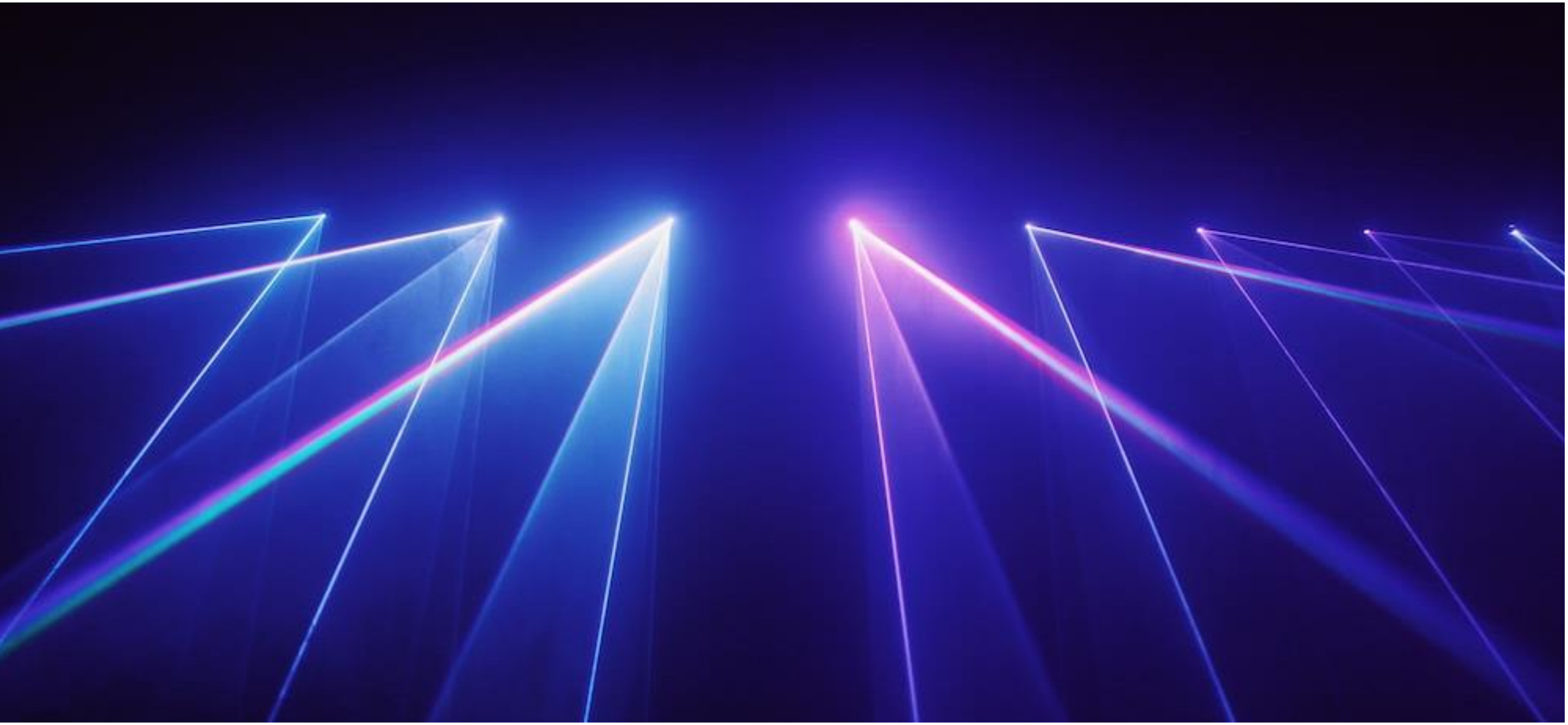
Epipolar Geometry + Calibration

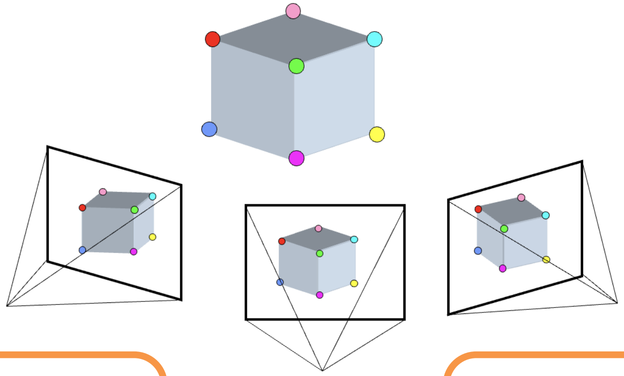


A lot of slides from Noah Snavely +
Shree Nayar's YT series: First principals of Computer Vision

CS180: Intro to Computer Vision and Comp. Photo
Angjoo Kanazawa & Alexei Efros, UC Berkeley, Fall 2025

Many problems in 3D

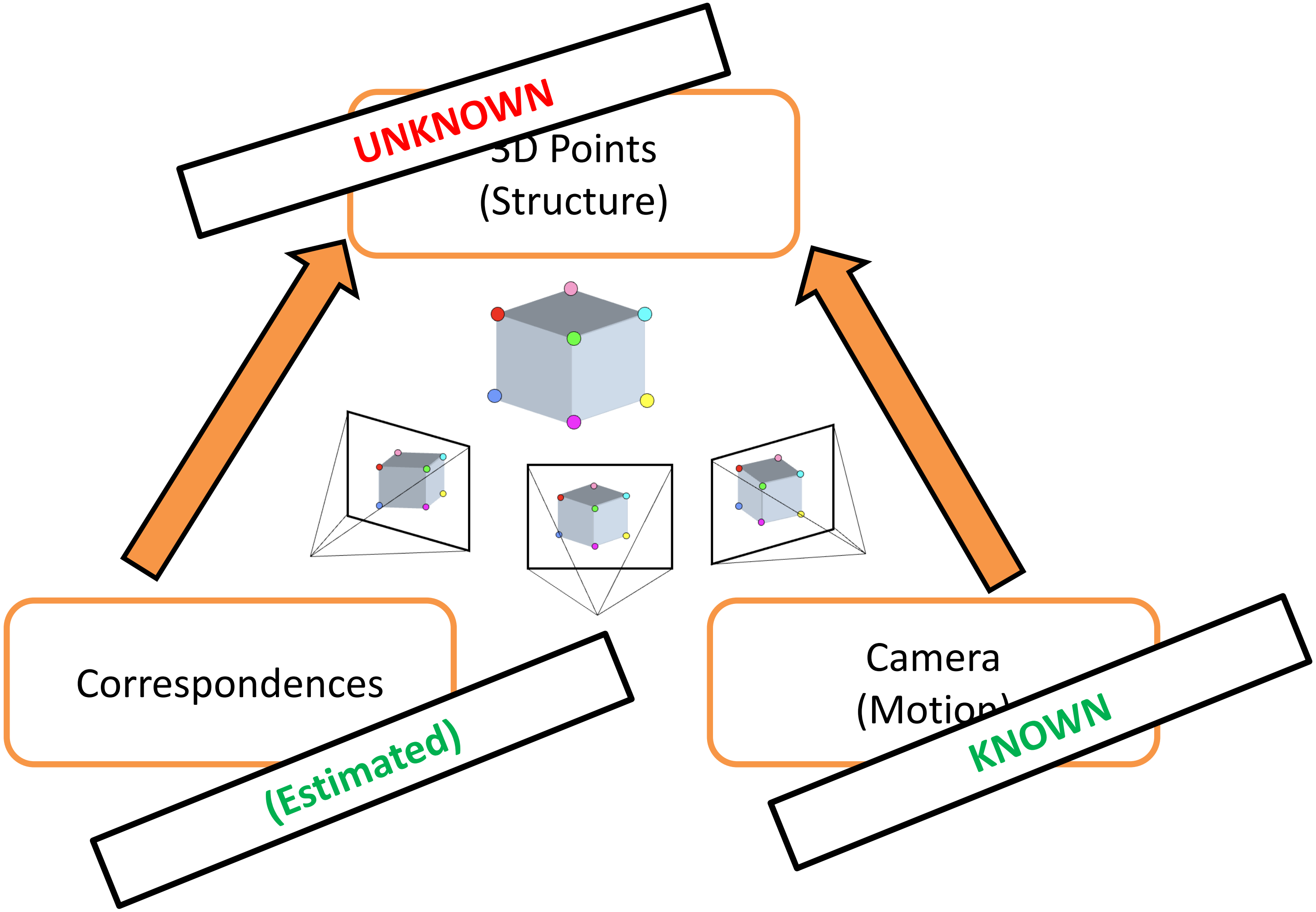
3D Points
(Structure)



Correspondences

Camera
(Motion)

Simple Stereo;
Corresp + Camera = Disparity = depth^{-1}



UNKNOWN

3D Points
(Structure)

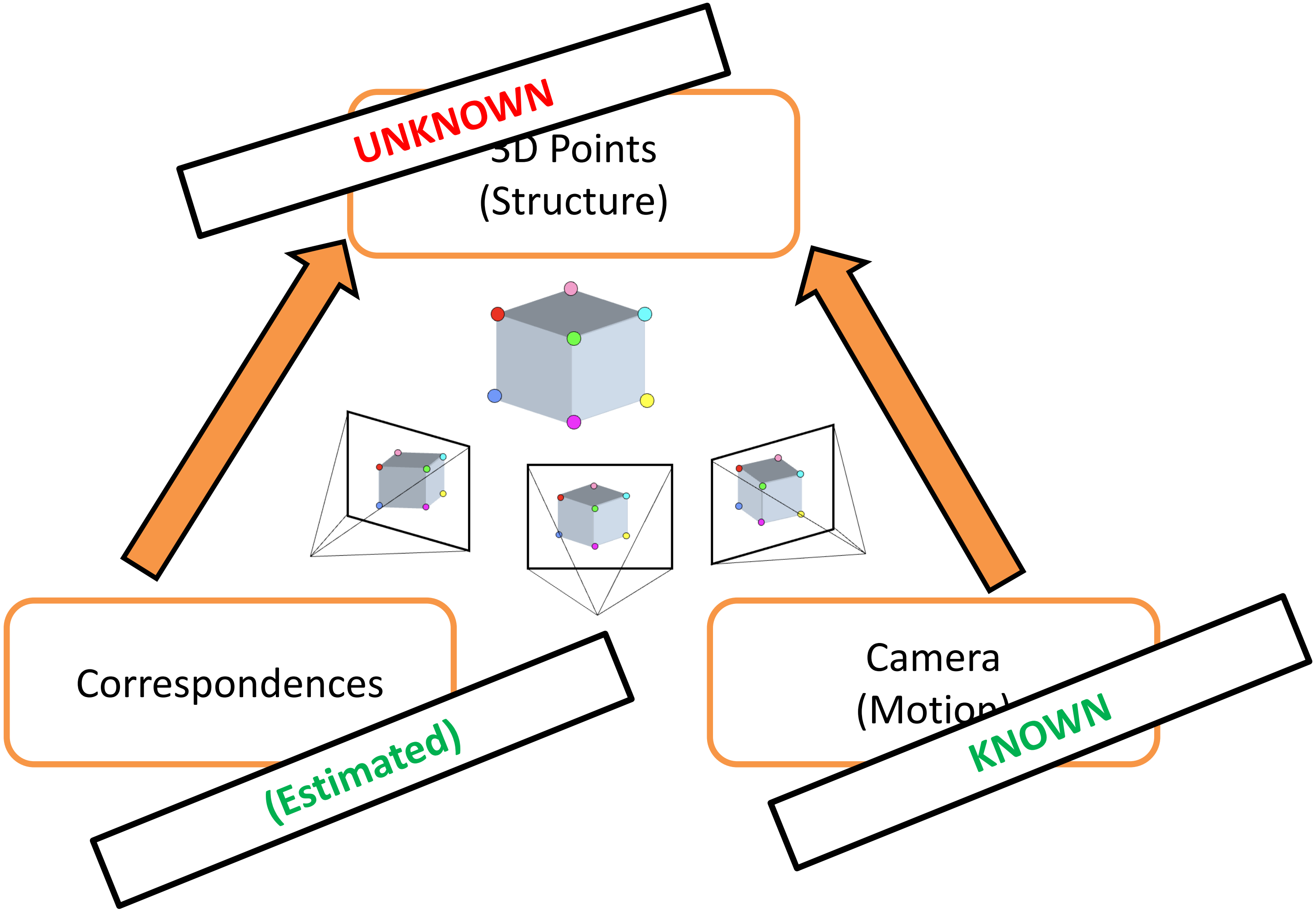
Correspondences

(Estimated)

Camera
(Motion)

KNOWN

Today: Arbitrary Stereo;
Corresp + Camera – Triangulation = depth



UNKNOWN

3D Points
(Structure)

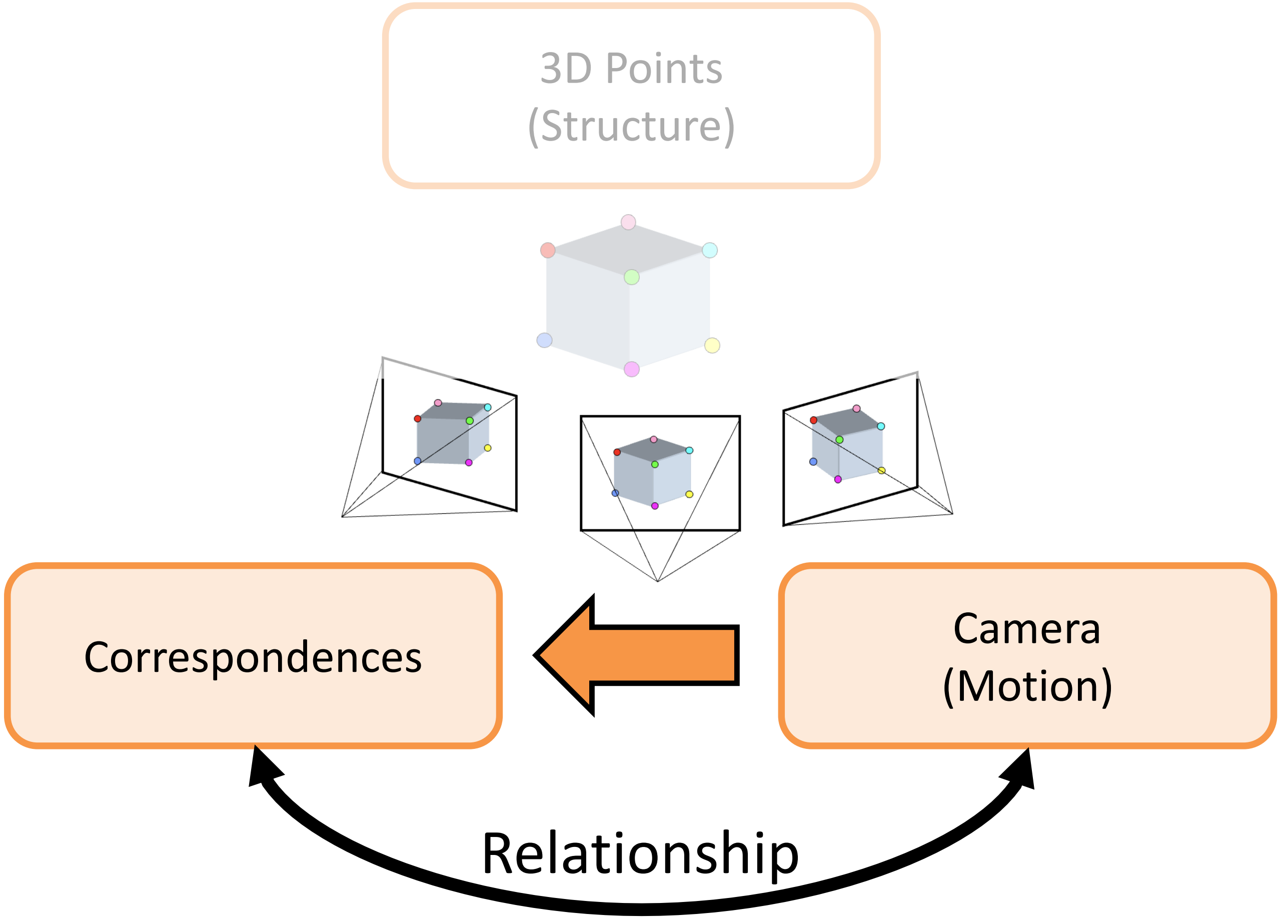
Correspondences

(Estimated)

Camera
(Motion)

KNOWN

Camera helps Correspondence: **Epipolar Geometry**



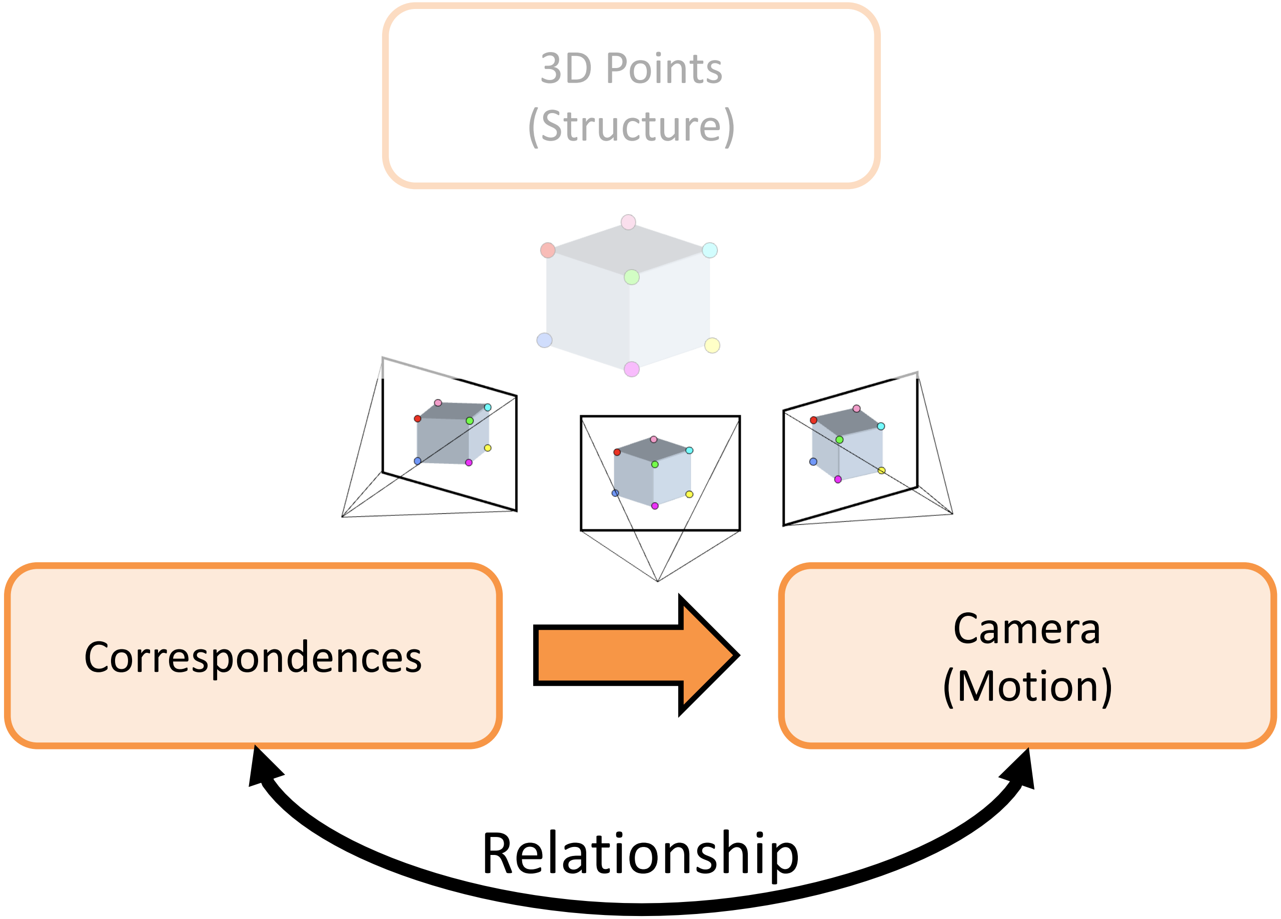
3D Points
(Structure)

Correspondences

Camera
(Motion)

Relationship

Correspondence gives camera: **Epipolar Geometry**



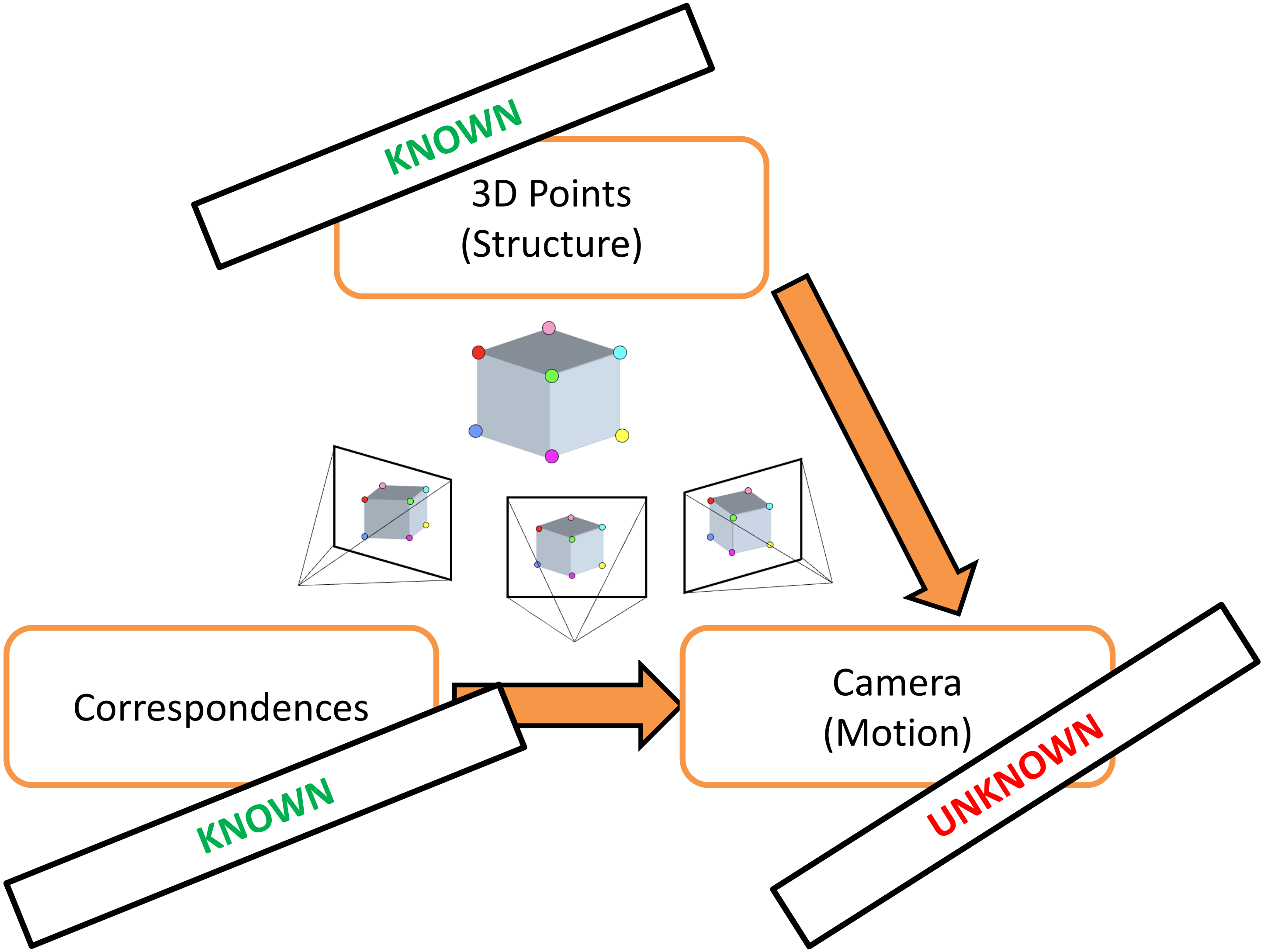
3D Points
(Structure)

Correspondences

Camera
(Motion)

Relationship

Next: Camera Calibration



KNOWN

3D Points
(Structure)

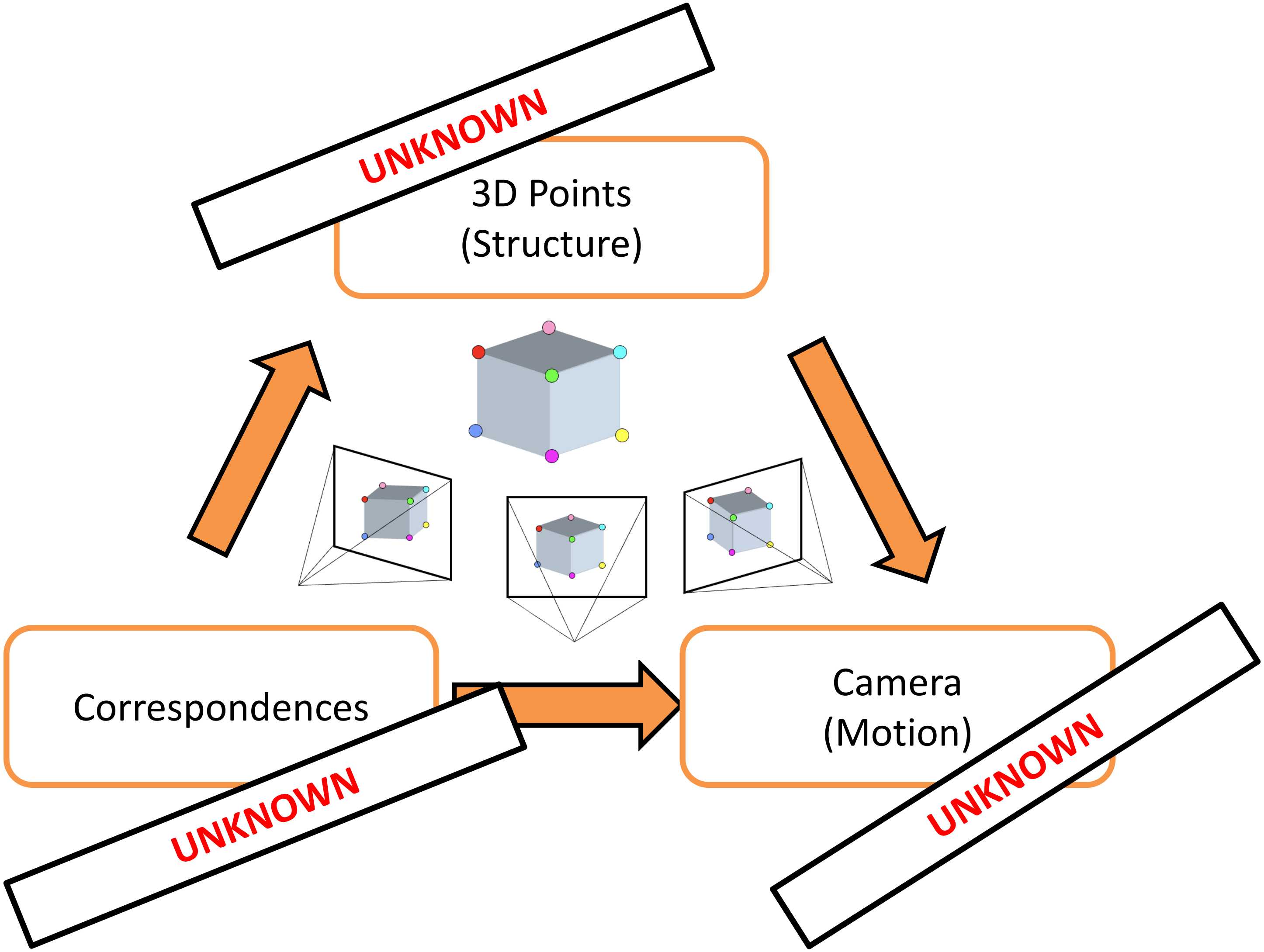
Correspondences

KNOWN

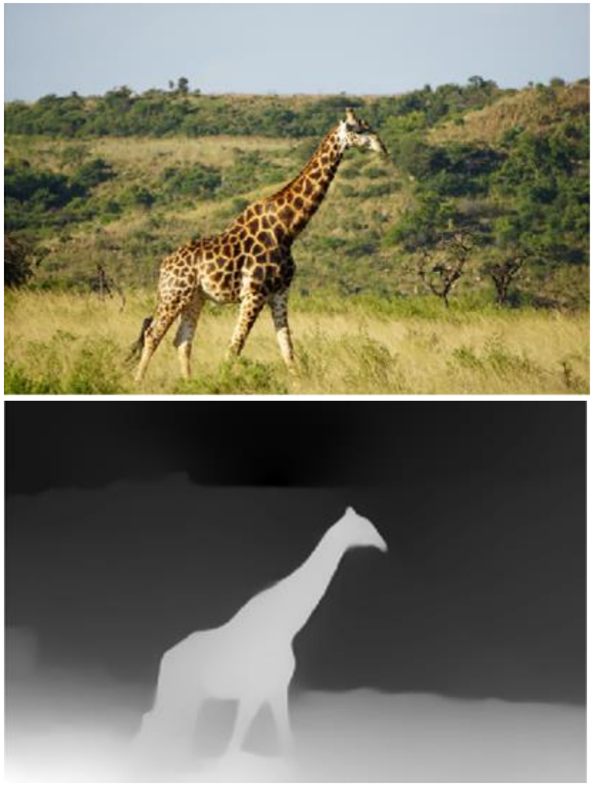
Camera
(Motion)

UNKNOWN

After that: Structure-from-Motion Everything* Unknown

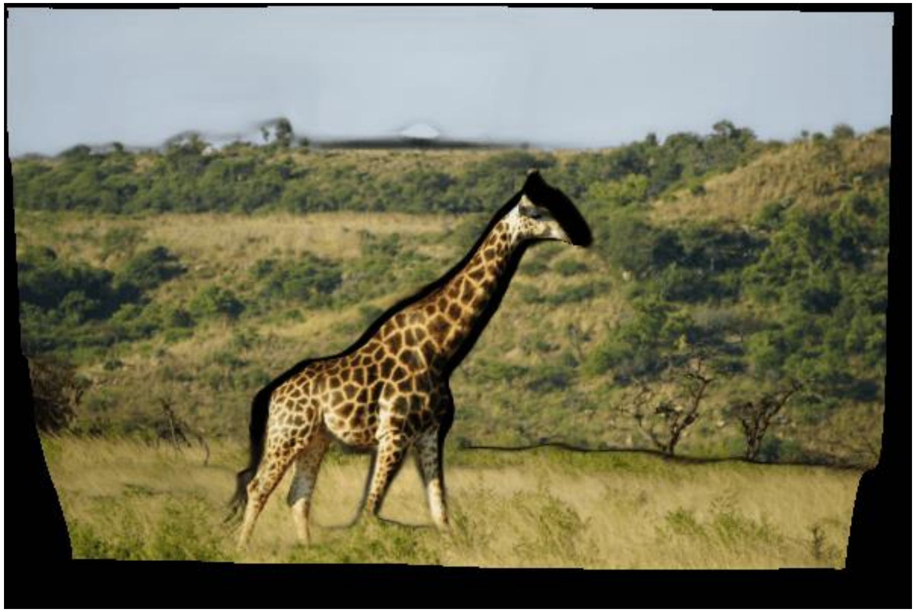


What Depth Map provides



Monocular Depth Prediction [Ranftl et al. PAMI'20]

warping the pixel based on its depth as you change the views



More cool things with Depth

Niklaus et al. ToG 2019

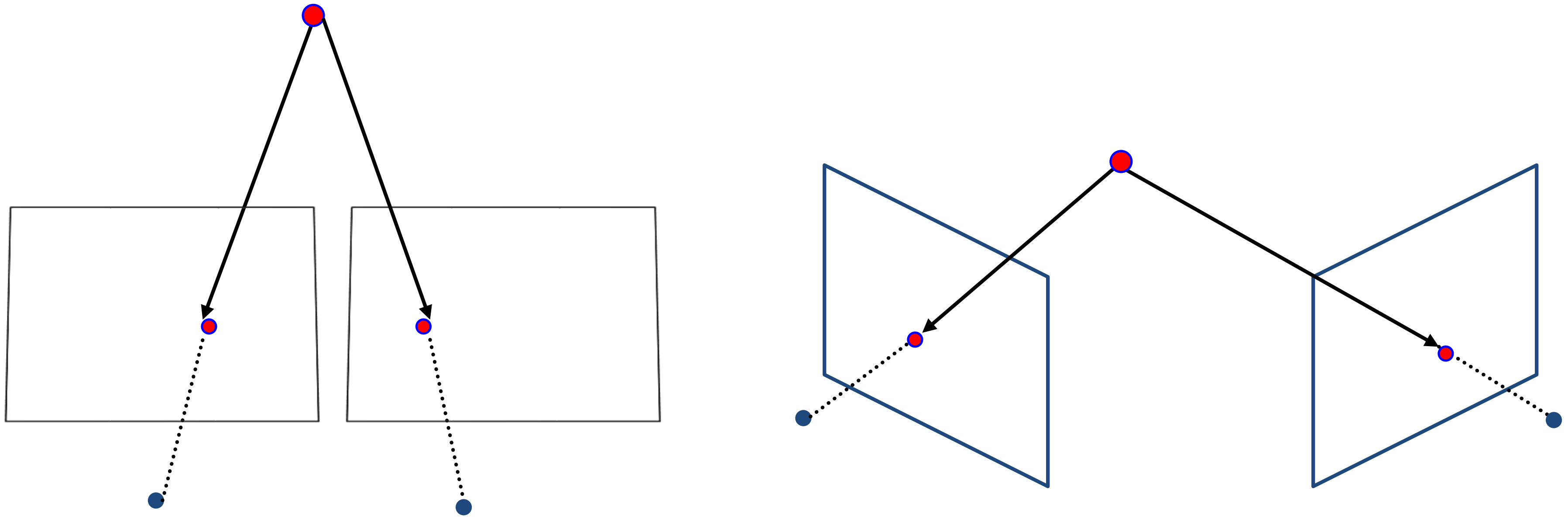
Shih et al. CVPR 2020

3D photo

AR

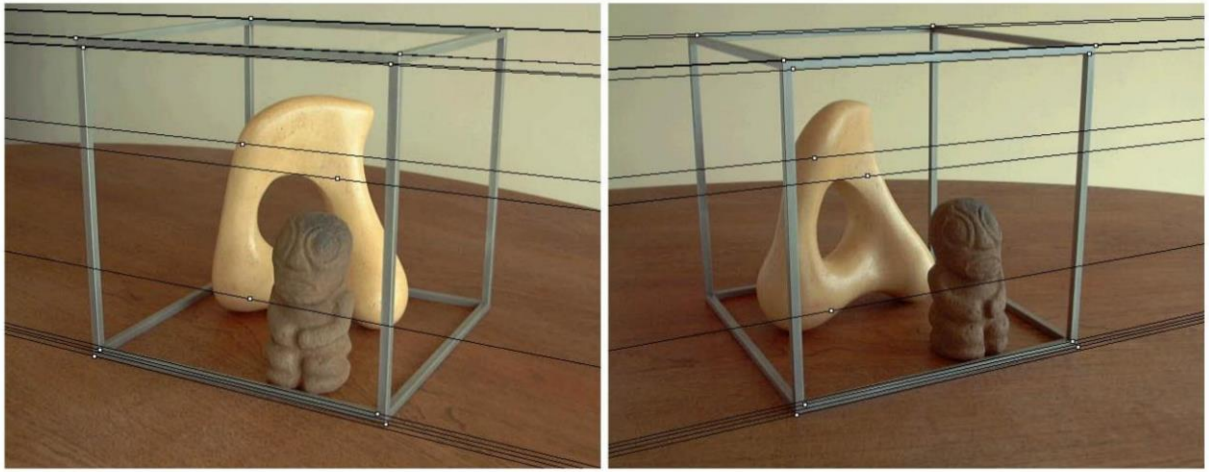
Next: General case

- The two cameras need not have parallel optical axes.
- Assume camera intrinsics are calibrated



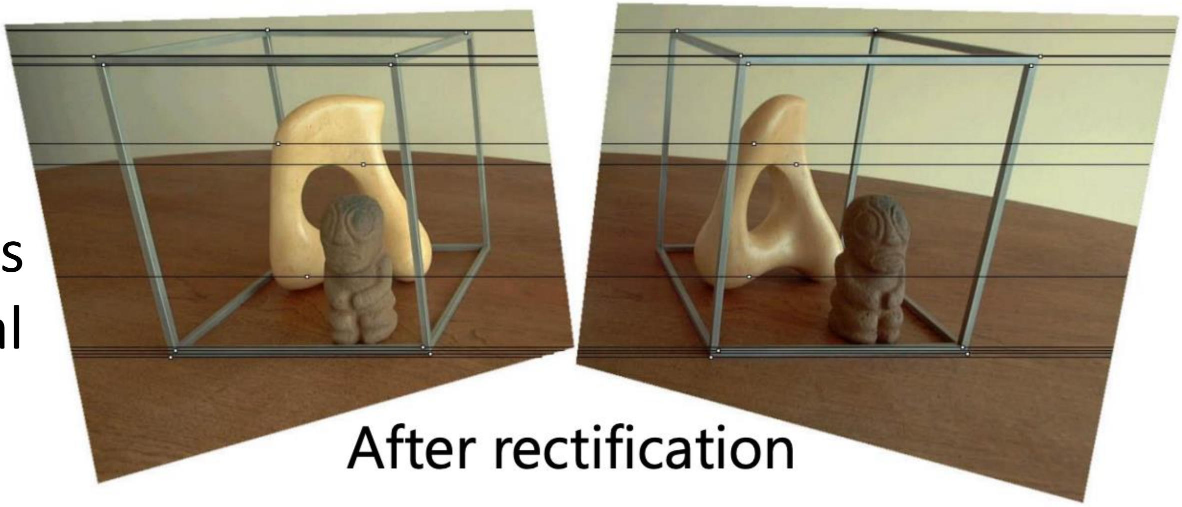
Same hammer:
Find the correspondences, then solve for structure

Option 1: Rectify via homography



Original stereo pair

Then find correspondences on the horizontal scan line



After rectification

Slide courtesy of Noah Snavely

General case, known camera, find depth: Option 2

1. Find correspondences
2. Triangulate

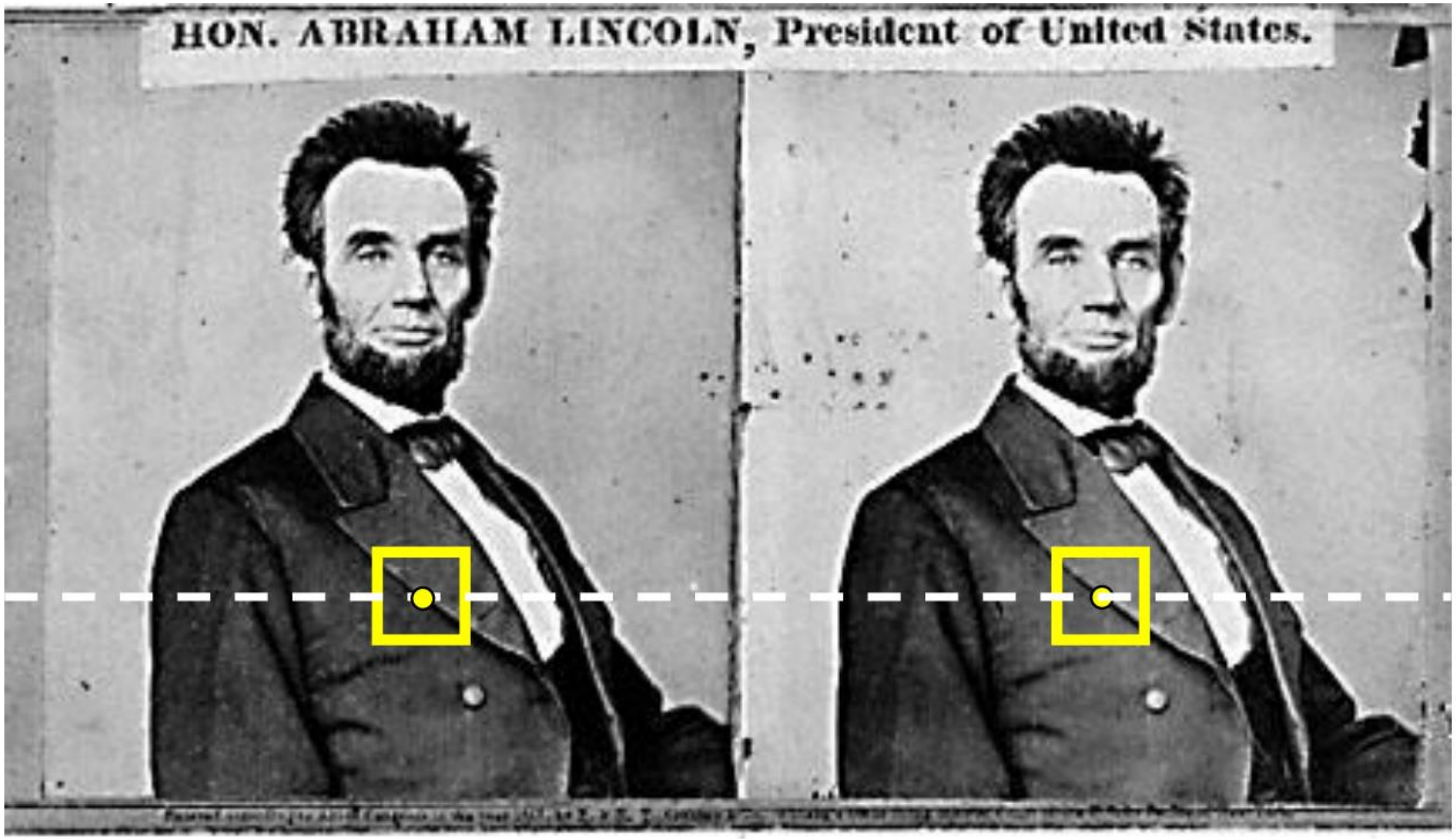
General case, known camera, find depth: Option 2

1. **Find correspondences**
2. Triangulate

Can we restrict the search space again to 1D?

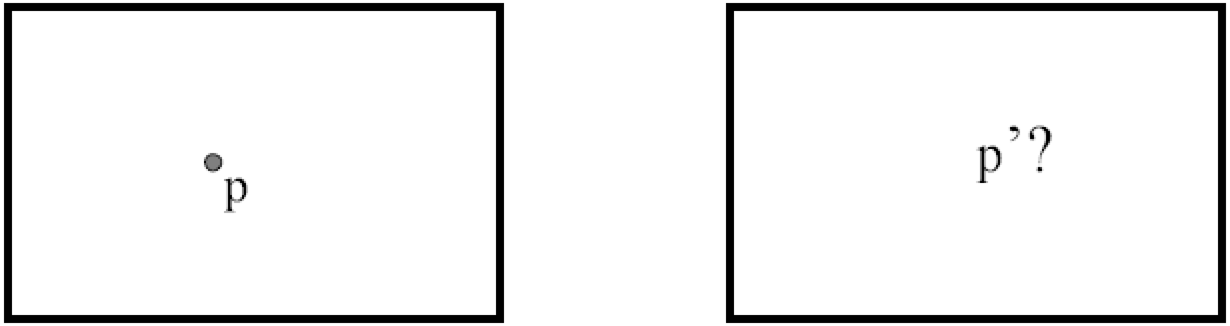
What is the relationship between the camera + the corresponding points?

Where do epipolar lines come from?



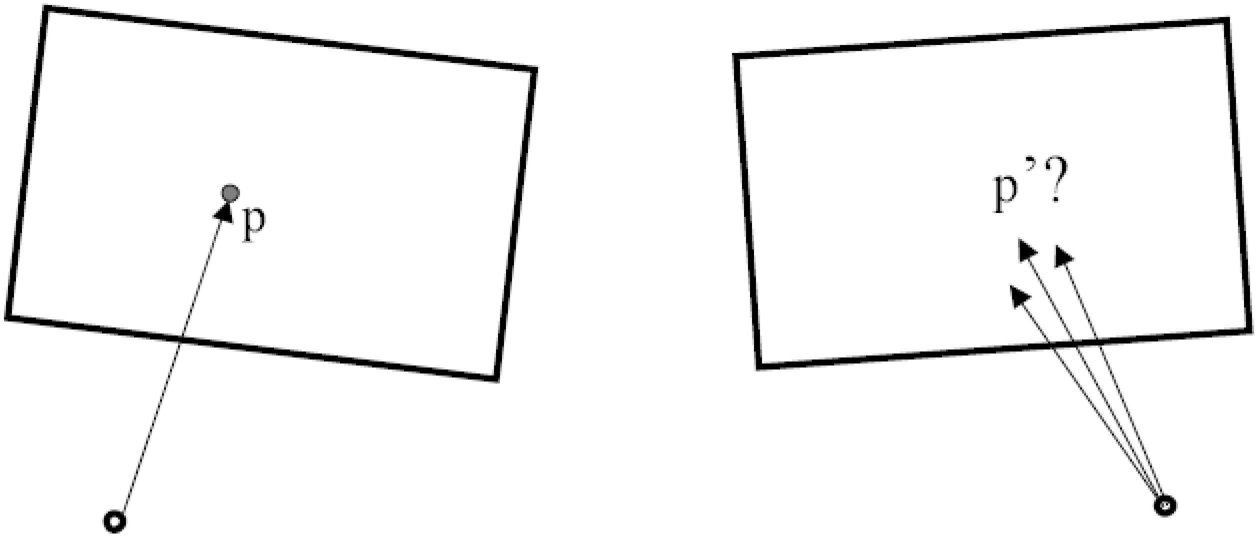
Slide courtesy of Noah Snavely

Stereo correspondence constraints



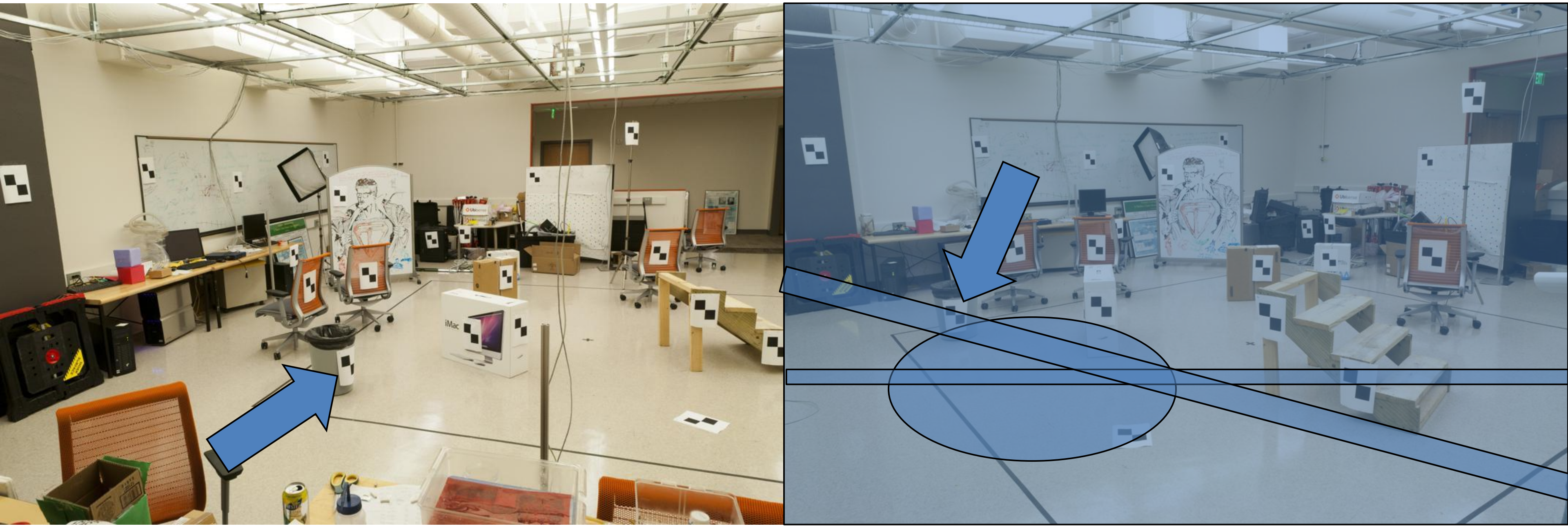
- Given p in left image, where can corresponding point p’ be?

Stereo correspondence constraints



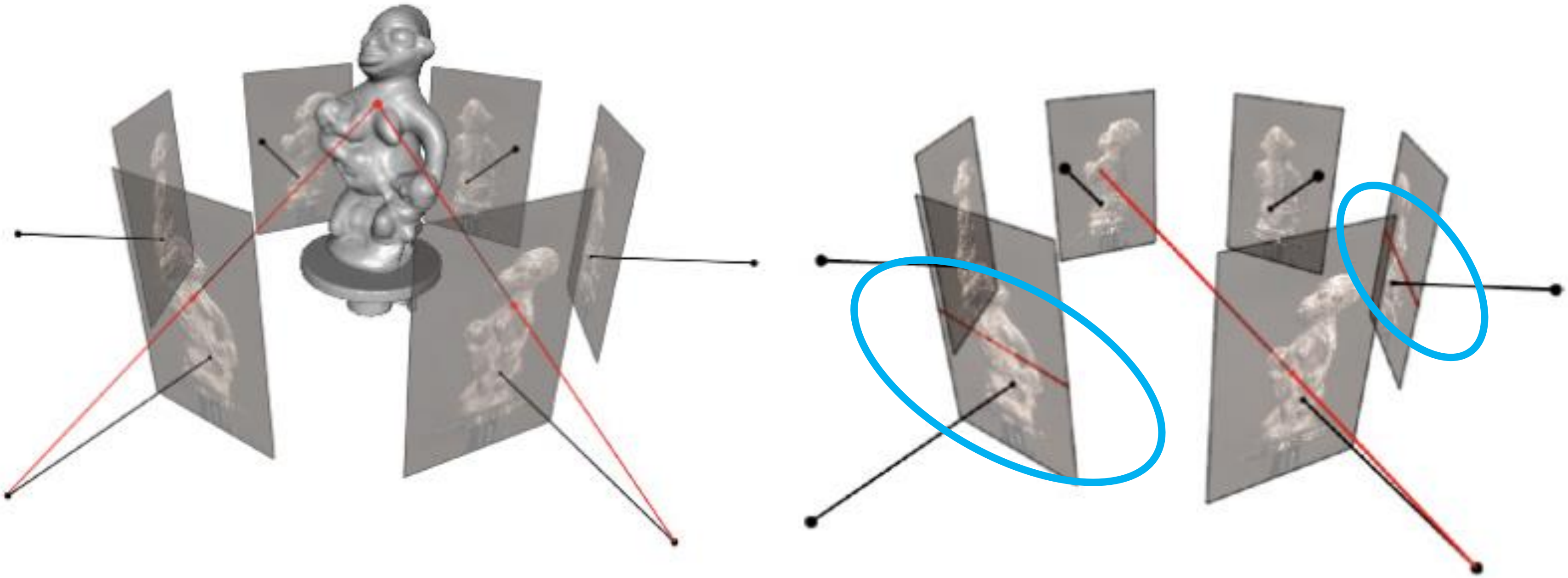
- Given p in left image, where can corresponding point p’ be?

Where do we need to search?



Slide by James Hays

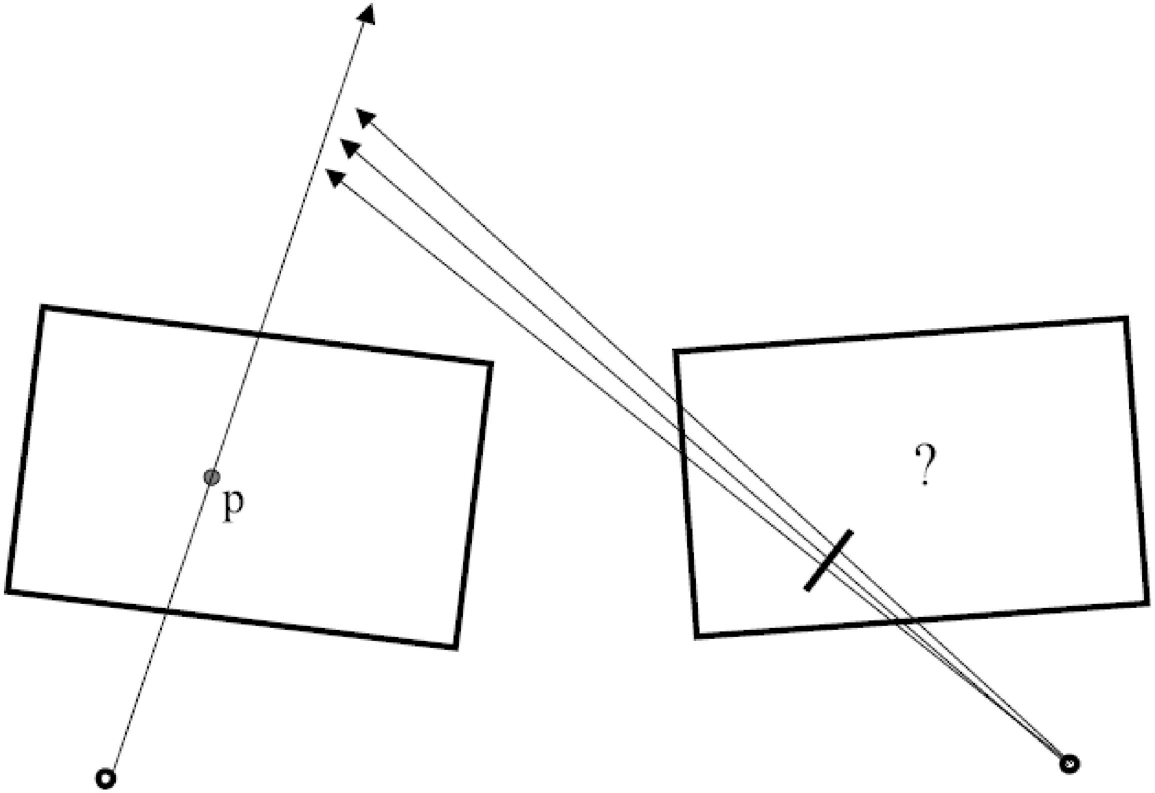
Epipolar Geometry



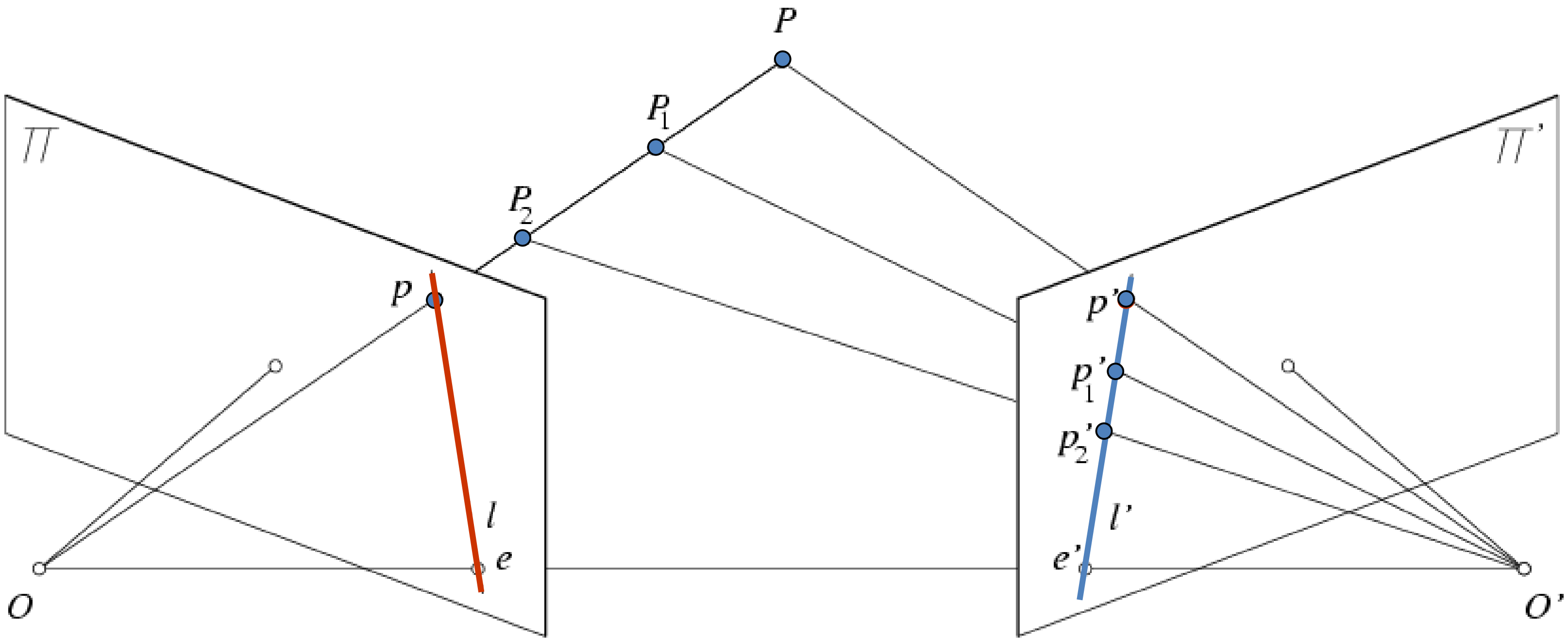
Figures by Carlos Hernandez

If you get confused with the following math, look at this picture again, it just describes this.

Stereo correspondence constraints



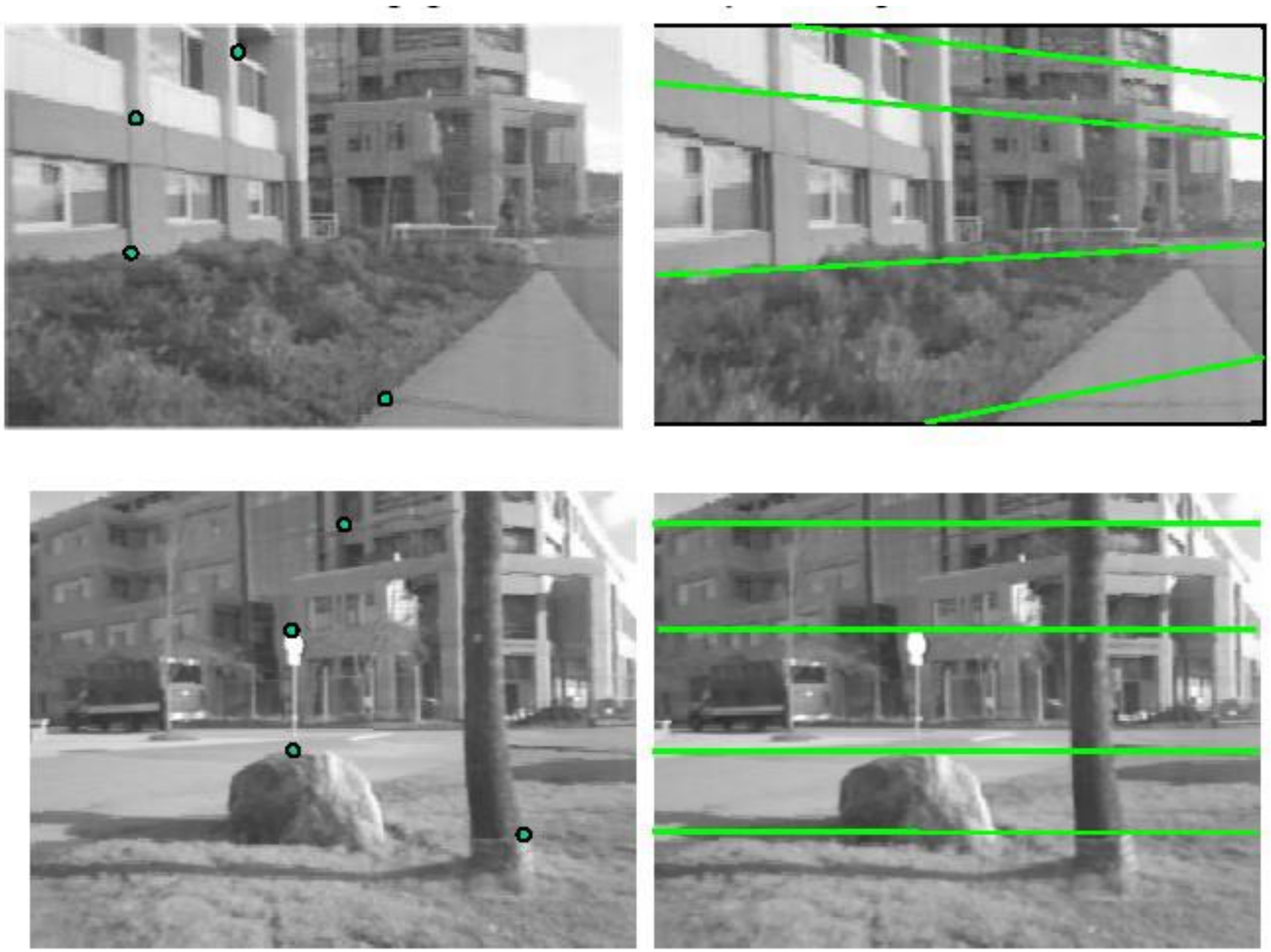
Epipolar constraint



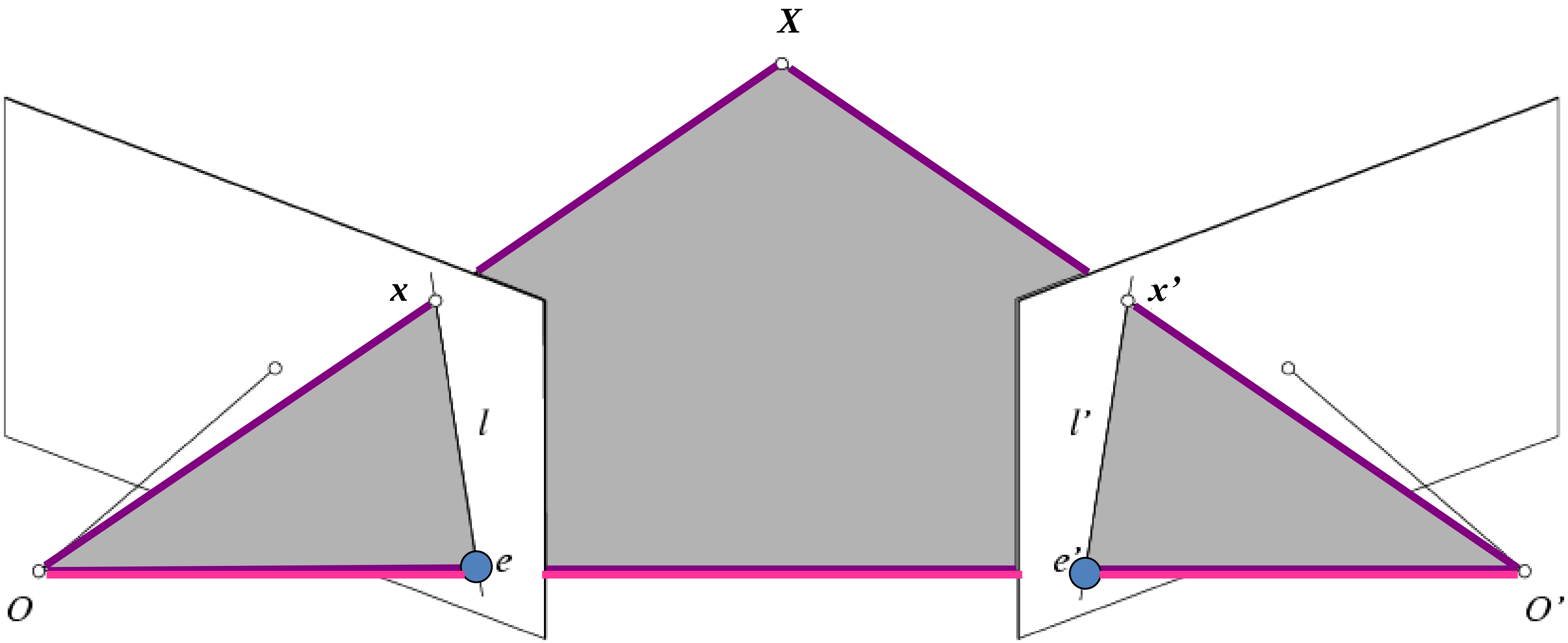
- Potential matches for *p* have to lie on the corresponding epipolar line *l'*.
- Potential matches for *p'* have to lie on the corresponding epipolar line *l*.

Source: M. Pollefeys

Example



Parts of Epipolar geometry



- **Baseline** – line connecting the two camera centers
- **Epipolar Plane** – plane containing baseline (1D family)
- **Epipoles**

= intersections of baseline with image planes
= projections of the other camera center
= vanishing points of the baseline

The Epipole

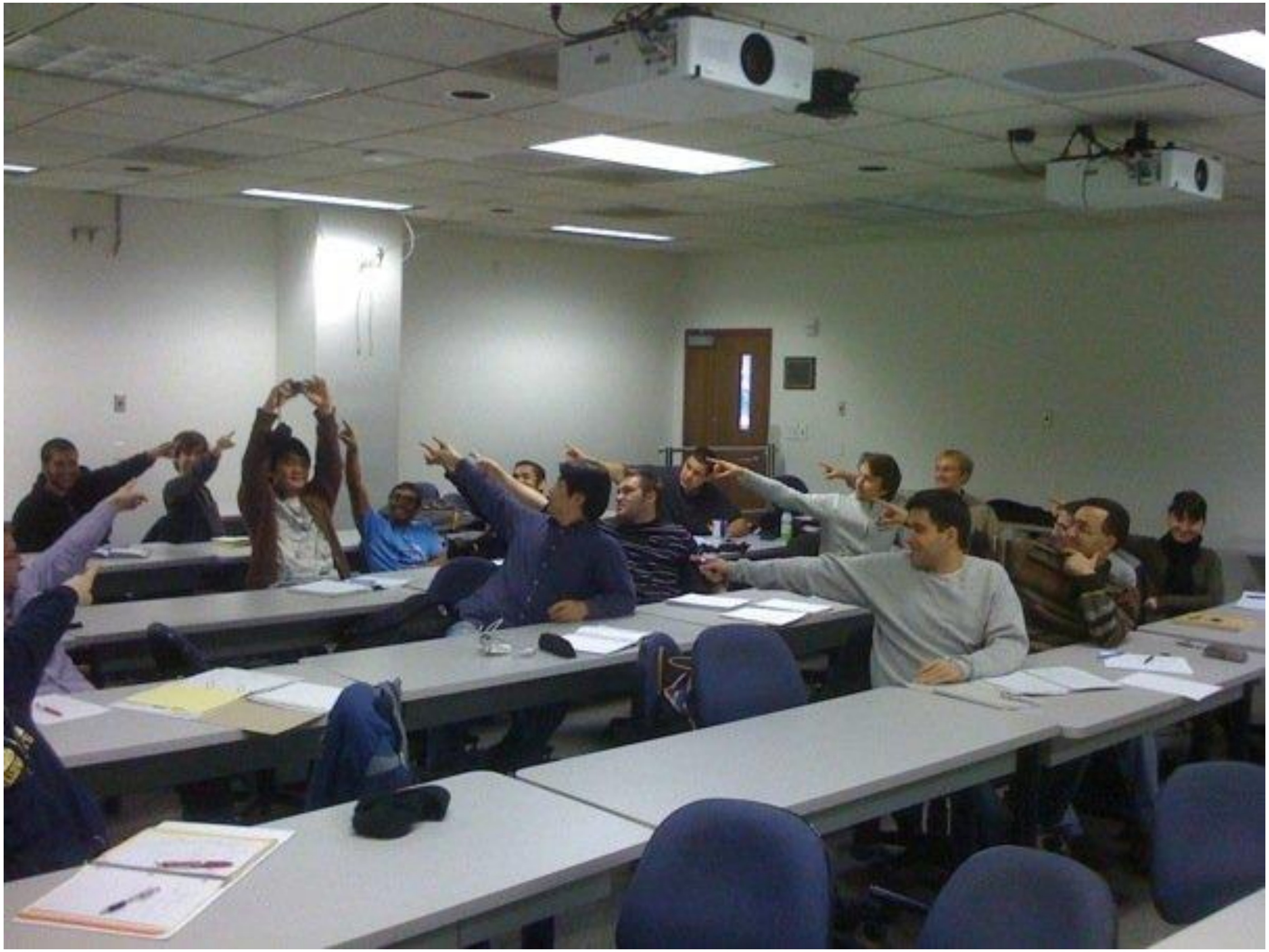
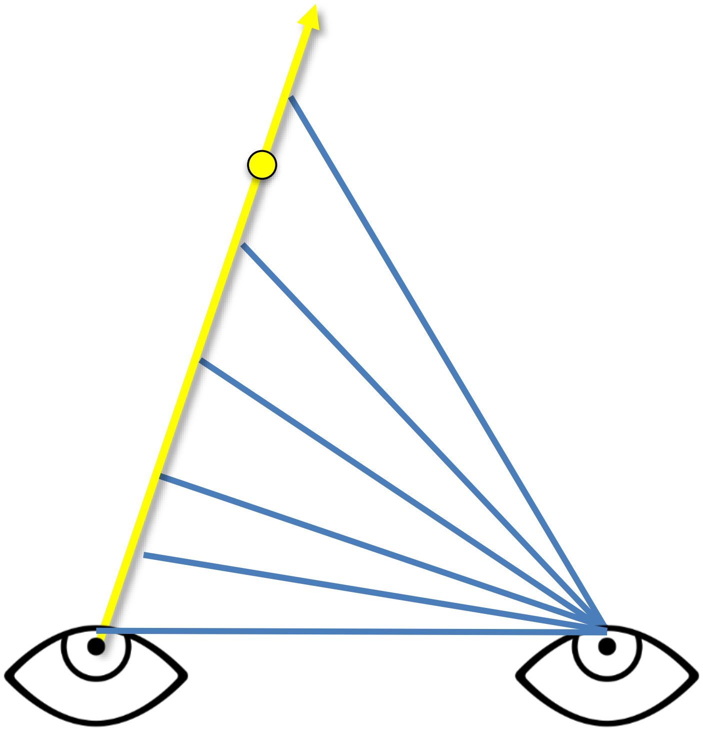
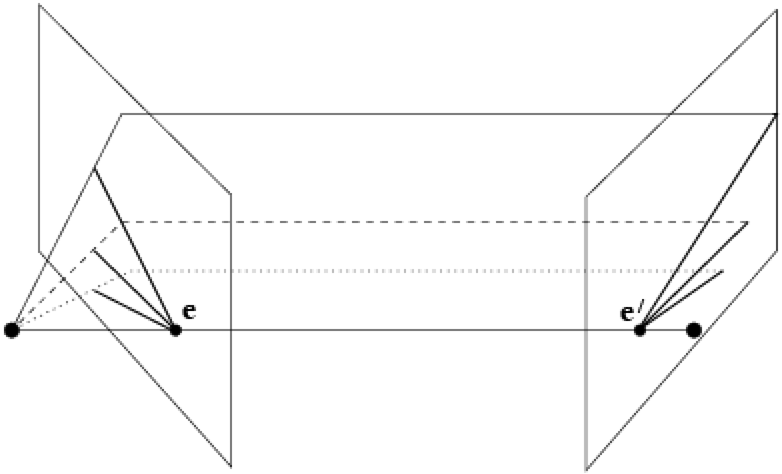


Photo by Frank Dellaert

The epipolar line has to go through the epipole



Example: converging cameras



As position of 3d point varies, epipolar lines “rotate” about the baseline

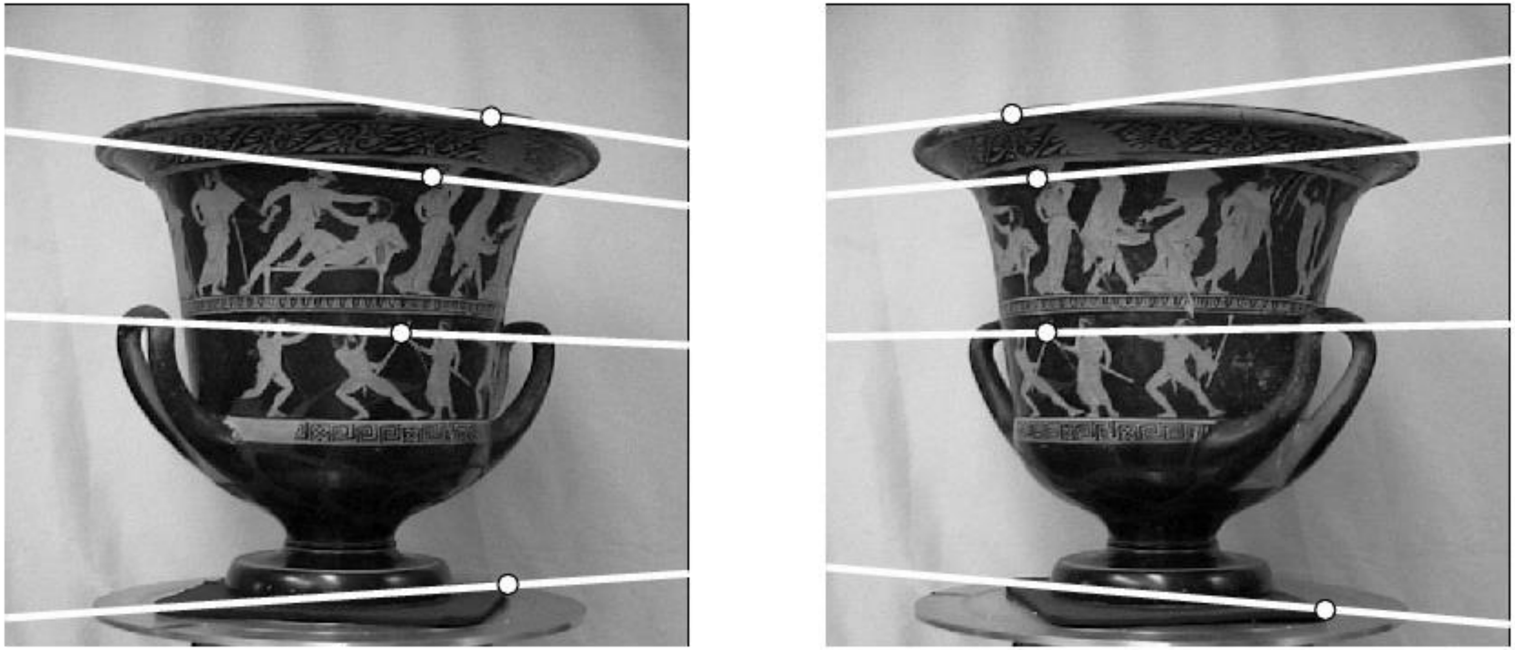
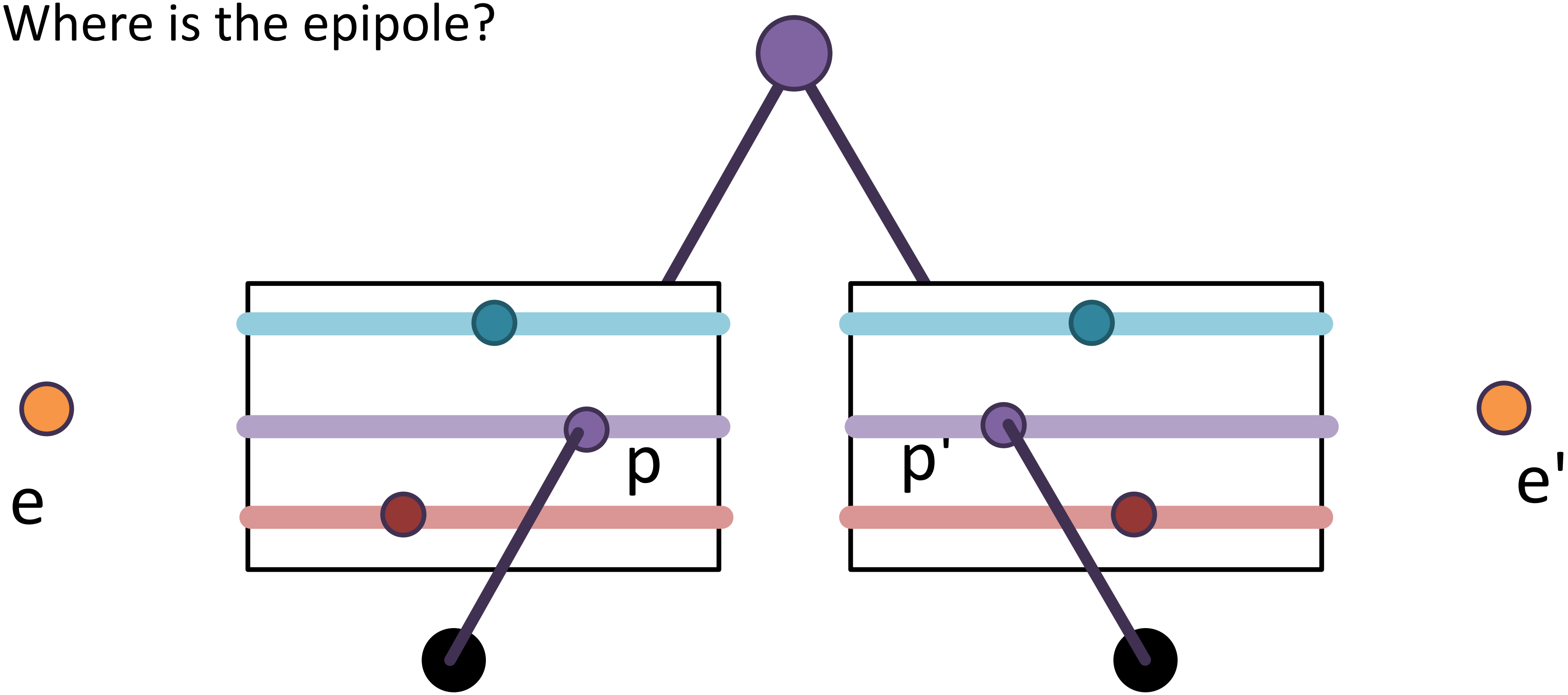


Figure from Hartley & Zisserman

Example: Parallel to Image Plane

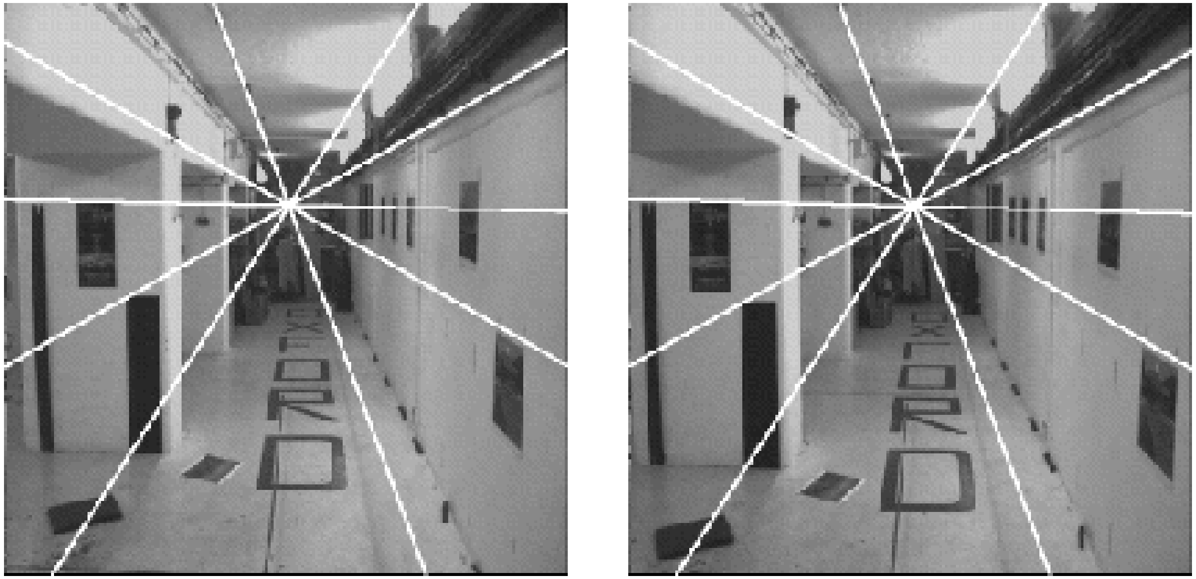
Where is the epipole?

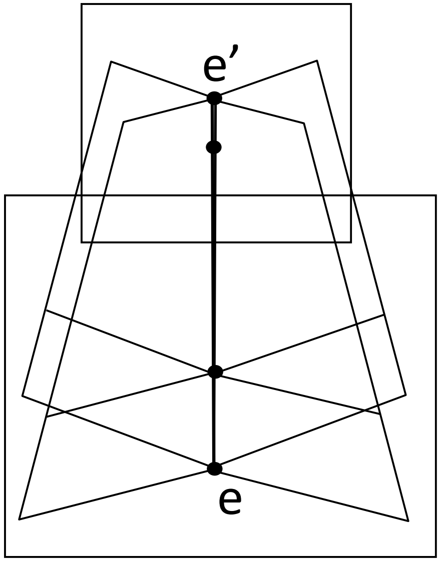


Epipoles *infinitely* far away, epipolar lines parallel

Slide credit: David Fouhey

Example: forward motion





Epipole has same coordinates in both images.
Points move along lines radiating from e: “Focus of expansion”

Figure from Hartley & Zisserman

Motion perpendicular to image plane

http://vimeo.com/48425421

Slide credit: David Fouhey

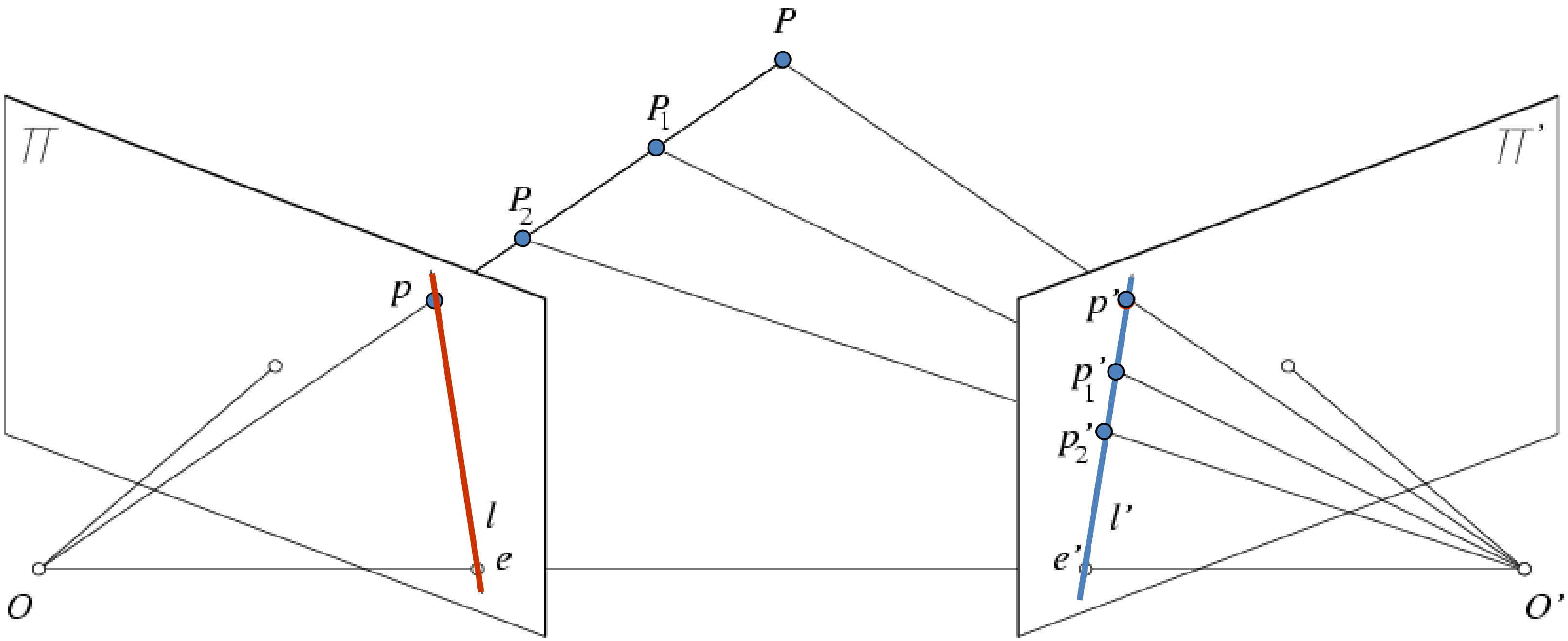
Ok so where were we?

- Setup: Calibrated Camera (both extrinsic & intrinsic)
- Goal: 3D reconstruction of corresponding points in the image
- We need to find correspondences!

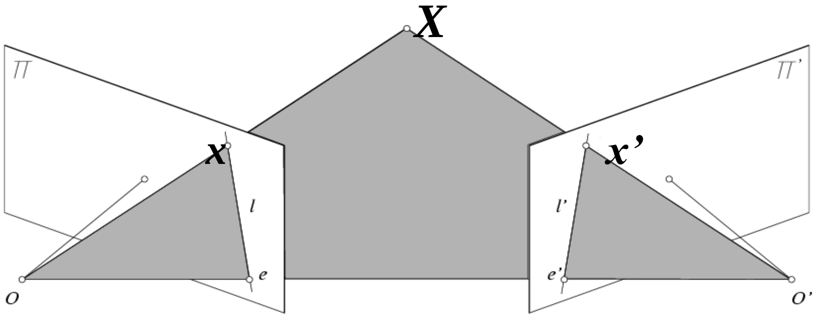
➔1D search along the epipolar line!

➔Need: Compute the epipolar line from camera

Ok so what exactly are l and l'?



Step 0: Factor out intrinsics



$$x = K[R \;\; t]X$$

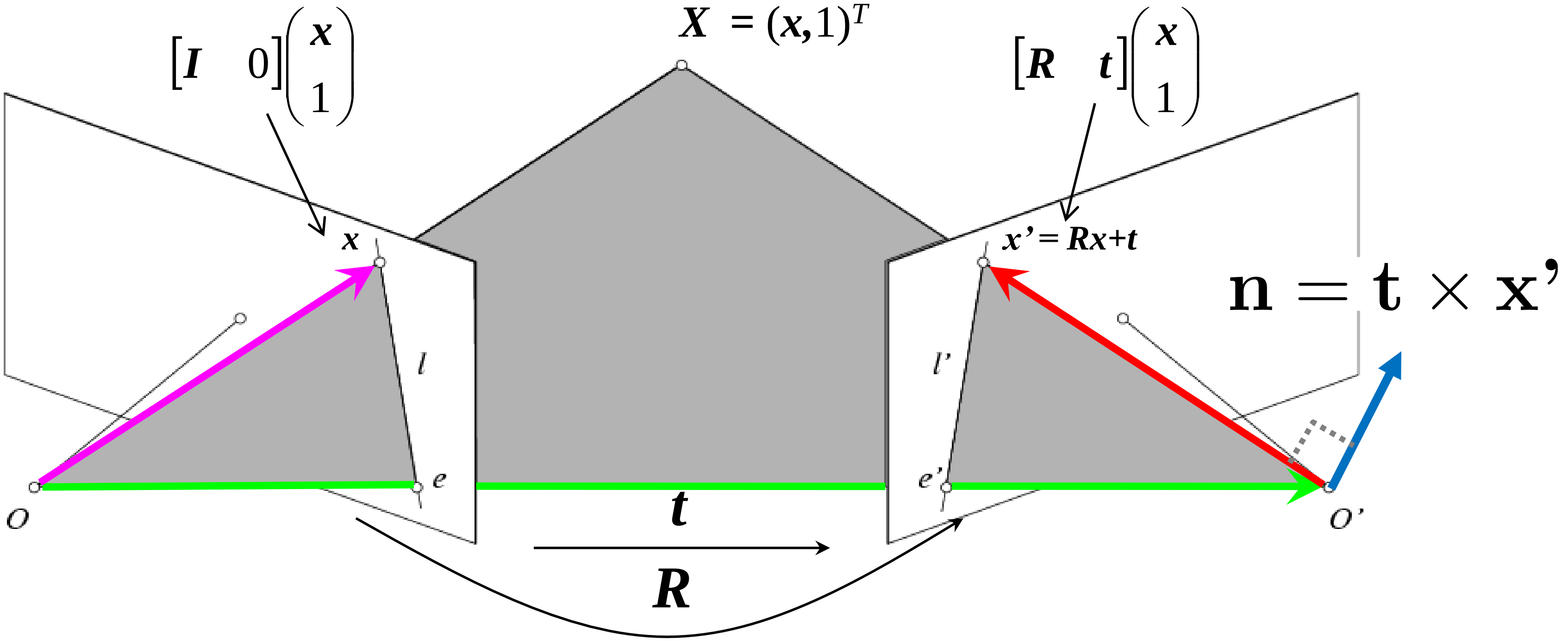
$$K^{-1}x = [R \;\; t]X$$

- Let's factor out the effect of K (do everything in 3D)
- Make it into a ray with K^{-1} and use depth = 1
- This is called the *normalized* image coordinates. It may be thought of as a set of points with identity K

$$\boldsymbol{x}_{\text{norm}} = \boldsymbol{K}^{-1}\boldsymbol{x}_{\text{pixel}} = [\boldsymbol{I} \;\; 0]\,\boldsymbol{X}, \qquad \boldsymbol{x}'_{\text{norm}} = \boldsymbol{K}'^{-1}\boldsymbol{x}'_{\text{pixel}} = [\boldsymbol{R} \;\; \boldsymbol{t}]\,\boldsymbol{X}$$

- Assume that the points are normalized from here on

Epipolar constraint: Calibrated case



$\begin{bmatrix} I & 0 \end{bmatrix} \begin{pmatrix} x \\ 1 \end{pmatrix}$
$X = (x, 1)^T$
$\begin{bmatrix} R & t \end{bmatrix} \begin{pmatrix} x \\ 1 \end{pmatrix}$
x
x' = Rx+t
$\mathbf{n} = \mathbf{t} \times \mathbf{x'}$
l
l'
e
e'
O
O'
t
R

The vectors *x*, *t*, and *x'* are coplanar

What can you say about their relationships, given $\mathbf{n} = \mathbf{t} \times \mathbf{x'}$?

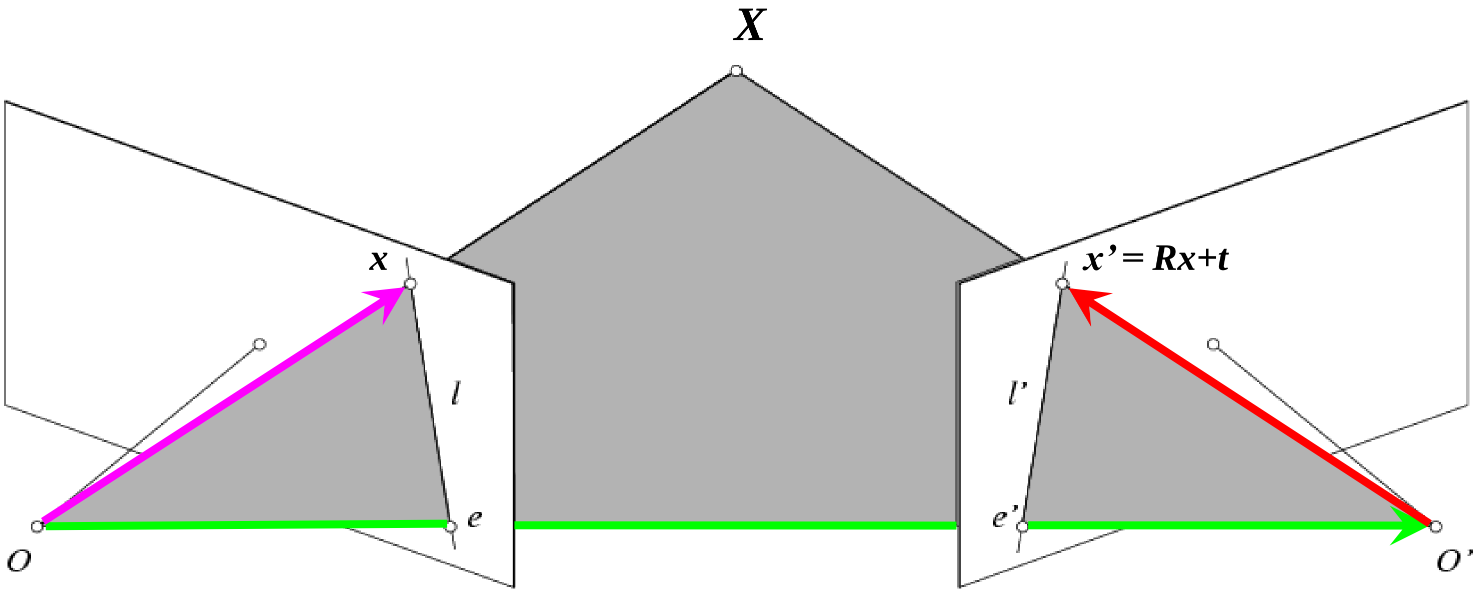
$$\boxed{\mathbf{x'} \cdot (\mathbf{t} \times \mathbf{x'}) = 0}$$

$$\mathbf{x'} \cdot (\mathbf{t} \times (R\mathbf{x} + \mathbf{t})) = 0$$

$$\mathbf{x'} \cdot (\mathbf{t} \times R\mathbf{x} + \cancelto{0}{\mathbf{t} \times \mathbf{t}})) = 0$$

$$\boxed{\mathbf{x'} \cdot (\mathbf{t} \times R\mathbf{x}) = 0}$$

Epipolar constraint: Calibrated case

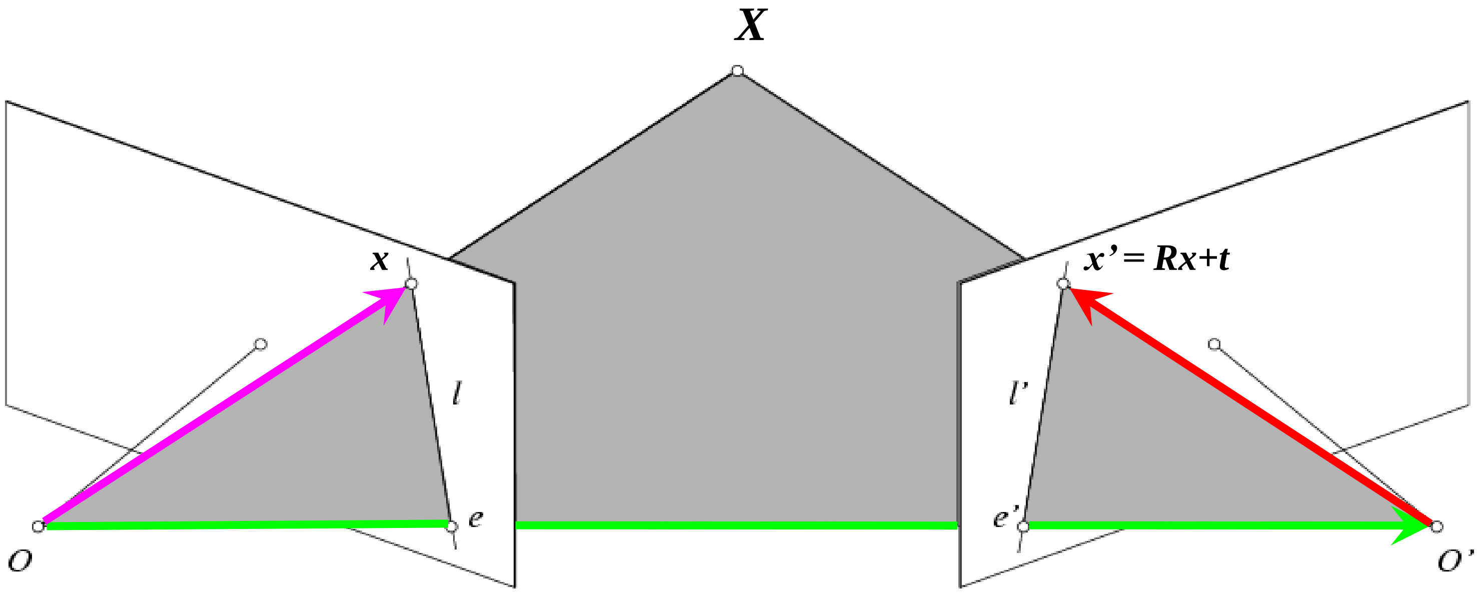


$$\boldsymbol{x}' \cdot [\boldsymbol{t} \times (\boldsymbol{R}\boldsymbol{x})] = 0 \quad \Rightarrow \quad \boldsymbol{x}'^T [\boldsymbol{t}_\times] \boldsymbol{R}\boldsymbol{x} = 0$$

Recall: $\mathbf{a} \times \mathbf{b} = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} = [\mathbf{a}_\times]\mathbf{b}$

The vectors ***x***, ***t***, and ***x’*** are coplanar

Epipolar constraint: Calibrated case



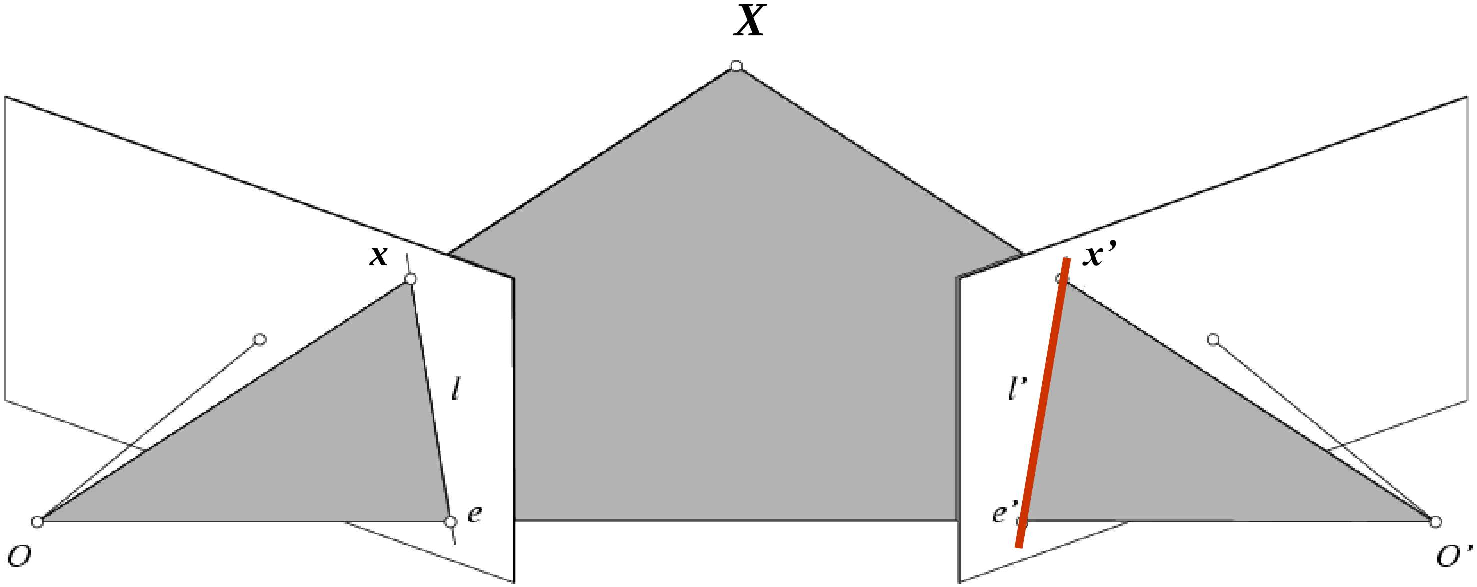
$$\mathbf{x}'\cdot[\boldsymbol{t}\times(\boldsymbol{R}\boldsymbol{x})]=0 \quad\Rightarrow\quad \boldsymbol{x}¢^{T}\underbrace{[\boldsymbol{t}_{\times}]}_{E}\boldsymbol{R}\boldsymbol{x}=0 \quad\Rightarrow\quad \boldsymbol{x}¢^{T}E\boldsymbol{x}=0$$

Recall: $\mathbf{a}\times\mathbf{b}=\begin{bmatrix}0 & -a_z & a_y\\ a_z & 0 & -a_x\\ -a_y & a_x & 0\end{bmatrix}\begin{bmatrix}b_x\\ b_y\\ b_z\end{bmatrix}=[\mathbf{a}_{\times}]\mathbf{b}$

Essential Matrix
(Longuet-Higgins, 1981)

The vectors ***x***, ***t***, and ***x'*** are coplanar

Epipolar constraint: Calibrated case

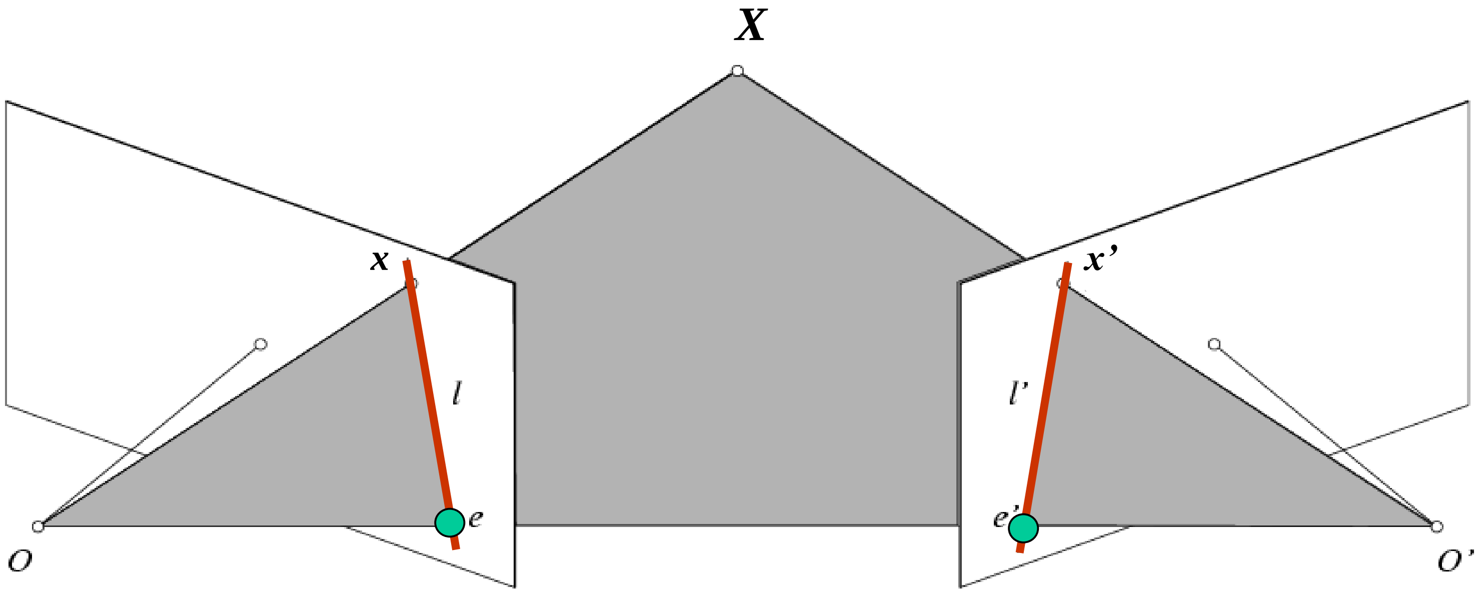


$$\boxed{\boldsymbol{x}\text{¢}^{T} E\, \boldsymbol{x} = 0}$$

- ***E x*** is the epipolar line associated with ***x*** (***l'*** = ***E x***)
  - Recall: a line is given by $ax + by + c = 0$ or

$$\mathbf{l}^T\mathbf{x} = 0 \quad \text{where} \quad \mathbf{l} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Epipolar constraint: Calibrated case



$$\boxed{\boldsymbol{x}\text{¢}^{T} E\,\boldsymbol{x} = 0}$$

- ***E x*** is the epipolar line associated with ***x*** (***l***′ = ***E x***)
- ***E***^{T}***x***′ is the epipolar line associated with ***x′*** (***l*** = ***E***^{T}***x***′)
- ***E e*** = 0   and   ***E***^{T}***e***′ = *0*
- ***E*** is singular (rank two)
- ***E*** has five degrees of freedom

Why is the epipolar matrix rank 2?

Question: With known camera (i.e. essential matrix),

can you map a point on the left image to the right?

A: No! You don’t know where exactly the point is, just that it will be on a line.

2D point → 1D line, lost a dimension

Why? Because camera does not give you depth. You need depth for precise point corresp

R, T ~ Essential Matrix

Recall, knowing the camera gives you the essential matrix (i.e. the plane per point)

So the DoF has to match up

Essential matrix: 3 x 3, 9 numbers, but rank2 means 2 columns fully define = 6 parameters -1 for scale = 5 DoF

Extrinsic Camera (R, T): 3 for rotation, 3 for translation, but -1 for scale = 5 DoF!

Epipolar constraint: Uncalibrated case

- Recall that we normalized the coordinates

$$x = K^{-1}\hat{x} \quad x' = K'^{-1}\hat{x}'$$

$$\hat{x} = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

where $\hat{x}$ is the image coordinates

- But in the *uncalibrated* case, ***K*** and ***K'*** are unknown!
- We can write the epipolar constraint in terms of *unknown* normalized coordinates:

$$x'^T E x = 0$$

$$(K'^{-1}\hat{x}')'^T E(K^{-1}\hat{x}) = 0$$

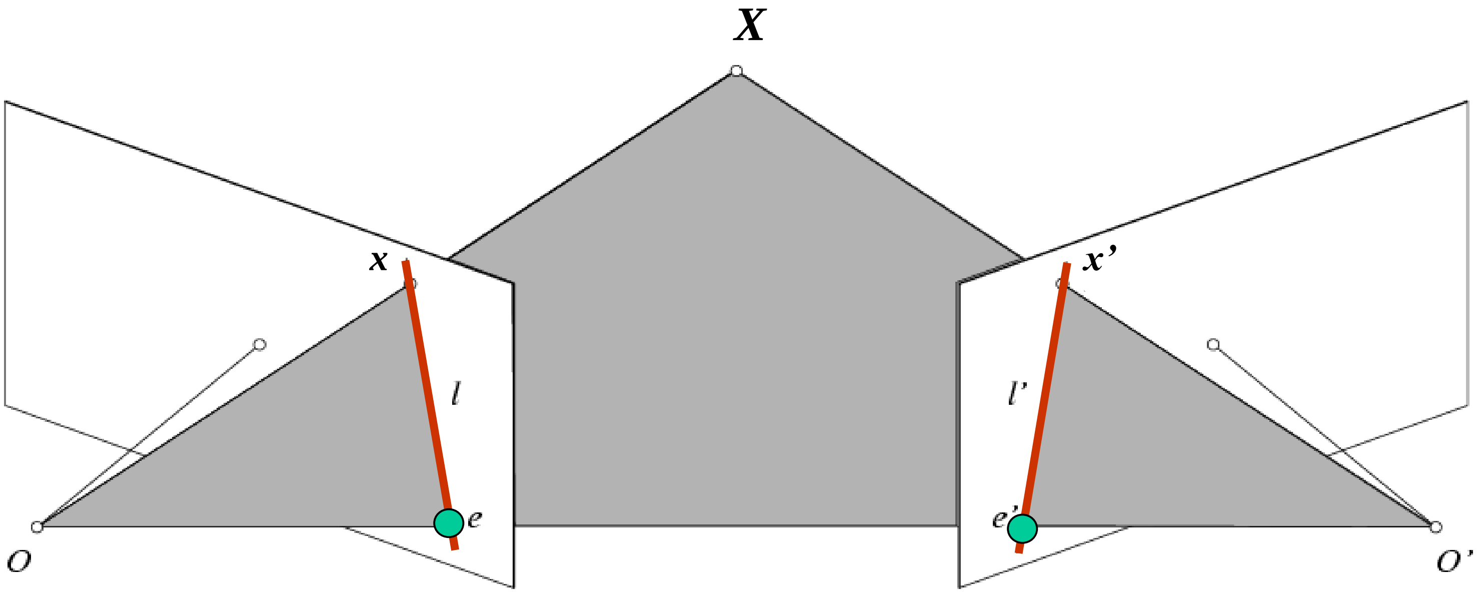
$$\hat{x}'^T \underbrace{K'^{-T}E(K^{-1}}\hat{x}) = 0$$

$$\hat{x}'^T F \hat{x} = 0$$

$$F = K'^{-T}EK^{-1}$$

Fundamental Matrix
(Faugeras and Luong, 1992)

Epipolar constraint: Uncalibrated case



$$x'^T E x = 0 \quad \Rightarrow \quad \hat{x}'^T F \hat{x} = 0 \quad \text{with} \quad F = K'^{-T} E K^{-1}$$

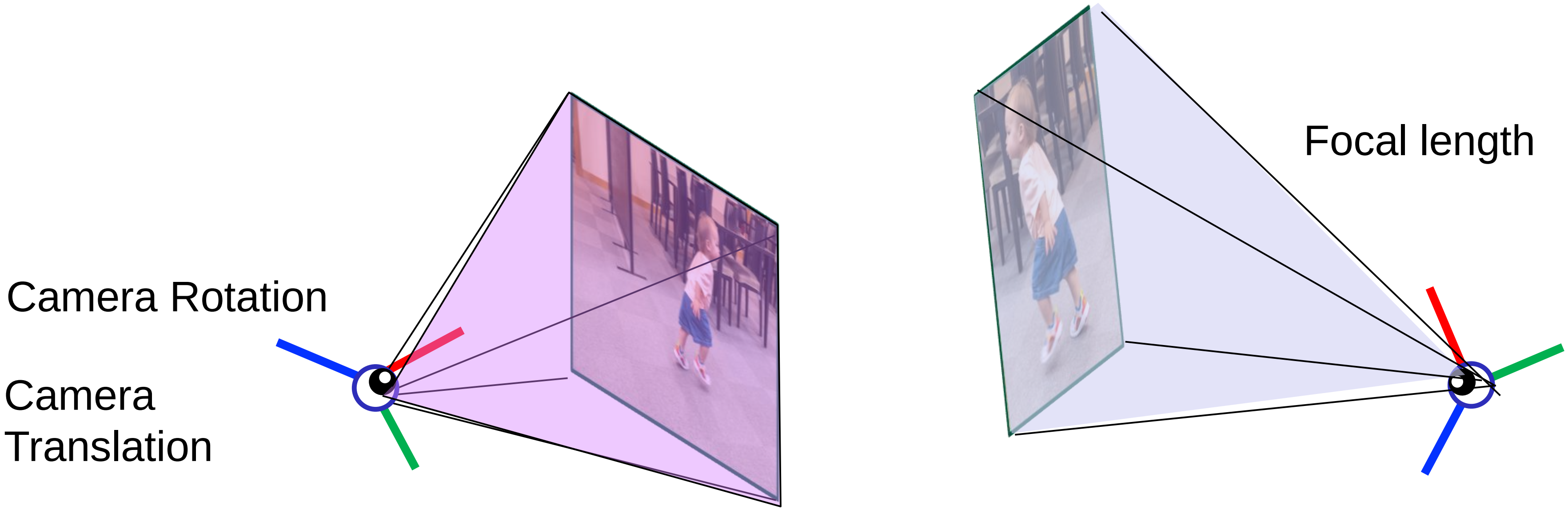
- $\boldsymbol{F}\,\hat{\boldsymbol{x}}$ is the epipolar line associated with $\hat{\boldsymbol{x}}$ ($\boldsymbol{l}' = \boldsymbol{F}\,\hat{\boldsymbol{x}}$)
- $\boldsymbol{F}^T\hat{\boldsymbol{x}}'$ is the epipolar line associated with $\hat{\boldsymbol{x}}'$ ($\boldsymbol{l} = \boldsymbol{F}^T\,\hat{\boldsymbol{x}}'$)
- $\boldsymbol{F}\,\boldsymbol{e} = 0$ and $\boldsymbol{F}^T\boldsymbol{e}' = 0$
- ***F*** is singular (rank two)
- ***F*** has *seven* degrees of freedom

Why 7 DoF?

Think about the frustum, this is the unknown!

It’s a scaling factor on the frustum, thus 1 unknown for each camera

5 of Essential matrix + 2 = 7.



Where are we? (in the original setup)

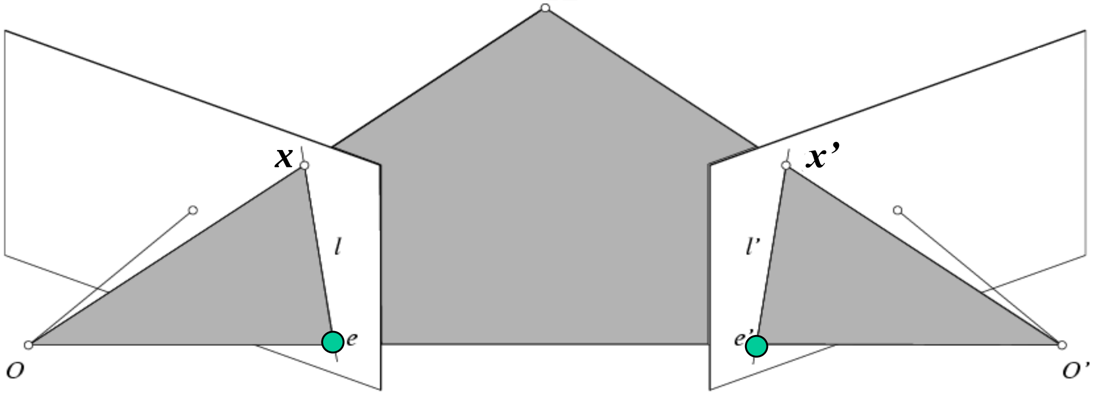
We have two images with calibrated cameras, want the 3D points!

1. Solve for correspondences using epipolar constraints from known camera (1D search)
   - **Now we know the exact equation of this line**
2. Triangulate to get depth!

Finally: computing depth by triangulation

We know about the camera, K_1, K_2 and [R t]:

$$\boldsymbol{X} = X' = RX + T$$



and found the corresponding points: $x \leftrightarrow x'$

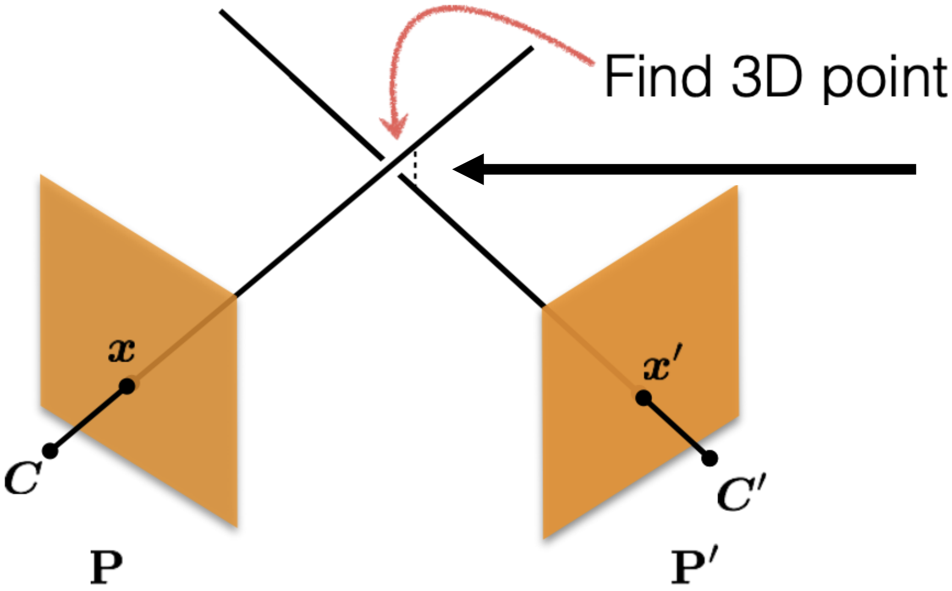
$$x = K\boxed{X} \qquad x' = K'X'$$
$$= K'(R\boxed{X} + T)$$

only unknowns!

How many unknowns
+ how many equations
do we have?

Solve by formulating
Ax=0, see H&Z ch.12

In practice the rays may not intersect!



Find 3D point

Ray’s don’t always intersect
because of noise!!!

$$\mathbf{X} \text{ s.t.}$$

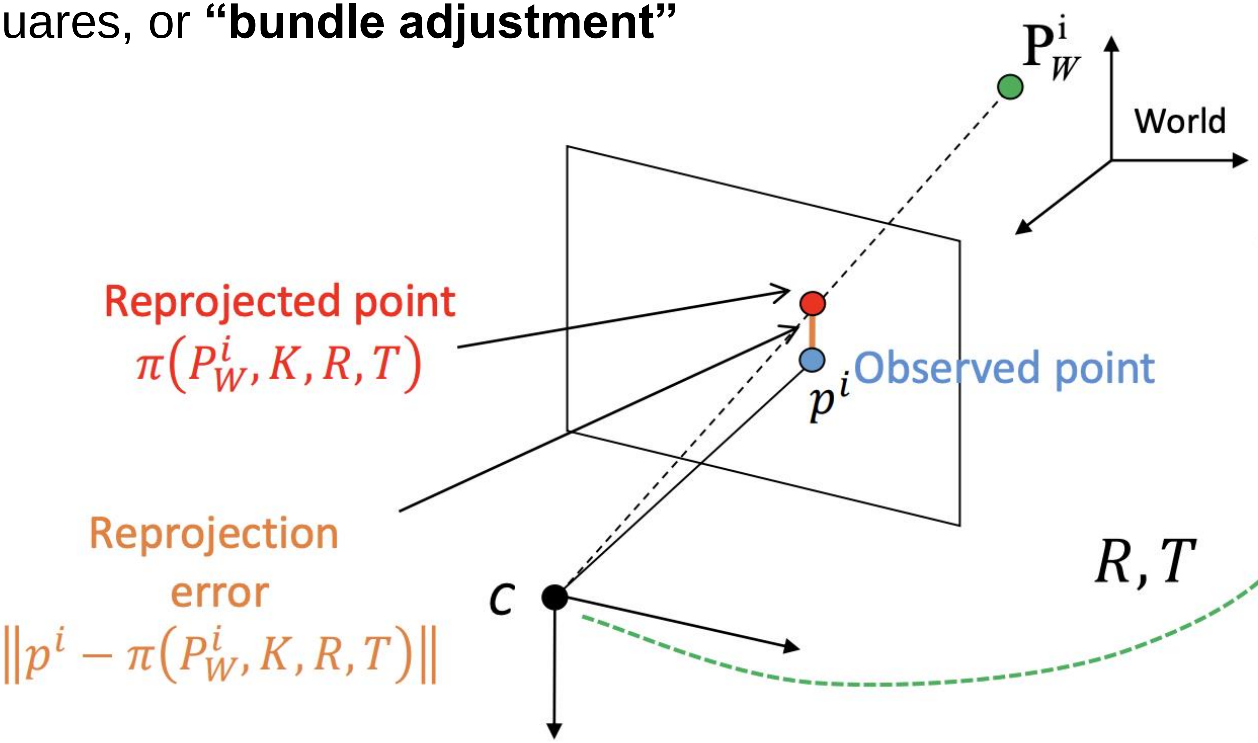
$$\mathbf{x} = \mathbf{P}\mathbf{X}, \;\; \mathbf{x}' = \mathbf{P}'\mathbf{X}$$

Slide credit: Shubham Tulsiani

Reprojection error

Even if you do everything right, you will still be off because of noise, this is called the Reprojection Error

In practice with noise, want to directly minimize this with non-linear least squares, or **“bundle adjustment”**



Solve with non-linear least squares, iteratively

Summary: Two-view, known camera

0. Assuming known camera intrinsics + extrinsics

1. Find correspondences:

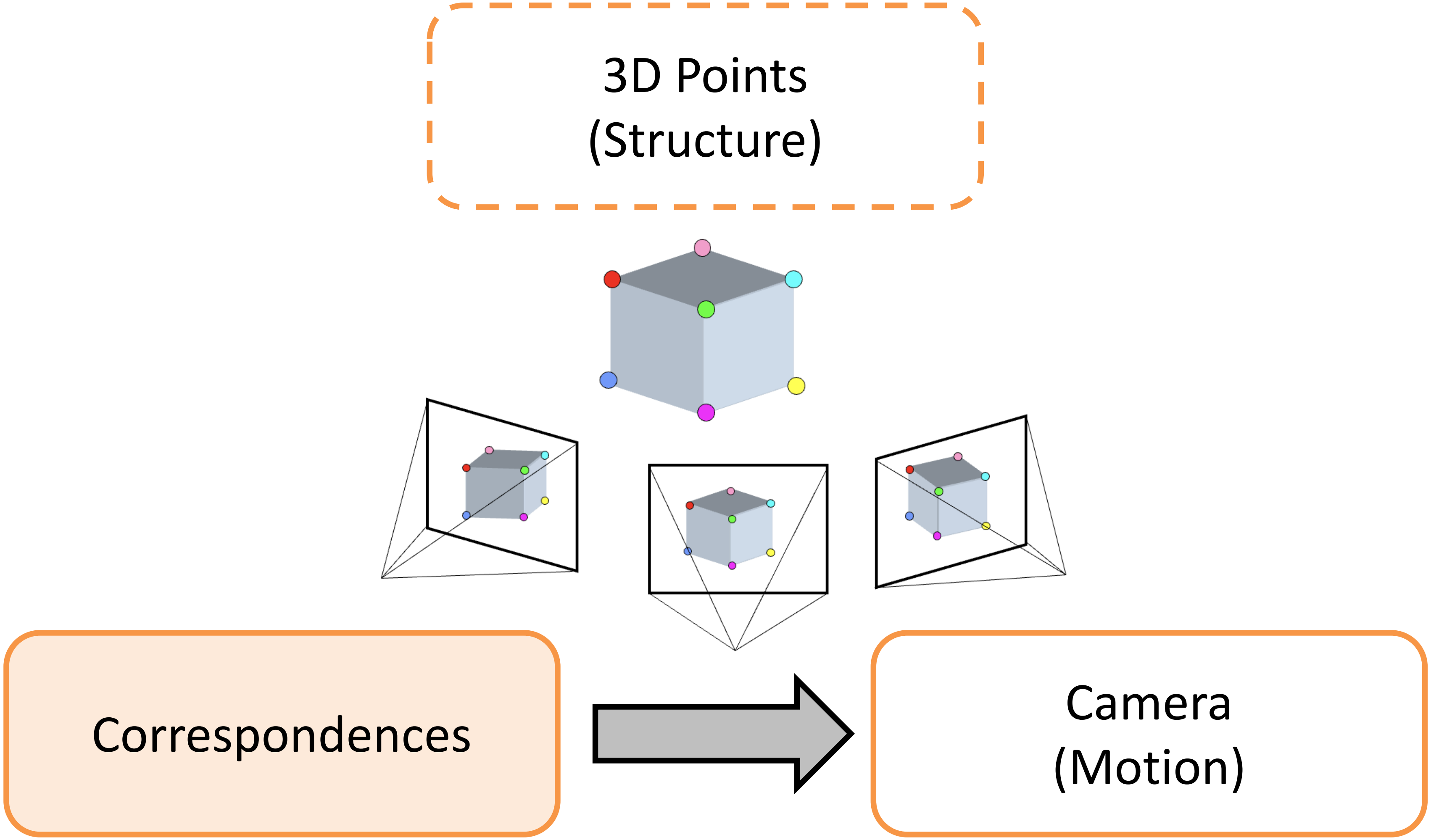
   - Reduce this to 1D search with Epipolar Geometry!

2. Get depth:

   - If simple stereo, disparity (difference of corresponding points) is inversely proportional to depth

   - In the general case, triangulate.

What if we don’t know the camera?



3D Points
(Structure)

Correspondences

Camera
(Motion)

What if we don’t know the camera?

Assume we know the correspondences ($\hat{x}'$, $\hat{x}$) :

Solve for F s.t.

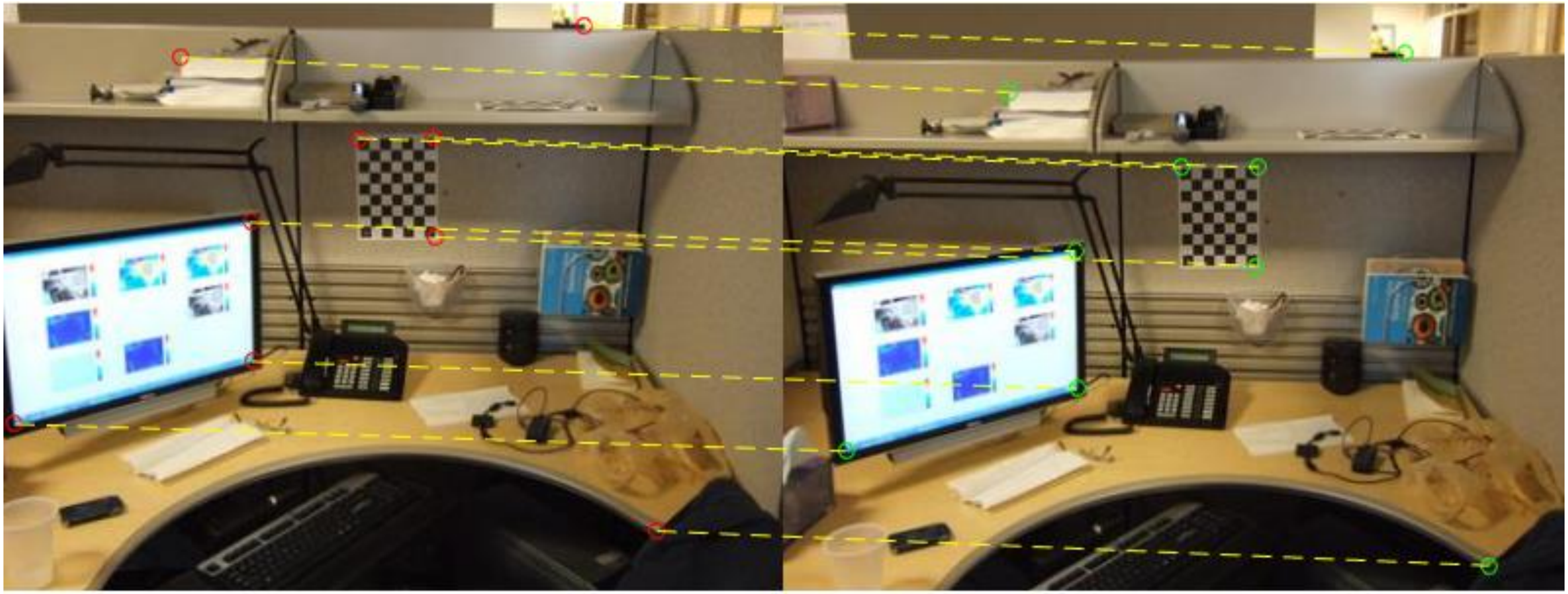
$$\hat{x}'^T F \hat{x} = 0$$

$$\hat{x} = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} u' & v' & 1 \end{bmatrix} \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = 0$$

How many correspondences do we need?

Estimating the fundamental matrix



The eight-point algorithm

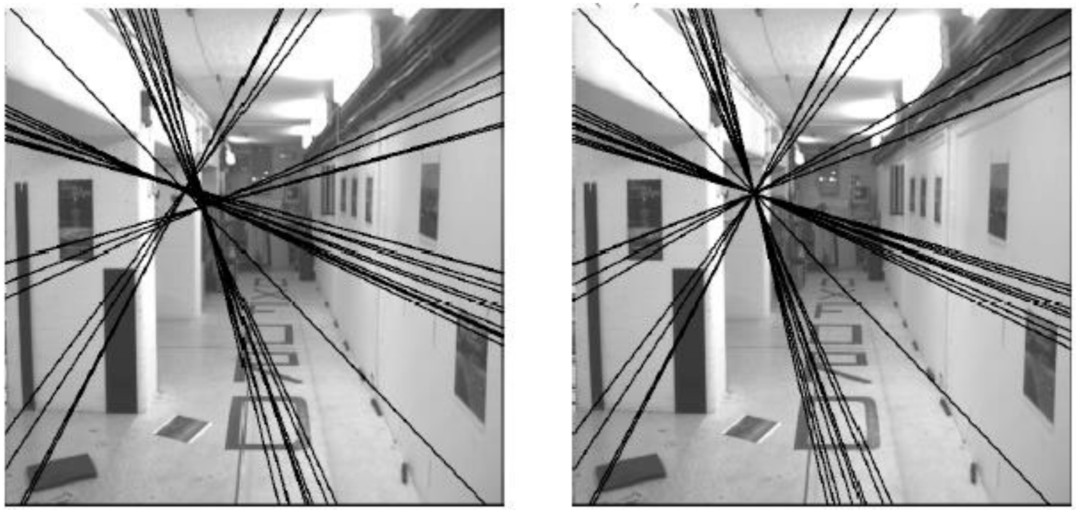
$$\boldsymbol{x} = (u, v, 1)^T, \quad \boldsymbol{x}' = (u', v', 1)$$

$$[u' \quad v' \quad 1]\begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix}\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = 0$$

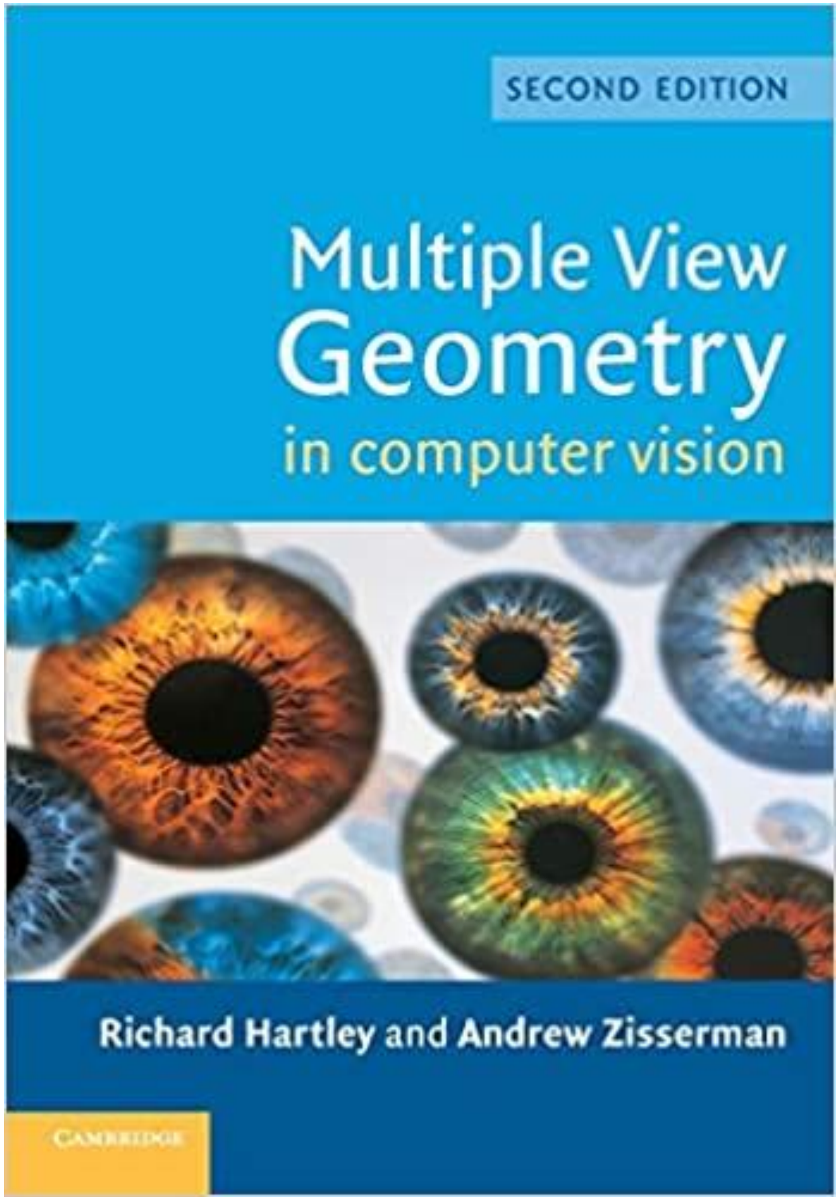
$$[u'u \quad u'v \quad u' \quad v'u \quad v'v \quad v' \quad u \quad v \quad 1]\begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = 0$$

Solve homogeneous linear system using eight or more matches

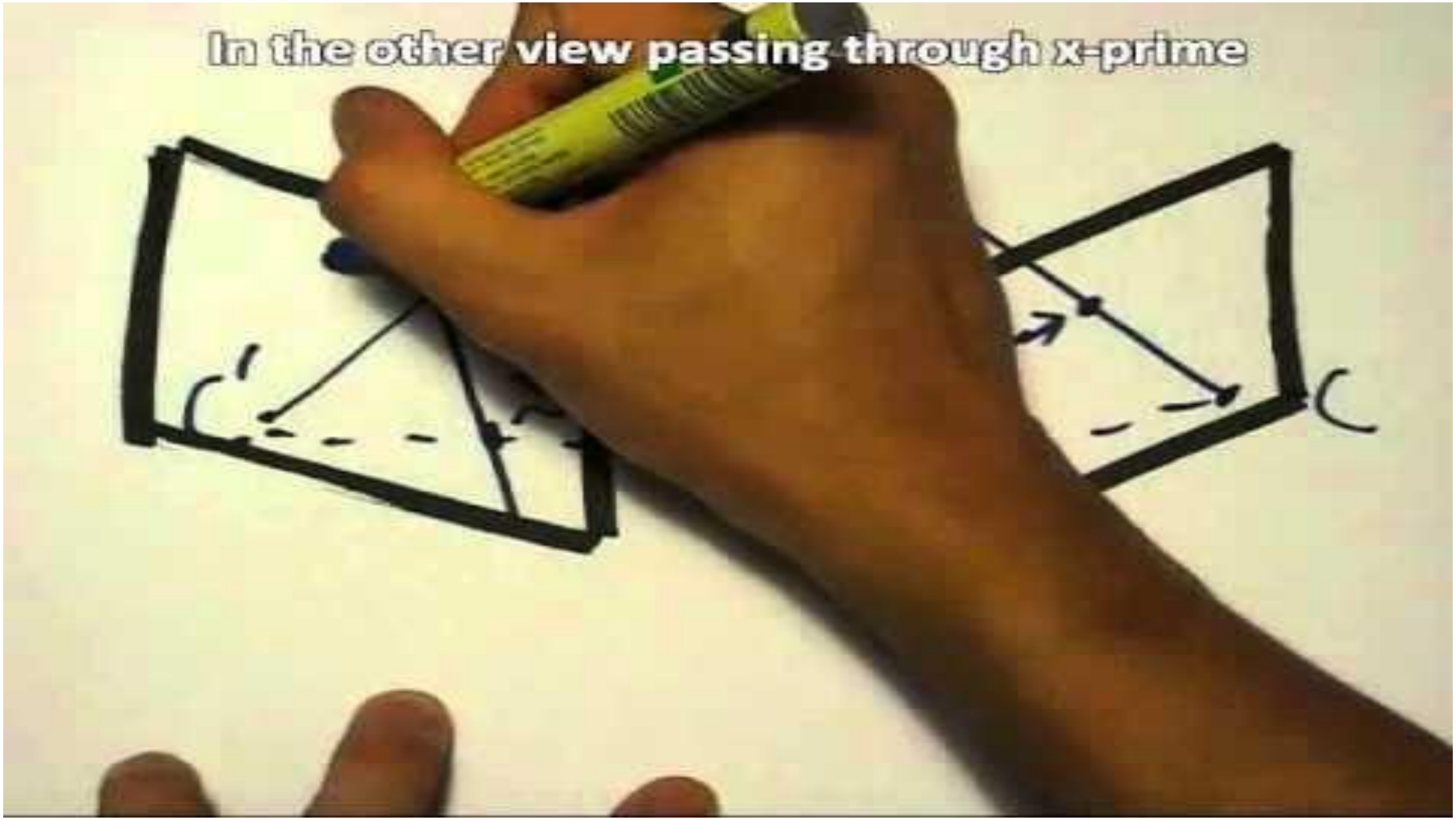
Enforce rank-2 constraint (take SVD of ***F*** and throw out the smallest singular value)



The Bible by Hartley & Zisserman


SECOND EDITION
Multiple View Geometry in computer vision
Richard Hartley and Andrew Zisserman
CAMBRIDGE

The Fundamental Matrix Song



http://danielwedge.com/fmatrix/
https://www.youtube.com/watch?time_continue=8&v=DgGV3l82NTk&feature=emb_title

Going from F -> E -> Camera

Get the essential matrix with K (or some estimates of K) …

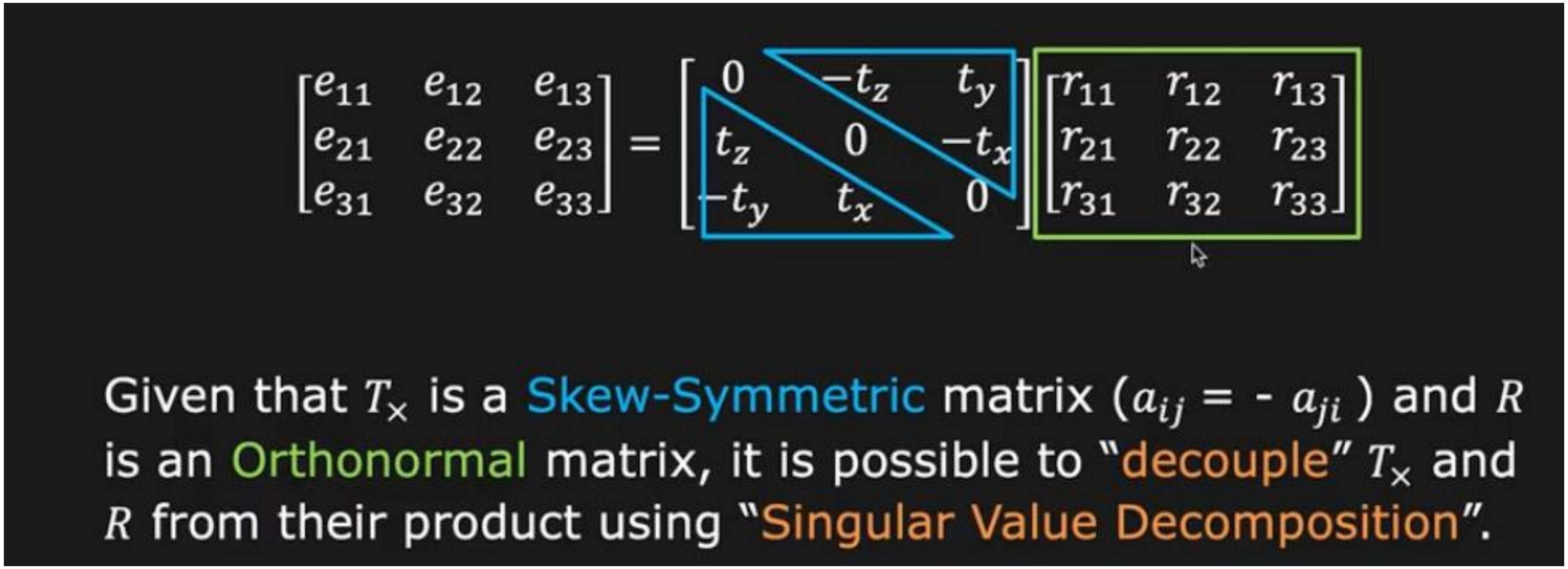
in practice you calibrate your cameras so you know K or have a very good estimate

$$\mathrm{E} = \mathrm{K}'^{\mathsf{T}}\mathrm{F}\mathrm{K}.$$

Essential matrix can be decomposed

$E = T_x R$

If we know E, we can recover t and R



$$\begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{bmatrix} = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix} \boxed{\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}}$$

Given that $T_\times$ is a Skew-Symmetric matrix (a_{ij} = - a_{ji}) and R is an Orthonormal matrix, it is possible to "decouple" $T_\times$ and R from their product using "Singular Value Decomposition".

Slide from Shree Nayar

What about more than two views?

The geometry of three views is described by a 3 x 3 x 3 tensor called the *trifocal tensor*

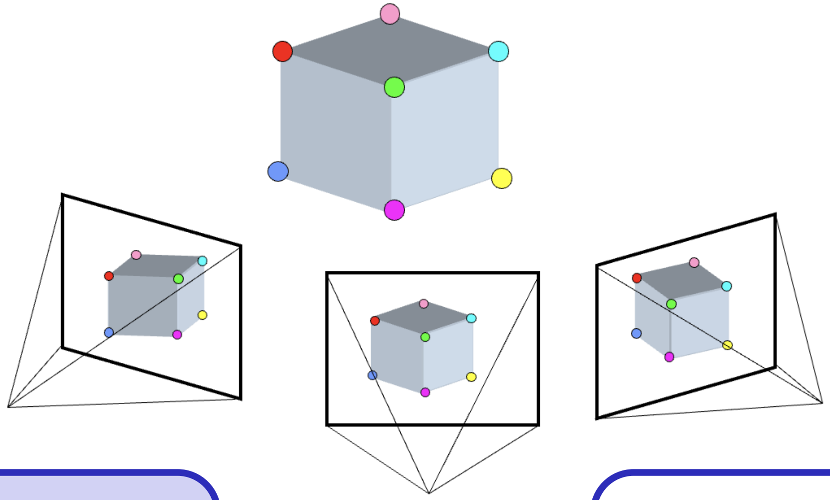
The geometry of four views is described by a 3 x 3 x 3 x 3 tensor called the *quadrifocal tensor*

After this it starts to get complicated…

Slide courtesy of Noah Snavely

This completes: Corresp to Camera

3D Points
(Structure)



Correspondences

Camera
(Motion)

How to estimate the camera?

1. Estimate the fundamental/essential matrix!

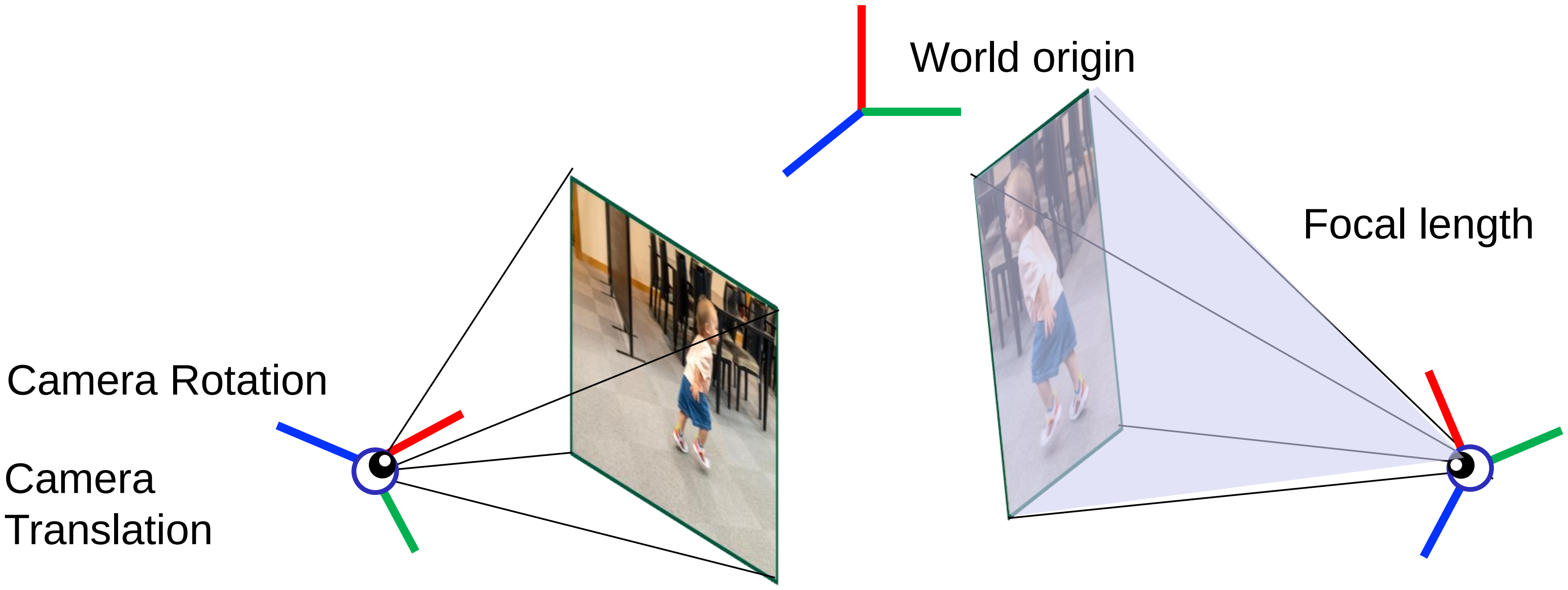
2. Another method: Calibration

Problem: Solve for the camera

What are the camerea parameters?

- Extrinsics (R, T)
- Intrinsics (K)

How am I situated in the world + what is the shape of the ray

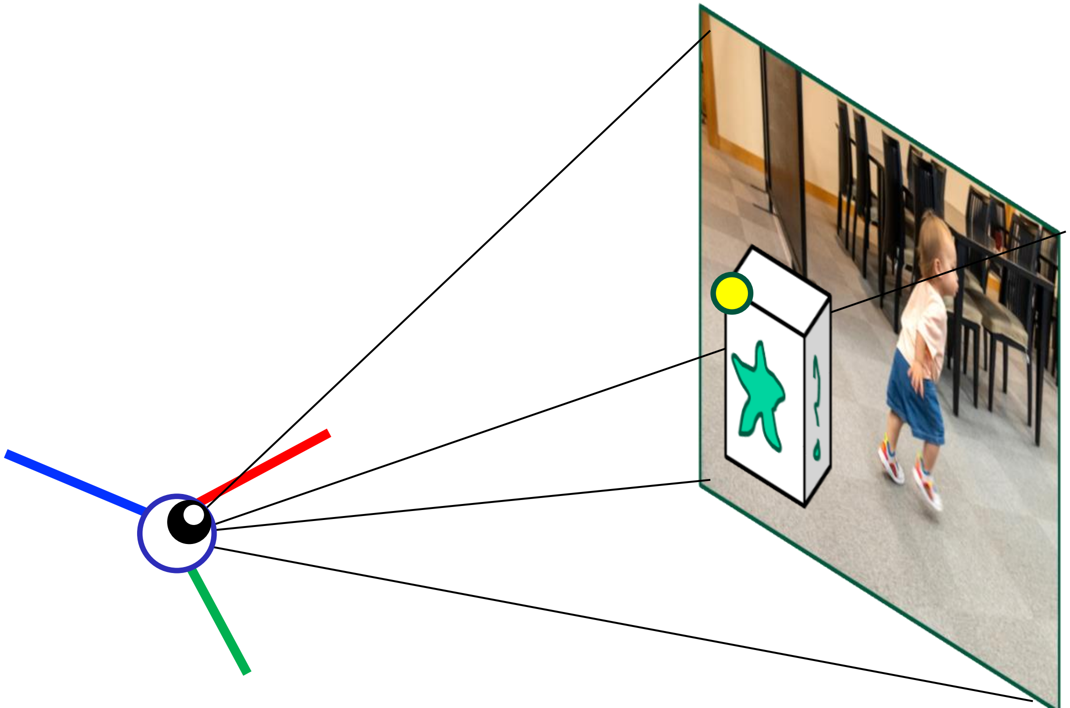


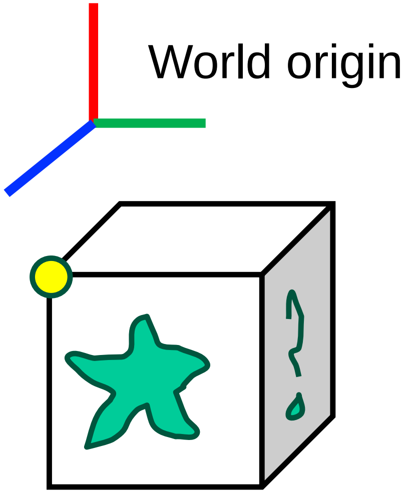
Calibration

Definition: Solve for camera using a known 3D structure + where it is in the image

Invasive / active

Can’t be done on existing pictures





Only have to do it once if the cameras are static

How to calibrate the camera?

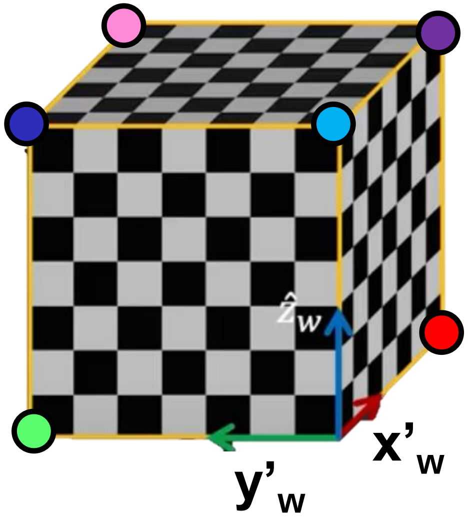
$$\mathbf{x} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}\mathbf{X}$$

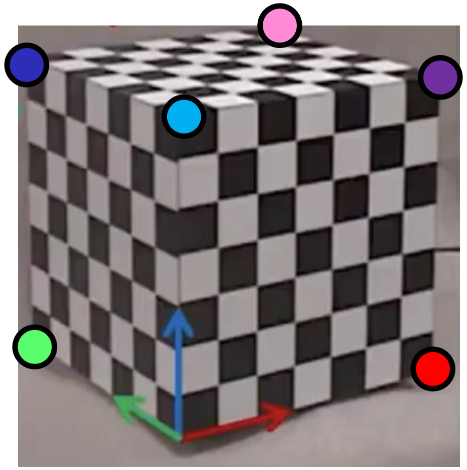
$$\begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

If we know the points in 3D we can estimate the camera!!

Step 1: With a known 3D object

1. Take a picture of an object with known 3D geometry

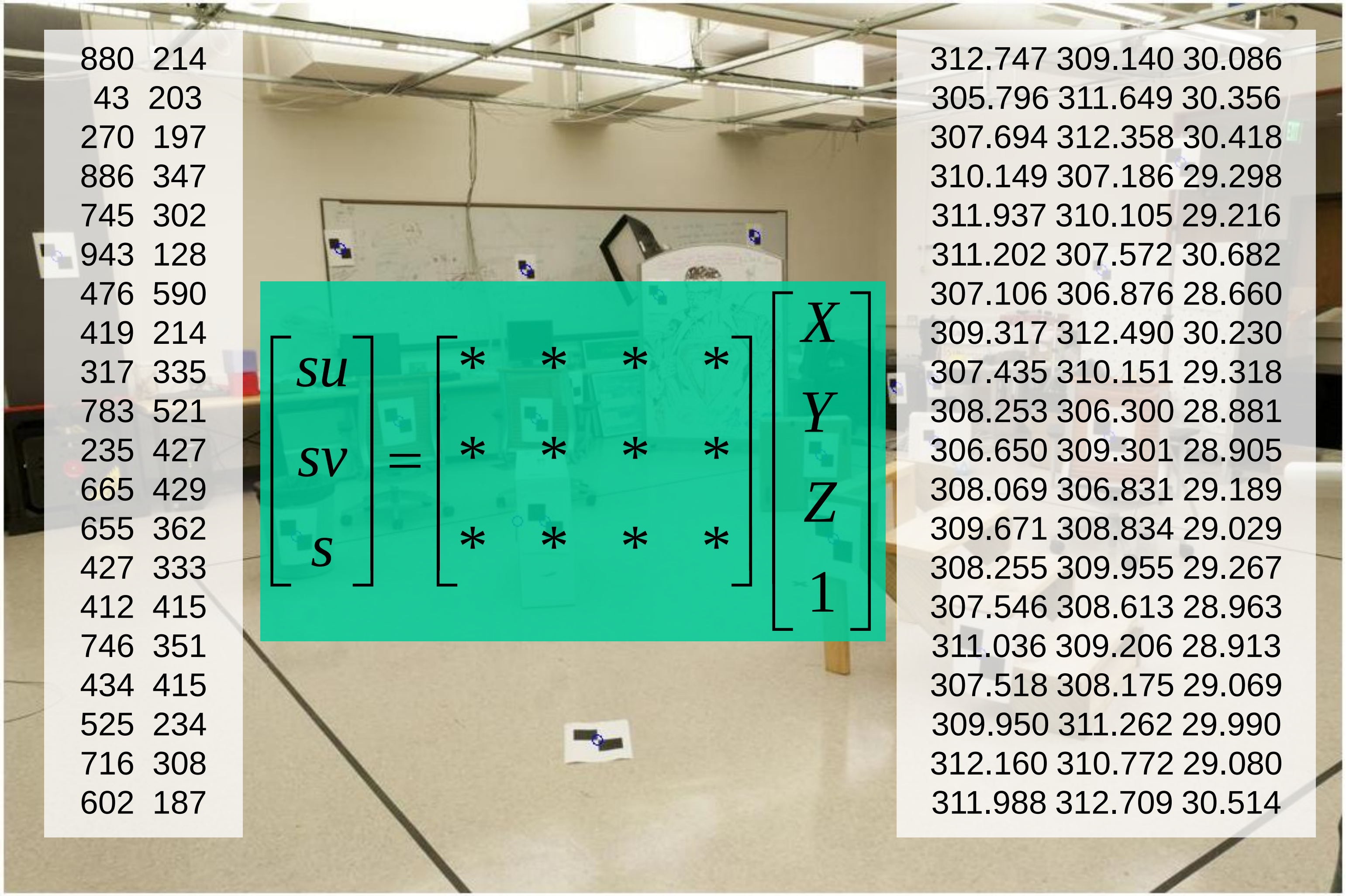




2. Identify correspondences

Slide from Shree Nayar

How do we calibrate a camera?



880	214	312.747	309.140	30.086
43	203	305.796	311.649	30.356
270	197	307.694	312.358	30.418
886	347	310.149	307.186	29.298
745	302	311.937	310.105	29.216
943	128	311.202	307.572	30.682
476	590	307.106	306.876	28.660
419	214	309.317	312.490	30.230
317	335	307.435	310.151	29.318
783	521	308.253	306.300	28.881
235	427	306.650	309.301	28.905
665	429	308.069	306.831	29.189
655	362	309.671	308.834	29.029
427	333	308.255	309.955	29.267
412	415	307.546	308.613	28.963
746	351	311.036	309.206	28.913
434	415	307.518	308.175	29.069
525	234	309.950	311.262	29.990
716	308	312.160	310.772	29.080
602	187	311.988	312.709	30.514

$$\begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Method: Set up a linear system

$$\begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Solve for m’s entries using linear least squares

Ax=0 form

$$\begin{bmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -u_1X_1 & -u_1Y_1 & -u_1Z_1 & -u_1 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -v_1X_1 & -v_1Y_1 & -v_1Z_1 & -v_1 \\ & & & & & & & & & & & \\ X_n & Y_n & Z_n & 1 & 0 & 0 & 0 & 0 & -u_nX_n & -u_nY_n & -u_nZ_n & -u_n \\ 0 & 0 & 0 & 0 & X_n & Y_n & Z_n & 1 & -v_nX_n & -v_nY_n & -v_nZ_n & -v_n \end{bmatrix} \begin{bmatrix} m_{11} \\ m_{12} \\ m_{13} \\ m_{14} \\ m_{21} \\ m_{22} \\ m_{23} \\ m_{24} \\ m_{31} \\ m_{32} \\ m_{33} \\ m_{34} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \\ 0 \\ 0 \end{bmatrix}$$

Similar to how you solved for homography!

Can we factorize M back to K [R | T]?

Yes.

Why? because K and R have a very special form:

$$\begin{bmatrix} f_x & s & o_x \\ 0 & f_y & o_y \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

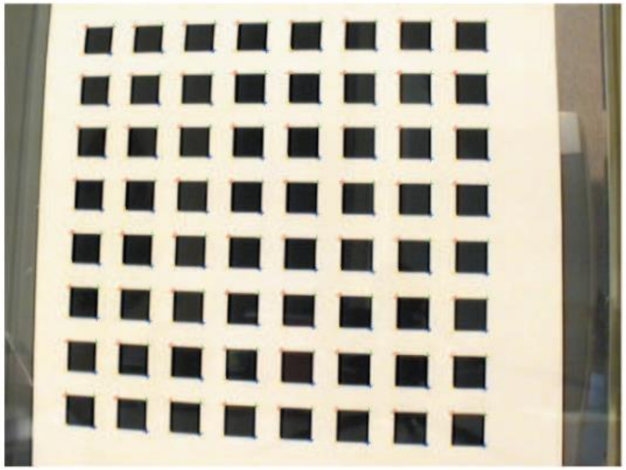
QR decomposition

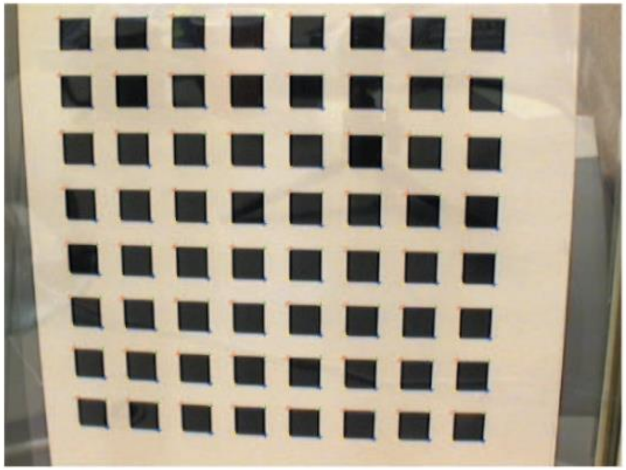
Practically, use camera calibration packages (there is a good one in OpenCV)

Inserting a 3D known object…

Also called “Tsai’scalibration” requires non-coplanar 3D points, is not very practical…

Modern day calibration uses a planar calibration target





Developed in 2000 by Zhang at Microsoft research

Zhang, A flexible new technique for camera calibration, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000

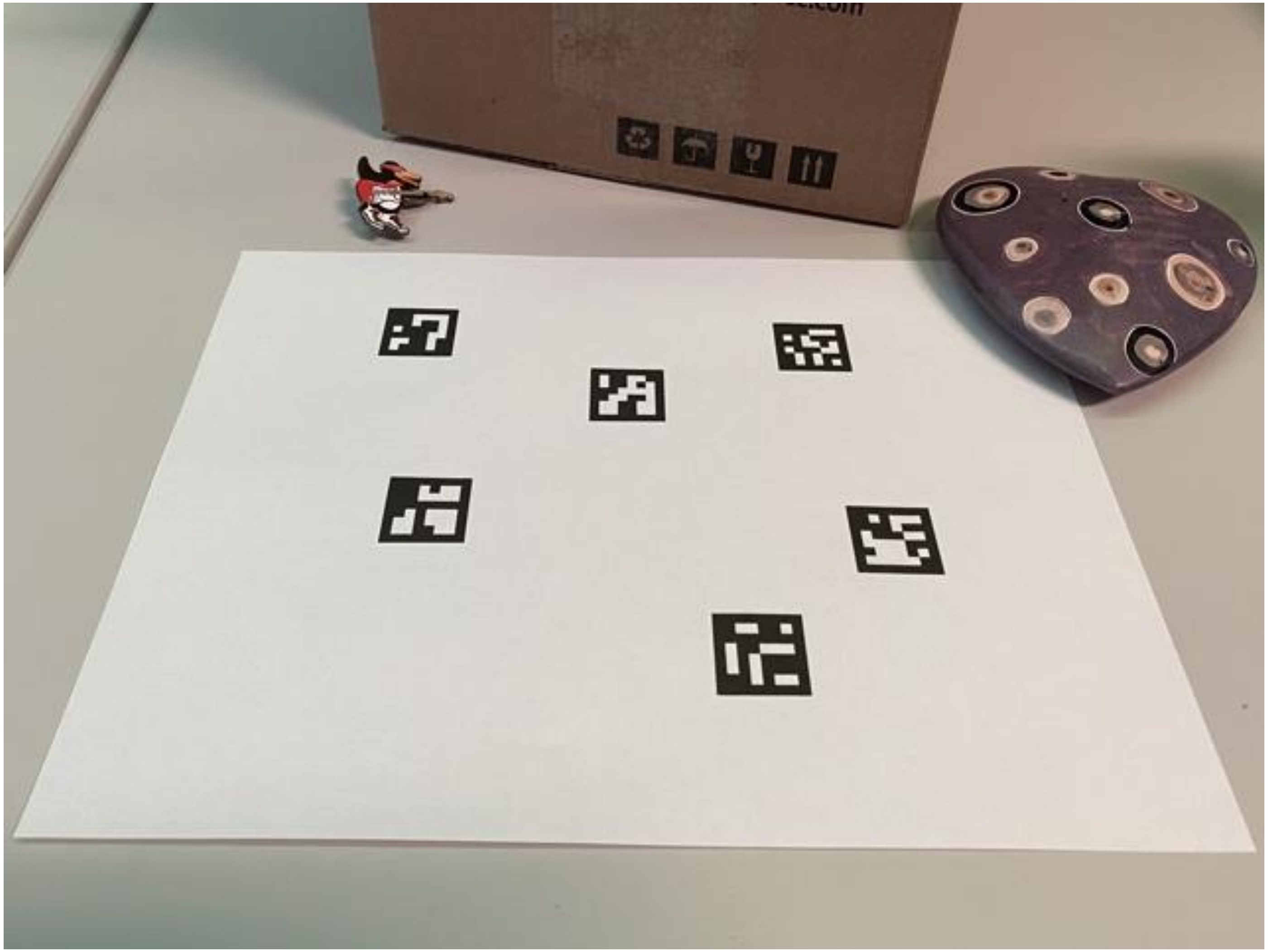
Doesn’t plane give you homography?

Yes! If it’s a plane, it’s only a homography, so instead of recovering 3x4 matrix, you will recover 3x3 in Zhang’s method

The 3x3 gives first two columns of R and T

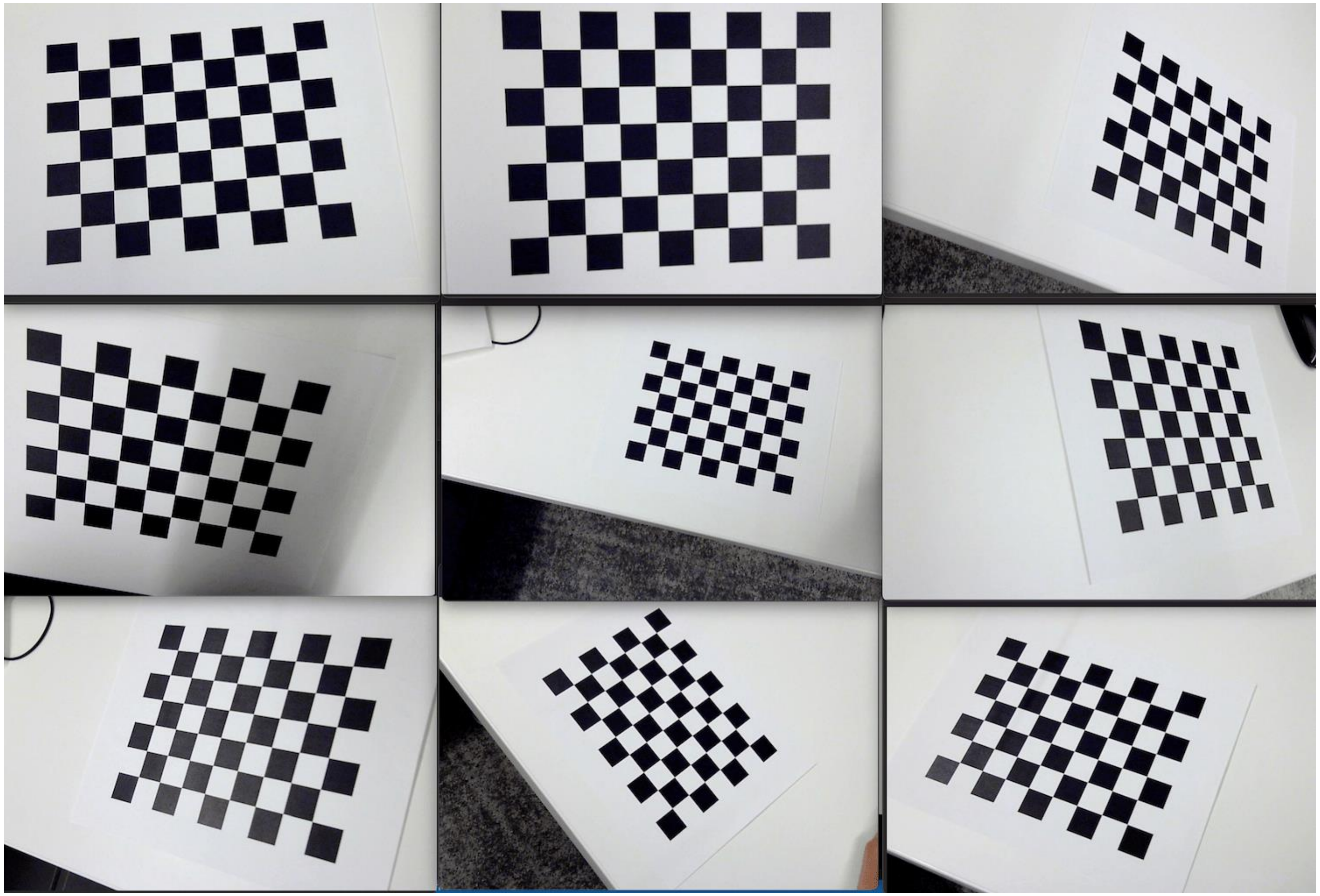
$$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} = \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} r_{11} & r_{12} & t_1 \\ r_{21} & r_{22} & t_2 \\ r_{31} & r_{32} & t_3 \end{bmatrix}$$

You will use Aruco tags



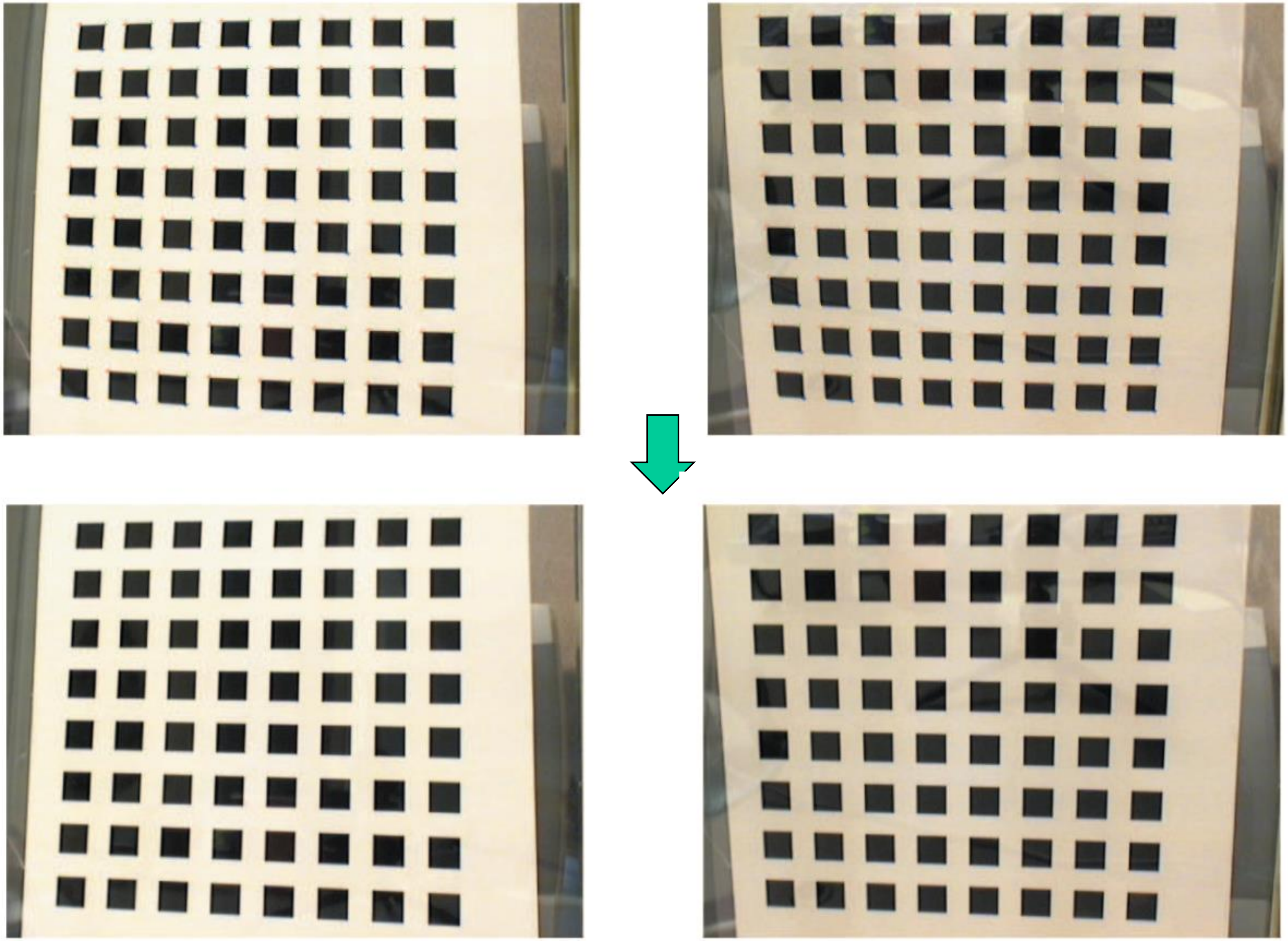
In practice: Step 0

Calibrate your intrinsics first, also estimates lens distortion (cv2.calibrateCamera)



Step 1: Undistort your image

Cv2.undistort()



Step 2: Estimate camera with PnP

PnP – “Perspective-n-Point” problem:

Estimate extrinsic parameters given n correspondences

Minimize reprojection loss with non-linear least squares

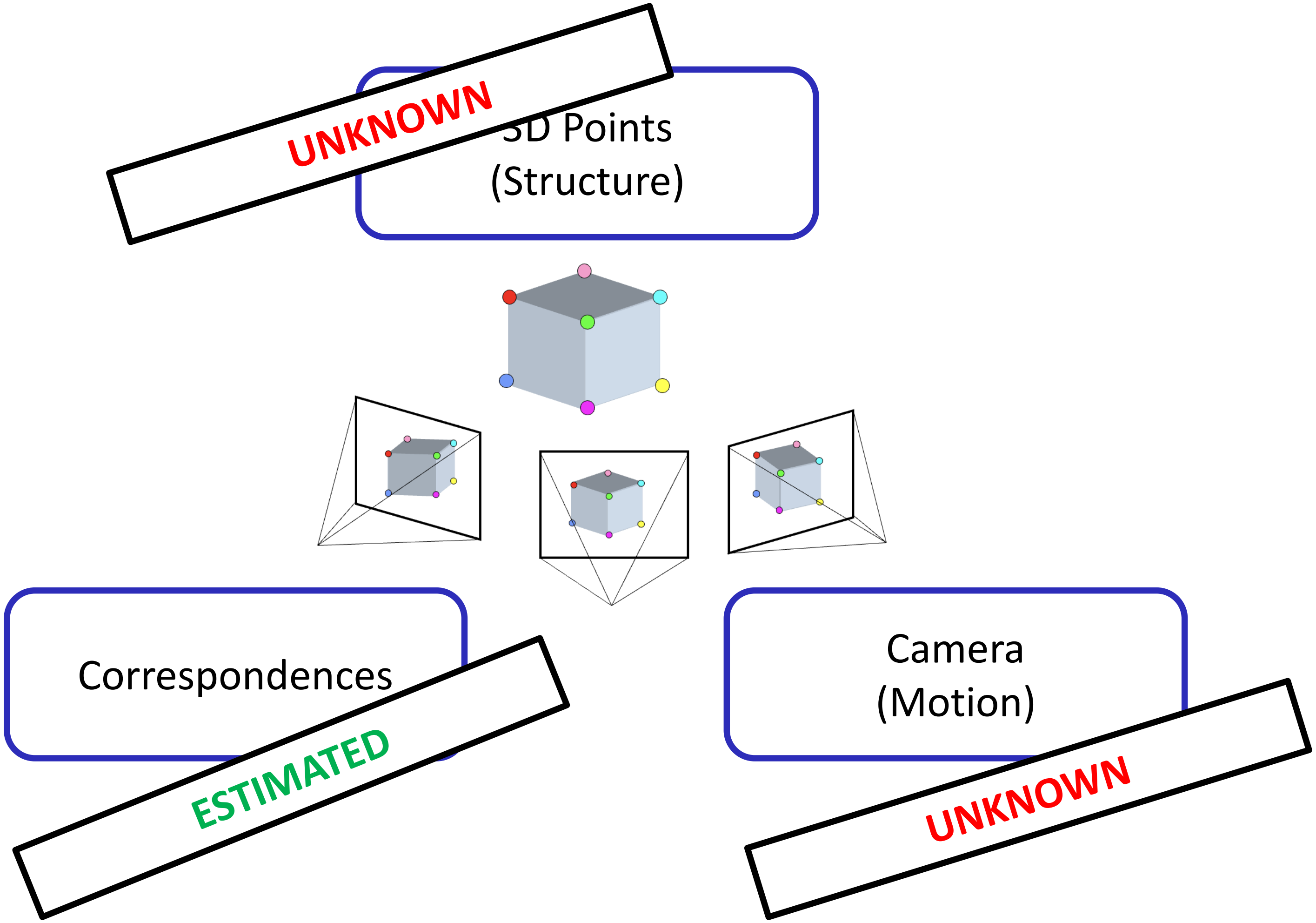
$$\min_{R,T} \sum \|x_i \; - K[R\;T]X\;\|^2$$

In general you do DLT first (Ax=0), then use that as initialization, or do other algorithms like Efficient PnP

Putting it all together

Structure-from-Motion: You know nothing!

(except ok maybe intrinsics)



3D Points
(Structure)

UNKNOWN

Correspondences

ESTIMATED

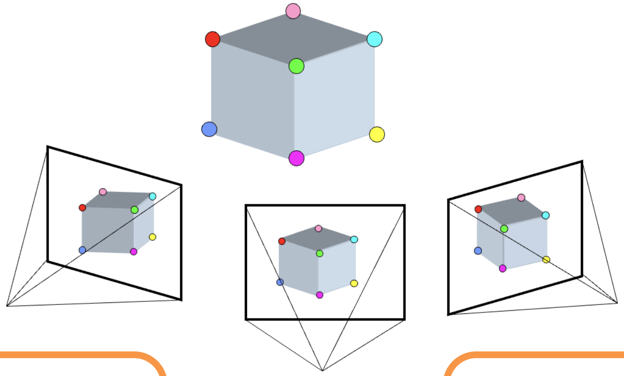
Camera
(Motion)

UNKNOWN

(Next lecture) Ultimate: Structure-from-Motion/SLAM

UNKNOWN

3D Points
(Structure)



Correspondences

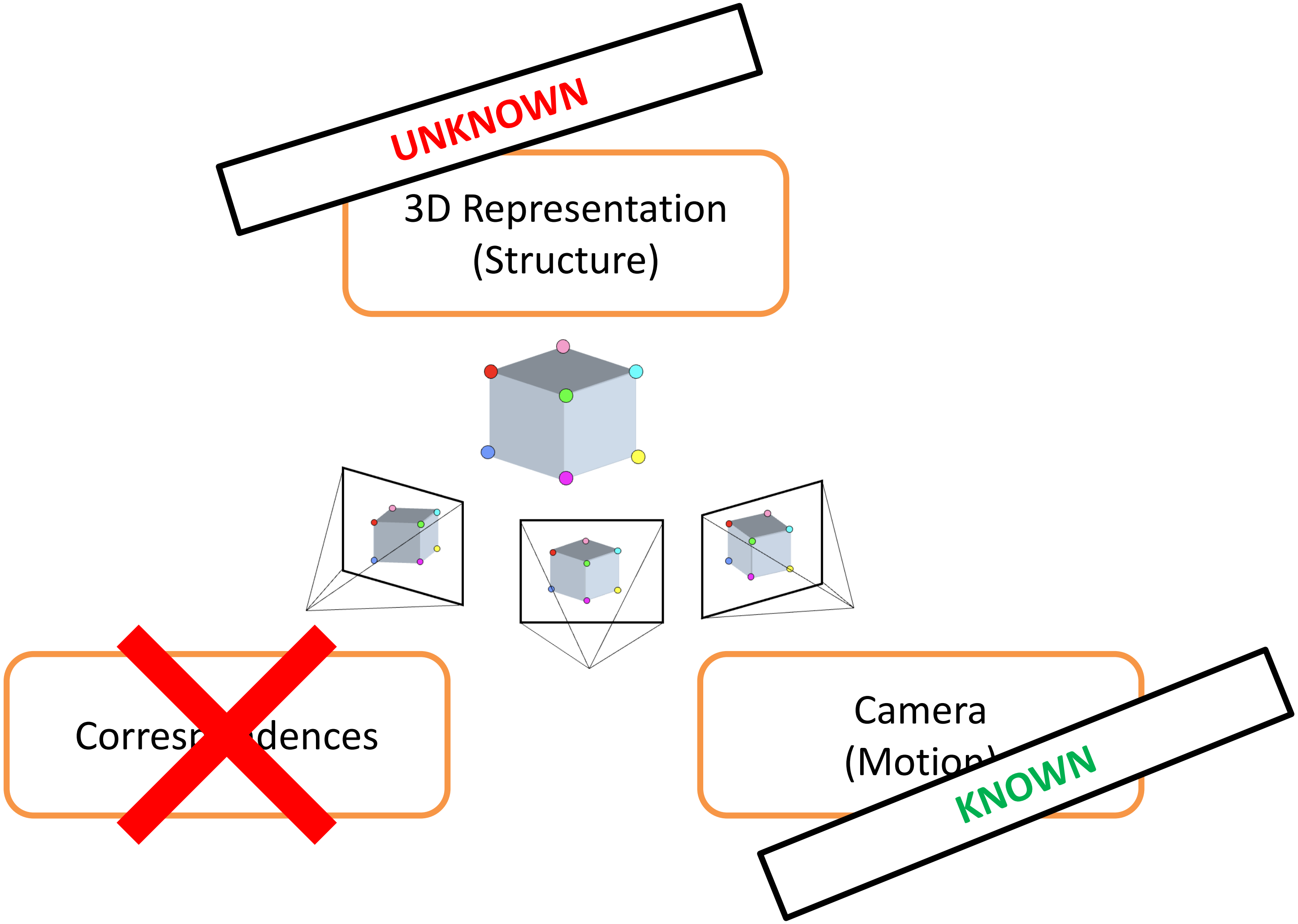
ESTIMATED

Camera
(Motion)

UNKNOWN

The starting point for all problems where you can’t calibrate actively

(after that): Neural Rendering



UNKNOWN

3D Representation
(Structure)

Correspondences

Camera
(Motion)

KNOWN

A form of multi-view stereo, more on this in the NeRF lecture.

Next: Large-scale structure from motion

Dubrovnik, Croatia. 4,619 images (out of an initial 57,845).

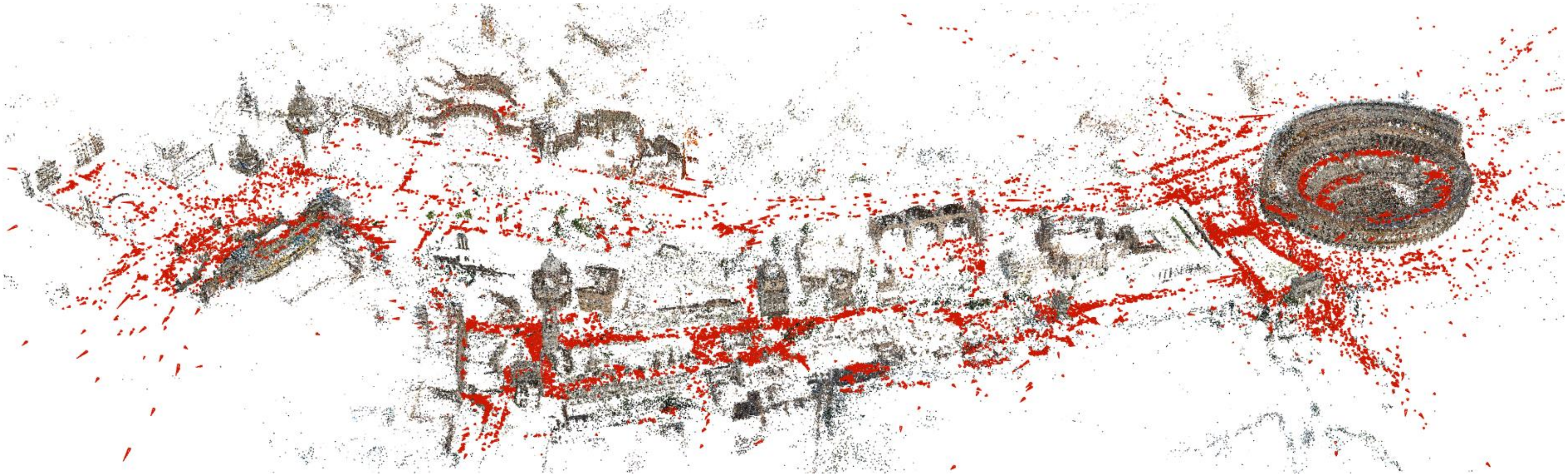
Total reconstruction time: 23 hours

Number of cores: 352

Building Rome in a Day, Agarwal et al. ICCV 2009

Slide courtesy of Noah Snavely

Large-scale structure from motion



Result using COLMAP: Schönberger et al. CVPR ’16