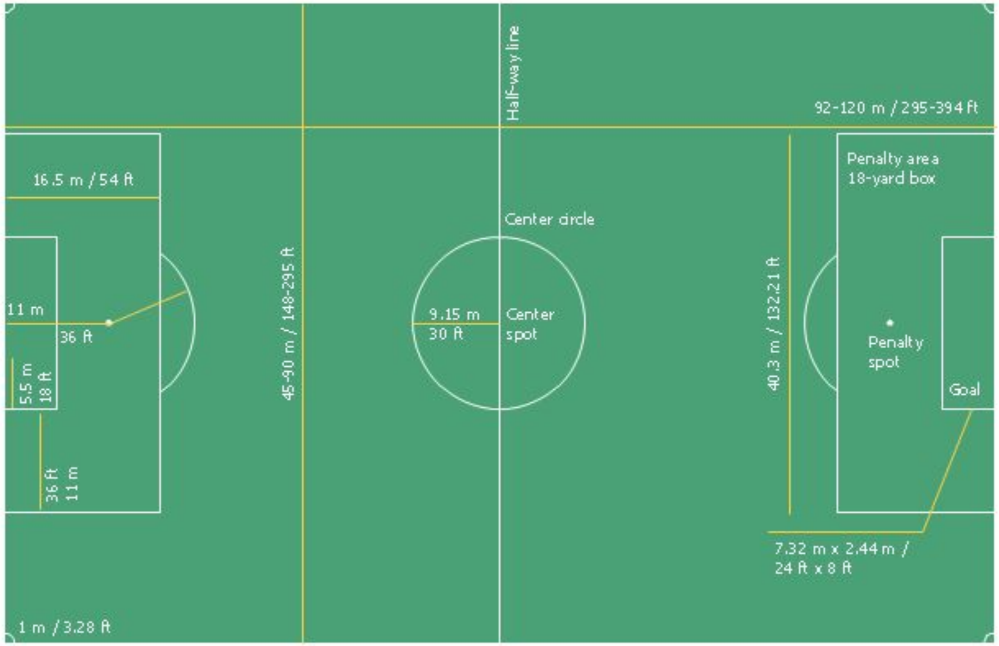
Recitation 6 - a lot of cameras

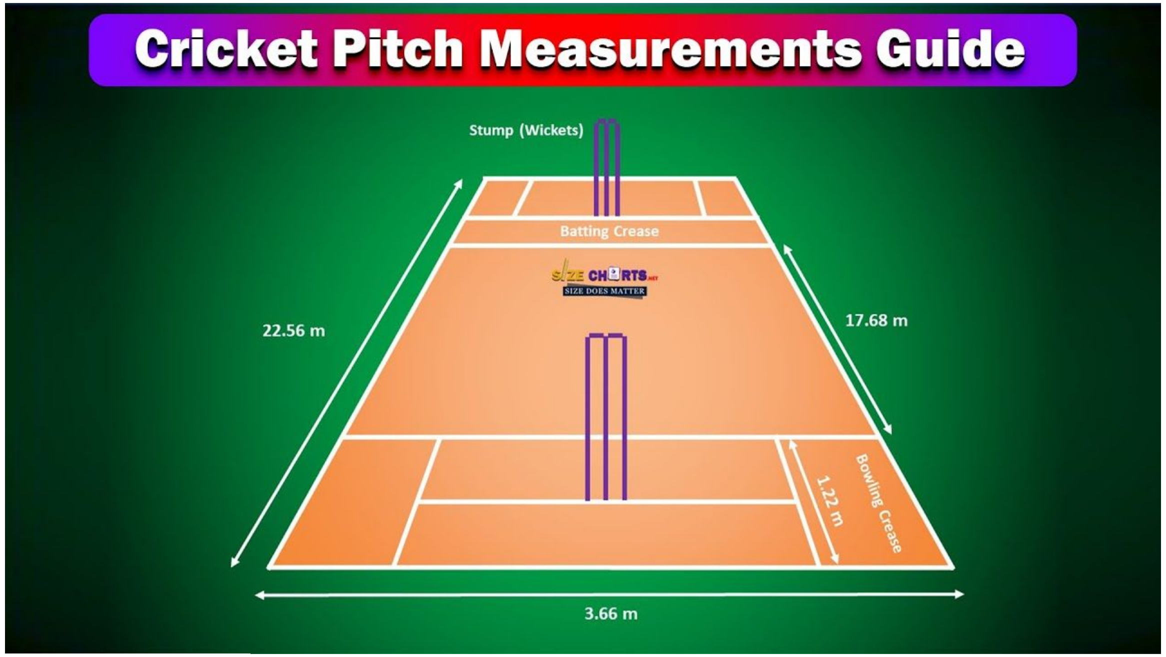
Raymond Yu

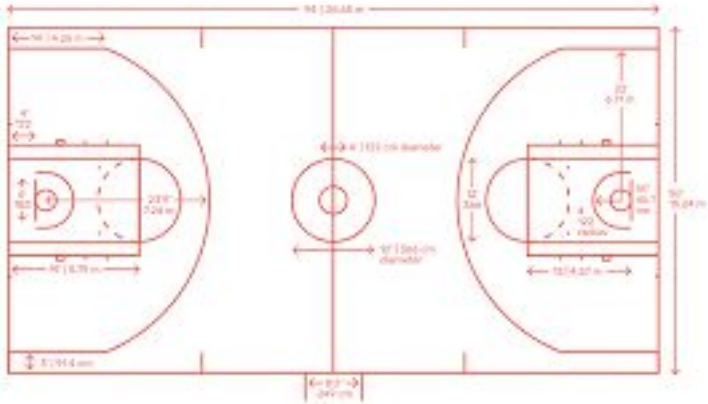
Recap on Camera Calibrations

- Want to find a way to relate real world coordinates and image coordinates
- Do that with a matrix that transforms between points
- Let’s build intuition from the ground up
- What’s a domain where you might have an important and well defined real world coordinate system?

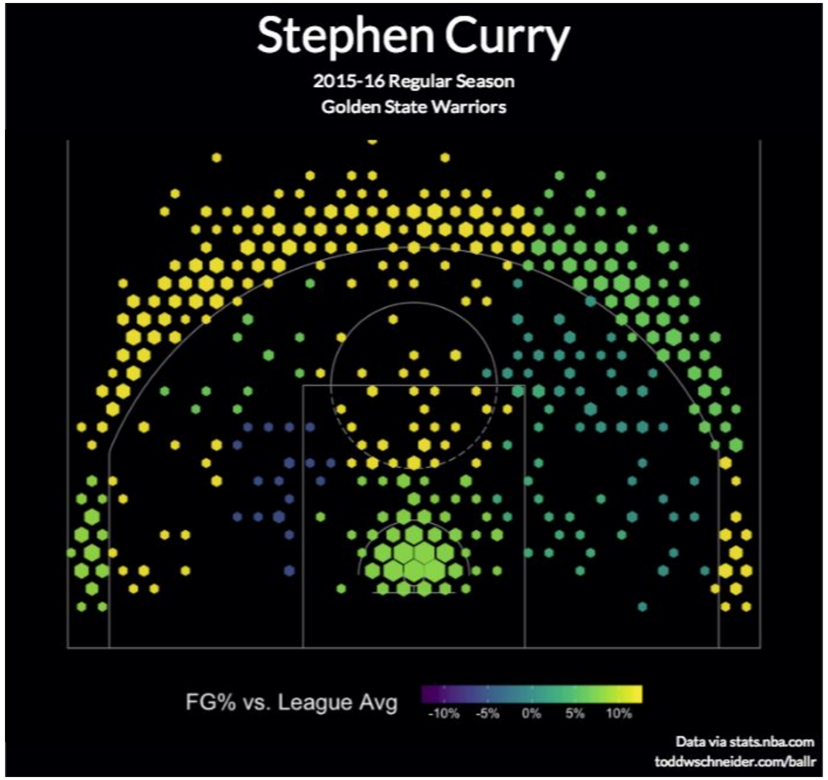
Sports!

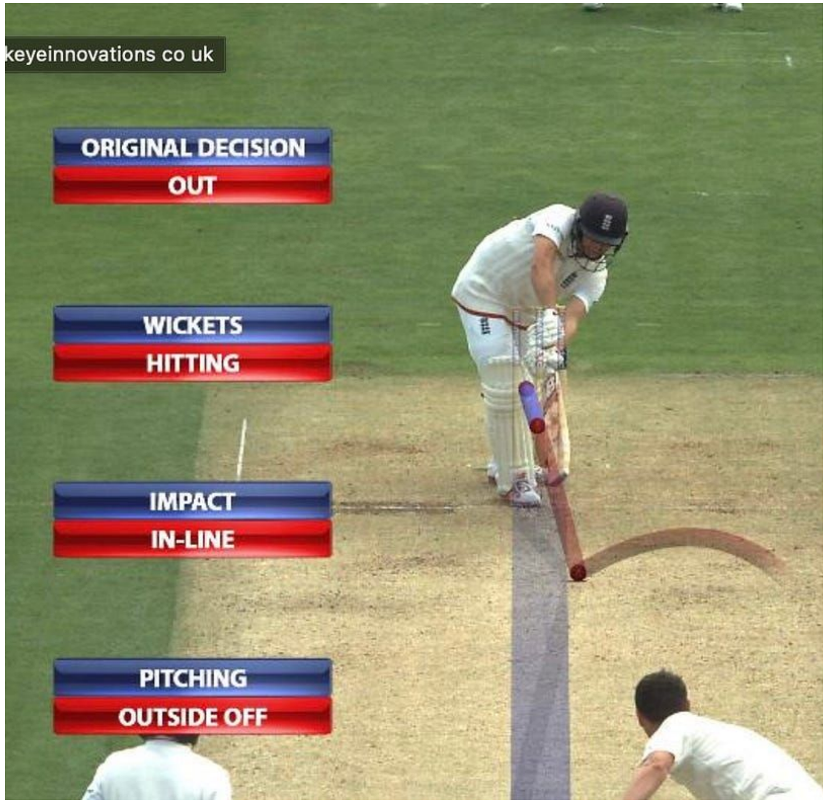

Half-way line
92-120 m / 295-394 ft
16.5 m / 54 ft
Penalty area
18-yard box
Center circle
11 m
36 ft
45-90 m / 148-295 ft
9.15 m
30 ft
Center
spot
40.3 m / 132.21 ft
Penalty
spot
5.5 m
18 ft
Goal
36 ft
11 m
7.32 m x 2.44 m /
24 ft x 8 ft
1 m / 3.28 ft


Cricket Pitch Measurements Guide
Stump (Wickets)
Batting Crease
SIZE DOES MATTER
22.56 m
17.68 m
1.22 m
Bowling Crease
3.66 m



Camera Calibration Underlies all Technology in Sports





Let’s Start With a Simple Problem

Let’s Start With a Simple Problem

Messi scored a goal from here. We have an image and want to know where he is

Object detector says he’s at pixel value (300, 600).... that’s not useful

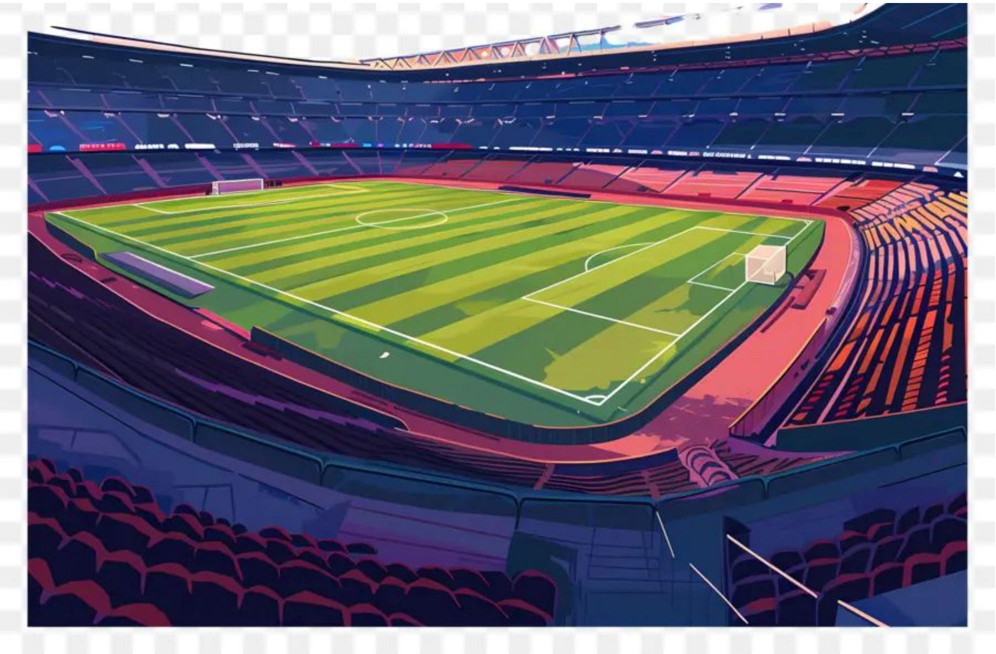
Now consider 3D

$$\mathbf{K} = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix}$$

K is called the camera intrinsics

Potential Issues?

Potential Issues?

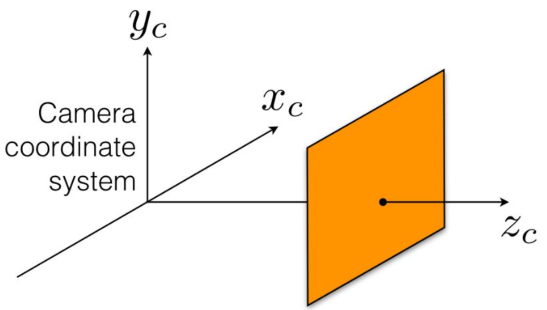


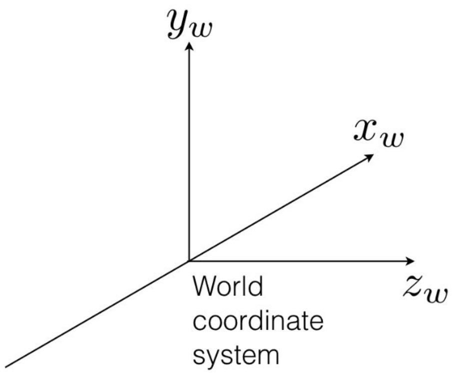
Real world coordinate system isn't aligned with camera coordinate system!

Assumes that the **camera** and **world** share the same coordinate system

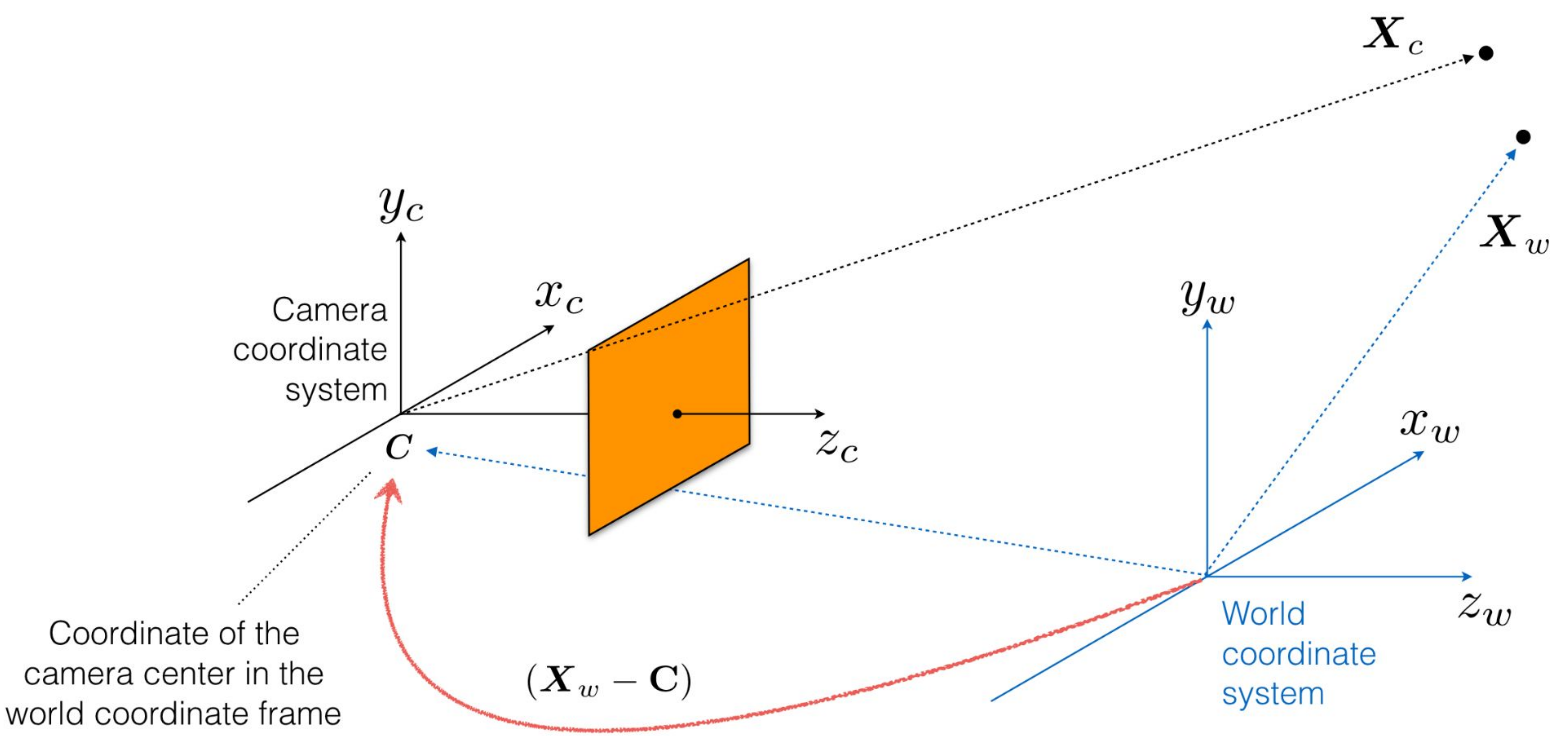
$$\mathbf{P} = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

What if they are different?
How do we align them?





https://www.cs.cmu.edu/~16385/s17/Slides/11.1_Camera_matrix.pdf

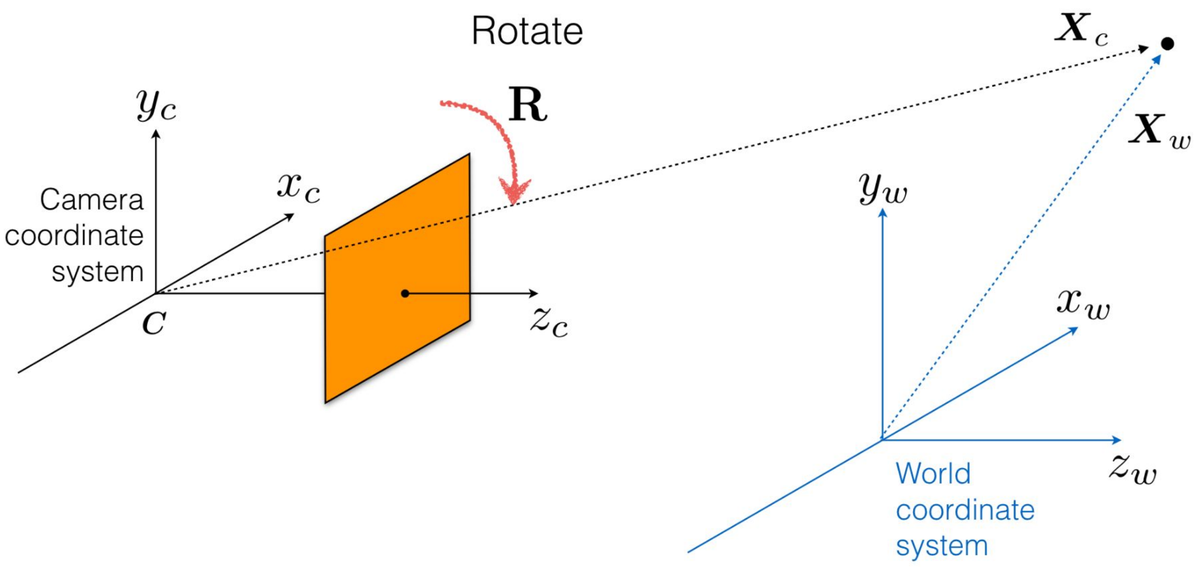


$$(\boldsymbol{X}_w - \mathbf{C})$$

Translate

https://www.cs.cmu.edu/~16385/s17/Slides/11.1_Camera_matrix.pdf

What happens to points after alignment?



$$\mathbf{R}(\boldsymbol{X}_w - \mathbf{C})$$

Rotate Translate

https://www.cs.cmu.edu/~16385/s17/Slides/11.1_Camera_matrix.pdf

What is the meaning of each matrix of the camera matrix decomposition?

$$\mathbf{P} = \mathbf{KR}[\mathbf{I} | -\mathbf{C}]$$

3x3
intrinsics

3x3
3D rotation

3x3
identity

3x1
3D translation

https://www.cs.cmu.edu/~16385/s17/Slides/11.1_Camera_matrix.pdf

The story so far

Model the 3D-to-2D camera projection: $\tilde{\mathbf{x}}^{\mathrm{I}} \sim \mathbf{P}\widetilde{\mathbf{X}}^{\mathrm{W}}$ with $\mathbf{P} = \mathbf{K}[\mathbf{R}|\mathbf{t}]$

Calibrate cameras (get $\mathbf{K}[\mathbf{R}|\mathbf{t}]$) from N 2D-**3D** correspondences $(\tilde{\mathbf{x}}_i^{\mathrm{I}},\ \widetilde{\mathbf{X}}_i^{\mathrm{W}})$

- cast constraints (2D-3D correspondences) as a linear system $\mathbf{A}\ \mathbf{p} = \mathbf{0}$
- total least squares ($argmin_{\mathbf{x}}\ \|\mathbf{A}\ \mathbf{p}\|^2 s.t.\ \|\mathbf{p}\|^2 = 1$) gives the best approximation
- closed-form solution via the SVD of $\mathbf{A}$ (its last right singular vector) & Cholesky
- refine by minimizing the 2D reprojection error $\sum_i \|\mathrm{proj}(\mathbf{K}[\mathbf{R}\mid\mathbf{t}]\mathbf{X}_\mathbf{i};\boldsymbol{\kappa}) - \boldsymbol{x}_i\|^2$

Quiz

- Why do we project real world 3D points to image points and not the other way?
- Does a point in an image defined a unique 3D point? What does it define?
- What additional information could you use to know the unique 3D point corresponding to an image?

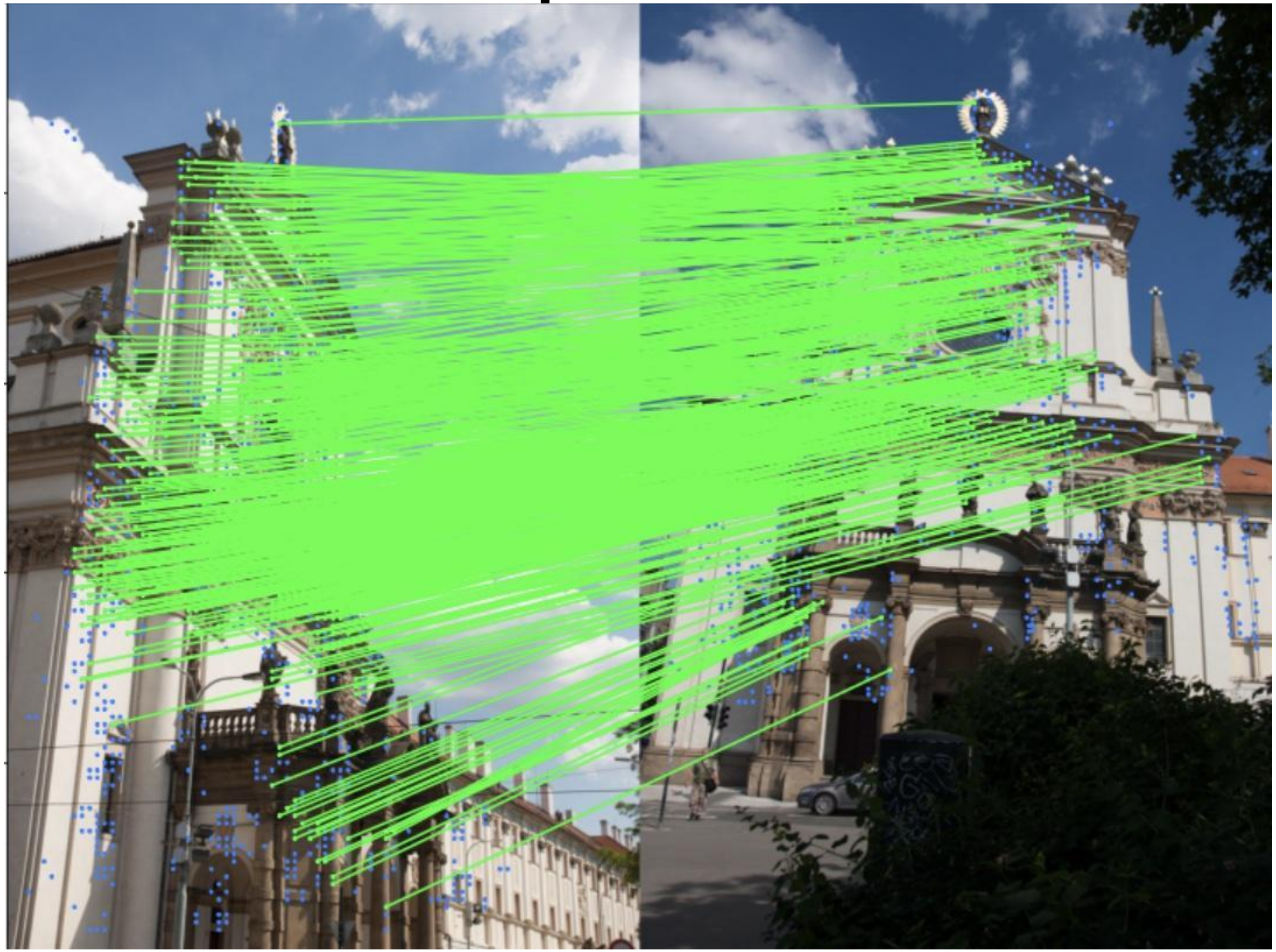
In general, we don’t have **3D** measurements…

… but more than 1 image!

What are the geometric constraints governing **multiple views** of the same scene?

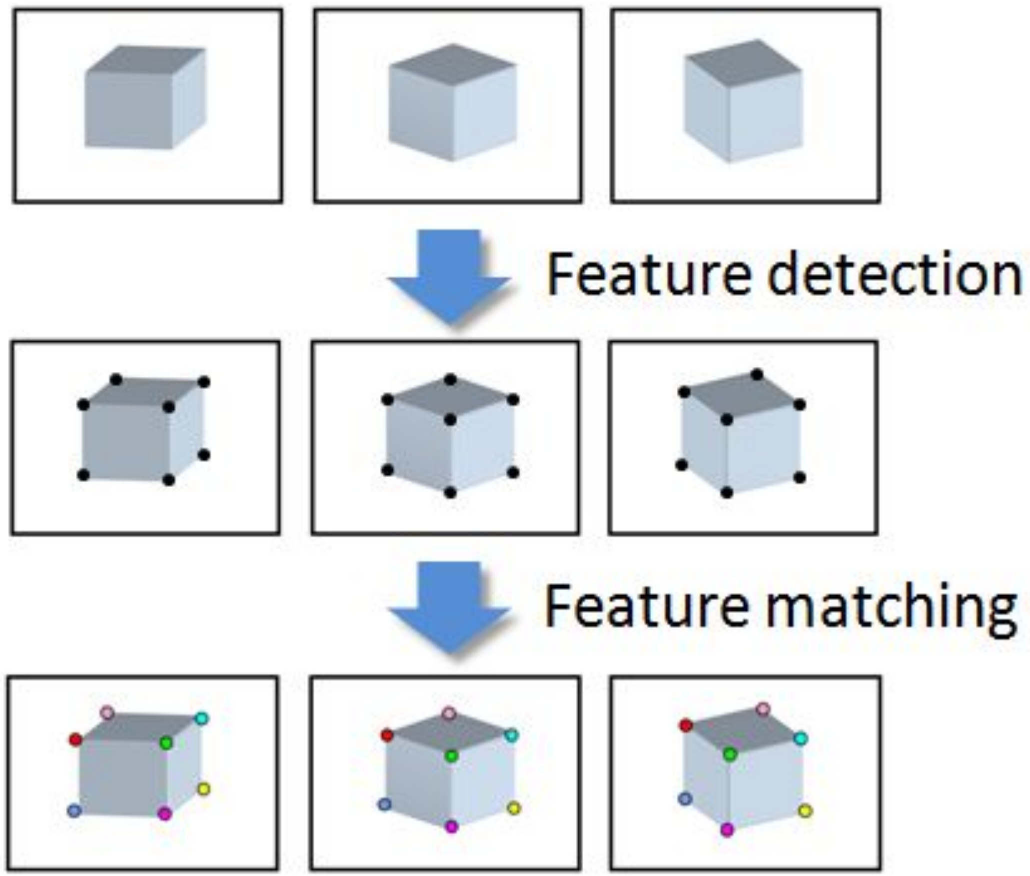
2D correspondences!

Get 3D structure & motion from 2D correspondences



https://kornia.readthedocs.io/en/latest/applications/image_matching.html

Correspondence estimation

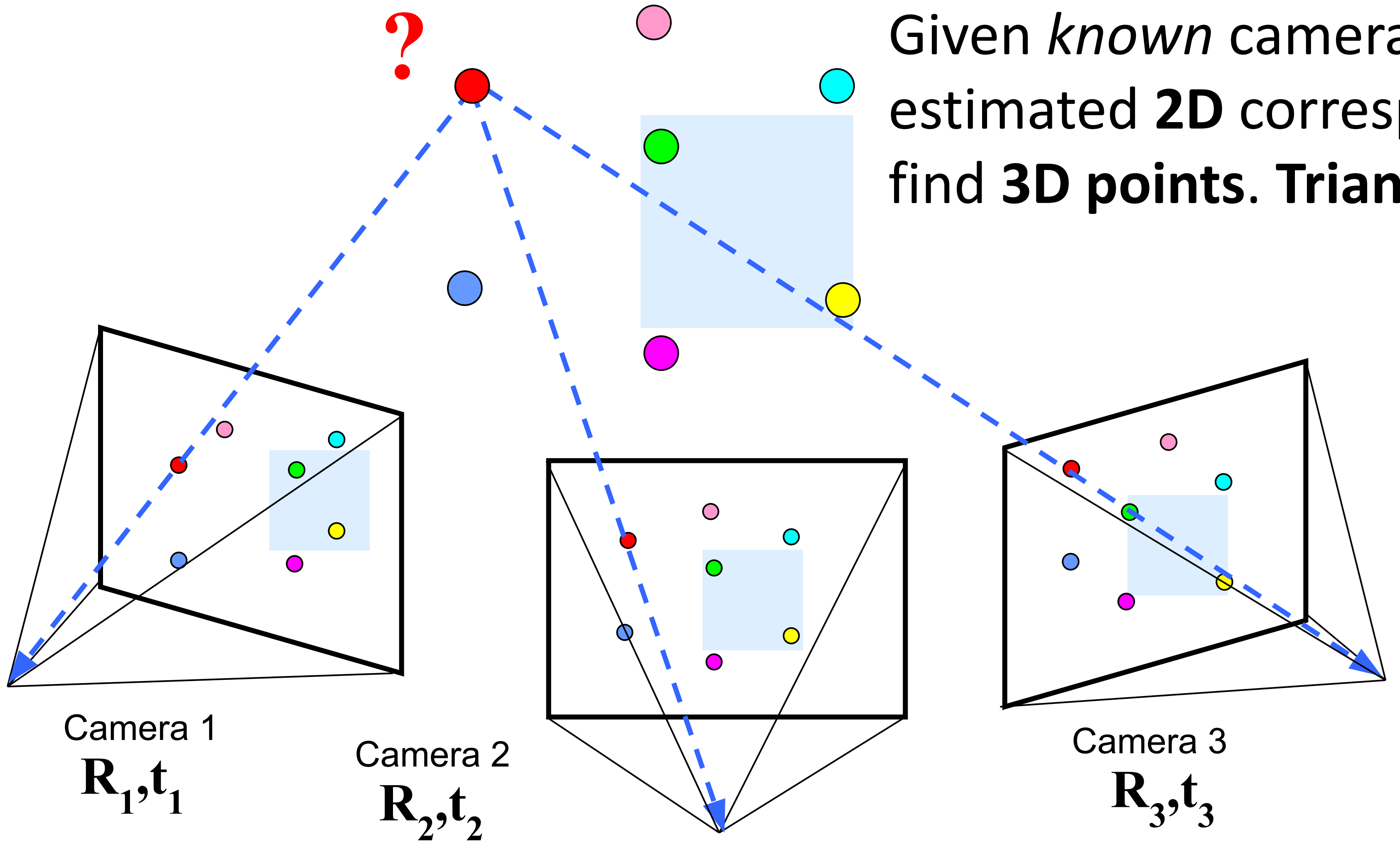


Why do we care about 3D reconstruction?

- Mapping, Localization, Navigation for Robots, Drones, Cars (cf. also visual SLAM)
- AR (e.g., Hololens) and VR (e.g., Oculus)
- Movies (special FX), Digital Preservation, "Photo Tourism", …
- Software: COLMAP (SfM), orb-slam2 / g2o / gtsam (SLAM)
- Hot topic in industry & academia (top category at CVPR)

Multi-view geometry problems

Recovering structure:
Given *known* cameras and estimated **2D** correspondences, find **3D points**. **Triangulation!**

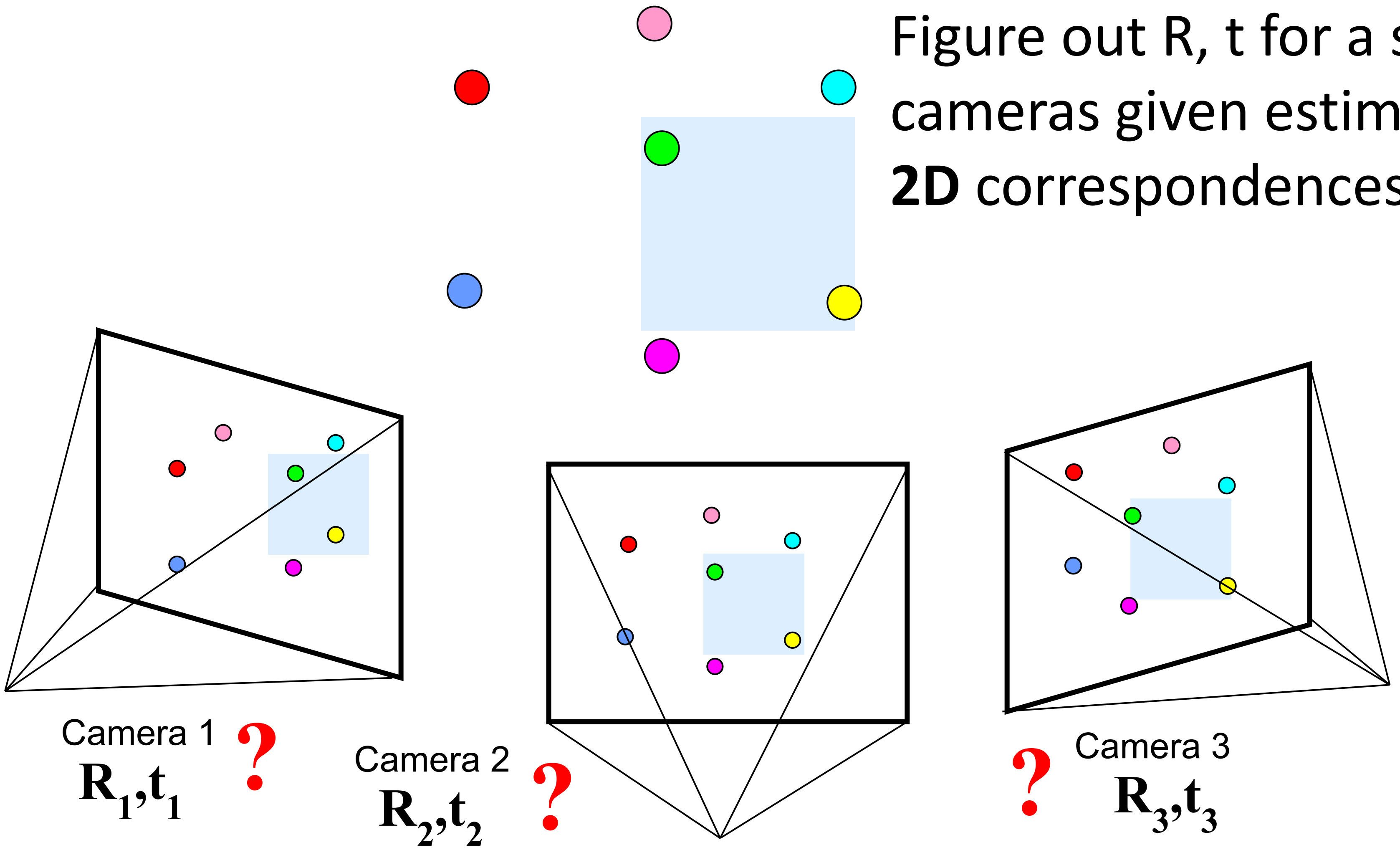


Slide credit:
Noah Snavely

Multi-view geometry problems

Motion:

Figure out R, t for a set of cameras given estimated **2D** correspondences



Slide credit: Noah Snavely

Why Are Multiple Cameras Important

https://twitter.com/overtime/status/1610436277069041664

What will we learn today?

Triangulation

Epipolar geometry

Stereo

Structure-from-Motion (SfM)

What will we learn today?

Triangulation

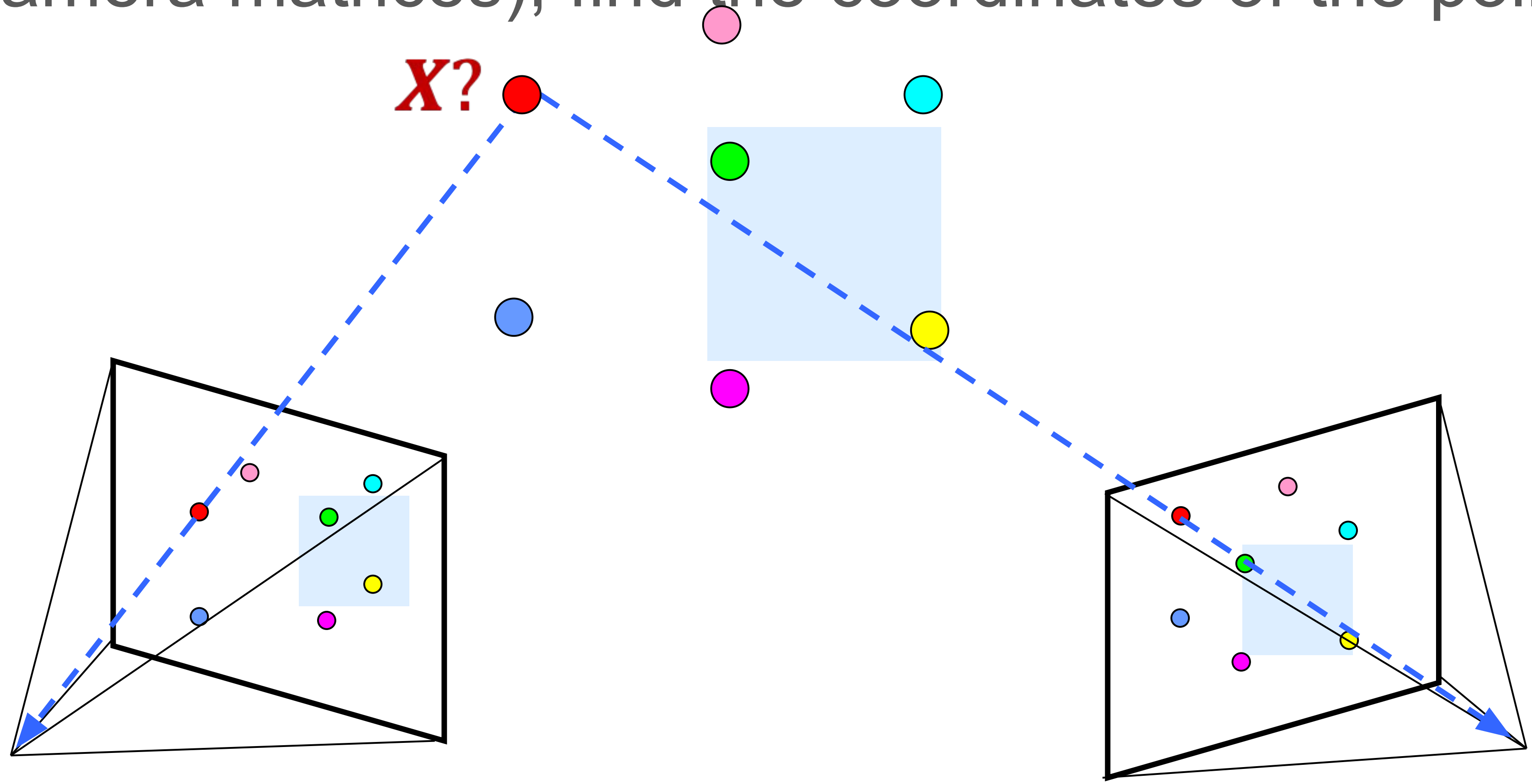
Epipolar geometry

Stereo

Structure-from-Motion (SfM)

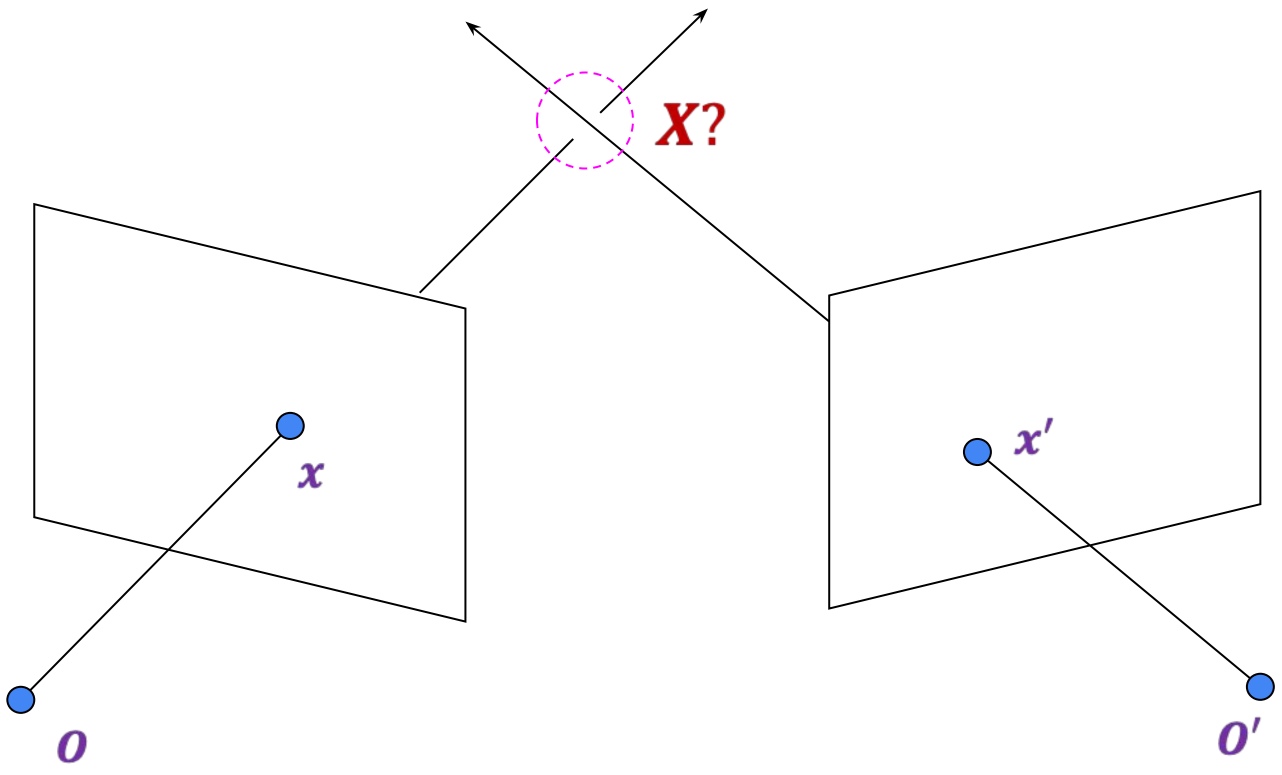
Triangulation

Given projections of a 3D point in two or more images (with known camera matrices), find the coordinates of the point



Triangulation

Given projections of a 3D point in two or more images (with known camera matrices), find the coordinates of the point

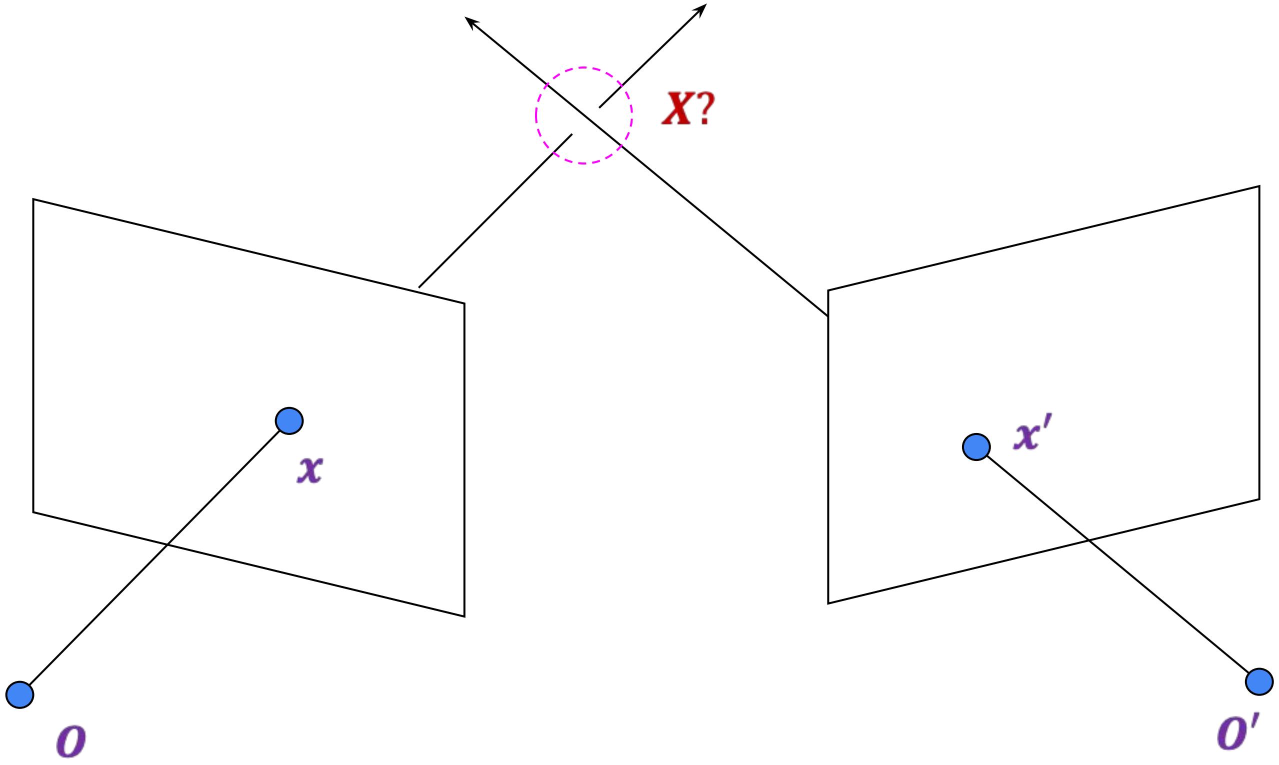


Triangulation

We want to intersect the two visual rays corresponding to $\boldsymbol{x}$ and $\boldsymbol{x}'$

But do they always intersect *exactly*?

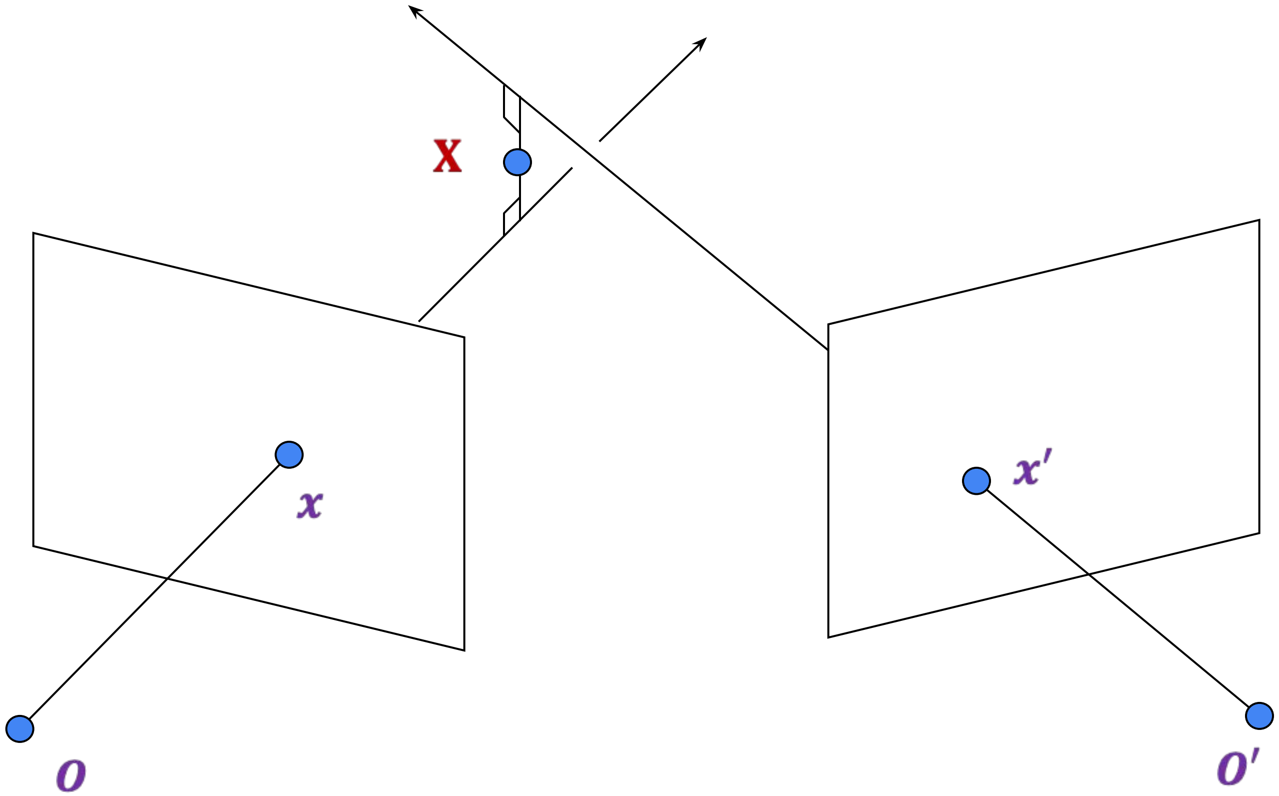
No! Noise in 2D matching or numerical errors



Triangulation: linear approach

Find the shortest segment connecting the two viewing rays

Let $\mathbf{X}$ be the midpoint of that segment: solve for $\mathbf{X}$!



what is $\mathbf{A}$?

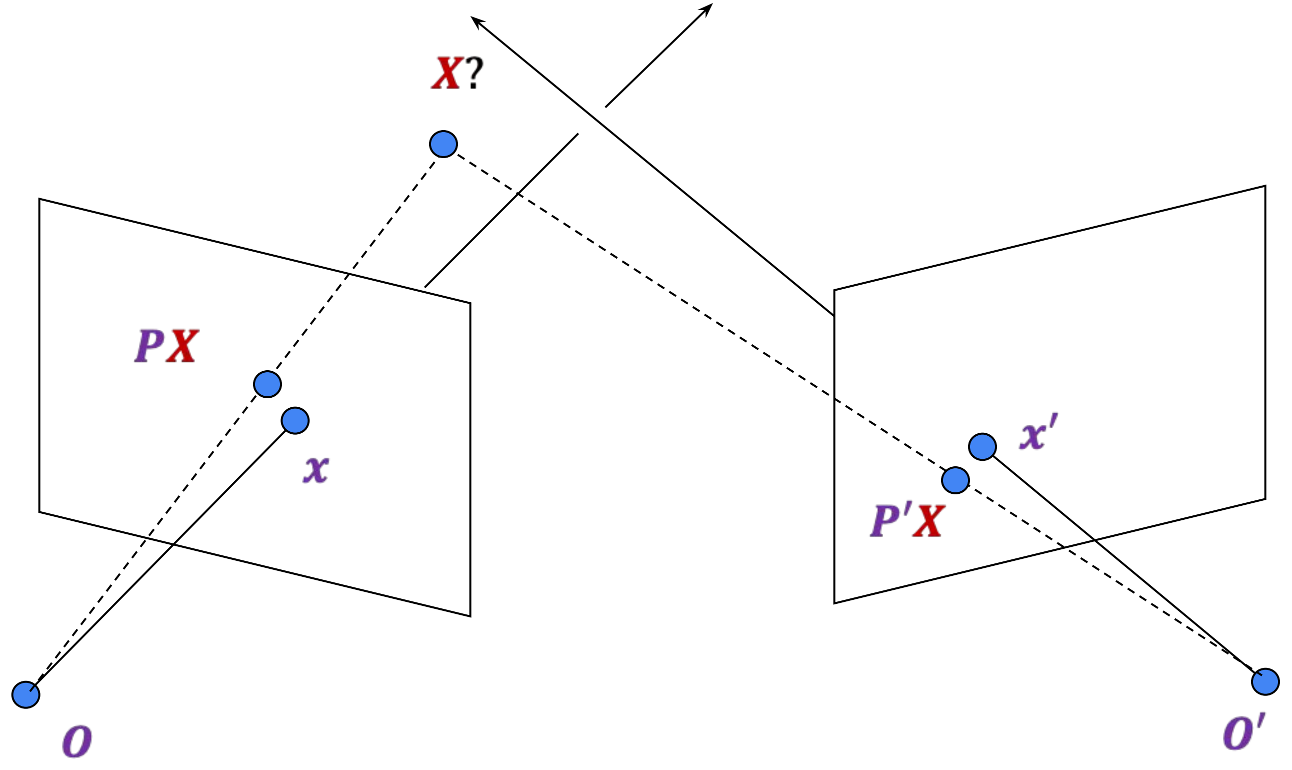
(answer in appendix)

As for calibration: constraints $(\mathbf{x} \sim \mathbf{P}\mathbf{X},\ \mathbf{x}' \sim \mathbf{P}'\mathbf{X}) \rightarrow \mathbf{A}\mathbf{X} = \mathbf{0} \rightarrow$ SVD of $\mathbf{A}$

Triangulation: *non-linear* approach

Find $\boldsymbol{X}$ that minimizes the 2D reprojection errors

$$\|\mathrm{proj}(\boldsymbol{P}\boldsymbol{X}) - \boldsymbol{x}\|^2 + \|\mathrm{proj}(\boldsymbol{P}'\boldsymbol{X}) - \boldsymbol{x}'\|^2$$



Applications of Triangulation

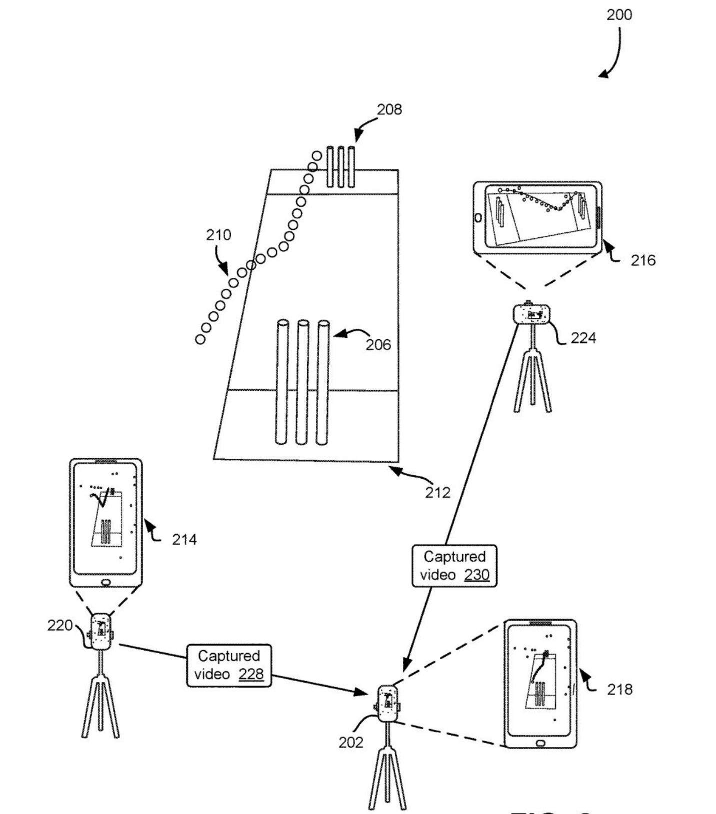


FIG. 2

3D Reconstruction

What will we learn today?

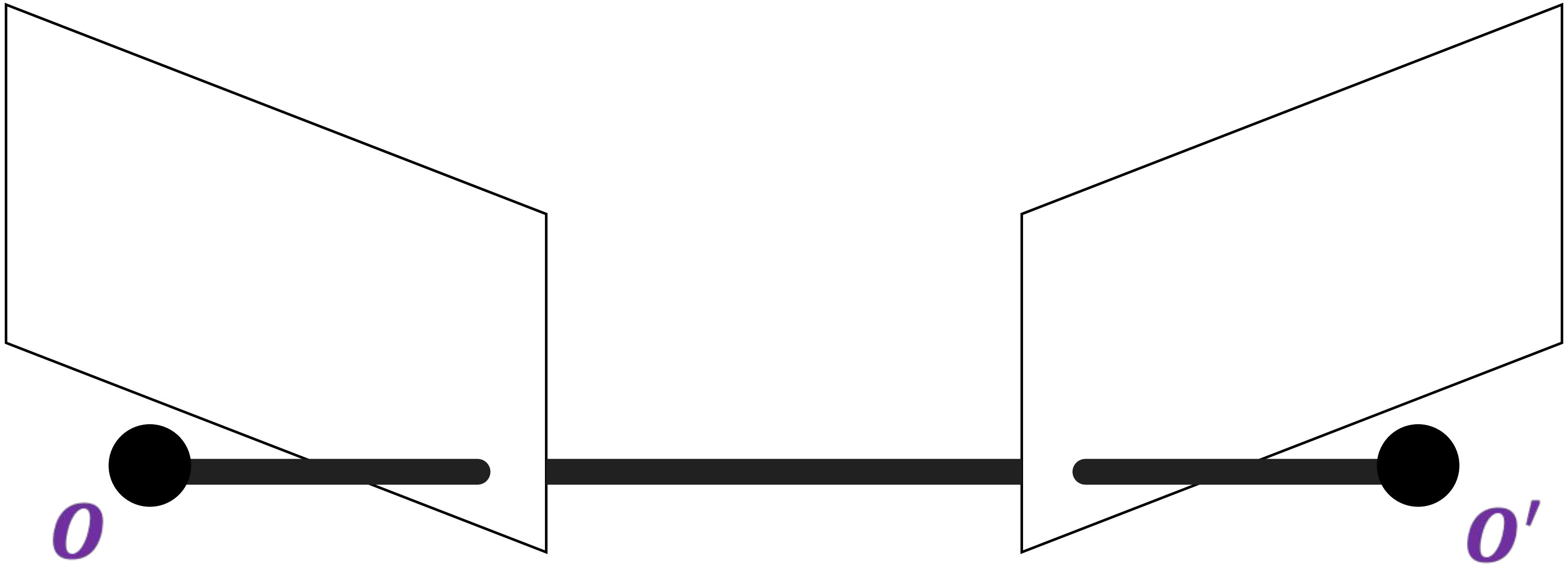
Triangulation

Epipolar geometry

Stereo

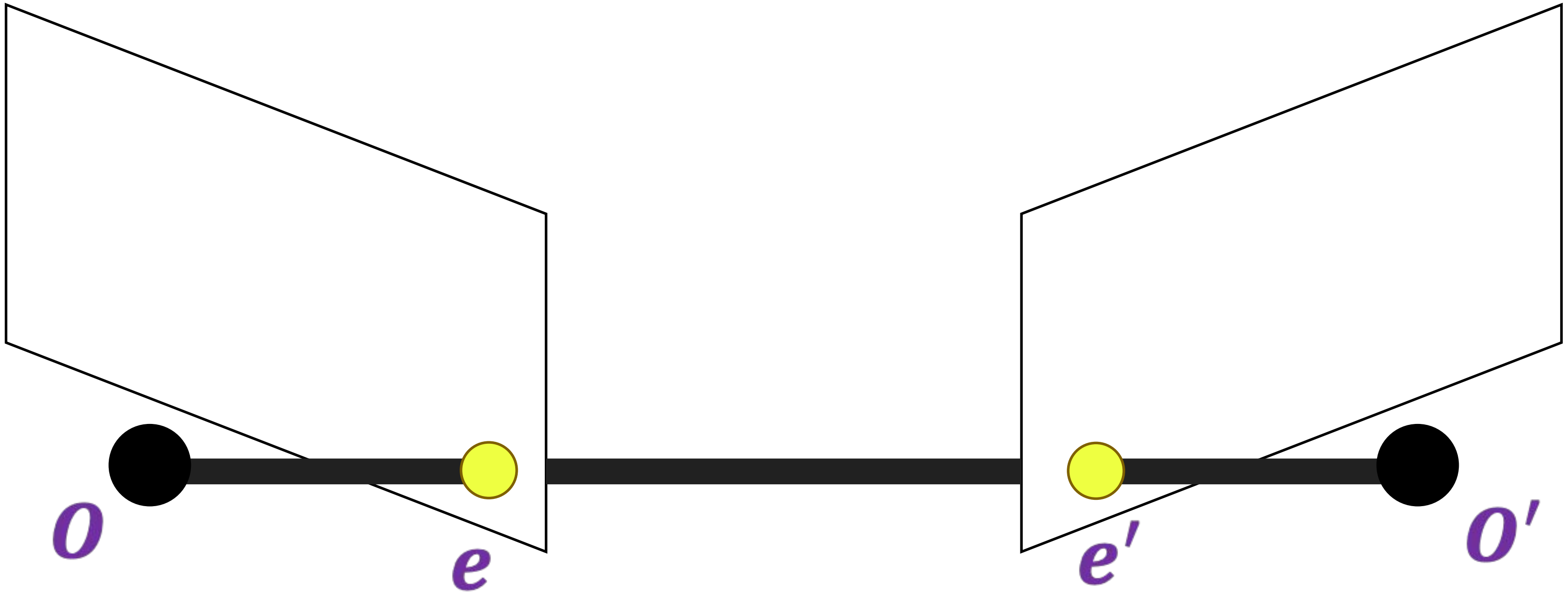
Structure-from-Motion (SfM)

Epipolar geometry setup



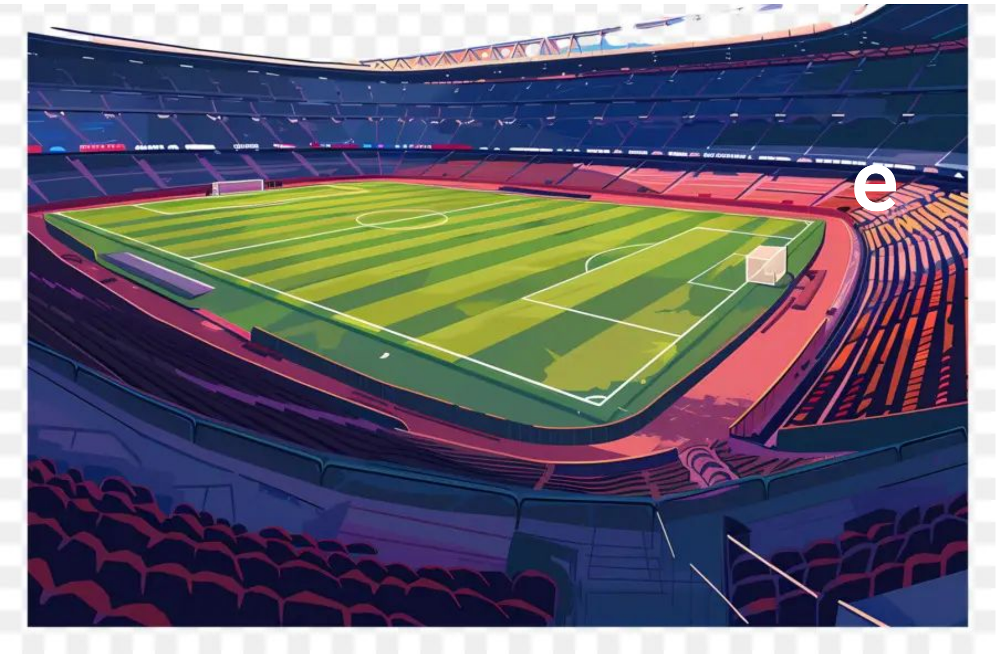
- Suppose we have two cameras with centers O, O'
- The **baseline** is the line connecting the origins

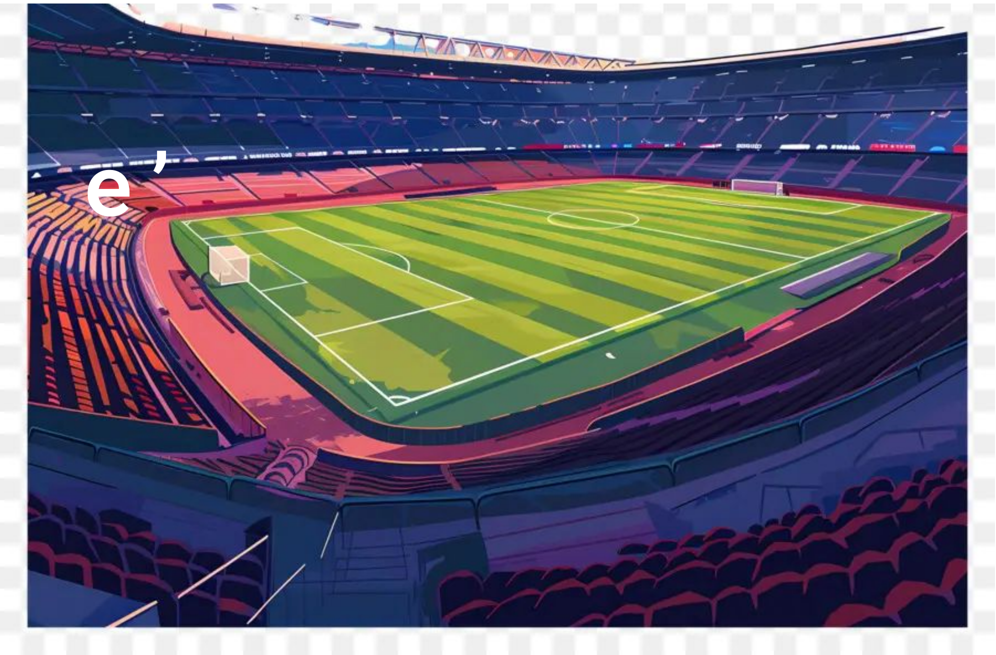
Epipolar geometry setup



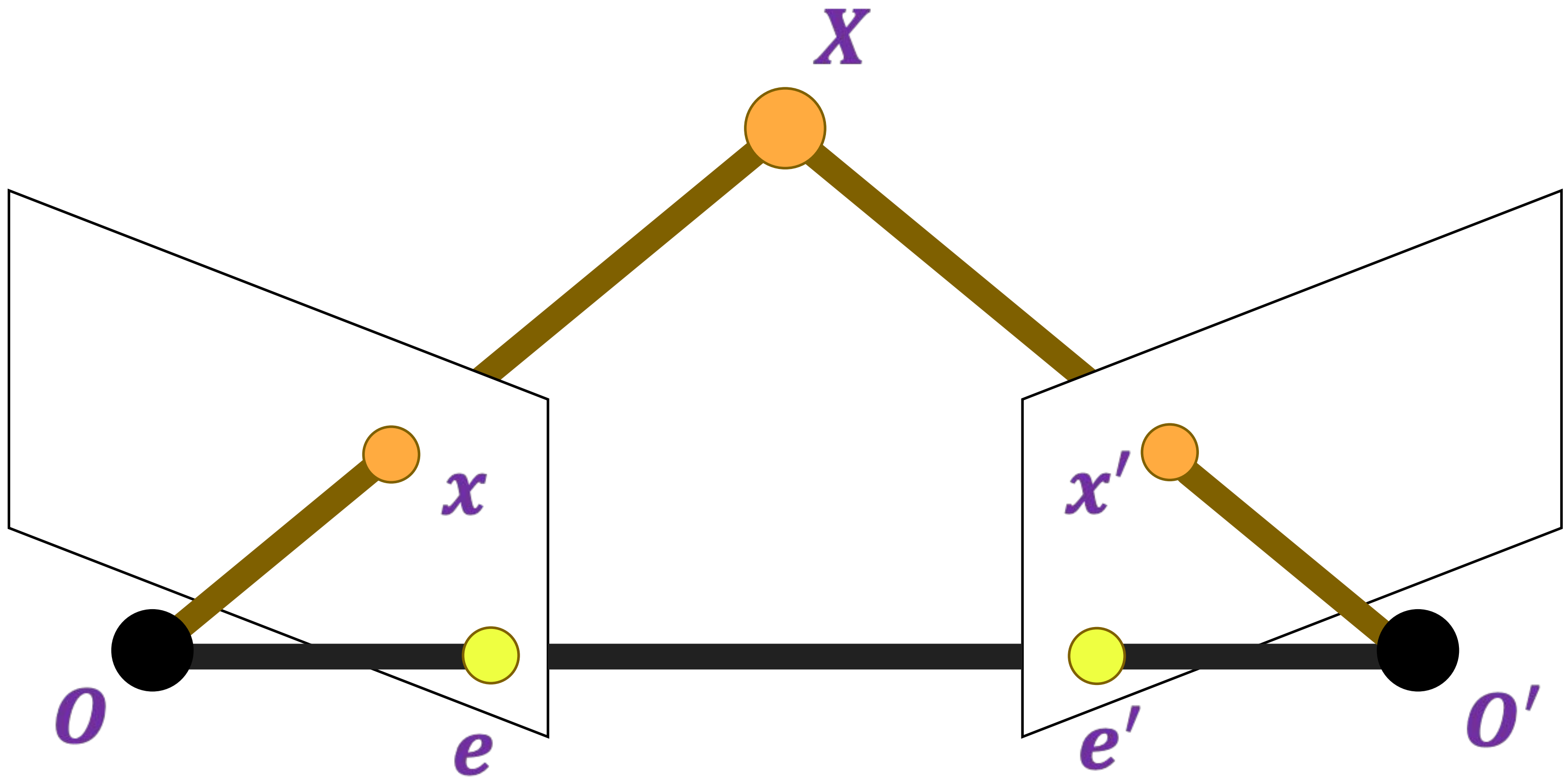
- **Epipoles** e, e' are where the baseline intersects the image planes
- Equivalently: epipoles are projections of the other camera in each view

Epipolar geometry setup



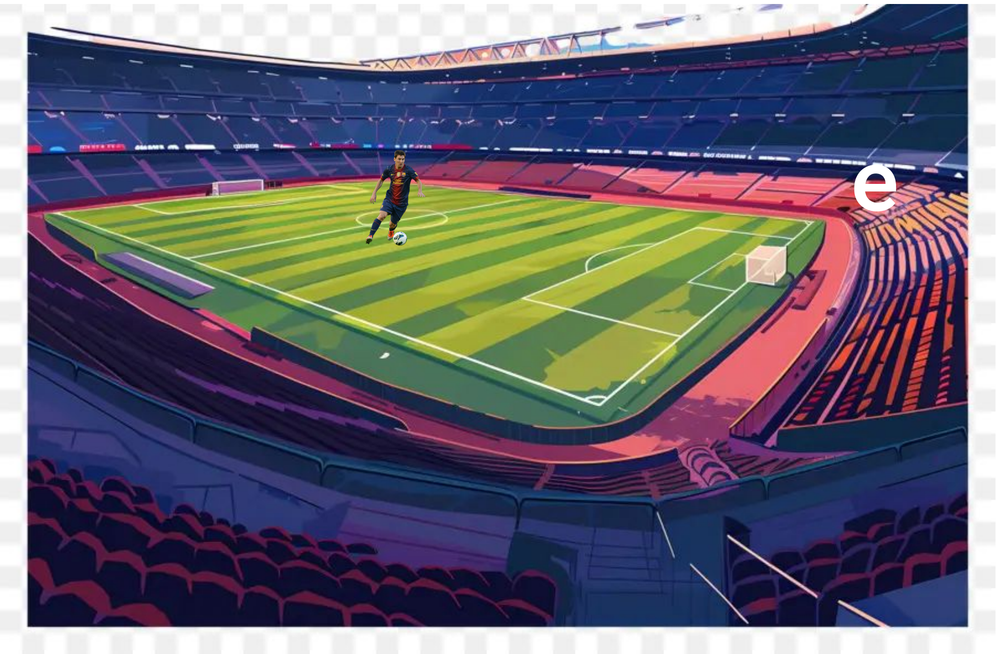


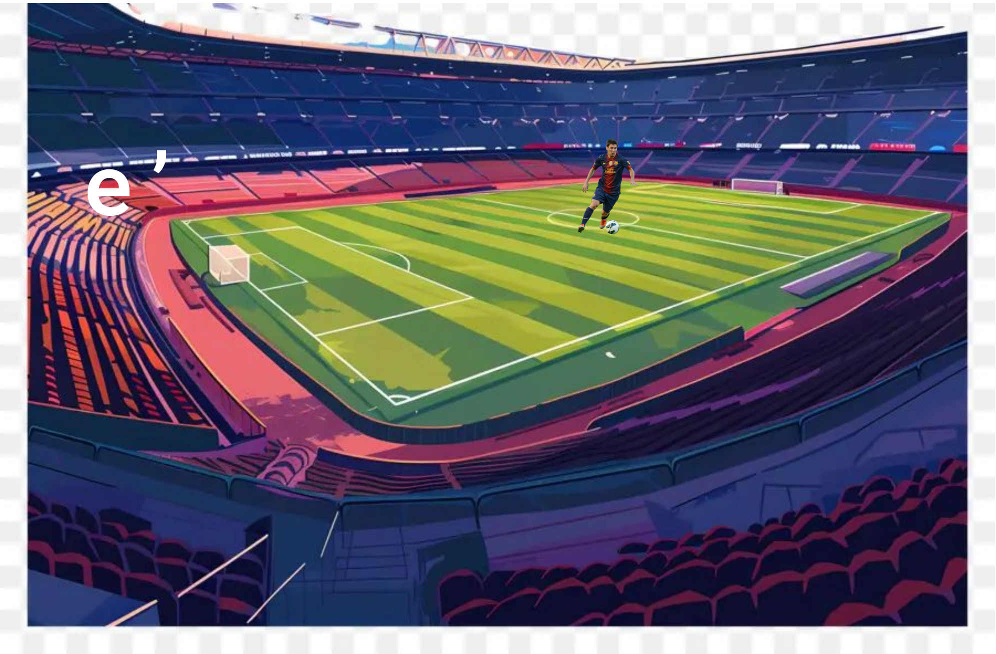
Epipolar geometry setup



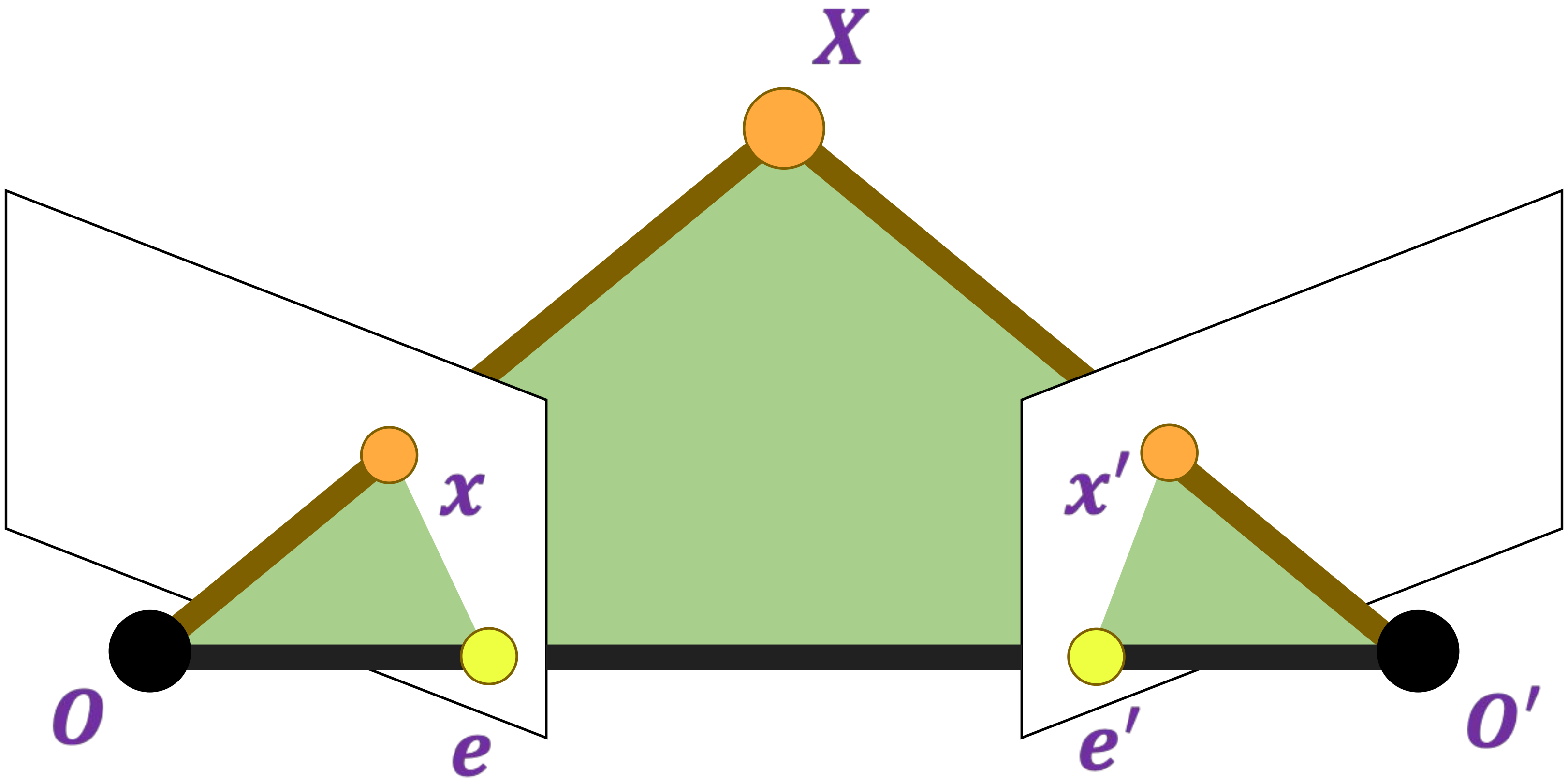
- Consider a **point** $\boldsymbol{X}$, which projects to $\boldsymbol{x}$ and $\boldsymbol{x'}$

Epipolar geometry setup



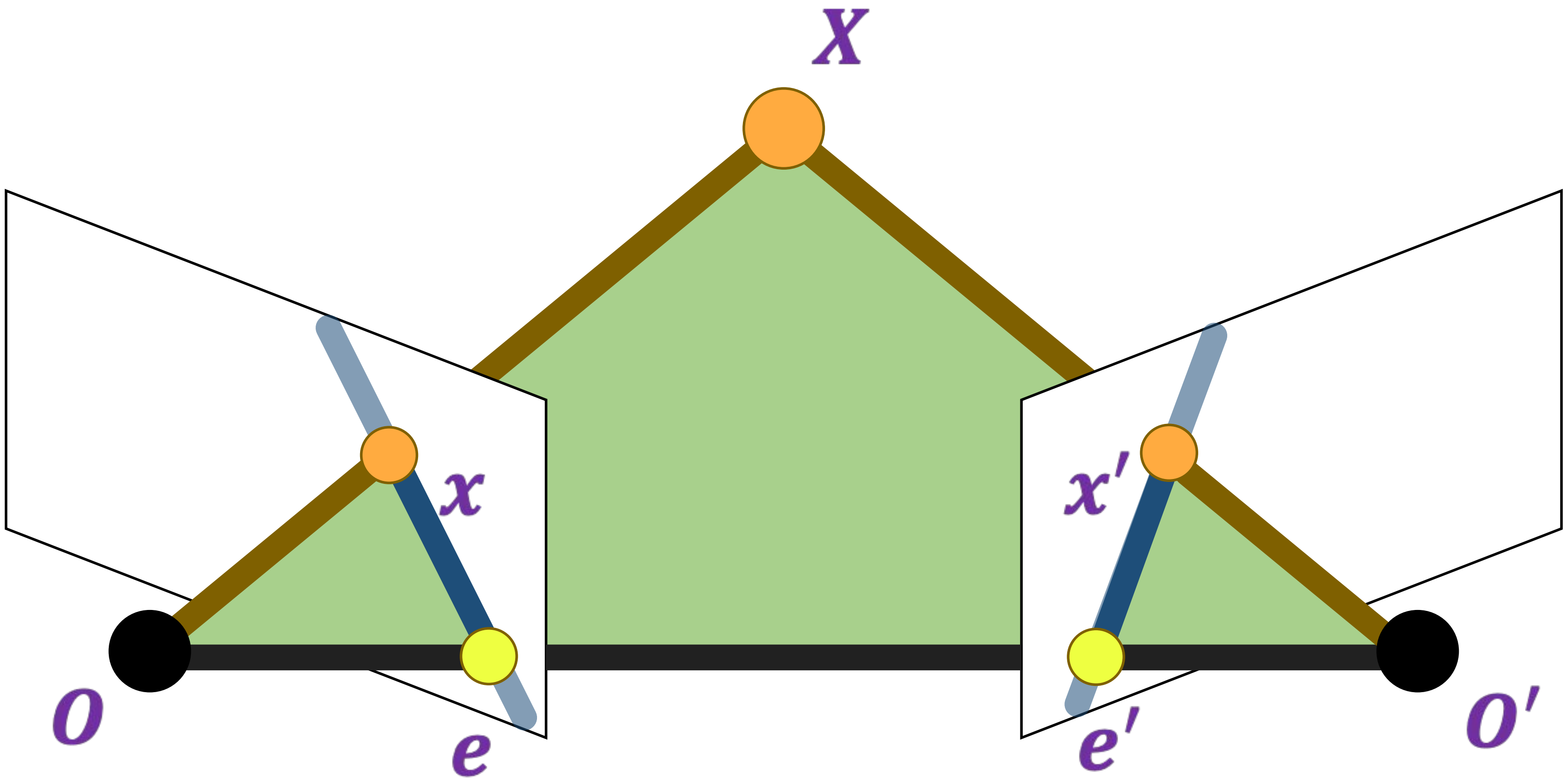


Epipolar geometry setup



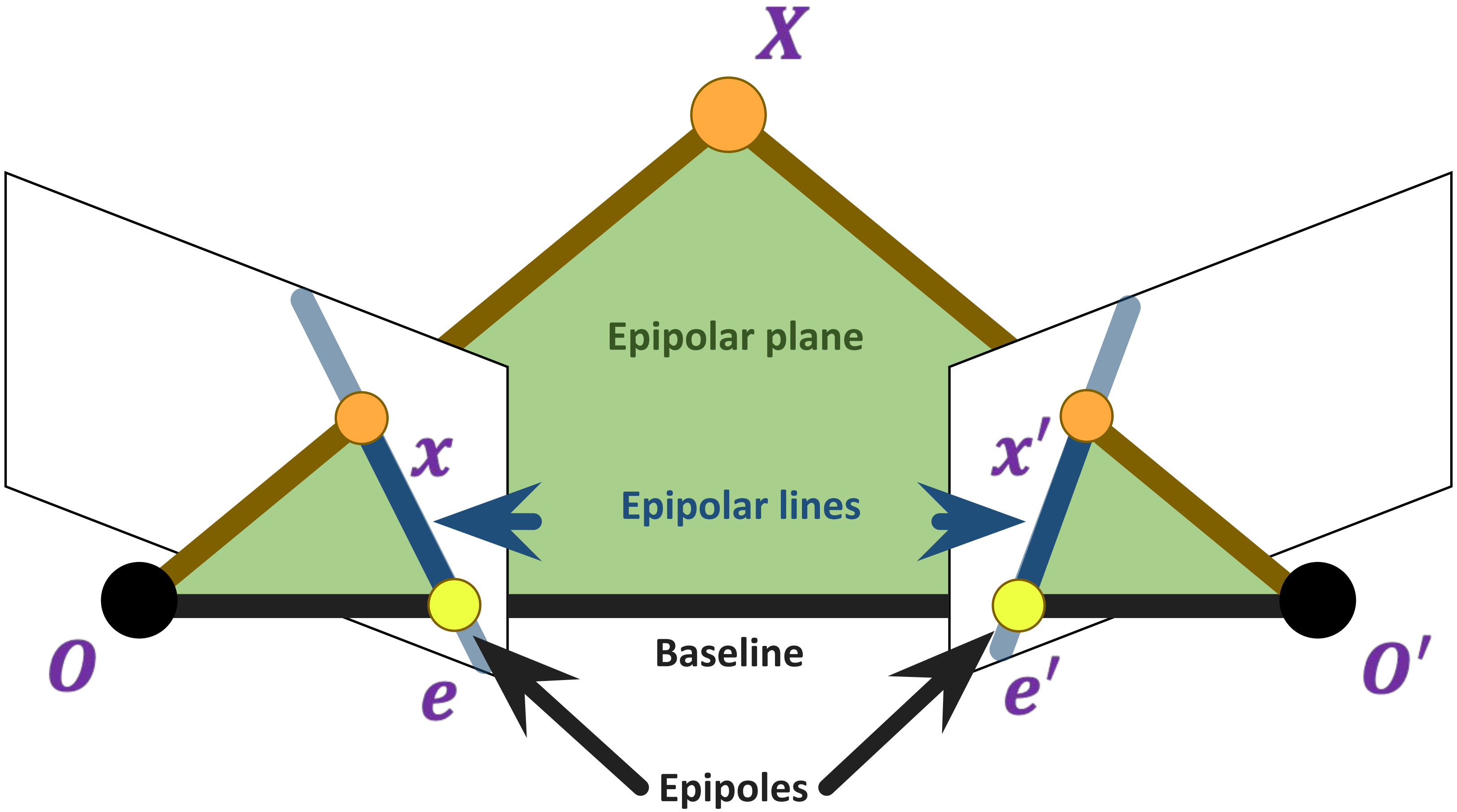
- The plane formed by $\boldsymbol{X}$, $\boldsymbol{O}$, and $\boldsymbol{O'}$ is called an **epipolar plane**

Epipolar geometry setup

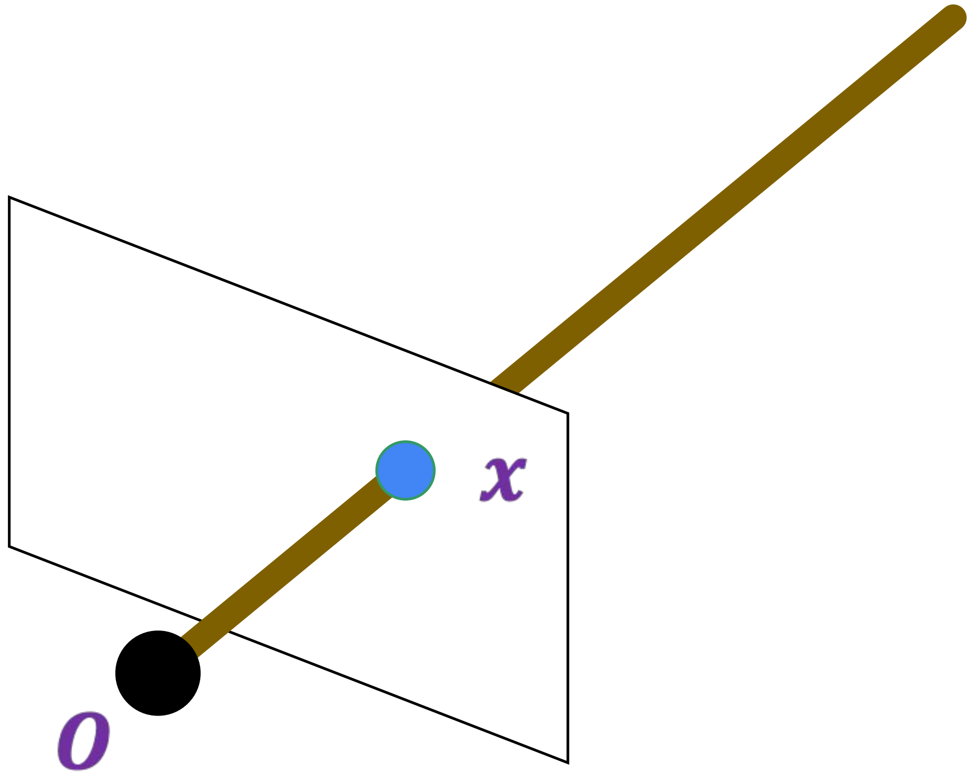


- **Epipolar lines** connect the epipoles to the projections of X
- Equivalently, they are intersections of the epipolar plane with the image planes, come in pairs (for x and x')

Epipolar geometry setup: Summary

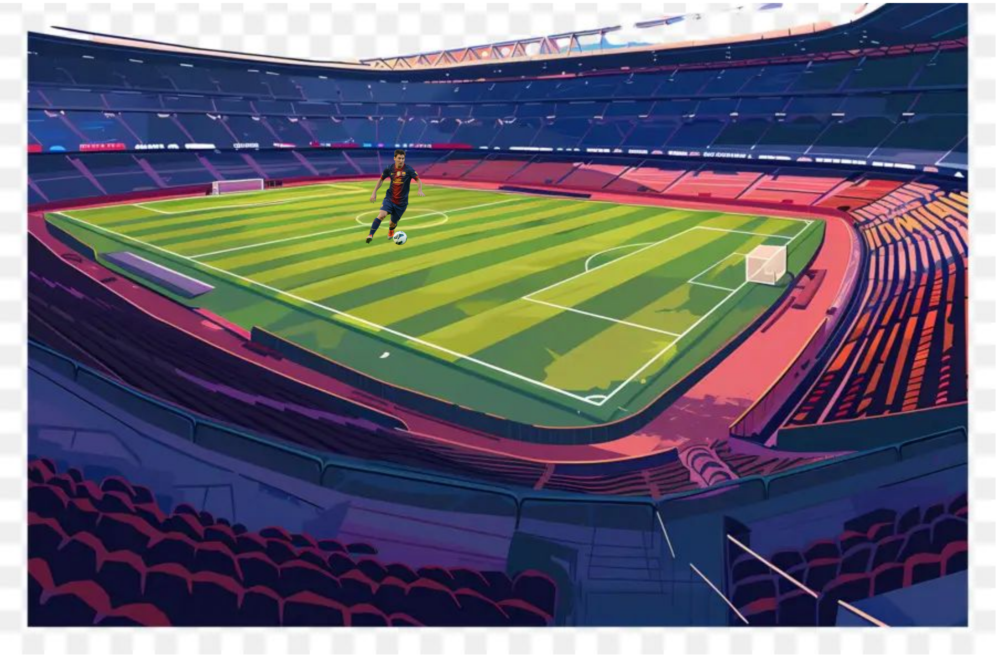


Epipolar constraint

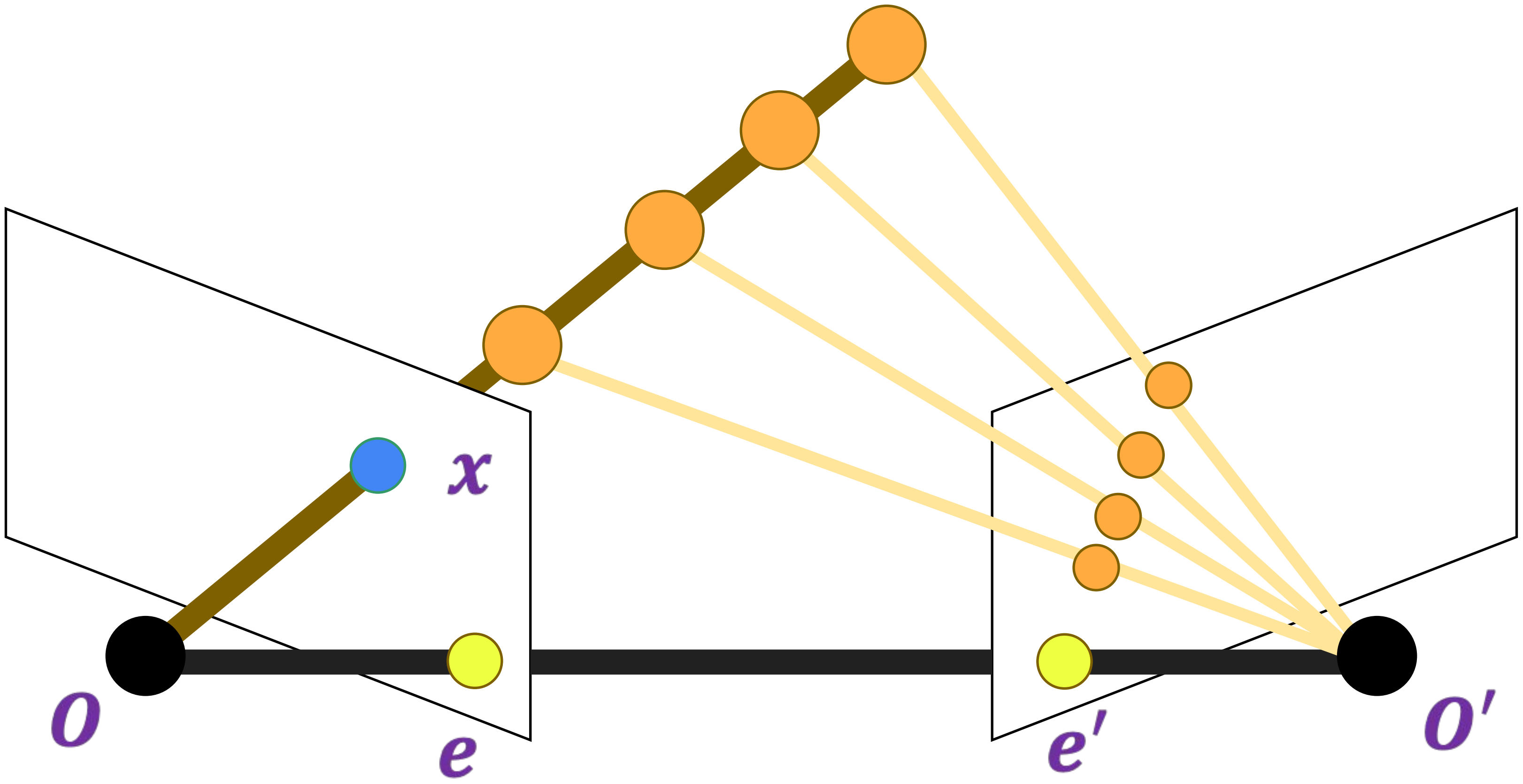


- Suppose we observe a single point x in one image

Epipolar geometry setup

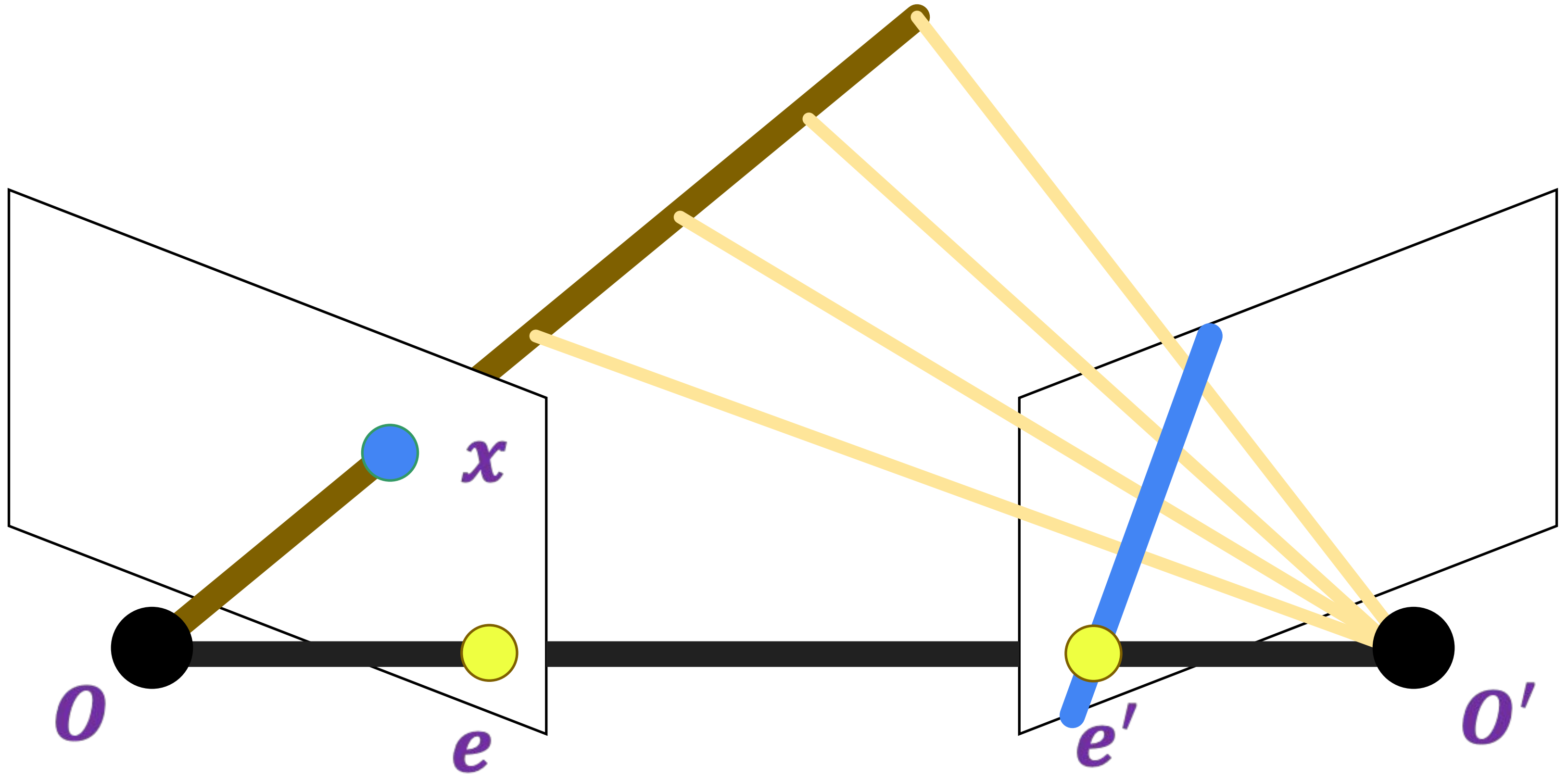


Epipolar constraint



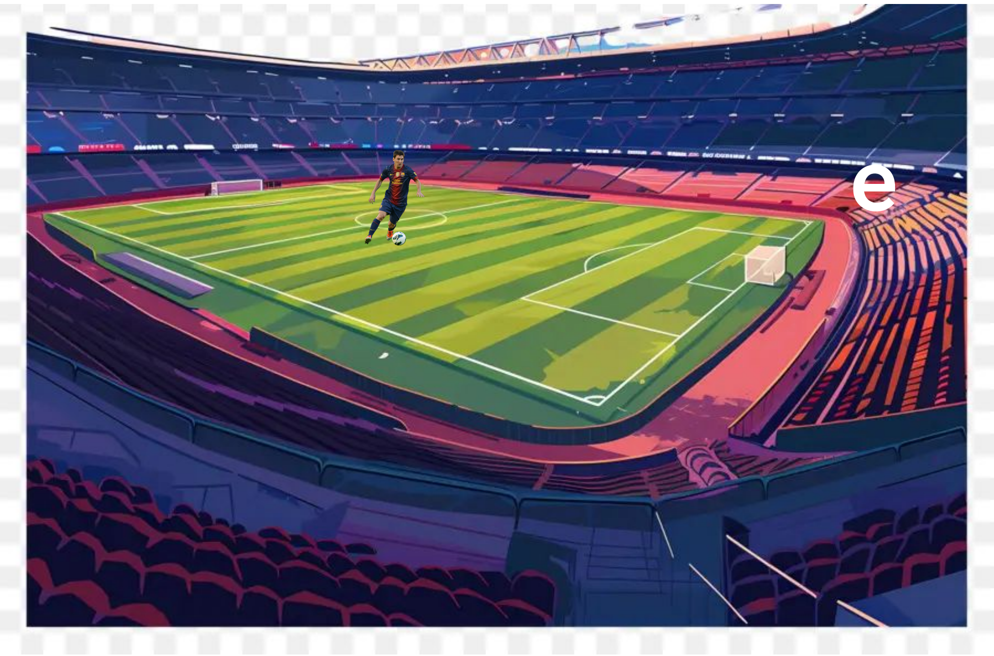
- Where can we find the x' corresponding to x in the other image?

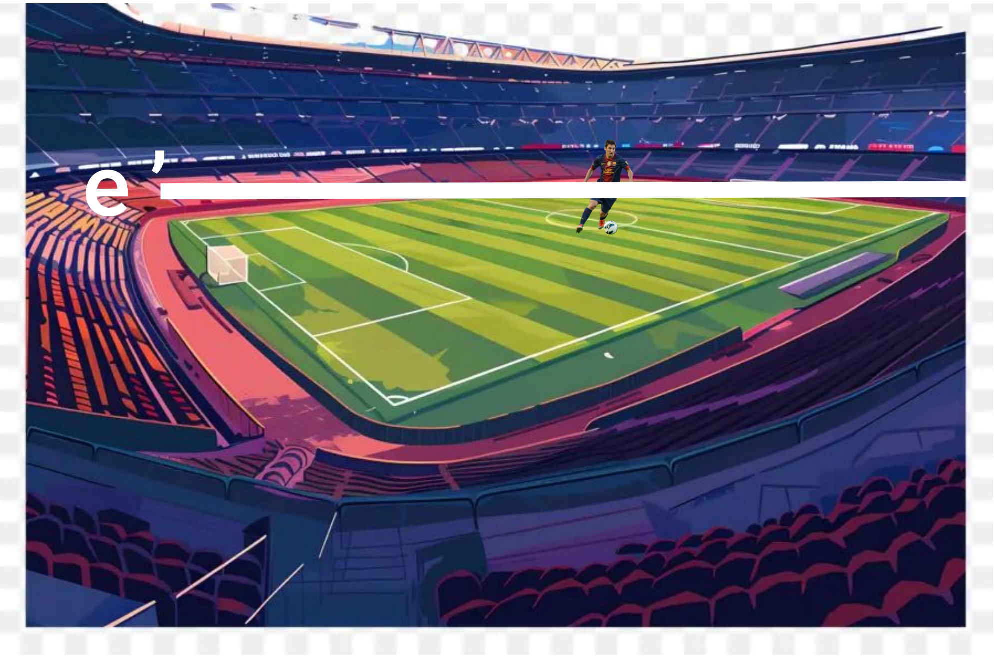
Epipolar constraint



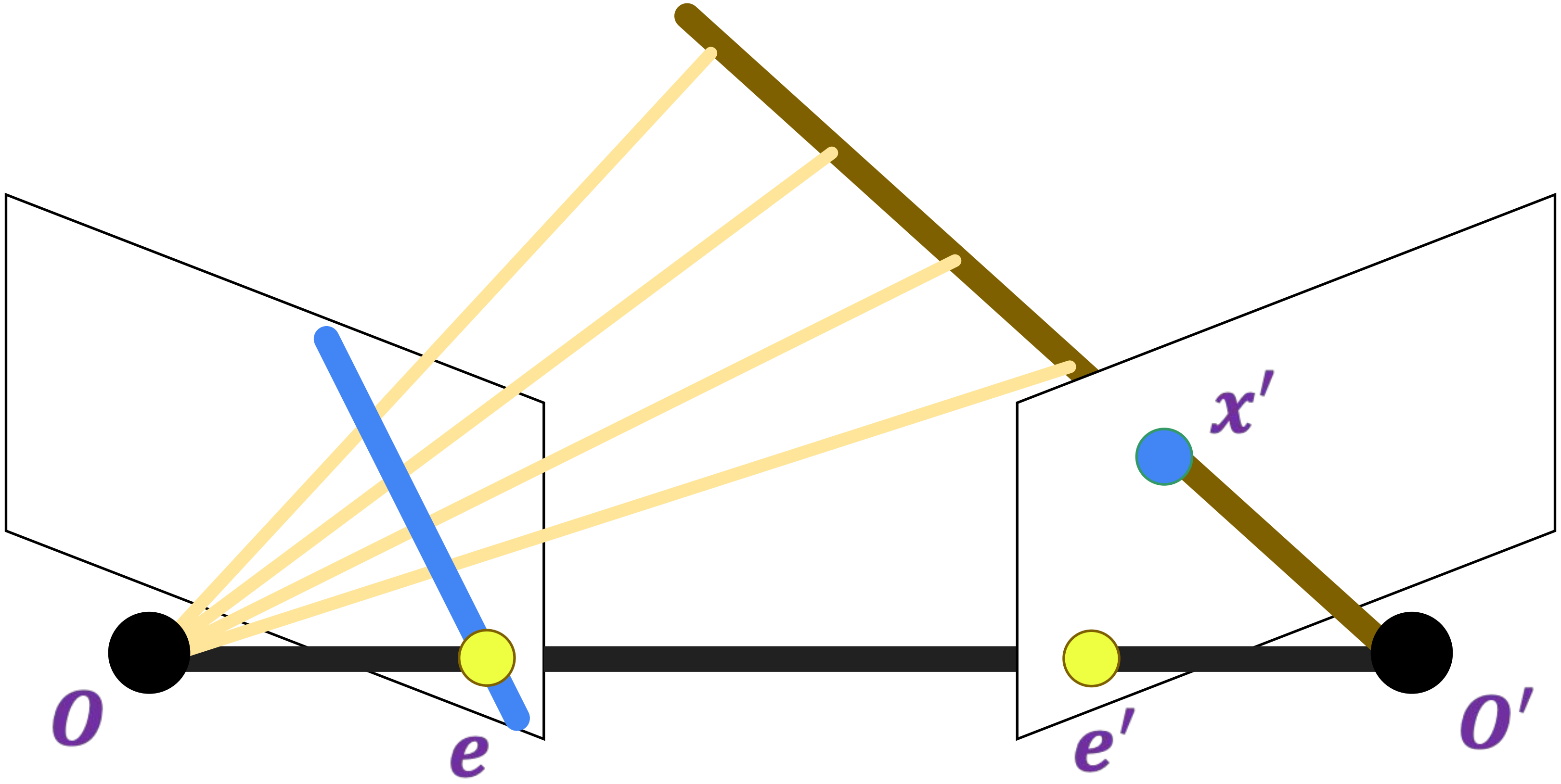
- Where can we find the x' corresponding to x in the other image?
- Along the **epipolar line** corresponding to x (projection of visual ray connecting O with x into the second image plane)

Epipolar geometry setup



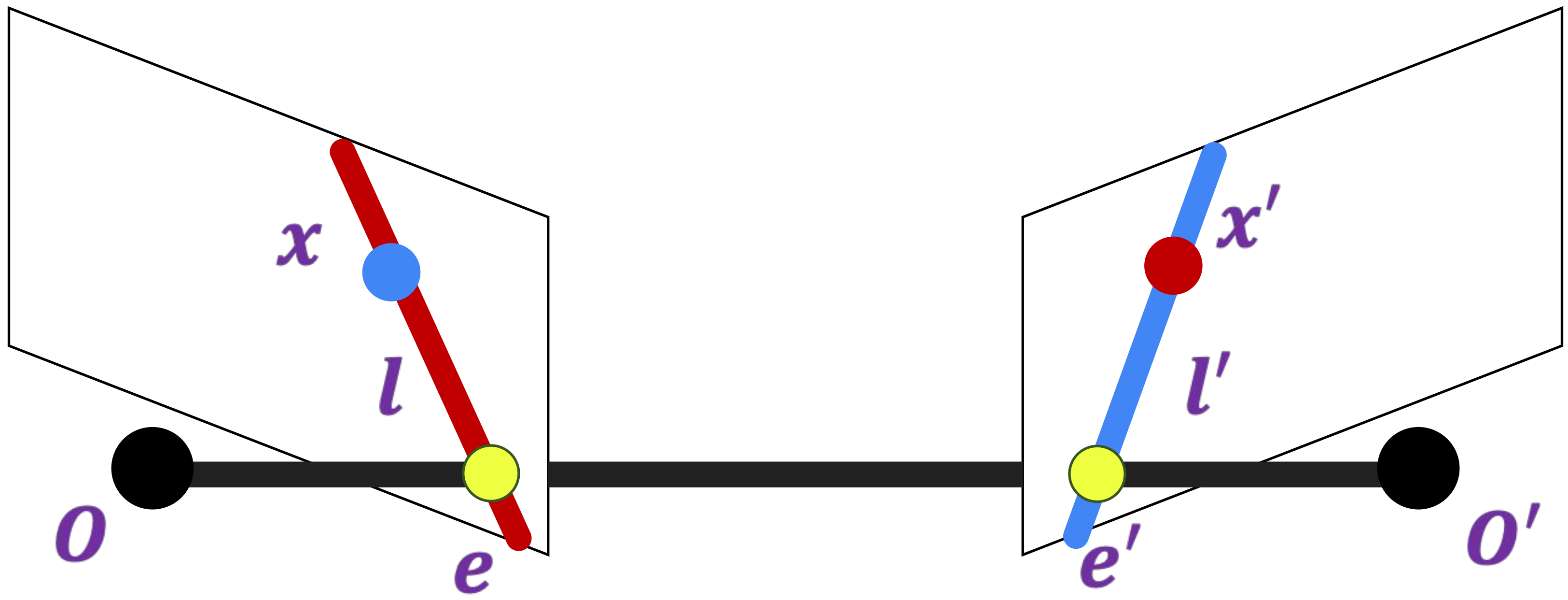


Epipolar constraint



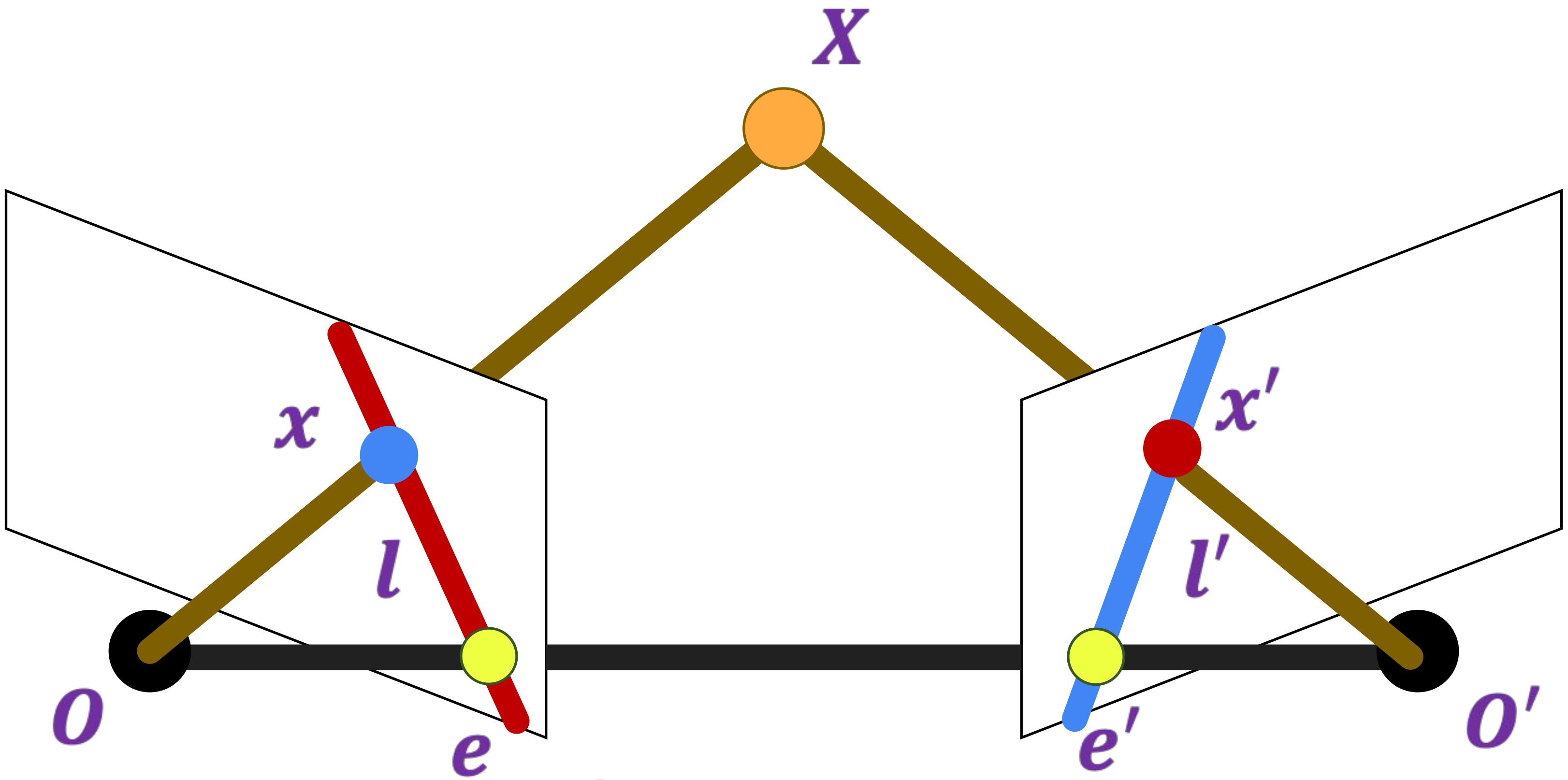
- Similarly, all points in the left image corresponding to x' have to lie along the epipolar line corresponding to x'

Epipolar constraint



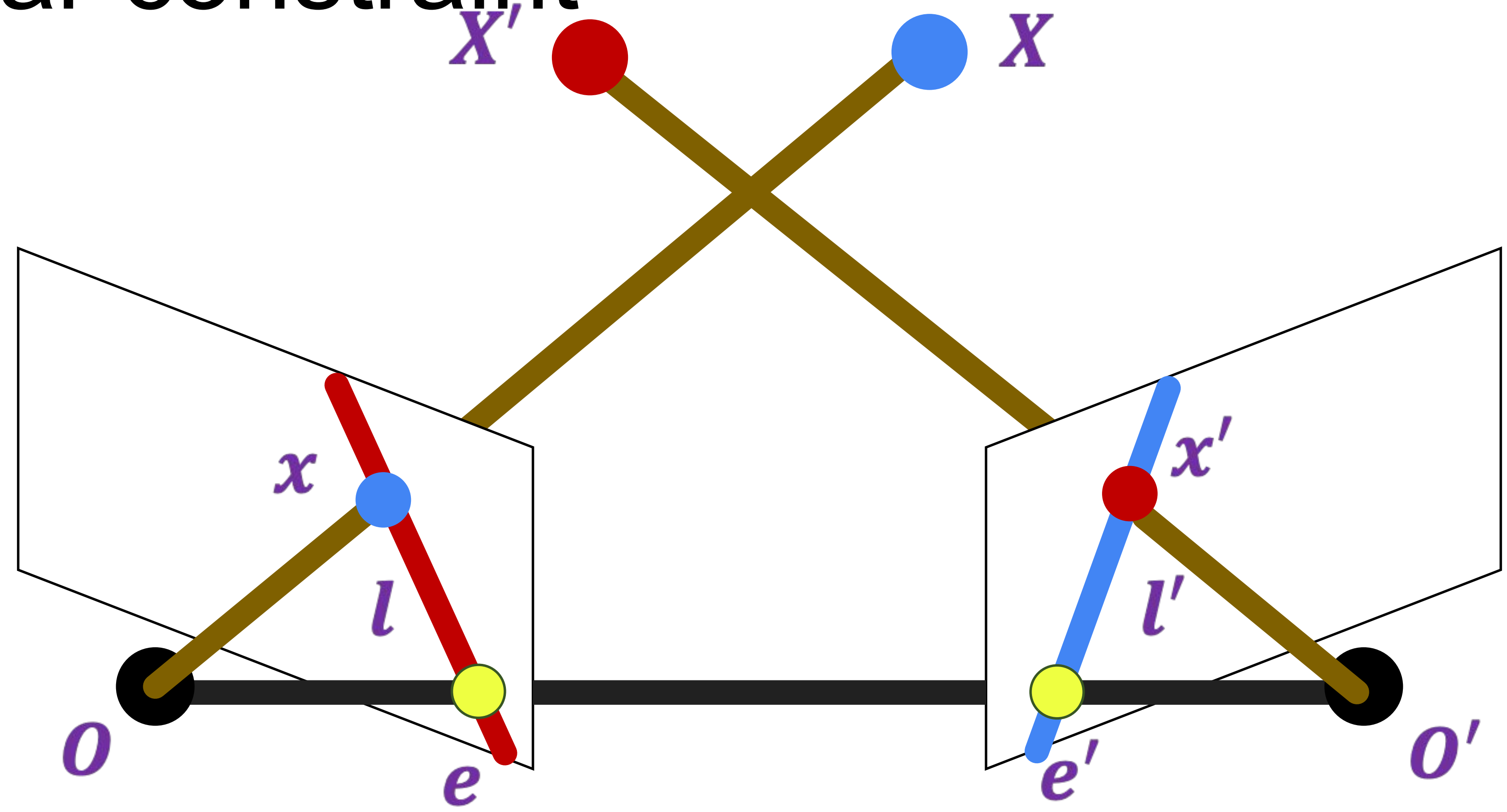
- Potential matches for x have to lie on the matching epipolar line l' and vice-versa → need only to search along 1D epipolar line for matching!

Epipolar constraint



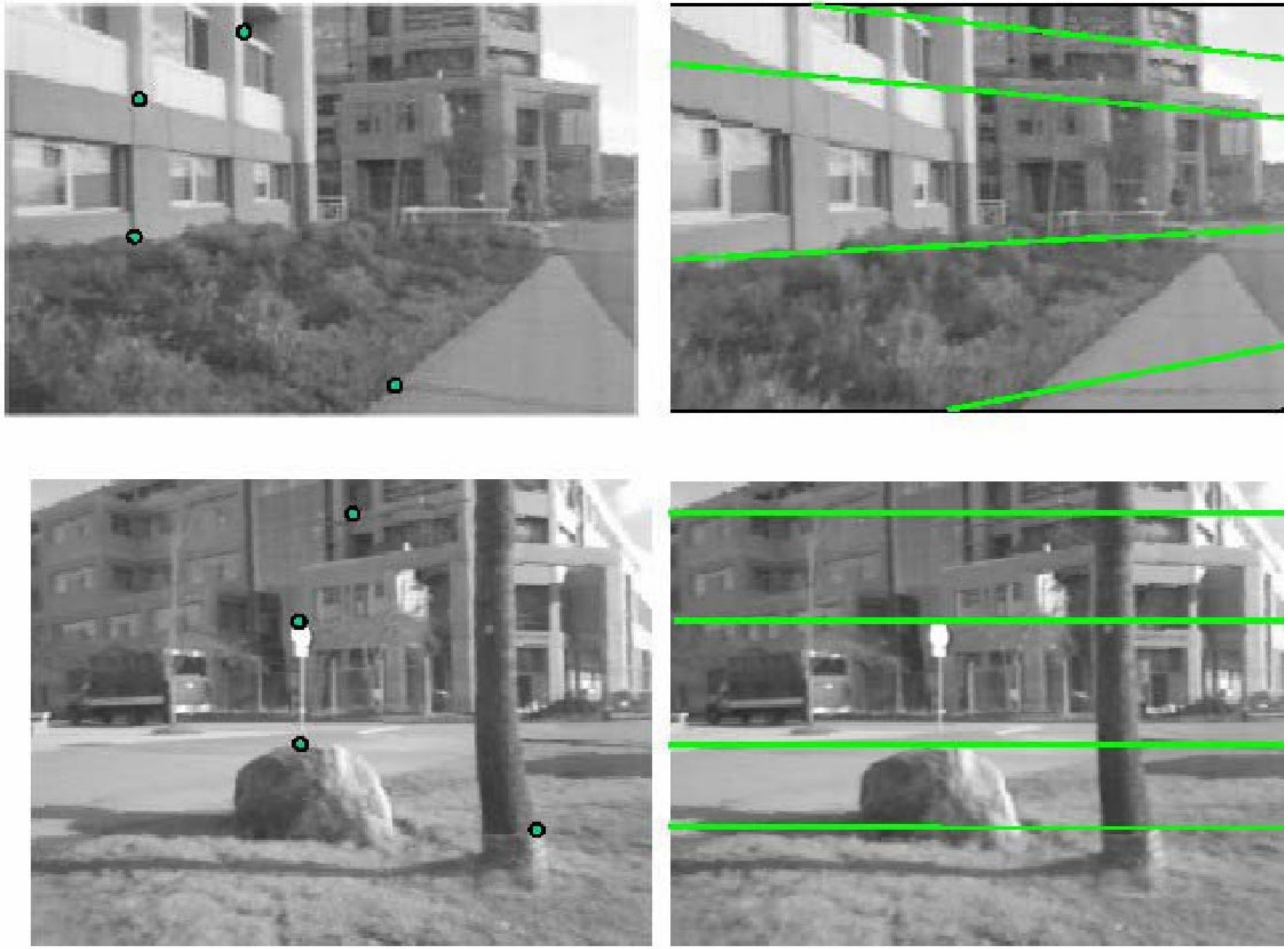
- Whenever two points x and x' lie on matching epipolar lines l and l', the ***visual rays*** corresponding to them meet in space, i.e., x and x' ***could be*** projections of the same 3D point X

Epipolar constraint

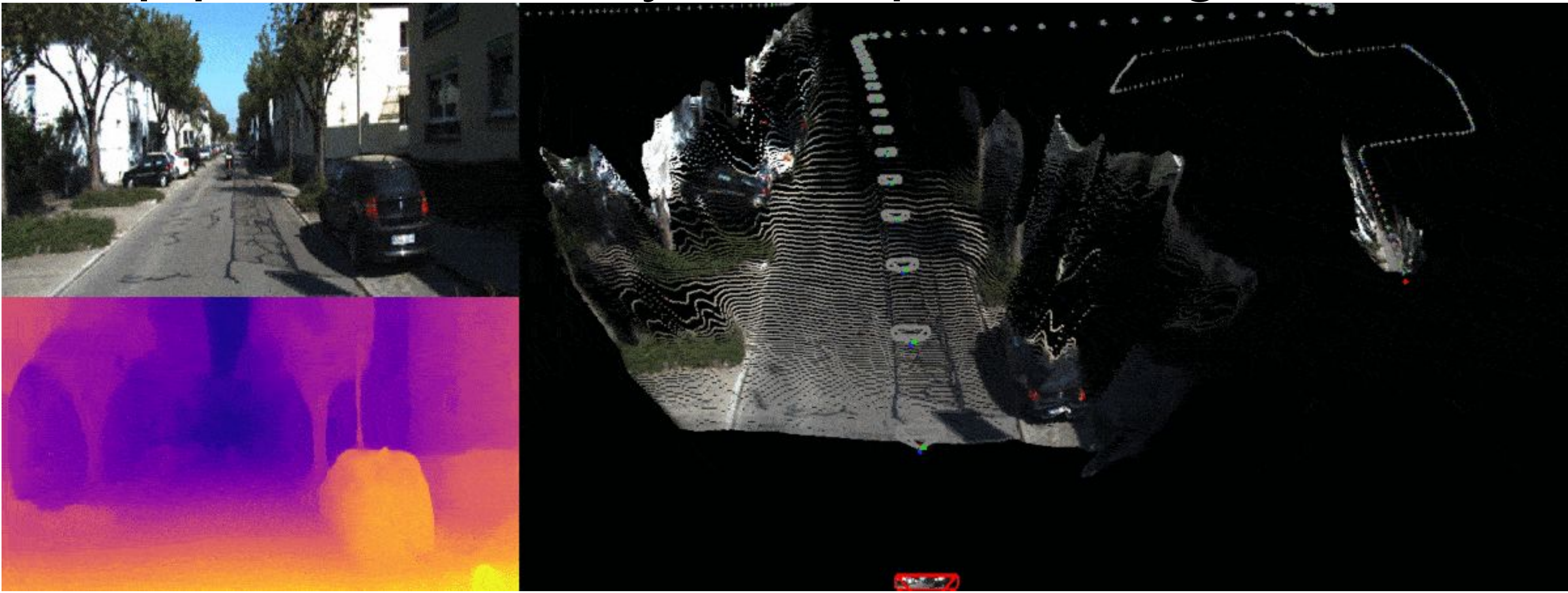


- Caveat: if x and x' satisfy the epipolar constraint, this doesn't mean they *have to be* projections of the same 3D point

Epipolar constraint: Example



Epipolar Geometry & Deep Learning



<u>Multi-Frame Self-Supervised Depth Estimation with Transformers</u> (CVPR 2022)

Vitor Guizilini, Rares Ambrus, Dian Chen, Sergey Zakharov, Adrien Gaidon

Epipolar Geometry & Deep Learning

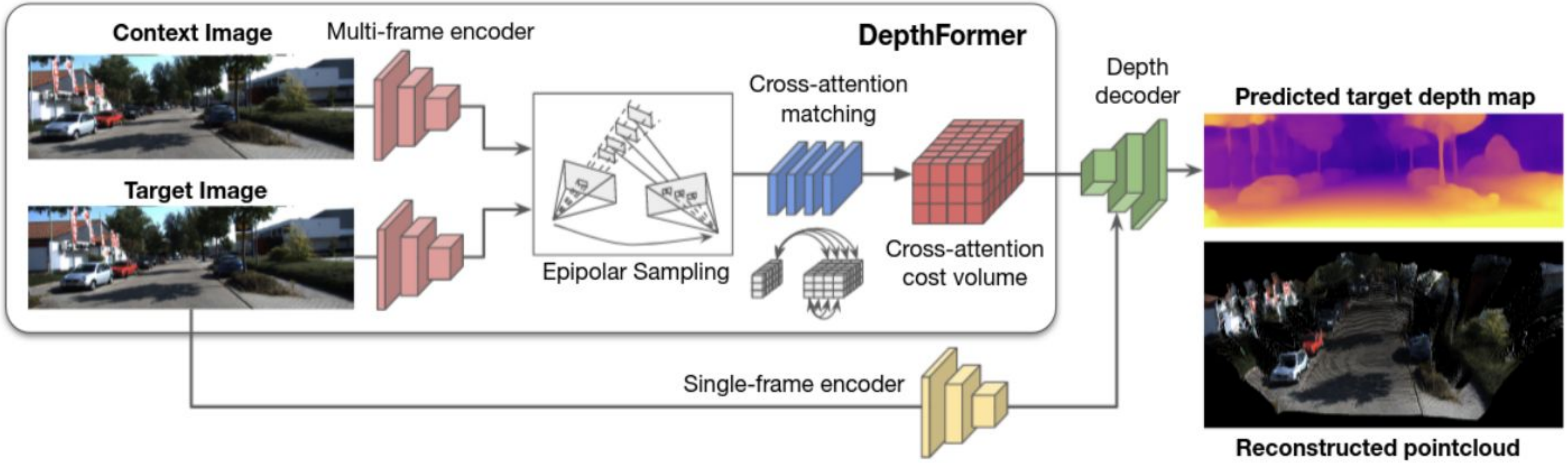
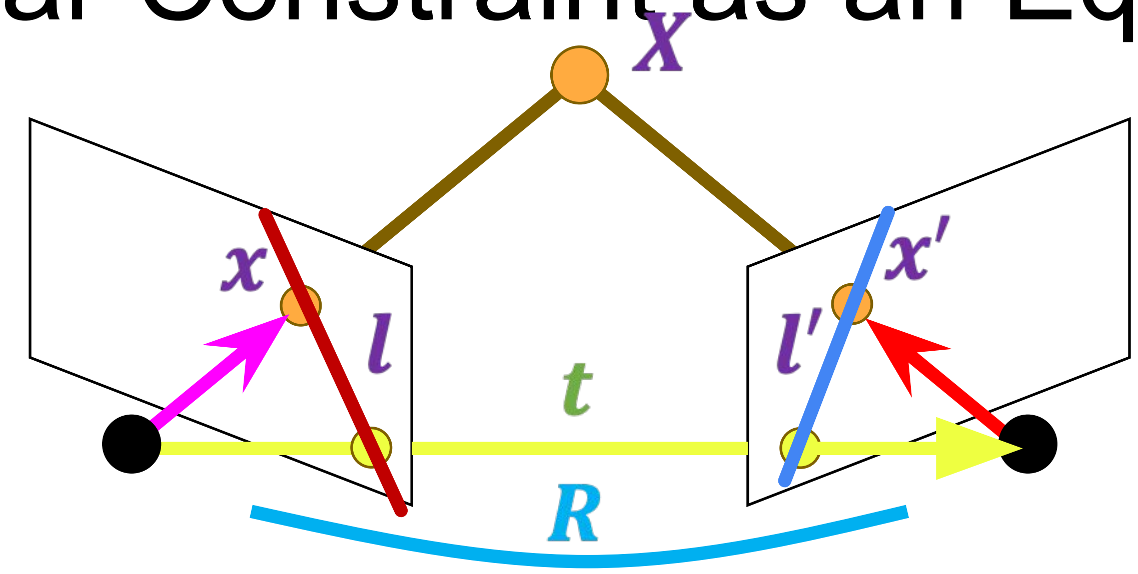


Figure 1. **Our DepthFormer architecture** achieves state-of-the-art multi-frame self-supervised monocular depth estimation by **improving feature matching** across images during cost volume generation.

Multi-Frame Self-Supervised Depth Estimation with Transformers (CVPR 2022)

Vitor Guizilini, Rares Ambrus, Dian Chen, Sergey Zakharov, Adrien Gaidon

The Epipolar Constraint as an Equation



$\boldsymbol{x}'^{T}\boldsymbol{F}\boldsymbol{x} = 0$ where $\boldsymbol{F} = \boldsymbol{K}'^{-T}\boldsymbol{E}\boldsymbol{K}^{-1}$ is called the ✨**Fundamental Matrix**✨ [1]

and $\boldsymbol{E} = [\boldsymbol{t}]_{\times}\boldsymbol{R}$ is the **Essential Matrix** [2]

$$(x', y', 1)\begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix}\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = 0$$

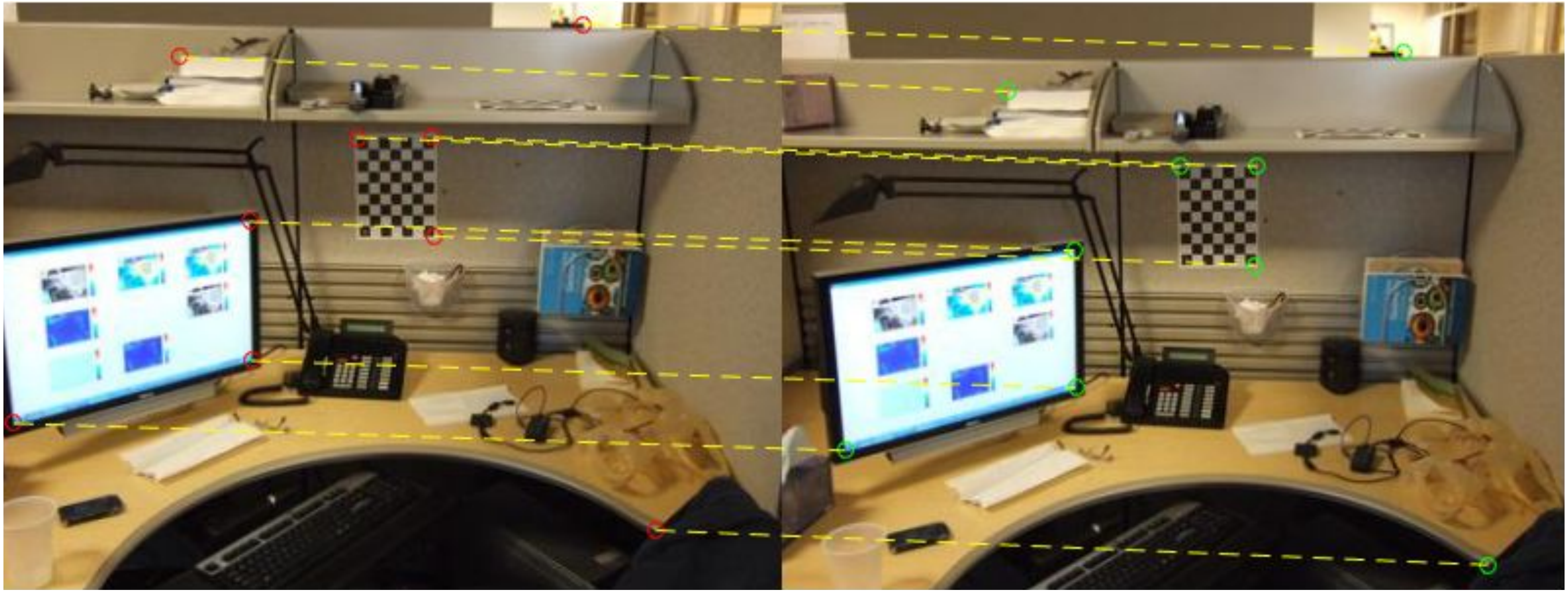
(sketch of proof in appendix)

[1] Faugeras et al., (1992), Hartley (1992)

[2] H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. Nature, 1981

Estimating the fundamental matrix - teaser

- Given: correspondences $\boldsymbol{x} = (x, y, 1)^T$ and $\boldsymbol{x}' = (x', y', 1)^T$



Estimating the fundamental matrix - teaser

- Given: 2D correspondences $\boldsymbol{x} = (x, y, 1)^T$ and $\boldsymbol{x}' = (x', y', 1)^T$
- Constraints: $\boldsymbol{x}'^T \boldsymbol{F} \boldsymbol{x} = 0$ (1 per correspondence, how many needed?)
- Boils down to another homogeneous linear equation $\boldsymbol{AX} = \boldsymbol{0}$
- Recast once more into total least squares (Sz.A.2.1) due to noise
- SVD gives the solution as usual + enables enforcing rank 2 constraint by replacing smallest singular value with 0
- This "algebraic" algorithm is called "[normalized] 8-point algorithm" (R. Hartley. In defense of the eight-point algorithm. TPAMI 1997)
- As in calibration and homography fitting: non-linear "geometric" optimization (of reprojected distances) is more precise
- Can be made robust to outliers via the RANSAC algorithm
- See appendix, H&Z ch. 9, Szeliski 11.3, or take ***CS231A*** for more!

From epipolar geometry to camera calibration

Estimating the fundamental matrix is known as “weak calibration”

If we know the calibration matrices (K, K') of the two cameras, we can estimate the ***essential matrix***: $\boldsymbol{E} = \boldsymbol{K}'^{T}\boldsymbol{F}\boldsymbol{K}$

The essential matrix gives us the relative rotation and translation between the cameras, or their ***extrinsic parameters*** ($\mathbf{E} = [\boldsymbol{t}]_{\times}\boldsymbol{R}$)

Alternatively, if the calibration matrices are known (or in practice, if good initial guesses of the intrinsics are available), the five-point algorithm can be used to estimate relative camera pose

D. Nister. An efficient solution to the five-point relative pose problem. IEEE Trans. PAMI, 2004

What will we learn today?

Triangulation

Epipolar geometry

Stereo

Structure-from-Motion (SfM)

Example configuration: Converging cameras

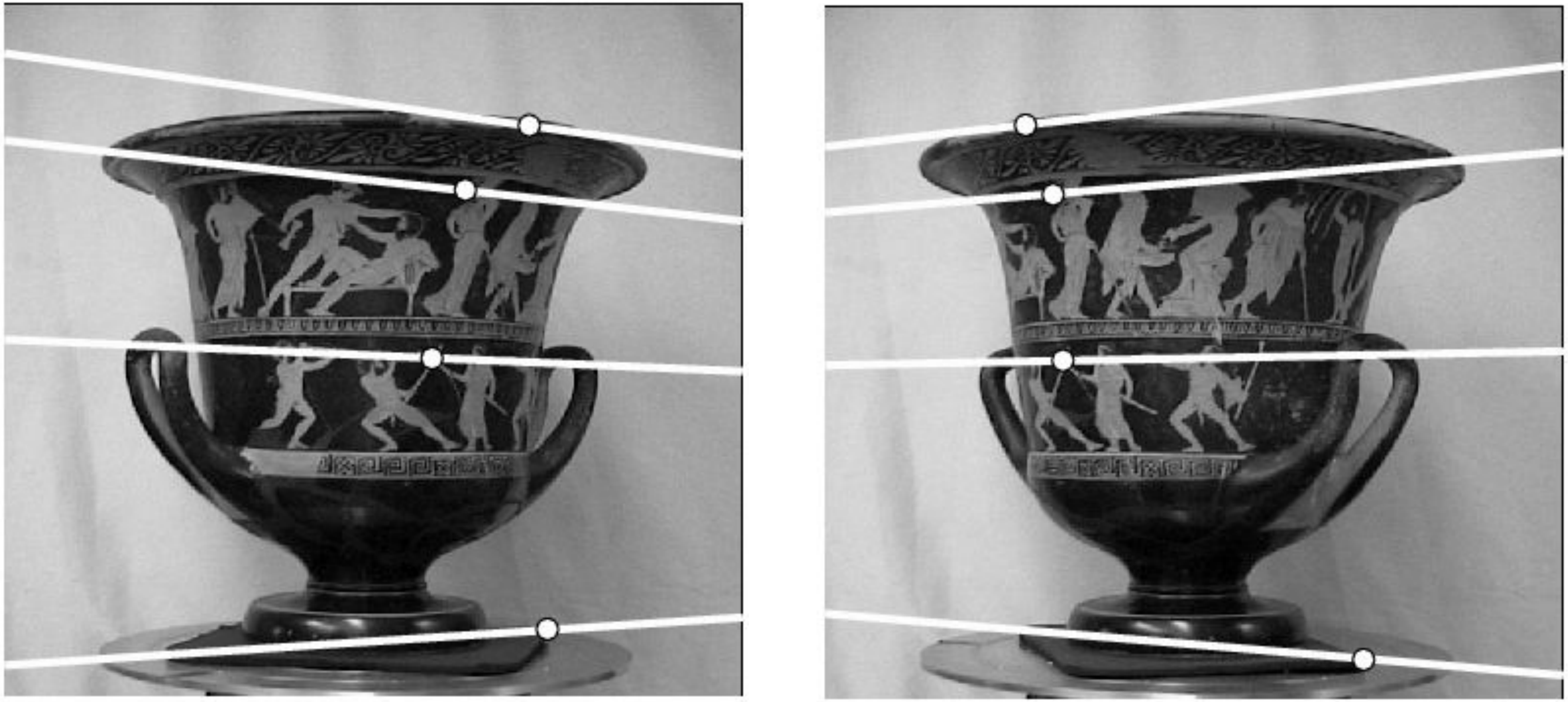
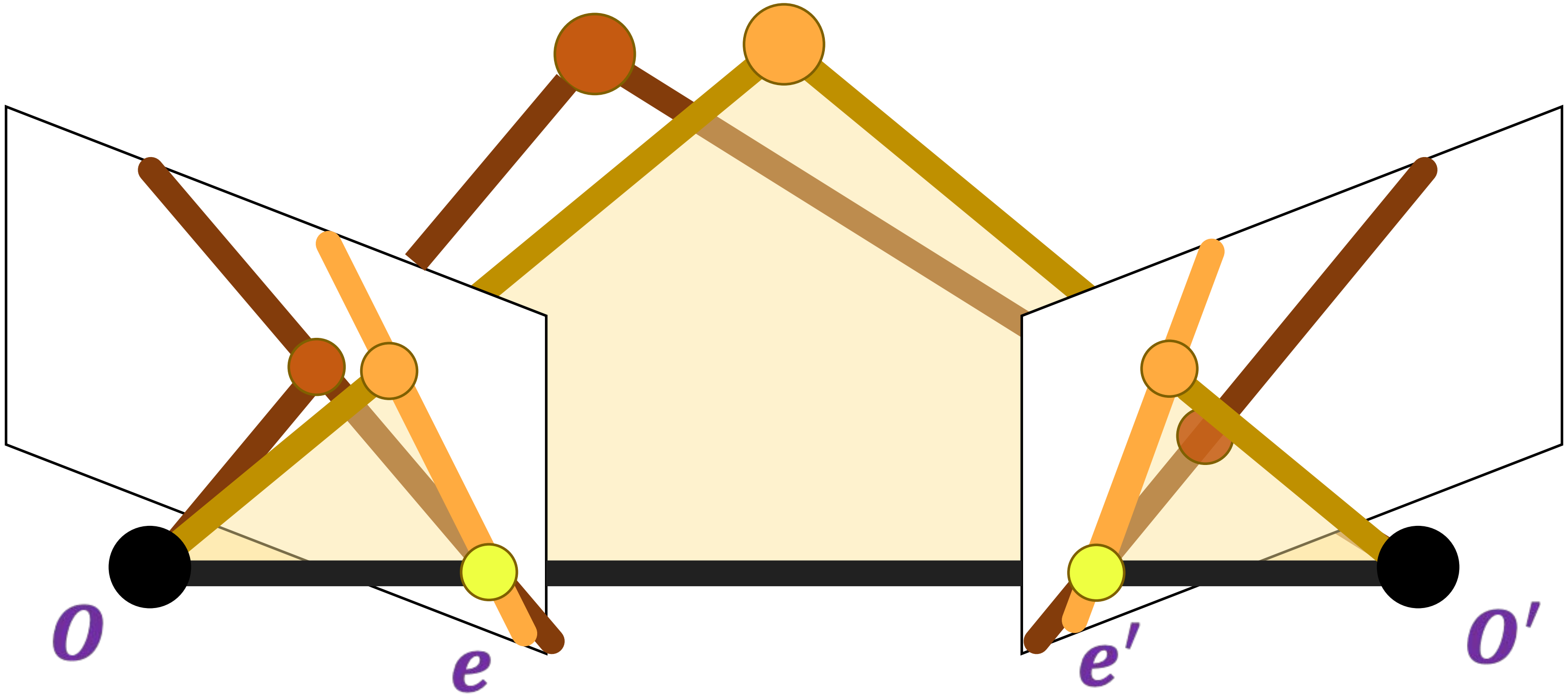


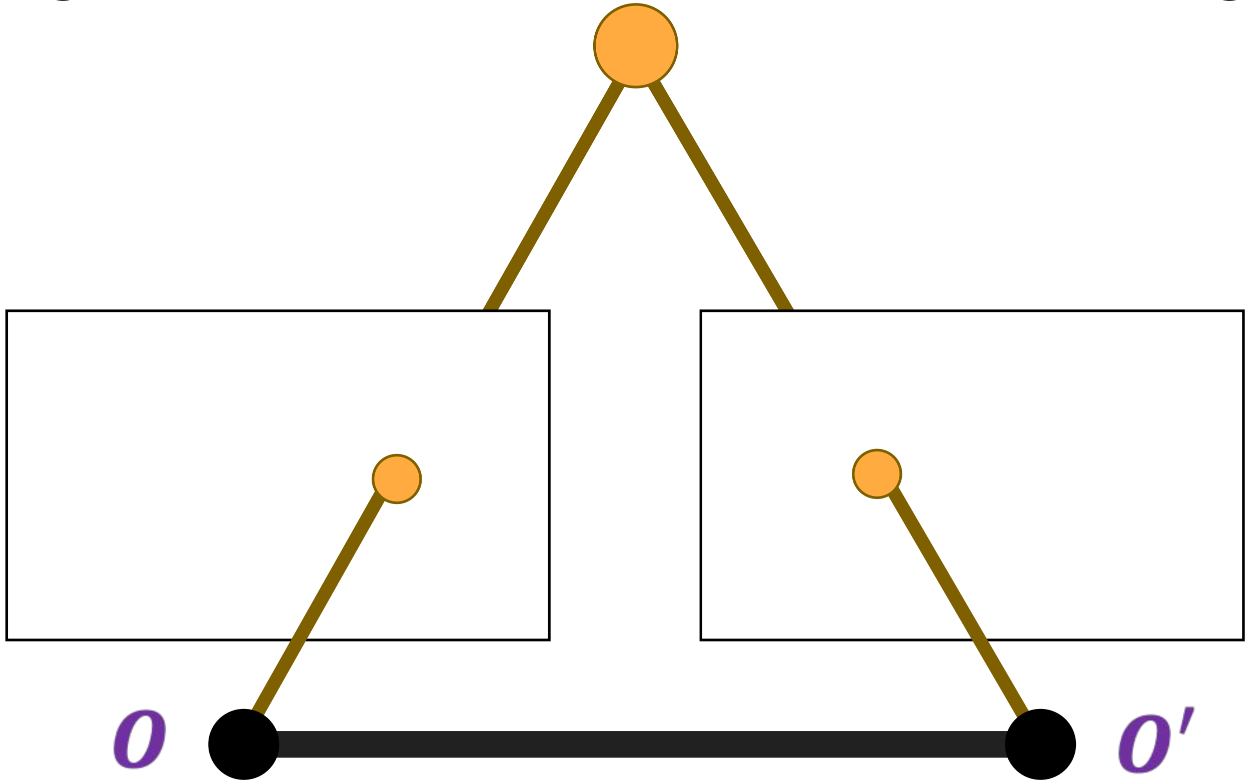
Image Credit: Hartley & Zisserman

Example configuration: Converging cameras



- Epipoles are finite, may be visible in the image

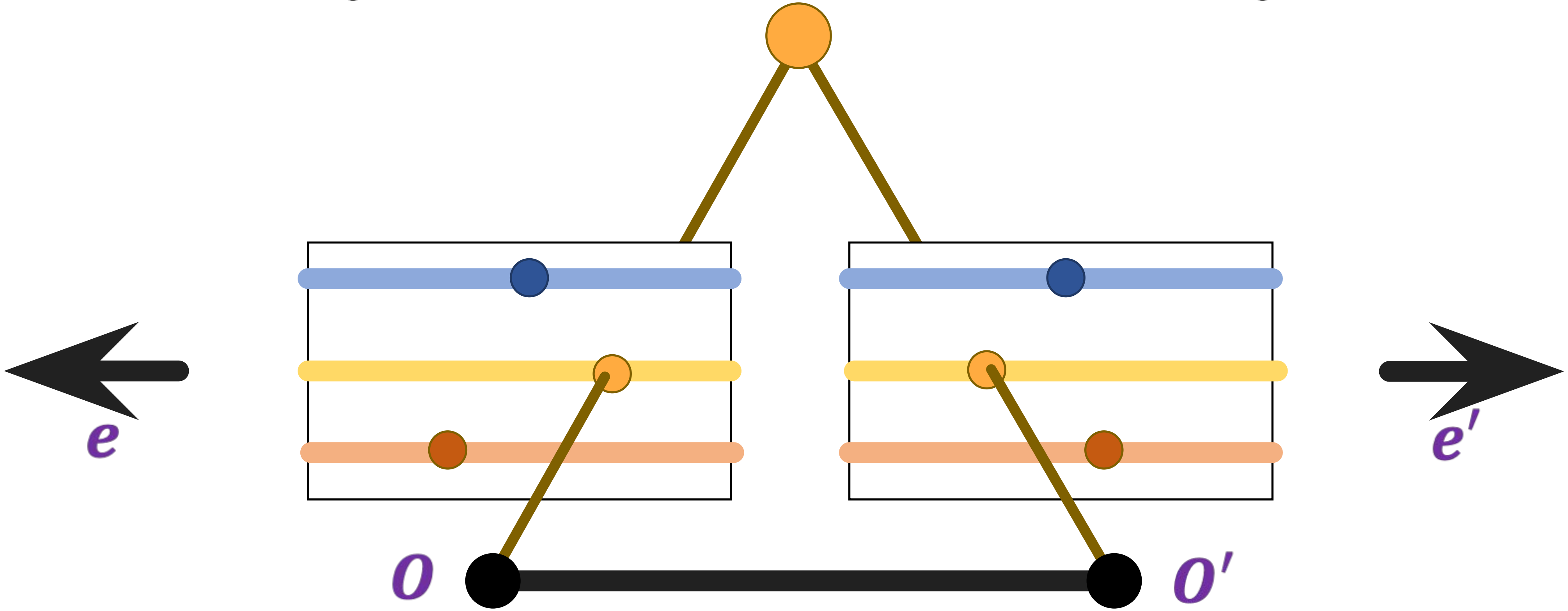
Example configuration: Motion parallel to image plane



Where are the epipoles?

What do the epipolar lines look like?

Example configuration: Motion parallel to image plane

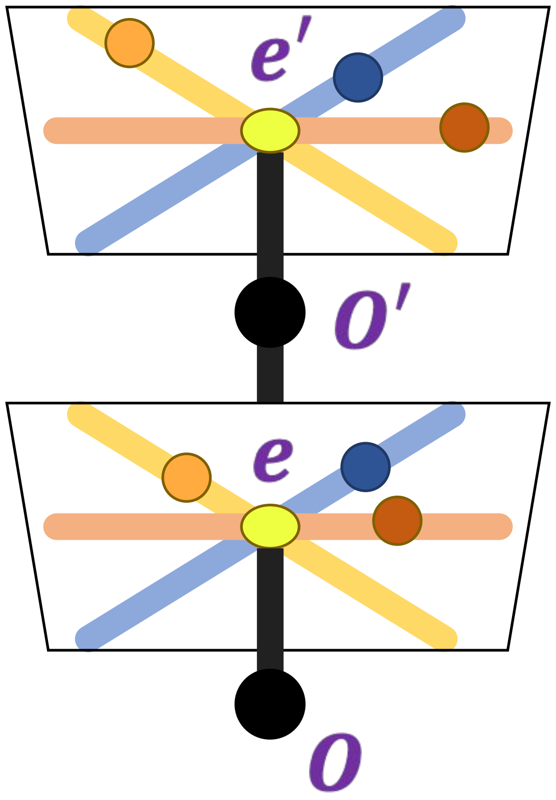


Epipoles *infinitely* far away!

Epipolar lines parallel: "scan lines"

Stereo = easier fronto-parallel special case!

Example configuration: Motion perpendicular to image plane

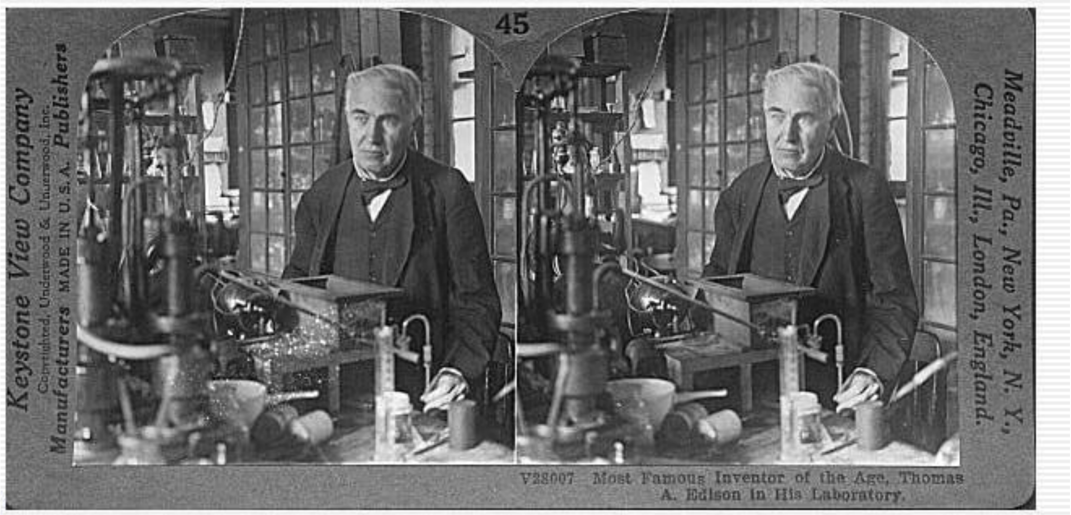


- Epipole is the “focus of expansion” and coincides with the principal point of the camera
- Epipolar lines go out from principal point

History: Stereograms

Humans can fuse pairs of images to get a sensation of depth

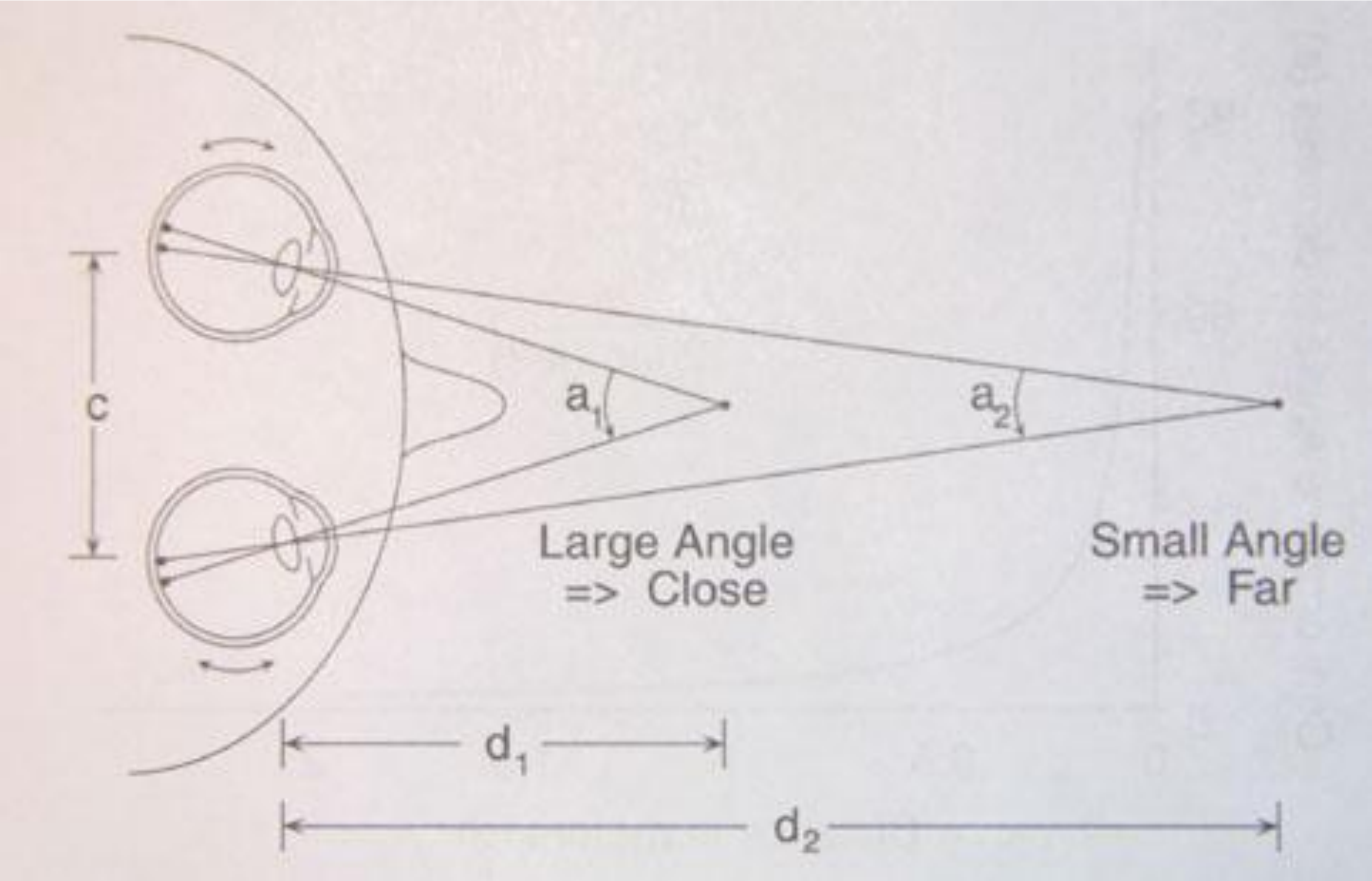
Vérascope 40



Stereograms: Invented by Sir Charles Wheatstone, 1838

https://en.wikipedia.org/wiki/Stereoscopy

Depth from convergence



$$d = \frac{c}{2tan(a/2)}$$

Slide credit: J. Hayes

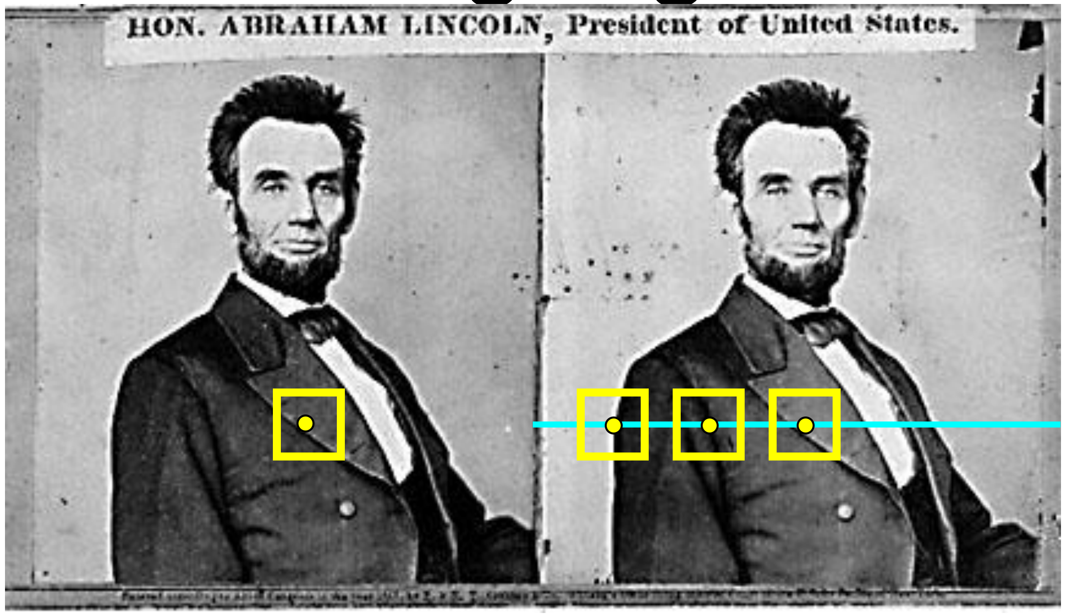
Stereo Matching for Depth Estimation

Given: stereo pair (assumed calibrated)

Wanted: dense depth map



Basic stereo matching algorithm

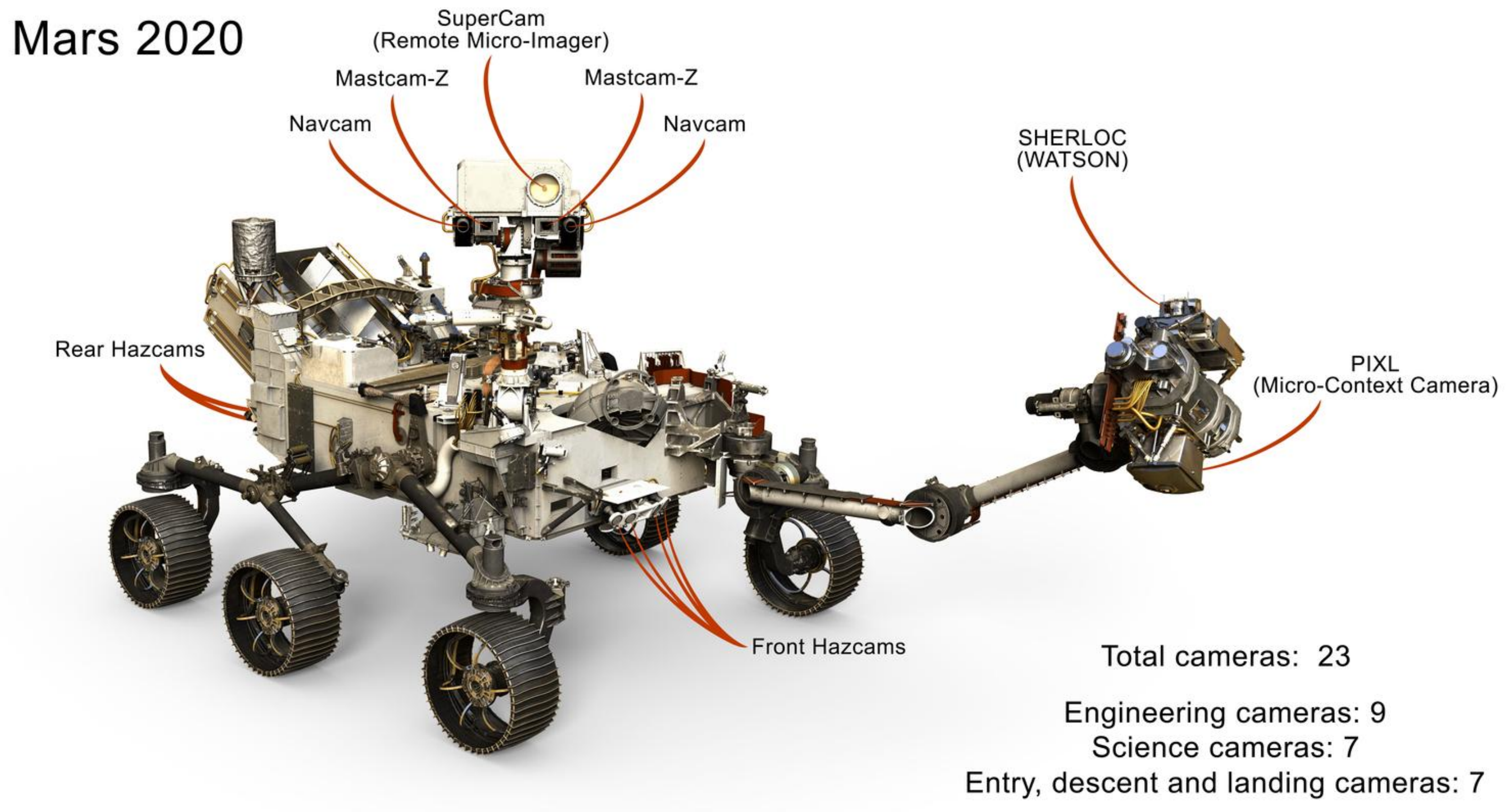


For each pixel in the first image

- Find corresponding epipolar line in the right image: same row!
- Examine all pixels on the epipolar line and pick the best match
- Triangulate the matches to get depth information

More details in appendix: rectification, matching, depth from disparity, etc

Stereo on the Perseverance Mars Rover



https://mars.nasa.gov/mars2020/spacecraft/rover/cameras/

What will we learn today?

Triangulation

Epipolar geometry

Stereo

Structure-from-Motion (SfM)

Reference: Szeliski 11, H&Z ch. 9

Most slides adapted from N. Snavely & S. Lazebnik

Structure-from-Motion

Given many images, how can we

a) figure out where they were all taken from?

b) build a 3D model of the scene?

N. Snavely, S. Seitz, and R. Szeliski, Photo tourism: Exploring photo collections in 3D, SIGGRAPH 2006.

http://phototour.cs.washington.edu/

Geometry of more than two views?

2 views: governed by the 3x3 **Fundamental Matrix** (how to go from one point in an image to the epipolar line in the 2nd image)

3 views: governed by the 3 x 3 x 3 **Trifocal Tensor**

4 views: governed by the 3 x 3 x 3 x 3 **Quadrifocal Tensor**

After this it starts to get complicated…

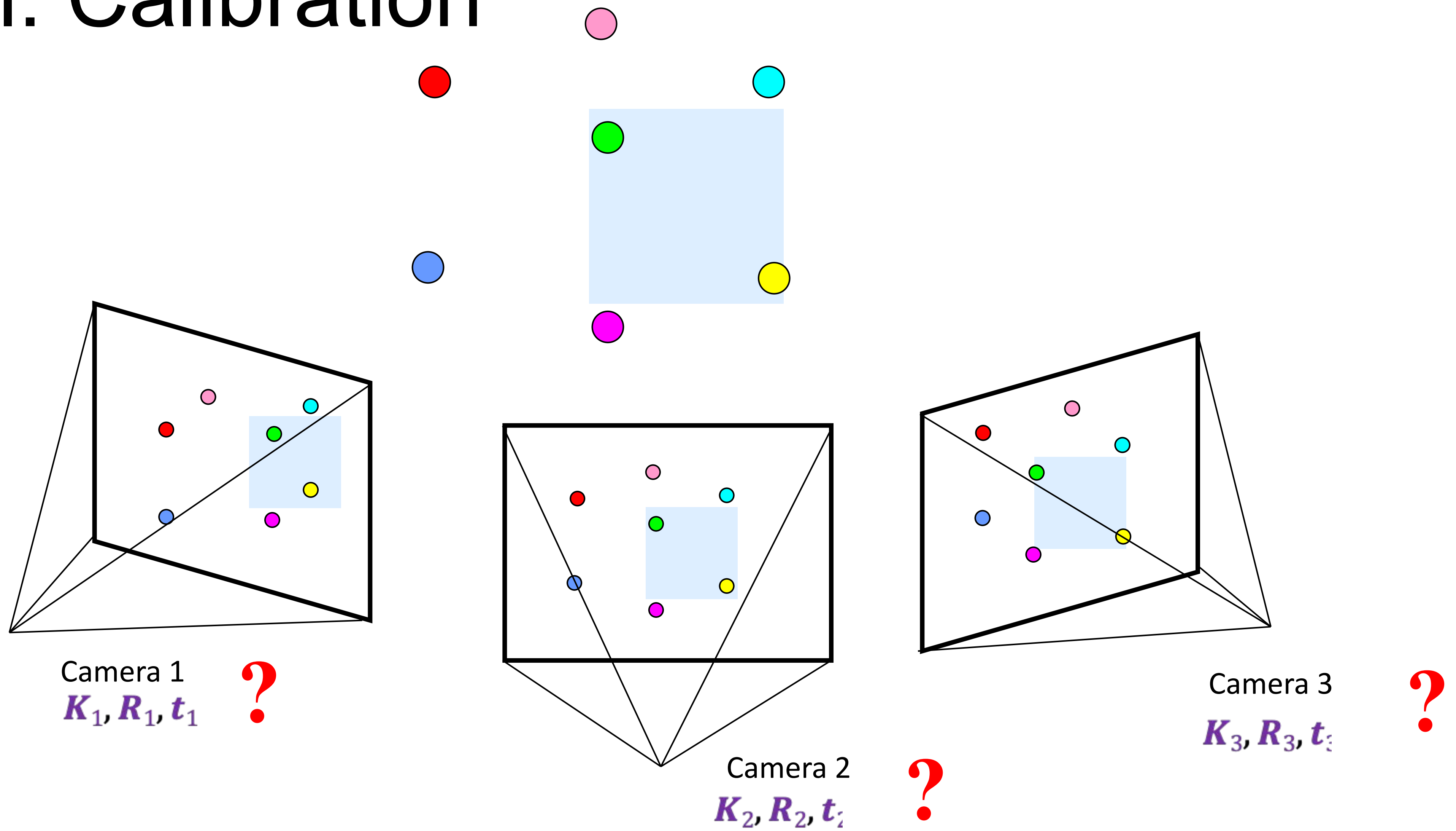
explicitly solve for camera poses ***and*** scene geometry

Large-scale structure-from-motion

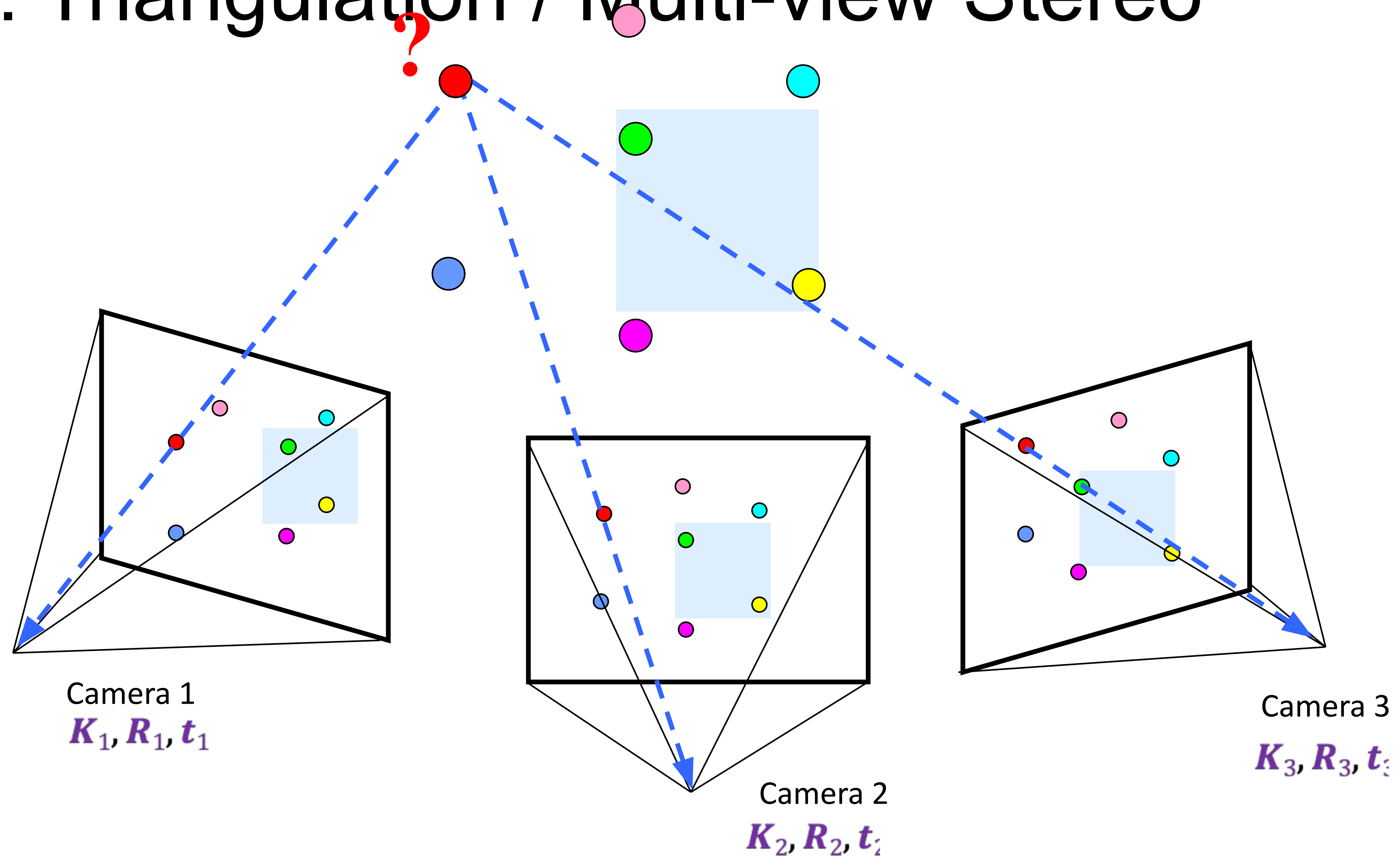
Dubrovnik, Croatia. 4,619 images (out of an initial 57,845 downloaded from Flickr). 3.5M points!
Total reconstruction time: 17.5 hours on 352 cores

Building Rome in a Day, *Agarwal et al, ICCV'09*
http://grail.cs.washington.edu/rome/

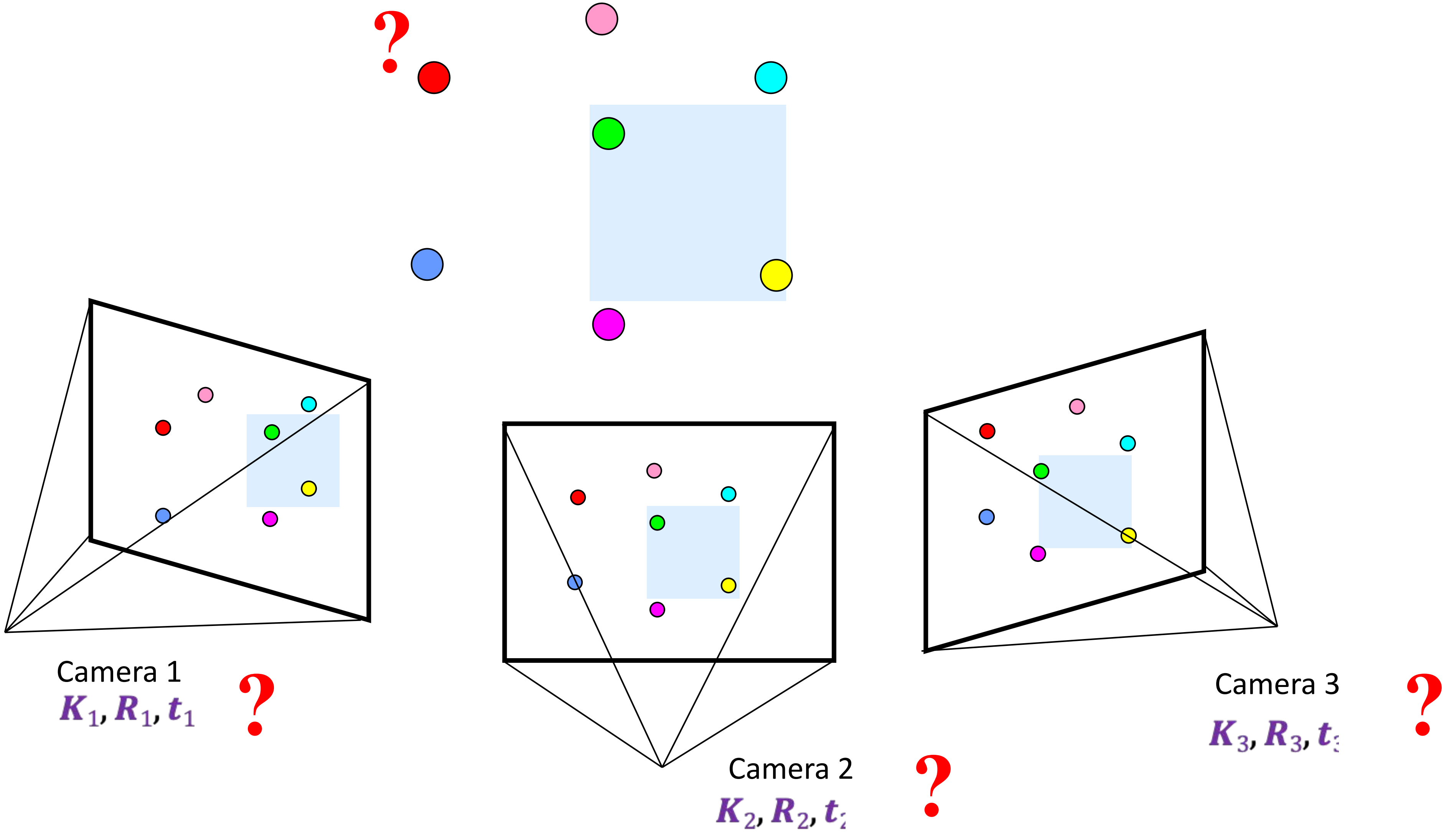
Recall: Calibration



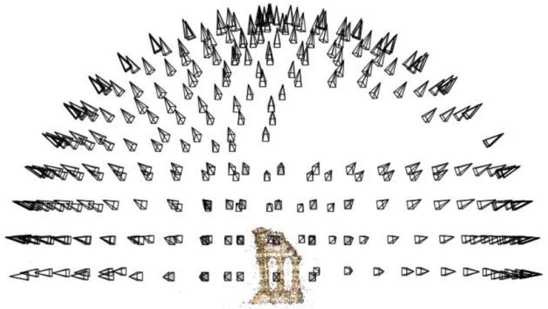
Recall: Triangulation / Multi-view Stereo



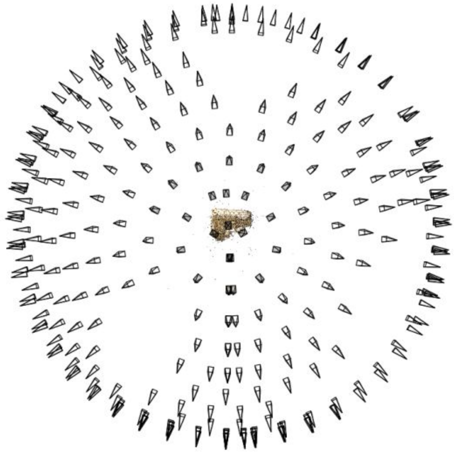
Structure-from-Motion



Structure-from-Motion



Reconstruction (side)



(top)

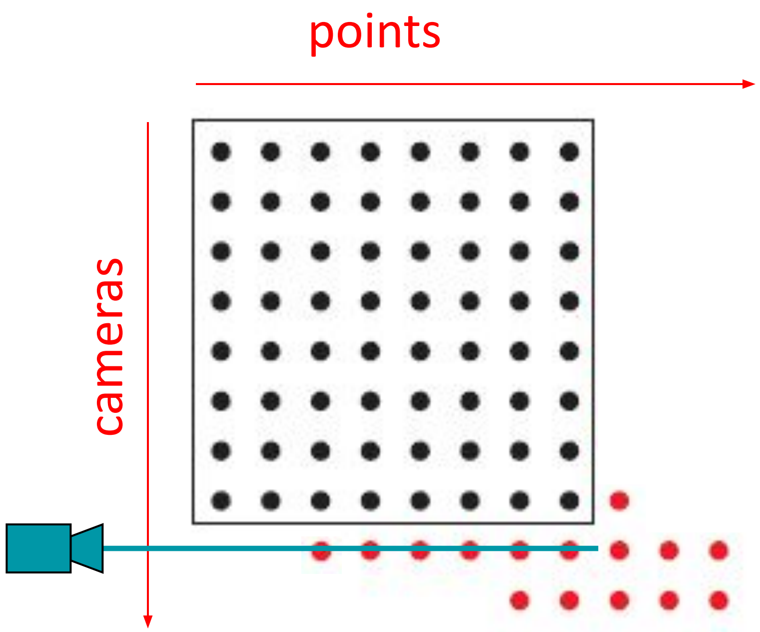
- Input: images with 2D points $\boldsymbol{x}_{ij}$ in correspondence

- Output (solved simultaneously now!)
  - structure: 3D location $\boldsymbol{X}_j$ for each point $\boldsymbol{x}_{ij}$
  - motion: camera parameters $\boldsymbol{R}_i$, $\boldsymbol{t}_i$ & possibly $\boldsymbol{K}_i$

- Objective function: minimize *reprojection error in 2D*

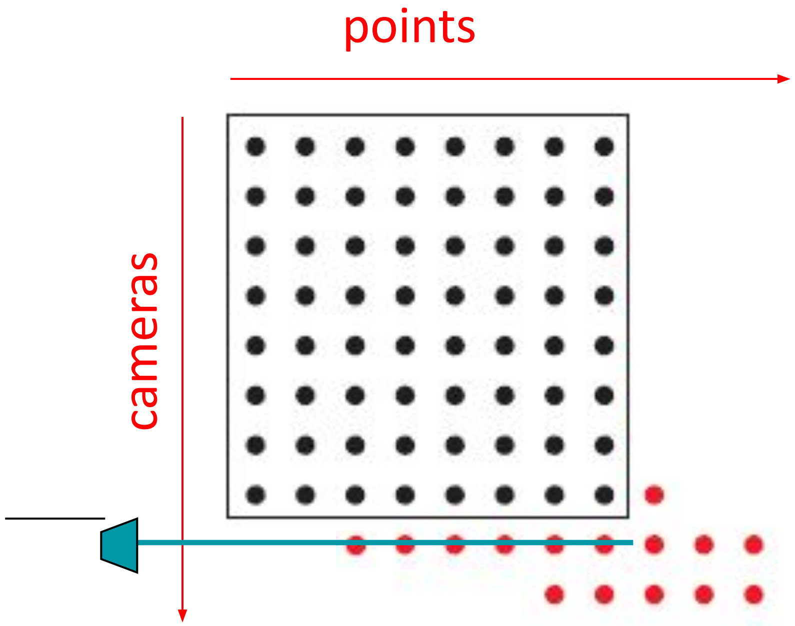
Incremental Structure-from-Motion

- Initialize motion from two images using the fundamental matrix
- Initialize structure by triangulation
- For each additional view:
  - Determine projection matrix of new camera using all the known 3D points that are visible in its image – calibration



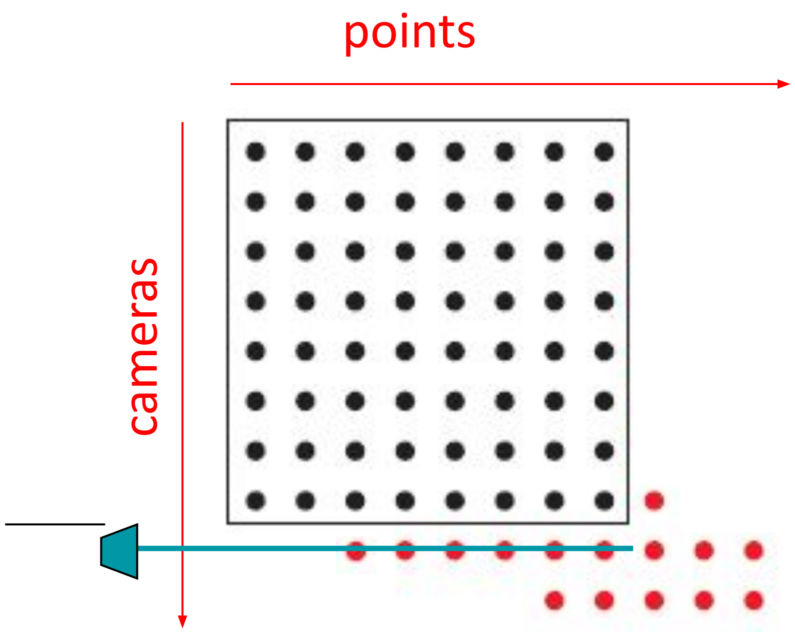
Incremental Structure-from-Motion

- Initialize motion from two images using the fundamental matrix
- Initialize structure by triangulation
- For each additional view:
  - Determine projection matrix of new camera using all the known 3D points that are visible in its image – calibration
  - Refine and extend structure: compute newly visible 3D points, re-optimize existing points that are also seen by this camera – triangulation



Incremental Structure-from-Motion

- Initialize motion from two images using the fundamental matrix
- Initialize structure by triangulation
- For each additional view:
  - Determine projection matrix of new camera using all the known 3D points that are visible in its image – calibration
  - Refine and extend structure: compute newly visible 3D points, re-optimize existing points that are also seen by this camera – triangulation
- Refine all cameras & points *jointly*: **bundle adjustment**



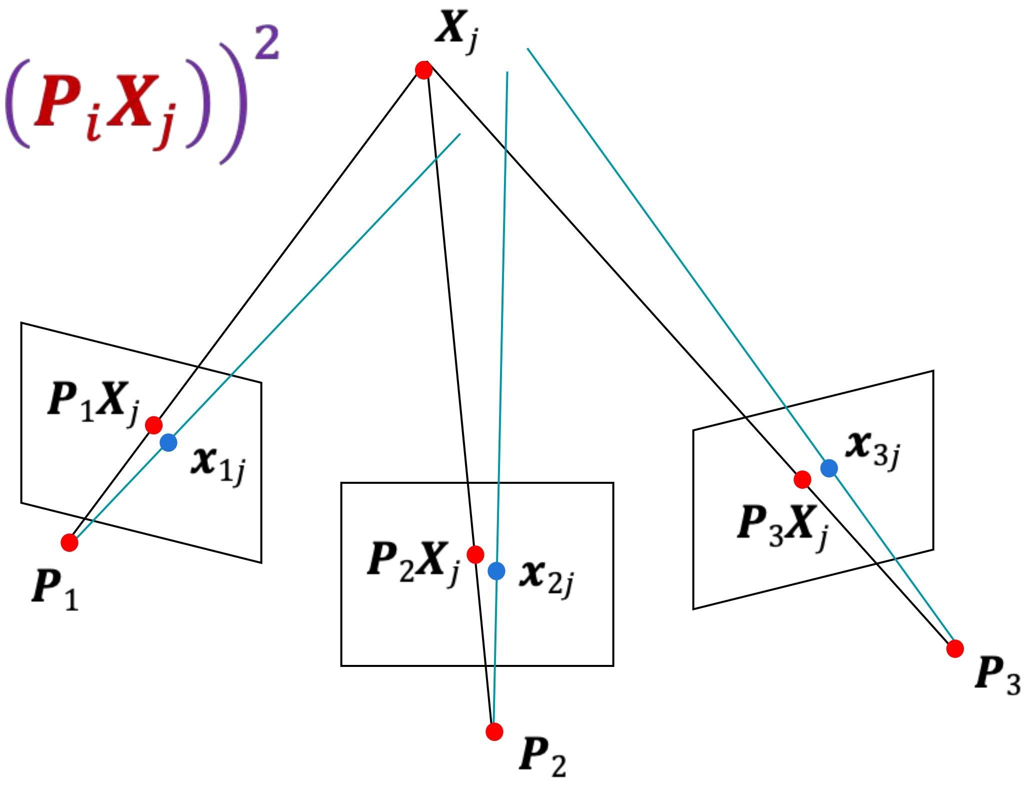
Bundle Adjustment

Non-linear method for refining structure ($\boldsymbol{X}_j$) and motion ($\boldsymbol{P}_i$)

Minimize reprojection error (with lots of bells and whistles):

$$\sum_{i=1}^{m}\sum_{j=1}^{n} w_{ij}\, d\left(\boldsymbol{x}_{ij} - \mathrm{proj}(\boldsymbol{P}_i\boldsymbol{X}_j)\right)^2$$

visibility flag: is point j visible in view i?



B. Triggs et al. Bundle adjustment – A modern synthesis. International Workshop on Vision Algorithms, 1999

Incremental SfM in Practice

- Pick a pair of images with lots of inliers (and good EXIF data)
  - Initialize intrinsic parameters (focal length, principal point) from EXIF
  - Estimate extrinsic parameters ($\boldsymbol{R}$ and $\boldsymbol{t}$) using five-point algorithm
  - Use triangulation to initialize model points
- While remaining images exist
  - Find an image with many feature matches with images in the model
  - Run RANSAC on feature matches to register new image to model
  - Triangulate new points
  - Perform bundle adjustment to re-optimize everything
  - Optionally, align with GPS from EXIF data or ground control points

Incremental structure from motion

Time-lapse reconstruction of Dubrovnik, Croatia, viewed from above

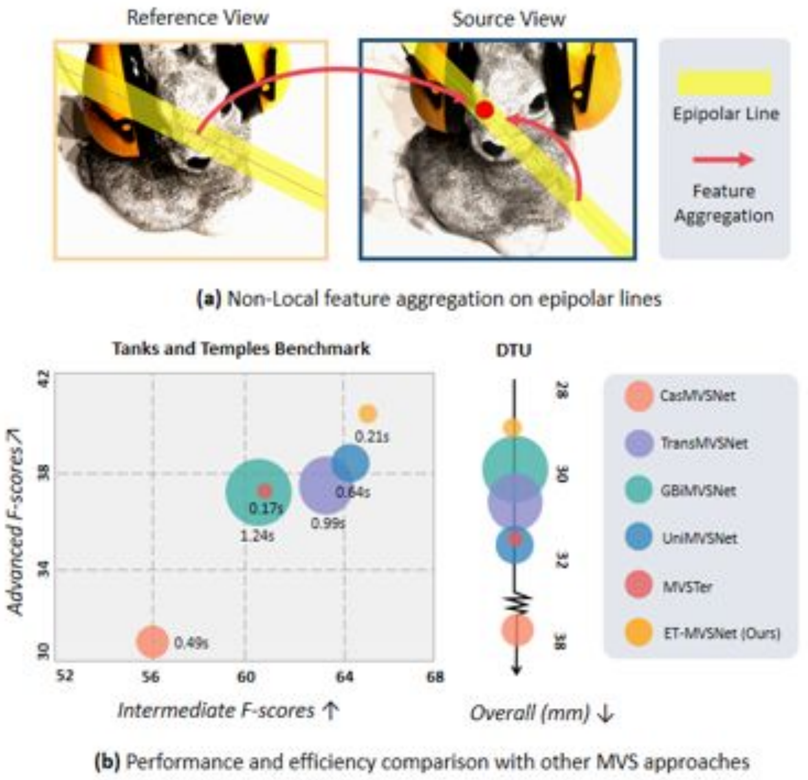
COLMAP

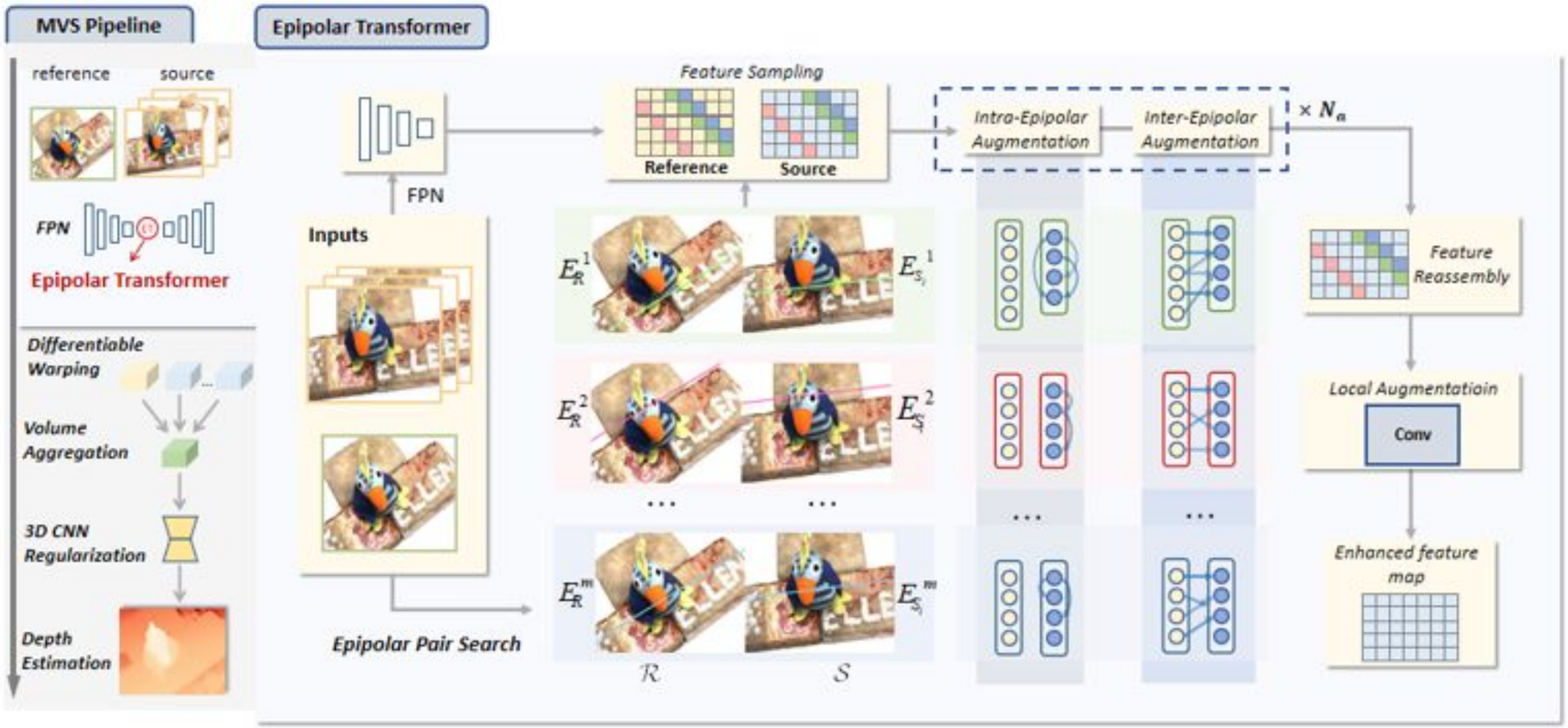
Sparse model of central Rome using 21K photos produced by COLMAP's SfM pipeline.

Dense models of several landmarks produced by COLMAP's MVS pipeline.

https://colmap.github.io/

SfM in the age of Deep Learning





ET-MVSNet: When Epipolar Constraint Meets Non-local Operators in Multi-View Stereo (ICCV'23)

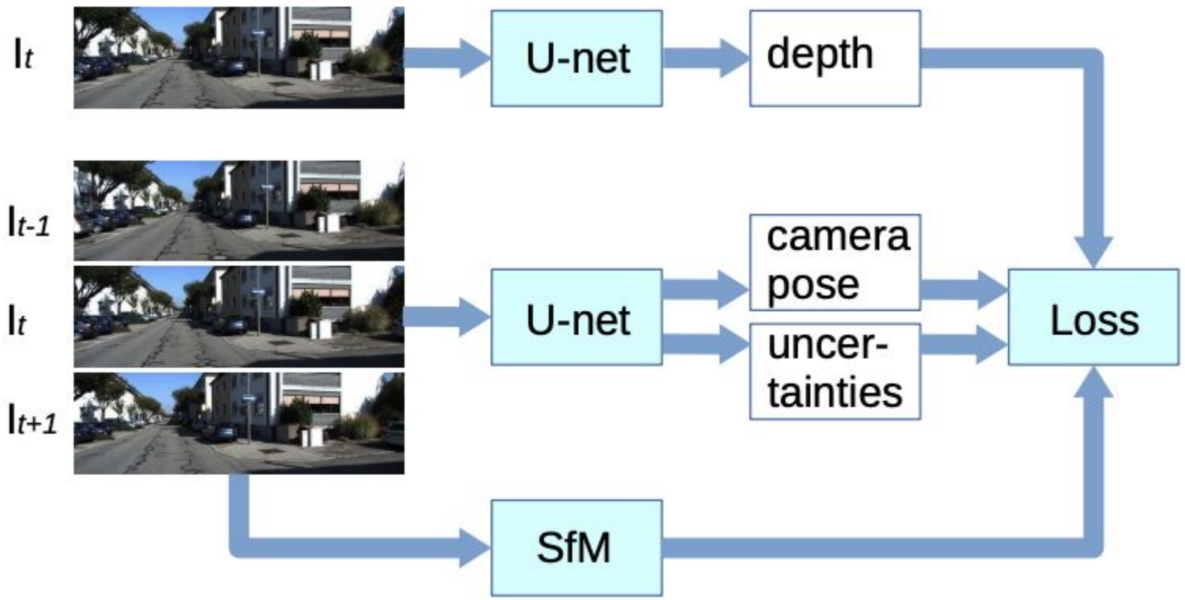
See also MVSFormer: Multi-View Stereo by Learning Robust Image Features and Temperature-based Depth (TMLR'23)

Supervising the new with the old: learning SFM from SFM

Maria Klodt[0000-0003-3015-9584] and Andrea Vedaldi[0000-0003-1374-2858]

Visual Geometry Group, University of Oxford
{klodt,vedaldi}@robots.ox.ac.uk

Abstract. Recent work has demonstrated that it is possible to learn deep neural networks for monocular depth and ego-motion estimation from unlabelled video sequences, an interesting theoretical development with numerous advantages in applications. In this paper, we propose a number of improvements to these approaches. First, since such self-supervised approaches are based on the brightness constancy assumption, which is valid only for a subset of pixels, we propose a probabilistic learning formulation where the network predicts distributions over variables rather than specific values. As these distributions are conditioned on the observed image, the network can learn which scene and object types are likely to violate the model assumptions, resulting in more robust learning. We also propose to build on dozens of years of experience in developing handcrafted structure-from-motion (SFM) algorithms. We do so by using an off-the-shelf SFM system to generate a supervisory signal for the deep neural network. While this signal is also noisy, we show that our probabilistic formulation can learn and account for the defects of SFM, helping to integrate different sources of information and boosting the overall performance of the network.



(b) proposed network architecture: the depth and pose-uncertainty networks are supervised by traditional SfM.

https://openaccess.thecvf.com/content_ECCV_2018/papers/Maria_Klodt_Supervising_the_new_ECCV_2018_paper.pdf

What did we learn today?

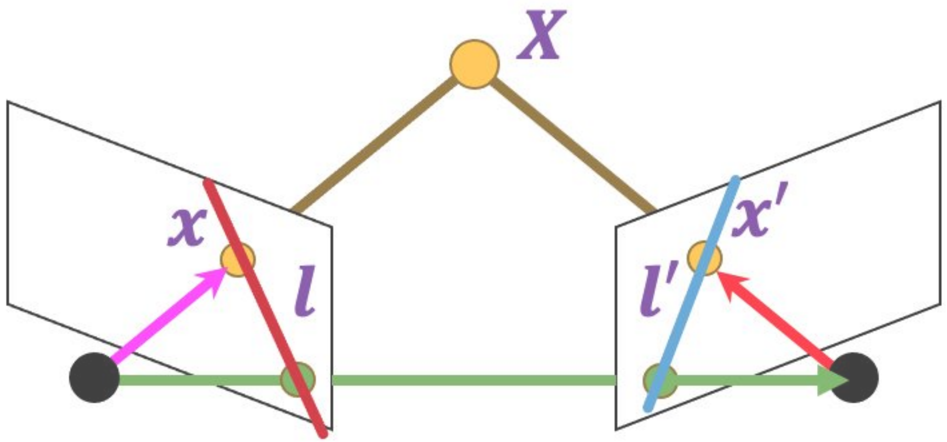
Triangulation

from calibrated cameras and 2D correspondences to 3D points

Epipolar geometry

$$x'^T F x = 0$$

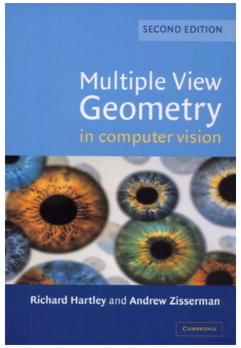
Epipolar constraint,
Essential & Fundamental matrices



Stereo (fronto-parallel special case)

Structure-from-Motion (SfM) (many images / video)

Note: this is just an introduction.



Wrapping up Geometric Vision

Homogeneous Coordinates & Projective Space

2D & 3D Transforms as Matrix Multiplication

Pinhole Camera Model P=K[R|t]

Calibration from known 3D-to-2D correspondences

Multi-view geom: fundamental matrix, stereo, SfM