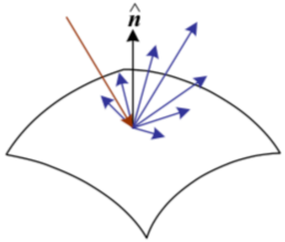
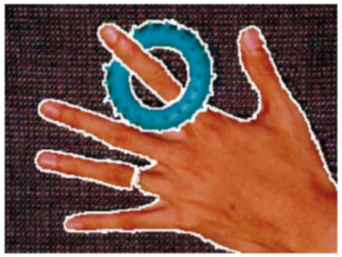


2. Image Formation

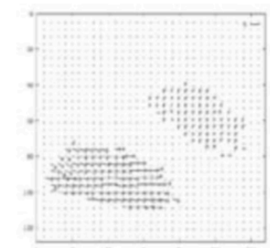
3. Image Processing

4. Features

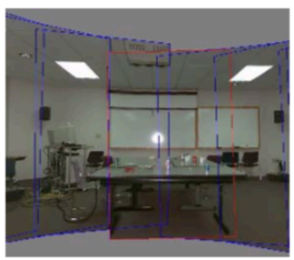


5. Segmentation

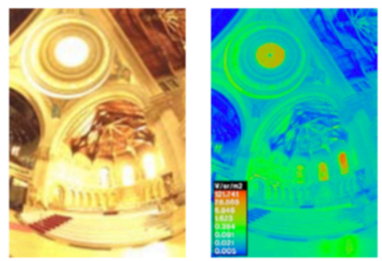
6-7. Structure from Motion



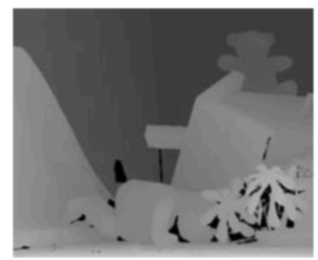
8. Motion



9. Stitching

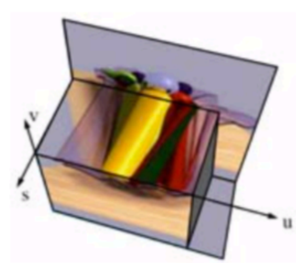


10. Computational Photography



11. Stereo

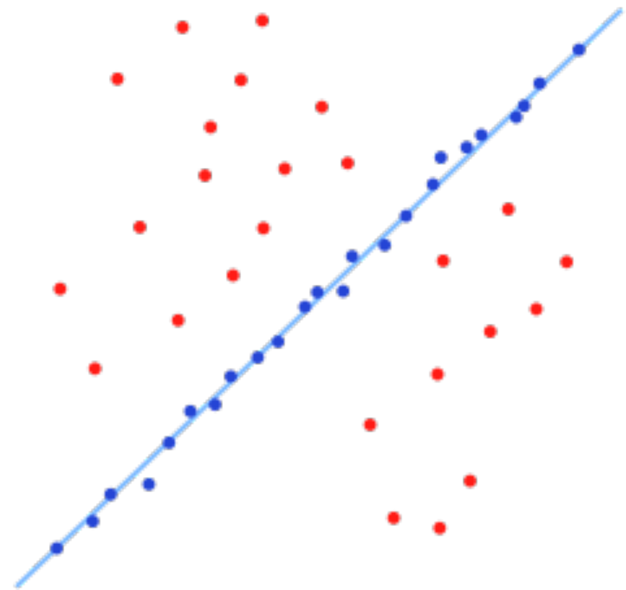
12. 3D Shape



13. Image-based Rendering

14. Recognition

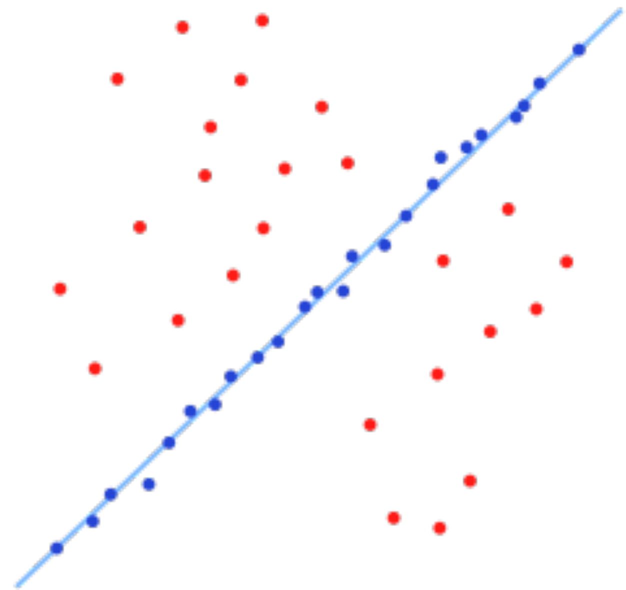
Review: RANSAC



- Objective:
  - Robust fit of a model to data D
- Algorithm
  - Randomly select s points
  - Instantiate a model
  - Get consensus set D_i
  - If $|D_i|>T$, terminate and return model
  - Repeat for N trials, return model with max $|D_i|$

Image credit Wikipedia

Adaptive N



- When etha is unknown ?
- Start with etha = 50%, N=inf
- Repeat:
  - Sample s, fit model
  - -> update etha as |outliers|/n
  - -> set N=f(etha, s, p)
- Terminate when N samples seen

Review: 2D Alignment

- Input:
  - A set of matches $\{(x_i, x_i')\}$
  - A parametric model $f(x; p)$
- Output:
  - Best model p^*
- How?

Now: 3D-2D Alignment

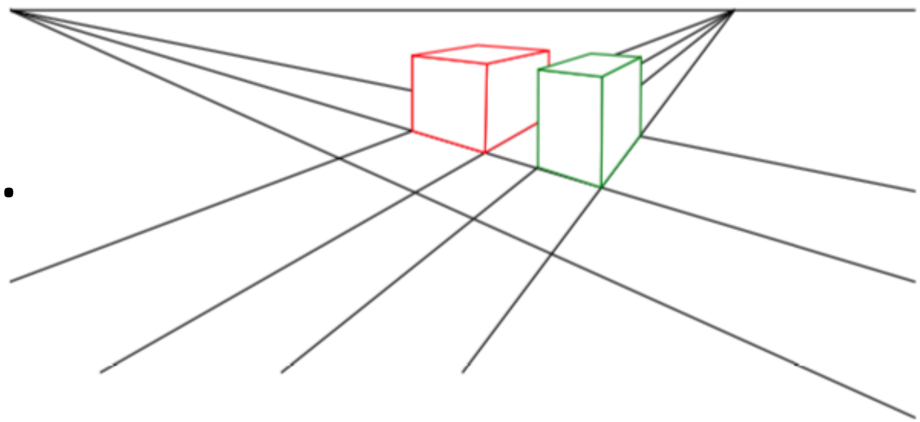
- Input:
  - A set of 3D->2D matches $\{(X_i, x_i)\}$
  - A parametric model $f(X; p)$
- Output:
  - Best model p^*
- How?

Pose Estimation

- Input:
  - A set of 2D measurements x_i of known 3D points3D X_i
  - Parametric model is camera matrix P, i.e., $x = f(X; P)$
- Output:
  - Best camera matrix P
- How?

Review: Projective Camera Matrix

- Chapter 2 in book
- Homogeneous coord.
- 3D TO 2D projection:



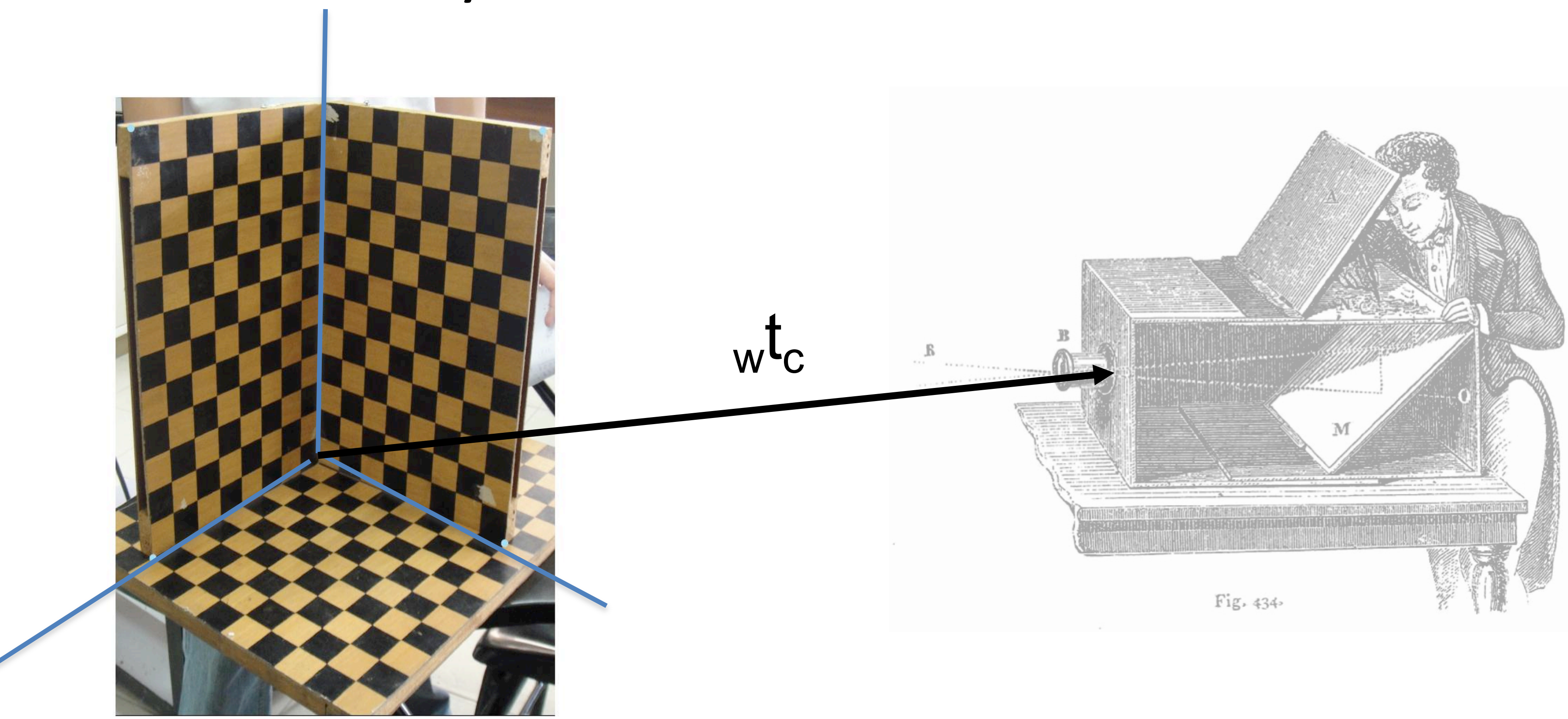
$$x = K[R|t]X = PX$$

where P = 3x4 camera matrix
and K the 3x3 calibration

$$\boldsymbol{K} = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

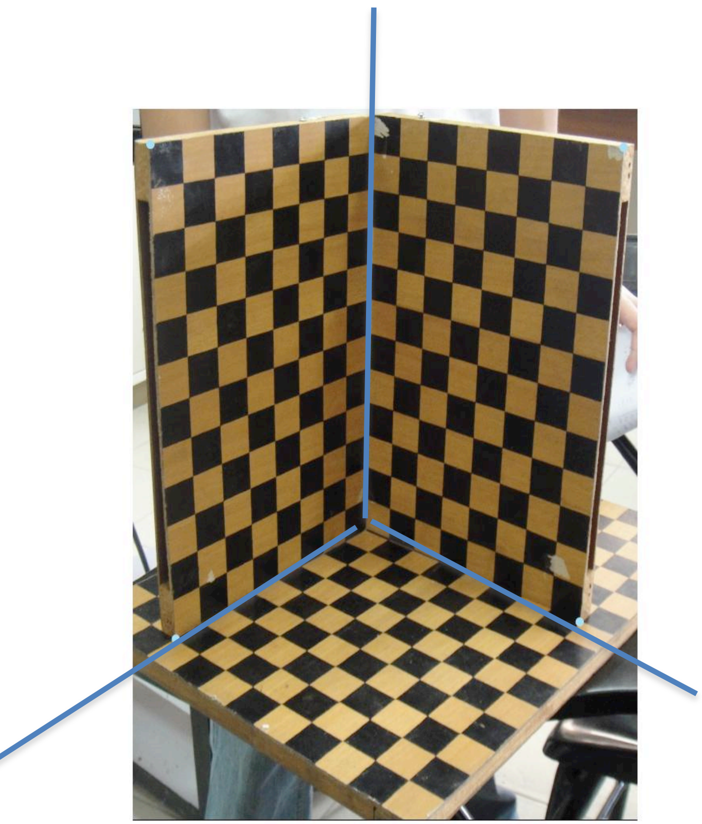
Camera Extrinsics: a Pose in 3D

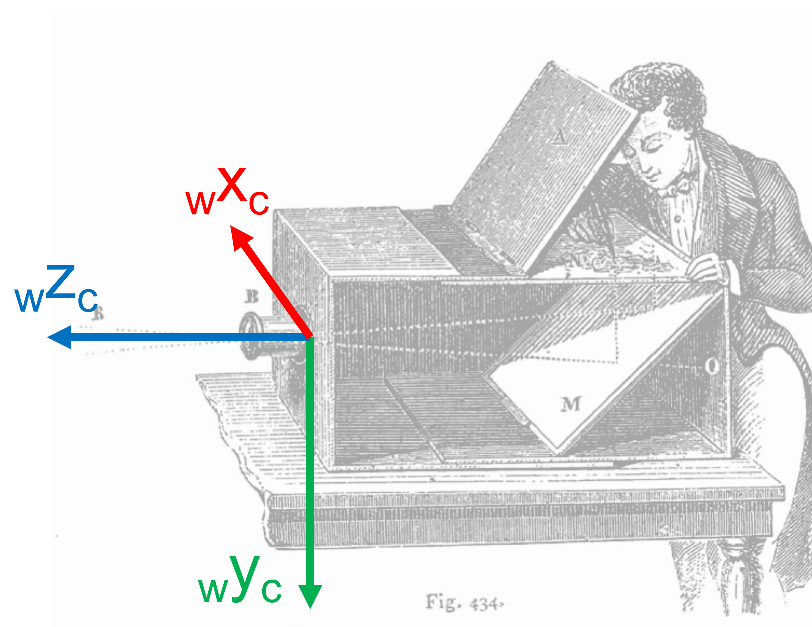
- What is the geometric meaning of R and t ??
- Intuitive: camera is at a position $_{w}t_{c}$
  Indices say: camera *in* world coordinate frame



Camera Extrinsics: a Pose in 3D

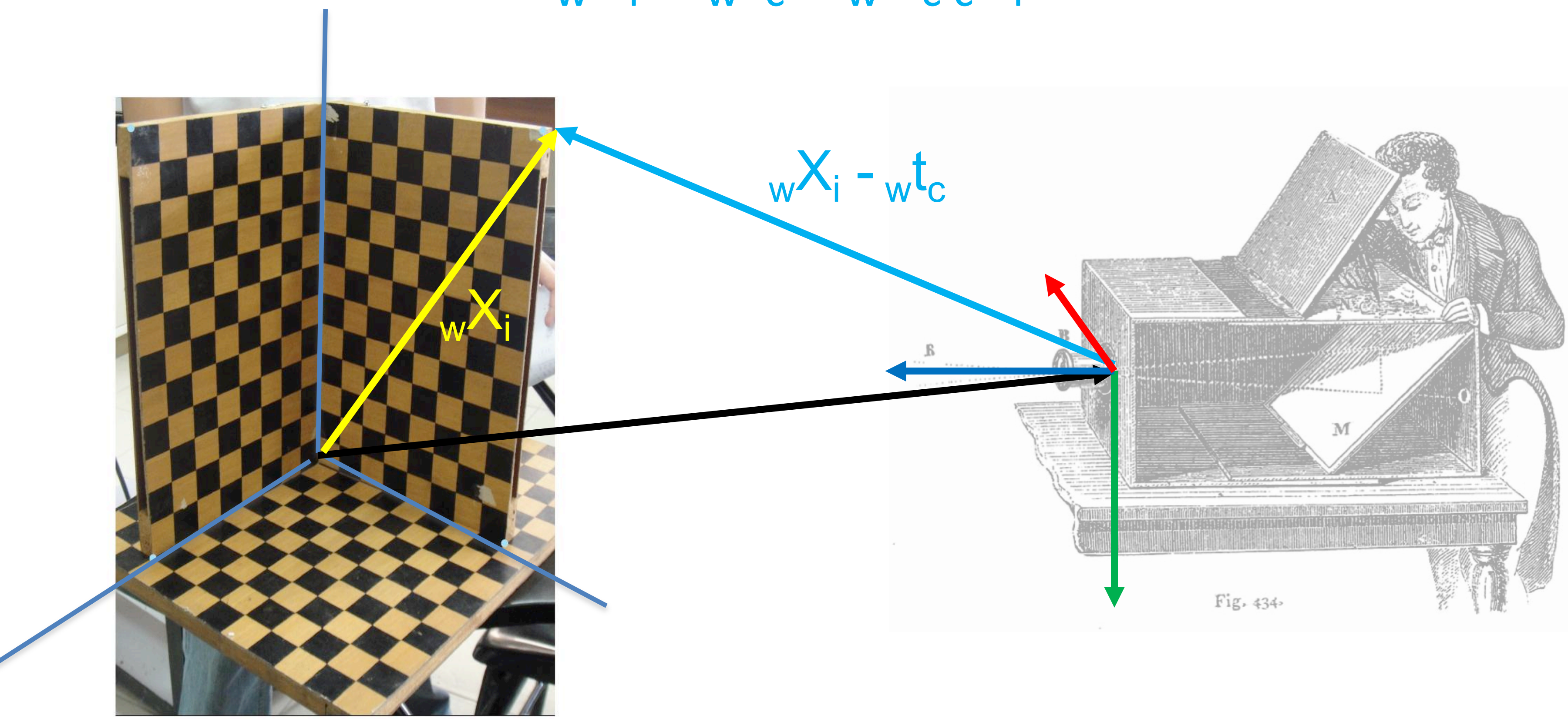
- What is the geometric meaning of R and t ??
- Rotation is given by 3x3 matrix ${}_wR_c$ whose *columns* are the camera axes ${}_wx_c$, ${}_wy_c$, ${}_wz_c$





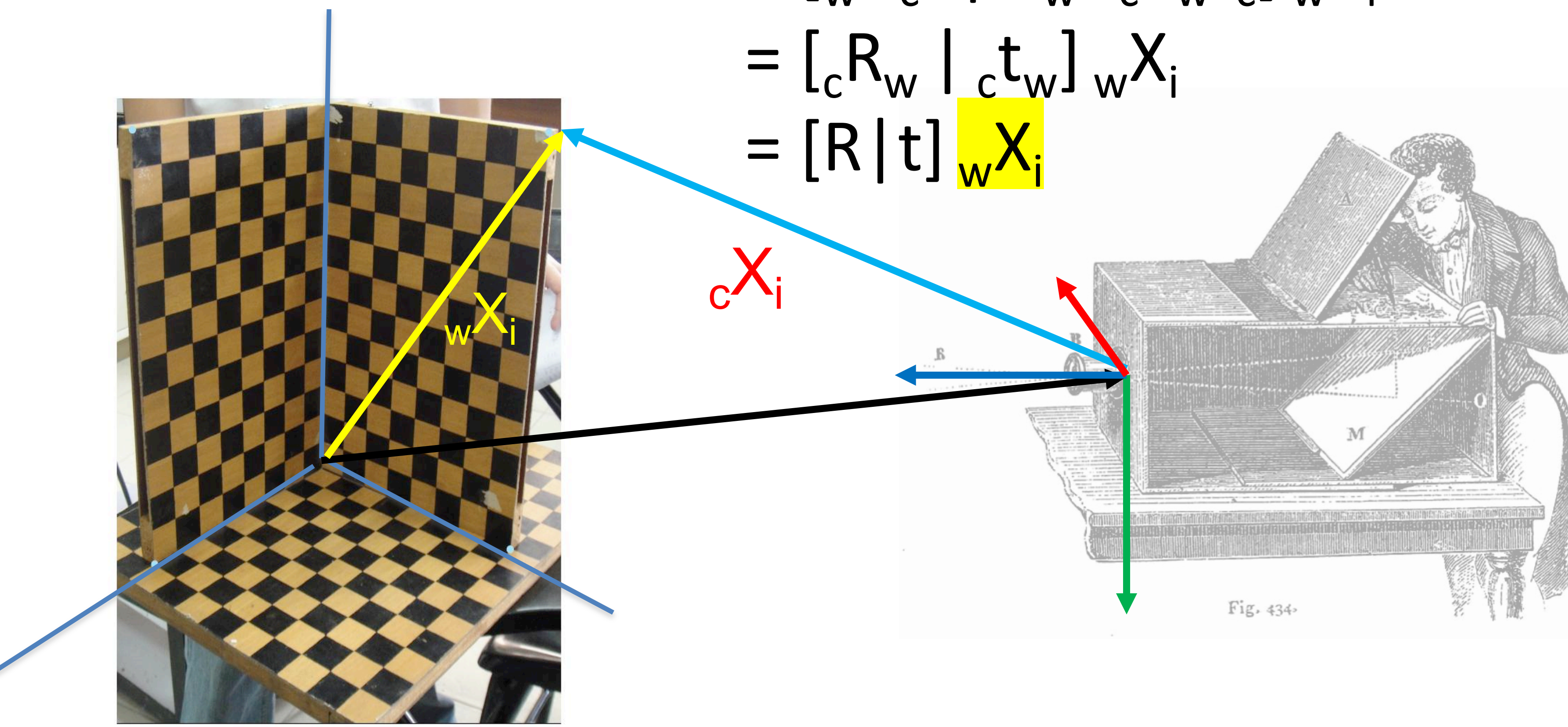
Camera Extrinsics: a Pose in 3D

- What is the geometric meaning of R and t ??
- Transforming point X_i from world to camera coordinates: ${}_wX_i - {}_wt_c = {}_wR_c\,{}_cX_i$



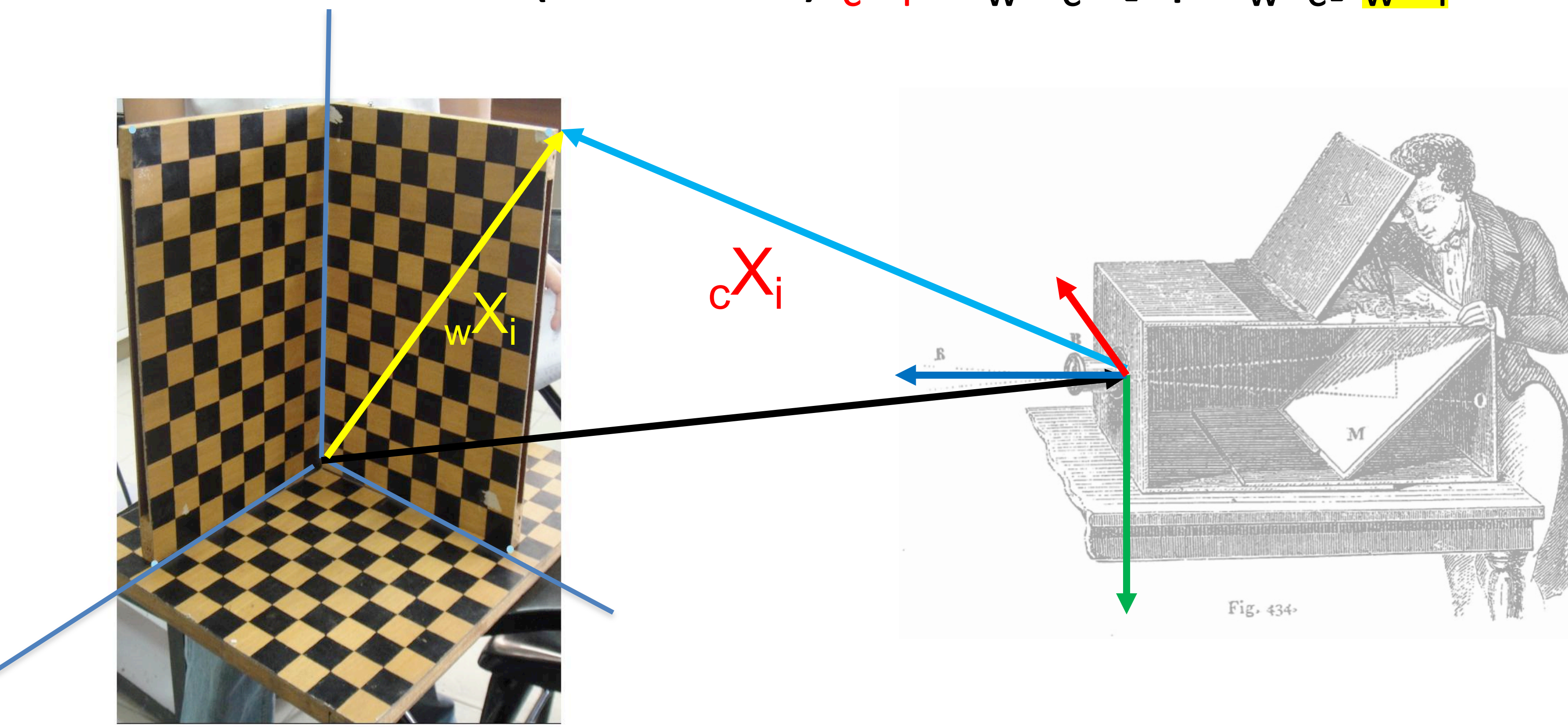
Camera Extrinsics: a Pose in 3D

- Expressed in homogeneous coordinates:
- $$
\begin{aligned}
{}_cX_i &= {}_wR_c^T\,({}_wX_i - {}_wt_c) = {}_wR_c^T\,[I \,|\, -{}_wt_c]\;{}_wX_i \\
&= [{}_wR_c^T \,|\, -{}_wR_c^T\,{}_wt_c]\;{}_wX_i \\
&= [{}_cR_w \,|\, {}_ct_w]\;{}_wX_i \\
&= [R|t]\;{}_wX_i
\end{aligned}
$$



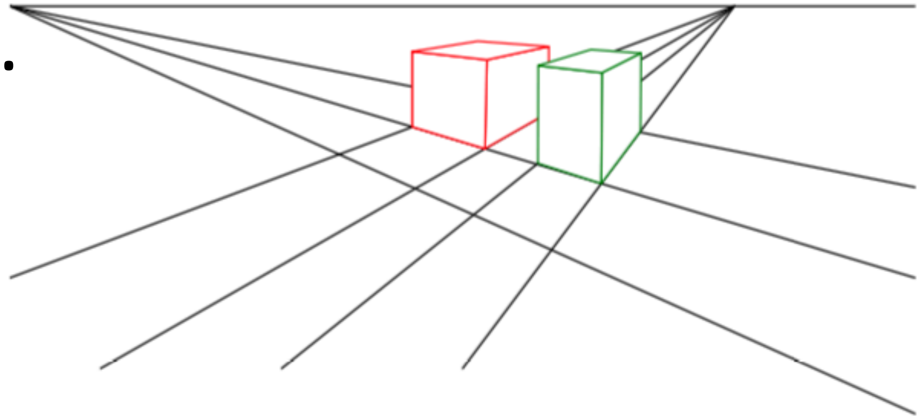
Camera Extrinsics: a Pose in 3D

- Conclusion: when people write ${}_cX_i = [R|t]\ {}_wX_i$ they are talking about (unintuitive) $[{}_cR_w \mid {}_ct_w]$
- We like use (intuitive) ${}_cX_i = {}_wR_c^T [I \mid -{}_wt_c]\ {}_wX_i$



Revision: Projective Camera Matrix

- Homogeneous coord.
- 3D TO 2D projection:



Camera-centric: $x = K[{}_cR_w \mid {}_ct_w]\,X = PX$

World-centric: $x = K\,{}_wR_c^T\,[I \mid -\,{}_wt_c]\,X = PX$

P = **same** 3x4 camera matrix

and K the 3x3 calibration $\boldsymbol{K} = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$

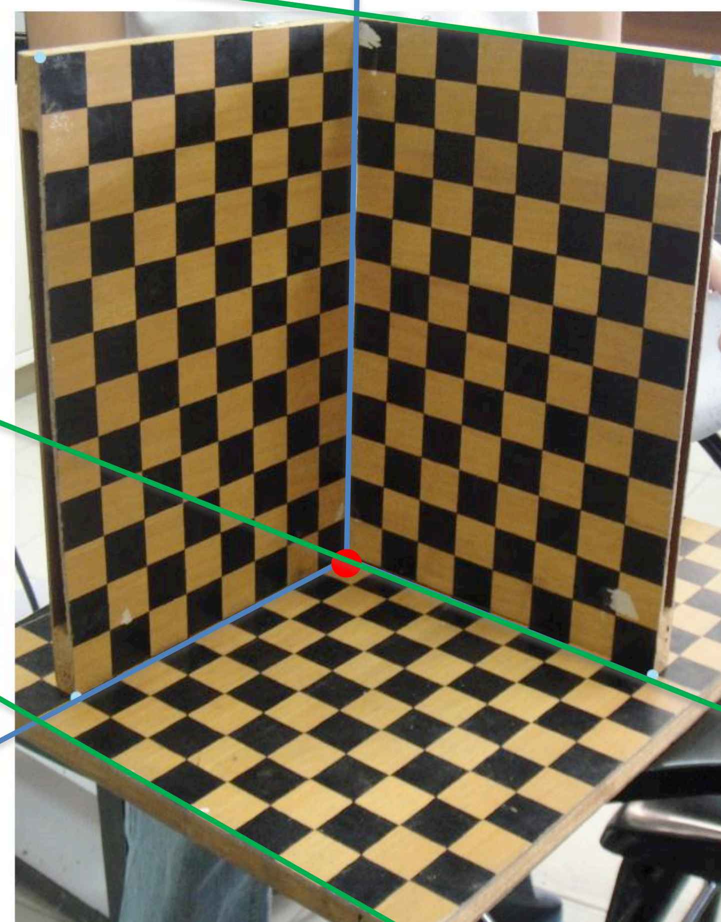
Looking at the (opaque) camera matrix

Can you interpret the columns of P with entities in the scene?

$$P = \left[P^1 \quad P^2 \quad P^3 \quad P^4\right]$$

Answer:
P^1 == the image of [1 0 0 0]
P^2 == the image of [0 1 0 0]
P^3 == the image of [0 0 1 0]
P^4 == the image of [0 0 0 1]

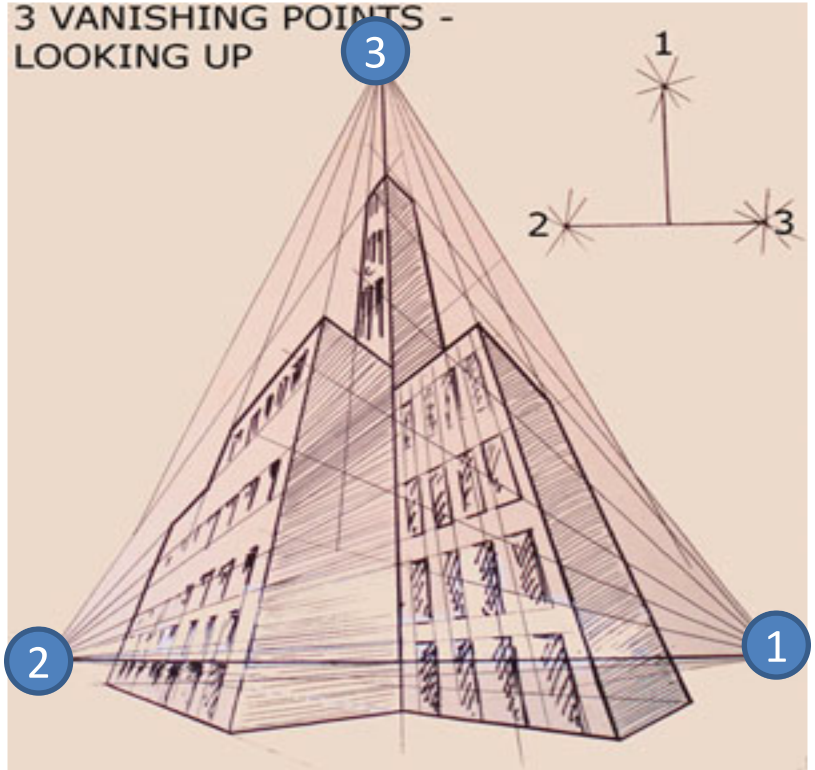
What are those ?
[0 0 0 1] is easy…

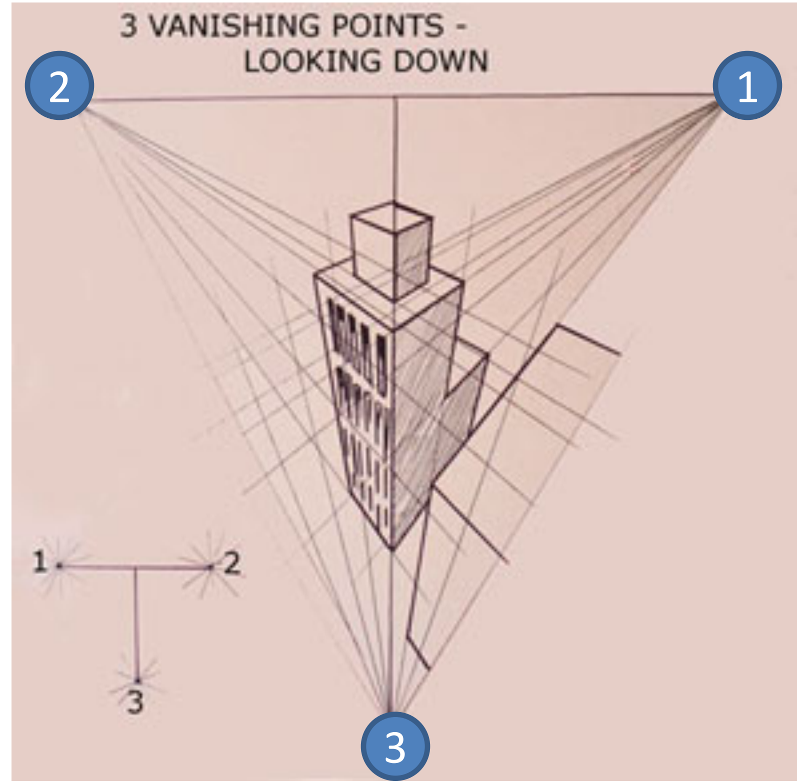


Answer:
[0 0 0 1] is the origin, so P4 is the image of the origin.

[1 0 0 0] is a point at infinity in the X-direction, so it is the vanishing point of all lines parallel with the X direction!

Vanishing points, revisited





Columns of P !

$$P = \begin{bmatrix} P^1 & P^2 & P^3 & P^4 \end{bmatrix}$$

P^4 is arbitrary: wherever you defined the world origin.

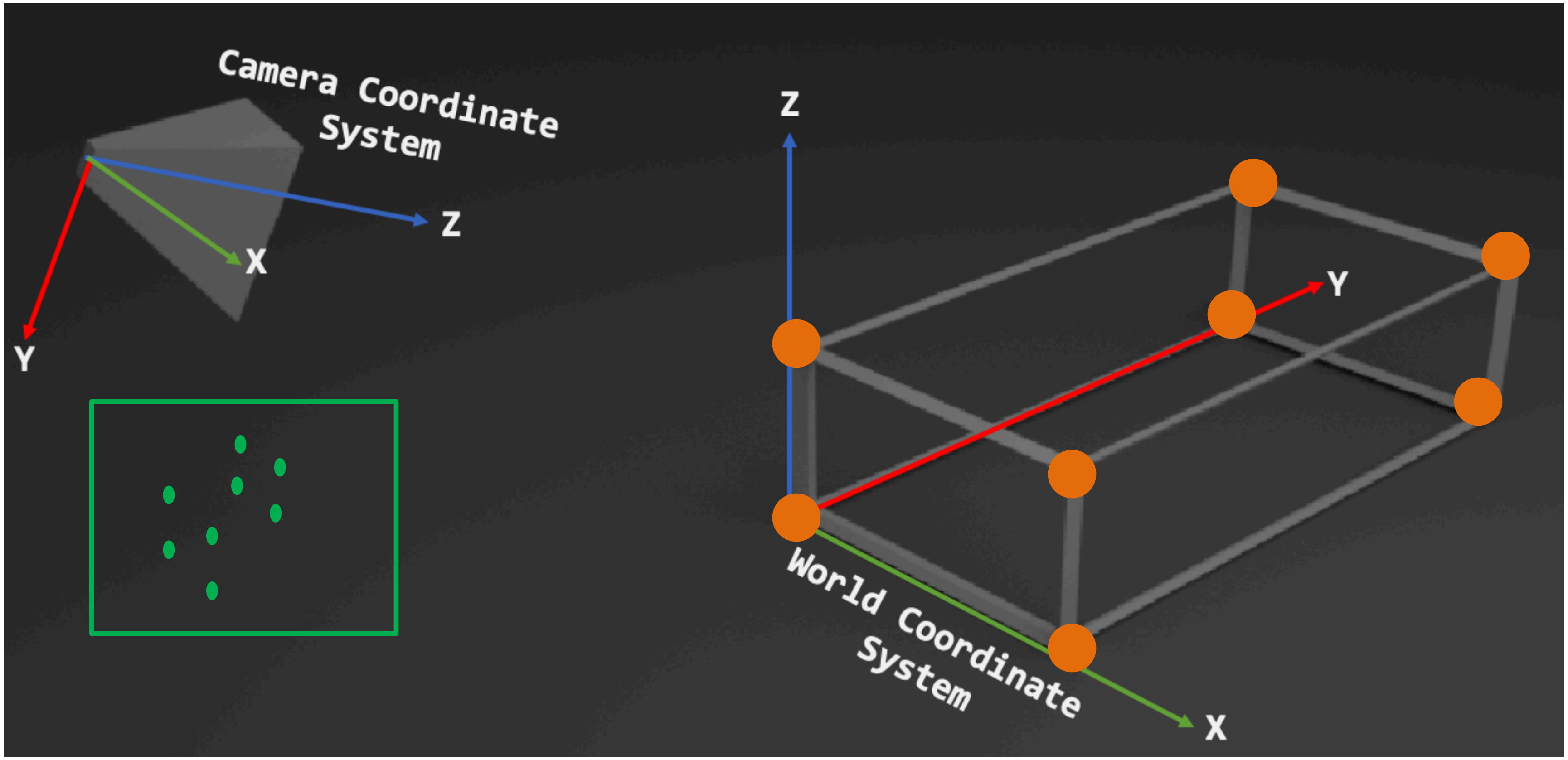
https://www.artinstructionblog.com/perspective-drawing-tutorial-for-artists-part-2

Back to Pose Estimation!

- Simple algorithm: just measure the coordinates of the origin and the three vanishing points?

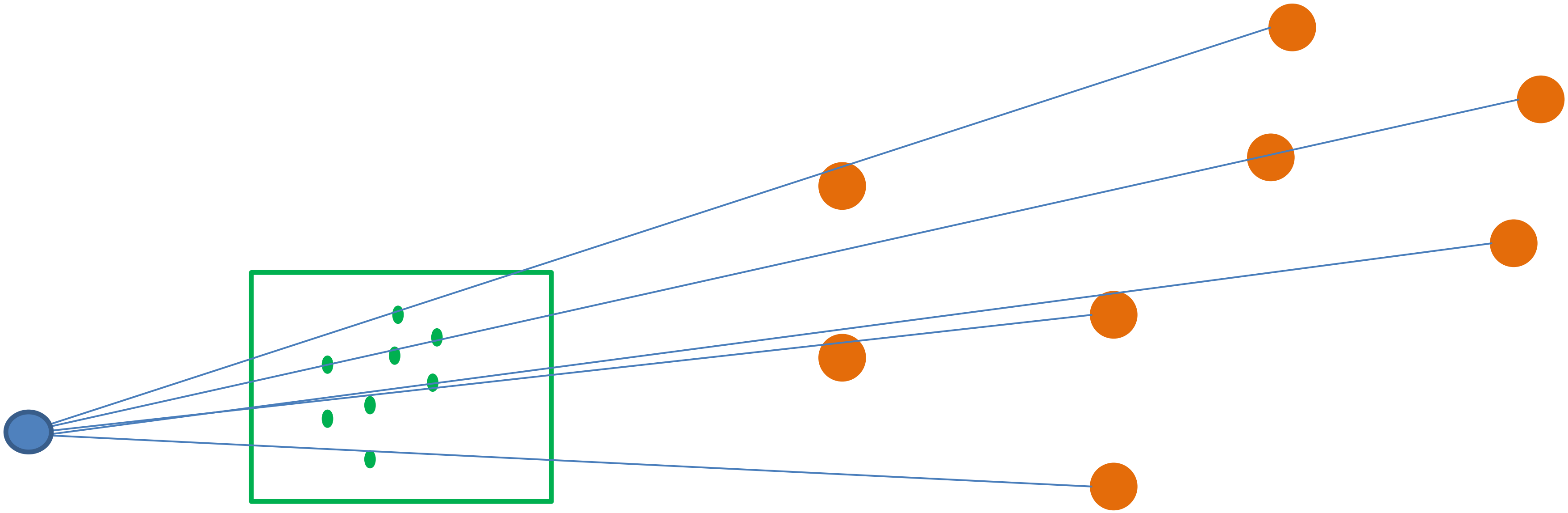
- Does not work ☹:
  - Columns are only measured up to a scale.
  - 4 points * 2DOF = only 8 DOF! Missing 11-8=3
  - 3 missing numbers are exactly those scales.

Least Squares Pose Estimation…



- Input:
  - A set of 2D measurements x_i of known 3D points3D X_i
  - Parametric model is camera matrix P, i.e., $x = f(X; P)$
- Output:
  - Best camera matrix P

Pose estimation = “Resectioning”



$$\mathbf{x} = f(\mathbf{X}_w; \mathbf{P}) = \mathbf{P}\mathbf{X}_w = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \cong \begin{bmatrix} s \cdot u \\ s \cdot v \\ s \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}$$

$$\arg\min_{\hat{\mathbf{P}}} \sum_{i=1}^{N} ||\hat{\mathbf{P}}\mathbf{X}_w^i - \mathbf{x}^i||_2.$$

- Opposite of triangulation.

Pose estimation

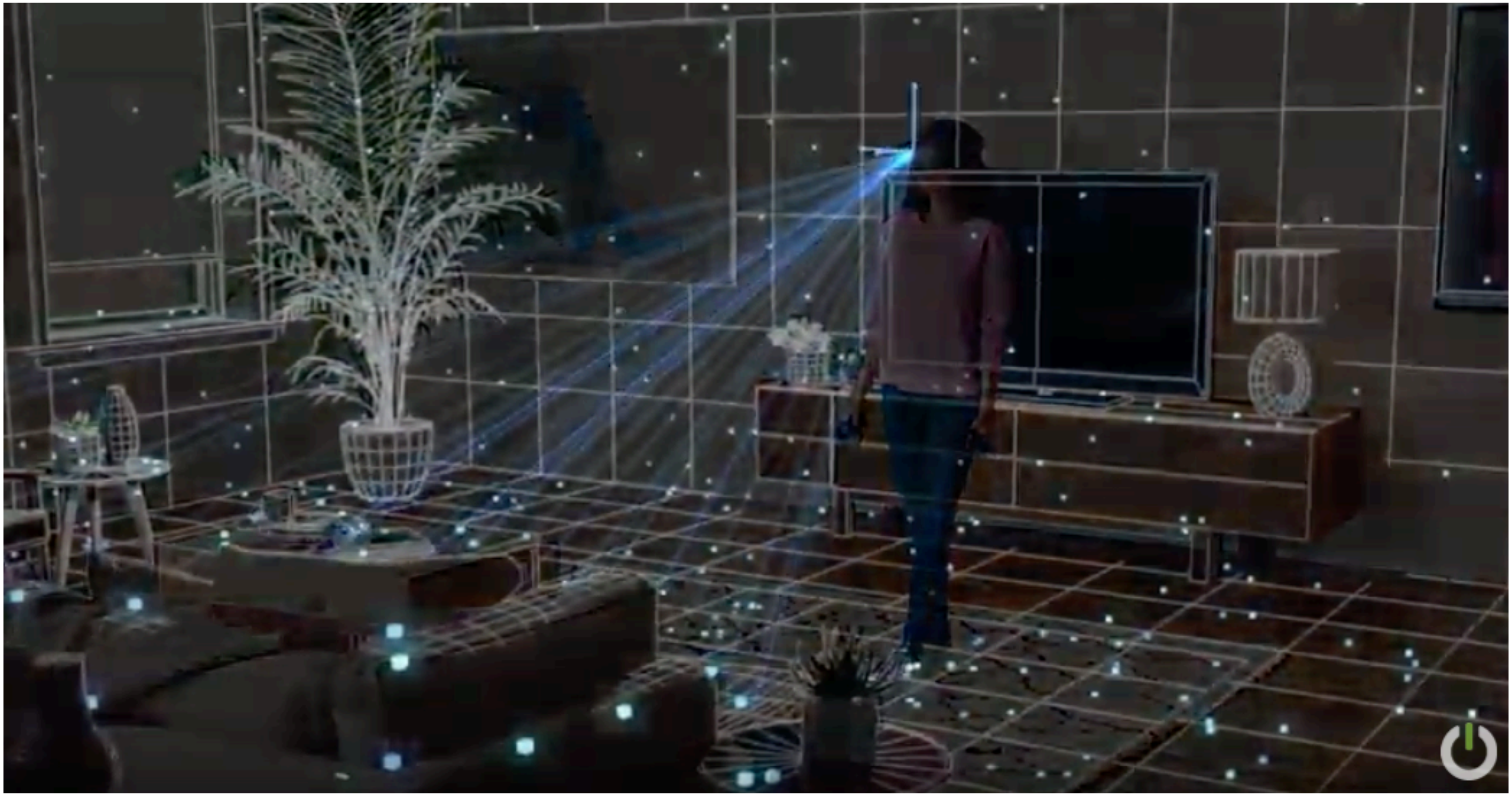
$$\arg\min_{\hat{\mathbf{P}}} \sum_{i=1}^{N} ||\hat{\mathbf{P}}\mathbf{X}_w^i - \mathbf{x}^i||_2.$$

- In project 3, you will use `scipy.optimize.least_squares` to do exactly that. Working knowledge of 3D poses will be required.
- Note before we compute the 2D reprojection error we need to convert back *PX* to non-homogeneous coordinates:

$$x_i = \frac{p_{00}X_i + p_{01}Y_i + p_{02}Z_i + p_{03}}{p_{20}X_i + p_{21}Y_i + p_{22}Z_i + p_{23}}$$
$$y_i = \frac{p_{10}X_i + p_{11}Y_i + p_{12}Z_i + p_{13}}{p_{20}X_i + p_{21}Y_i + p_{22}Z_i + p_{23}}$$

https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.least_squares.html

Application: VR



https://youtu.be/nrj3JE-NHMw