2 Loss**
1	(-0 3 6)	(1314.55 614.653)	(1314 616)	2.12059
2	(-0 11 8)	(1652.15 857.862)	(1652 858)	0.0429897
3	(-0 11 1)	(1439.17 1220.46)	(1441 1220)	3.56067
4	(7 0 9)	(884.586 323.879)	(884 323)	1.1154
5	(1 0 1)	(1105.13 744.65)	(1106 744)	1.18492
6	(10 0 3)	(632.924 656.145)	(634 656)	1.17786
7	(6 3 -0)	(882.664 910.721)	(881 912)	4.40355
8	(3 8 -0)	(1157.08 1125.93)	(1156 1125)	2.03786
9	(11 11 0)	(686.162 1363.71)	(687 1364)	0.788174
Reprojection Error (L2 Loss)				**16.432**

Table 4: Result and projection error of *test_data_2*

$\|\|P\mathbf{m}\|\|^2 = 2.93001 \times 10^{-12}$				
Pair No.	**3D Coordinates (world)**	**Our Projection**	**Ground Truth**	**Pairwise L2 Loss**
1	(9.13784 9.21271 0.00326691)	(851.997 170)	(852 170)	$1.07681 * 10^{-5}$
2	(4.8685 8.85815 0.00370201)	(679.999 402)	(680 402)	$6.5628 * 10^{-7}$
3	(-0.0132064 6.19356 3.70991)	(664 868)	(664 868)	$7.18366 * 10^{-8}$
4	(-0.0107006 8.0542 11.8768)	(526.002 1450)	(526 1450)	$5.40817 * 10^{-6}$
5	(-0.0105981 9.97279 10.127)	(378.003 1280)	(378 1280)	$9.82891 * 10^{-6}$
6	(-0.0119576 10.6602 3.88994)	(362 796)	(362 796)	$1.33897 * 10^{-7}$
7	(-0.013227 1.76027 8.53246)	(936.002 1240)	(936 1240)	$2.8727 * 10^{-6}$
8	(9.17019 -0.00357617 10.895)	(1586 1222)	(1586 1222)	$5.79168 * 10^{-6}$
9	(10.164 -0.00379389 7.0957)	(1576 882.001)	(1576 882)	$3.03204 * 10^{-6}$
10	(3.29906 -0.00443328 8.49047)	(1186 1182)	(1186 1182)	$1.99076 * 10^{-6}$
11	(5.53453 -0.0044679 4.957)	(1262 886)	(1262 886)	$7.40196 * 10^{-7}$
Reprojection Error (L2 Loss)				$\mathbf{4.12946 * 10^{-5}}$

Table 5: Result and projection error of *test_data_3*

3 Discussion and Reflection

3.1 Discussion on sign determination of ρ

$$\mathbf{t} = \rho K^{-1}\mathbf{b} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} = \rho \begin{bmatrix} (K^{-1}\mathbf{b})_x \\ (K^{-1}\mathbf{b})_y \\ (K^{-1}\mathbf{b})_z \end{bmatrix} \tag{7}$$

- The vector $\mathbf{t}$ represents the translation from the world coordinates to its position in the camera coordinate system.
- The scalar ρ is a scale factor that affects the direction and magnitude of $\mathbf{t}$.

The sign of ρ in the equation is determined based on the z-component of the translation vector t, ensuring that it is positive when the world coordinate system's origin is in front of the camera, which is at the origin of the checkerboard in this case. The camera follows a right-handed coordinate system, where the positive z-axis points forward. So, we can check the solution of t and identify what is the direction of t_z.

- If $\left(K^{-1}\mathbf{b}\right)_z$ is positive, then ρ should also be positive to maintain the correct direction.
- If $\left(K^{-1}\mathbf{b}\right)_z$ is negative, then ρ must be negative to flip the direction and ensure $t_z > 0$.

3.2 Reflection on feedback

We had a discussion with another group about both our code and theirs. They were positive about our implementation, finding it well-structured and effective. For their code, we suggested adding a case to determine the sign of ρ based on the sign of a_3 within their program. We also proposed to them to use a similar method like ours for obtaining quantitative results and helped them on how to potentially incorporate that.

Appendix

Link to github repository: https://github.com/MCHU-1999/geo-1016-assignments.

Reprojection error

The reprojection error quantifies the accuracy of the estimates of the camera parameters by measuring how well the reprojected 3D world points align with their 2D image points. Once the projection is obtained, reprojection error is then measured by comparing projected 2D points and actual 2D image correspondences. It is given by:

$$E = \sum_{i=1}^{N} \left((x_i - x_i^{\text{true}})^2 + (y_i - y_i^{\text{true}})^2 \right) \tag{8}$$

where:

- (x_i, y_i) are the projected 2D points.
- $(x_i^{\text{true}}, y_i^{\text{true}})$ are the actual 2D image points.
- N is the total number of points.

Our understanding of the vector t

If we simplify the original formula by substitute K with identity matrix I, we get a simple camera model and the equation $\mathbf{p} = R\mathbf{P} + \mathbf{t}$. When P is the origin in world coordinates, $\mathbf{p} = R\mathbf{0} + \mathbf{t} = \mathbf{t}$. This means that when a point in the world is at the origin, the corresponding point in the camera's coordinate system is simply the translation vector $\mathbf{t}$. And for our case, where the origin of the world coordinates and selected points are all in front of the camera's image plane, t_z should always be greater than 0.

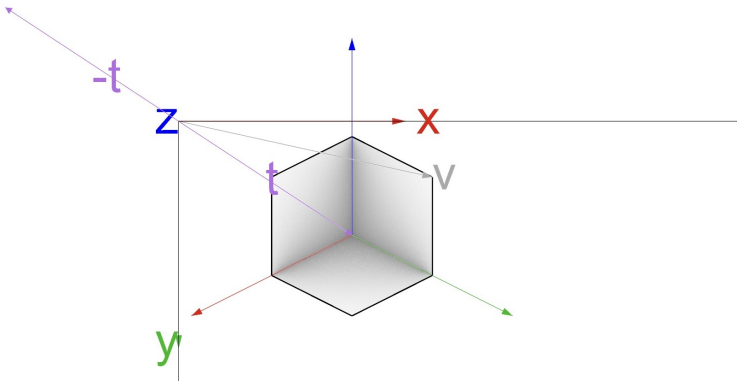


Figure 6: Relationship between world coordinate system, camera coordinate system and translation matrix $\mathbf{t}$

Credit

- Ming-Chieh Hu (6186416): data collection, data analysis, major part of programming, report revising, discussion on ρ, "effect of zooming" paragraph.
- Neelabh Singh (6052045): Reprojection Error calculations and programming, discussion on ρ and report writing
- Daan Schlosser (5726042): Wrote parts of methodology chapter, did data collection and visualised the results, discussed and compared the code and concepts with another group, e.g. on how we determine the sign of ρ, verified the report on consistency and conciseness.