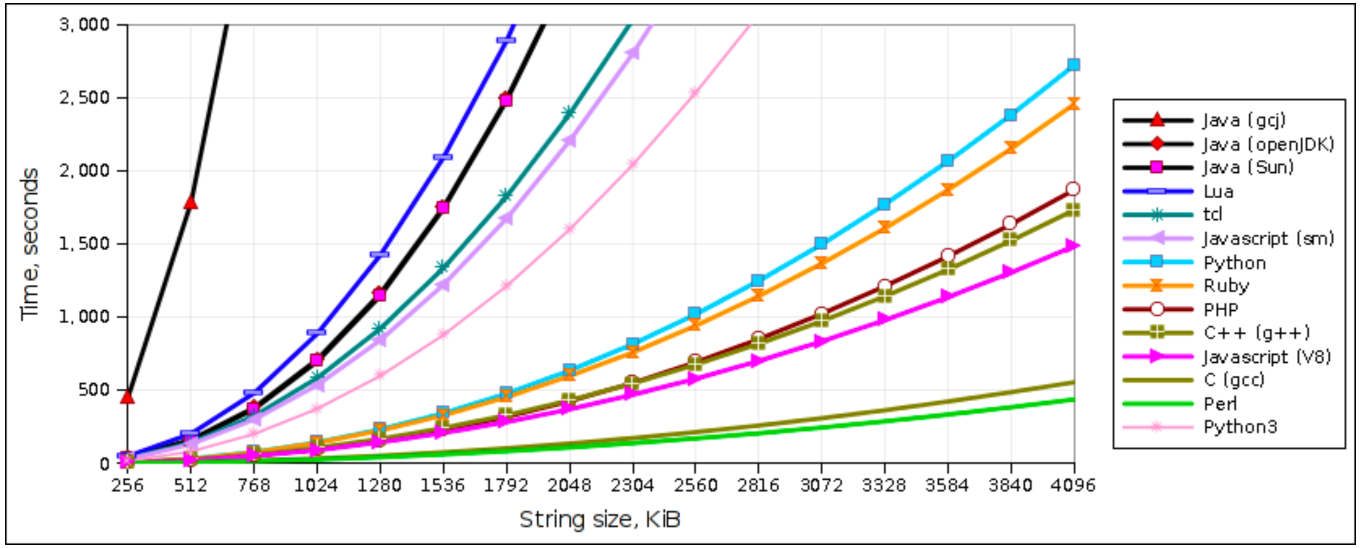
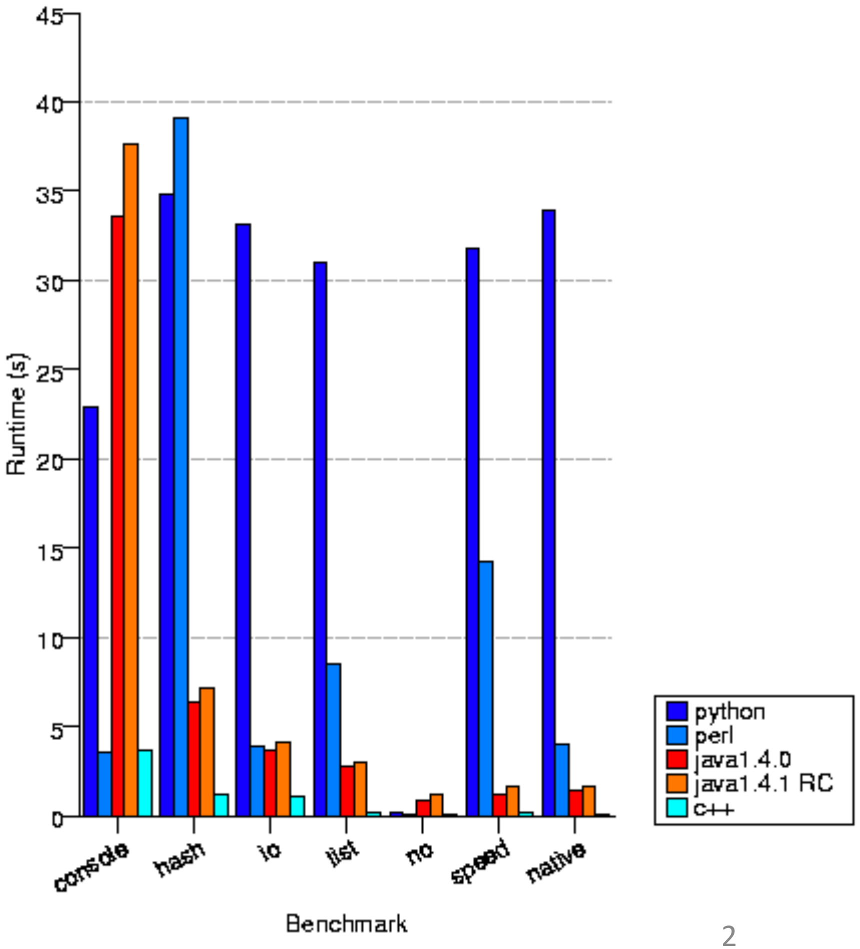
# Feedback and Discussion

- Why C++?
- Why do I have to submit full source code?
- What is “K * [R, **t**]”?
- A1: Calibration
  - What is the main source the error in camera calibration?
  - Why cannot I crop the image (but snapshots of the full viewer)?
  - How to determine the sign of rho?
- A2: Triangulation
  - Scaling of F doesn’t matter – why?
  - Determine the correct R-t pair
  - The effect of errors in K on reconstruction
  - The accuracy of reconstruction from two similar views

# Why C++?

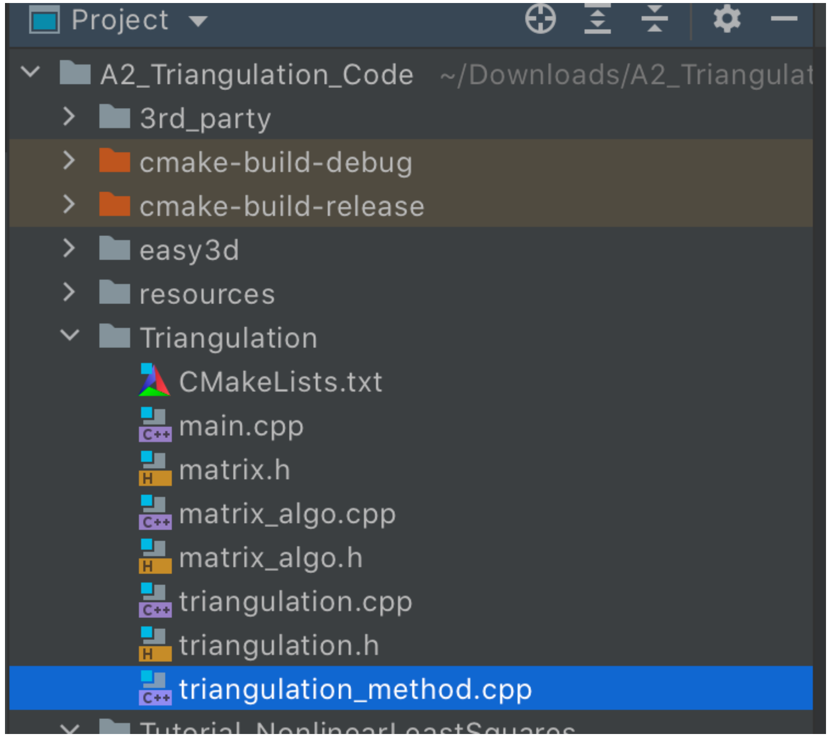
- Performance is critical
  - Large images, point clouds
  - SfM, MVS
- “bilingual” ☺



# Why do I have to submit full source code?

- Minimize the effort of evaluation
  - Other files are modified by students
  - Teacher improved the framework and the code not compatible



# What is “K * [R, **t**]”?

- It is not K * R * **t**
- [R, **t**] denotes the concatenation of R and **t**
  - Appending **t** to R so **t** becomes its last column
- If T instead of **t**
  - T is a matrix
  - T and **t** represent the same translation transformation
  - K * [R, **t**] = K * T * R
    - Be careful: it is not K * R * T

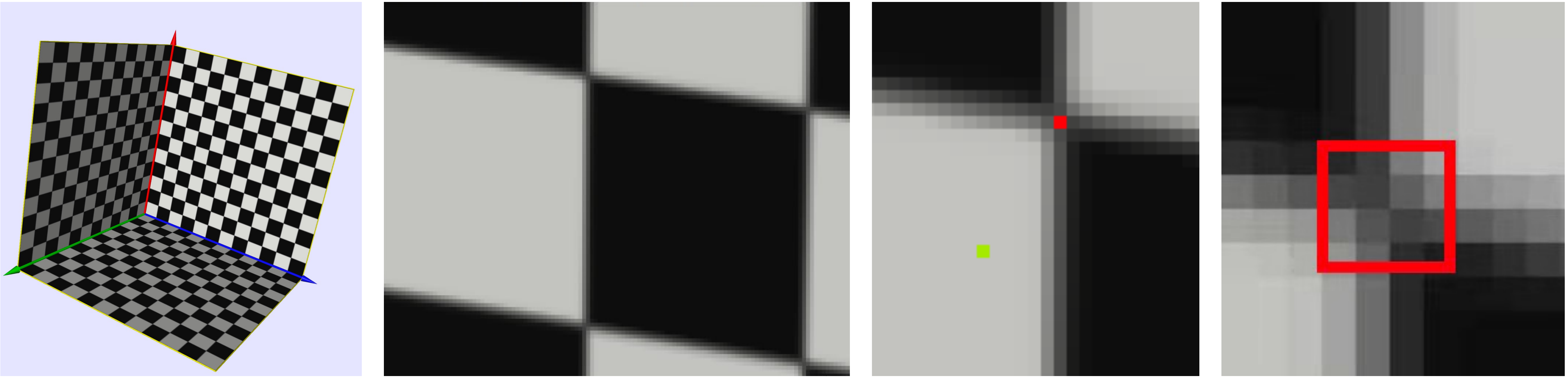
$$\mathbf{P}' = \mathbf{T} \cdot R \cdot \mathbf{P} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & t_x \\ \sin\theta & \cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

# A1: Calibration

- What is the main source the error in camera calibration?

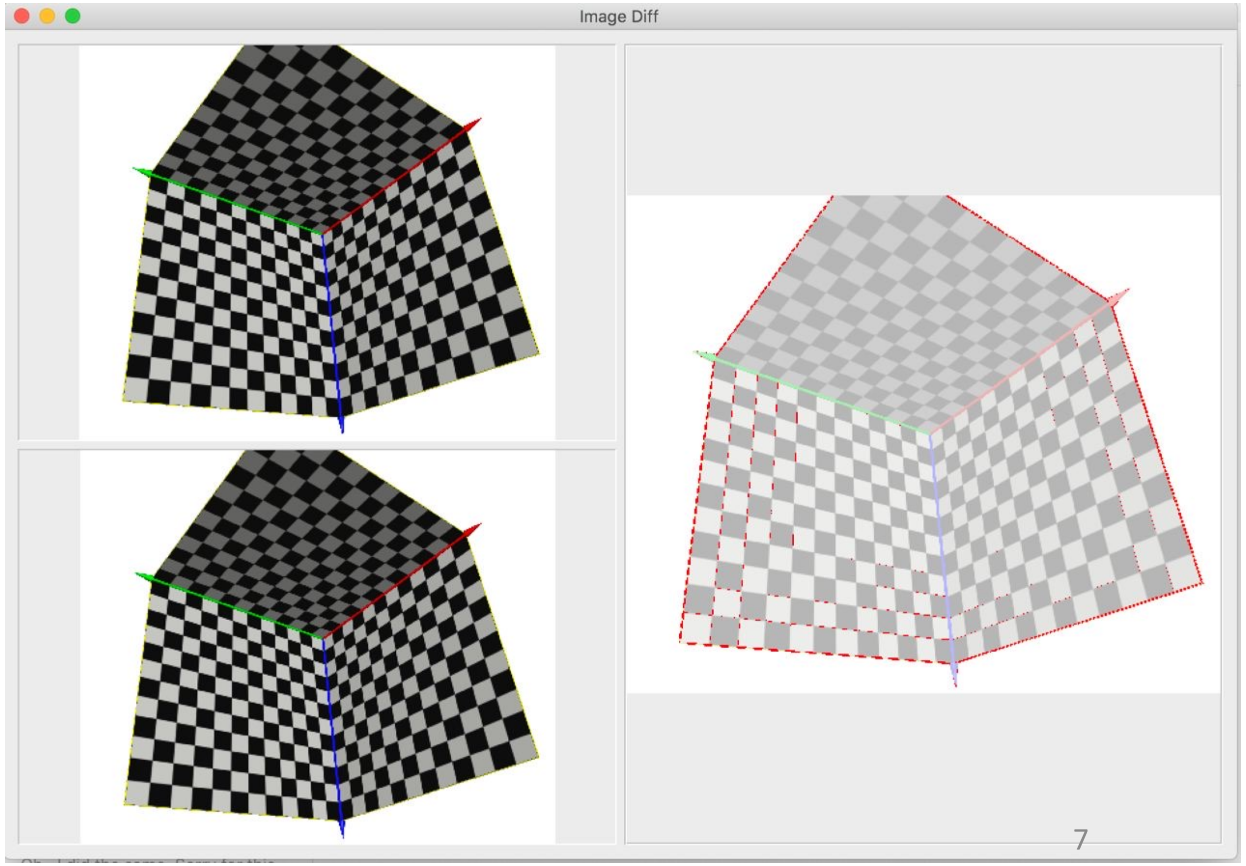
# A1: Calibration

- What is the main source the error in camera calibration?
  - Manual marking of the pixels



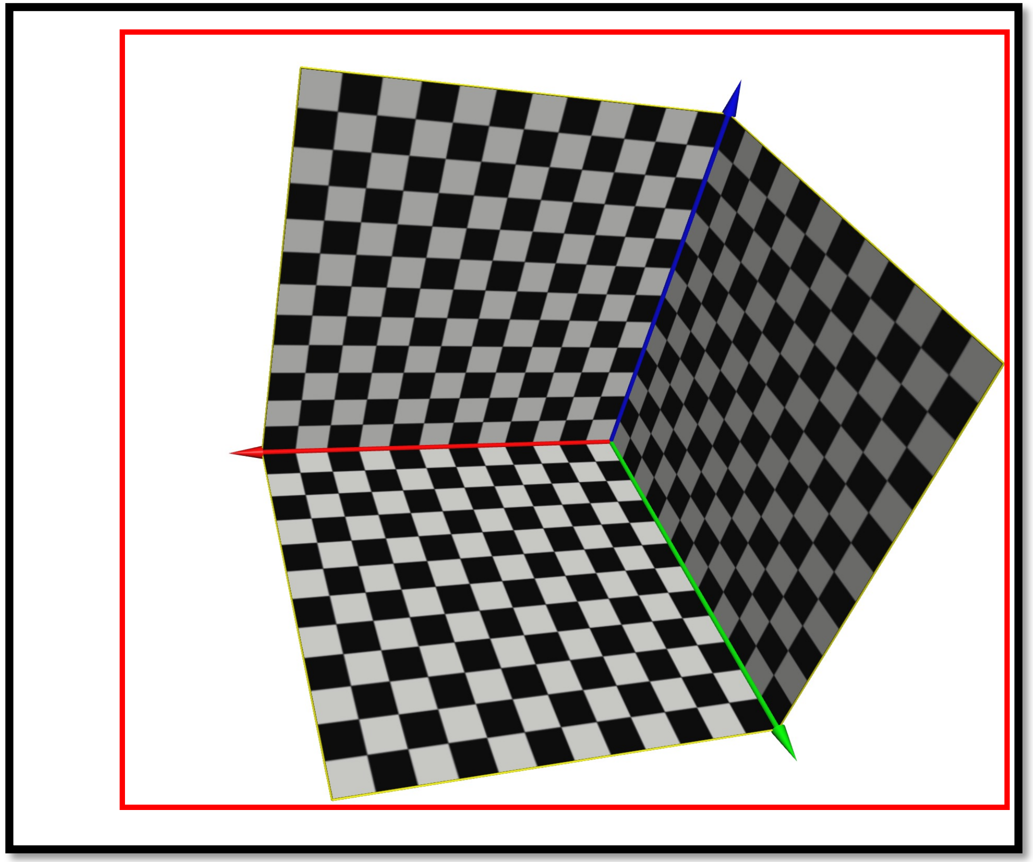
# A1: Calibration

- What is the main source the error in camera calibration?
  - Manual marking of the pixels



# A1: Calibration

- Why cannot I crop the image (but using the snapshot of the full viewer)?



# A1: Calibration

- How to determine the sign of rho?

Ask the student to explain

> The sign of rho needs to be figured out. As $\boldsymbol{\rho} = 1/Z$, with Z the distance from the camera center to the 3D point along the z axis, $\boldsymbol{\rho}$ has to be positive. Otherwise it means that the object is behind what we are looking at so it makes non sense.

$$P\mathbf{m} = 0 \quad \Rightarrow \quad \begin{array}{ll} \underset{\mathbf{m}}{\text{minimize}} & \|P\mathbf{m}\|^2 \\ \text{subject to} & \|\mathbf{m}\|^2 = 1 \end{array}$$

$$M = K\begin{bmatrix} R & \mathbf{t} \end{bmatrix}$$

$$K = \begin{bmatrix} f_x & s & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \alpha & -\alpha\cot\theta & u_0 \\ 0 & \frac{\beta}{\sin\theta} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad R = \begin{bmatrix} \mathbf{r}_1^T \\ \mathbf{r}_2^T \\ \mathbf{r}_3^T \end{bmatrix}, \ \mathbf{t} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$$

$$M = \begin{bmatrix} \alpha\mathbf{r}_1^T - \alpha\cot\theta\mathbf{r}_2^T + u_0\mathbf{r}_3^T & \alpha t_x - \alpha\cot\theta t_y + u_0 t_z \\ \frac{\beta}{\sin\theta}\mathbf{r}_2^T + v_0\mathbf{r}_3^T & \frac{\beta}{\sin\theta}t_y + v_0 t_z \\ \mathbf{r}_3^T & t_z \end{bmatrix}$$

SVD-solved projection matrix is known up to scale, i.e., $\rho\mathcal{M} = M$

SVD-solved projection matrix → $\mathcal{M}$

The true values of project matrix → $M$

$$\mathcal{M} = \frac{1}{\rho}M = \frac{1}{\rho}\begin{bmatrix} \alpha\mathbf{r}_1^T - \alpha\cot\theta\mathbf{r}_2^T + u_0\mathbf{r}_3^T & \alpha t_x - \alpha\cot\theta t_y + u_0 t_z \\ \frac{\beta}{\sin\theta}\mathbf{r}_2^T + v_0\mathbf{r}_3^T & \frac{\beta}{\sin\theta}t_y + v_0 t_z \\ \mathbf{r}_3^T & t_z \end{bmatrix}$$

# A1: Calibration

- How to determine the sign of rho?
  - Size of rho: the scale factor between the SVD-solved projection matrix and the actual projection matrix
  - Sign of rho

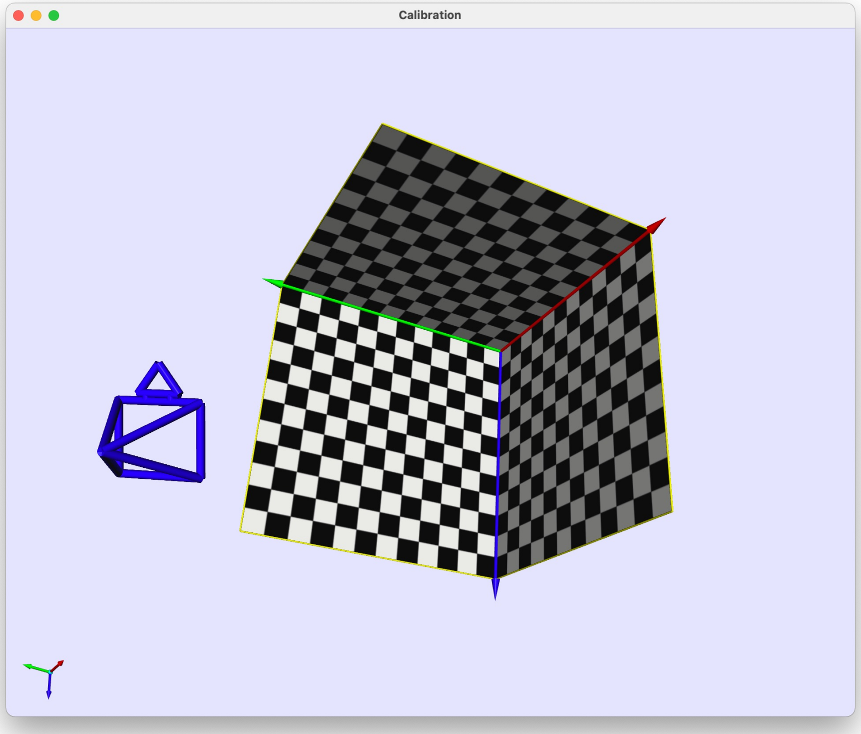
Extrinsic parameters:

$$\mathbf{r}_1 = \frac{\mathbf{a}_2 \times \mathbf{a}_3}{\|\mathbf{a}_2 \times \mathbf{a}_3\|}$$

$$\mathbf{r}_2 = \mathbf{r}_3 \times \mathbf{r}_1$$

$$\mathbf{r}_3 = \rho \mathbf{a}_3$$

$$\boxed{\mathbf{t} = \rho K^{-1} \mathbf{b}}$$



# A2: Triangulation

- Scaling of F doesn’t matter – why?
- Determine the correct R-t pair
- The effect of errors in K on reconstruction
- The accuracy of reconstruction from two similar views

# A2: Triangulation

- Scaling of F doesn’t matter – why?
  - F is a homogenenous matrix

$$p'^T F \, p = 0$$

# A2: Triangulation

- Determine the correct R-**t** pair
  - The one having the largest number of 3D points in front of **BOTH** cameras

✗

```
const vec3 p3d = triangulate(M, M_prime, p1, p2);
const vec4 p3d_h = vec4{ p3d.x, p3d.y, p3d.z,  w_in: 1.0 };

// First camera check
if (p3d.z > 0) found++;
// Second camera check
if ((Rt * p3d_h).z > 0) found++;
```

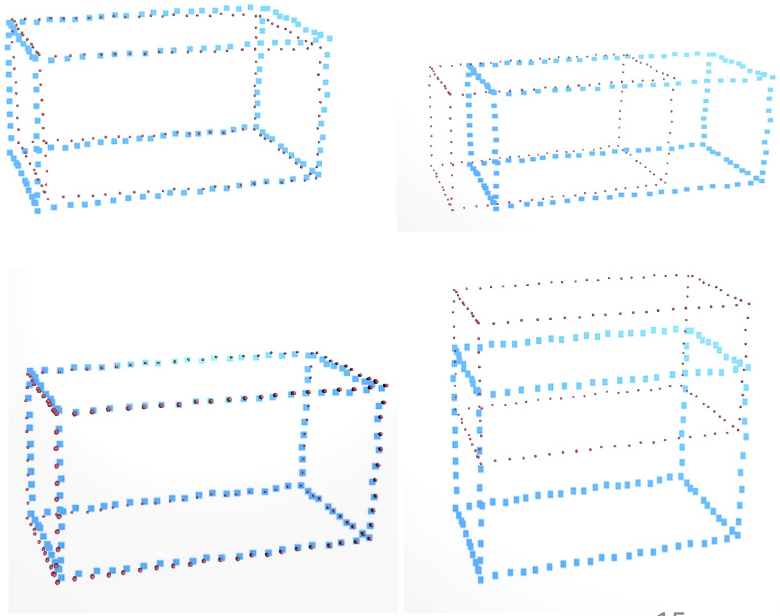
✓

```
const vec3 p3d = triangulate(M, M_prime, p1, p2);
const vec4 p3d_h = vec4{ p3d.x, p3d.y, p3d.z,  w_in: 1.0 };

if (p3d.z > 0 && (Rt * p3d_h).z > 0)
    found++;
```

# A2: Triangulation

- The effect of errors in K on reconstruction
  - How K can be obtained in practice?
  - Does K have to be very accurate?



*Accurate Camera Calibration from Multi-View Stereo and Bundle Adjustment*
Furukawa and Ponce. CVPR 2008

# A2: Triangulation

- The accuracy of reconstruction from two views

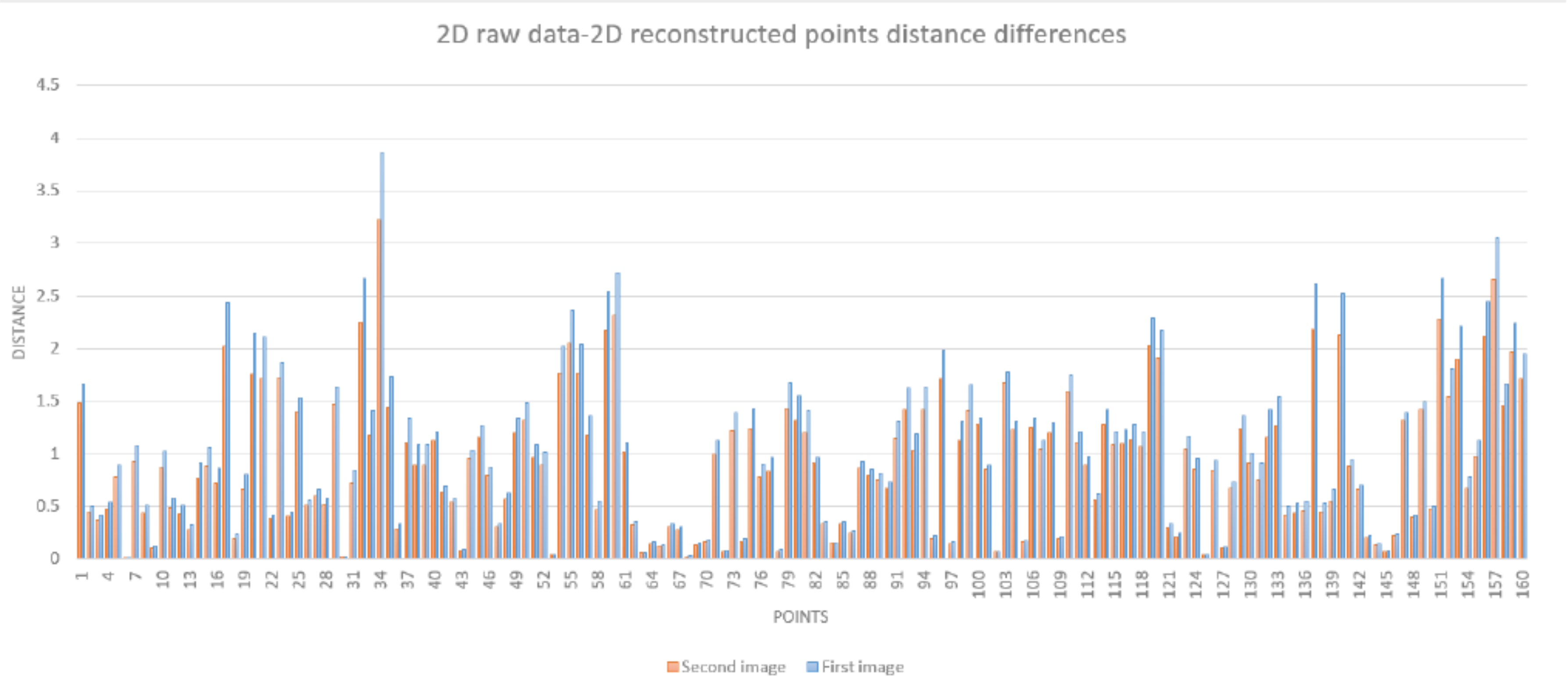



*Figure 3: Accuracy assessment- Distance differences between raw data points and 2d reconstructed points for the two images.*

# A2: Triangulation

- The accuracy of reconstruction from two views
  - Why?
  - How to improve?