CSE 152: Computer Vision

Hao Su

Lecture 19: Final Review

Coverage

- Anything taught this quarter may appear
- >90% on materials from L12 (3D Deep Learning)

Form

- 3 big problems that need calculation
  - Understand your homework well

- A few short Q&A

Principle No. 1

- No cheating!
- We will curve and be generous in grading
- But we will not tolerate any cheating

CSE 152: Computer Vision

Hao Su

Lecture 12: 3D Deep Learning

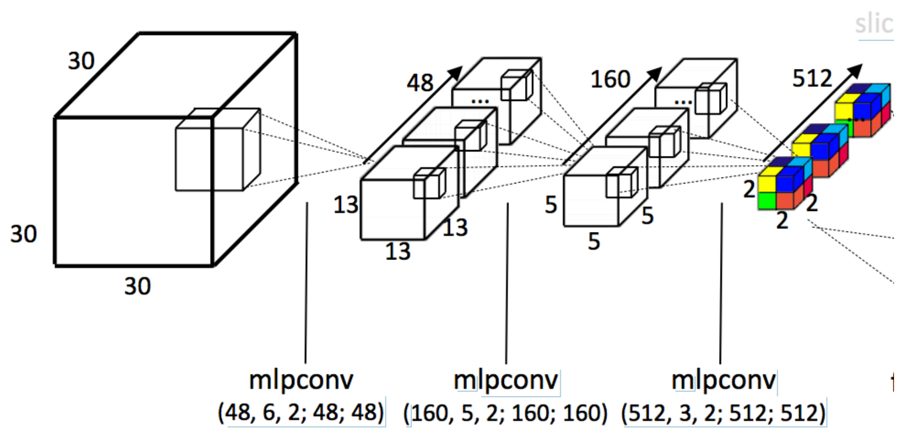
Credit: Stanford CS231n, L13

Voxelization

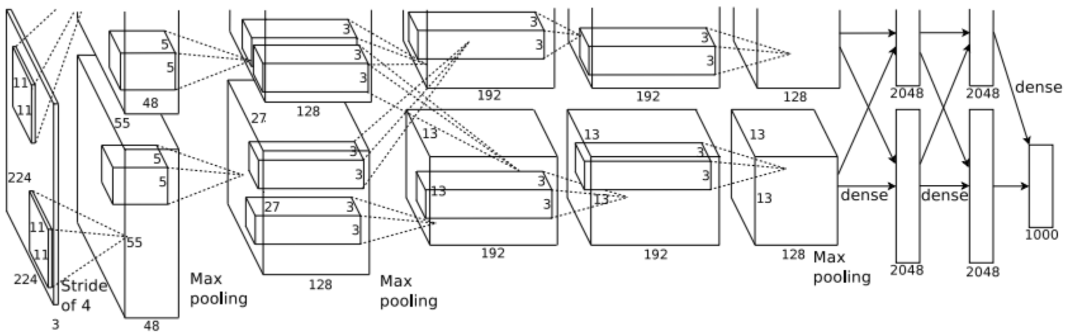
Represent the occupancy of regular 3D grids

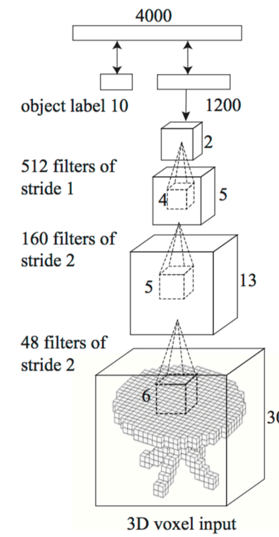
3D CNN on Volumetric Data

3D convolution uses 4D kernels



Complexity Issue





AlexNet, 2012

Input resolution: 224x224

224x224=50176

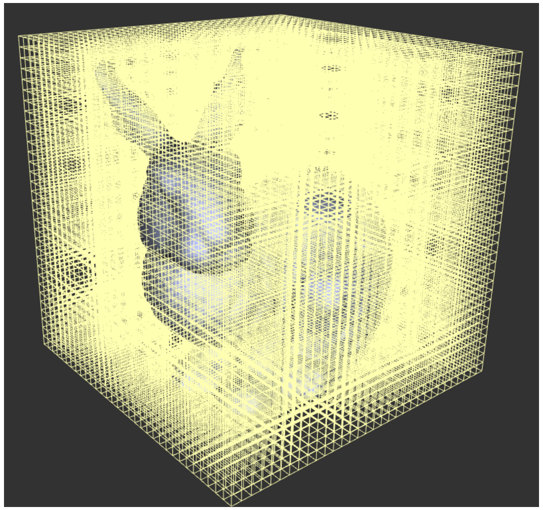
3DShapeNets, 2015

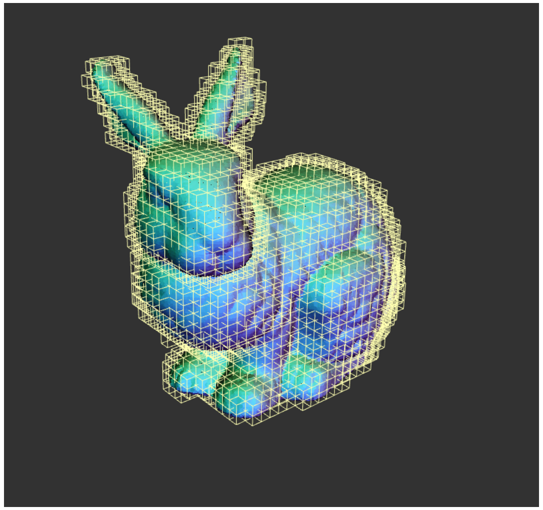
Input resolution: 30x30x30

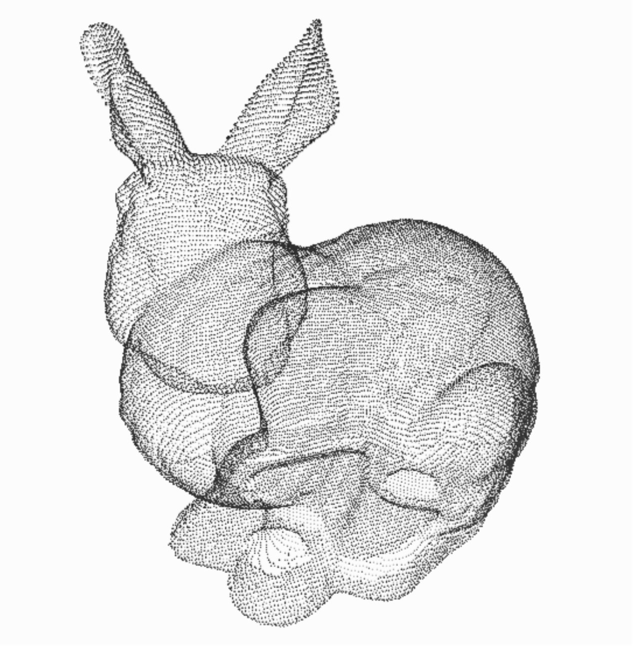
224x224=27000

Store only the Occupied Grids

- Store the sparse surface signals
- Constrain the computation near the surface





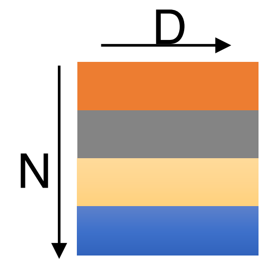


Point cloud

(The most common 3D sensor data)

Properties of a Desired Point Network

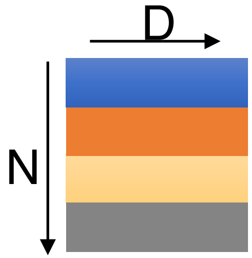
Point cloud: N **orderless** points, each represented by a D dim coordinate



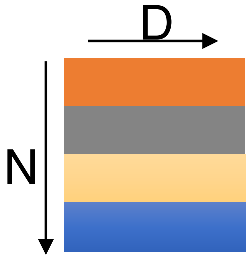
2D array representation

Permutation Invariance

Point cloud: N **orderless** points, each represented by a D dim coordinate



represents the same **set** as

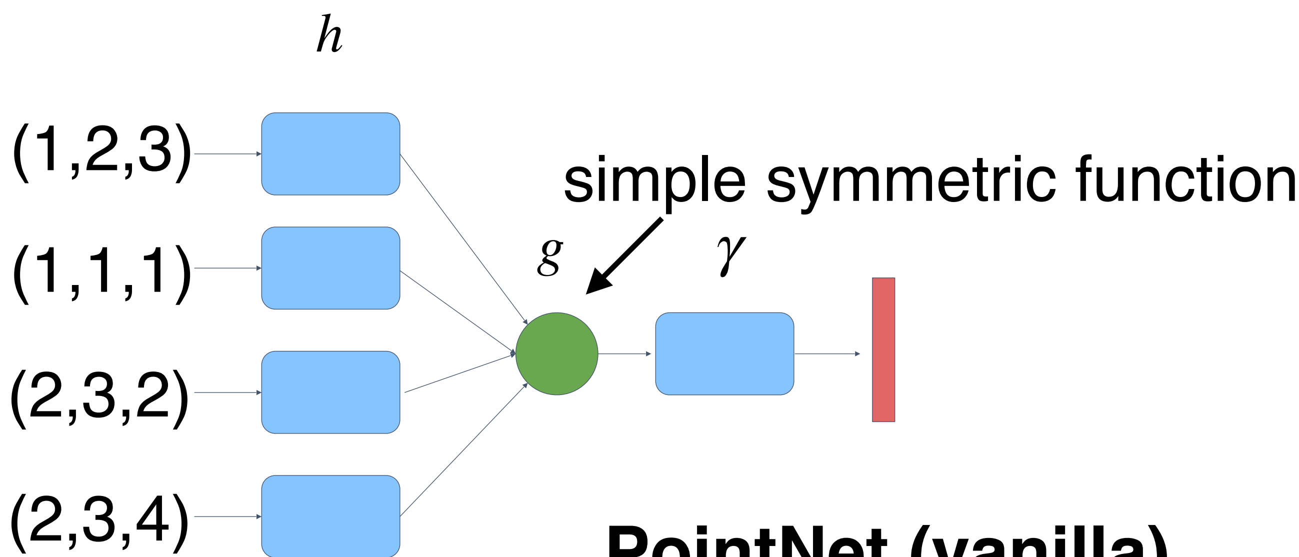


2D array representation

Construct a Symmetric Function

Observe:

$f(x_1, x_2, \ldots, x_n) = \gamma \circ g(h(x_1), \ldots, h(x_n))$ is symmetric if g is symmetric



PointNet (vanilla)

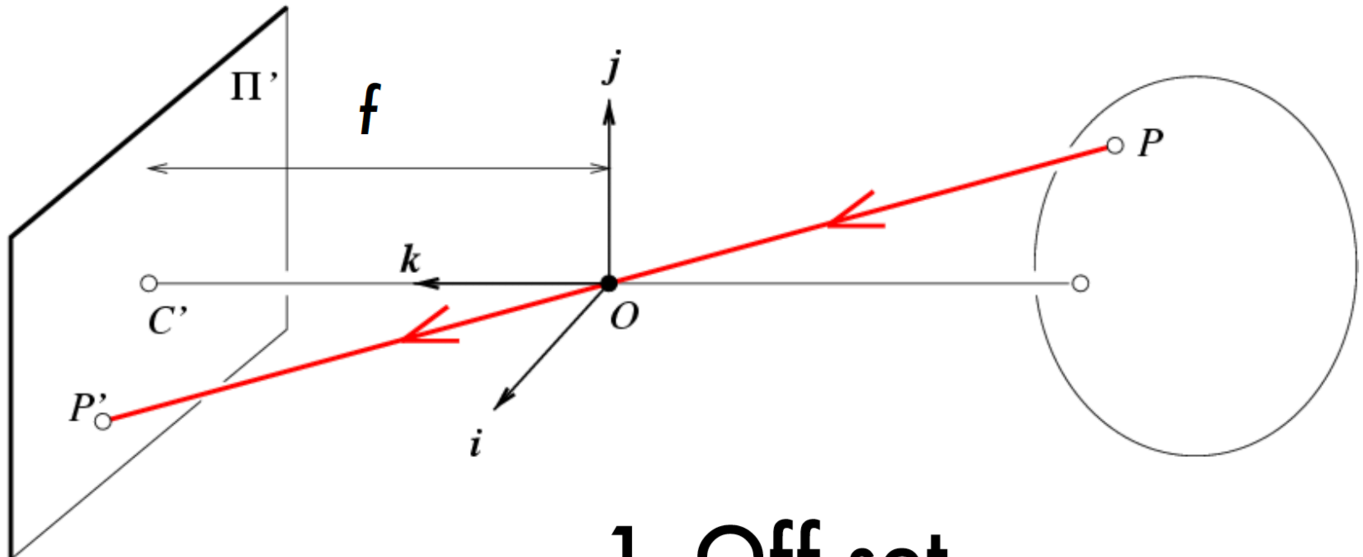
CSE 152: Computer Vision

Hao Su

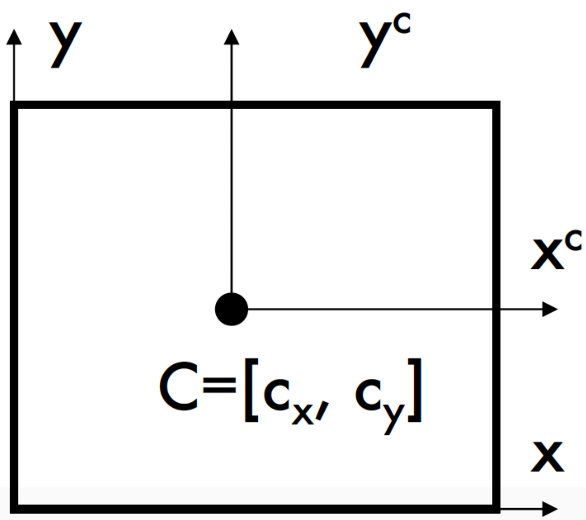
Lecture 13: Camera Models

Credit: CS231a, Stanford, Silvio Savarese

Converting to pixels



1. Off set
2. From metric to pixels



$$(x, y, z) \to \left(\underbrace{\boxed{f\,k}}_{\alpha}\frac{x}{z} + c_x, \underbrace{\boxed{f\,l}}_{\beta}\frac{y}{z} + c_y\right)$$

[Eq. 6]

Units: k,l : pixel/m

f : m

Non-square pixels

α, β : pixel

Projective transformation in the homogenous coordinate system

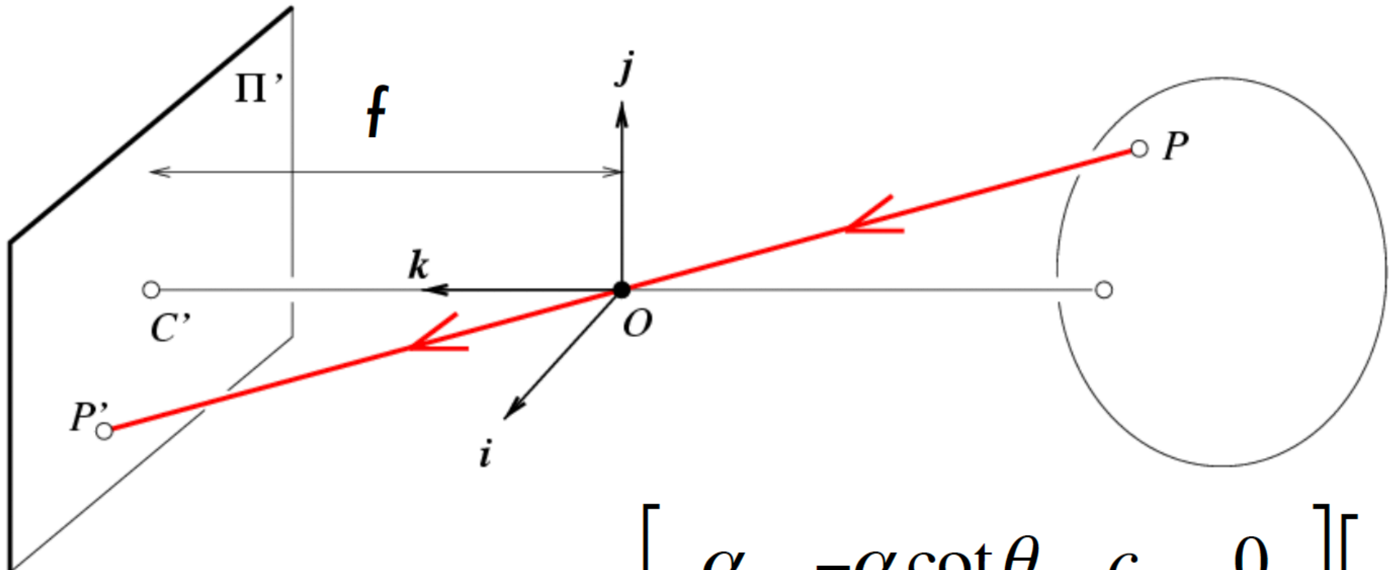
$$P_h' = \begin{bmatrix} \alpha x + c_x z \\ \beta y + c_y z \\ z \end{bmatrix} = \begin{bmatrix} \alpha & 0 & c_x & 0 \\ 0 & \beta & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \underbrace{\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}}_{P_h}$$

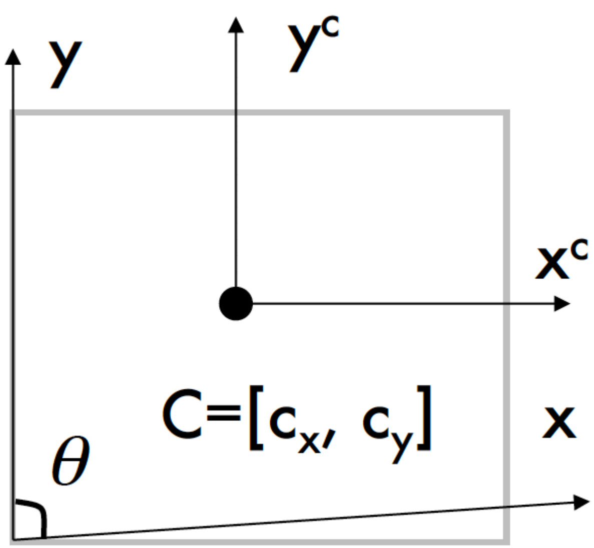
[Eq.8]

$$M = \begin{bmatrix} \alpha & 0 & c_x & 0 \\ 0 & \beta & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\overbrace{P_h'}^{\text{Homogenous}} \rightarrow \overbrace{P' = (\alpha \frac{x}{z} + c_x, \beta \frac{y}{z} + c_y)}^{\text{Euclidian}}$$

Camera Skewness

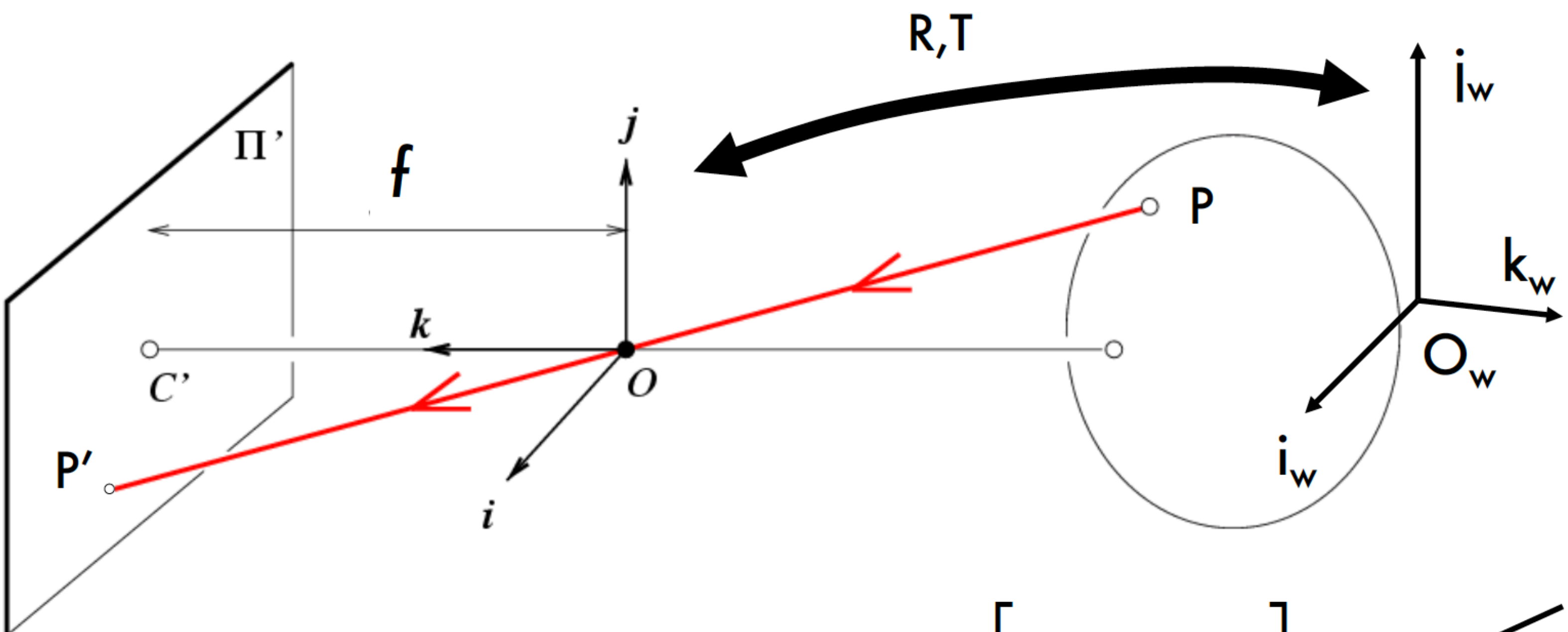




$$P' = \begin{bmatrix} \alpha & -\alpha \cot\theta & c_x & 0 \\ 0 & \dfrac{\beta}{\sin\theta} & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

How many degree does K have?
5 degrees of freedom!

World reference system



In 4D homogeneous coordinates: $P = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix}_{4\times4} P_w$ $\qquad P_w = \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}$

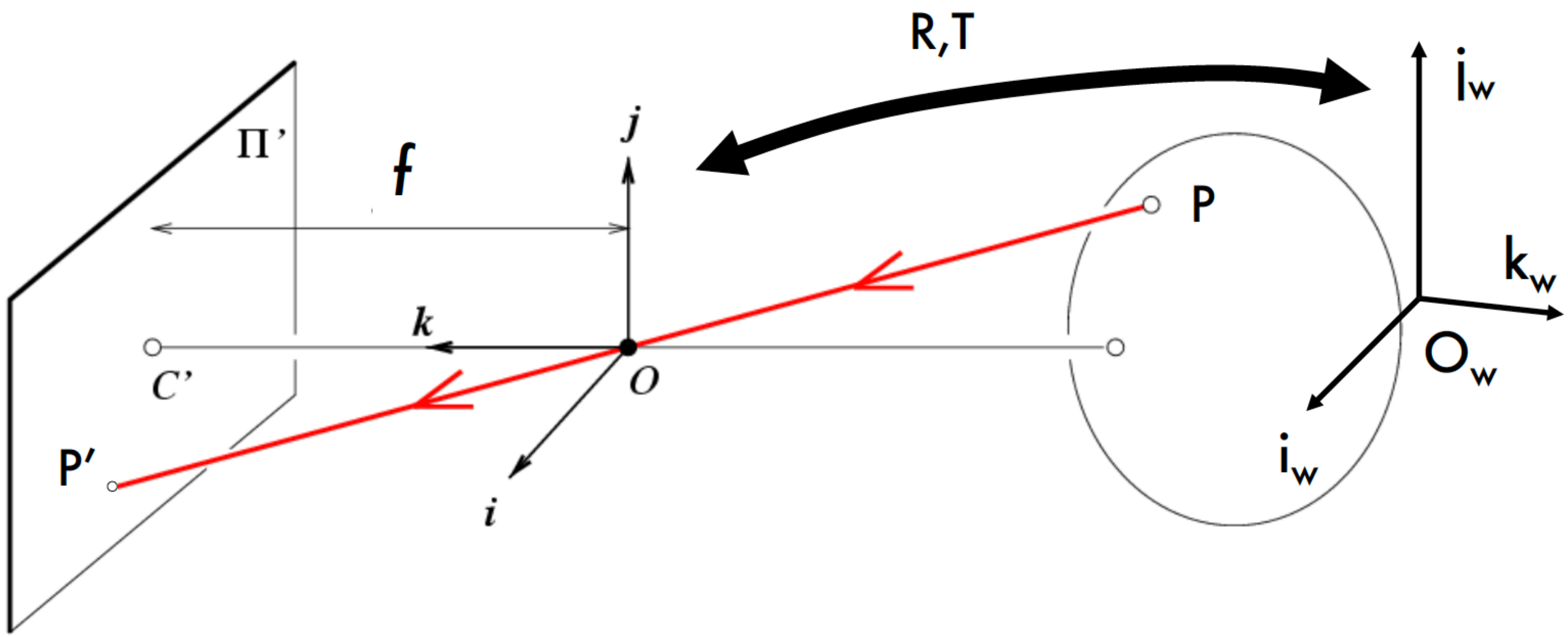
intrinsic parameters

extrinsic parameters

$$P' = K\begin{bmatrix} I & 0 \end{bmatrix} P = K\begin{bmatrix} I & 0 \end{bmatrix}\begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix}_{4\times4} P_w = \underbrace{\boxed{K\begin{bmatrix} R & T \end{bmatrix}}}_{M} P_w$$

[Eq.11]

The projective transformation



$$P'_{3\times1} = M_{3x4}\, P_w = K_{3\times3}\begin{bmatrix} R & T \end{bmatrix}_{3\times4} P_{w\,4\times1}$$

How many degrees of freedom?

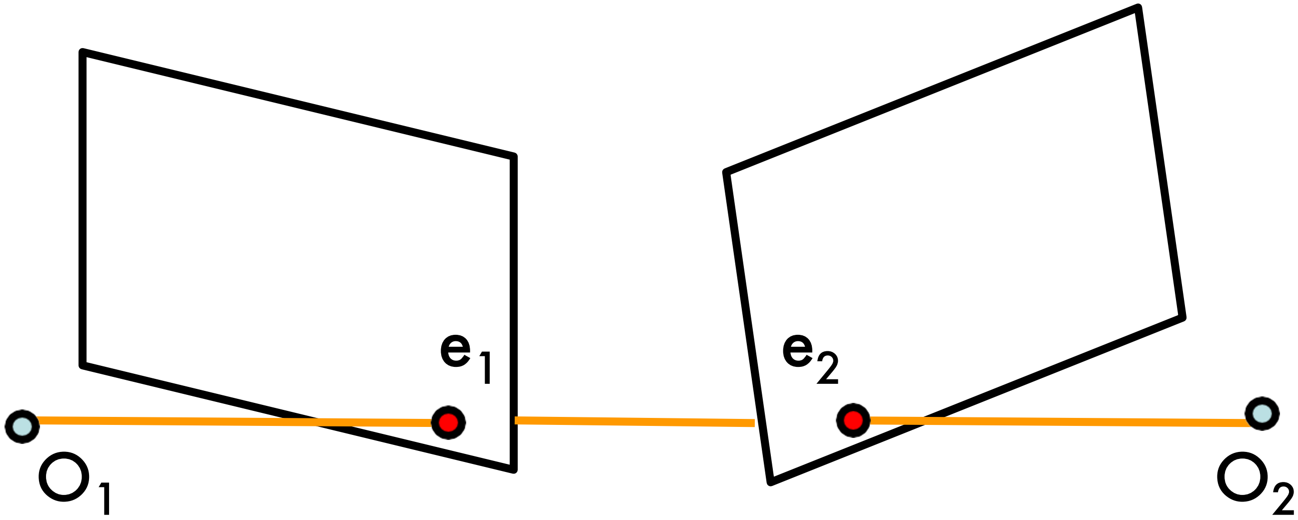
5 + 3 + 3 =11!

CSE 152: Computer Vision

Hao Su

Lecture 14: Multiview Geometry

Epipolar Geometry

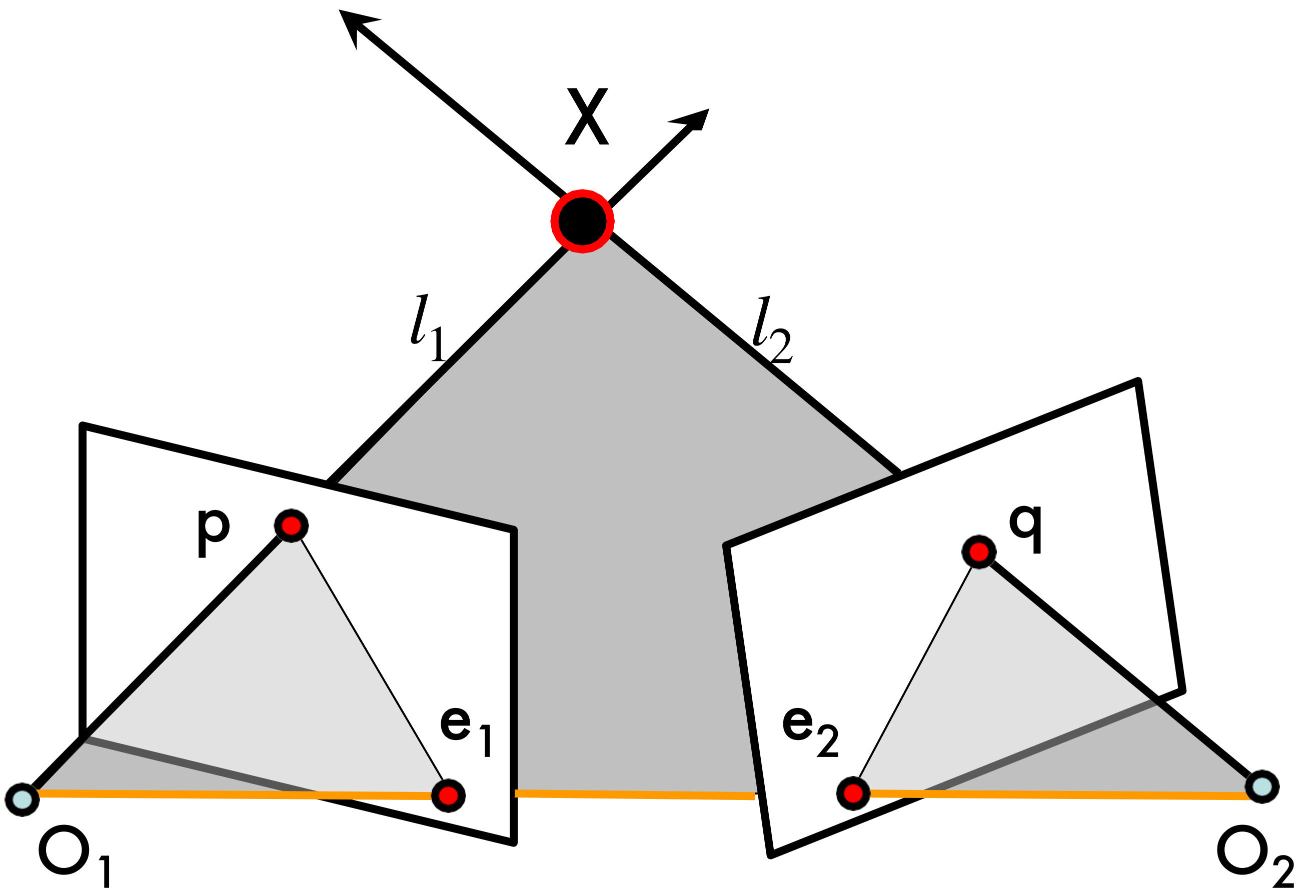


- Baselines
- Epipoles: e_1 , e_2

  = intersections of baseline with image planes

  = projections of the other camera center

Epipolar Geometry



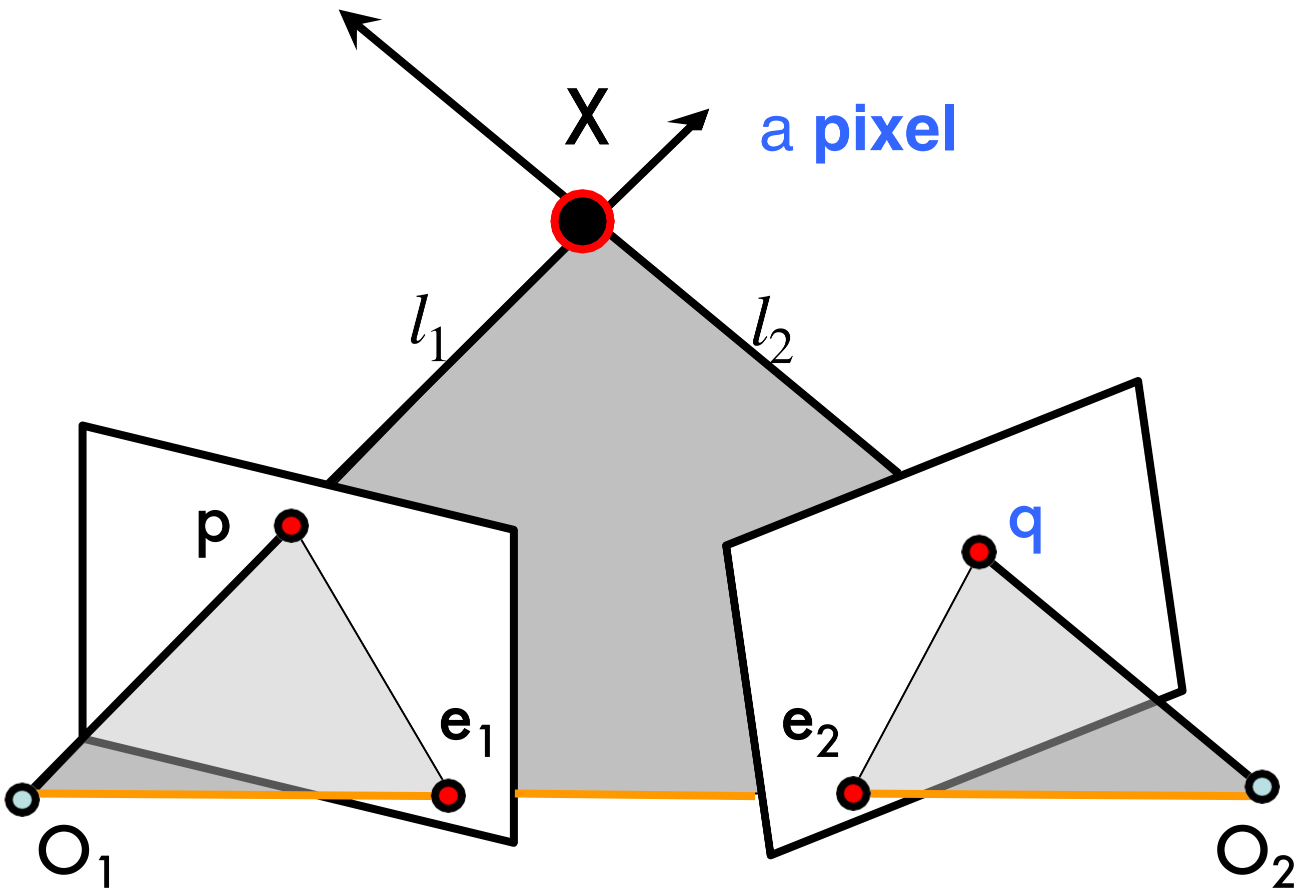
- Baselines
- Epipolar plane

- Epipoles: e_1 , e_2

  = intersections of baseline with image planes

  = projections of the other camera center

Epipolar Geometry

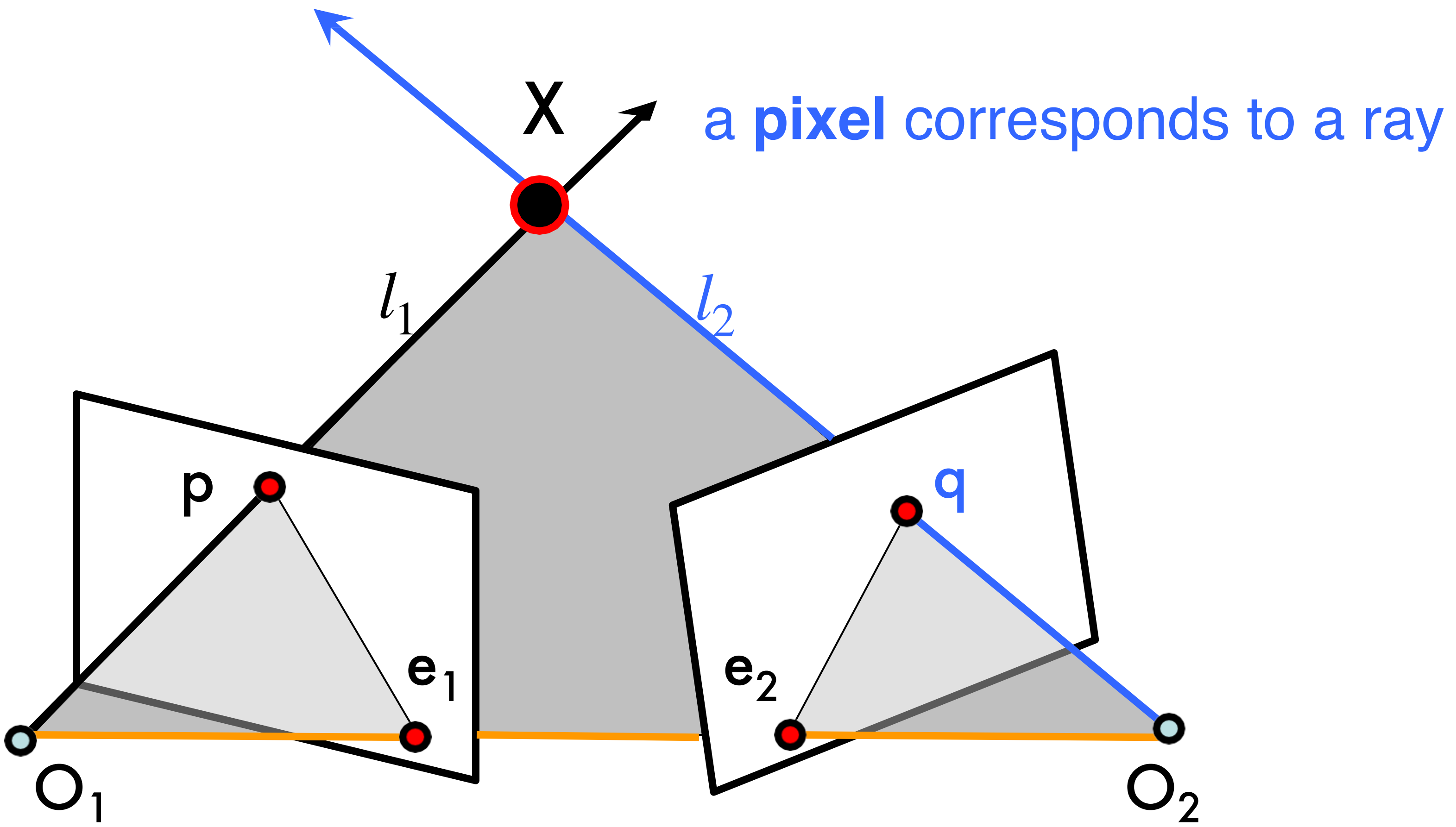


- Baselines
- Epipolar plane
- Epipoles: e_1 , e_2

  = intersections of baseline with image planes

  = projections of the other camera center

Epipolar Geometry



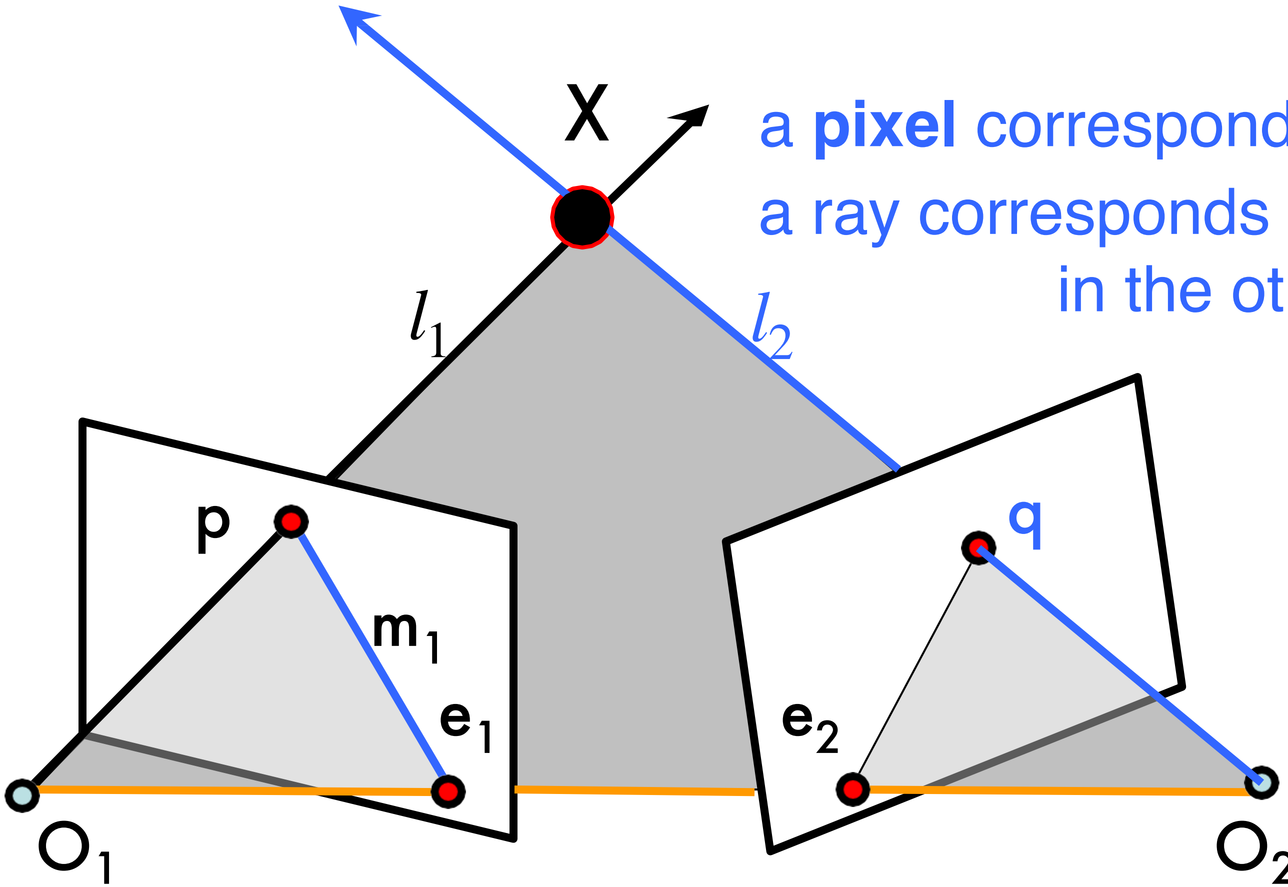
- Baselines
- Epipolar plane

- Epipoles: e_1 , e_2

  = intersections of baseline with image planes

  = projections of the other camera center

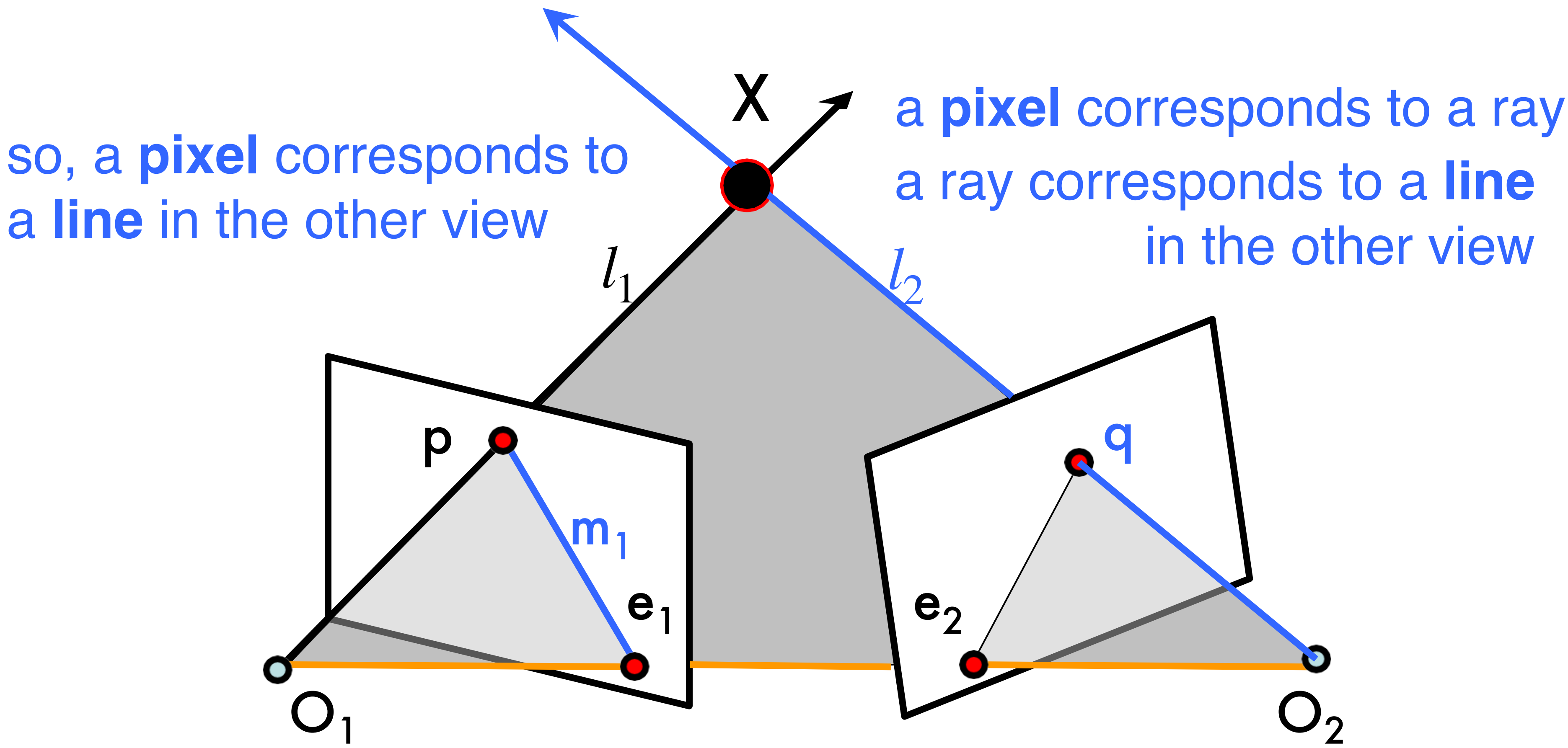
Epipolar Geometry



a **pixel** corresponds to a ray
a ray corresponds to a **line** in the other view

- Baselines
- Epipolar plane
- Epipoles: e_1 , e_2
  = intersections of baseline with image planes
  = projections of the other camera center

Epipolar Geometry



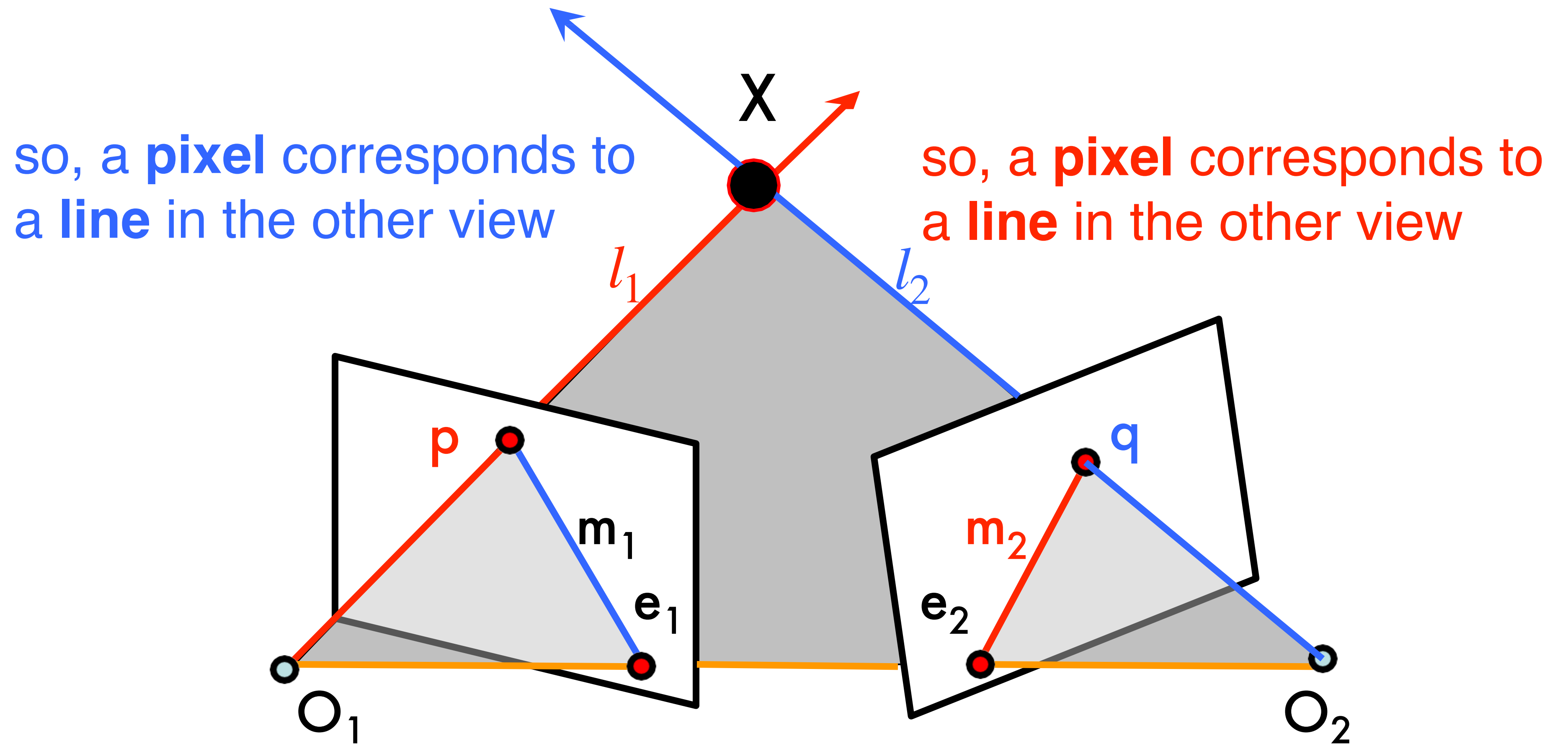
- Baselines
- Epipolar plane

- Epipoles: e_1 , e_2

  = intersections of baseline with image planes

  = projections of the other camera center

Epipolar Geometry



- Baselines
- Epipolar plane
- Epipolar line

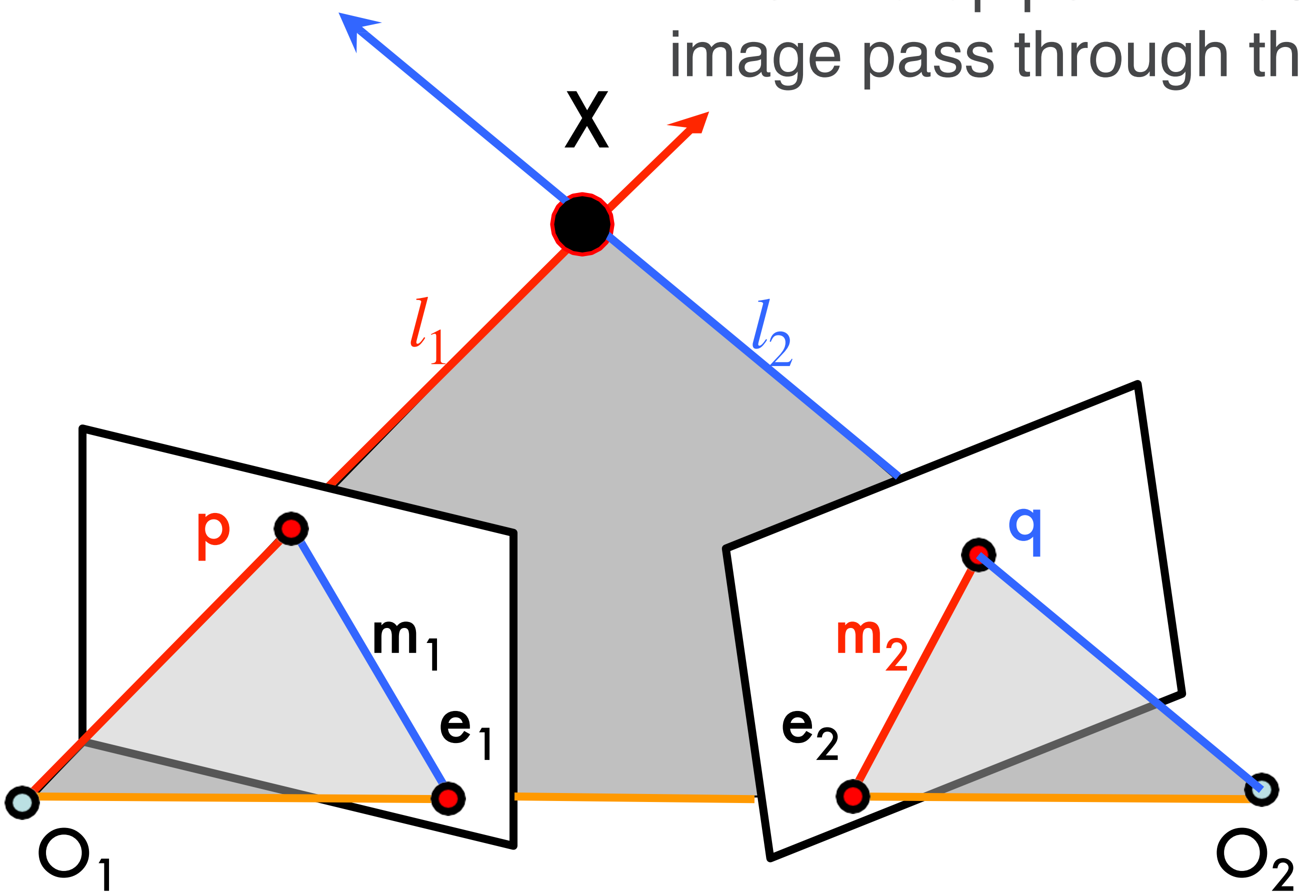
- Epipoles: e_1 , e_2

  = intersections of baseline with image planes

  = projections of the other camera center

Epipolar Geometry

All of the epipolar lines in an image pass through the epipole.



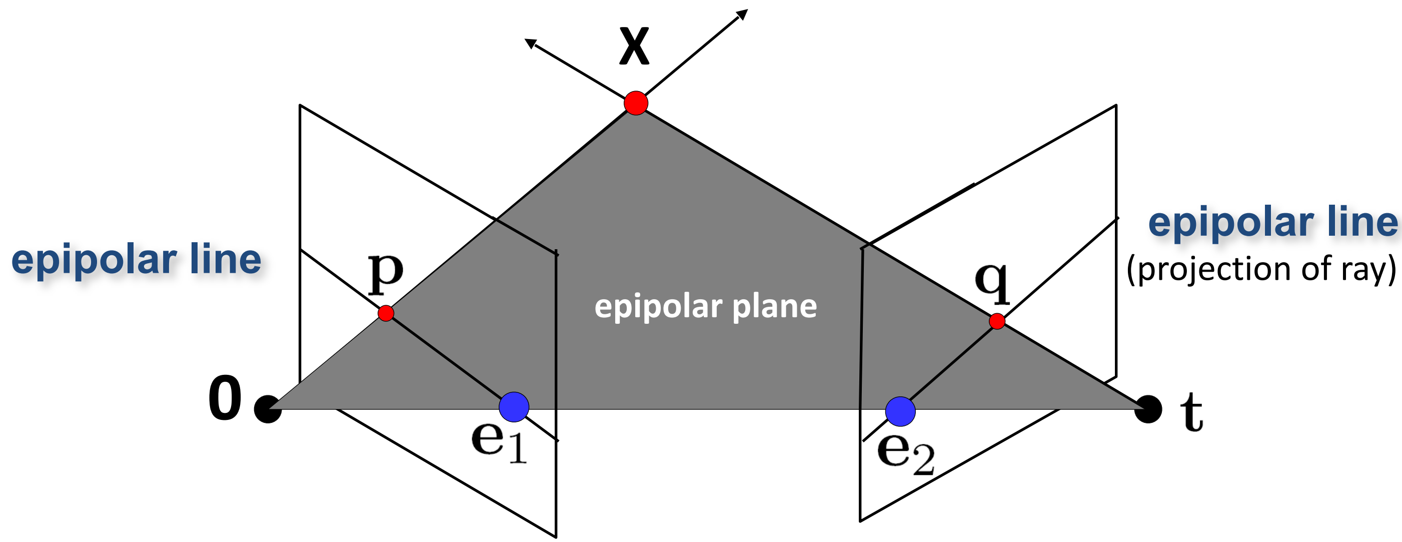
- Baselines
- Epipolar plane
- Epipolar line

- Epipoles: e_1 , e_2

  = intersections of baseline with image planes

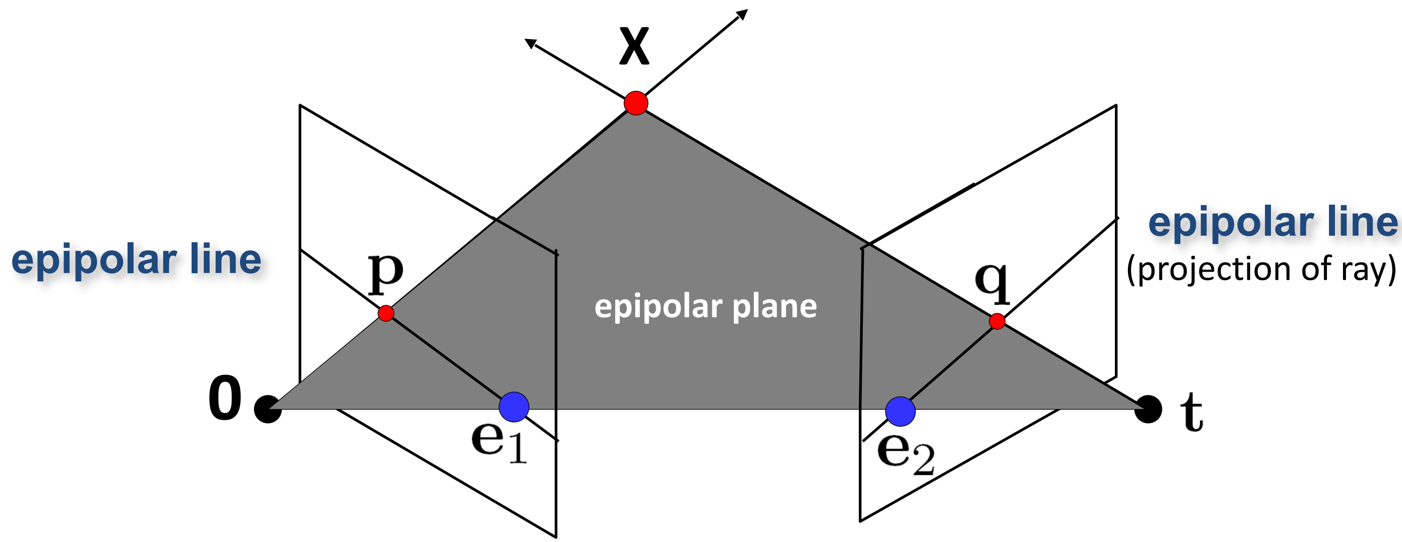
  = projections of the other camera center

Essential Matrix



- Assume p and q in $\mathbb{R}^3$ are two points on the (virtual) image plane of two cameras
- Denoted by the pinhole frame coordinate in the corresponding cameras

Essential Matrix



- We have: $(p^1)^T[t]_\times R q^2 = 0$
- Define: $\mathbf{E} = [\mathbf{t}]_\times \mathbf{R}$

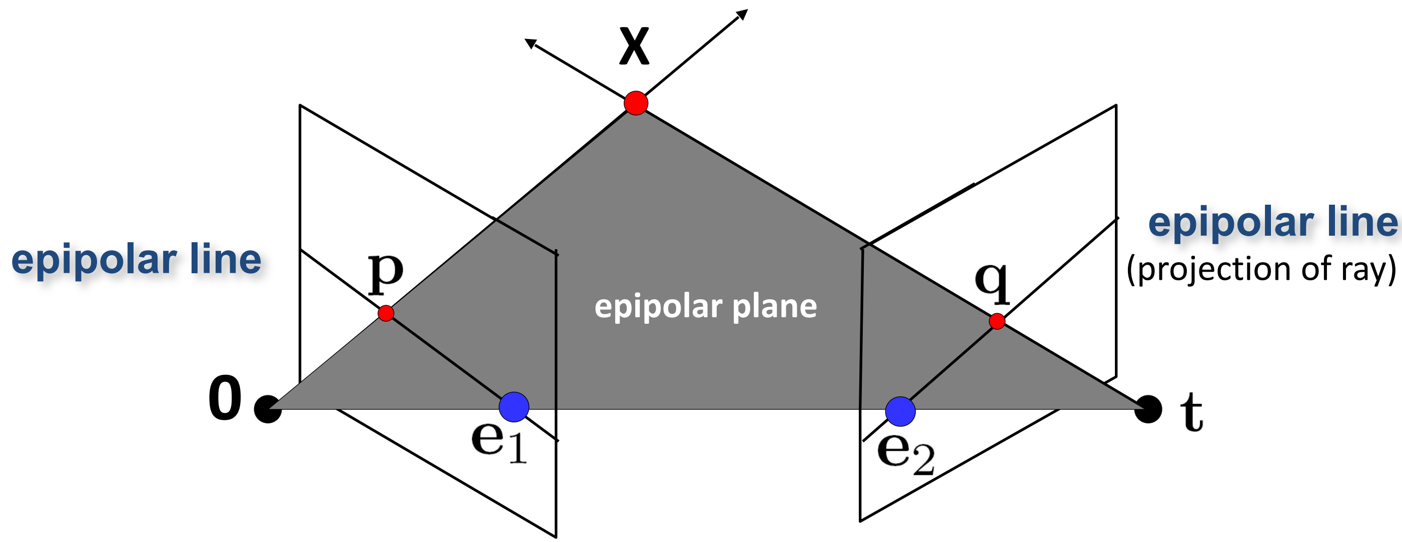
Essential matrix

- Then, we have:

rank(E)=2

$$(p^1)^T E q^2 = 0 \xrightarrow[\text{superscript}]{\text{omit}} p^T E q = 0$$

Fundamental Matrix



- Consider intrinsic camera matrices
- Then, **p** and **q** are in the pinhole frame and pixel counterparts are:

$$\mathbf{p}' = \mathbf{K}_1\mathbf{p} \qquad \mathbf{q}' = \mathbf{K}_2\mathbf{q}$$

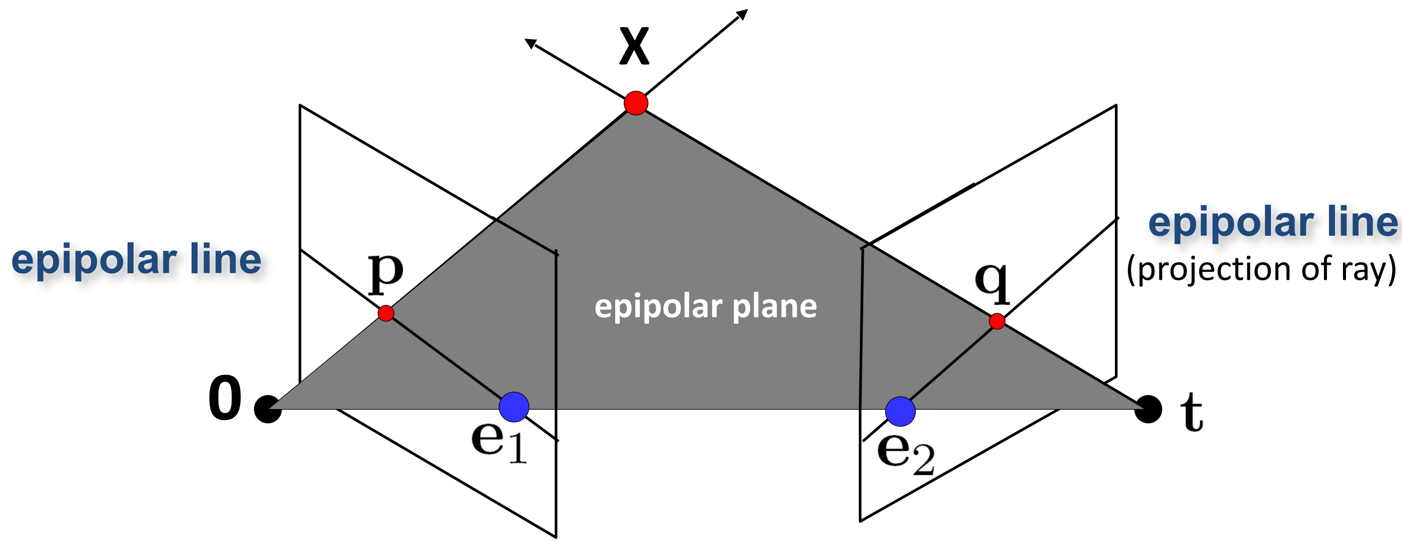
- Recall essential matrix constraint:

$$\boxed{p^T E q = 0}$$

- Substituting, we have:

$$(K_1^{-1}p')^T E(K_2^{-1}q') = 0$$

Fundamental Matrix



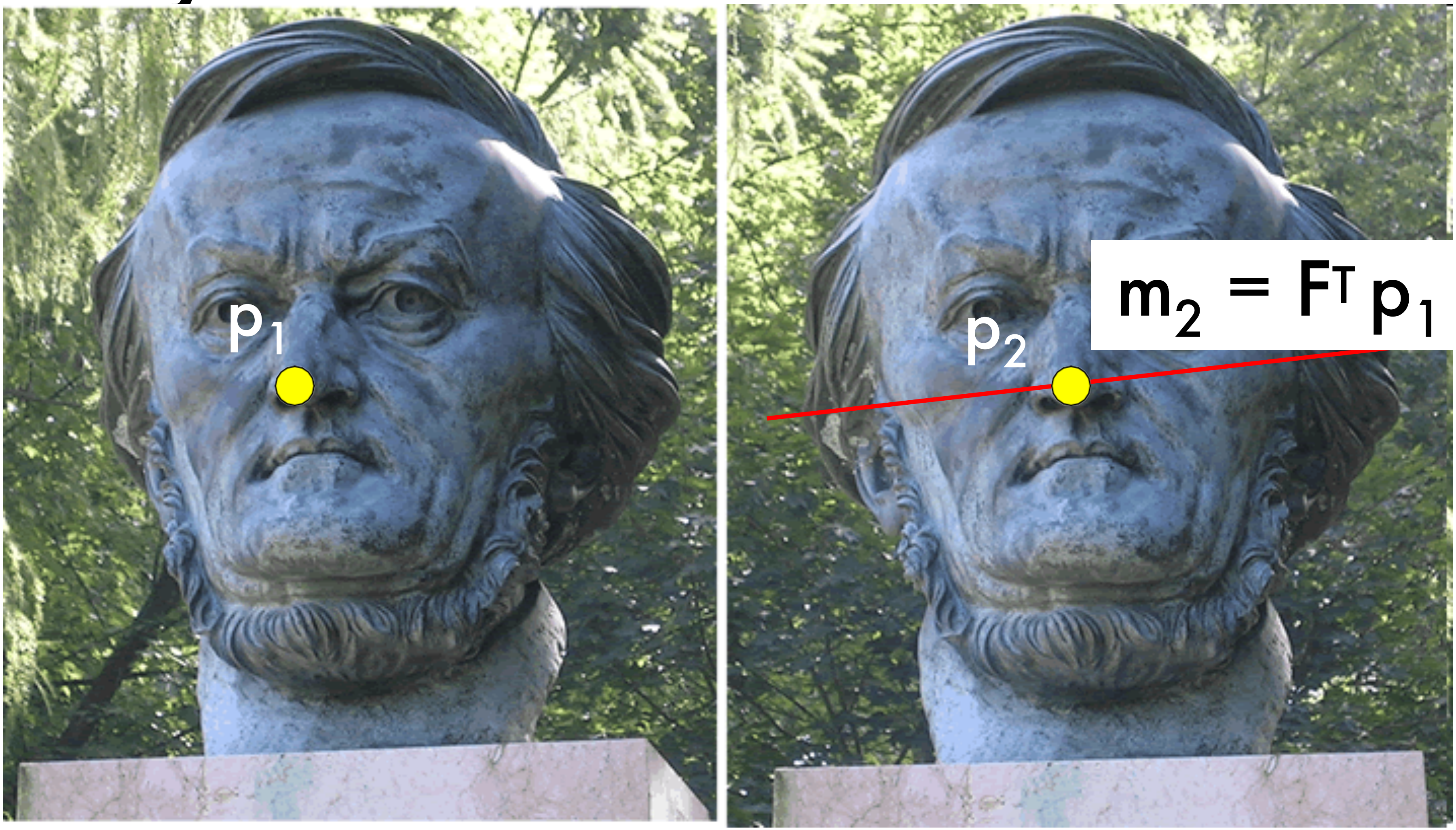
- Essential matrix constraint in pixel space: $(K_1^{-1}p')^T E(K_2^{-1}q').= 0$
- Rearranging: $p'^T K_1^{-T} E K_2^{-1} q' = 0$
- Define: $F = K_1^{-T} E K_2^{-1}$ Fundamental matrix rank(F)=2
- Then, we have: $\boxed{p'^T F q' = 0}$

Epipolar Constraint

$$p_1^T \cdot Fp_2 = 0$$

- $w_1 = Fp_2$ defines an equation $w_1^T p_1 = 0$
- Note that, p_1 is the corresponding point of p_2 by the derivation of F
- So, $w_1 = Fp_2$ defines the epipolar line of p_2

Why F is useful?



- Suppose F is known
- No additional information about the scene and camera is given
- Given a point on left image, we can compute the corresponding epipolar line in the second image

Estimating F

Suppose we have a pair of corresponding points:

$$p = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \quad p' = \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix}$$

[Eq. 13] $$\mathrm{p}^{\mathrm{T}}\,\mathrm{F}\,\mathrm{p}' = 0 \quad \Rightarrow$$

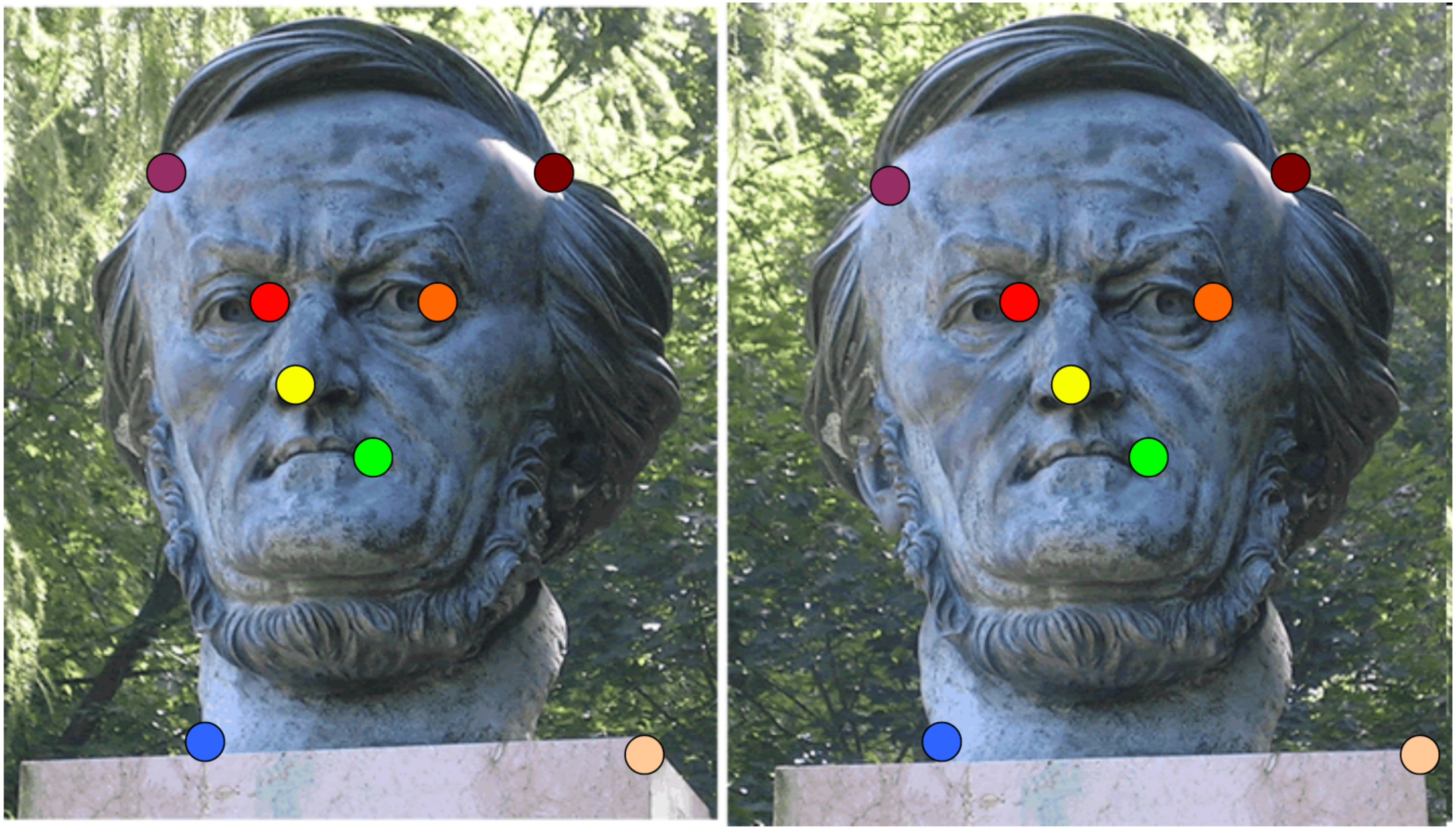
$$(u, v, 1)\begin{pmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{pmatrix}\begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} = 0$$

$$\Rightarrow \quad (uu', uv', u, vu', vv', v, u', v', 1)\begin{pmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{pmatrix} = 0$$

[Eq. 14]

Let's take 8 corresponding points

Estimating F



Estimating F

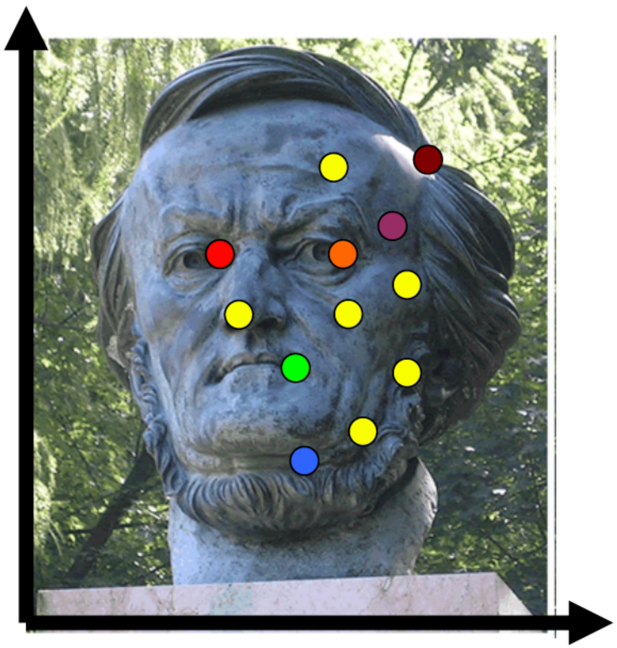
$$\mathbf{W} \underbrace{\begin{pmatrix} u_1u'_1 & u_1v'_1 & u_1 & v_1u'_1 & v_1v'_1 & v_1 & u'_1 & v'_1 & 1 \\ u_2u'_2 & u_2v'_2 & u_2 & v_2u'_2 & v_2v'_2 & v_2 & u'_2 & v'_2 & 1 \\ u_3u'_3 & u_3v'_3 & u_3 & v_3u'_3 & v_3v'_3 & v_3 & u'_3 & v'_3 & 1 \\ u_4u'_4 & u_4v'_4 & u_4 & v_4u'_4 & v_4v'_4 & v_4 & u'_4 & v'_4 & 1 \\ u_5u'_5 & u_5v'_5 & u_5 & v_5u'_5 & v_5v'_5 & v_5 & u'_5 & v'_5 & 1 \\ u_6u'_6 & u_6v'_6 & u_6 & v_6u'_6 & v_6v'_6 & v_6 & u'_6 & v'_6 & 1 \\ u_7u'_7 & u_7v'_7 & u_7 & v_7u'_7 & v_7v'_7 & v_7 & u'_7 & v'_7 & 1 \\ u_8u'_8 & u_8v'_8 & u_8 & v_8u'_8 & v_8v'_8 & v_8 & u'_8 & v'_8 & 1 \end{pmatrix}}_{} \underbrace{\begin{pmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{pmatrix}}_{\mathbf{f}} = 0$$

[Eqs. 15]

- Homogeneous system $\mathbf{W}\,\mathbf{f} = 0$
- Rank 8 ⟶ A non-zero solution exists (unique)
- If N>8 ⟶ Lsq. solution by SVD! ⟶ $\hat{\mathrm{F}}$

  $\|\mathbf{f}\| = 1$

Flow of the 8-Point Algorithm



$$Wf = 0,\ \|f\| = 1$$

↓ Least-square

$$\underset{f}{\textbf{minimize}} \quad \|Wf\|^2 \qquad s.t. \quad \|f\| = 1$$

↓

$$\underset{f}{\textbf{minimize}} \quad f^T W^T W f \qquad s.t. \quad f^T f = 1$$

↓

Do you remember how to solve the problem?
Hint: Check your HW1 (by the SVD of W)

Find F that minimizes $\|F - \hat{F}\|$ Frobenius norm (*)

Subject to det(F)=0

$$F = U \begin{bmatrix} s_1 & 0 & 0 \\ 0 & s_2 & 0 \\ 0 & 0 & 0 \end{bmatrix} V^T$$

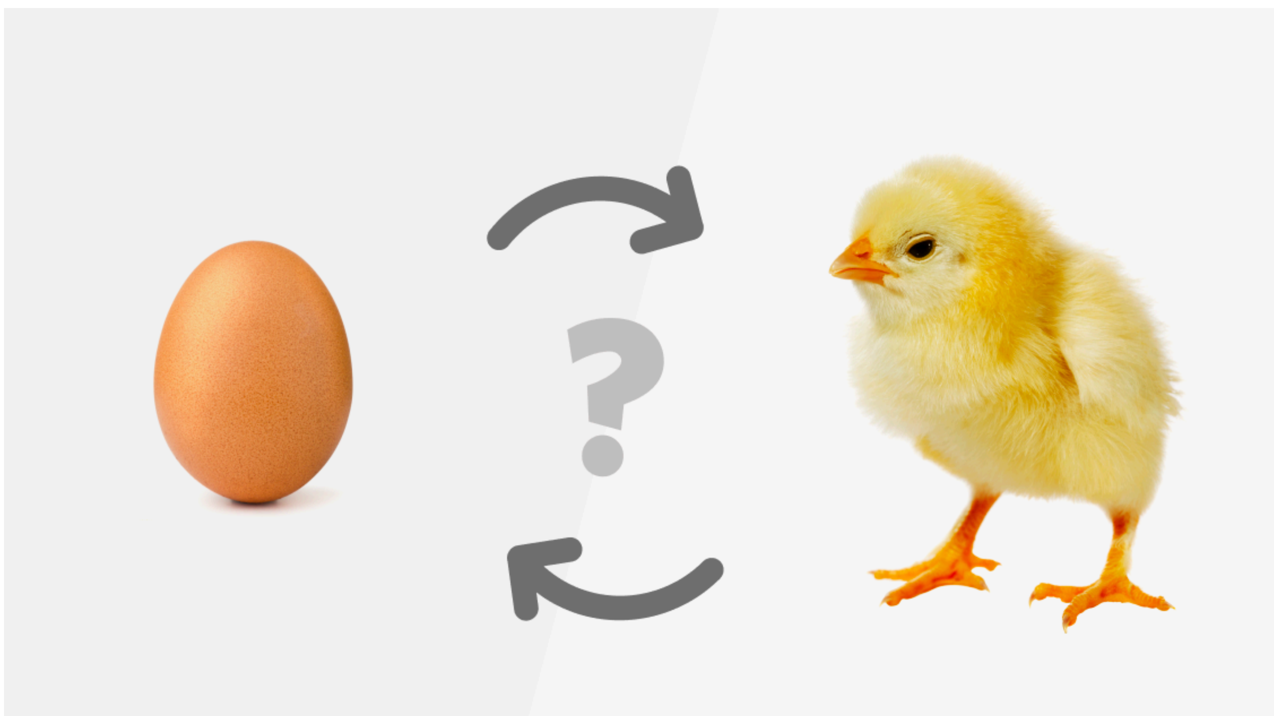
Where:

$$U \begin{bmatrix} s_1 & 0 & 0 \\ 0 & s_2 & 0 \\ 0 & 0 & s_3 \end{bmatrix} V^T = SVD(\hat{F})$$

[HZ] pag 281, chapter 11, "Computation of F"

Simultaneous Correspondence and F Estimation

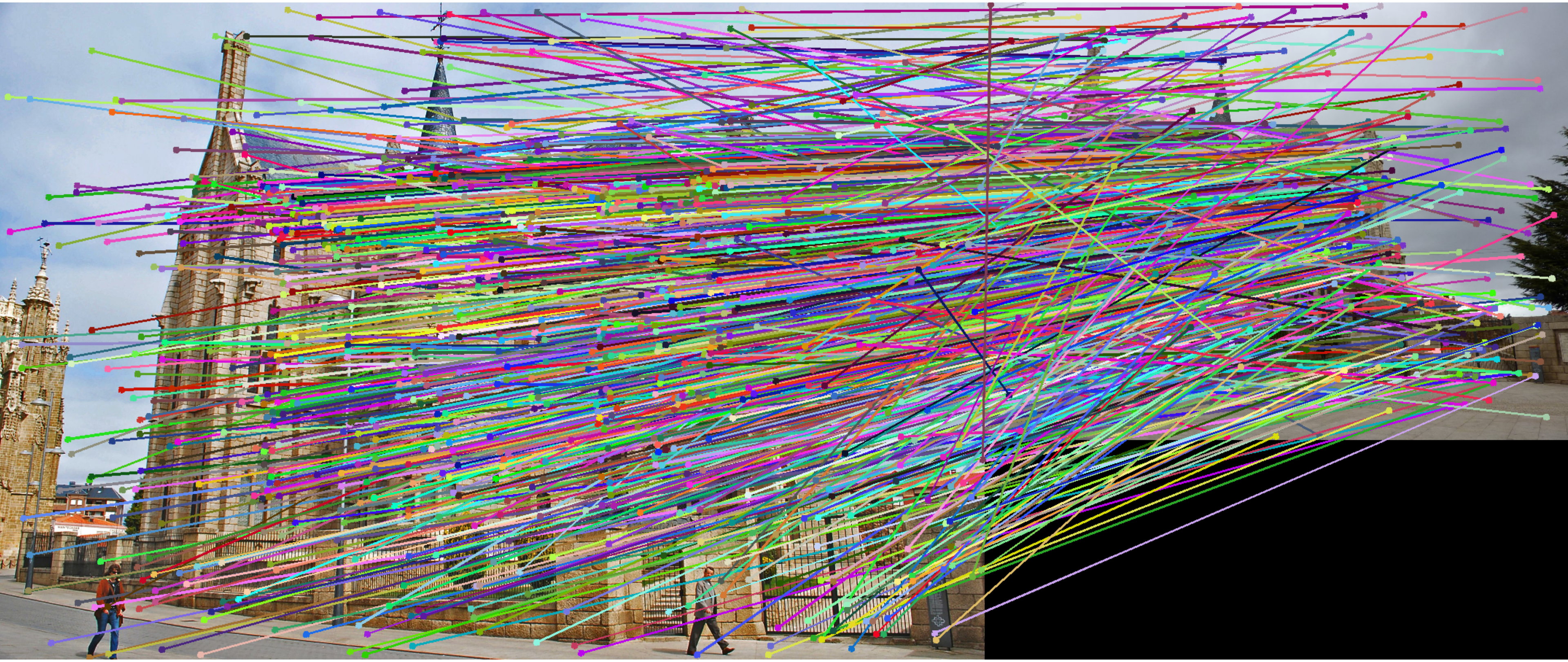
- ***Simultaneous*** Correspondence and F Estimation
  - With F, it is easier to compute correspondence
  - With correspondence, we can estimate F
- A Chicken-or-Egg problem



Basic Pipeline: Feature detection

Detect features using, for example, SIFT [Lowe, IJCV 2004] or learning-based keypoint detector

VLFeat's 800 most confident matches among 10,000+ local features.



Basic Pipeline: Repeat the above steps (RANSAC)

```
for i in range(n):
        randomly choose some pairs
        repeat for m times:
                based on the inliers, estimate F
                based on F, remove pairs with big errors
```

CSE 152: Computer Vision

Hao Su

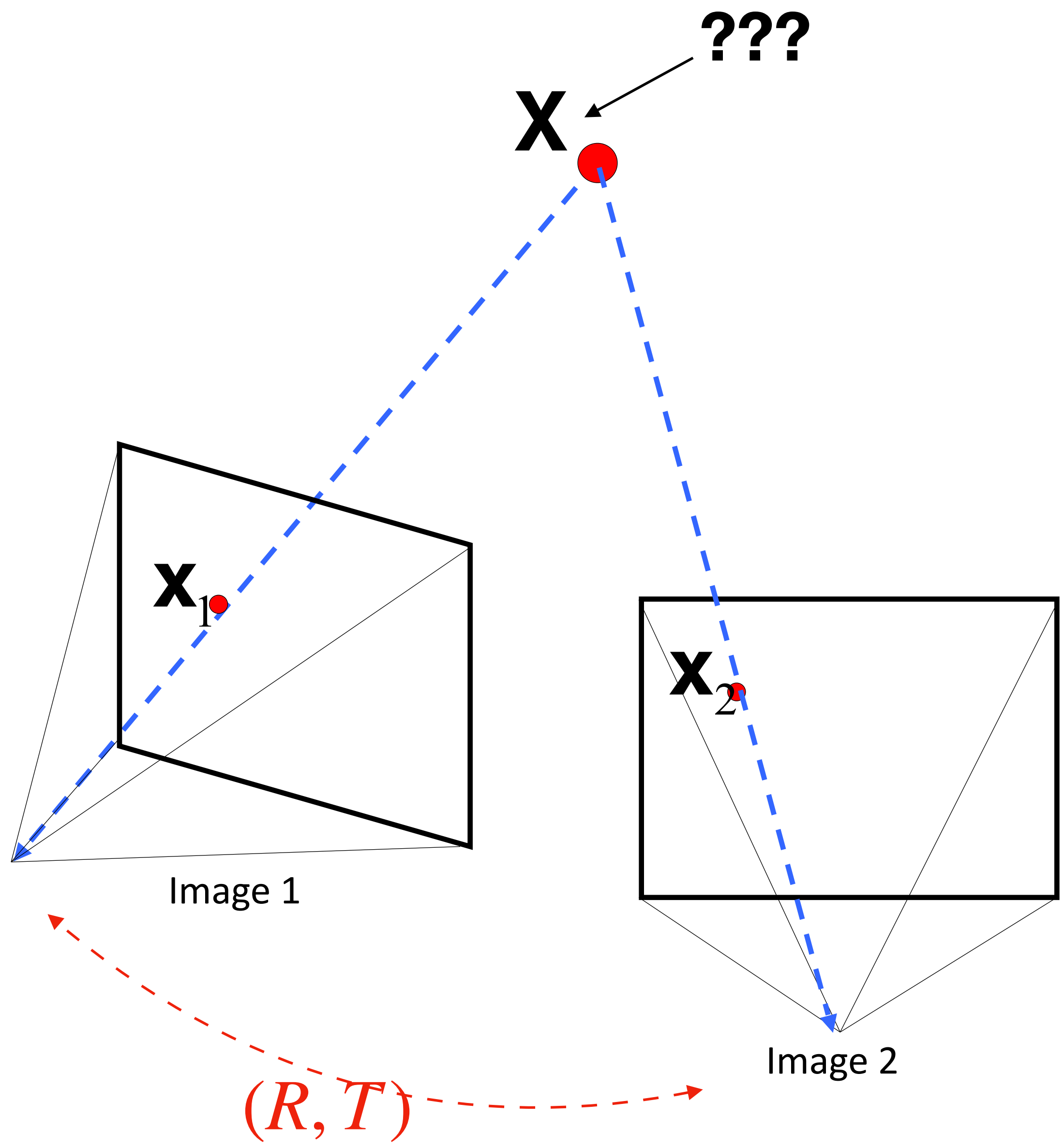
Lecture 16: Stereo Reconstruction

Problem setup

- Known:
  - Two views of the same scene
  - Corresponding points between views
  - Intrinsic camera matrices (K_1, K_2), i.e., camera calibration has been done
  - Fundamental matrix F

- Question: Point coordinates in 3D space

Step I: Estimate (R,T) Between Cameras



$\mathbf{x}_1 \leftrightarrow \mathbf{x}_2$

$\mathbf{x}_1^T \mathbf{F} \mathbf{x}_2 = 0$

Step I: Estimate (R,T) Between Cameras

- Get E from F:

$$F = K_1^{-T} E K_2^{-1}$$
$$E = K_1^{T} E K_2$$

- Decompose E into skew-symmetric and rotation matrices:

$$\mathbf{E} = [\mathbf{t}]_\times \mathbf{R}$$

Step II: Reprojection Error Minimization

- With K_i, R, t, we can compute the projection matrices for both cameras:

$$P' = M\,P_w = \underbrace{\boxed{K}}_{\text{Internal parameters}}\underbrace{\boxed{[R \quad T]}}_{\text{External parameters}}P_w$$

- The projective projection equation:

$$\boxed{p_i = \begin{bmatrix} u_i \\ v_i \end{bmatrix} = \begin{bmatrix} \dfrac{\mathbf{m}_1 P_i}{\mathbf{m}_3 P_i} \\ \dfrac{\mathbf{m}_2 P_i}{\mathbf{m}_3 P_i} \end{bmatrix}} \qquad M = \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{bmatrix}$$

in pixels

A non-linear transformation, denoted by $p_i = f(P_i)$

Step II: Reprojection Error Minimization

- Minimize sum of squared reproduction errors:

$$\underset{\{P_i\}}{\textbf{minimize}} \quad \sum_{i=1}^{n} \sum_{j=1}^{2} \underbrace{w_{ij}}_{\substack{\textit{indicator variable}:\\ \text{whether point } i \text{ visible in image } j}} \left\| \underbrace{f(P_i; R_j, t_j)}_{\substack{\textit{predicted}\\ \text{image location}}} - \underbrace{\begin{bmatrix} u_{i,j} \\ v_{i,j} \end{bmatrix}}_{\substack{\textit{observed}\\ \text{image location}}} \right\|^2$$

- Optimized with non-linear least squares
- LM algorithm (Levenberg-Marquardt) is a popular choice

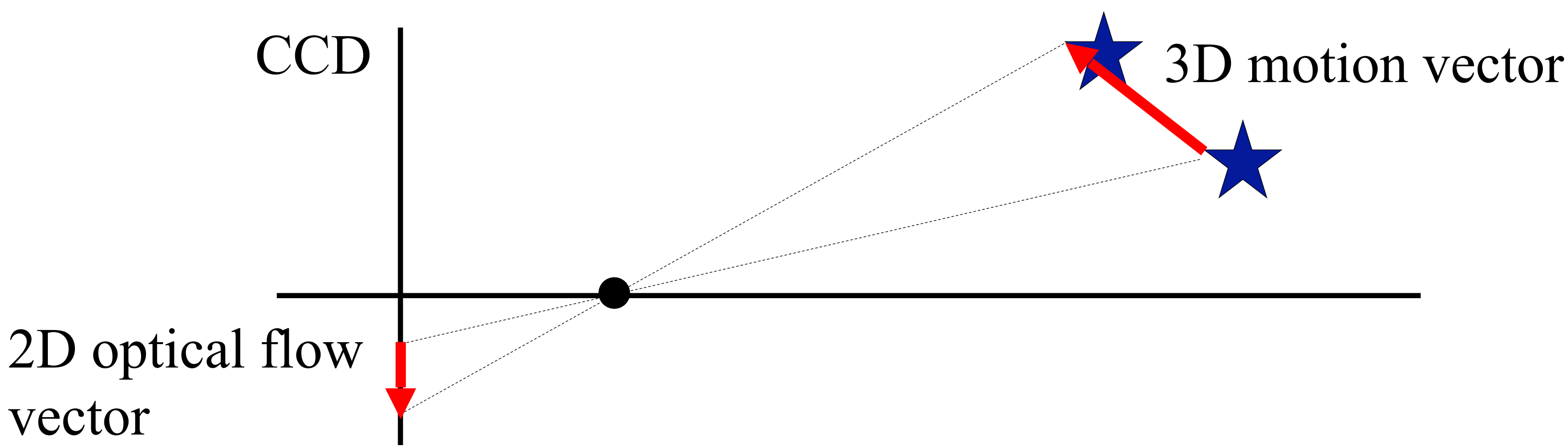
CSE 152: Computer Vision

Hao Su

Lecture 17: Motion Estimation

Motion Field & Optical Flow Field

- Motion Field = Real world 3D motion
- Optical Flow Field = Projection of the motion field onto the 2d image

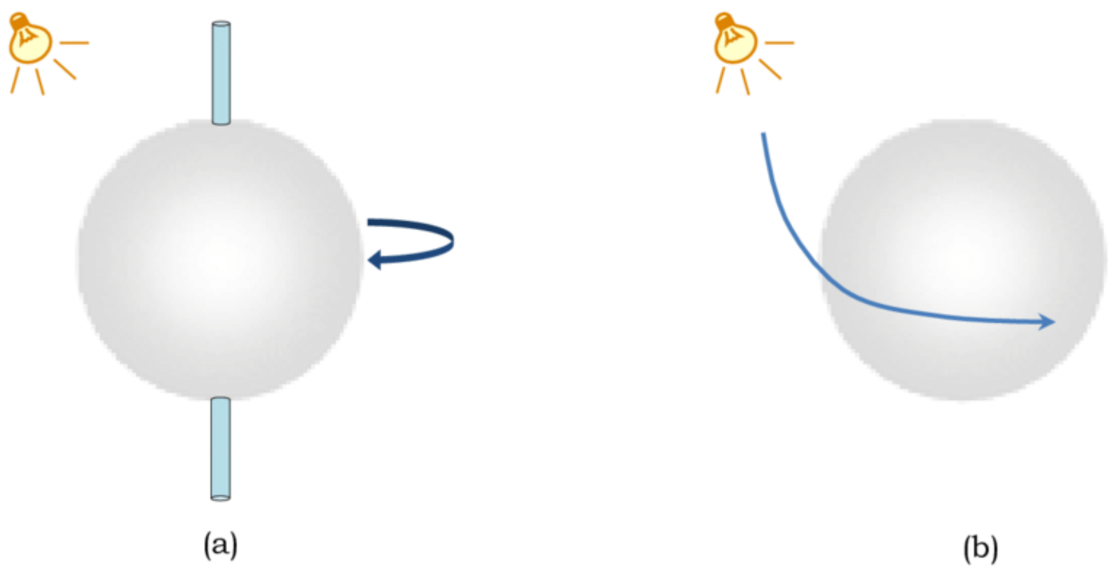


$$\vec{\mathrm{u}} = (u, v)$$

Slide adapted from Savarese.

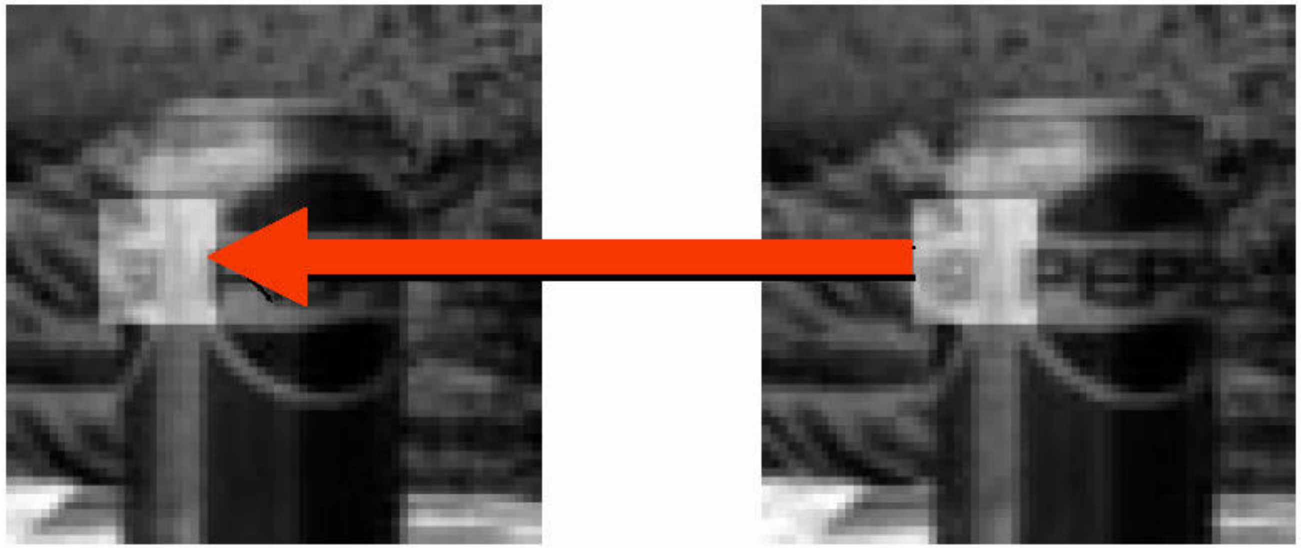
Apparent motion

- Optical flow differs from actual motion field:
  - (a) intensity remains constant, so that no motion is perceived;
  - (b) no object motion exists, however moving light source produces shading changes.



Source: Silvio Savarese

Key Assumptions: Brightness Constancy



Assumption

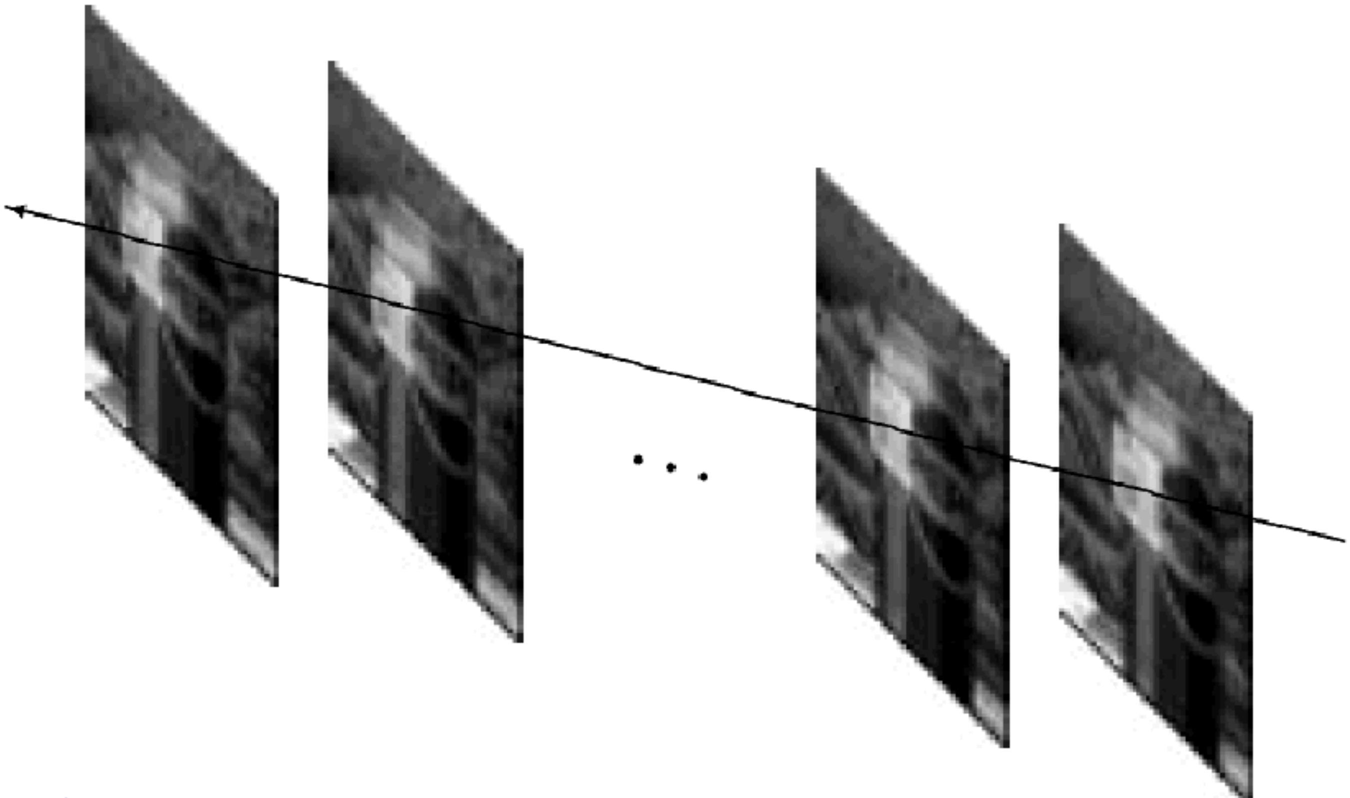
Image measurements (e.g. brightness) in a small region remain the same although their location may change.

$$I(x+u, y+v, t+1) = I(x, y, t)$$

(assumption)

* Slide from Michael Black, CS143 2003

Key Assumptions: Small Motions

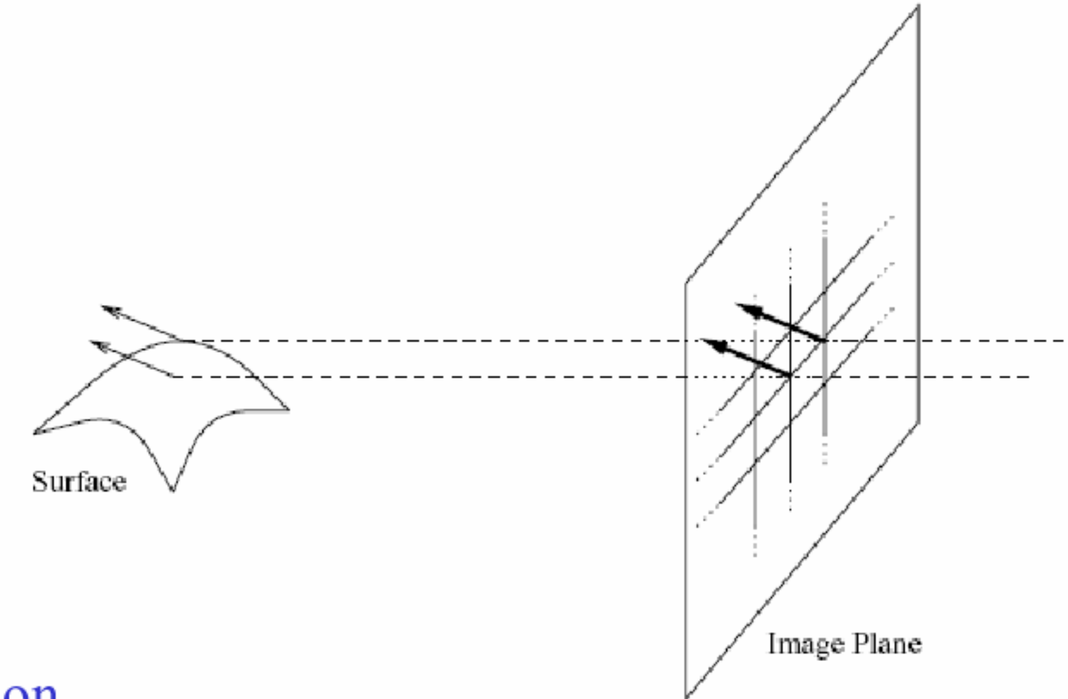


Assumption:

The image motion of a surface patch changes gradually over time.

* Slide from Michael Black, CS143 2003

Key Assumptions: Spatial Coherence

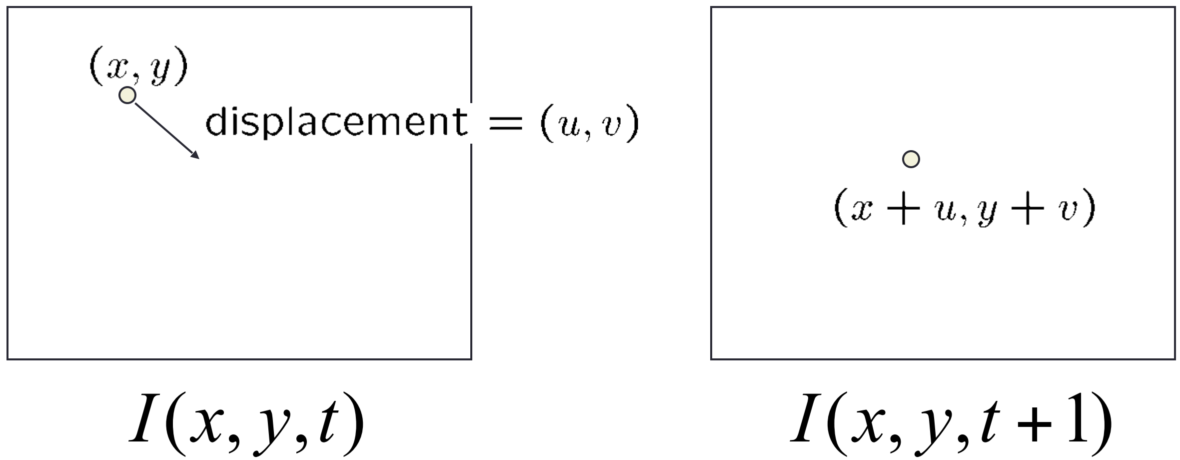


Assumption

* Neighboring points in the scene typically belong to the same surface and hence typically have similar motions.
* Since they also project to nearby points in the image, we expect spatial coherence in image flow.

* Slide from Michael Black, CS143 2003

Optical Flow Constraints (grayscale images)



- Let's look at these constraints more closely
  - Brightness constancy constraint (equation)

    $$I(x, y, t) = I(x+u, y+v, t+1)$$

  - Small motion: (u and v are less than 1 pixel, or smoothly varying)
    Taylor series expansion of I:

$$I(x+u, y+v, t+1) = I(x, y, t) + \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \frac{\partial I}{\partial t} + o(1)$$

The Brightness Constancy Constraint

Can we use this equation to recover image motion (u,v) at each pixel?

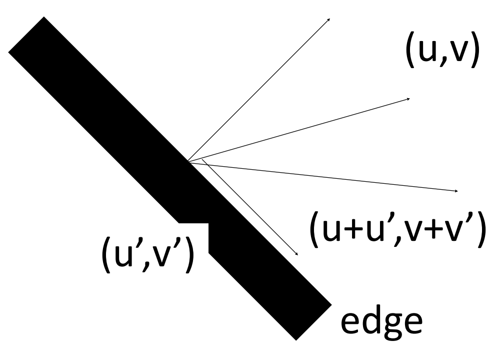
$$0 = I_t + \nabla I \cdot < u, v >$$

- How many equations and unknowns per pixel?
  - One equation (this is a scalar equation!), two unknowns (u,v)

The component of the flow perpendicular to the gradient (i.e., parallel to the edge) cannot be measured

If (u, v) satisfies the equation, so does (u+u', v+v') if

$$\nabla I \cdot \begin{bmatrix} u' & v' \end{bmatrix}^T = 0$$



Source: Silvio Savarese

Solving the Ambiguity…

B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 674–679, 1981.

- How to get more equations for a pixel?
- **Spatial coherence constraint**

  Assume the pixel's neighbors have the same (u,v)

  If we use a 5x5 window, that gives us 25 equations per pixel

$$0 = I_t(\mathbf{p_i}) + \nabla I(\mathbf{p_i}) \cdot [u \;\; v]$$

$$\begin{bmatrix} I_x(\mathbf{p_1}) & I_y(\mathbf{p_1}) \\ I_x(\mathbf{p_2}) & I_y(\mathbf{p_2}) \\ \vdots & \vdots \\ I_x(\mathbf{p_{25}}) & I_y(\mathbf{p_{25}}) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(\mathbf{p_1}) \\ I_t(\mathbf{p_2}) \\ \vdots \\ I_t(\mathbf{p_{25}}) \end{bmatrix}$$

Matching Matches Across Images

- Overconstrained linear system

$$\begin{bmatrix} I_x(\mathbf{p}_1) & I_y(\mathbf{p}_1) \\ I_x(\mathbf{p}_2) & I_y(\mathbf{p}_2) \\ \vdots & \vdots \\ I_x(\mathbf{p}_{25}) & I_y(\mathbf{p}_{25}) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(\mathbf{p}_1) \\ I_t(\mathbf{p}_2) \\ \vdots \\ I_t(\mathbf{p}_{25}) \end{bmatrix} \qquad \underset{25\text{x}2}{A} \ \underset{2\text{x}1}{d} = \underset{25\text{x}1}{b}$$

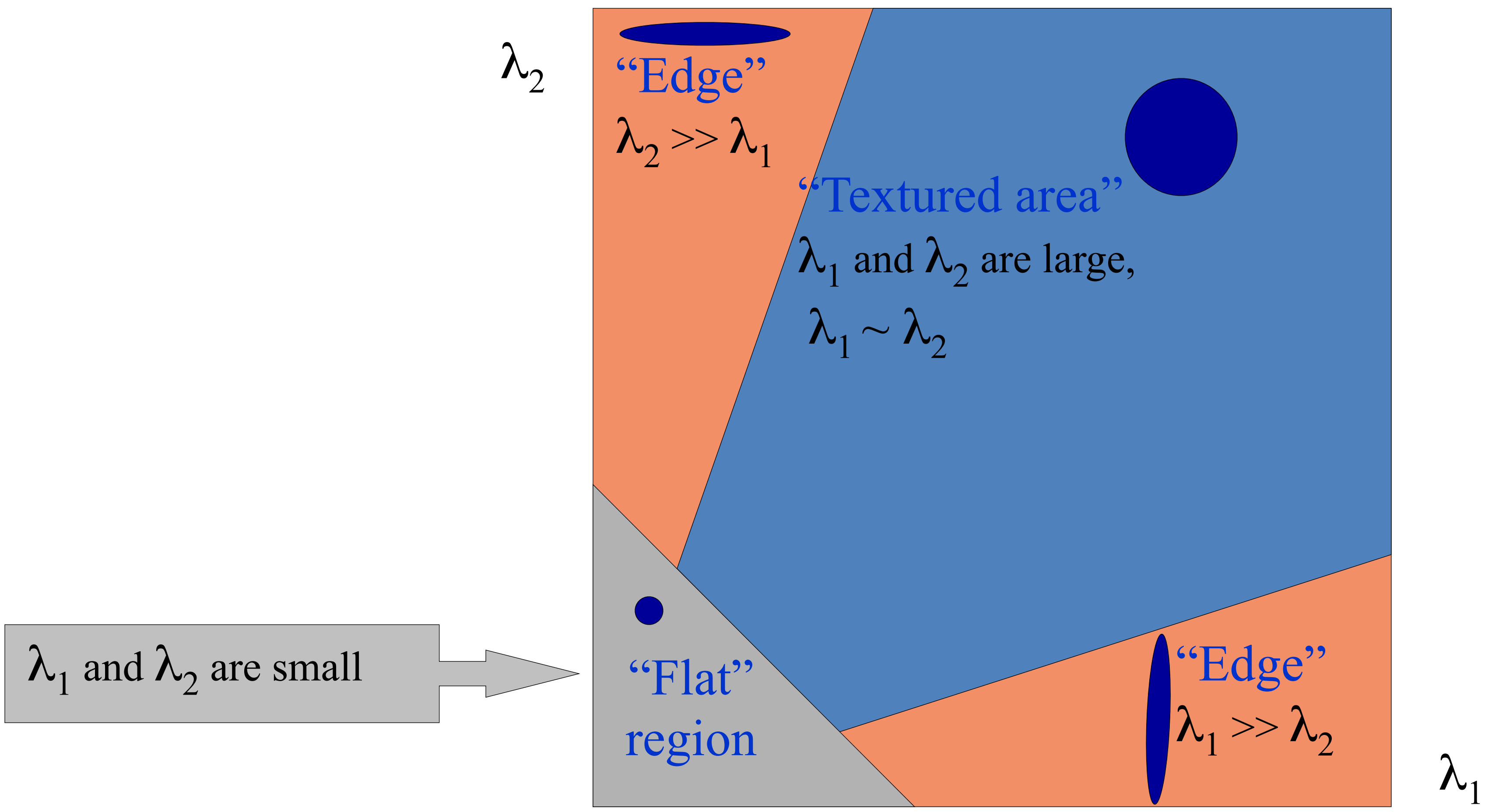
Least squares solution for *d* given by $(A^T A)\ d = A^T b$

$$A = \begin{bmatrix} (\nabla I(p_1))^T \\ \vdots \\ (\nabla I(p_n))^T \end{bmatrix} \Rightarrow A^T A = \sum_i \nabla I(p_i)(\nabla I(p_i))^T = \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix}$$

The summations are over all pixels in the K x K window

Interpreting the Eigenvalues

Classification of image points using eigenvalues of $A^T A$



Harris Corner Detector

Cornerness

$$C = \lambda_1\lambda_2 - \alpha(\lambda_1 + \lambda_2)^2$$

α: **constant (0.04 to 0.06)**