Review of Last Lectures

CSE P576
Vitaly Ablavsky

Mathematical Morphology

$$\mathbf{X} = \mathbb{Z}_m^+ \times \mathbb{Z}_n^+ = \left\{(x_1, x_2) \in \mathbb{Z}^2 \,:\, 1 \leq x_1 \leq m,\ 1 \leq x_2 \leq n\right\}.$$

We follow standard practice and represent these rectangular point sets by listing the points in matrix form. Figure 1.2.1 provides a graphical representation of the point set $\mathbf{X} = \mathbb{Z}_m^+ \times \mathbb{Z}_n^+$.

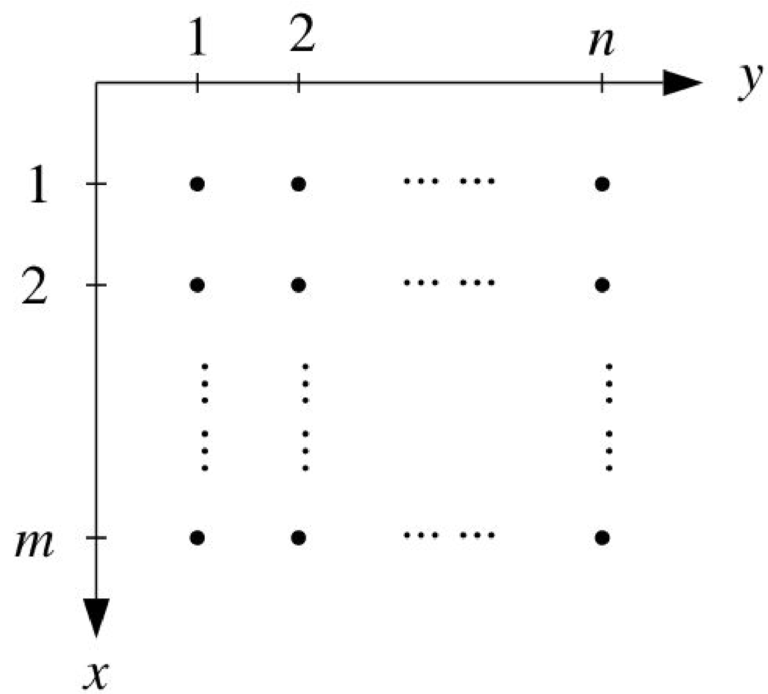


Figure 1.2.1. The rectangular point set $\mathbf{X} = \mathbb{Z}_m \times \mathbb{Z}_n$

[Ritter and Wilson, Handbook of Computer Vision Algorithms in Image Algebra, 1996]

Mathematical Morphology

COMPUTER VISION, GRAPHICS, AND IMAGE PROCESSING 35, 283–305 (1986)

Introduction to Mathematical Morphology

JEAN SERRA

E. N. S. M. de Paris, Paris, France

Received October 6, 1983; revised March 20, 1986

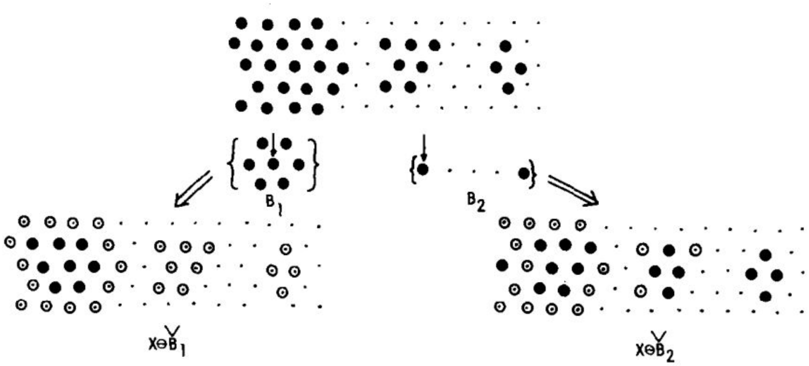


FIG. 6. Two different B's extract different features when eroding the same X (the arrow indicates the location of the origin in the structuring element).

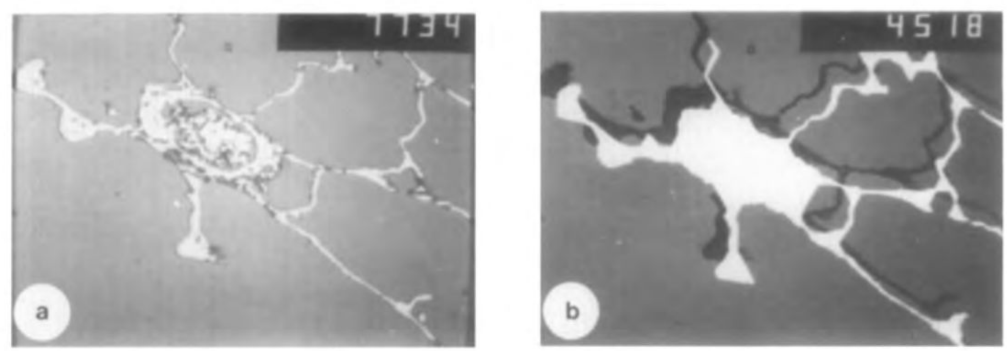


FIG. 9. (a) Thin section of a human lung. The presence of a vessel disturbs the measurement of the alveolae perimeter. (b) An opening of the background (i.e., closing at the alveolae) allows a correct computation. (Photographs taken on the monitor of a LEITZ T.A.S.)

Mathematical Morphology

IEEE TRANSACTIONS ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, VOL. ASSP-35, NO. 8, AUGUST 1987 1153

Morphological Filters—Part I: Their Set-Theoretic Analysis and Relations to Linear Shift-Invariant Filters

PETROS MARAGOS, MEMBER, IEEE, AND RONALD W. SCHAFER, FELLOW, IEEE

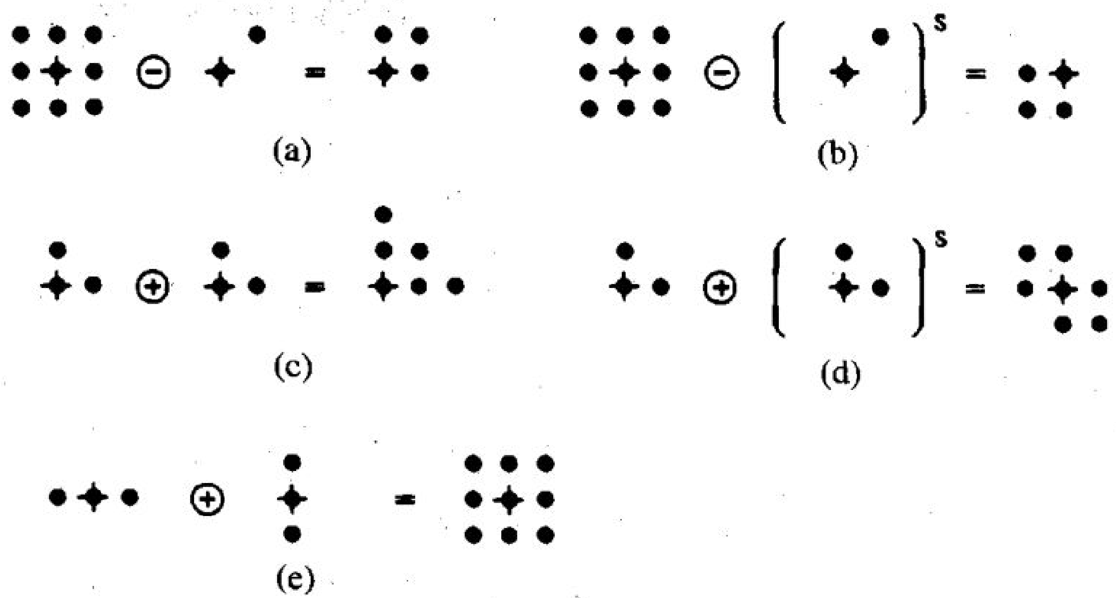


Fig. 6. Dilations and erosions of discrete sets: (a) Minkowski subtraction; (b) erosion; (c) Minkowski addition, (d) dilation; (e) forming larger sets as the Minkowski sum of simpler sets. (● = set points; + marks origin $(0, 0)$ of $\mathbf{Z}^2$.)

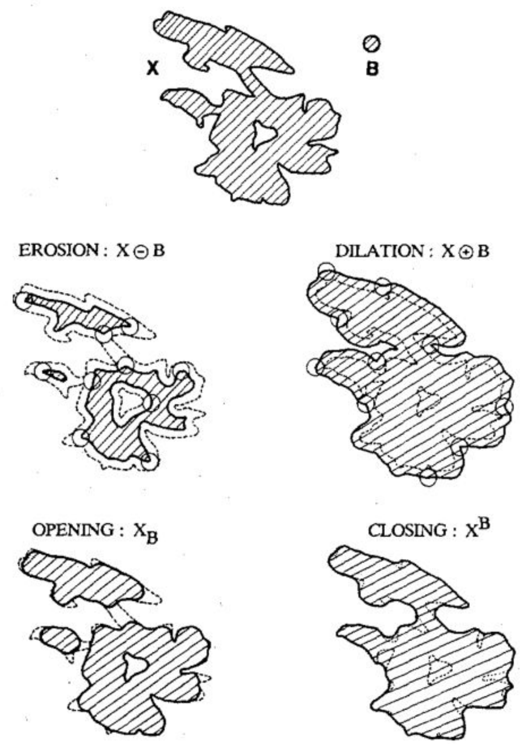


Fig. 5. Erosion, dilation, opening, and closing of X by B (the shaded areas correspond to the interior of the sets, the dark solid curve to the boundary of the transformed sets, and the dashed curve to the boundary of the original set).

Corner Detection Revisited

Without loss of generality, we will assume a grayscale 2-dimensional image is used. Let this image be given by I. Consider taking an image patch $(x, y) \in W$ (window) and shifting it by $(\Delta x, \Delta y)$. The *sum of squared differences* (SSD) between these two patches, denoted f, is given by:

$$f(\Delta x, \Delta y) = \sum_{(x_k, y_k) \in W} \left(I(x_k, y_k) - I(x_k + \Delta x, y_k + \Delta y)\right)^2$$

$I(x + \Delta x, y + \Delta y)$ can be approximated by a Taylor expansion. Let I_x and I_y be the partial derivatives of I, such that

$$I(x + \Delta x, y + \Delta y) \approx I(x, y) + I_x(x, y)\Delta x + I_y(x, y)\Delta y$$

This produces the approximation

$$f(\Delta x, \Delta y) \approx \sum_{(x,y) \in W} \left(I_x(x, y)\Delta x + I_y(x, y)\Delta y\right)^2,$$

what's the size of W?
do all pixels contribute equally?

which can be written in matrix form:

$$f(\Delta x, \Delta y) \approx \begin{pmatrix} \Delta x & \Delta y \end{pmatrix} M \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix},$$

where M is the structure tensor,

$$M = \sum_{(x,y) \in W} \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} = \begin{bmatrix} \sum_{(x,y)\in W} I_x^2 & \sum_{(x,y)\in W} I_x I_y \\ \sum_{(x,y)\in W} I_x I_y & \sum_{(x,y)\in W} I_y^2 \end{bmatrix}$$

computations of $I_x^2, I_x I_y$, etc. are *per-pixel*

For $x \ll y$, one has $\frac{x \cdot y}{x + y} = x \frac{1}{1 + x/y} \approx x$. In this step, we compute the smallest eigenvalue of the structure tensor using that approximation:

$$\lambda_{min} \approx \frac{\lambda_1 \lambda_2}{(\lambda_1 + \lambda_2)} = \frac{\det(M)}{\operatorname{tr}(M)}$$

with the trace $\operatorname{tr}(M) = m_{11} + m_{22}$.

Another commonly used Harris response calculation is shown as below,

$$R = \lambda_1 \lambda_2 - k \cdot (\lambda_1 + \lambda_2)^2 = \det(M) - k \cdot \operatorname{tr}(M)^2$$

where k is an empirically determined constant; $k \in [0.04, 0.06]$.

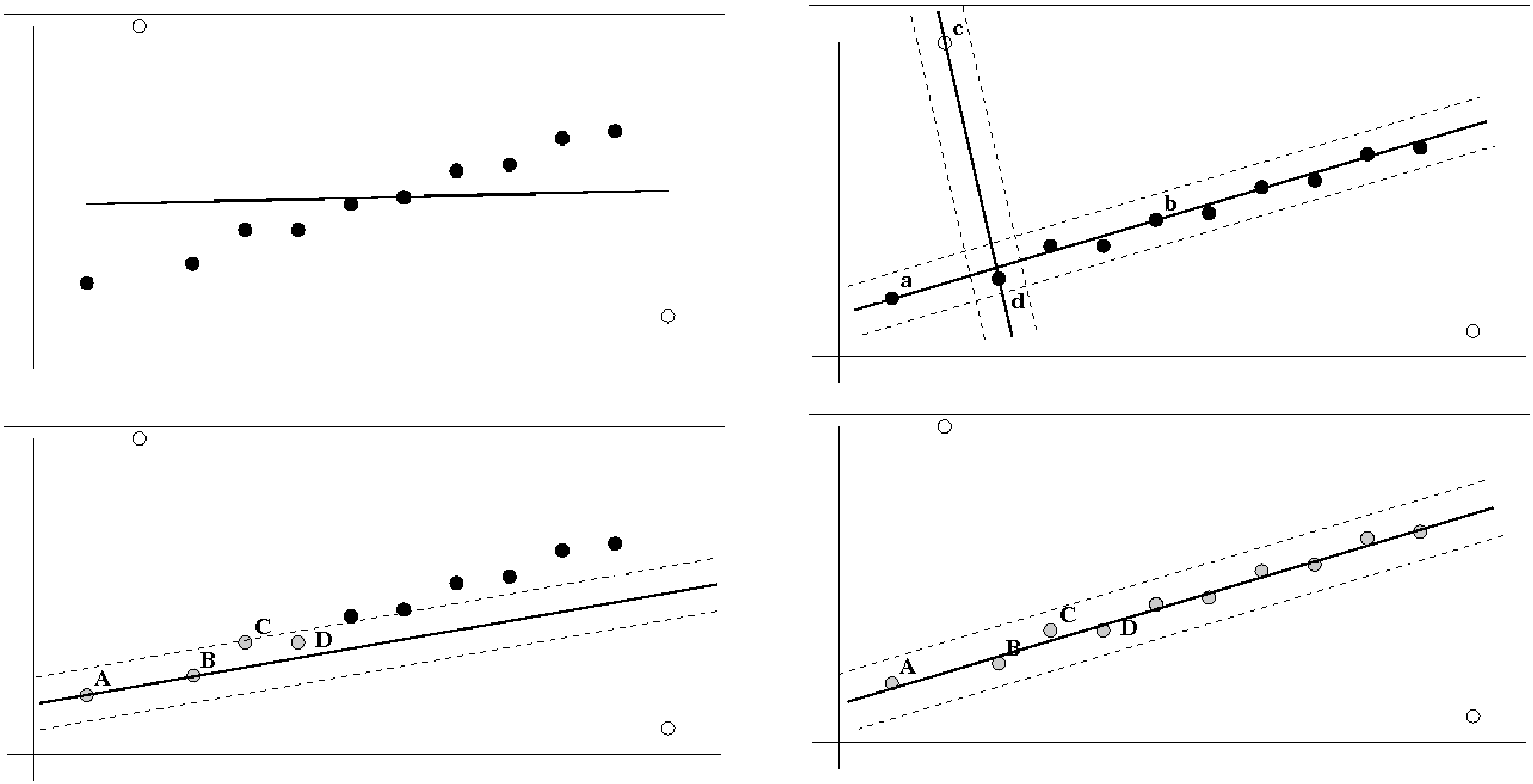
Credit: https://en.wikipedia.org/wiki/Harris_corner_detector

Peak-finding: Practicalities

A 1D example:

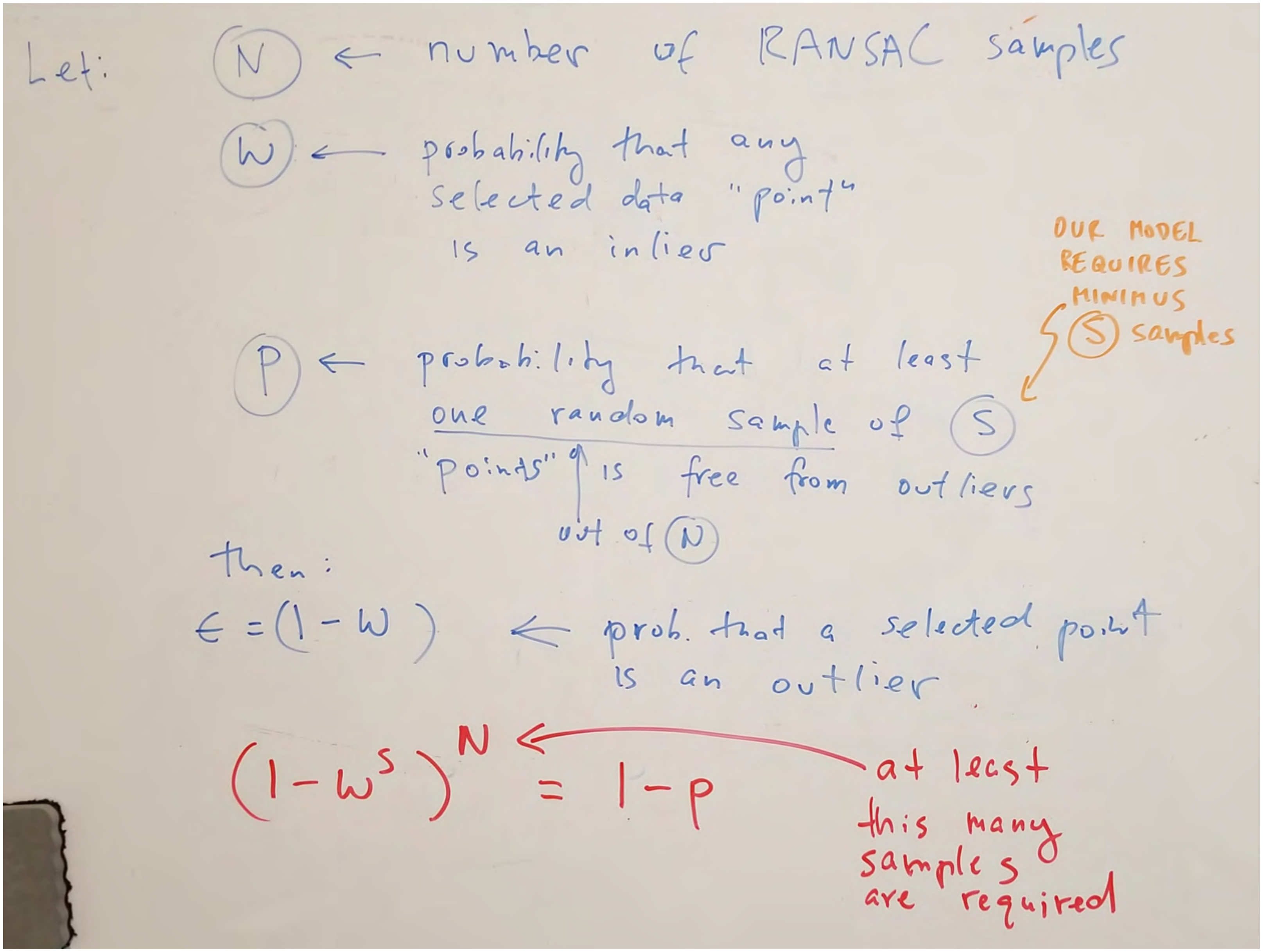
```
(Pdb) a2
array([0, 1, 2, 3, 4, 5, 5, 5, 5, 4, 5, 4, 7, 6, 2])
(Pdb) se2
array([ 1.000,  1.000,  1.000])
(Pdb) a2dse2 = ndimage.grey_dilation(a2,
footprint=se2)
(Pdb) a2dse2
array([1, 2, 3, 4, 5, 5, 5, 5, 5, 5, 5, 7, 7, 7, 6])
(Pdb) a2peaks_se2 = np.where((a2dse2 - a2) == 0,1,0)
(Pdb) a2peaks_se2
array([0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 0])
(Pdb) a2peaks_skimg =
skimage.feature.peak_local_max(a2,footprint=se2)
(Pdb) a2peaks_skimg
array([[12],
       [ 5],
       [ 6],
       [ 7],
       [ 8],
       [10]])
(Pdb) idx_1 = np.sort(a2peaks_skimg.flatten())
(Pdb) idx_1
array([ 5,  6,  7,  8, 10, 12])
(Pdb) idx_0 = np.where(a2peaks_se2)
(Pdb) idx_0
(array([ 5,  6,  7,  8, 10, 12]),)
(Pdb)
```

Random Sampling Consensus (RANSAC)



[Hartley and Zisserman, Ch. 4]

RANSAC:
How Many Samples Are Needed?



Epipolar Geometry

CSE P576

Vitaly Ablavsky

These slides were developed by Dr. Matthew Brown for CSEP576 Spring 2020 and adapted (slightly) for Fall 2021
credit → Matt
blame → Vitaly

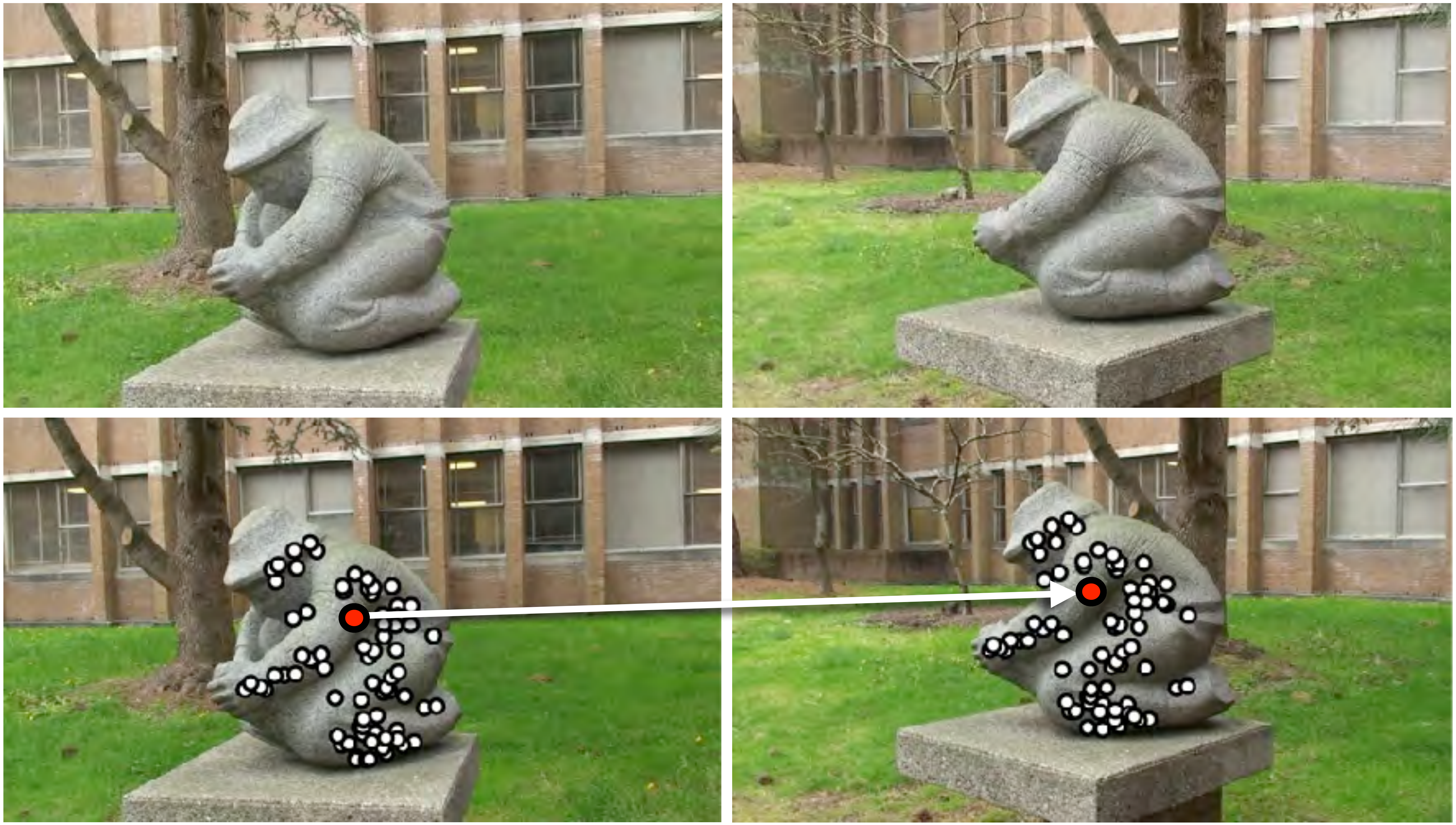
Epipolar Geometry

- Epipolar Lines, Plane Constraint
- Fundamental Matrix, Linear solution
- RANSAC for F, 2-view SFM

[Szeliski Chapters 7+11]

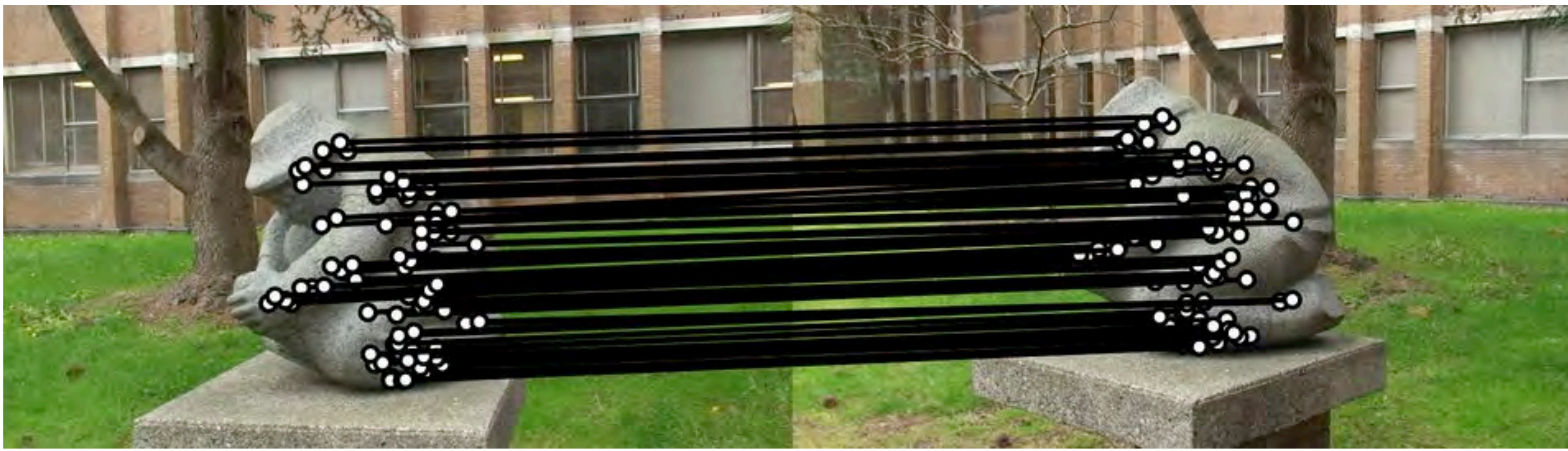
Correspondence

- Find all matches between views



Geometric Constraints

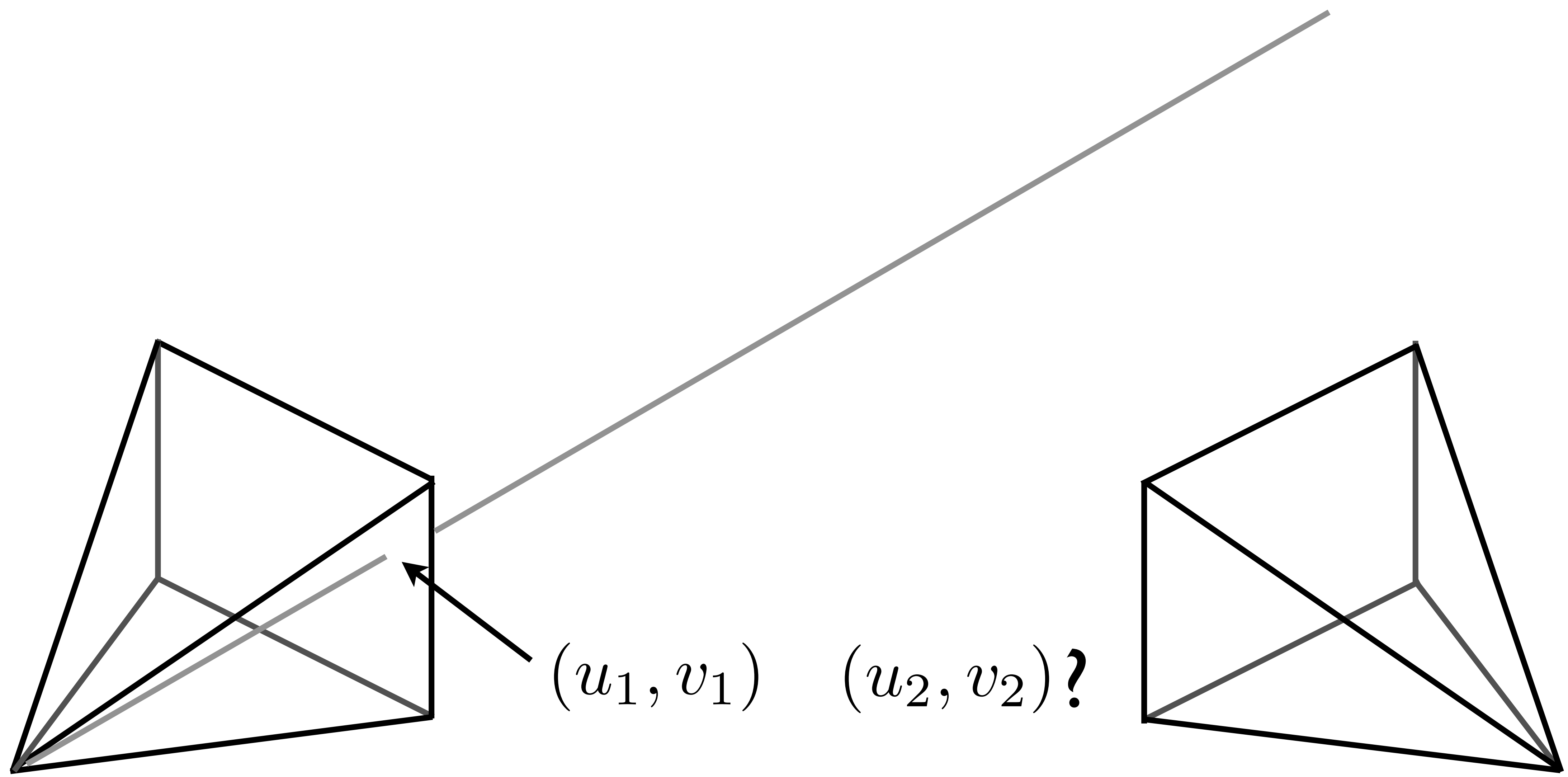
- Find subset of matches that are consistent with a geometric transformation



Consistent matches can be used for subsequent stages, e.g., 3D reconstruction, object recognition etc.

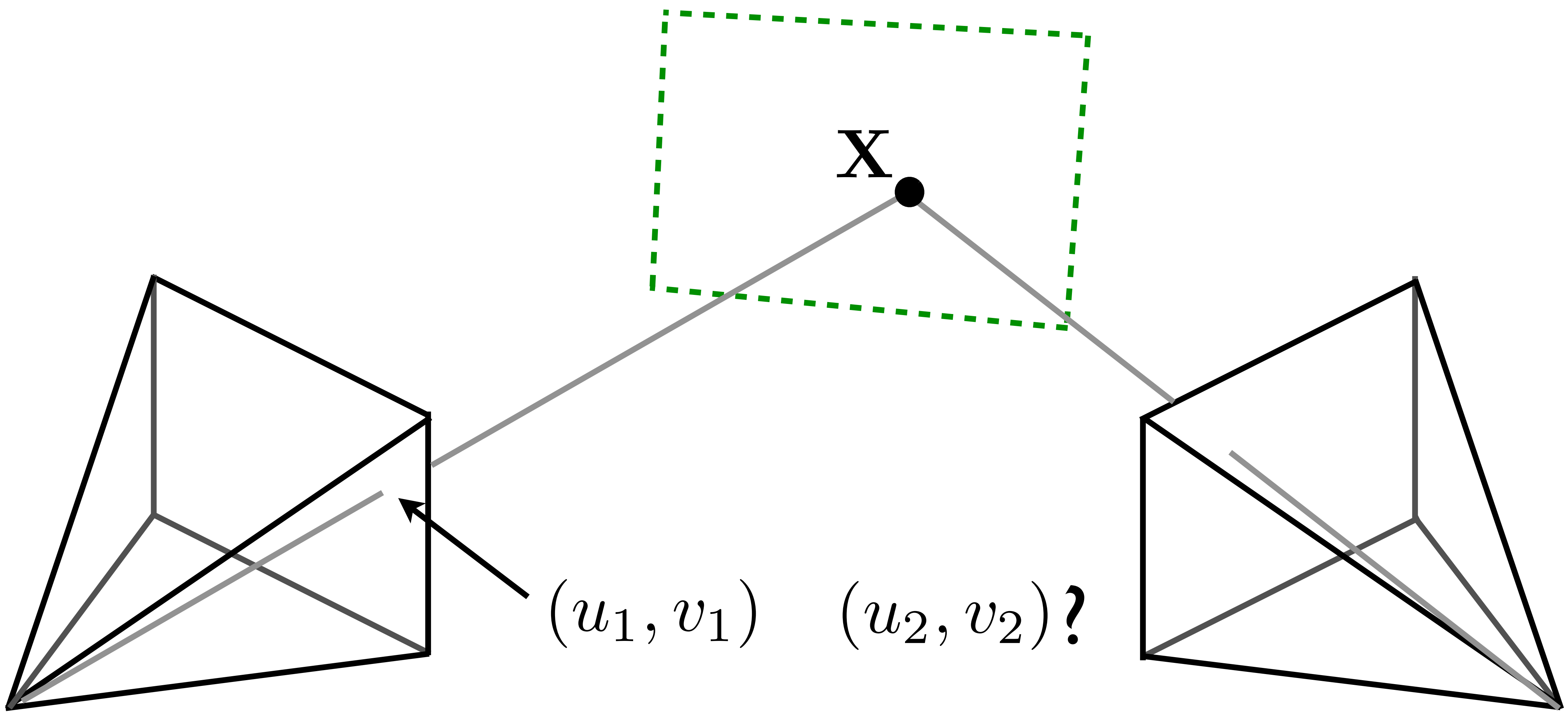
2-view Geometry

- How do we transfer points between 2 views?



2-view Geometry

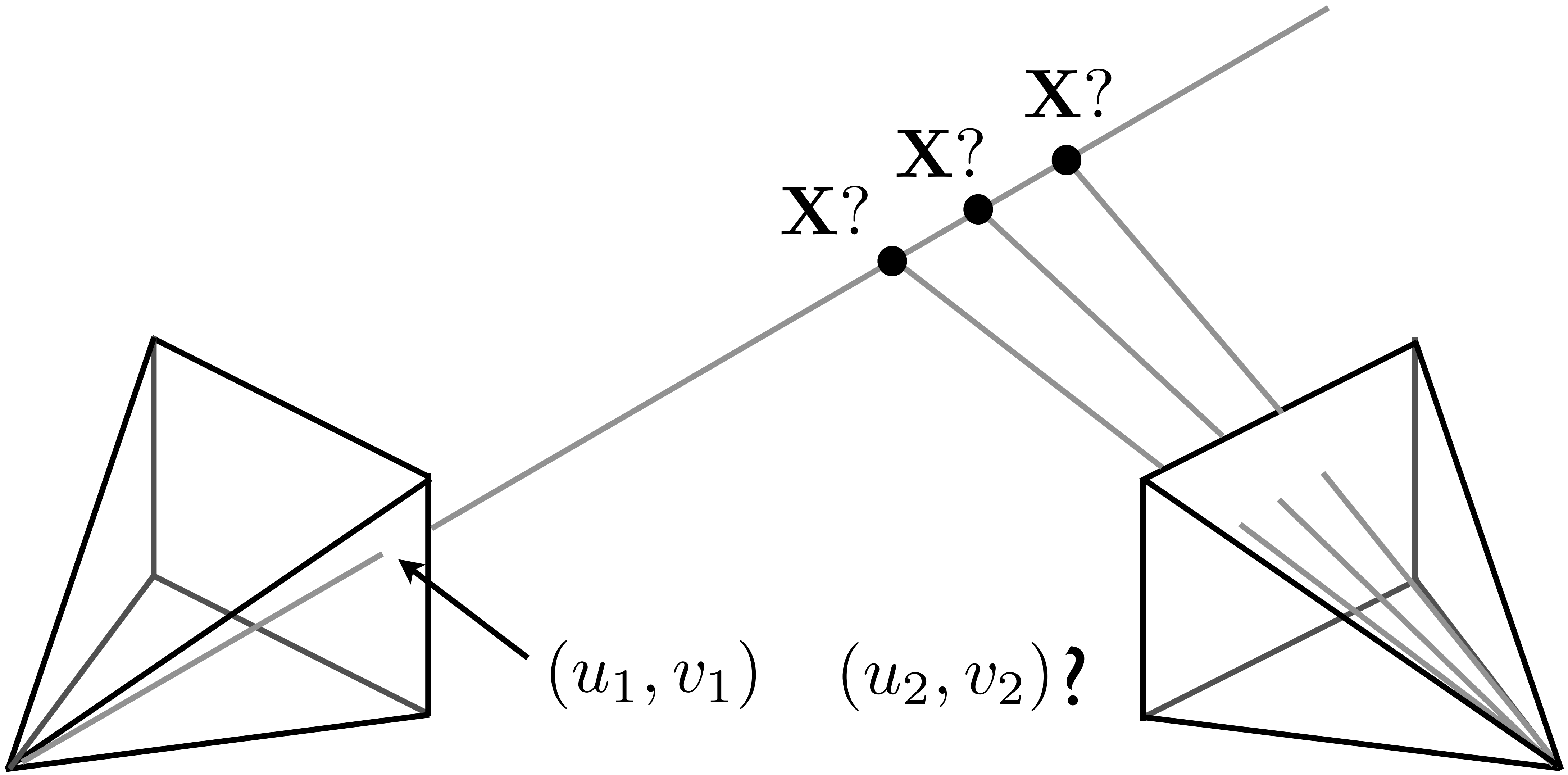
- How do we transfer points between 2 views? (planar case)



Planar case: one-to-one mapping via plane (Homography)

2-view Geometry

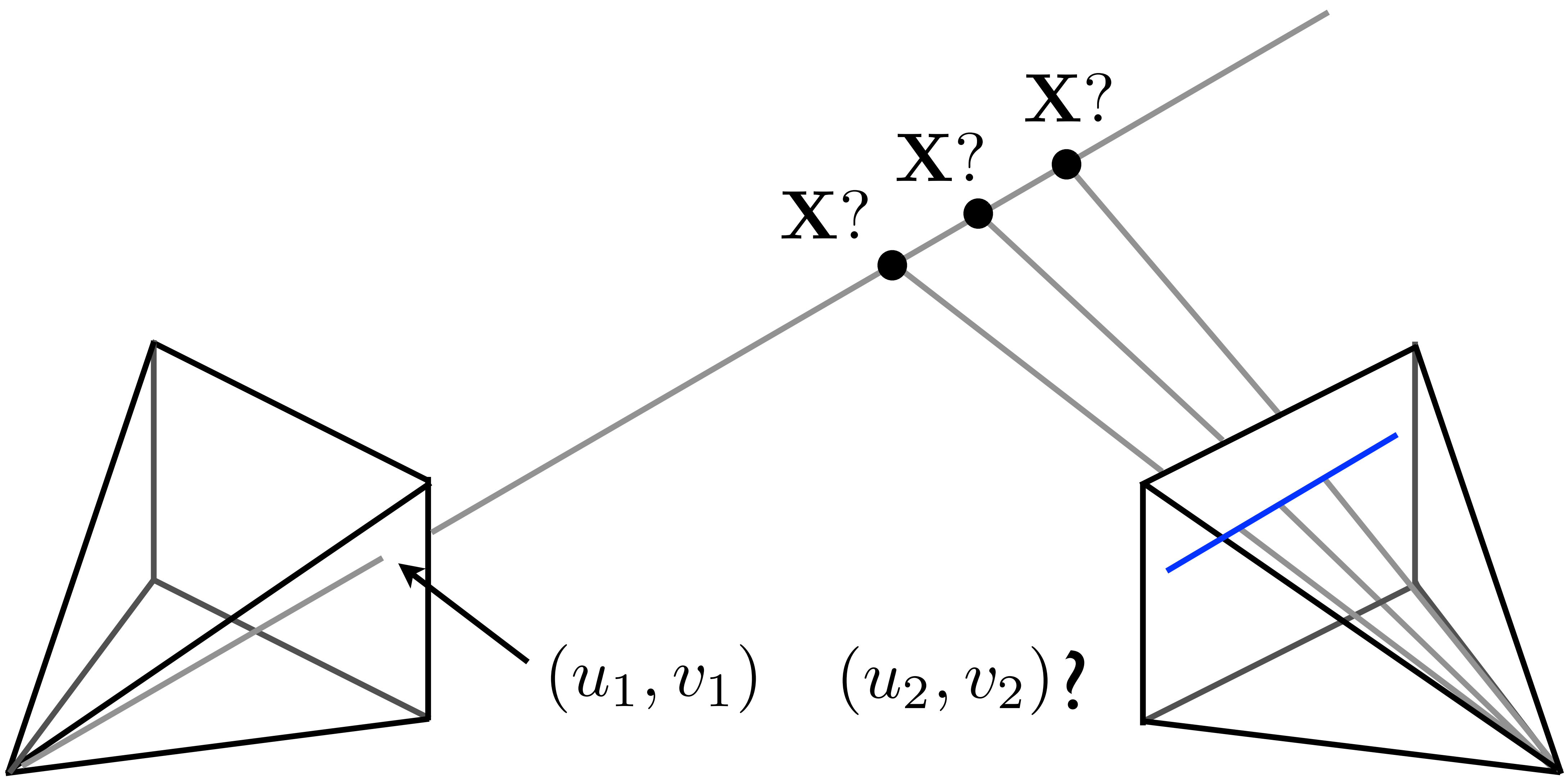
- How do we transfer points between 2 views? (non-planar)



Non-planar case: depends on the depth of the 3D point

Epipolar Line

- How do we transfer points between 2 views? (non-planar)

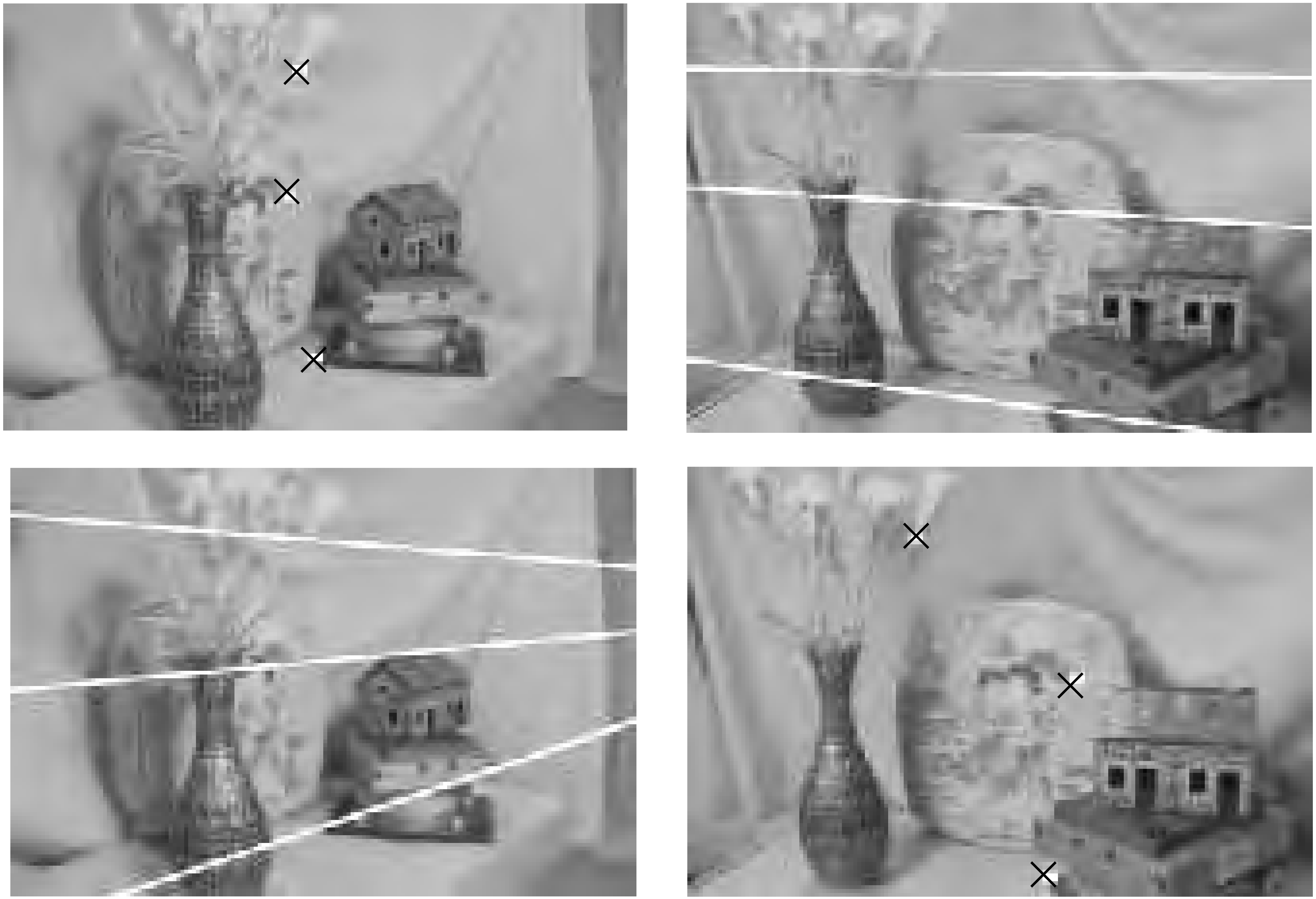


A point in image 1 gives a **line** in image 2

Epipolar Lines from F

- What is the equation of the epipolar line for point x?

Epipolar Lines

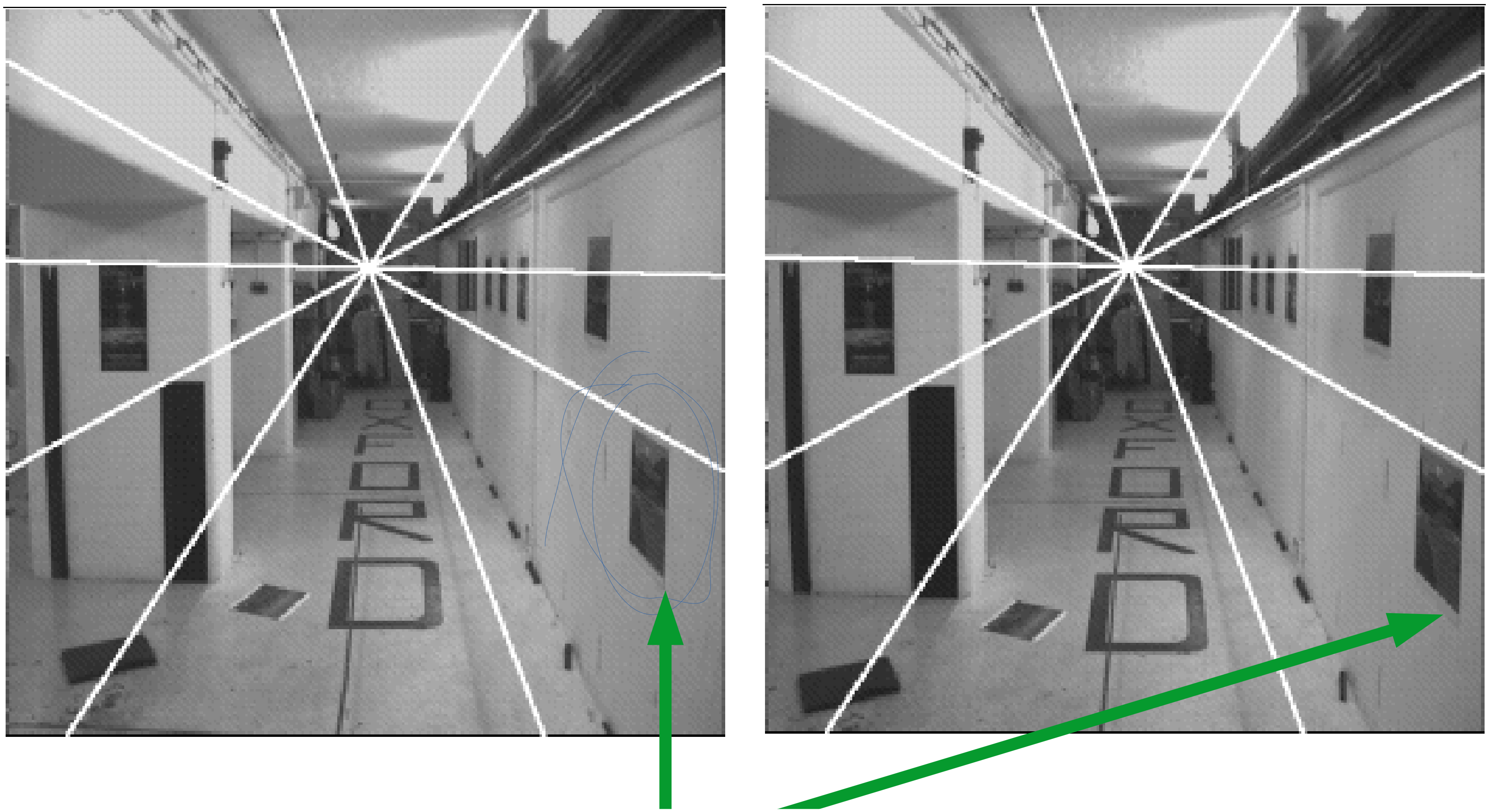


[R. Cipolla]

Epipolar Lines

[R. Cipolla]

Focus of Expansion

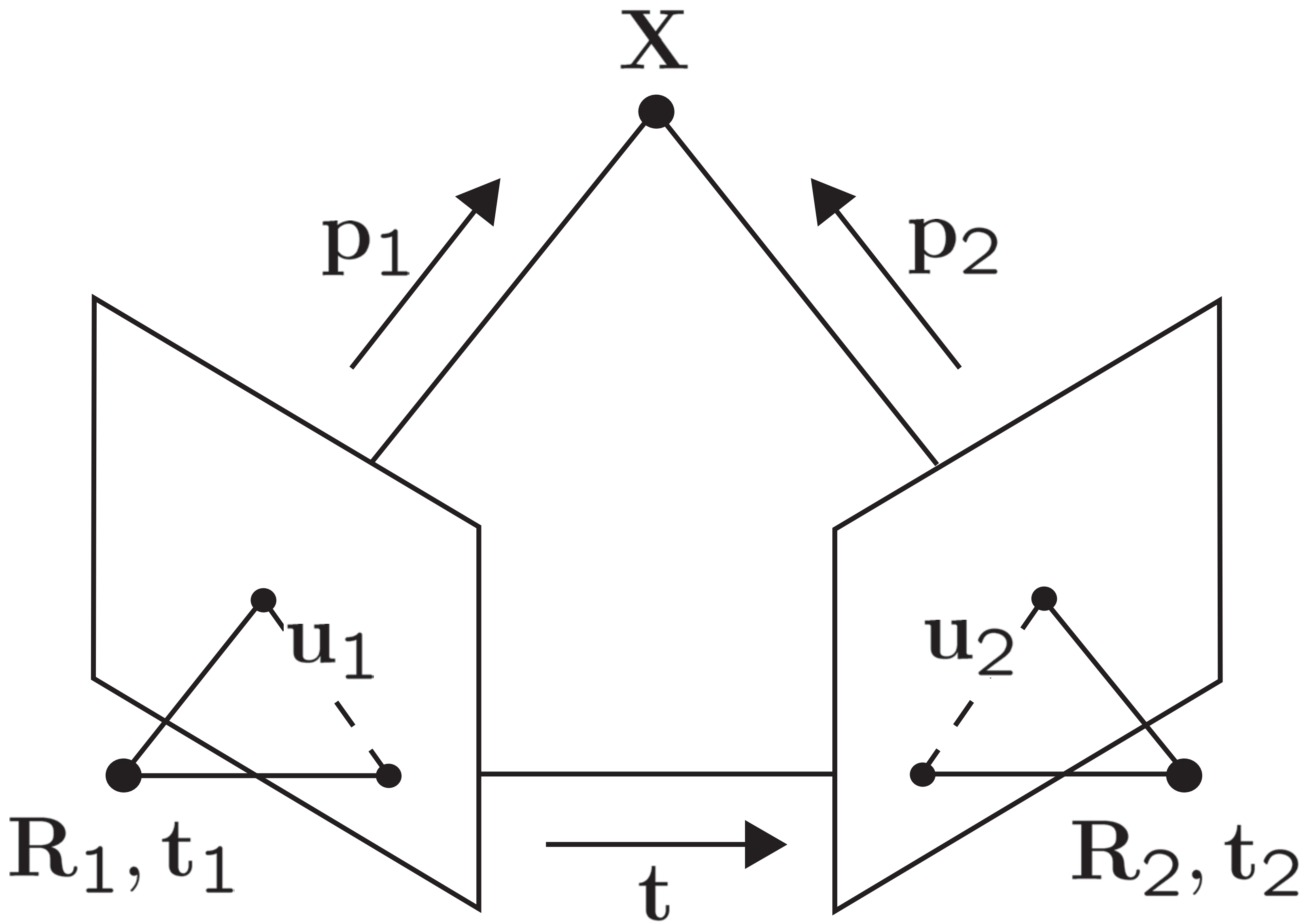


forward motion

[Hartley and Zisserman, Ch. 9]

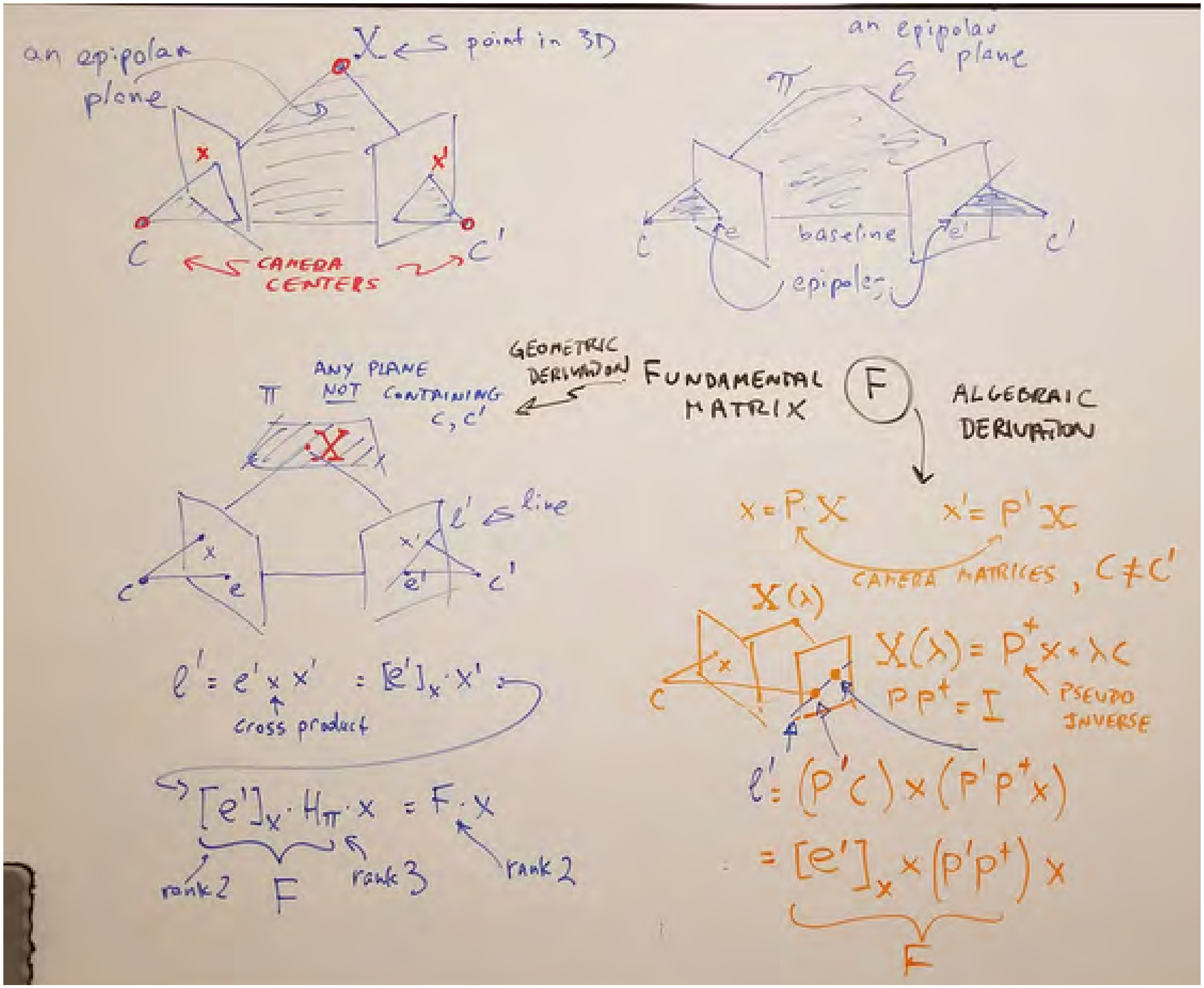
The Epipolar Constraint

- For rays to intersect at a point (X), the two rays and the camera translation must lie in the same plane



[Szeliski 11.3]

Epipolar geometry: geometric and algebraic derivations



Computing F

- Single correspondence gives us one equation

$$\begin{bmatrix} u_1 & v_1 & 1 \end{bmatrix} \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = 0$$

- Multiply out

$$u_1 x_1 f_{11} + u_1 y_1 f_{12} + u_1 f_{13} + v_1 x_1 f_{21} + v_1 y_1 f_{22} \\ + v_1 f_{23} + x_1 f_{31} + y_1 f_{32} + f_{33} = 0$$

Computing F

- Rearrange for unknowns, add points by stacking rows

$$\begin{bmatrix} u_1x_1 & u_1y_1 & u_1 & v_1x_1 & v_1y_1 & v_1 & x_1 & y_1 & 1 \\ u_2x_2 & u_2y_2 & u_2 & v_2x_2 & v_2y_2 & v_2 & x_2 & y_2 & 1 \\ u_3x_3 & u_3y_3 & u_3 & v_3x_3 & v_3y_3 & v_3 & x_3 & y_3 & 1 \\ u_4x_4 & u_4y_4 & u_4 & v_4x_4 & v_4y_4 & v_4 & x_4 & y_4 & 1 \\ u_5x_5 & u_5y_5 & u_5 & v_5x_5 & v_5y_5 & v_5 & x_5 & y_5 & 1 \\ u_6x_6 & u_6y_6 & u_6 & v_6x_6 & v_6y_6 & v_6 & x_6 & y_6 & 1 \\ u_7x_7 & u_7y_7 & u_7 & v_7x_7 & v_7y_7 & v_7 & x_7 & y_7 & 1 \\ u_8x_8 & u_8y_8 & u_8 & v_8x_8 & v_8y_8 & v_8 & x_8 & y_8 & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = \begin{bmatrix} 0 \end{bmatrix}$$

- This is a linear system of the form $\mathbf{Af} = \mathbf{0}$

  can be solved using Singular Value Decomposition (SVD)

Epipolar Geometry

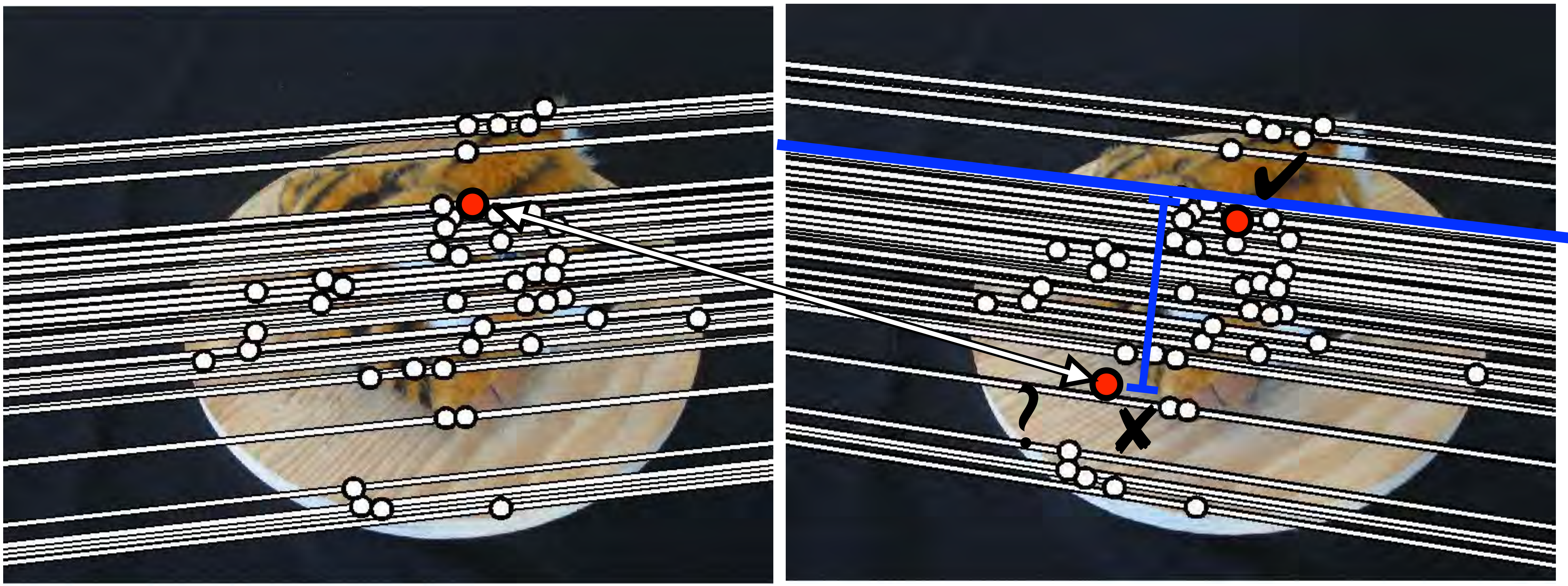
- Example: 2-view matching in 3D


Epipolar Geometry

- Raw SIFT matches

Epipolar Geometry

- Epipolar lines



Can use RANSAC to find inliers with small distance from epipolar line

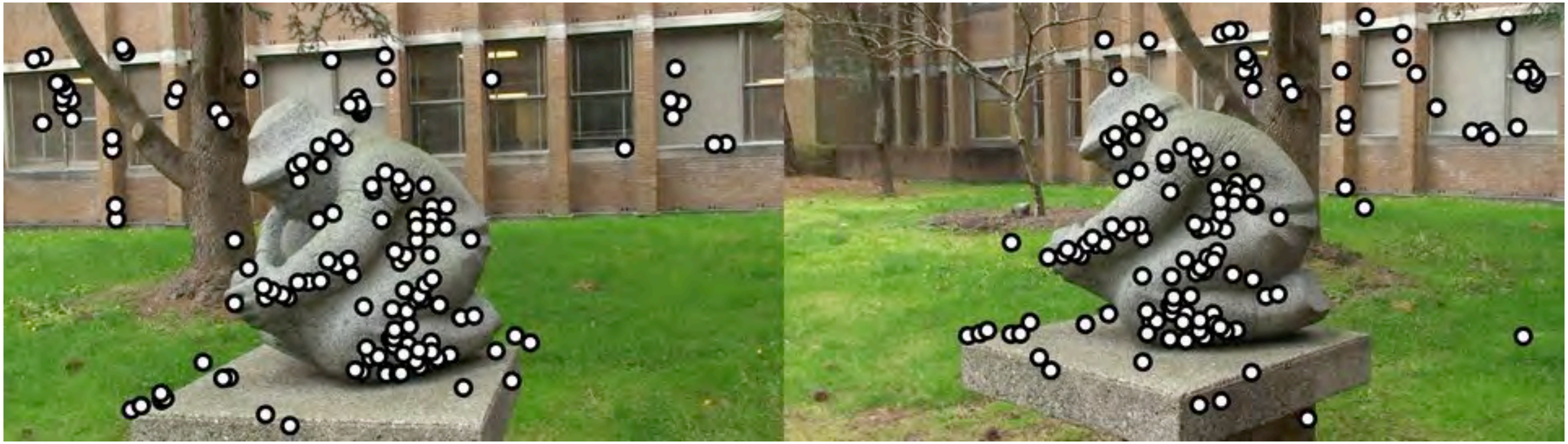
Epipolar Geometry

- Consistent matches


RANSAC for F

1. Match Features between 2 views
2. Randomly select set of 8 matches
3. Compute F using 8-point algorithm (SVD to solve Af=0)
4. Check consistency of all points with F, compute distances to epipolar lines and count #inliers with distance < threshold
5. Repeat steps 2-4 to maximise #inliers

RANSAC for F

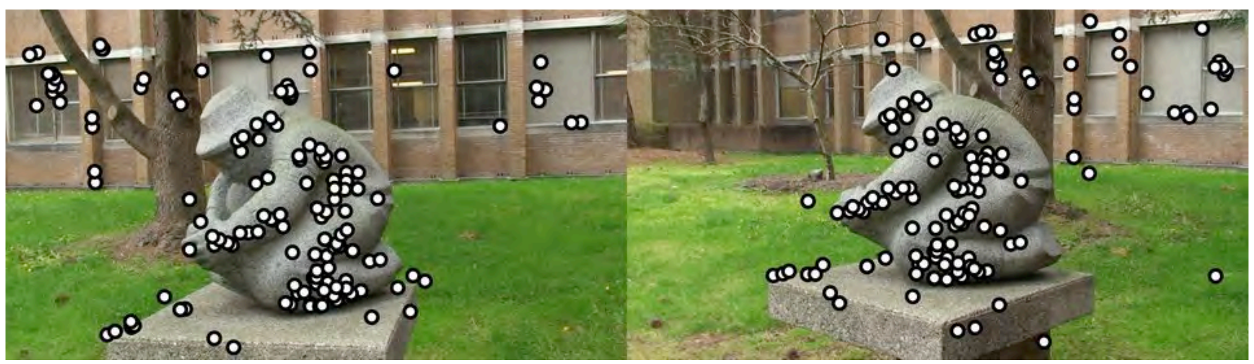


Raw feature matches (after ratio test filtering)

Solved for F and RANSAC inliers

2-view Structure from Motion

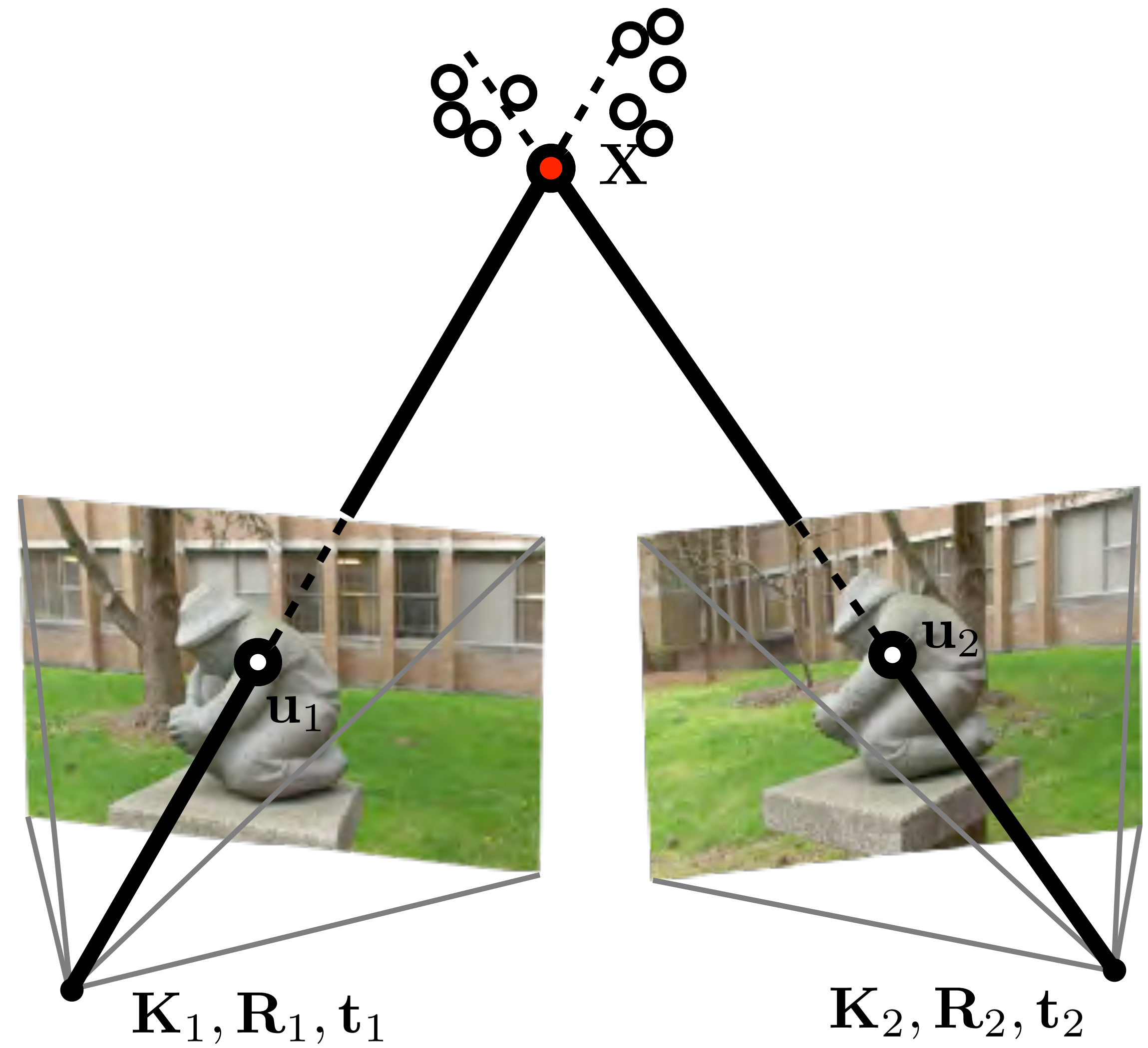
- We can use the combination of SIFT/RANSAC and triangulation to compute 3D structure from 2 views



Raw SIFT matches

RANSAC for F

Extract R, t



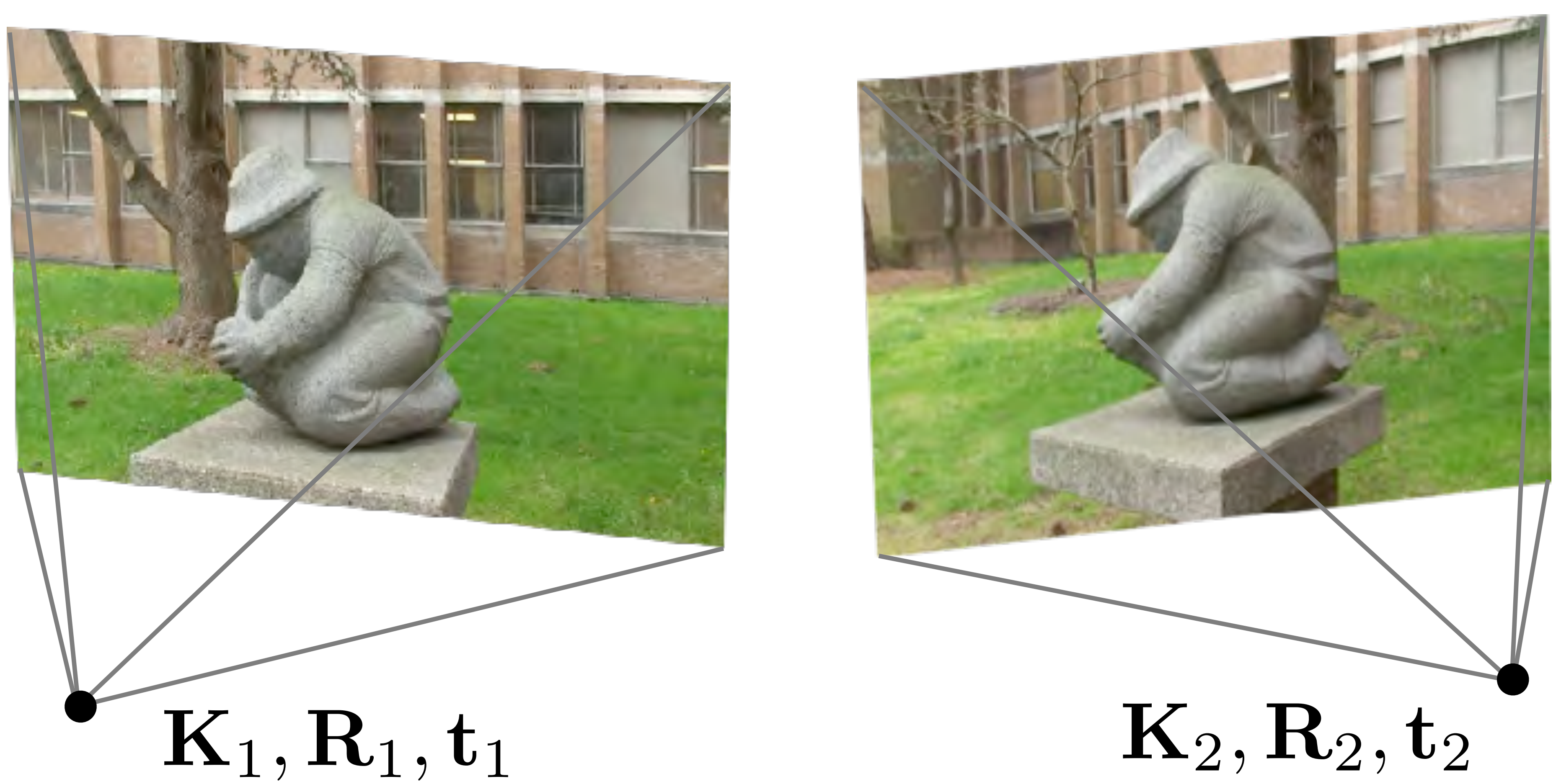
Triangulate to 3D Point Cloud

Cameras from F

- The Fundamental matrix is derived from the cameras

$$\mathbf{u}_2^T \underbrace{\mathbf{K}_2^{-T}\mathbf{R}_2^T(\mathbf{t}_2 - \mathbf{t}_1) \times \mathbf{R}_1\mathbf{K}_1^{-1}}_{\mathbf{F}}\mathbf{u}_1 = 0$$

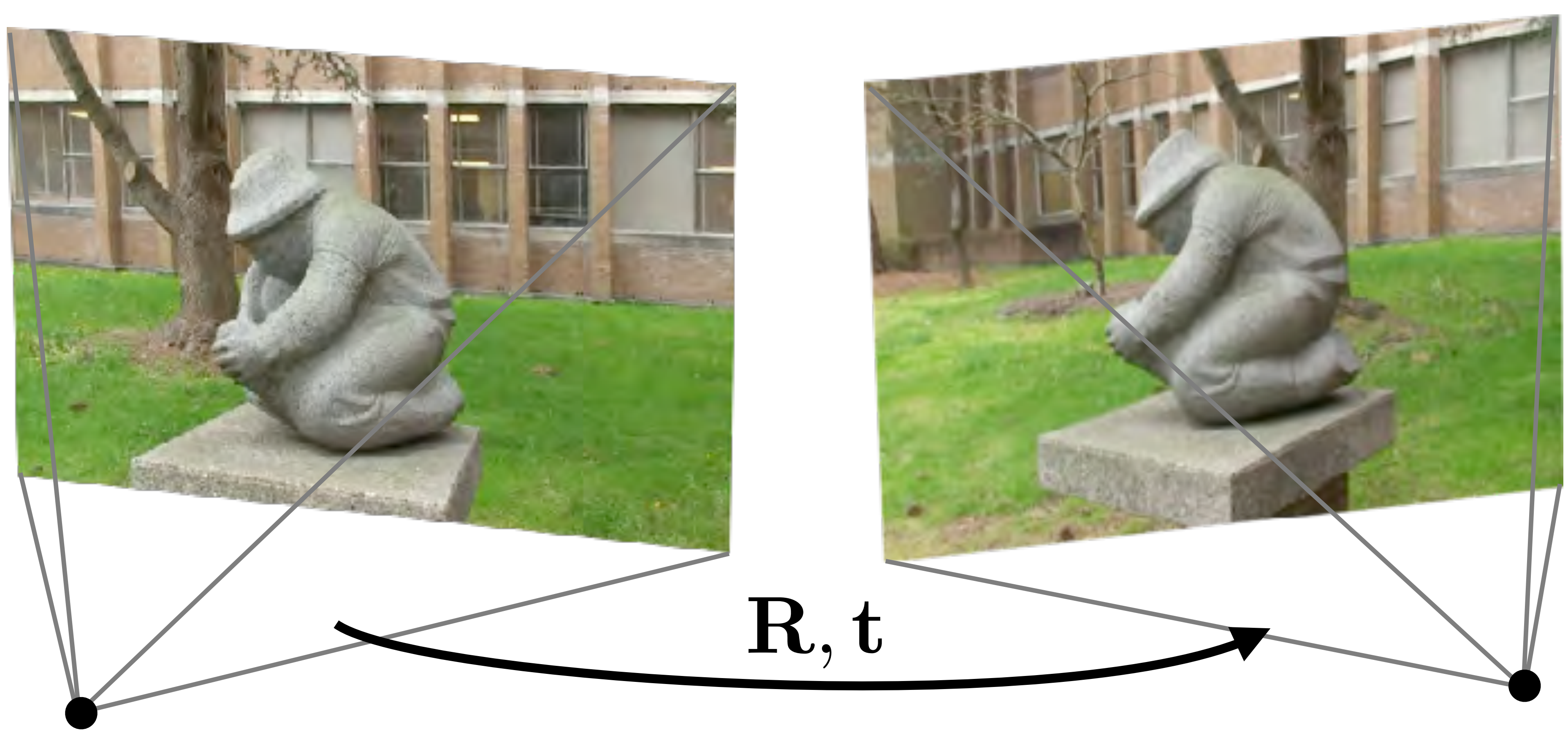
Can we invert it to get the cameras from F?



Cameras from F

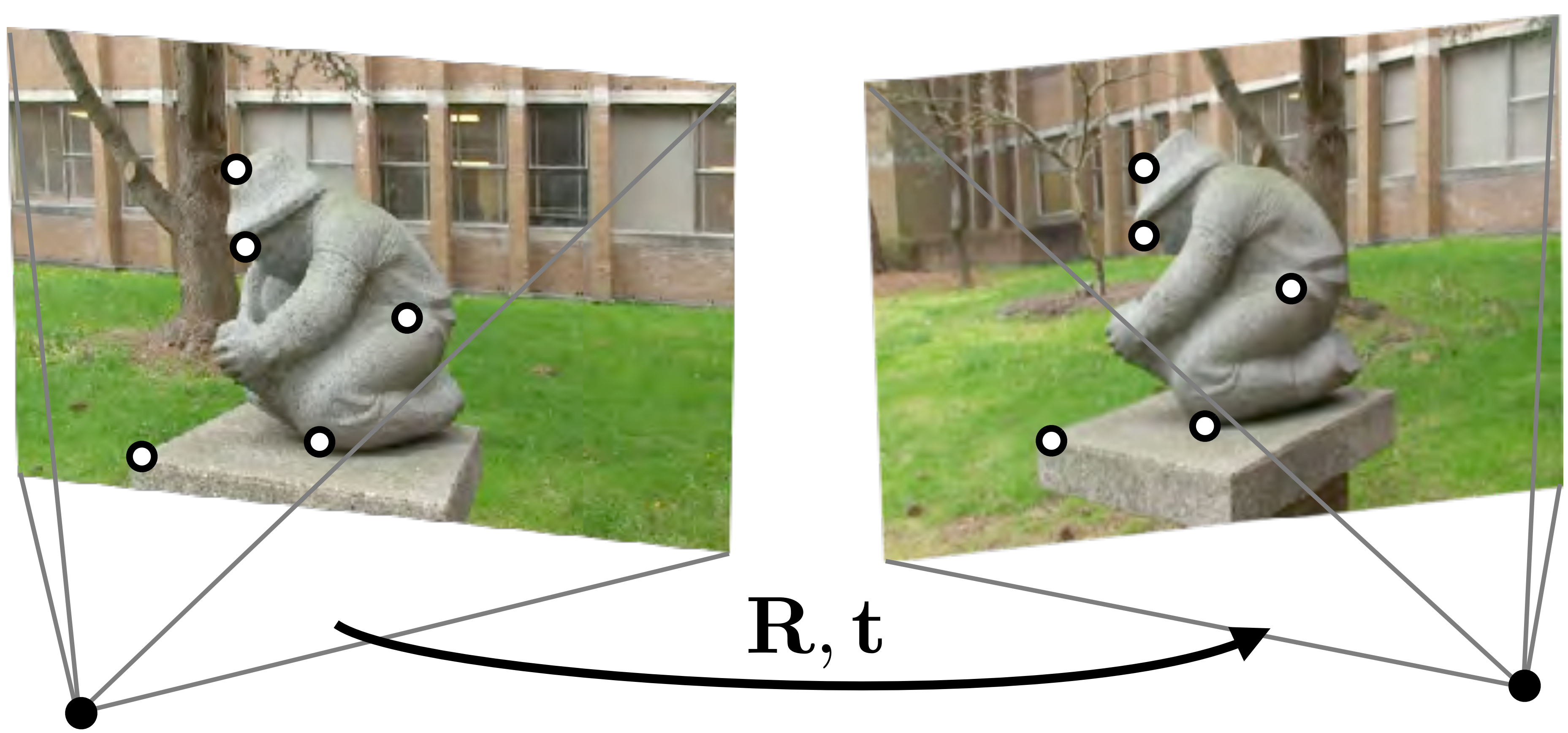
- First simplify by writing in terms of relative translation/ rotation and assume $\mathbf{K}_1, \mathbf{K}_2$ are known

$\mathbf{E} = [\mathbf{t}]_\times \mathbf{R}$ can be solved for **t**, **R** [Szeliski p350]



5 Point Algorithm

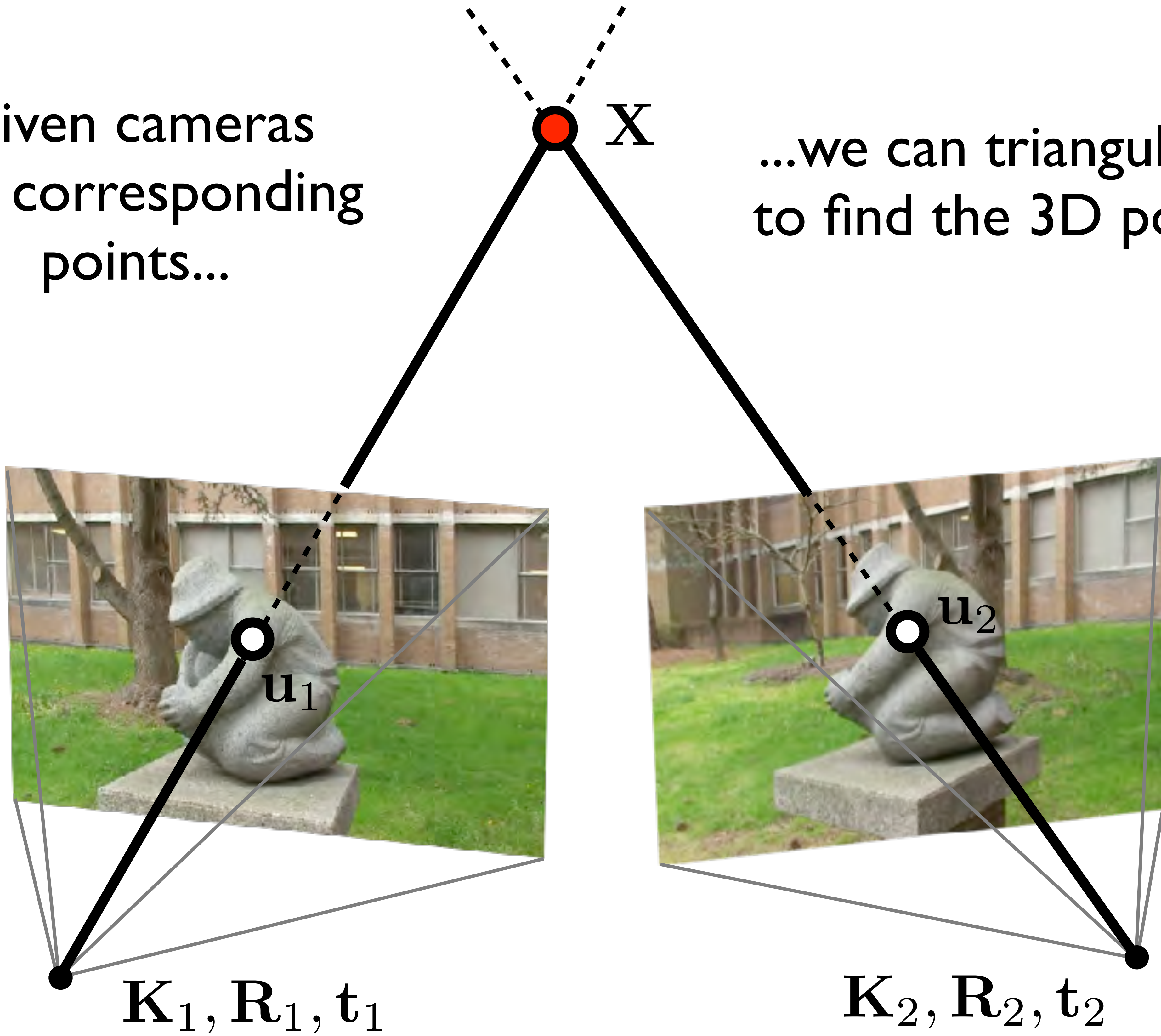
- Instead of using the 8 point algorithm to solve for F, we can directly solve for R and t using only 5 correspondences
- This involves solving a 10th degree polynomial [Nister 2004]
- Often we can guess the focal length (e.g., guess field of view), and solve for it later using bundle adjustment



Triangulation

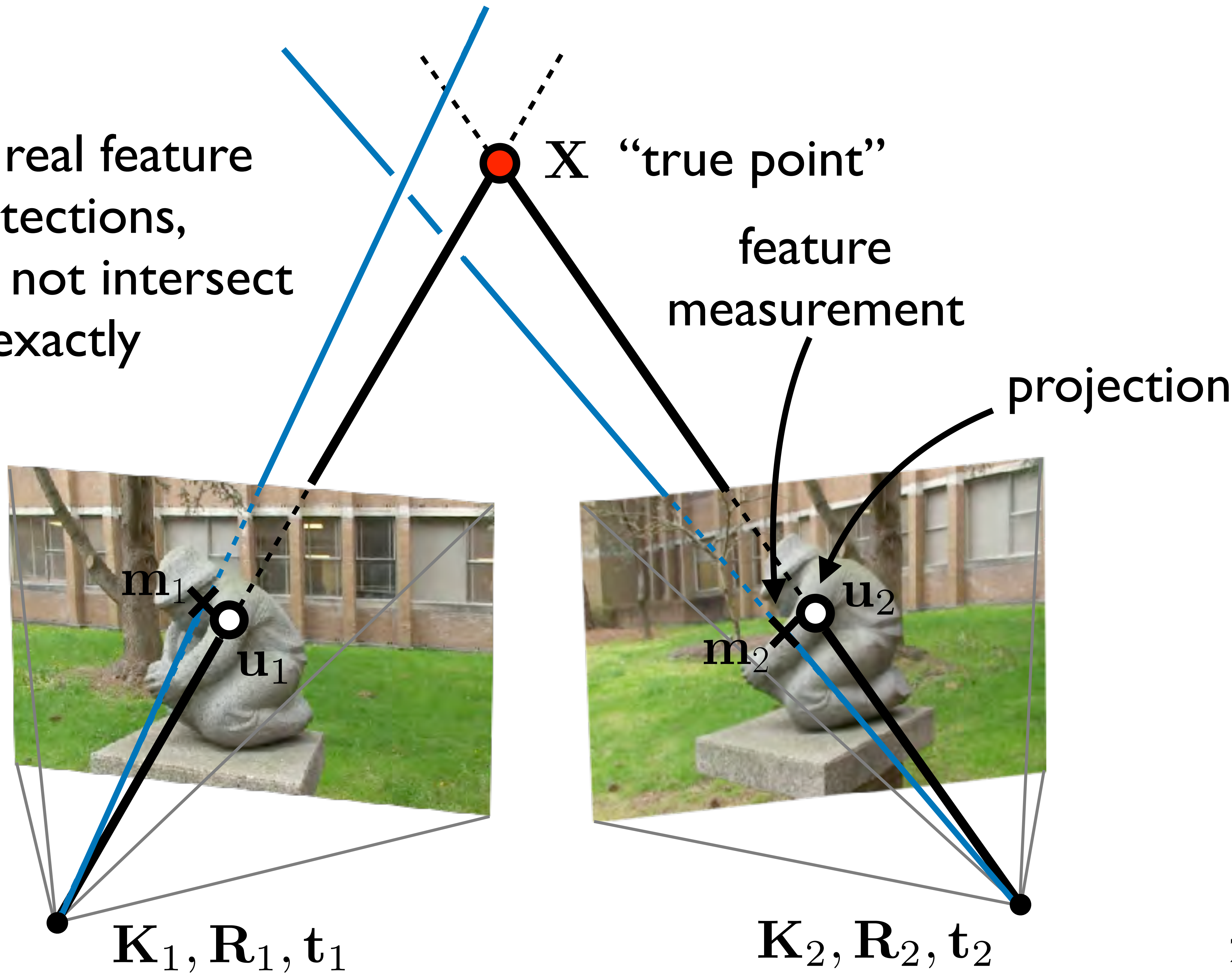
Given cameras and corresponding points...

...we can triangulate to find the 3D point



Triangulation

With real feature detections, rays do not intersect exactly



Triangulation

- We can solve for the 3D point X by minimising the closest approach of the rays in 3D (linear), or better find an X such that image measurement errors are minimised (non-linear)

[Szeliski 7.1]

Recap: 2-view Geometry

- Planar geometry: one to one mapping of points

$$\mathbf{u} = \mathbf{H}\mathbf{x}$$

viewing a plane, rotation

Recap: 2-view Geometry

- Epipolar (3D) geometry: point to line mapping

$$\mathbf{u}^T \mathbf{F} \mathbf{x} = 0$$

moving camera, 3D scene

Next Lecture

- Multiview alignment, structure from motion