Lecture 6
Stereo Systems
Multi-view geometry

Professor Silvio Savarese

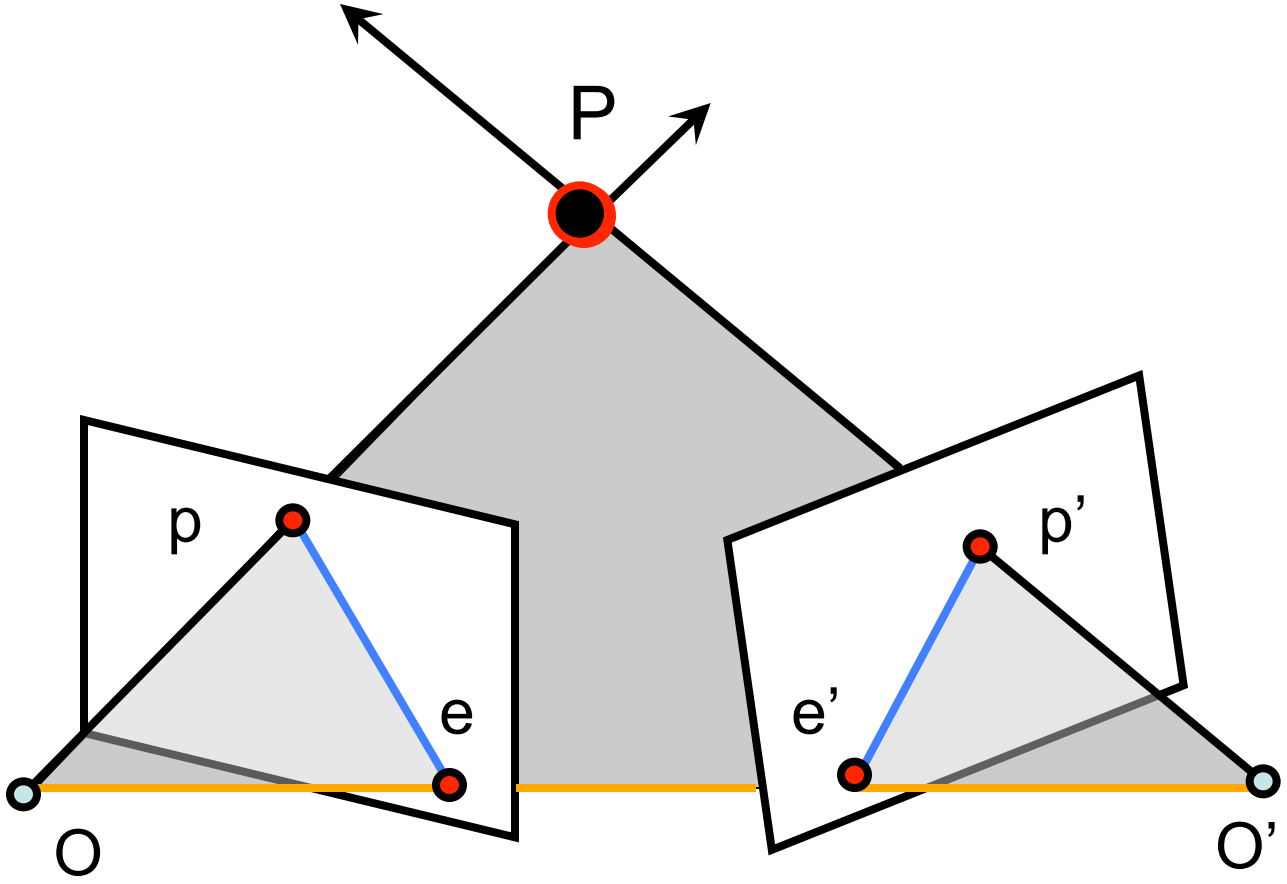
Computational Vision and Geometry Lab

Lecture 6
Stereo Systems
Multi-view geometry

- Stereo systems
  - Rectification
  - Correspondence problem
- Multi-view geometry
  - The SFM problem
  - Affine SFM

Reading:
[AZ] Chapter: 9 "Epip. Geom. and the Fundam. Matrix Transf."
[AZ] Chapter: 18 "N view computational methods"
[FP] Chapters: 7 "Stereopsis"
[FP] Chapters: 8 "Structure from Motion"

Epipolar geometry

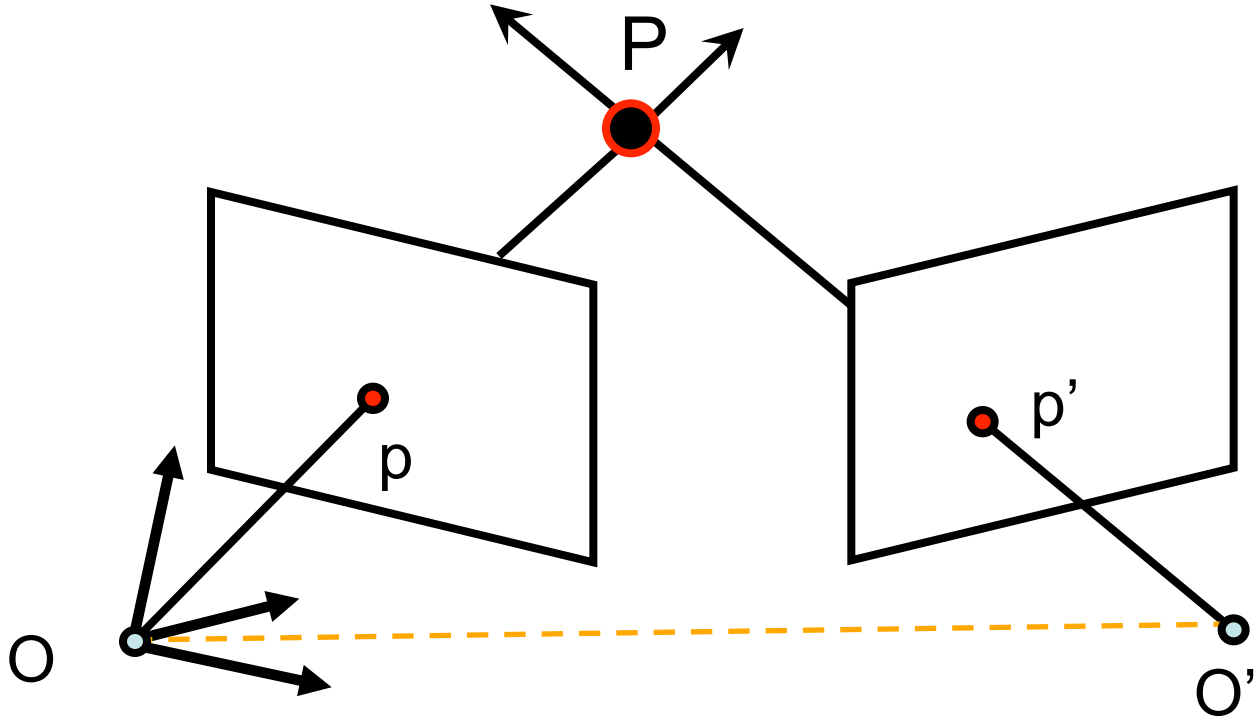


- Epipolar Plane
- Baseline
- Epipolar Lines

- Epipoles e, e’

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

Epipolar Constraint



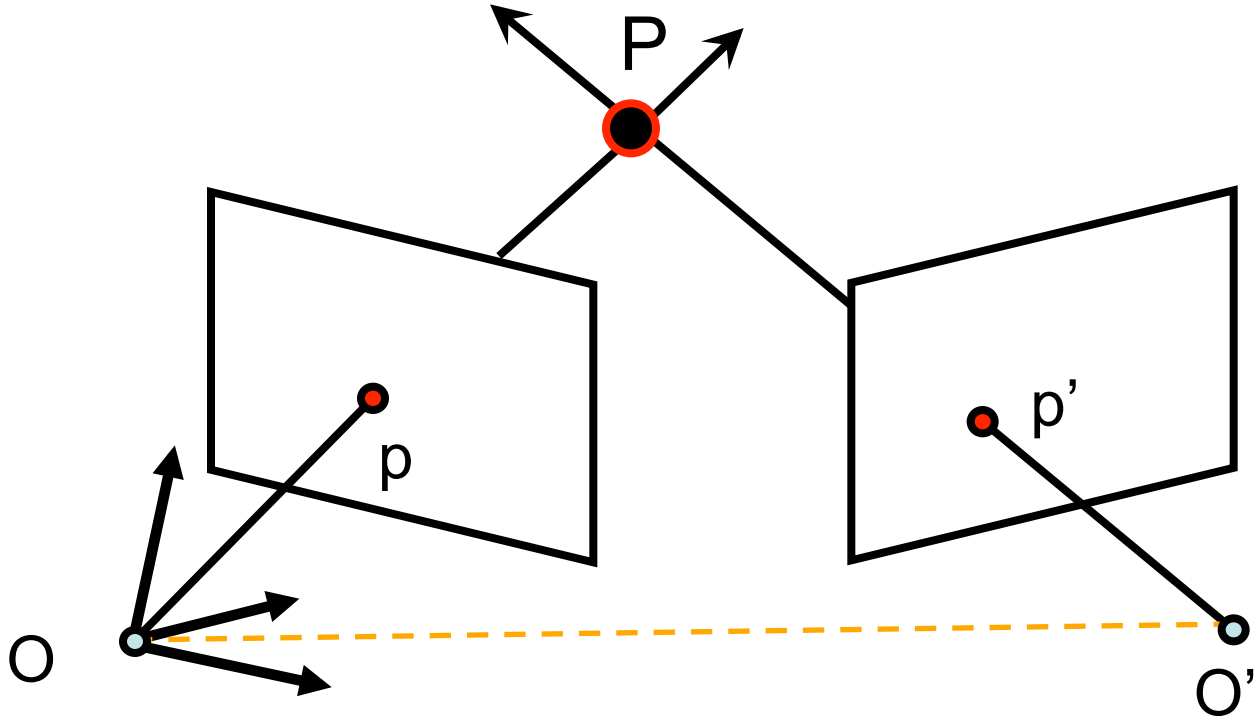
$$p^T \; E \; p' = 0$$

$$E = \left[T_\times\right] \cdot R$$

E = Essential Matrix

(Longuet-Higgins, 1981)

Epipolar Constraint



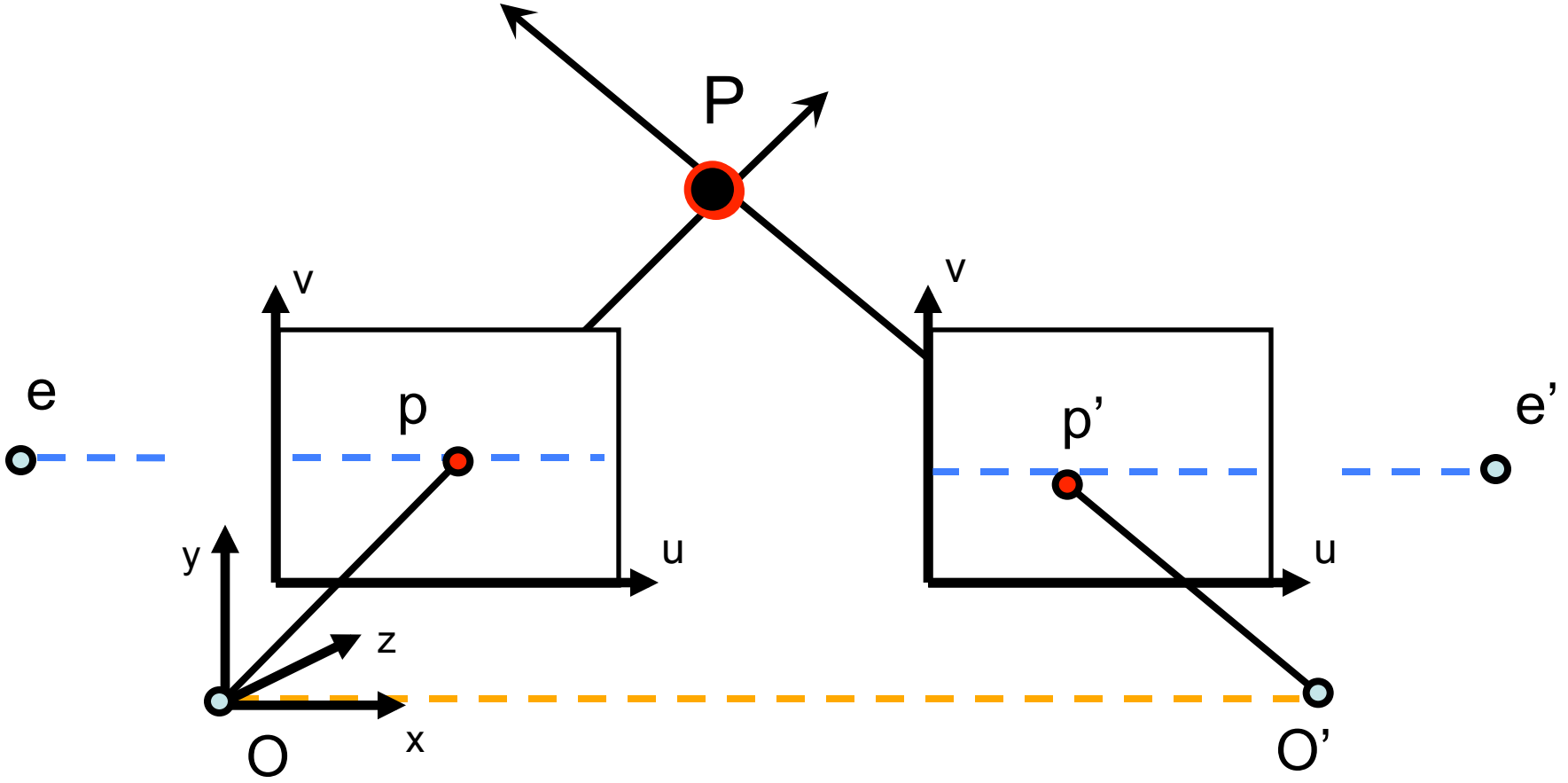
$$p^T F p' = 0$$

$$F = K^{-T} \cdot [T_\times] \cdot R\, K'^{-1}$$

F = Fundamental Matrix

(Faugeras and Luong, 1992)

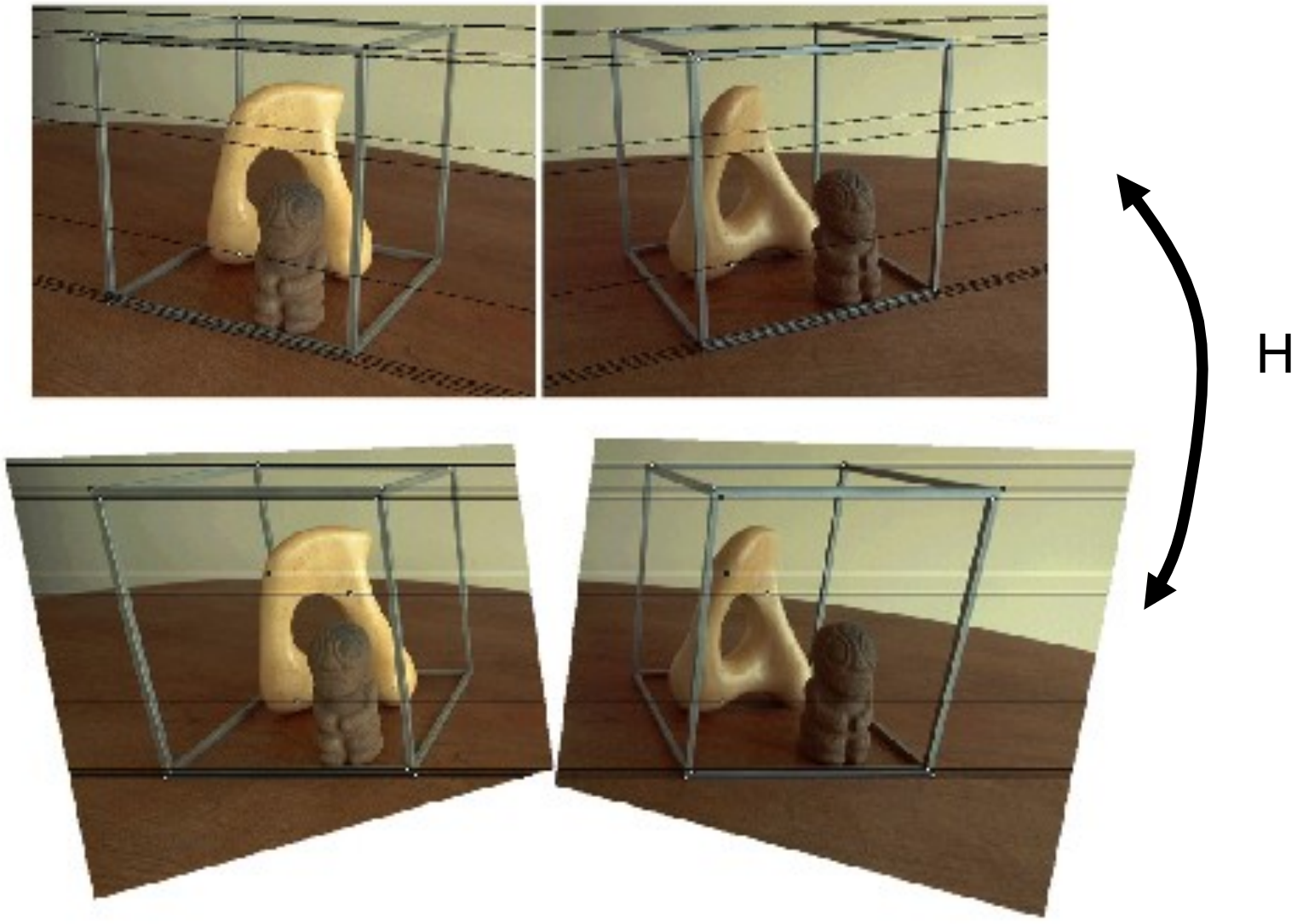
Example: Parallel image planes



- Epipolar lines are horizontal
- Epipoles go to infinity
- v-coordinates are equal

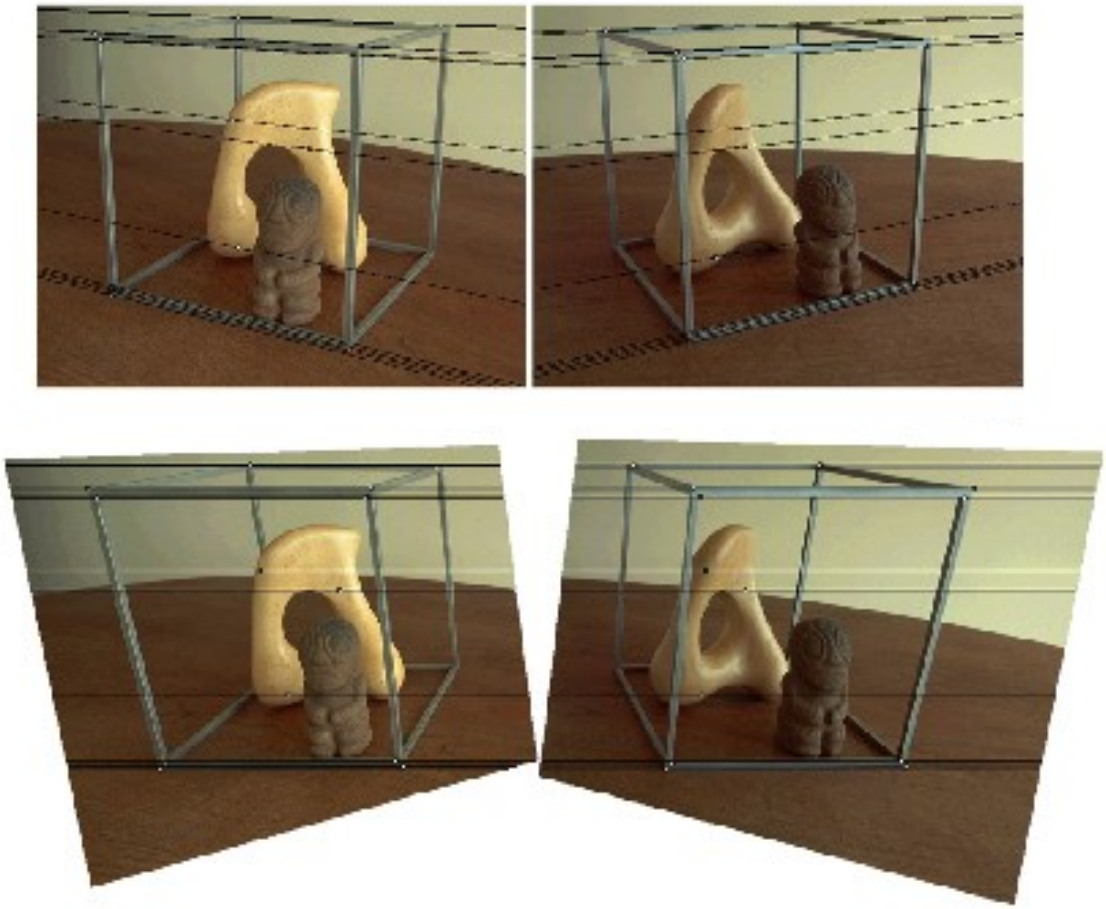
$$p = \begin{bmatrix} p_u \\ p_v \\ 1 \end{bmatrix} \quad p' = \begin{bmatrix} p'_u \\ p_v \\ 1 \end{bmatrix}$$

Rectification: making two images “parallel”



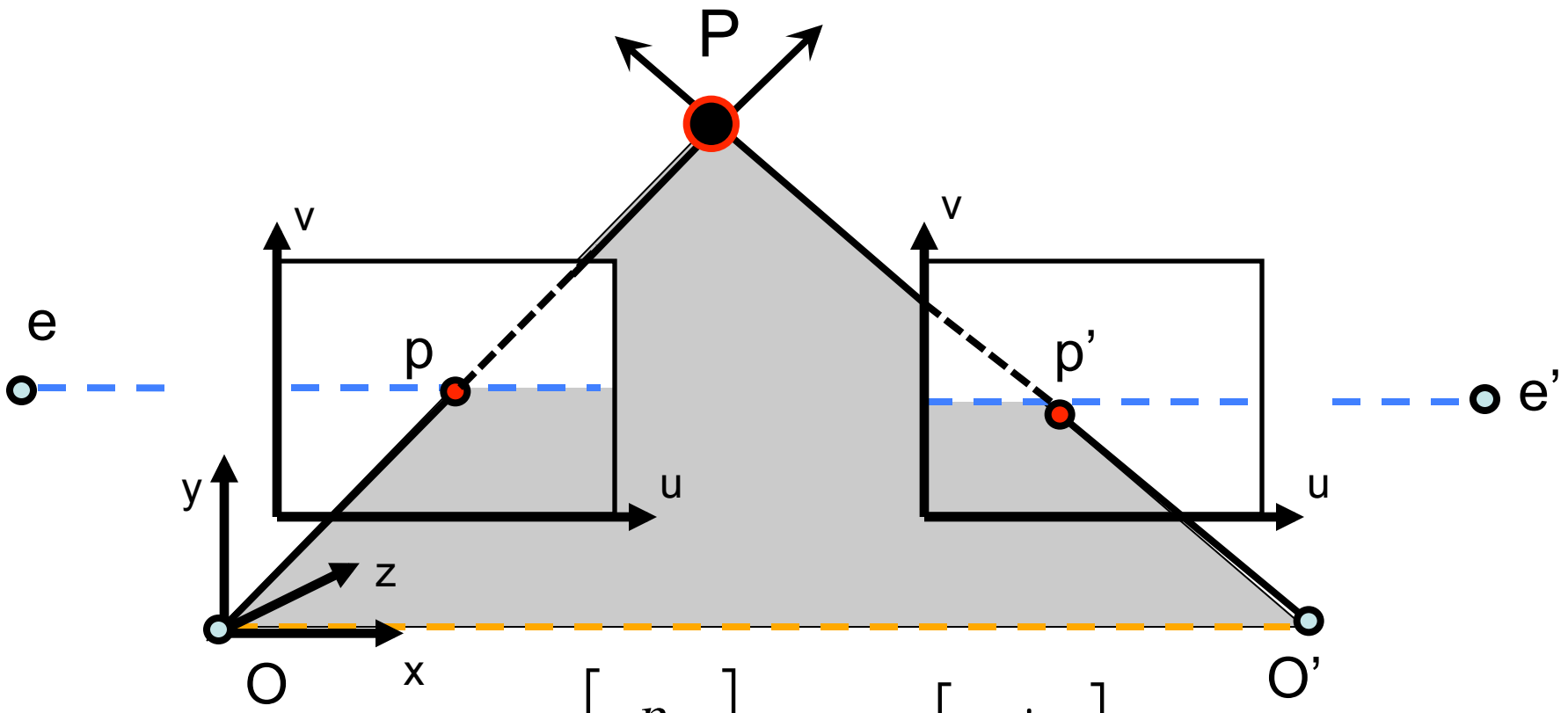
Courtesy figure S. Lazebnik

Why are parallel images useful?



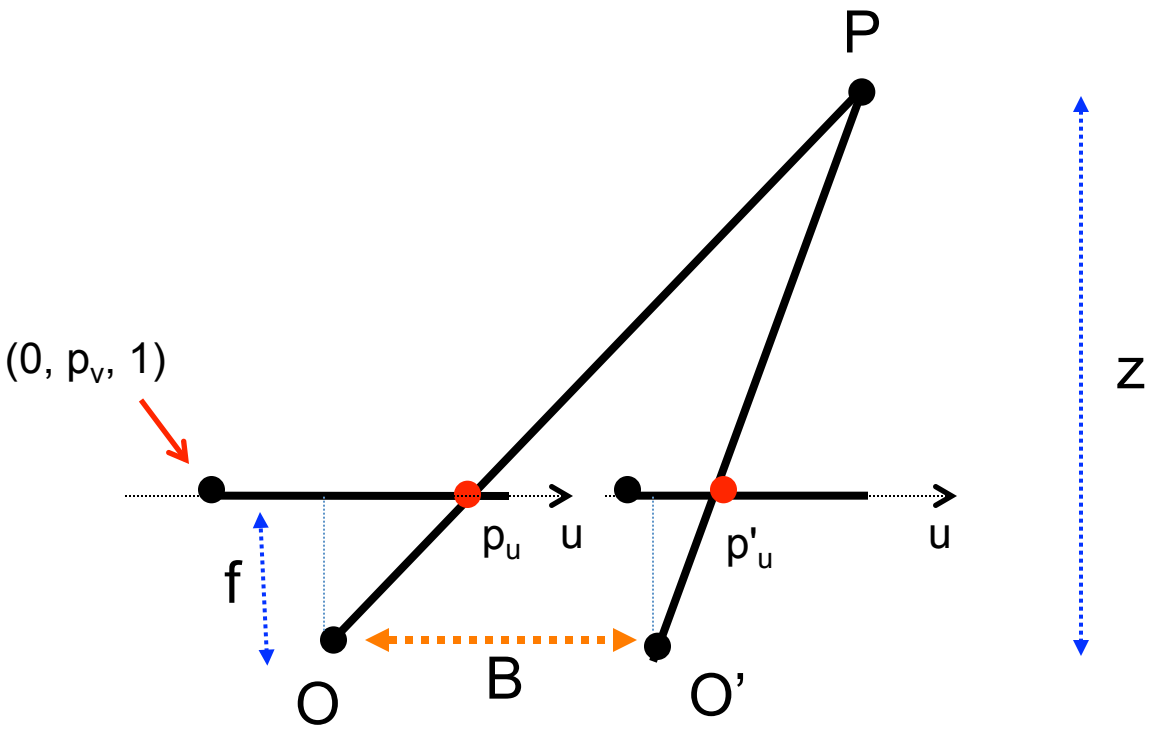
- Makes triangulation easy
- Makes the correspondence problem easier

Point triangulation



$$p = \begin{bmatrix} p_u \\ p_v \\ 1 \end{bmatrix} \qquad p' = \begin{bmatrix} p'_u \\ p_v \\ 1 \end{bmatrix}$$

Computing depth



disparity = $p_u - p'_u \propto \frac{B \cdot f}{z}$ [Eq. 1]

Note: Disparity is inversely proportional to depth

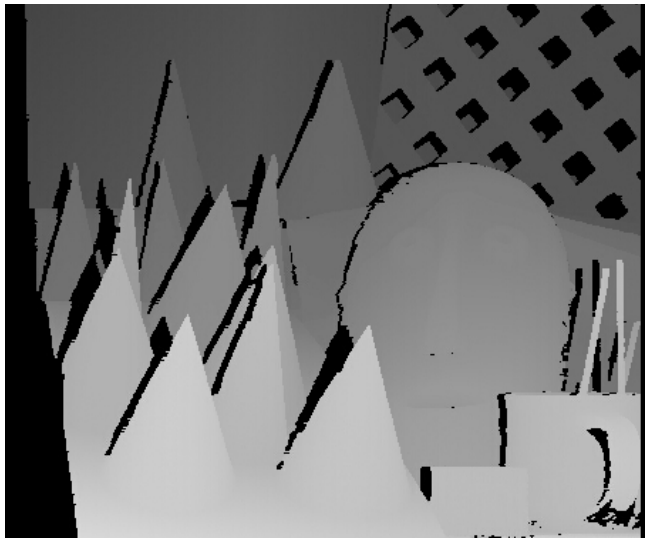
Disparity maps

http://vision.middlebury.edu/stereo/

$$p_u - p'_u \propto \frac{B \cdot f}{z}$$

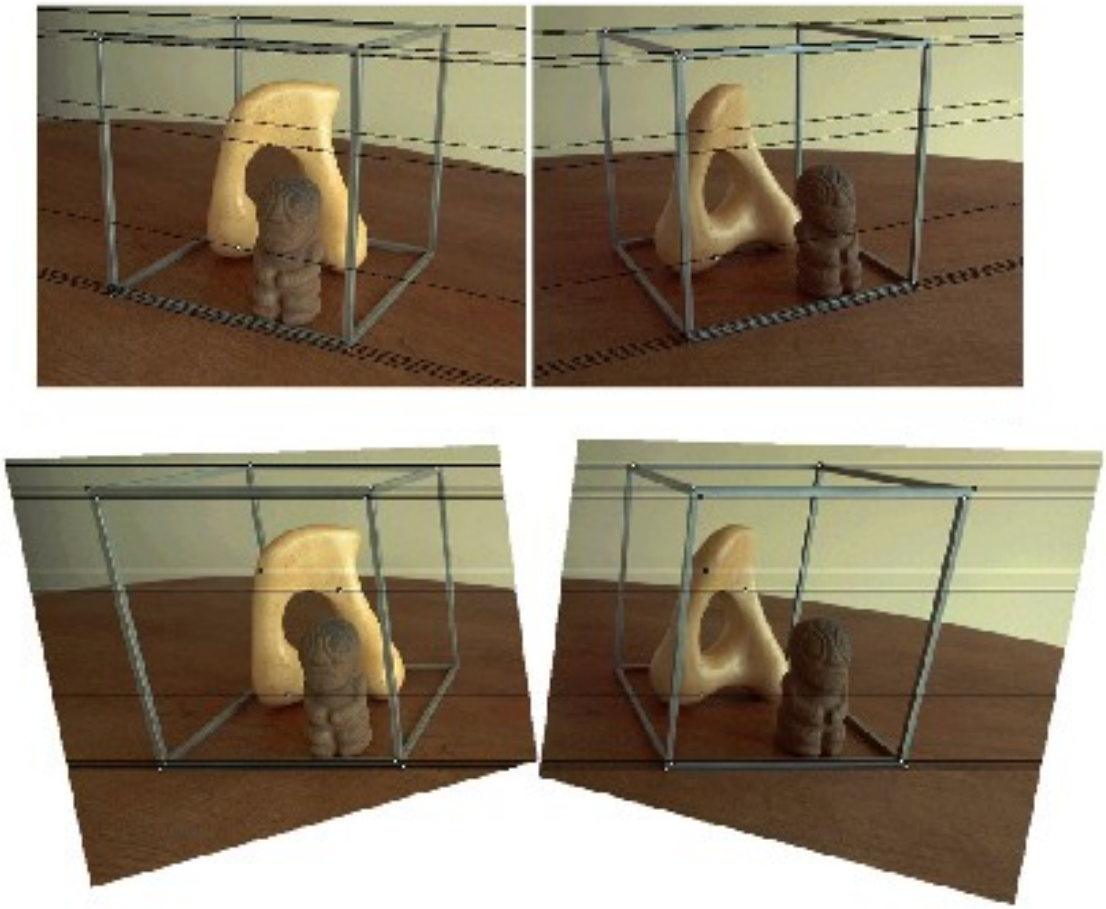
[Eq. 1]

Stereo pair



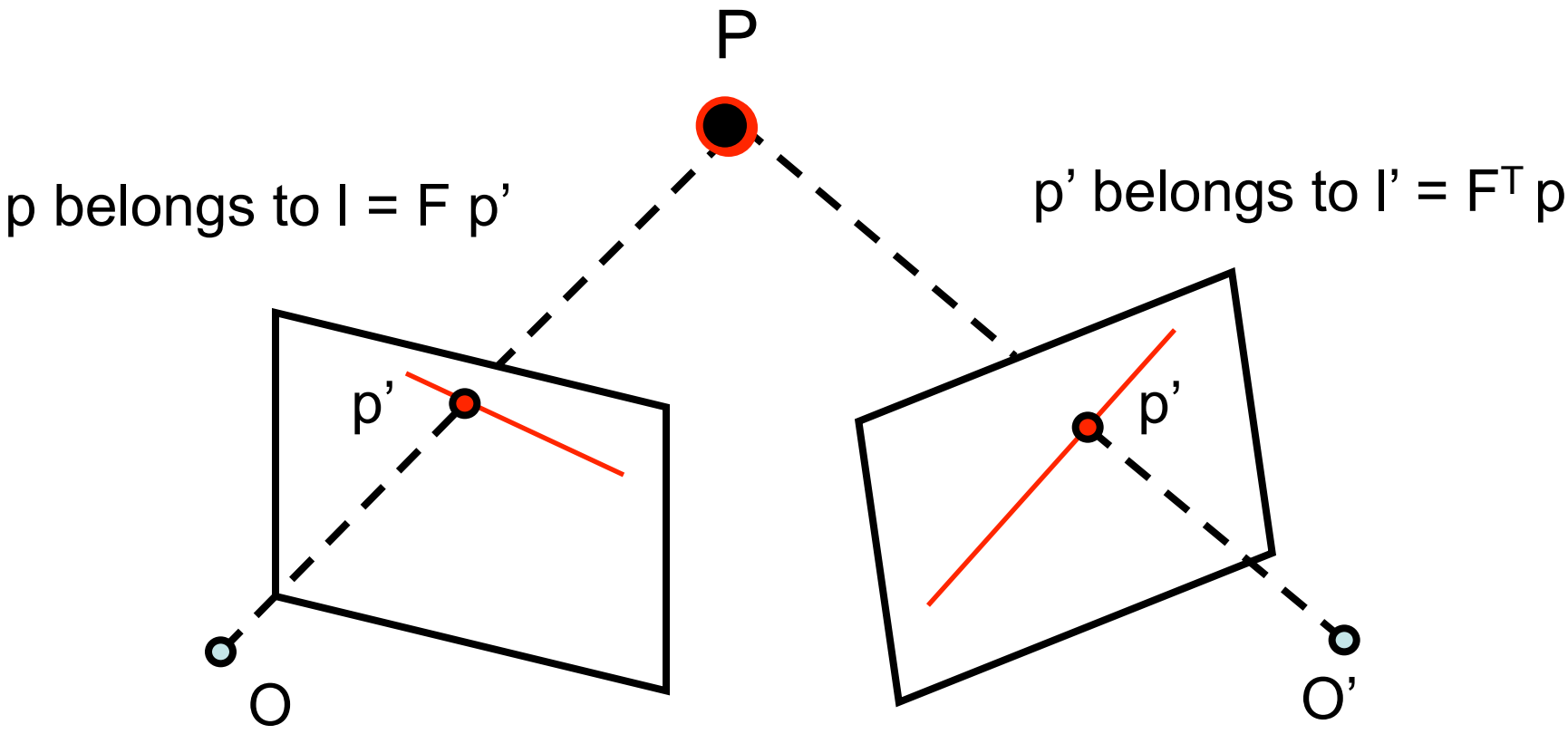
Disparity map / depth map

Why are parallel images useful?



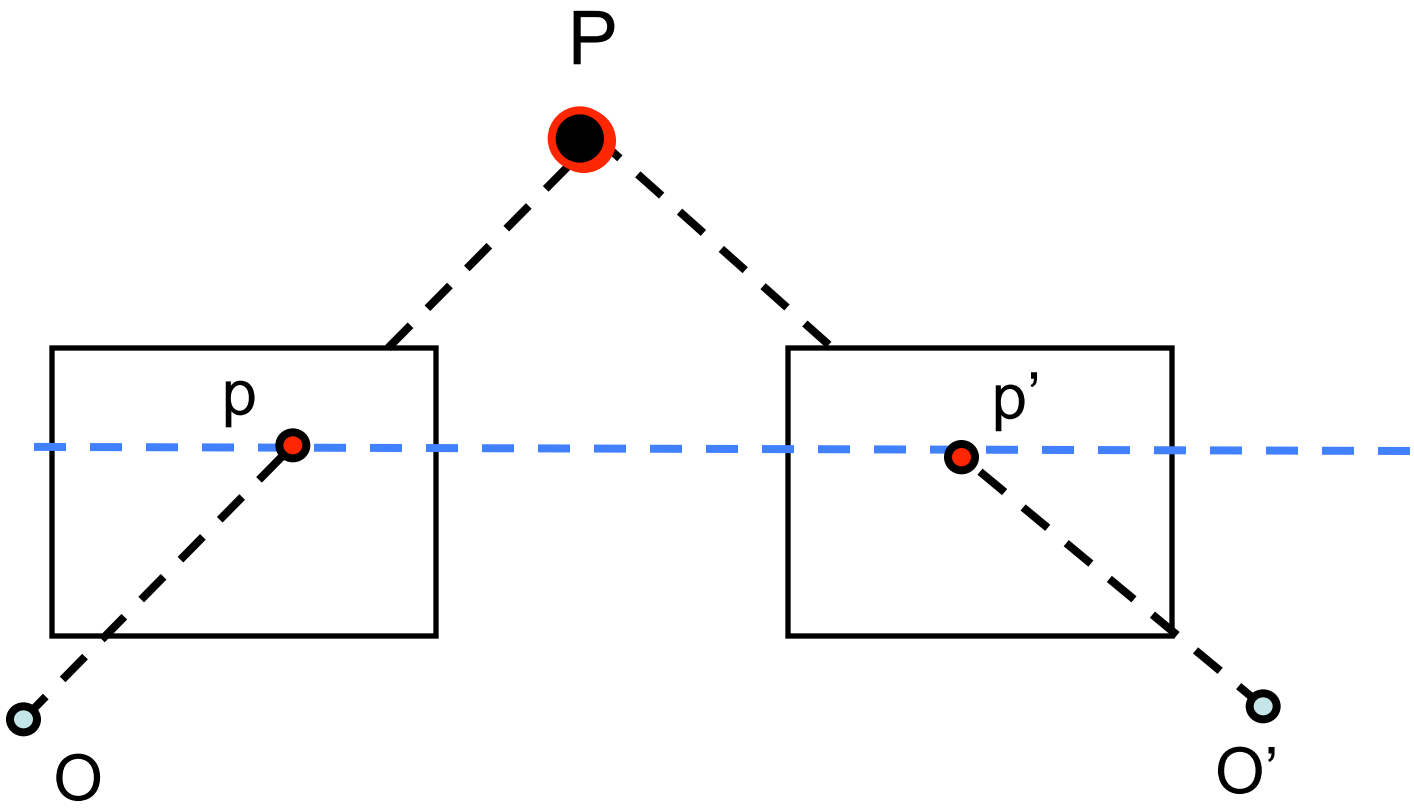
- Makes triangulation easy
- Makes the correspondence problem easier

Correspondence problem



Given a point in 3D, discover corresponding observations in left and right images [also called binocular fusion problem]

Correspondence problem



When images are rectified, this problem is much easier!

Correspondence problem

- A Cooperative Model (Marr and Poggio, 1976)
- Correlation Methods (1970--)
- Multi-Scale Edge Matching (Marr, Poggio and Grimson, 1979-81)

[FP] Chapters: 7

Correlation Methods (1970--)

image 1

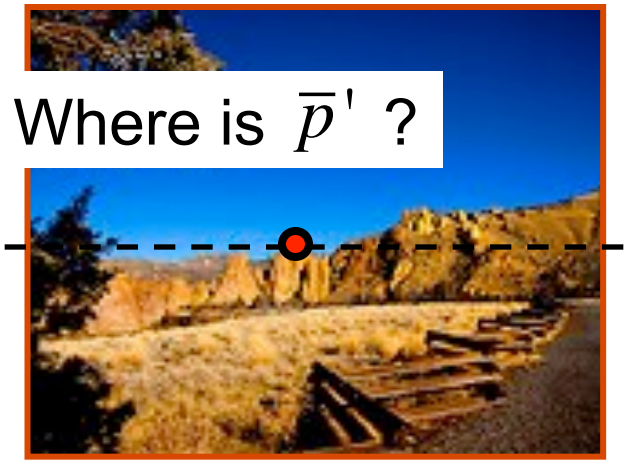
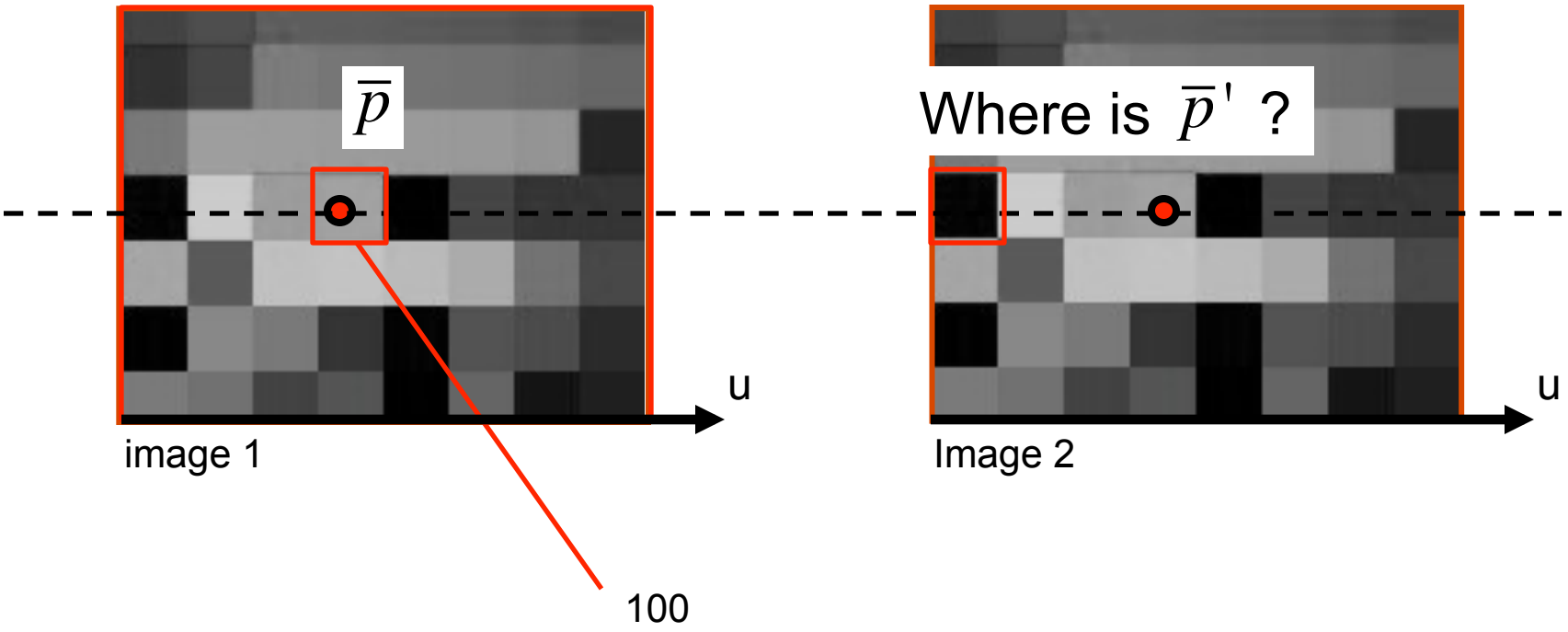


Image 2

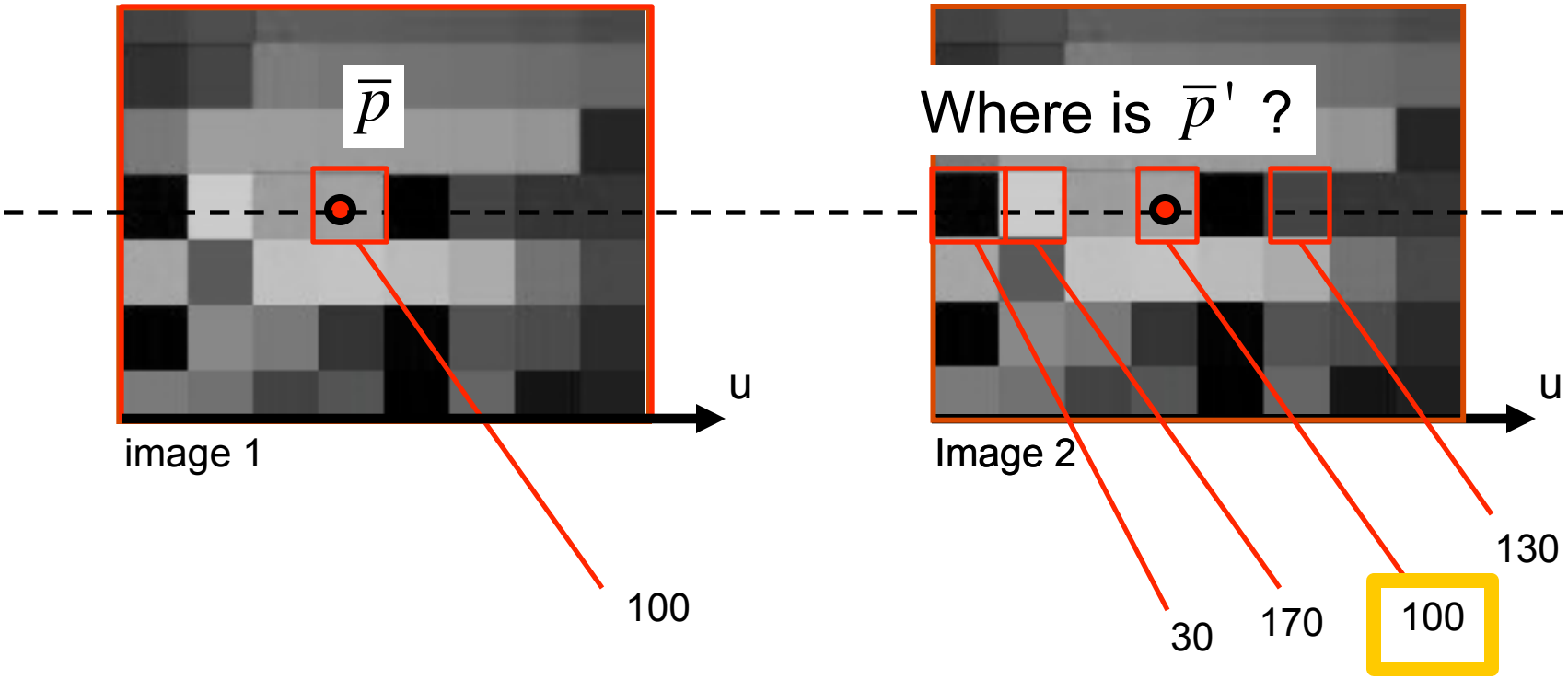
$$\bar{p} = \begin{bmatrix} \bar{u} \\ \bar{v} \\ 1 \end{bmatrix} \qquad \bar{p}' = \begin{bmatrix} \bar{u}' \\ \bar{v} \\ 1 \end{bmatrix}$$

Correlation Methods (1970--)



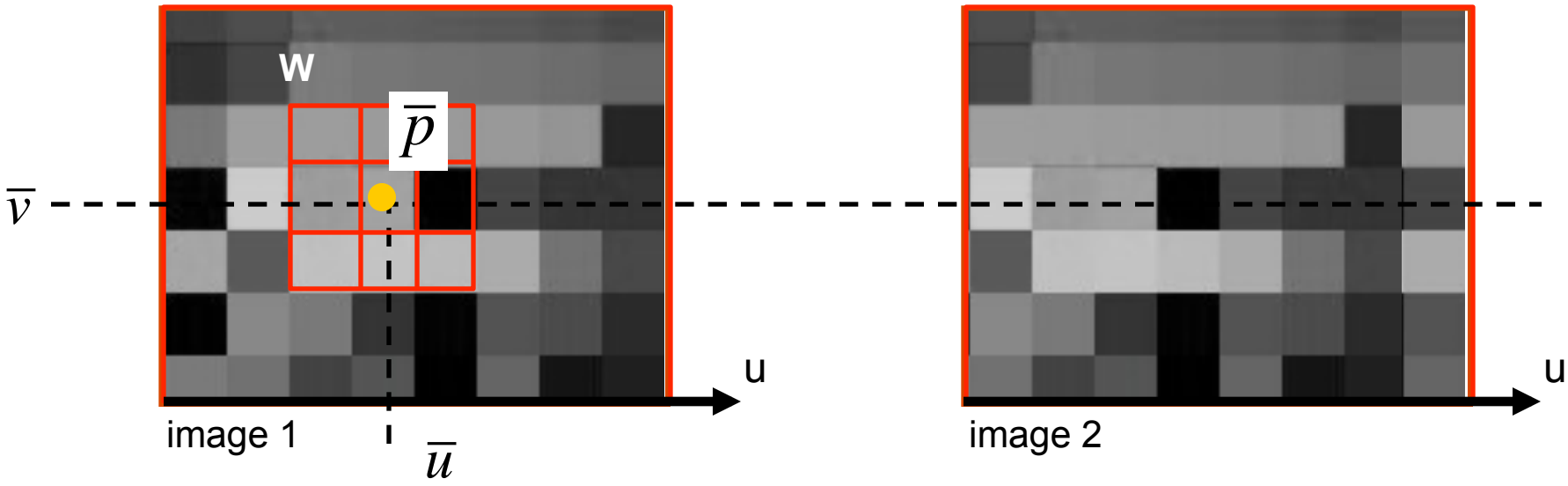
$$\overline{p} = \begin{bmatrix} \overline{u} \\ \overline{v} \\ 1 \end{bmatrix} \qquad \overline{p}' = \begin{bmatrix} \overline{u}' \\ \overline{v} \\ 1 \end{bmatrix}$$

Correlation Methods (1970--)



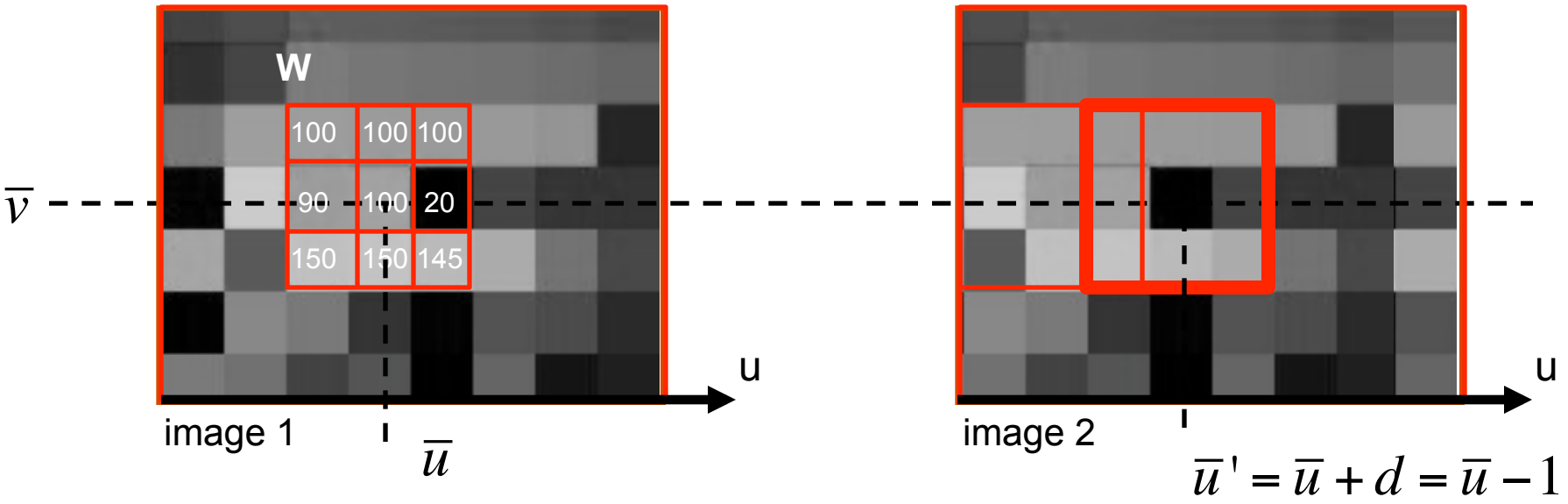
What's the problem with this?

Window-based correlation



- Pick up a window **W** around $\bar{p} = (\bar{u}, \bar{v})$
- Build vector **w**

Window-based correlation



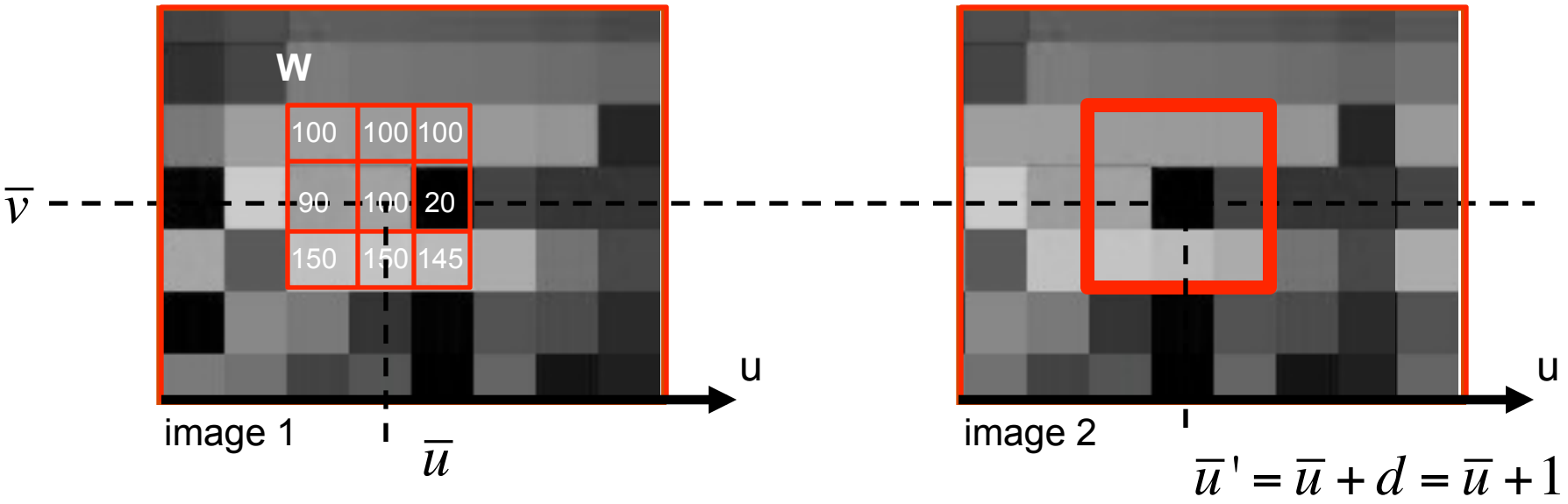
Example: **W** is a 3x3 window in red

w is a 9x1 vector

w = [100, 100, 100, 90, 100, 20, 150, 150, 145]^T

- Pick up a window **W** around $\overline{p} = (\overline{u}, \overline{v})$
- Build vector **w**
- Slide the window **W** along v = $\overline{v}$ in image 2 and compute **w**' (u) for each u
- Compute the dot product $\mathbf{w}^T$ **w'**(u) for each u and retain the max value

Window-based correlation



Example: **W** is a 3x3 window in red

w is a 9x1 vector

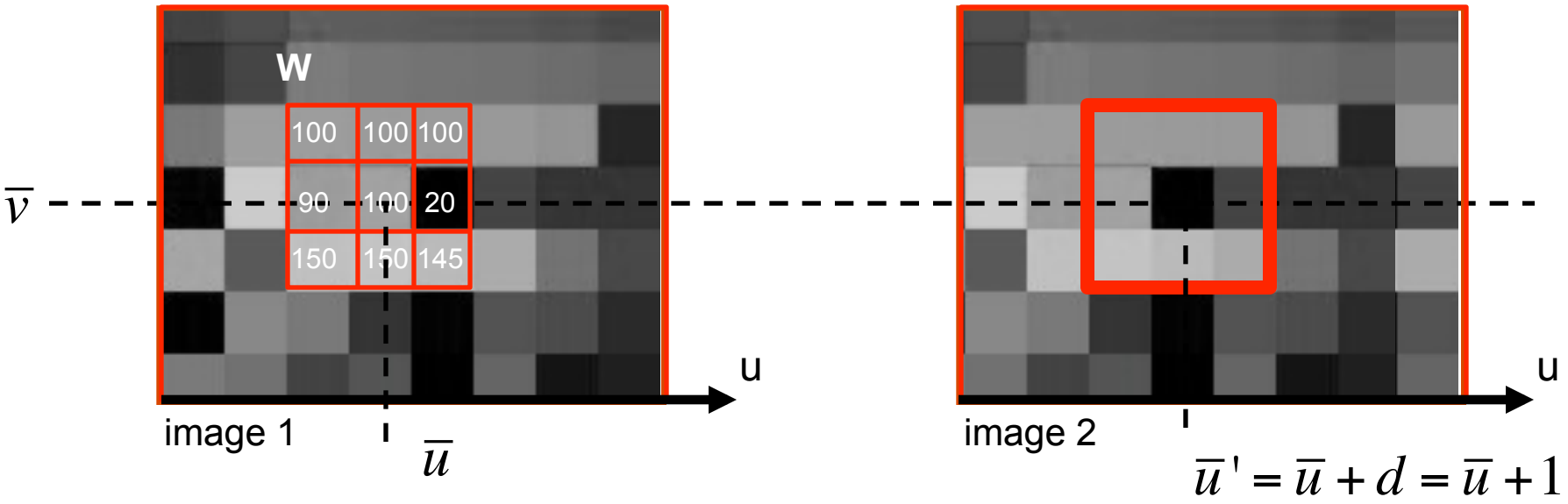
w = [100, 100, 100, 90, 100, 20, 150, 150, 145]^T

What's the problem with this?

Changes of brightness/exposure

Changes in the mean and the variance of intensity values in corresponding windows!

Normalized cross-correlation



Find u that maximizes:

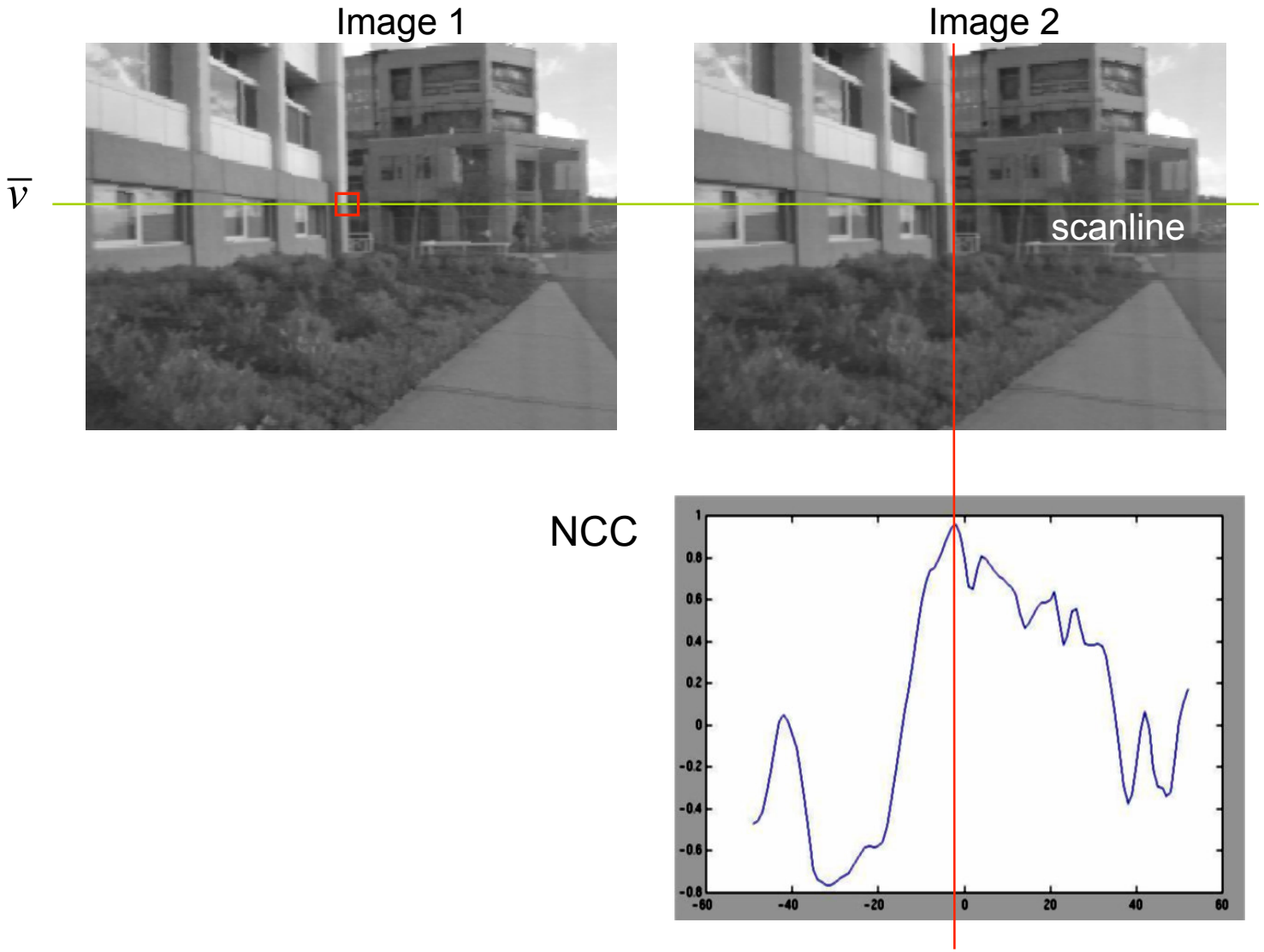
$$\frac{(w-\bar{w})^T (w'(u)-\bar{w}')}{\left\|(w-\bar{w})\right\| \left\|(w'(u)-\bar{w}')\right\|}$$

[Eq. 2]

$\bar{w}$ = mean value within **W** located at u^bar in image 1

$\bar{w}'(u)$ = mean value within **W** located at u in image 2

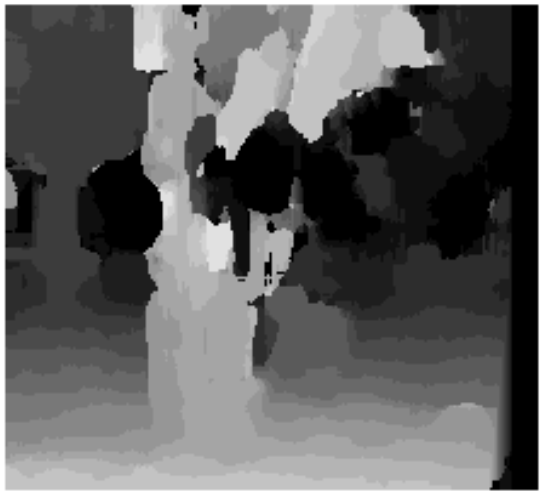
Example



Credit slide S. Lazebnik

Effect of the window's size

Window size = 3



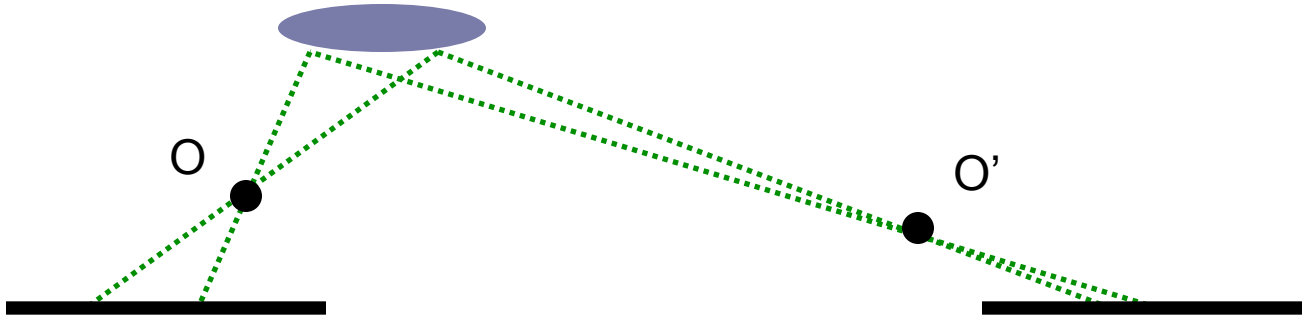
Window size = 20

- Smaller window
  - More detail
  - More noise
- Larger window
  - Smoother disparity maps
  - Less prone to noise

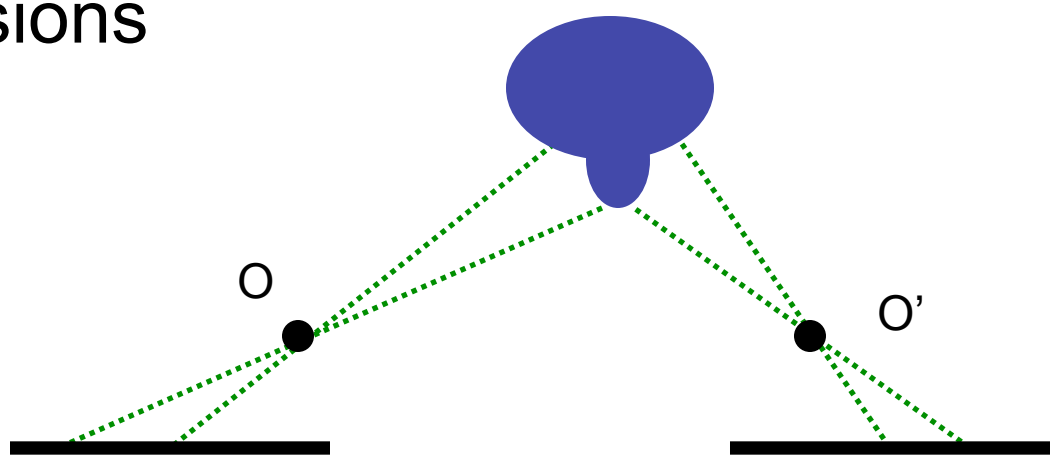
Credit slide S. Lazebnik

Issues

- Fore shortening effect

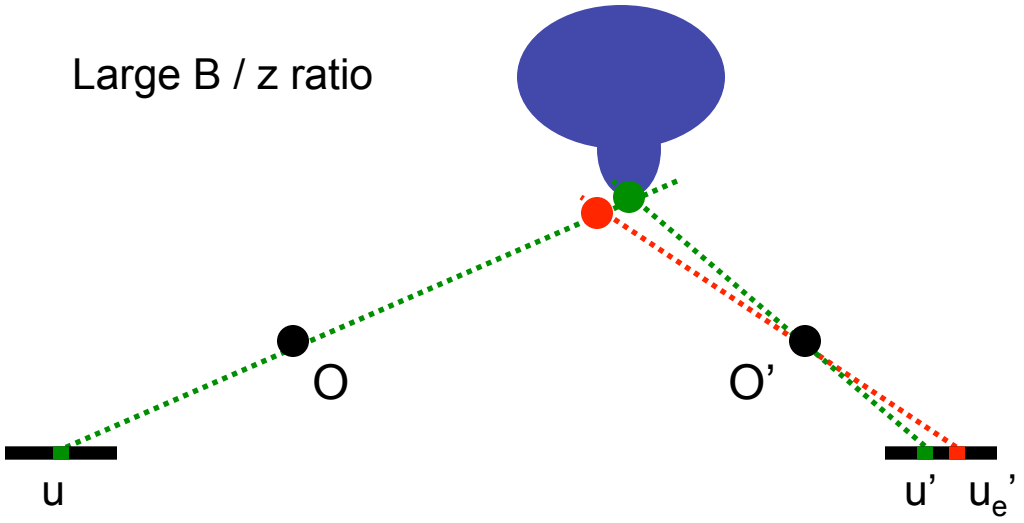


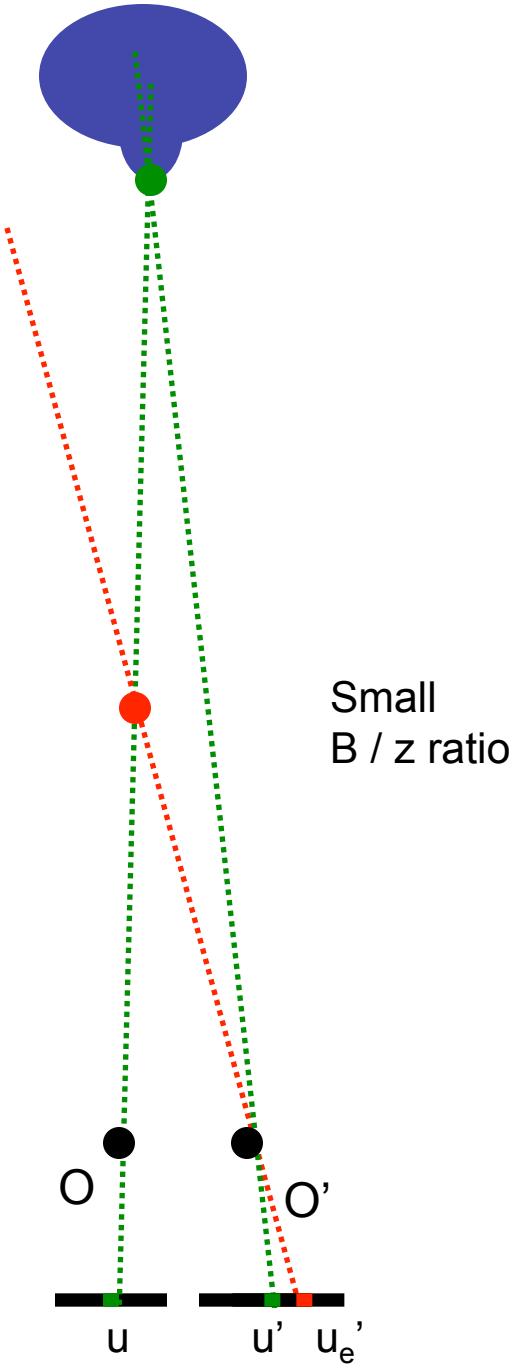
- Occlusions



Issues

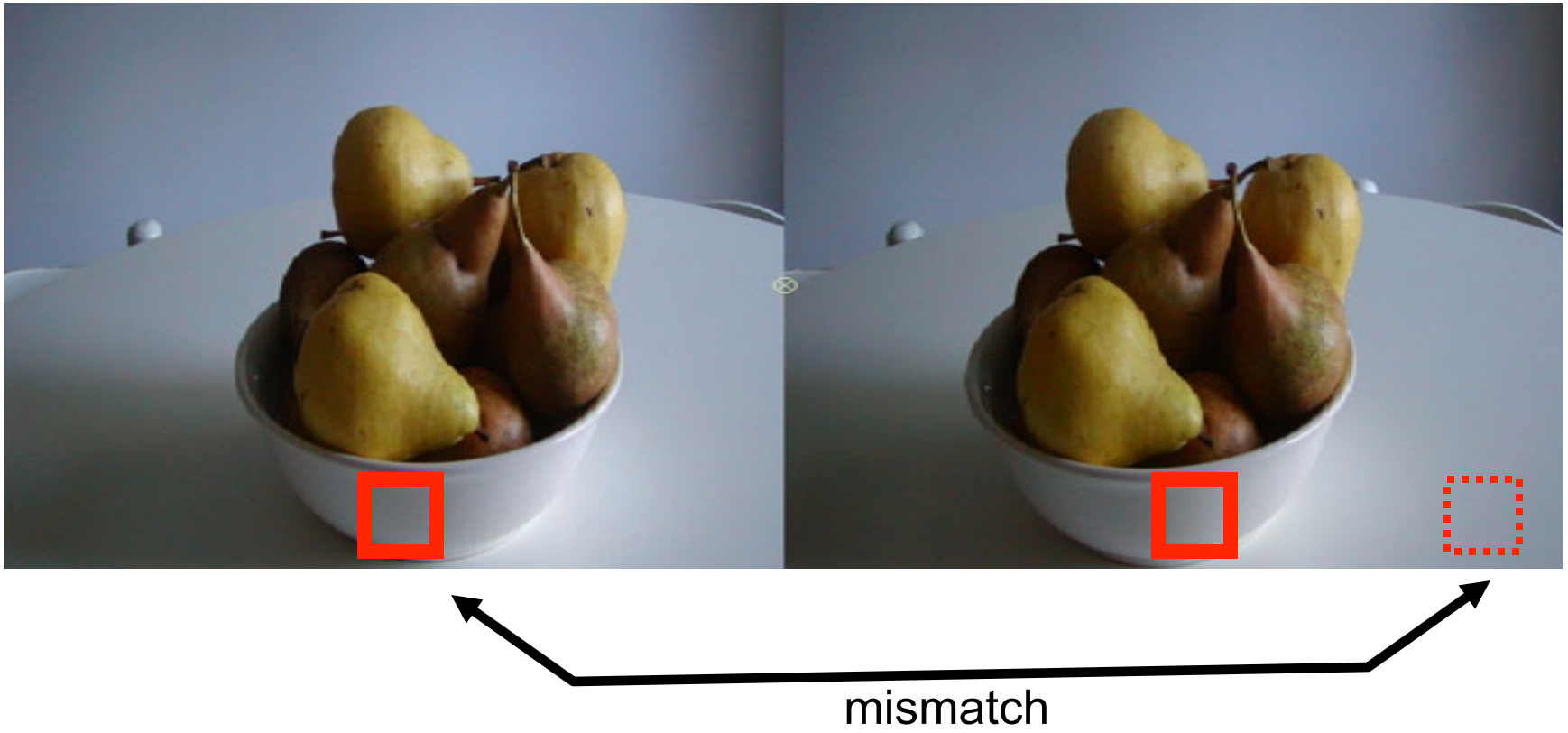
- To reduce the effect of foreshortening and occlusions, it is desirable to have small B / z ratio!
- However, when B/z is small, small errors in measurements imply large error in estimating depth





Issues

- Homogeneous regions



Issues

- Repetitive patterns

Correspondence problem is difficult!

- Occlusions
- Fore shortening
- Baseline trade-off
- Homogeneous regions
- Repetitive patterns

Apply non-local constraints to help enforce the correspondences

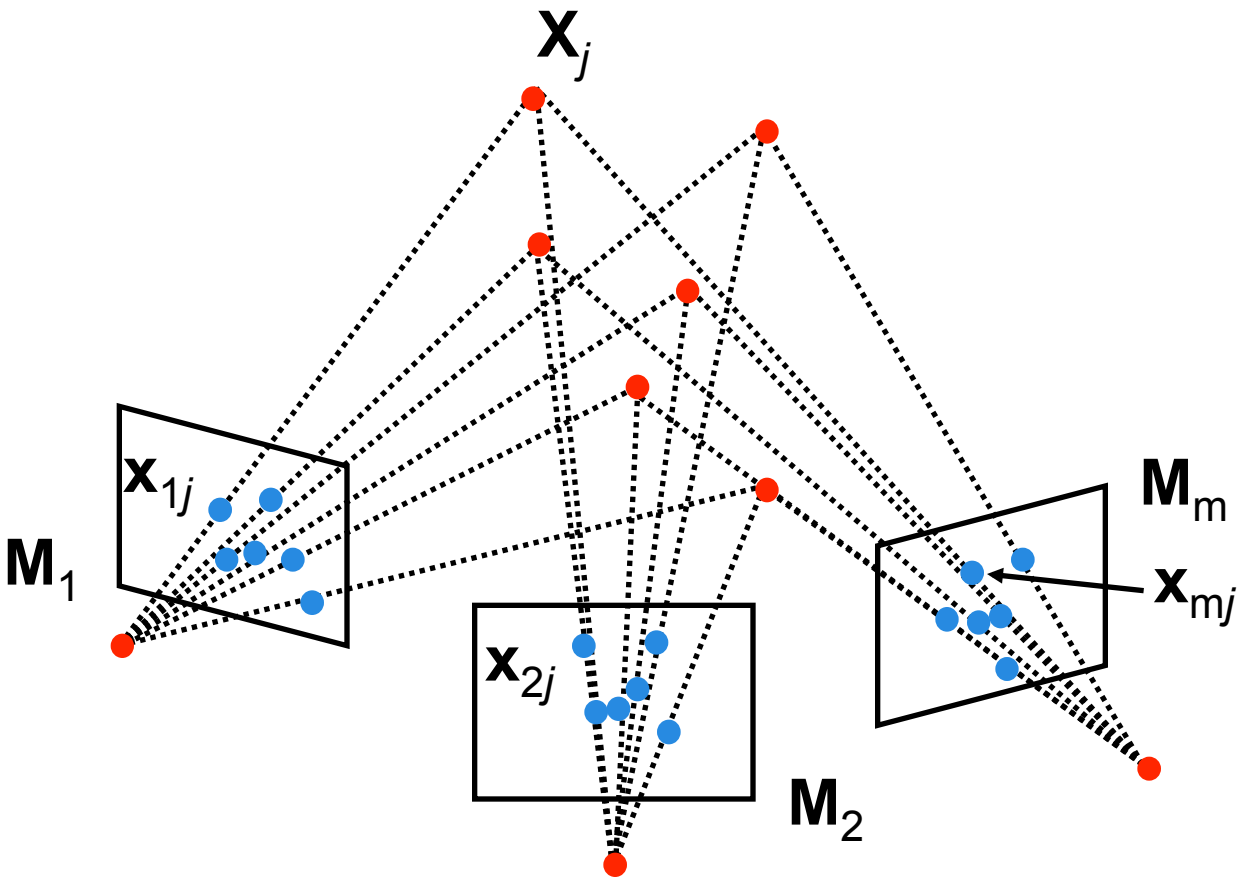
Non-local constraints

- Uniqueness
  - For any point in one image, there should be at most one matching point in the other image
- Ordering
  - Corresponding points should be in the same order in both views
- Smoothness
  - Disparity is typically a smooth function of x (except in occluding boundaries)

Lecture 6
Stereo Systems
Multi-view geometry

- Stereo systems
  - Rectification
  - Correspondence problem
- Multi-view geometry
  - The SFM problem
  - Affine SFM

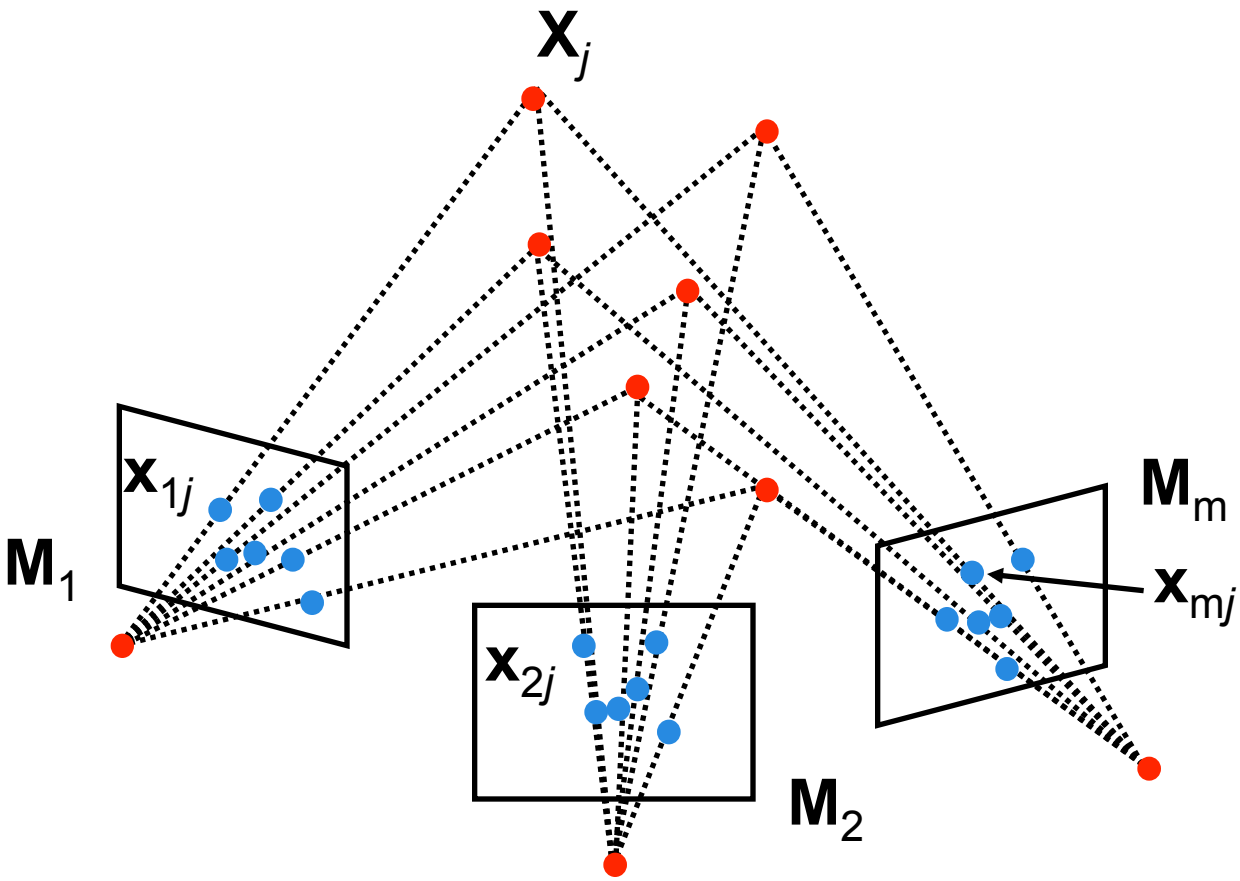
Structure from motion problem



Given *m* images of *n* fixed 3D points

- $\mathbf{x}_{ij} = \mathbf{M}_i \mathbf{X}_j, \quad i = 1, \ldots, m, \quad j = 1, \ldots, n$

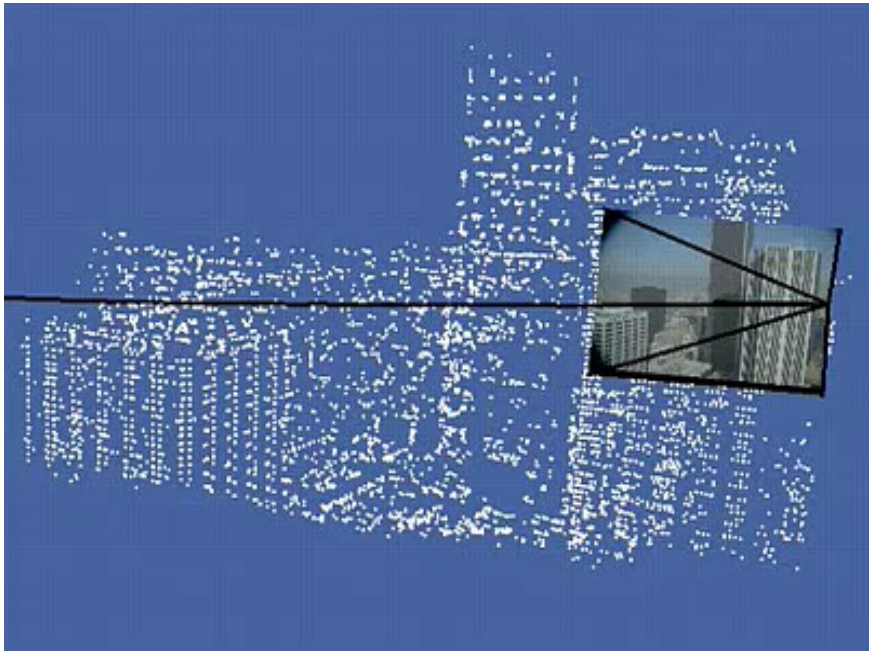
Structure from motion problem



From the mxn observations $\mathbf{x}_{ij}$, estimate:

•*m* projection matrices $\mathbf{M}_i$	motion
•*n* 3D points $\mathbf{X}_j$	structure

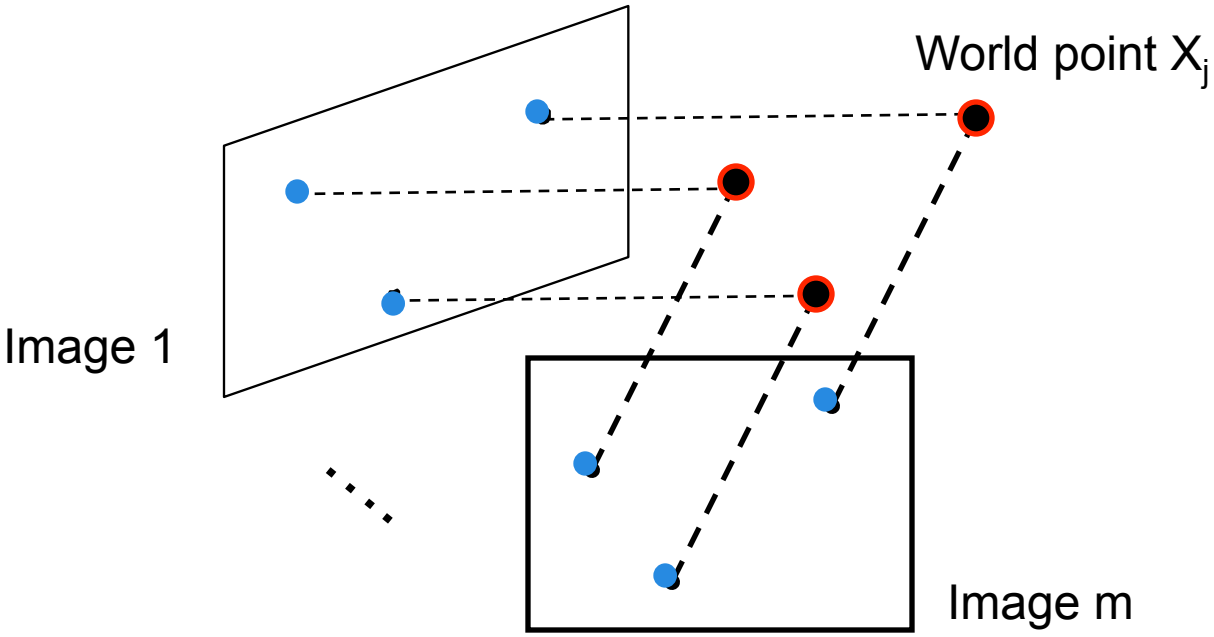
Structure from motion problem



Courtesy of Oxford **Visual Geometry Group**

Affine structure from motion

(simpler problem)



From the mxn observations $\mathbf{x}_{ij}$, estimate:

- *m* projection matrices $\mathbf{M}_i$ (affine cameras)
- *n* 3D points $\mathbf{X}_j$

Perspective

$$\mathbf{x} = M\,\mathbf{X} = \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{bmatrix} \mathbf{X} = \begin{bmatrix} \mathbf{m}_1\mathbf{X} \\ \mathbf{m}_2\mathbf{X} \\ \mathbf{m}_3\mathbf{X} \end{bmatrix}$$

$$M = \begin{bmatrix} \mathbf{A} & \mathbf{b} \\ \mathbf{v} & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{bmatrix}$$

$$\mathbf{x}^E = \left(\frac{\mathbf{m}_1\mathbf{X}}{\mathbf{m}_3\mathbf{X}}, \frac{\mathbf{m}_2\mathbf{X}}{\mathbf{m}_3\mathbf{X}}\right)$$

Affine

$$\mathbf{x} = M\,\mathbf{X} = \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{bmatrix} \mathbf{X} = \begin{bmatrix} \mathbf{m}_1\mathbf{X} \\ \mathbf{m}_2\mathbf{X} \\ 1 \end{bmatrix}$$

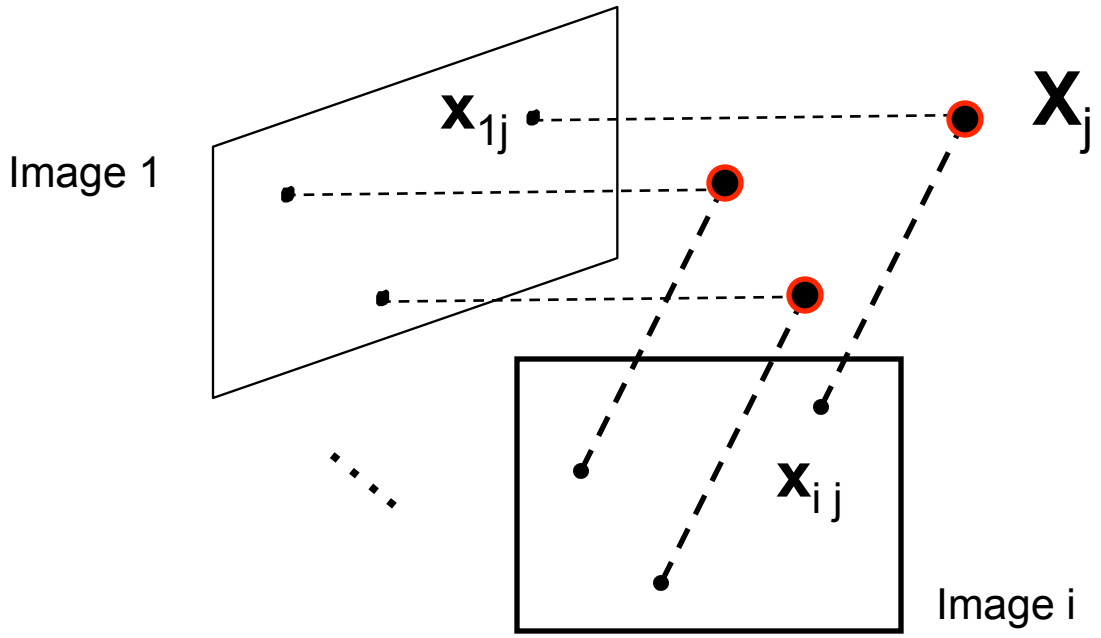
$$M = \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{2x3} & \mathbf{b}_{2x1} \\ \mathbf{0}_{1x3} & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ 0 \quad 0 \quad 0 \quad 1 \end{bmatrix}$$

$$\mathbf{x}^E = (\mathbf{m}_1\mathbf{X}, \mathbf{m}_2\mathbf{X}) = \begin{bmatrix} \mathbf{A} & \mathbf{b} \end{bmatrix} \mathbf{X} = \begin{bmatrix} \mathbf{A} & \mathbf{b} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \mathbf{A}\mathbf{X}^E + \mathbf{b} \qquad \mathbf{X}^E = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

magnification

[Eq. 3]

Affine cameras



Camera matrix $\mathbf{M}^a$ for the affine case (in Euclidean space)

$$\mathbf{x}_{ij} = \mathbf{A}_i\mathbf{X}_j + \mathbf{b}_i = \mathbf{M}_i^a \begin{bmatrix} \mathbf{X}_j \\ 1 \end{bmatrix}; \qquad \mathbf{M}^a = \begin{bmatrix} \mathbf{A} & \mathbf{b} \end{bmatrix}$$

[Eq. 4]

The Affine Structure-from-Motion Problem

Given *m* images of *n* fixed points $\mathbf{X}_j$ we can write

$$\mathbf{x}_{ij} = \mathbf{A}_i\mathbf{X}_j + \mathbf{b}_i$$ for i = 1, …,m and j = 1, … ,n

N. of cameras N. of points

Problem: estimate m matrices A_i, m matrices b_i and the n positions $\mathbf{X}_j$ from the m×n observations $\mathbf{x}_{ij}$.

How many equations and how many unknown?

2m × n equations in 8m+3n unknowns

With m=2 cameras, I need at least n=16 3D points
With m=3 cameras, I need at least n=8 3D points

Two approaches:

- Algebraic approach (affine epipolar geometry; estimate F; cameras; points)
- Factorization method

A factorization method – Tomasi & Kanade algorithm

C. Tomasi and T. Kanade
Shape and motion from image streams under orthography: A factorization method. *IJCV*, 9(2):137-154, November 1992.

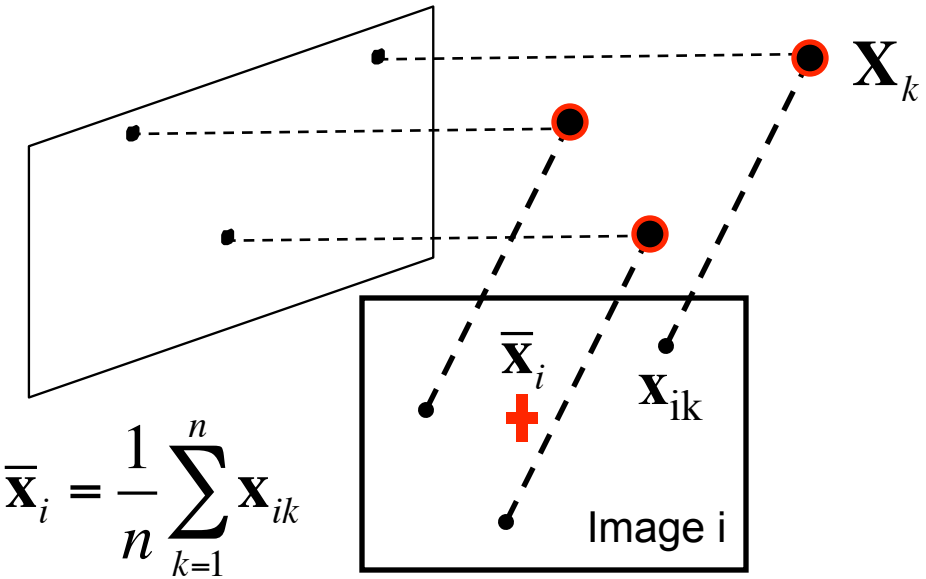
- Data centering
- Factorization

A factorization method - Centering the data

Centering: subtract the centroid of the image points

[Eq. 6] $$\hat{\mathbf{x}}_{ij} = \mathbf{x}_{ij} - \frac{1}{n}\sum_{k=1}^{n}\mathbf{x}_{ik}$$

$\frac{1}{n}\sum_{k=1}^{n}\mathbf{x}_{ik} = \bar{\mathbf{x}}_i$



[Eq. 5] $$\bar{\mathbf{x}}_i = \frac{1}{n}\sum_{k=1}^{n}\mathbf{x}_{ik}$$

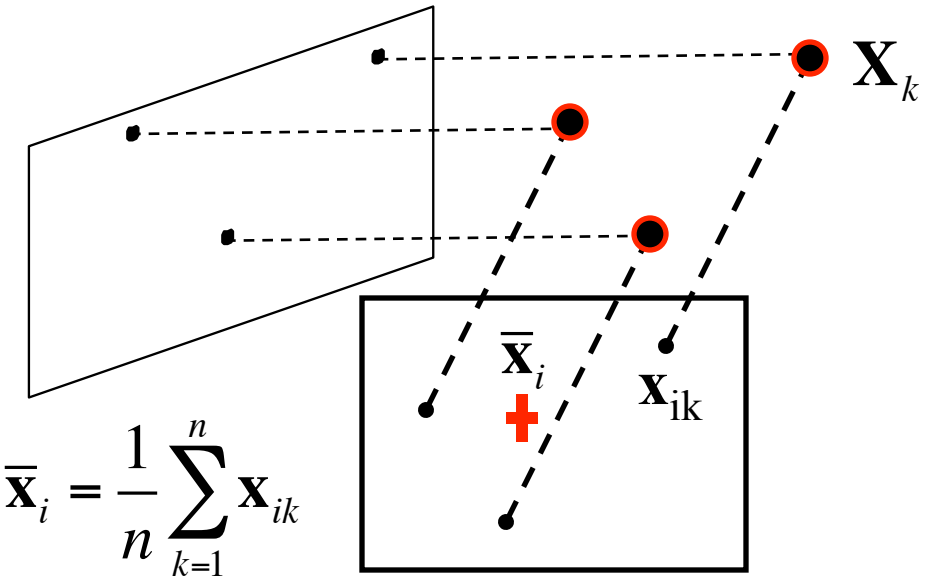
A factorization method - Centering the data

Centering: subtract the centroid of the image points

[Eq. 6] $$\hat{\mathbf{x}}_{ij} = \mathbf{x}_{ij} - \frac{1}{n}\sum_{k=1}^{n}\mathbf{x}_{ik} = \mathbf{A}_i\mathbf{X}_j + \mathbf{b}_i - \frac{1}{n}\sum_{k=1}^{n}\mathbf{A}_i\mathbf{X}_k - \frac{1}{n}\sum_{k=1}^{n}\mathbf{b}_i$$

$$\mathbf{x}_{ik} = \mathbf{A}_i\mathbf{X}_k + \mathbf{b}_i$$

[Eq. 4]



[Eq. 5] $$\bar{\mathbf{x}}_i = \frac{1}{n}\sum_{k=1}^{n}\mathbf{x}_{ik}$$

A factorization method - Centering the data

Centering: subtract the centroid of the image points

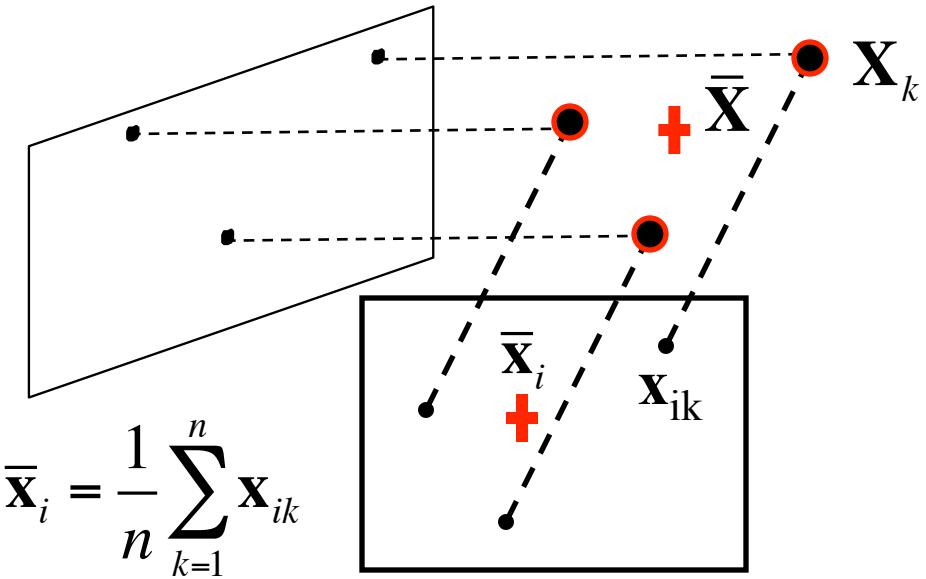
$$\hat{\mathbf{x}}_{ij} = \mathbf{x}_{ij} - \frac{1}{n}\sum_{k=1}^{n}\mathbf{x}_{ik} = \mathbf{A}_i\mathbf{X}_j + \mathbf{b}_i - \frac{1}{n}\sum_{k=1}^{n}\mathbf{A}_i\mathbf{X}_k - \frac{1}{n}\sum_{k=1}^{n}\mathbf{b}_i$$

$$= \mathbf{A}_i\left(\mathbf{X}_j - \frac{1}{n}\sum_{k=1}^{n}\mathbf{X}_k\right) = \mathbf{A}_i\left(\mathbf{X}_j - \bar{\mathbf{X}}\right)$$

$$= \mathbf{A}_i\hat{\mathbf{X}}_j \quad \text{[Eq. 8]}$$

$$\mathbf{x}_{ik} = \mathbf{A}_i\mathbf{X}_k + \mathbf{b}_i$$

[Eq. 4]



$$\bar{\mathbf{x}}_i = \frac{1}{n}\sum_{k=1}^{n}\mathbf{x}_{ik}$$

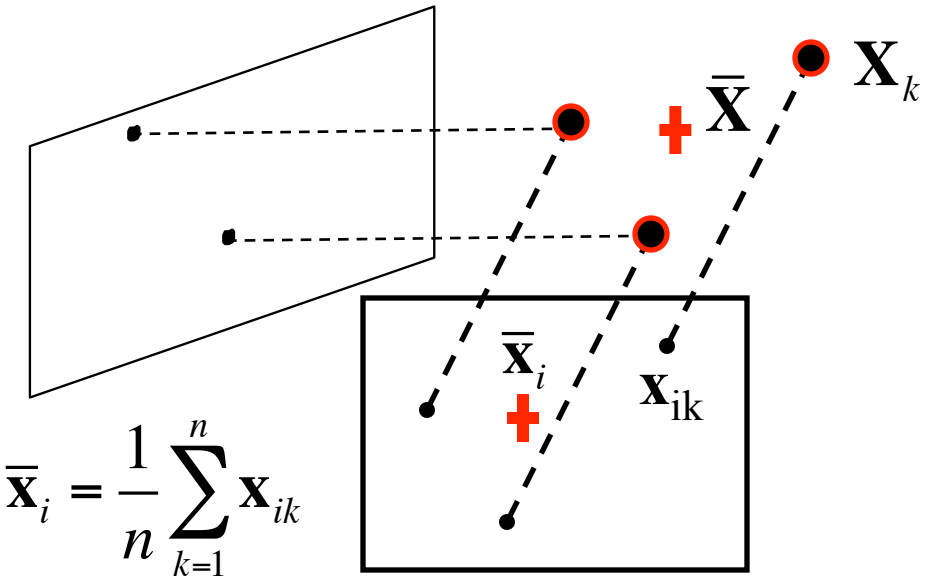
$$\bar{\mathbf{X}} = \frac{1}{n}\sum_{k=1}^{n}\mathbf{X}_k \quad \text{[Eq. 7]}$$

Centroid of 3D points

A factorization method - Normalizing the data

Thus, after centering, each **normalize**d observed point is related to the 3D point by

$$\hat{\mathbf{x}}_{ij} = \mathbf{A}_i \hat{\mathbf{X}}_j \quad \text{[Eq. 8]}$$



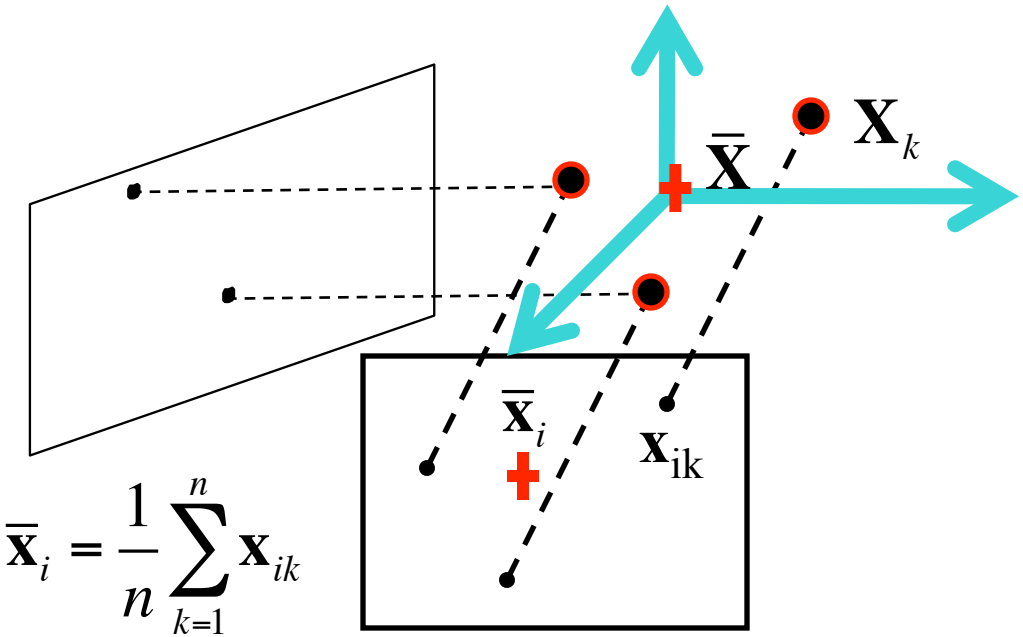
$$\bar{\mathbf{X}} = \frac{1}{n}\sum_{k=1}^{n} \mathbf{X}_k \quad \text{[Eq. 7]}$$

Centroid of 3D points

A factorization method - Normalizing the data

If the centroid of points in 3D = center of the world reference system

$$\hat{\mathbf{x}}_{ij} = \mathbf{A}_i \hat{\mathbf{X}}_j = \mathbf{A}_i \mathbf{X}_j \quad \text{[Eq. 9]}$$



$$\bar{\mathbf{x}}_i = \frac{1}{n}\sum_{k=1}^{n} \mathbf{x}_{ik}$$

$$\bar{\mathbf{X}} = \frac{1}{n}\sum_{k=1}^{n} \mathbf{X}_k \quad \text{[Eq. 7]}$$

Centroid of 3D points

A factorization method - factorization

Let's create a 2m × n data (measurement) matrix:

$$\mathbf{D} = \begin{bmatrix} \hat{\mathbf{x}}_{11} & \hat{\mathbf{x}}_{12} & \cdots & \hat{\mathbf{x}}_{1n} \\ \hat{\mathbf{x}}_{21} & \hat{\mathbf{x}}_{22} & \cdots & \hat{\mathbf{x}}_{2n} \\ & & \ddots & \\ \hat{\mathbf{x}}_{m1} & \hat{\mathbf{x}}_{m2} & \cdots & \hat{\mathbf{x}}_{mn} \end{bmatrix}$$

cameras ($2m$)

points (n)

Each $\hat{\mathbf{x}}_{ij}$ entry is a 2x1 vector!

A factorization method - factorization

Let's create a 2m × n data (measurement) matrix:

$$\mathbf{D} = \begin{bmatrix} \hat{\mathbf{x}}_{11} & \hat{\mathbf{x}}_{12} & \cdots & \hat{\mathbf{x}}_{1n} \\ \hat{\mathbf{x}}_{21} & \hat{\mathbf{x}}_{22} & \cdots & \hat{\mathbf{x}}_{2n} \\ & & \ddots & \\ \hat{\mathbf{x}}_{m1} & \hat{\mathbf{x}}_{m2} & \cdots & \hat{\mathbf{x}}_{mn} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \\ \vdots \\ \mathbf{A}_m \end{bmatrix} \begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \cdots & \mathbf{X}_n \end{bmatrix}$$

(2*m* × n)

cameras (2*m* × 3) — M

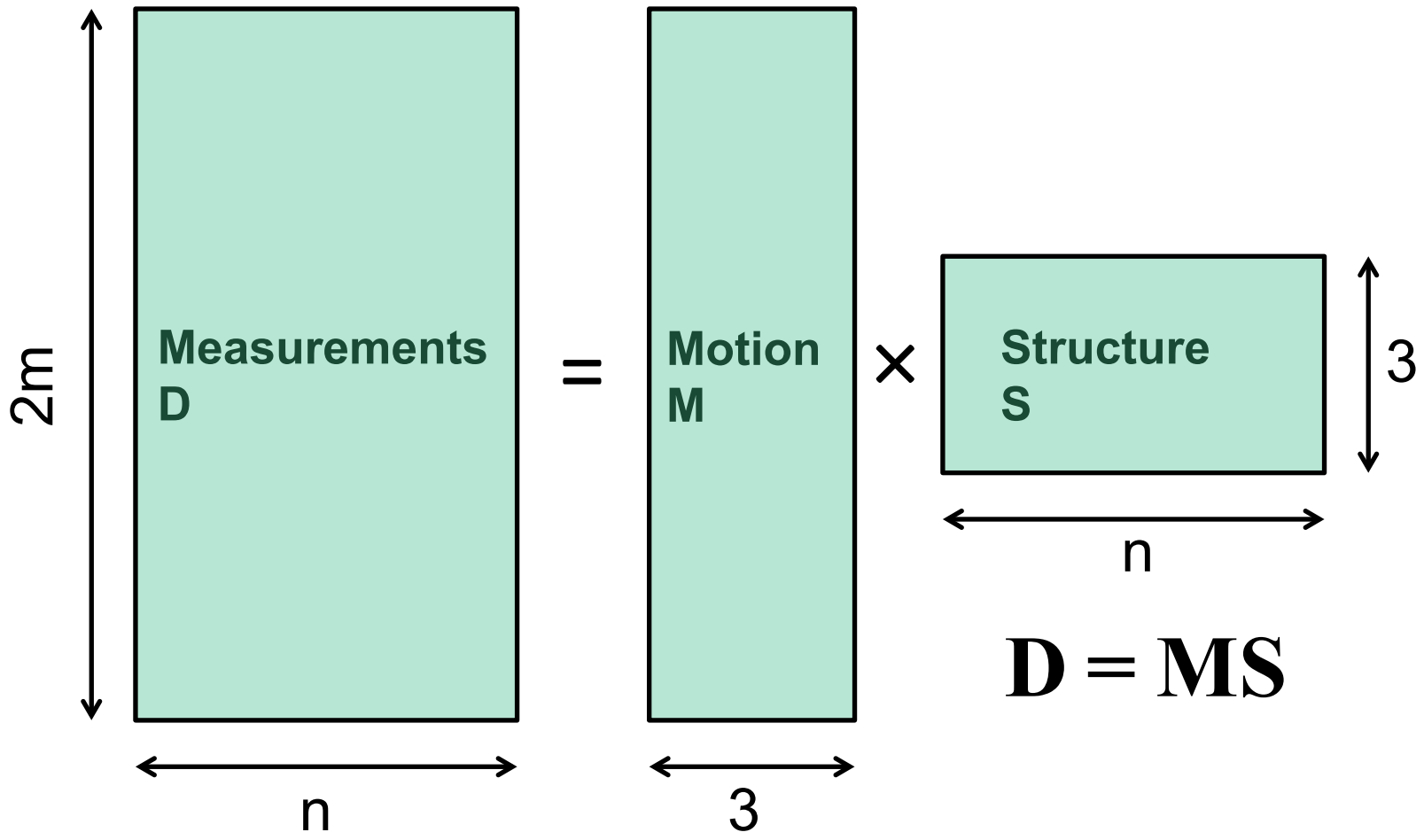
points (3 × *n*) — S

[Eq. 10]

Each $\hat{\mathbf{x}}_{ij}$ entry is a 2x1 vector!
$\mathbf{A}_i$ is 2x3 and $\mathbf{X}_j$ is 3x1

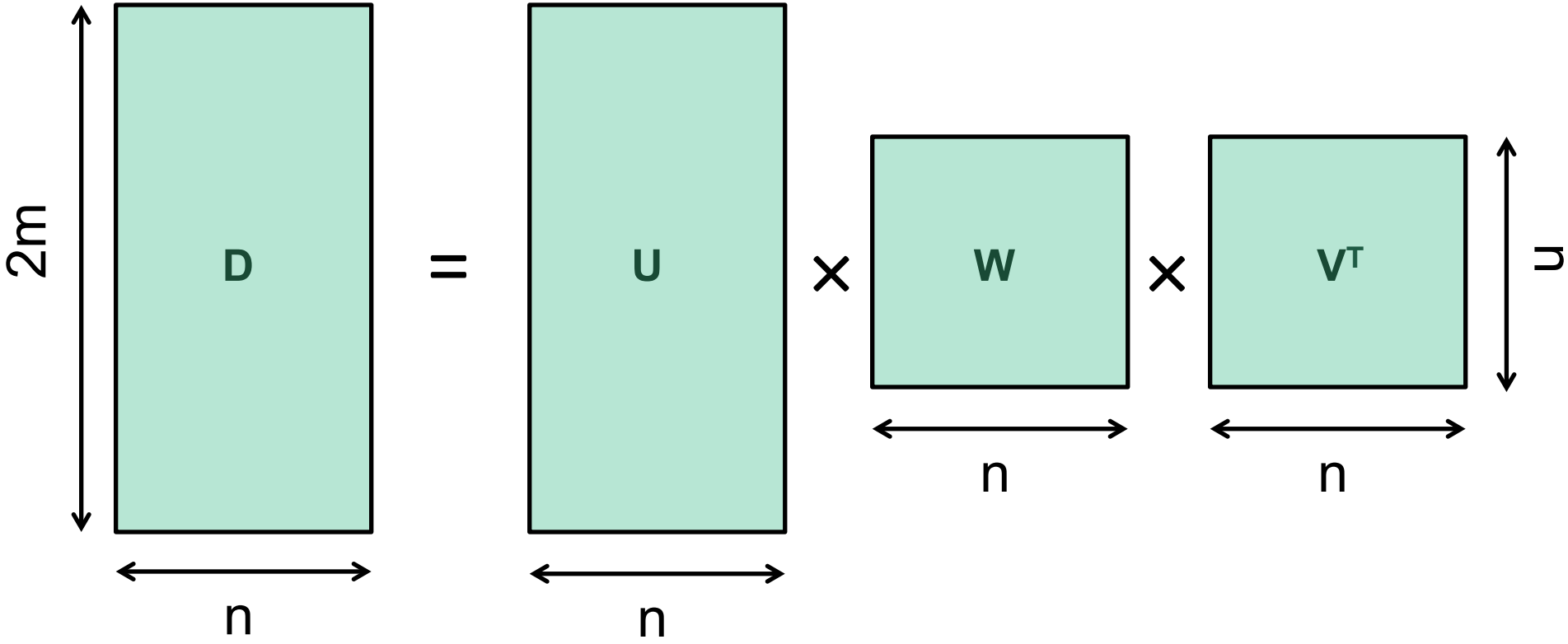
The measurement matrix **D = M S** has rank 3
(it's a product of a 2mx3 matrix and 3xn matrix)

Factorizing the Measurement Matrix



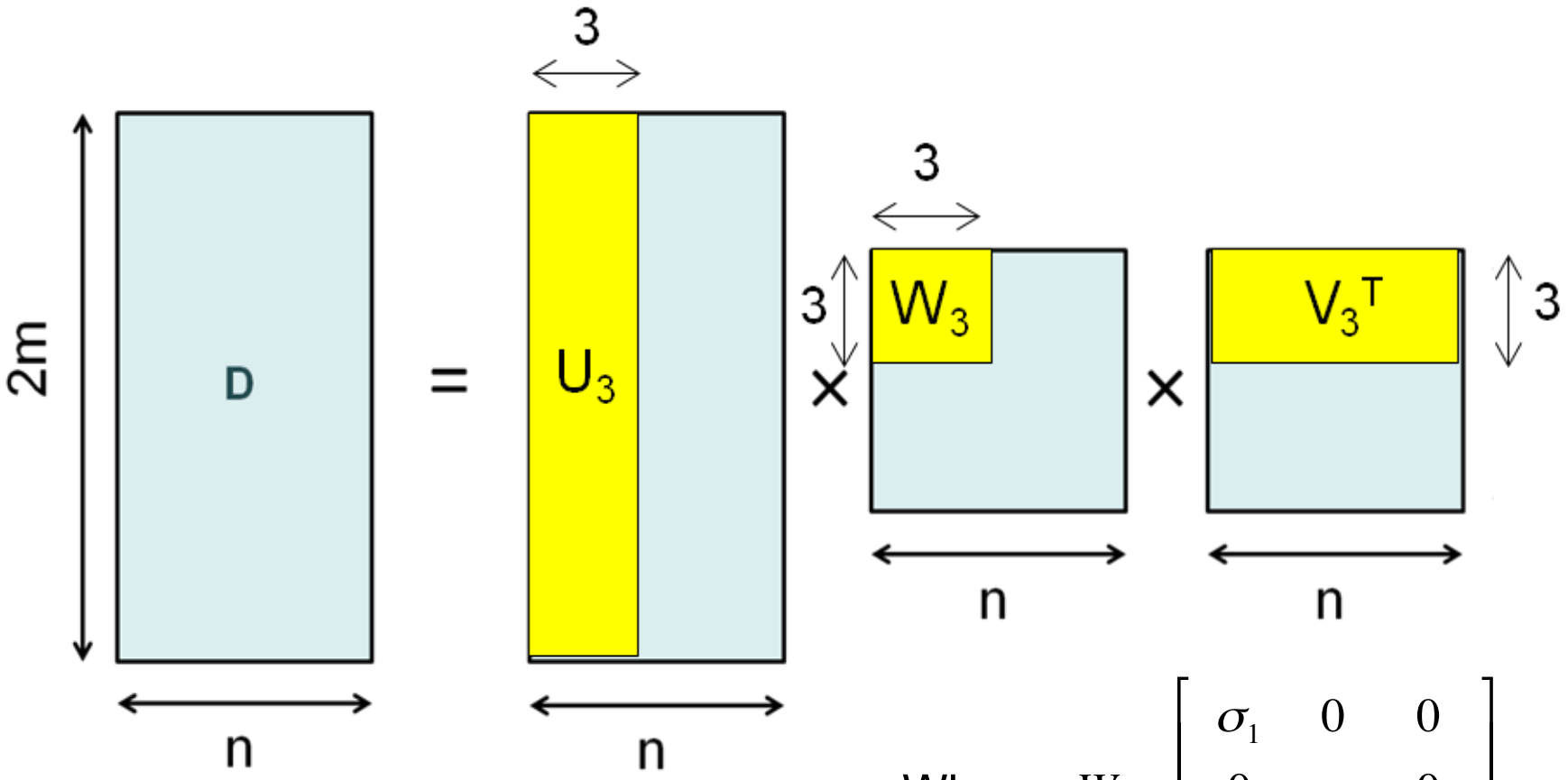
Factorizing the Measurement Matrix

- How to factorize D? By computing the Singular value decomposition of D!



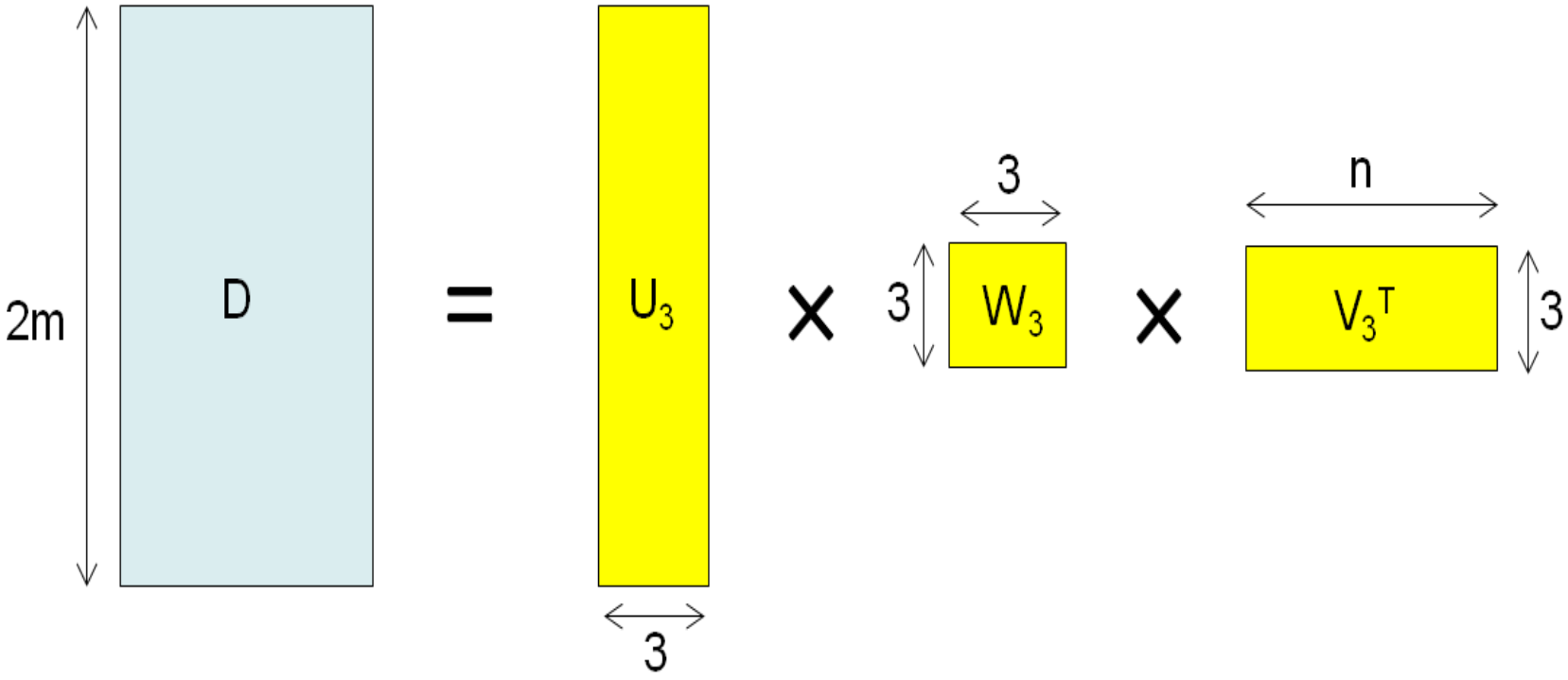
Factorizing the Measurement Matrix

Since rank (D)=3, there are only 3 non-zero singular values σ_1 , σ_2 and σ_3

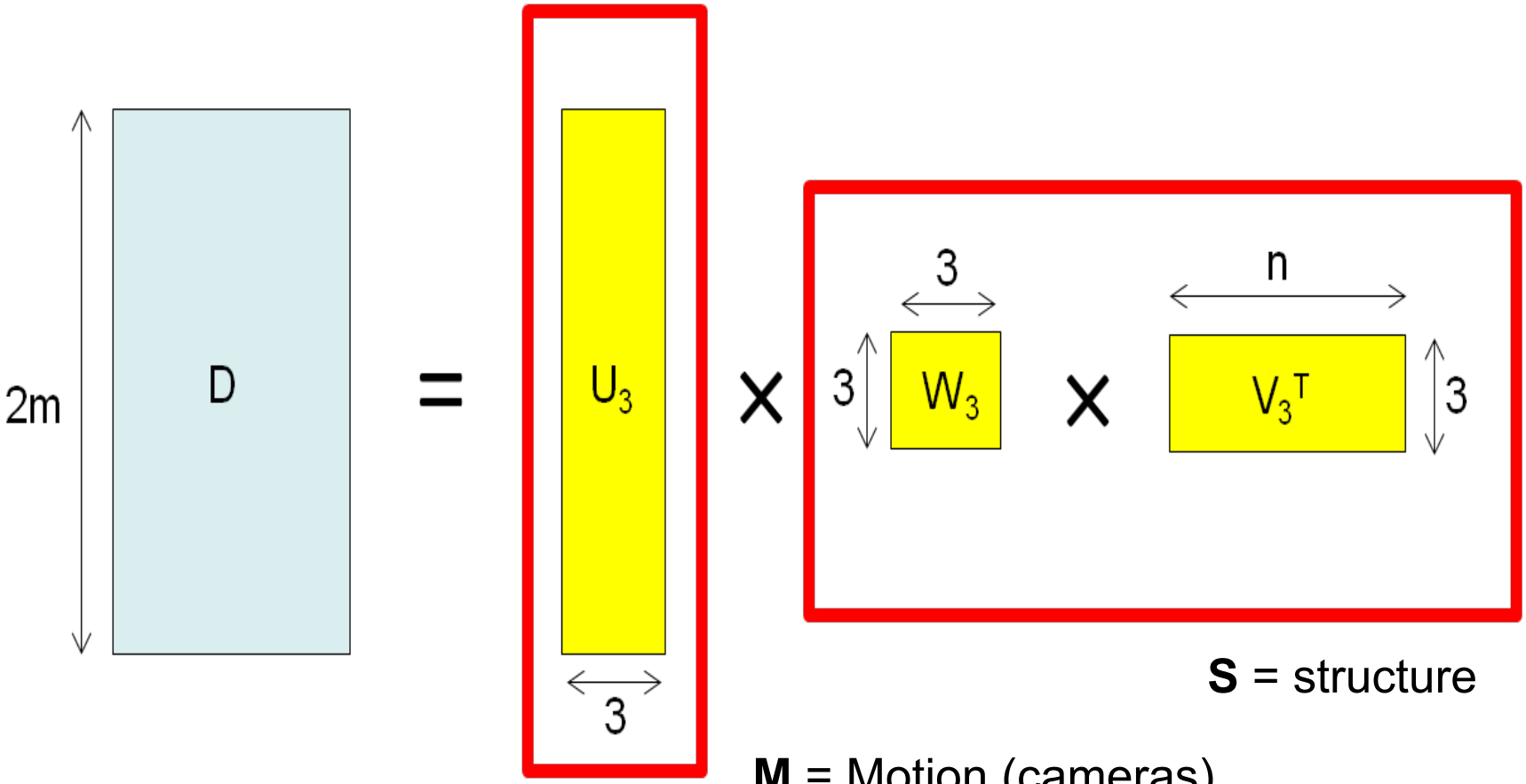


Where $W_3 = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix}$ [Eq. 11]

Factorizing the Measurement Matrix



Factorizing the Measurement Matrix



S = structure

M = Motion (cameras)

$$\mathbf{D} = \mathbf{U}_3 \mathbf{W}_3 \mathbf{V}_3^T = \mathbf{U}_3 (\mathbf{W}_3 \mathbf{V}_3^T) = \mathbf{M}\,\mathbf{S} \quad \text{[Eq. 12]}$$

Factorizing the Measurement Matrix

$$\mathbf{D} = \mathbf{U}_3 \mathbf{W}_3 \mathbf{V}_3^T = \mathbf{U}_3 (\mathbf{W}_3 \mathbf{V}_3^T) = \mathbf{M}\,\mathbf{S} \quad \text{[Eq. 12]}$$

What is the issue here?

D has rank>3 because of:

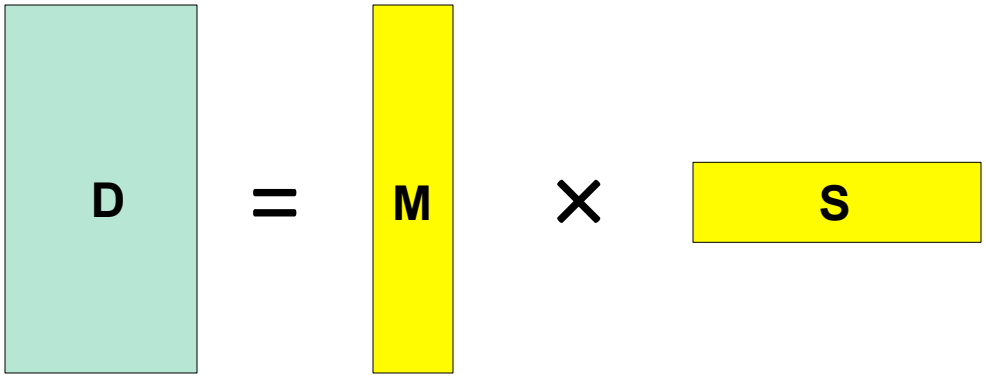
- measurement noise
- affine approximation

Theorem: When $\mathbf{D}$ has a rank greater than 3, $\mathbf{U}_3\mathbf{W}_3\mathbf{V}_3^T$ is the best possible rank-3 approximation of **D** in the sense of the Frobenius norm.

$$\mathbf{D} = \mathbf{U}_3\mathbf{W}_3\mathbf{V}_3^T \quad \begin{cases} \mathbf{M} \approx \mathbf{U}_3 \\ \mathbf{S} \approx \mathbf{W}_3\mathbf{V}_3^T \end{cases}$$

$$\|A\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|^2} = \sqrt{\sum_{i=1}^{\min\{m,n\}}\sigma_i^2}$$

Affine Ambiguity



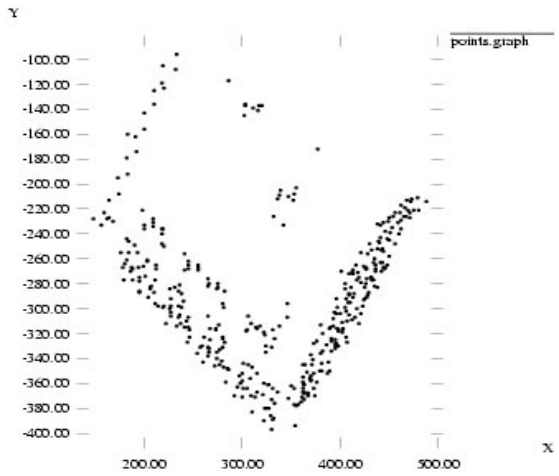
- The decomposition is not unique. We get the same **D** by using any 3×3 matrix **C** and applying the transformations **M** → **MC**, S →C^{-1}S

- We can enforce some Euclidean constraints to resolve this ambiguity (more on next lecture!)

Reconstruction results

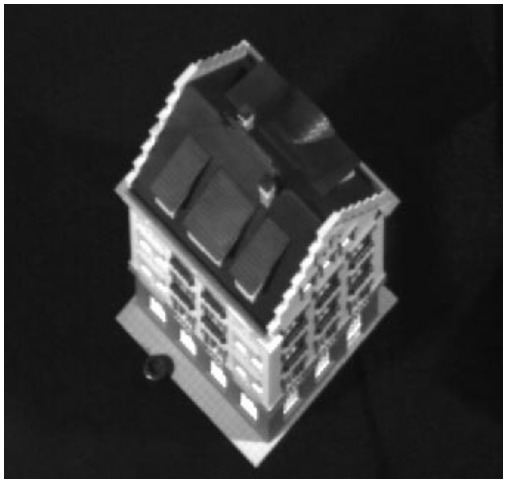
1

60



120

150



C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization method. *IJCV*, 9(2):137-154, November 1992.

Next lecture

Multiple view geometry

Perspective structure from Motion