# Stereo Vision (Correspondences)

EECS 598-08 Fall 2014
Foundations of Computer Vision

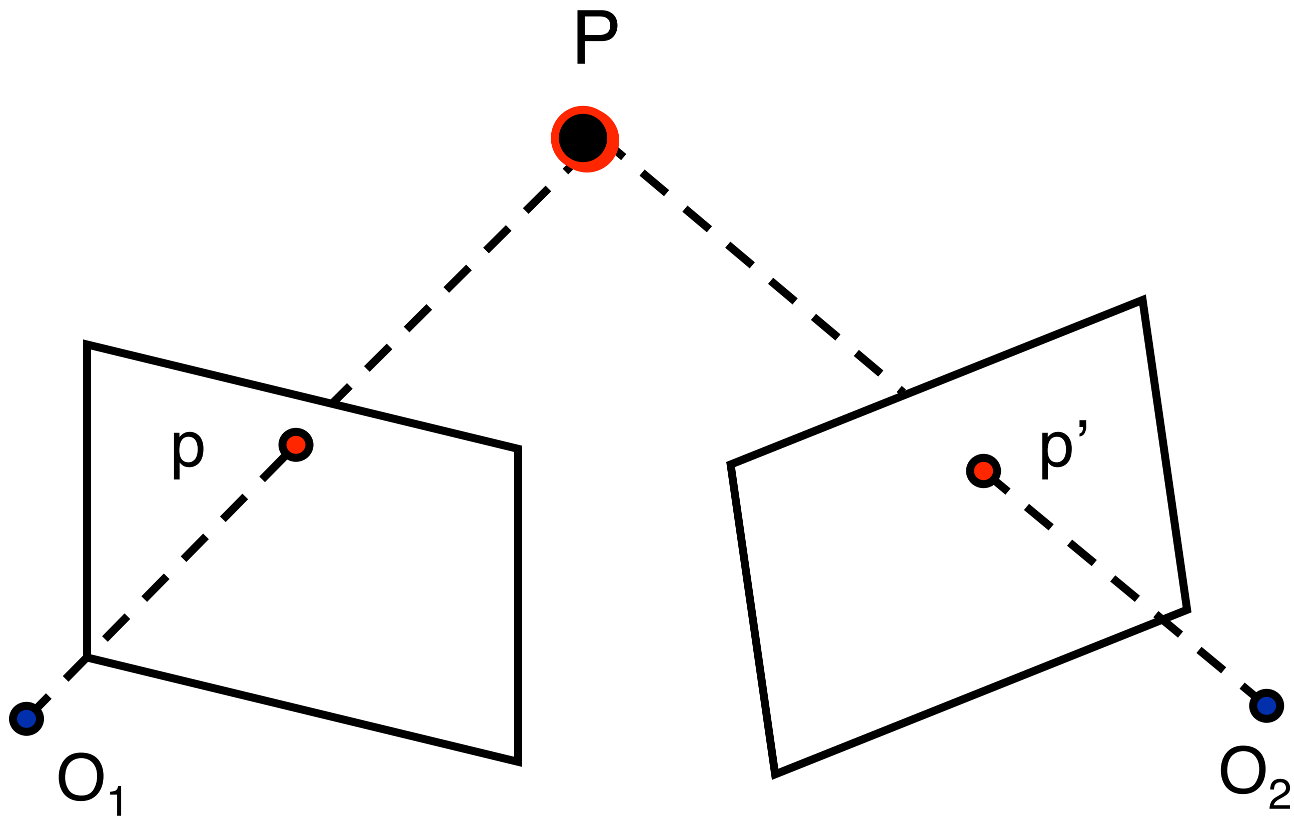
Instructor: Jason Corso (jjcorso)
web.eecs.umich.edu/~jjcorso/t/598F14

**Readings:** FP 7; SZ 11; TV 7
**Date:** 10/27/14

# Plan

- Stereo vision
- Rectification
- Correspondence problem
- Active stereo vision systems

# Stereo Vision

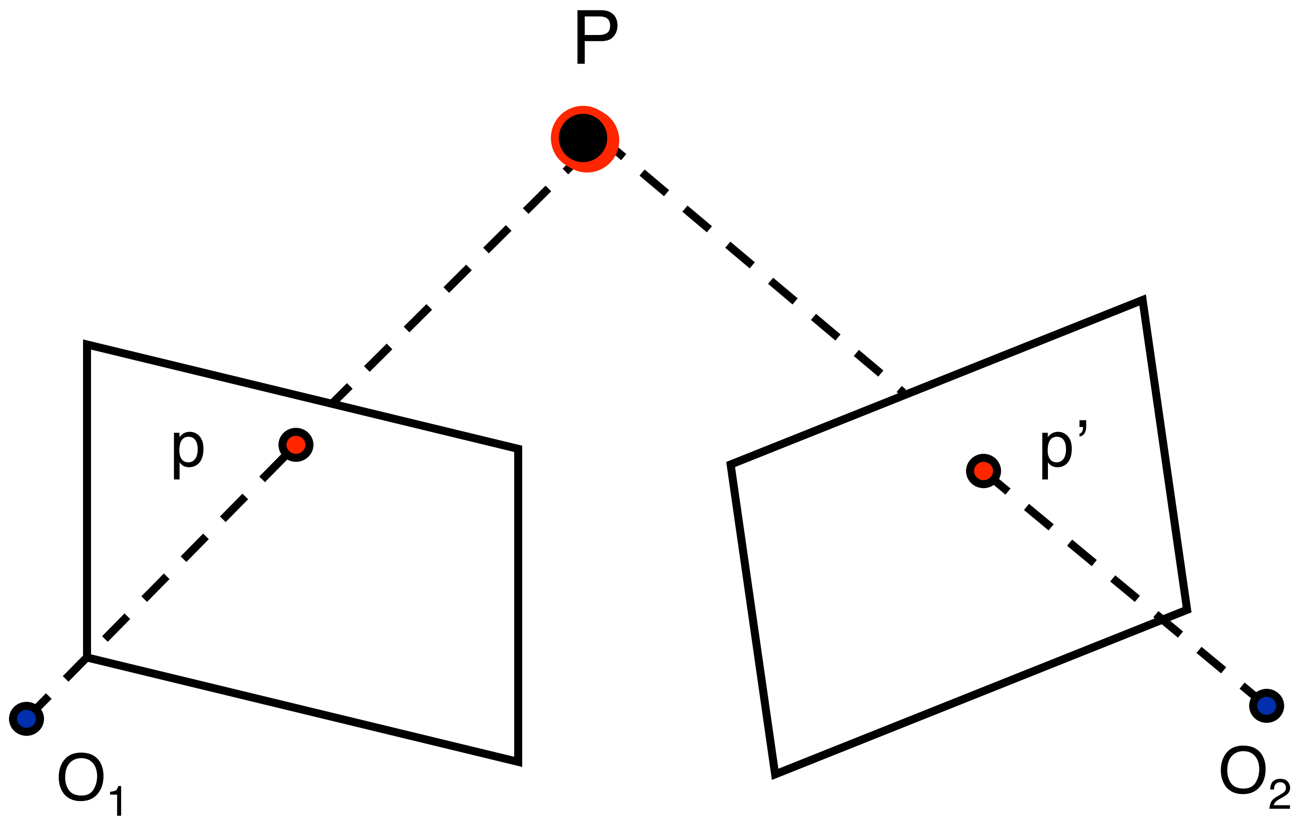



Goal: estimate the position of P given the observation of P from two view points

Assumptions: known camera parameters and position (K, R, T)

Slide source: S. Savarese.

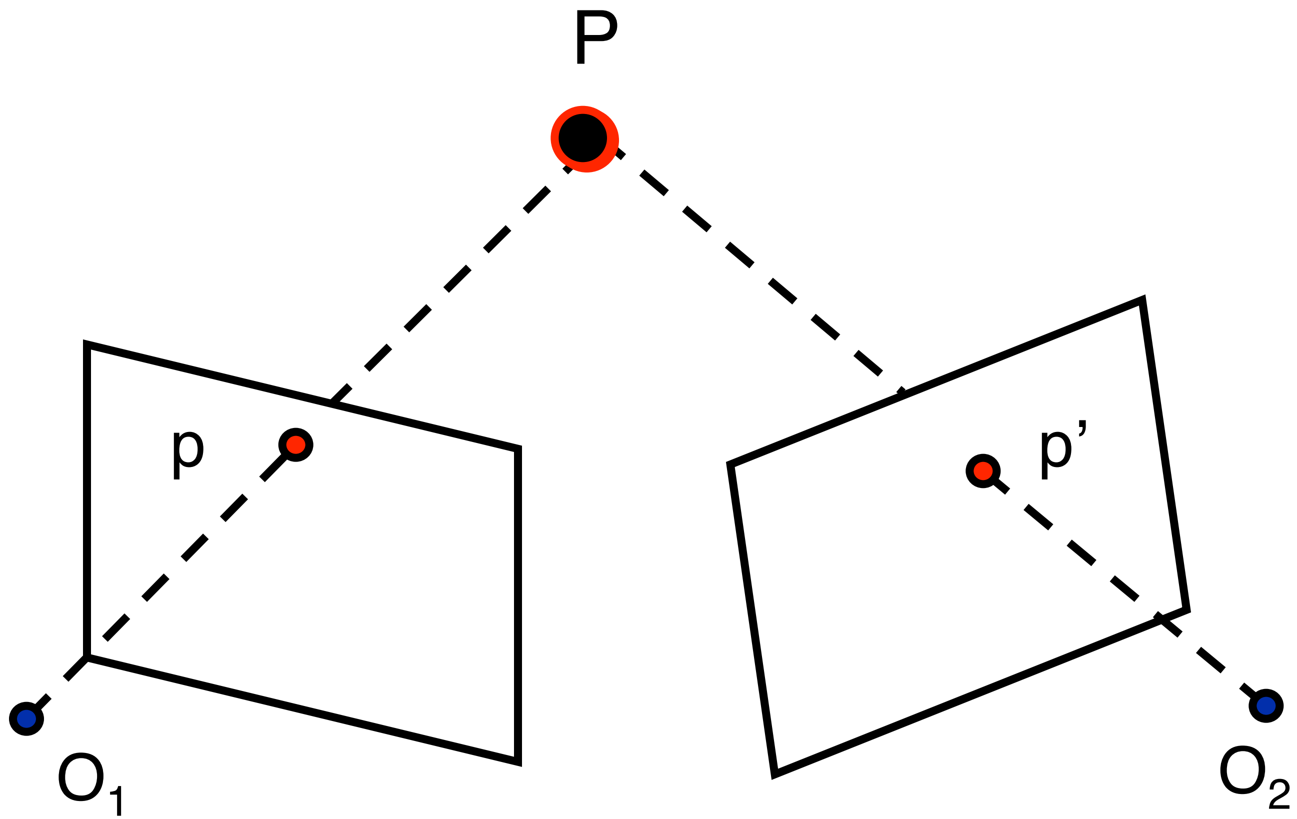
# Stereo Vision




Subgoals:
- Solve the correspondence problem
- Use corresponding observations to triangulate

Slide source: S. Savarese.

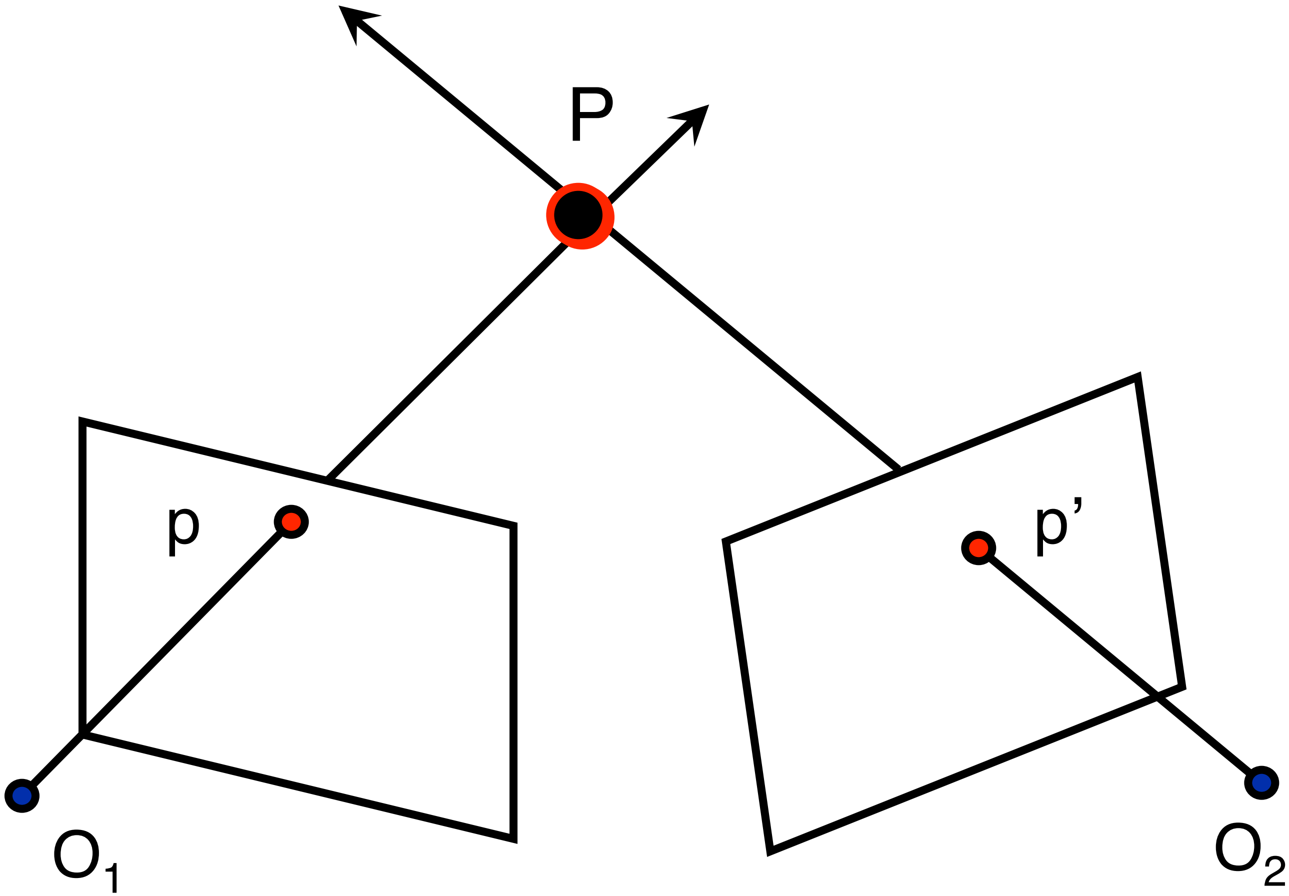
# The Correspondence Problem




• Given a point in 3d, discover corresponding observations in left and right images

Slide source: S. Savarese.

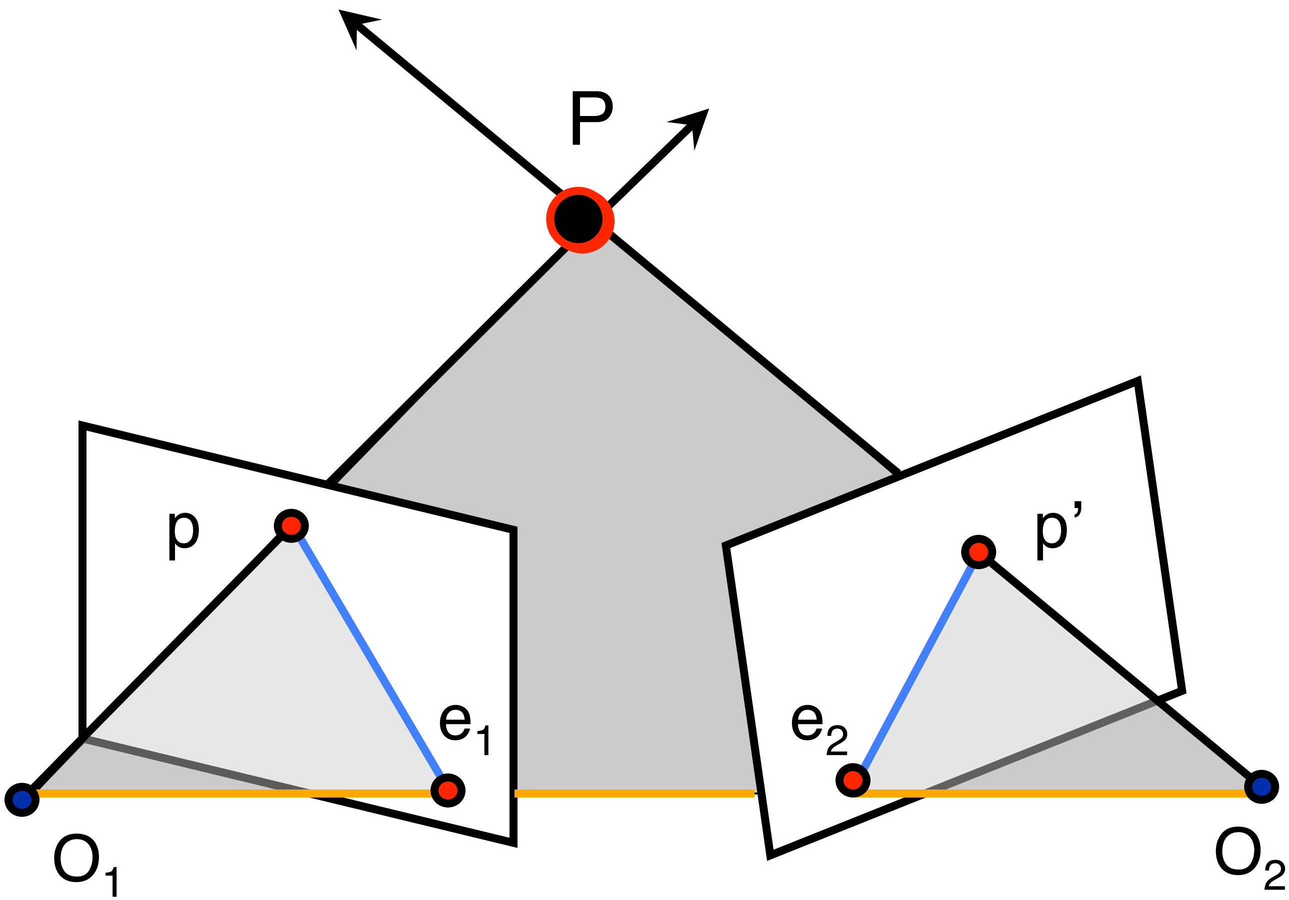
# Triangulation




- Intersecting the two lines of sight gives rise to P

Slide source: S. Savarese.

# Epipolar Geometry




- Epipolar Plane
- Baseline
- Epipolar Lines

- Epipoles $e_1$, $e_2$

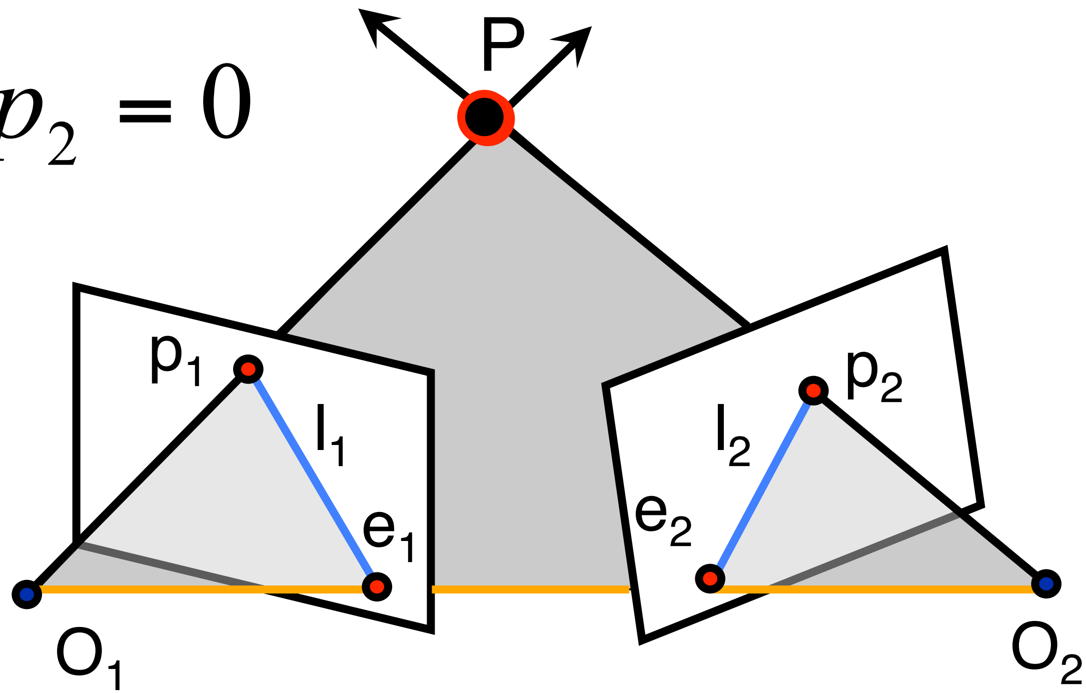
  = intersections of baseline with image planes

  = projections of the other camera center

Slide source: S. Savarese.

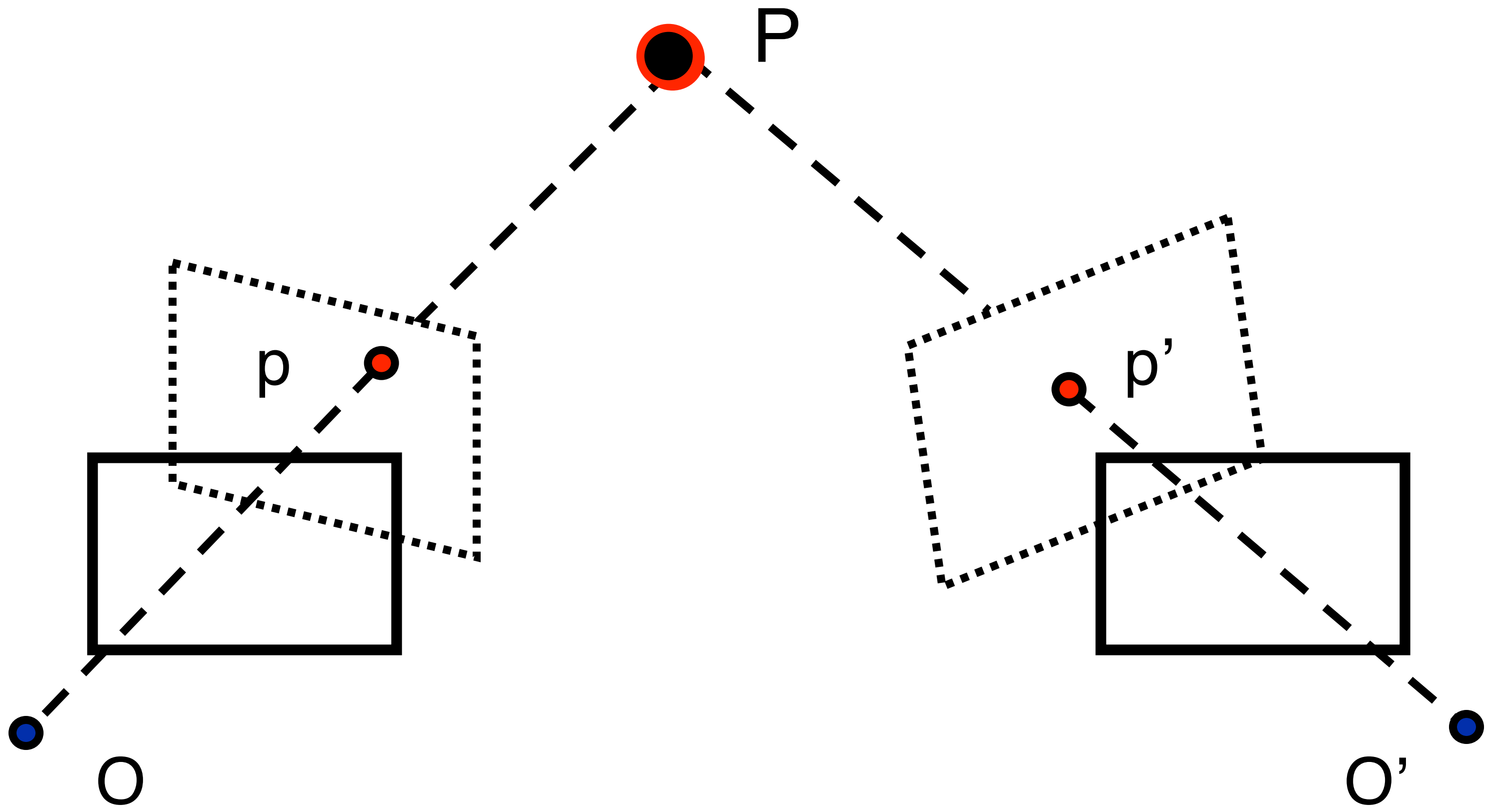
# The Epipolar Constraint

$$p_1^{\,T} \cdot E\, p_2 = 0$$




- $E\,p_2$ is the epipolar line associated with $p_2$ ($l_1 = E\,p_2$)
- $E^T p_1$ is the epipolar line associated with $p_1$ ($l_2 = E^T p_1$)
- $E\,e_2 = 0$ and $E^T e_1 = 0$
- E is 3x3 matrix; 5 DOF
- E is singular (rank two)

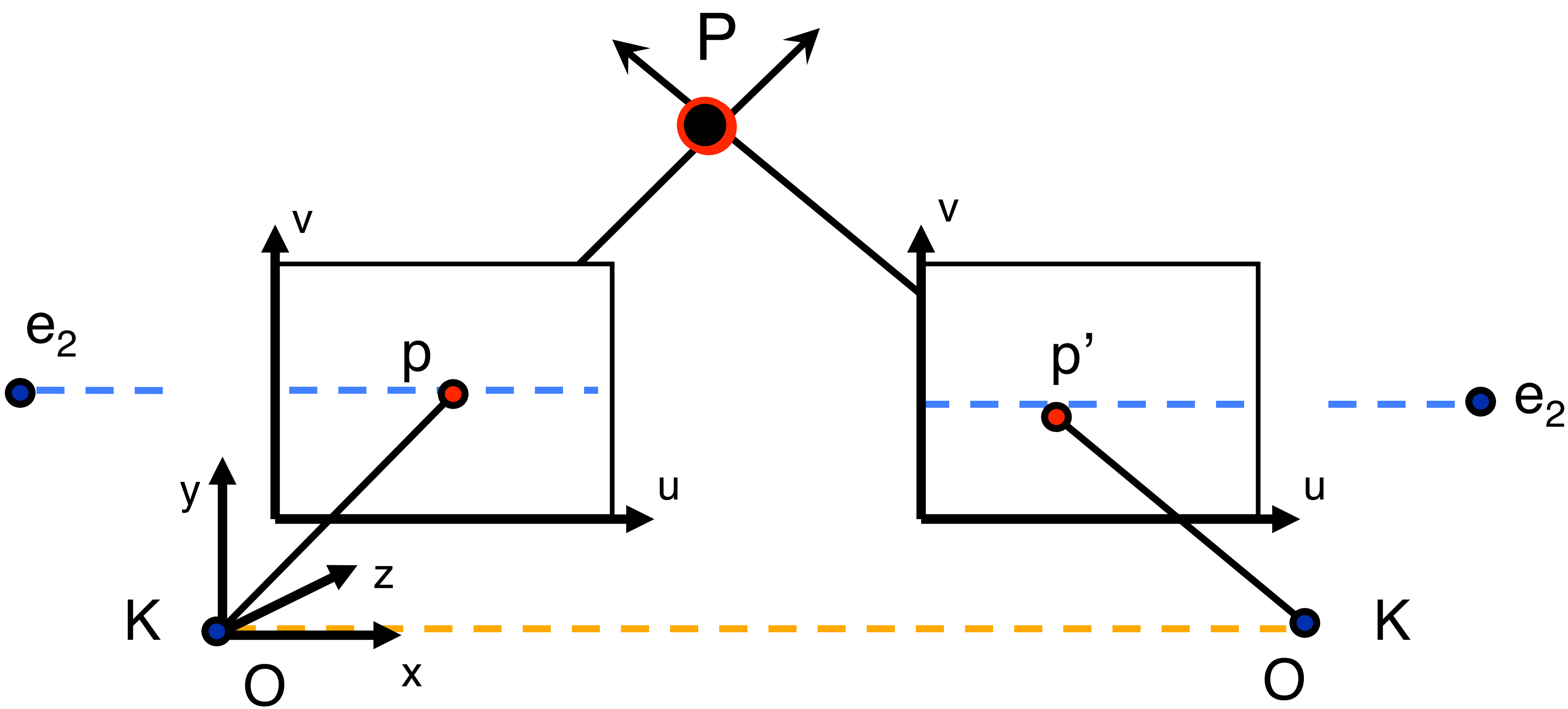
# Parallel Image Planes




• When views are **parallel** both correspondence and triangulation become much easier!

Slide source: S. Savarese.

# Parallel Image Planes

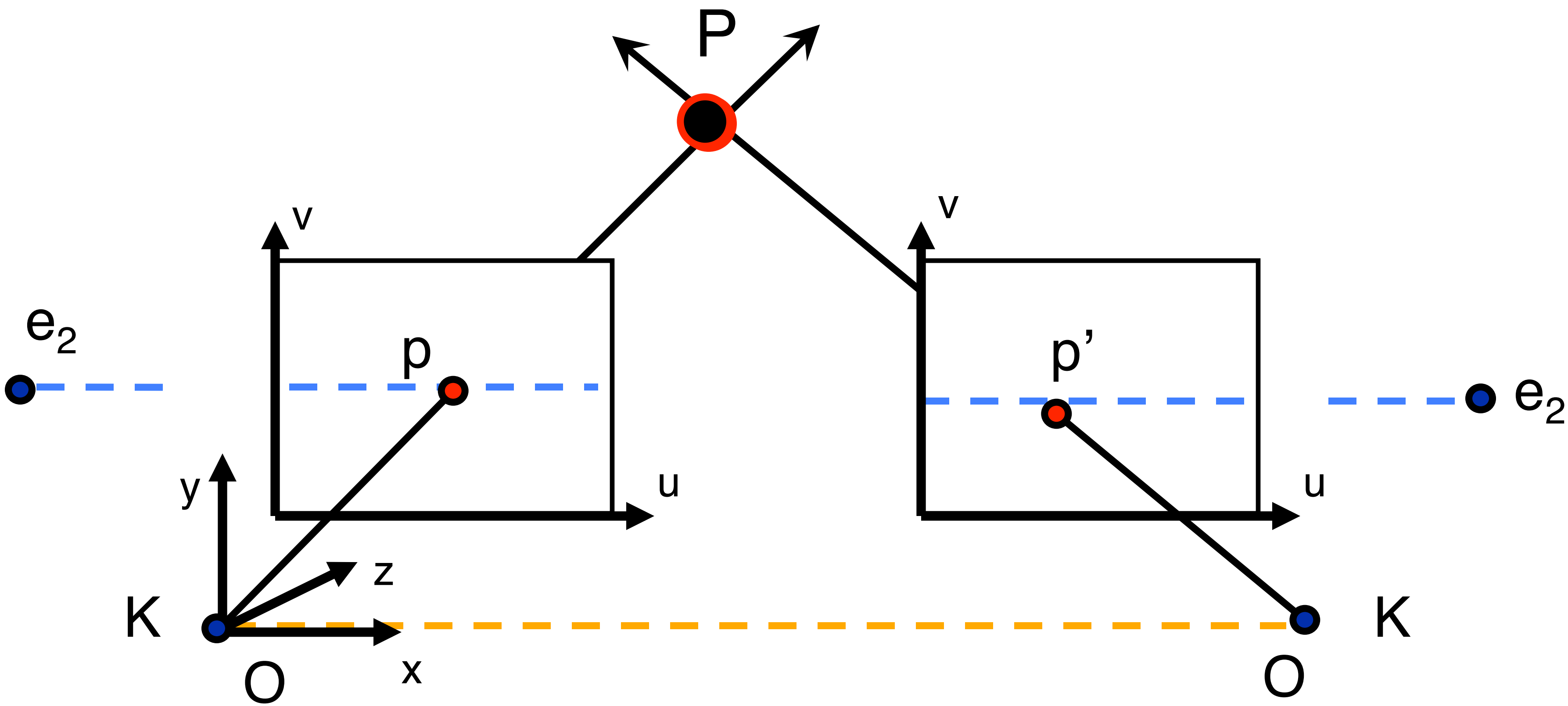



- Parallel epipolar lines
- Epipoles at infinity
- v = v'

Rectification: making two images "parallel"

Slide source: S. Savarese.

# Parallel Image Planes




$K_1 = K_2$ = known

x parallel to $O_1O_2$

$$E = [t_\times]R$$

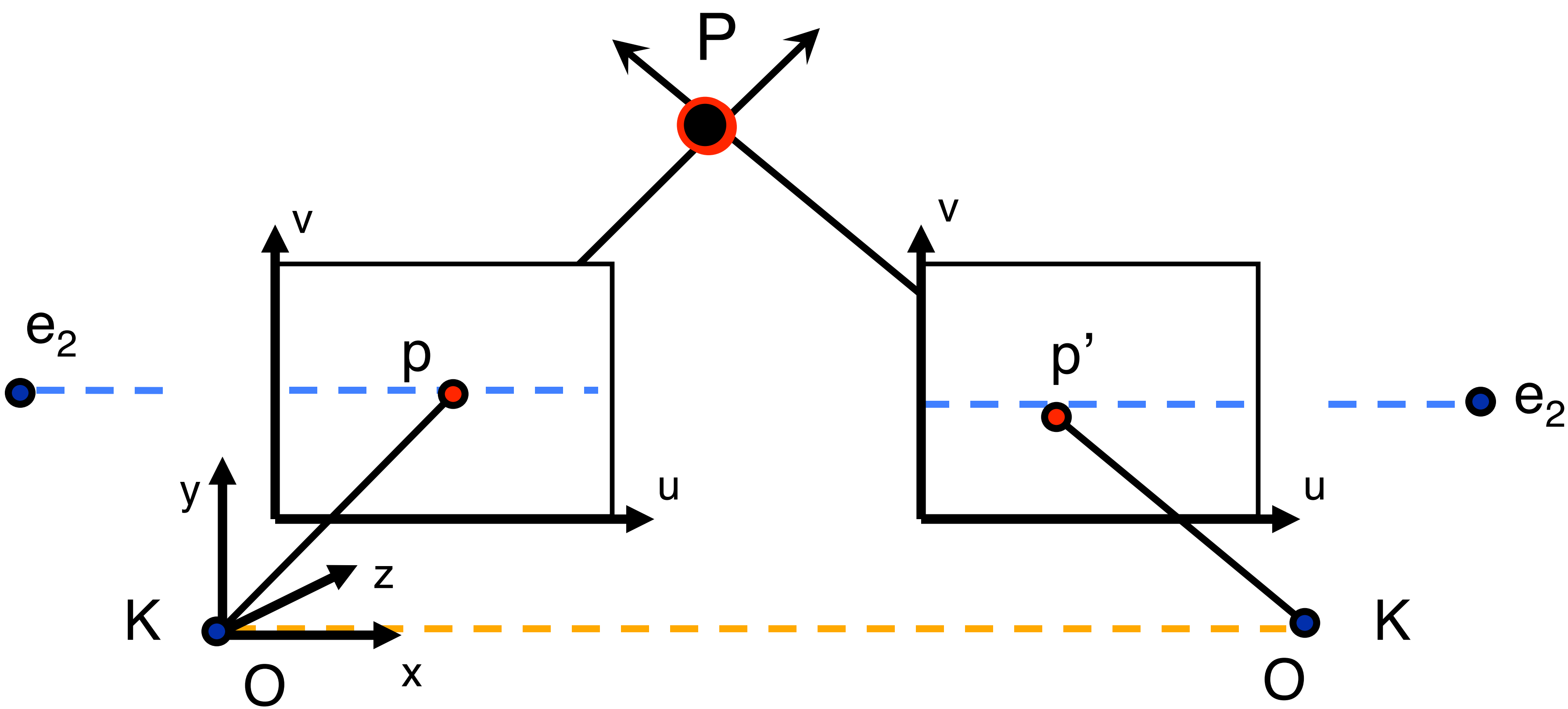
Slide source: S. Savarese.

# Cross product as matrix multiplication

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

Slide source: S. Savarese.

# Parallel Image Planes




$K_1$=$K_2$ = known

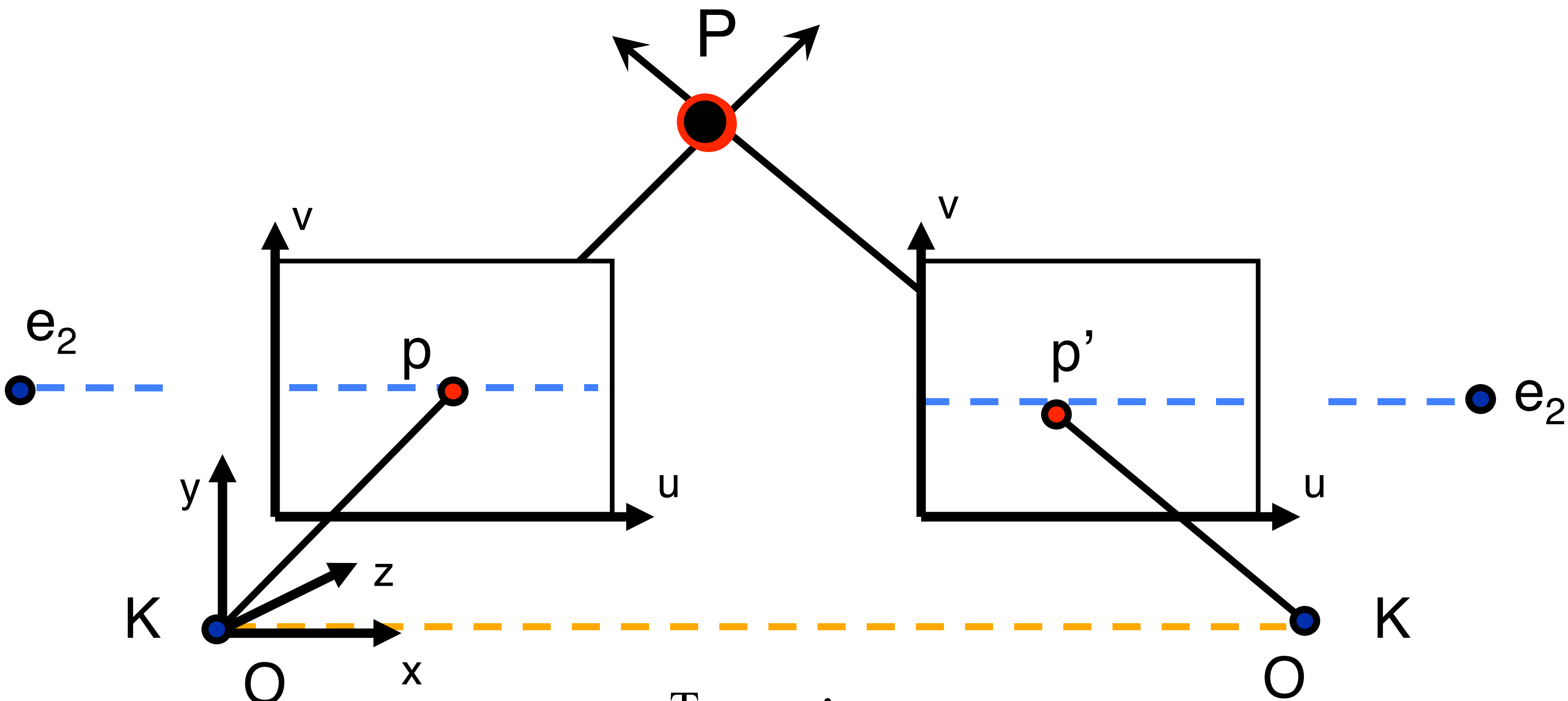
x parallel to $O_1O_2$

$$E = [t_\times]R = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -T \\ 0 & T & 0 \end{bmatrix}$$

→ v = v'?

Slide source: S. Savarese.

# Parallel Image Planes

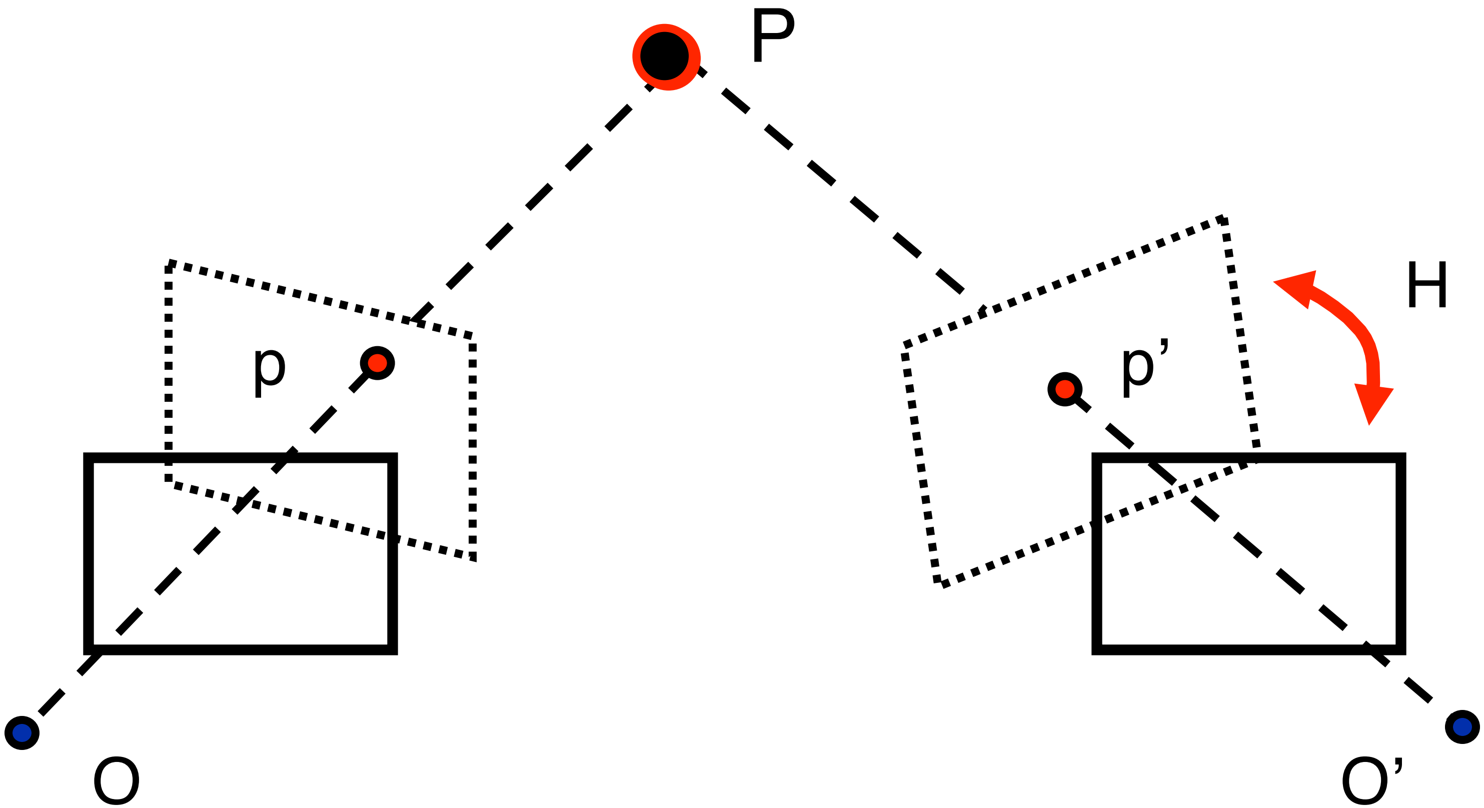



$$p^T E p' = 0$$

$$\begin{pmatrix} u & v & 1 \end{pmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -T \\ 0 & T & 0 \end{bmatrix} \begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} = 0 \qquad \begin{pmatrix} u & v & 1 \end{pmatrix} \begin{pmatrix} 0 \\ -T \\ Tv' \end{pmatrix} = 0 \qquad Tv = Tv'$$

Slide source: S. Savarese.

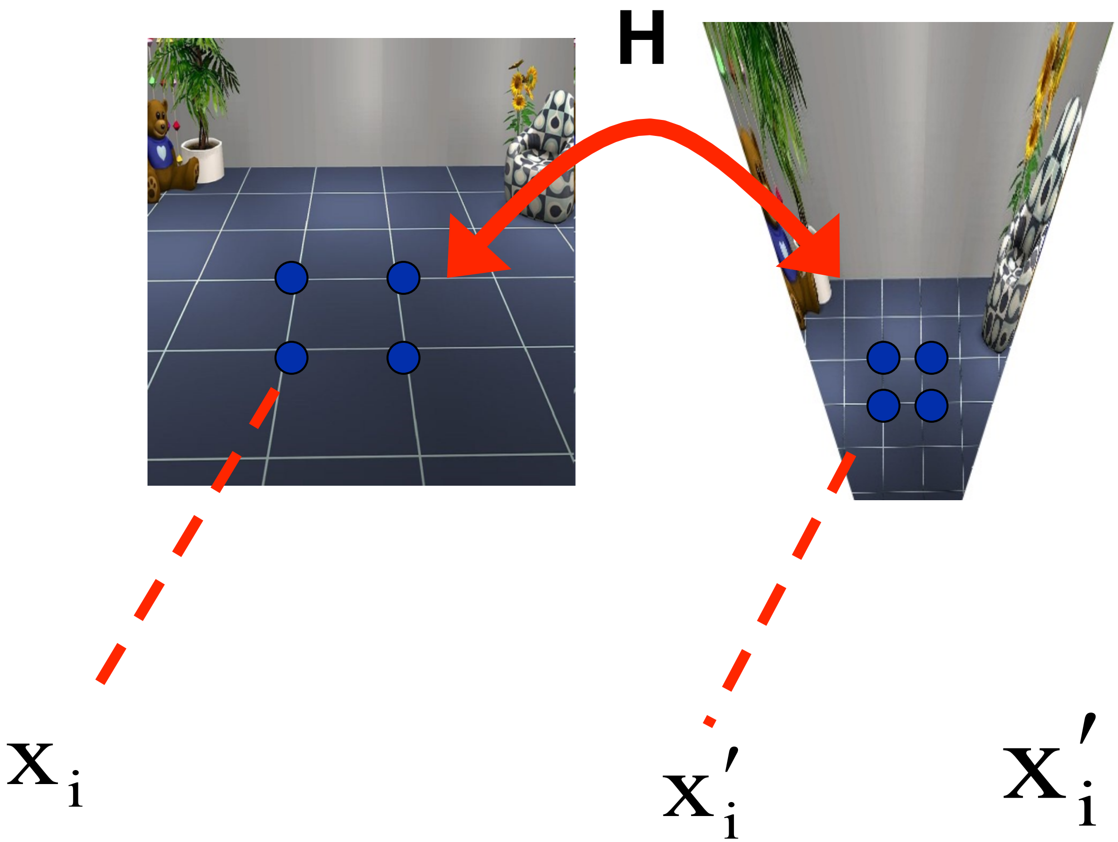
# Making Image Planes Parallel




GOAL: Estimate the perspective transformation H that makes the images parallel

Slide source: S. Savarese.

# Making the Image Planes Parallel
## The Projective Transformation

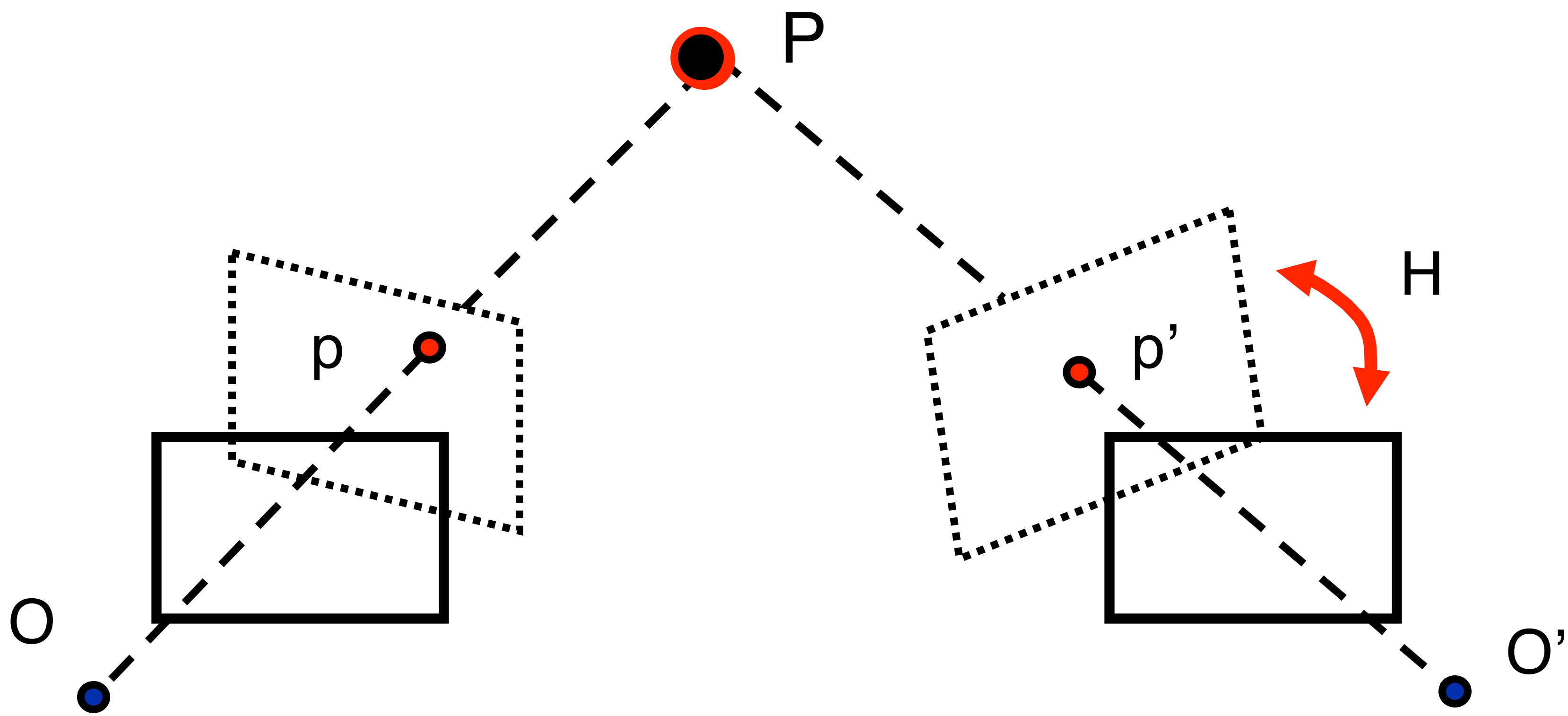



$$x'_i = H\,x_i$$

Now we don’t have the destination image ☹

Slide source: S. Savarese.

# Making the Image Planes Parallel




**GOAL:** Estimate the perspective transformation H that makes images parallel

Impose v'=v

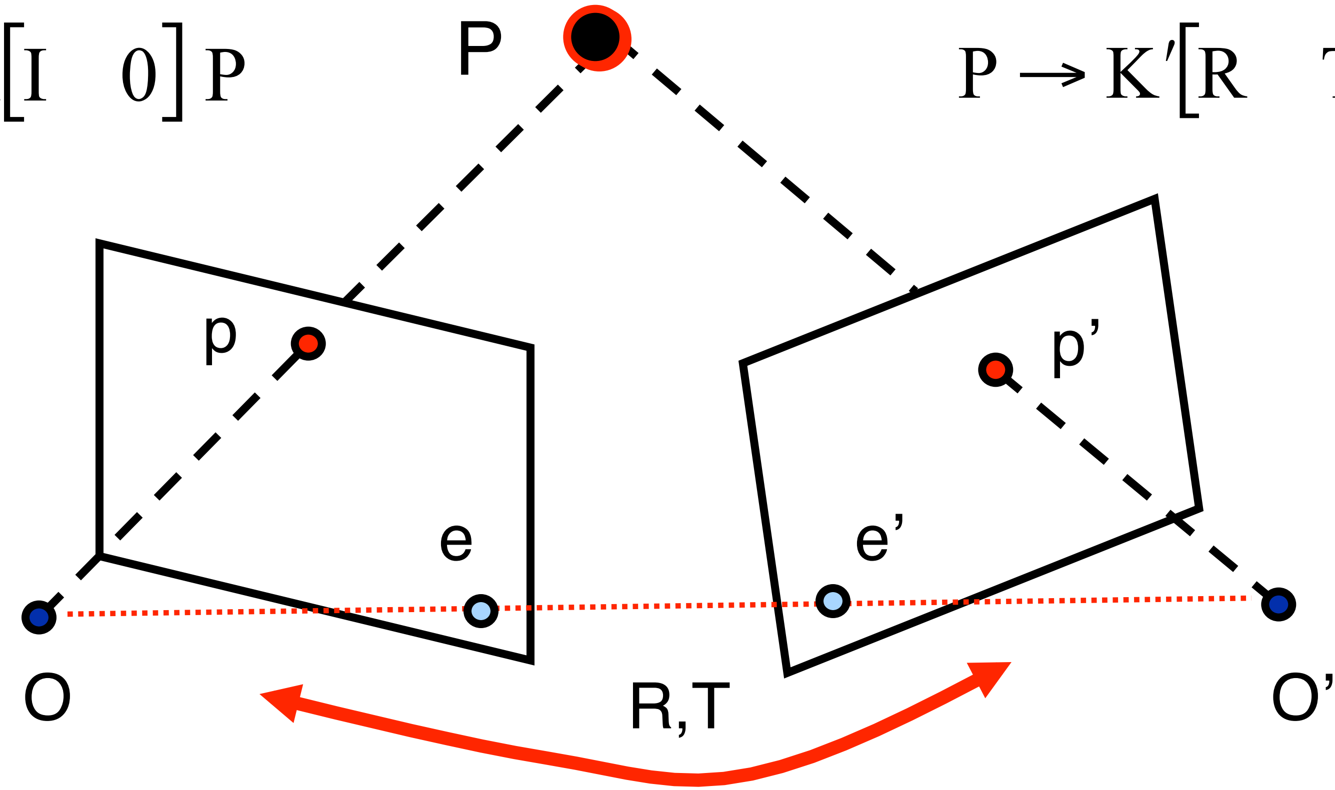
- This leaves degrees of freedom for determining H
- If an inappropriate H is chosen, severe projective distortions on image take place
- We impose a number of restrictions while computing H

Slide source: S. Savarese.

# Making the Image Planes Parallel

$$P \to K[I \quad 0]\, P$$

$$P \to K'[R \quad T]\, P$$

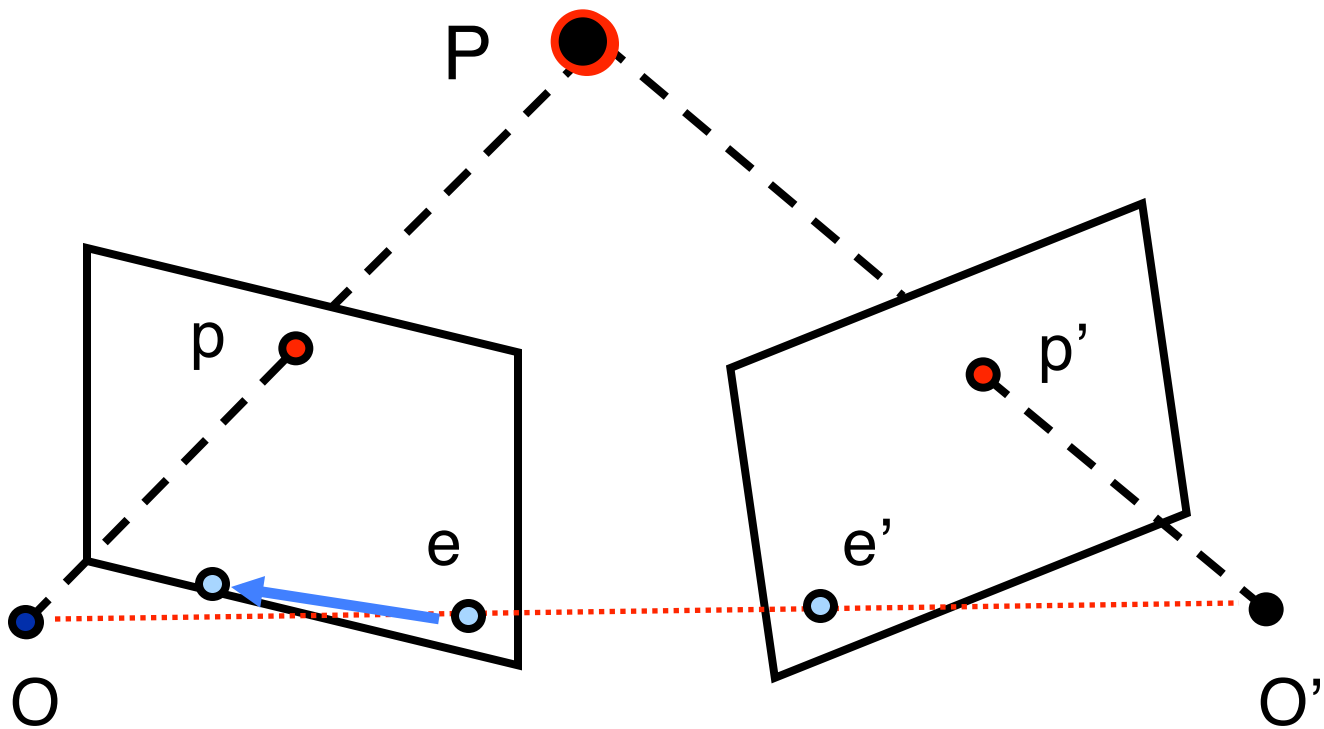



0. Compute epipoles

$$e = [e_1 \quad e_2 \quad 1]^T = K\, R^T\, T \qquad e' = K'\, T$$

Slide source: S. Savarese.

# Making the Image Planes Parallel



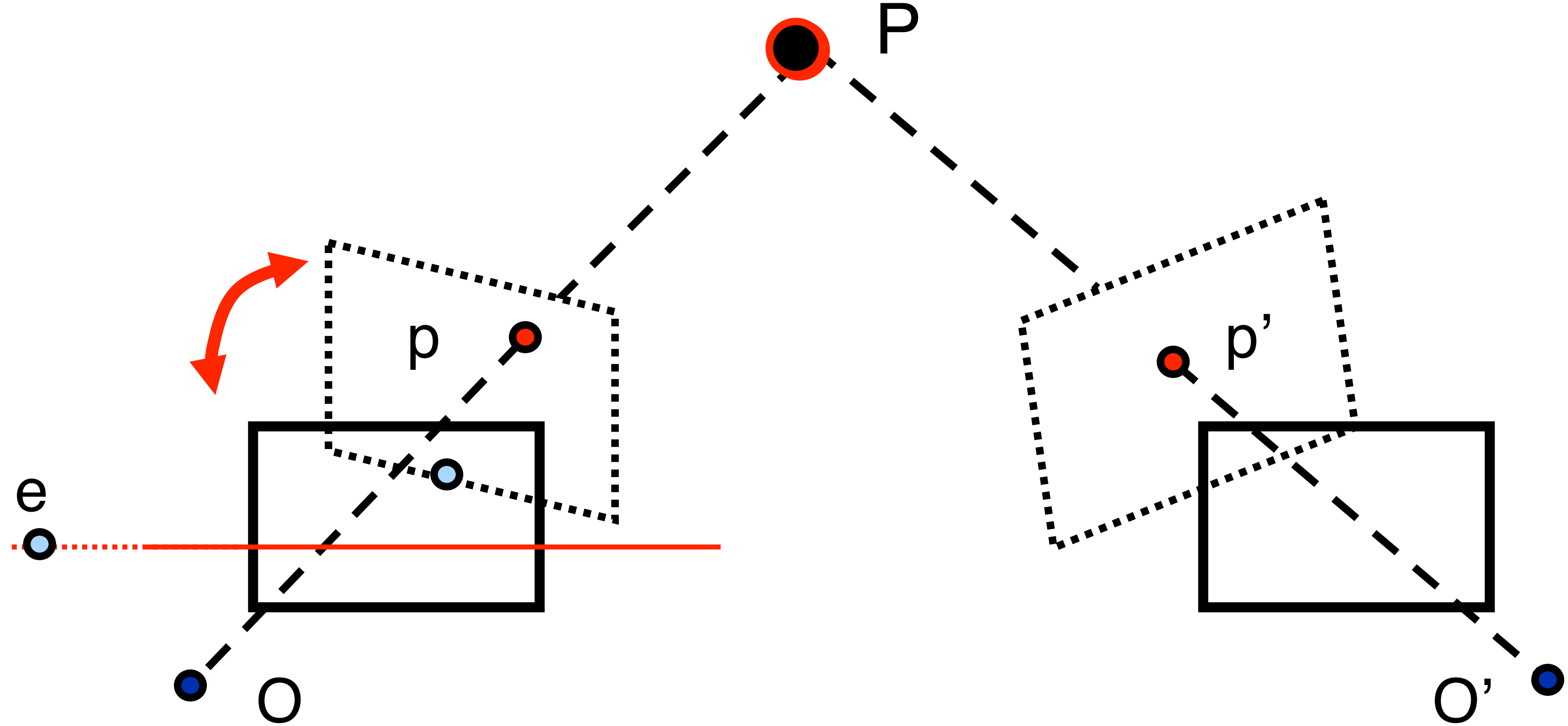

1. Map e to the x-axis at location $[1,0,1]^T$ (normalization)

$$e = [e_1 \quad e_2 \quad 1]^T \rightarrow [1 \quad 0 \quad 1]^T$$

$$H_1 = R_H T_H$$

Slide source: S. Savarese.

# Making the Image Planes Parallel




2. Send epipole to infinity:

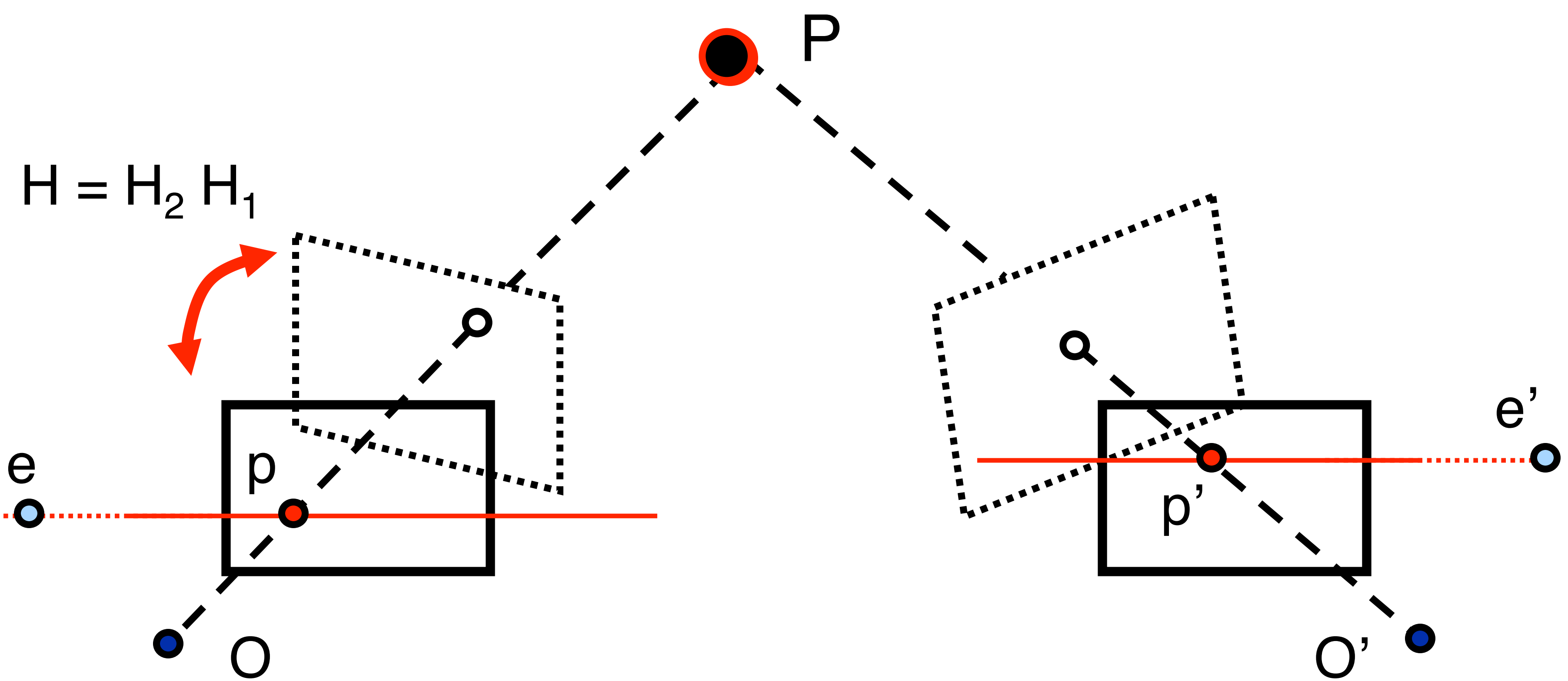
$$e = \begin{bmatrix} 1 & 0 & 1 \end{bmatrix}^T \rightarrow \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T$$

$$H_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}$$

Minimizes the distortion in a neighborhood (approximates id. mapping)

Slide source: S. Savarese.

# Making the Image Planes Parallel



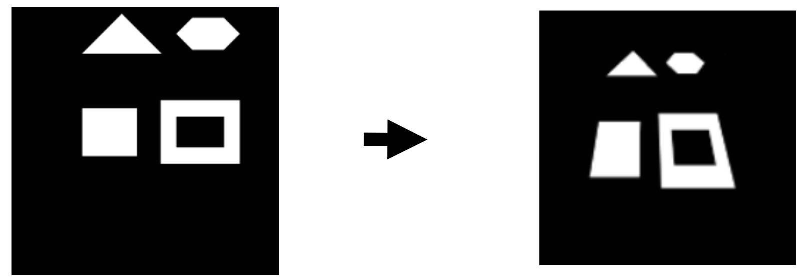

3. Define: $H = H_2 H_1$

4. Align epipolar lines

Slide source: S. Savarese.

# Projective Transformation of a Line

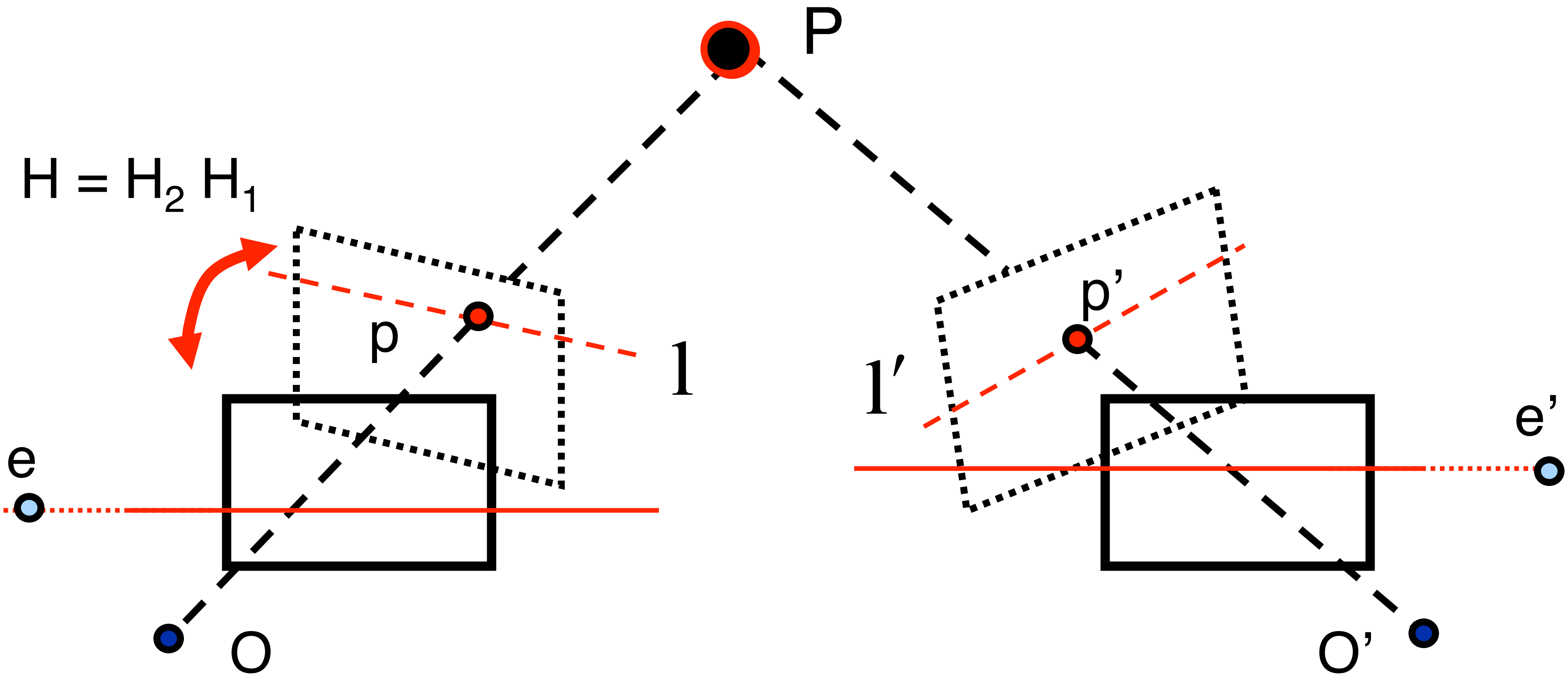
$$H = \begin{bmatrix} A & t \\ v & b \end{bmatrix}$$



$$l' \rightarrow H^{-T}\, l$$

Slide source: S. Savarese.

# Making the Image Planes Parallel




3. Define: $H = H_2 H_1$

4. Align epipolar lines

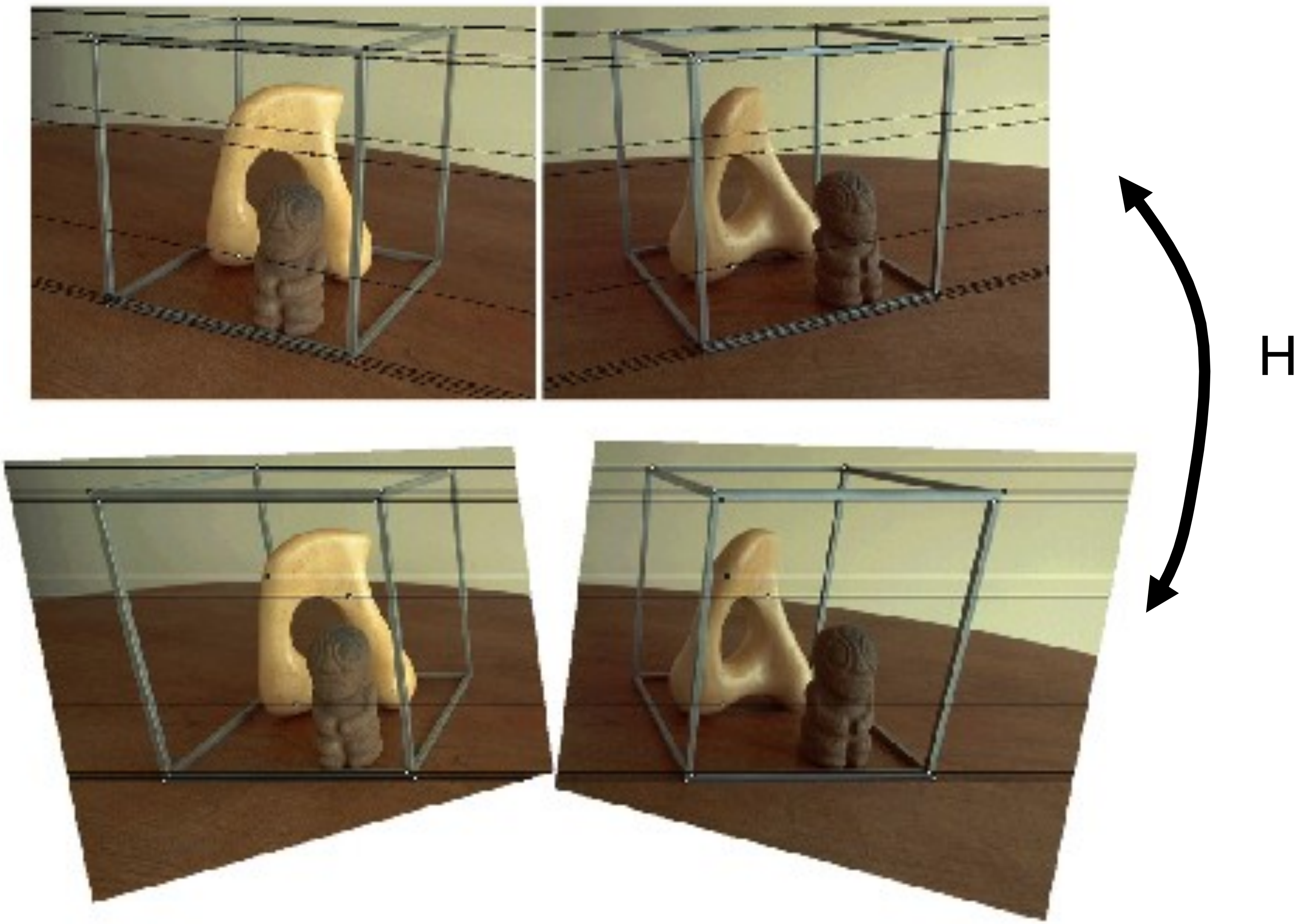
$$\overline{H}'^{-T} l' = \overline{H}^{-T} \, l$$

These are called matched pair of transformation

[HZ] Chapters: 11 (sec. 11.12)

Slide source: S. Savarese.

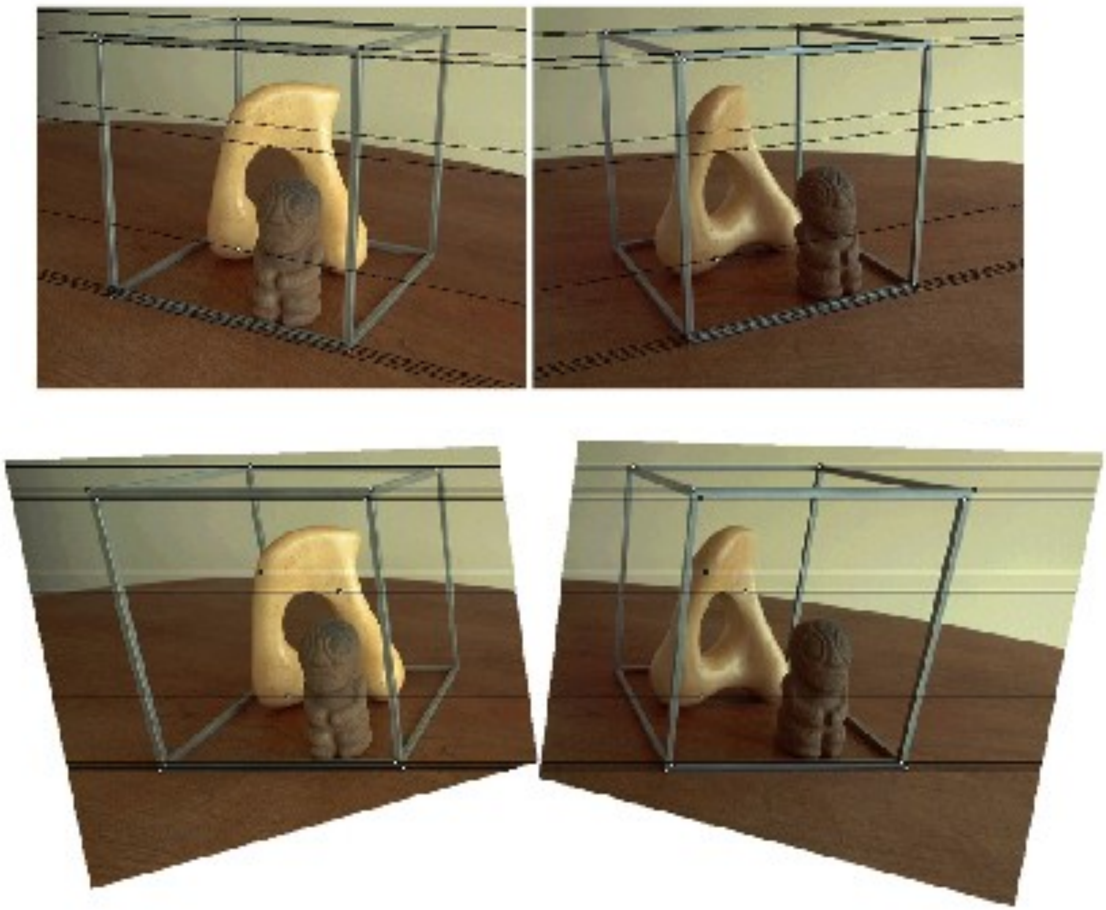
# Making the Image Planes Parallel




Courtesy figure S. Lazebnik

Slide source: S. Savarese.

# Why is Rectification Useful?

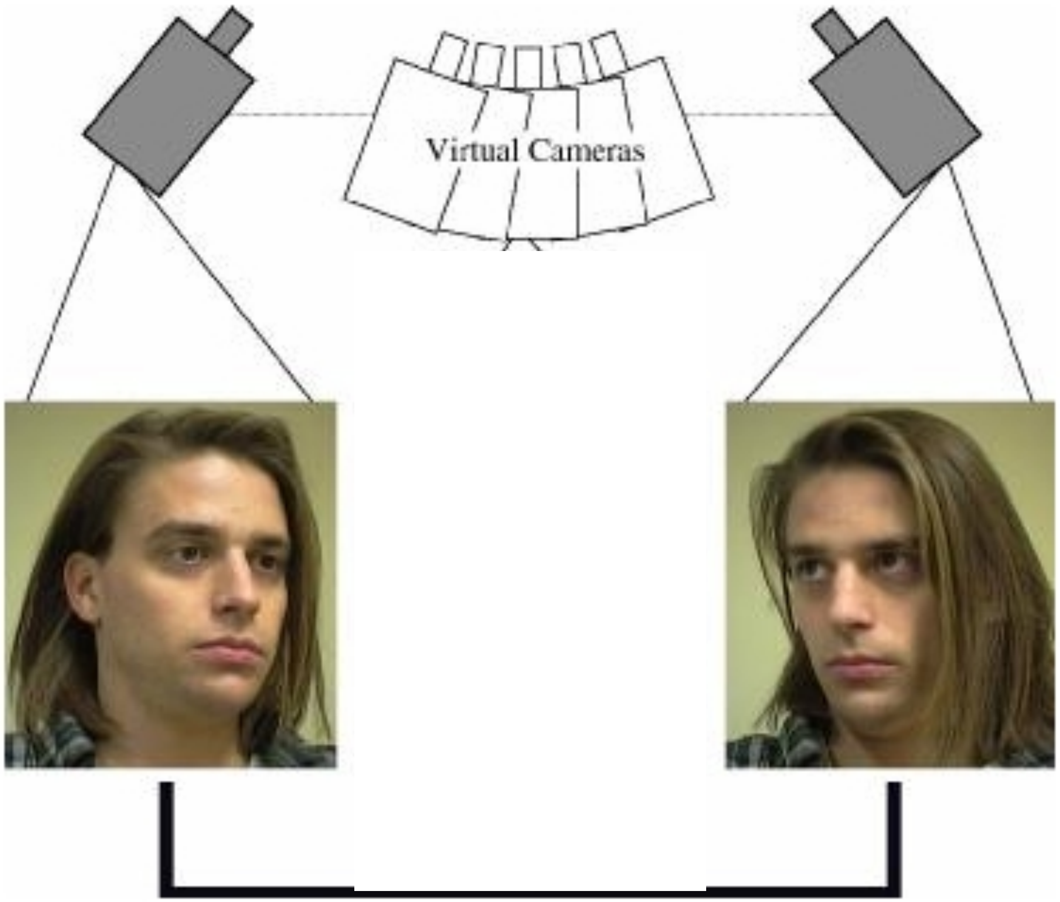


- Makes the correspondence problem easier
- Makes triangulation easy

Slide source: S. Savarese.

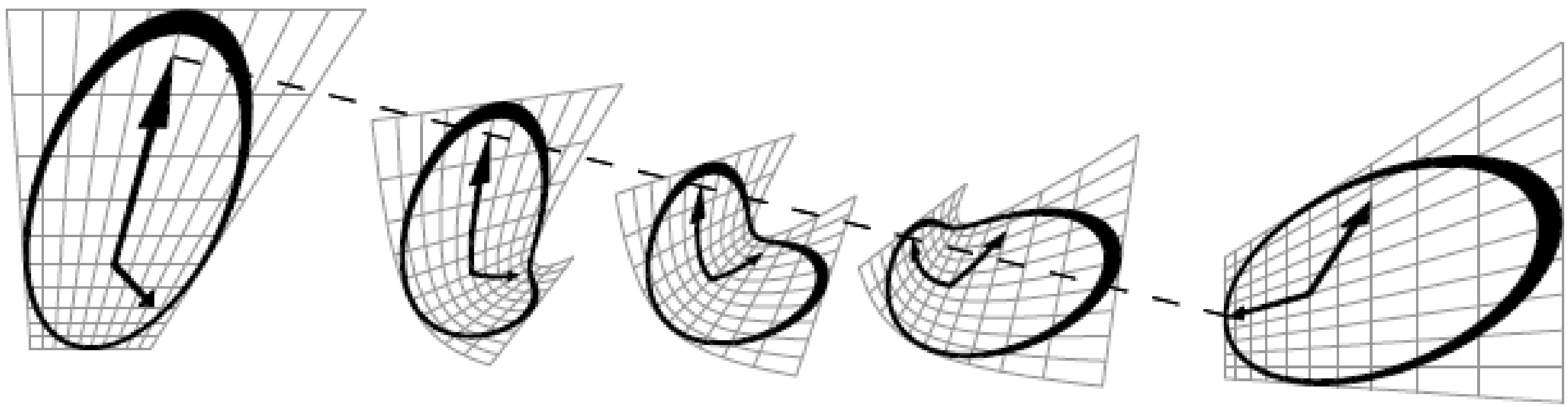
# Application: View Morphing

S. M. Seitz and C. R. Dyer, *Proc. SIGGRAPH 96*, 1996, 21-30



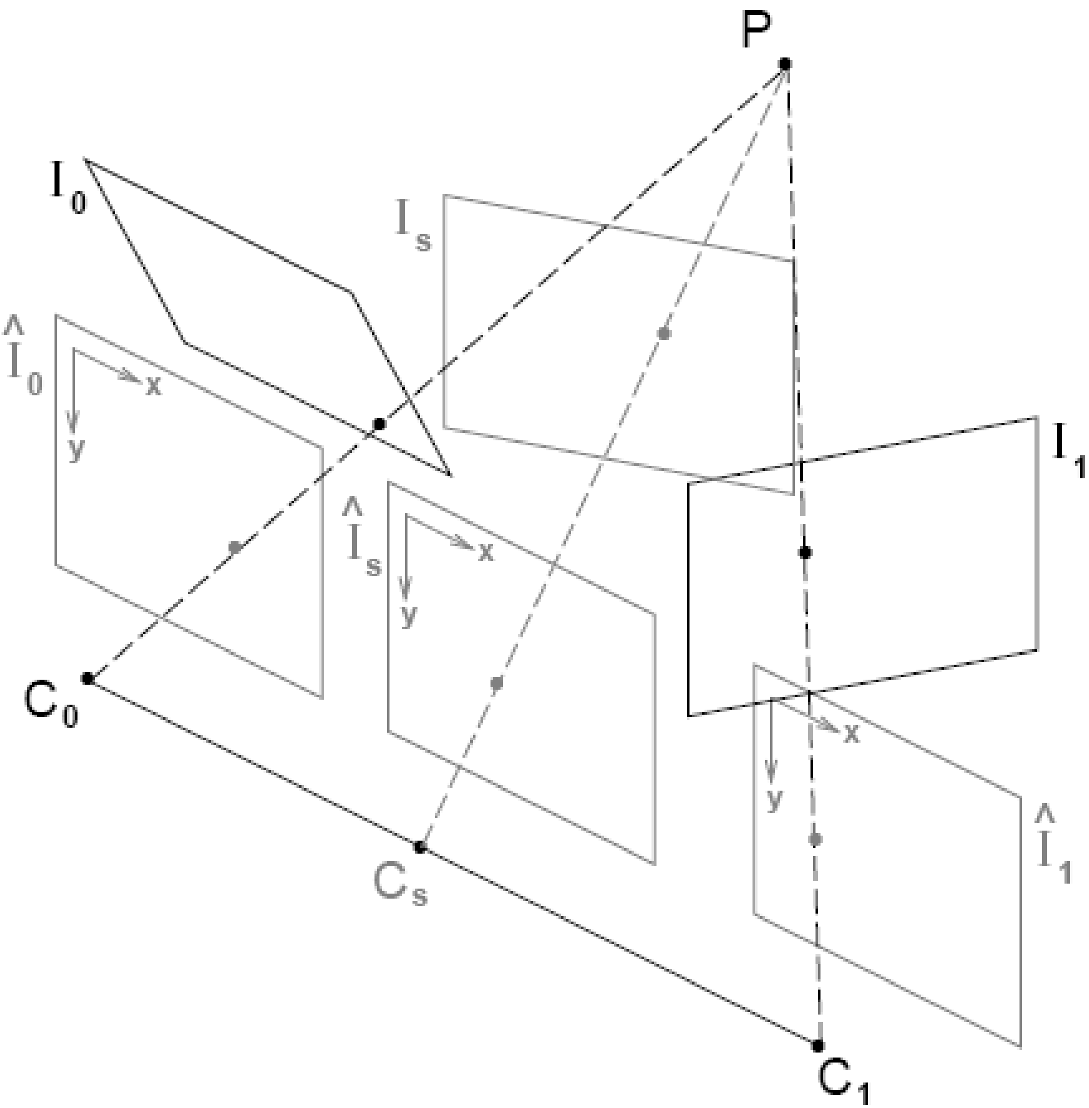

Slide source: S. Savarese.

# Morphing Without Using Geometry

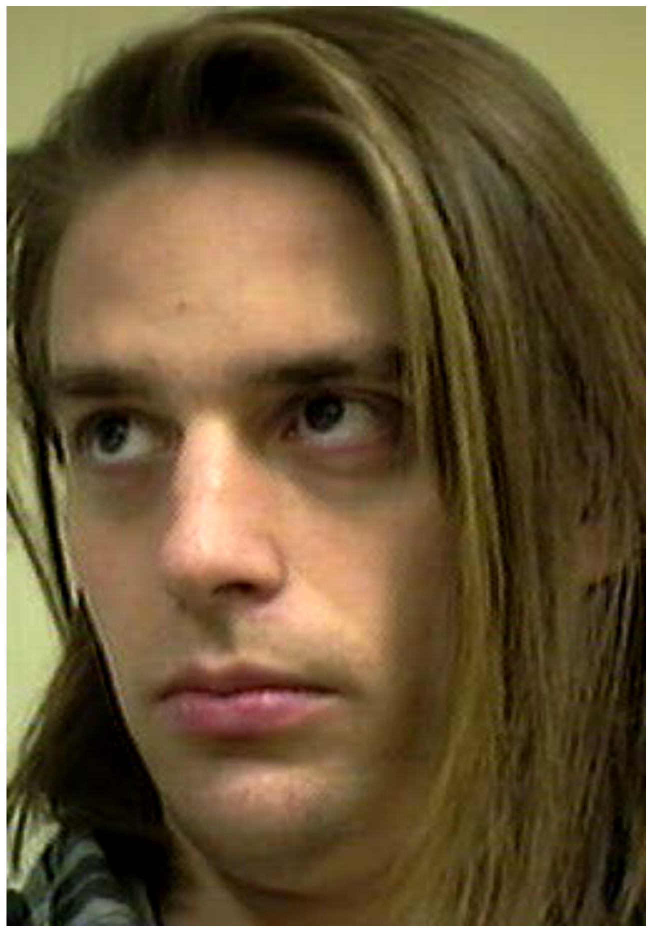


Slide source: S. Savarese.

# Rectification

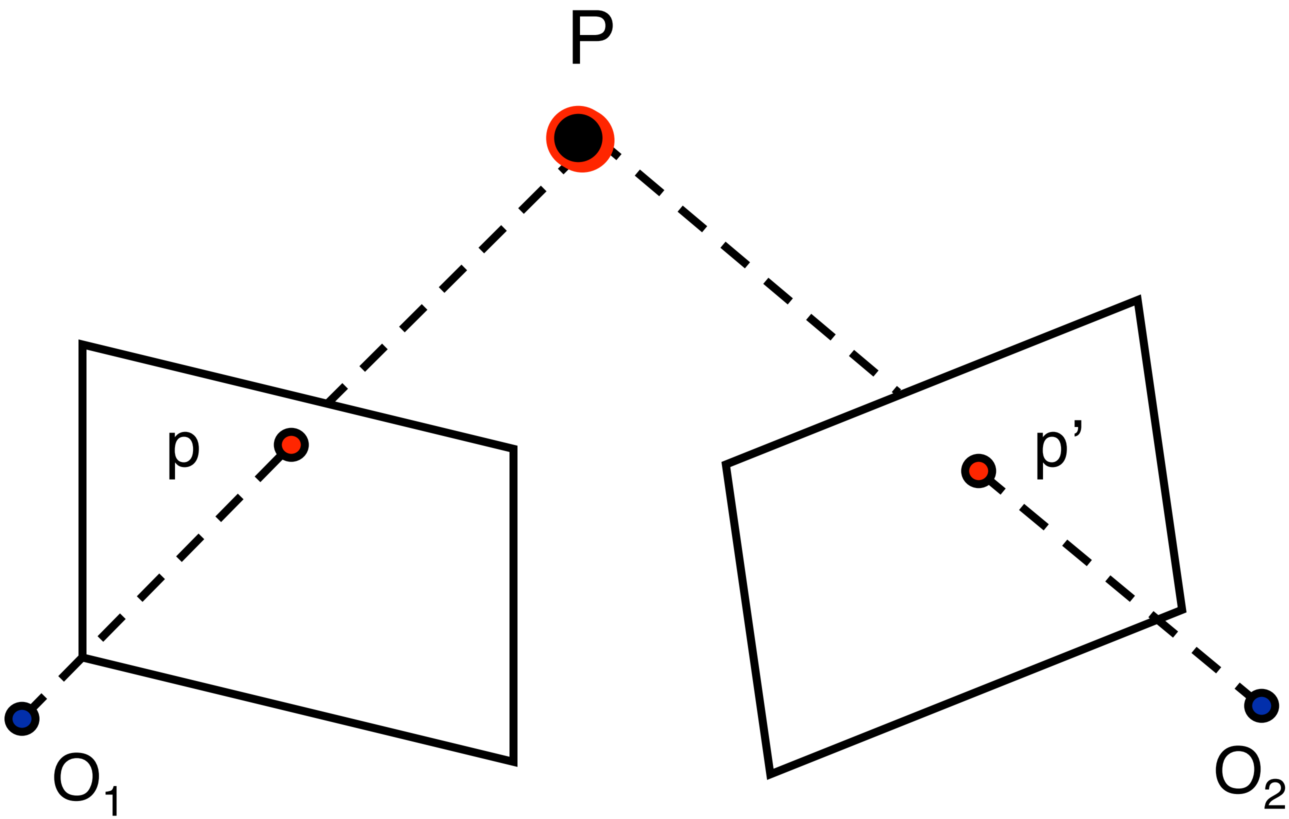



Slide source: S. Savarese.



Slide source: S. Savarese.

# Stereo Vision

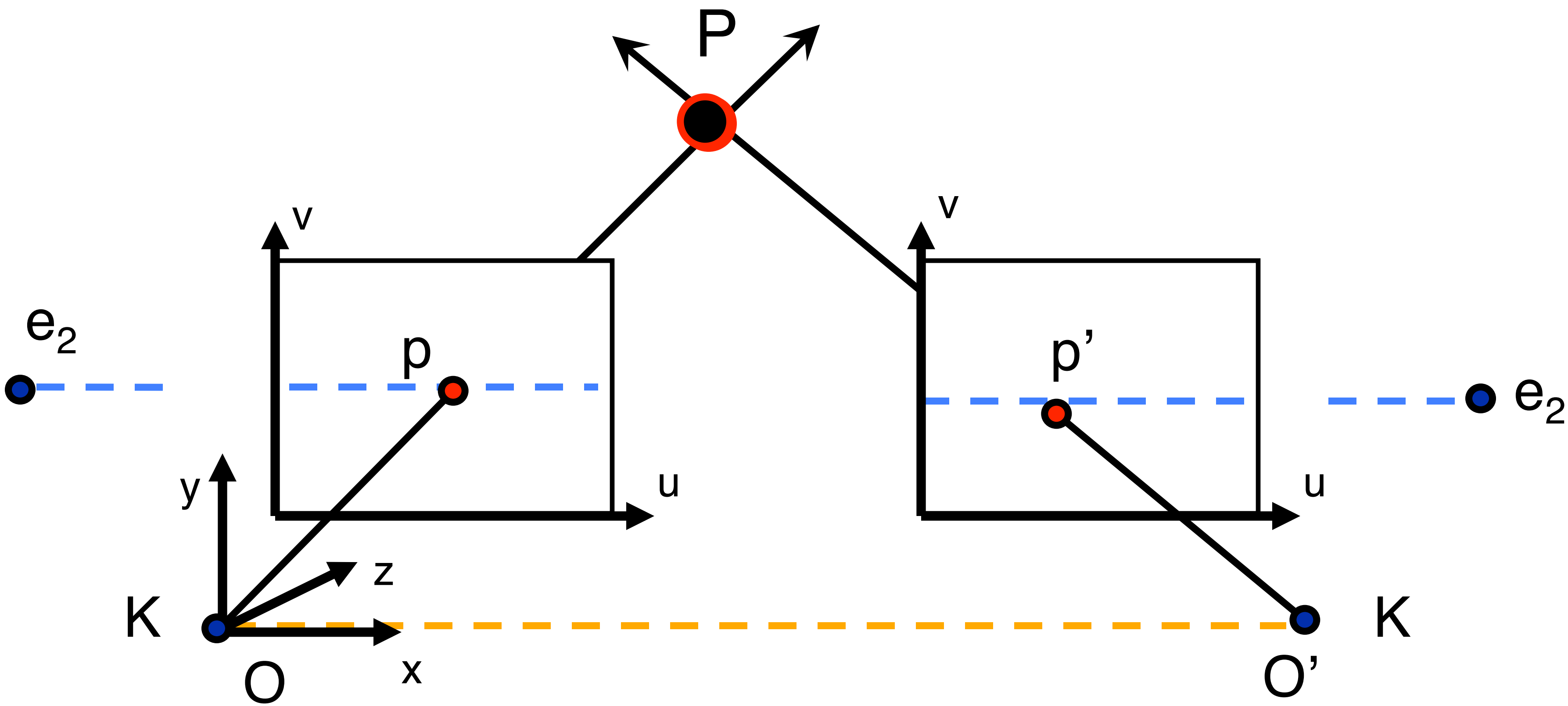



Subgoals:

- Solve the correspondence problem
- Use corresponding observations to triangulate

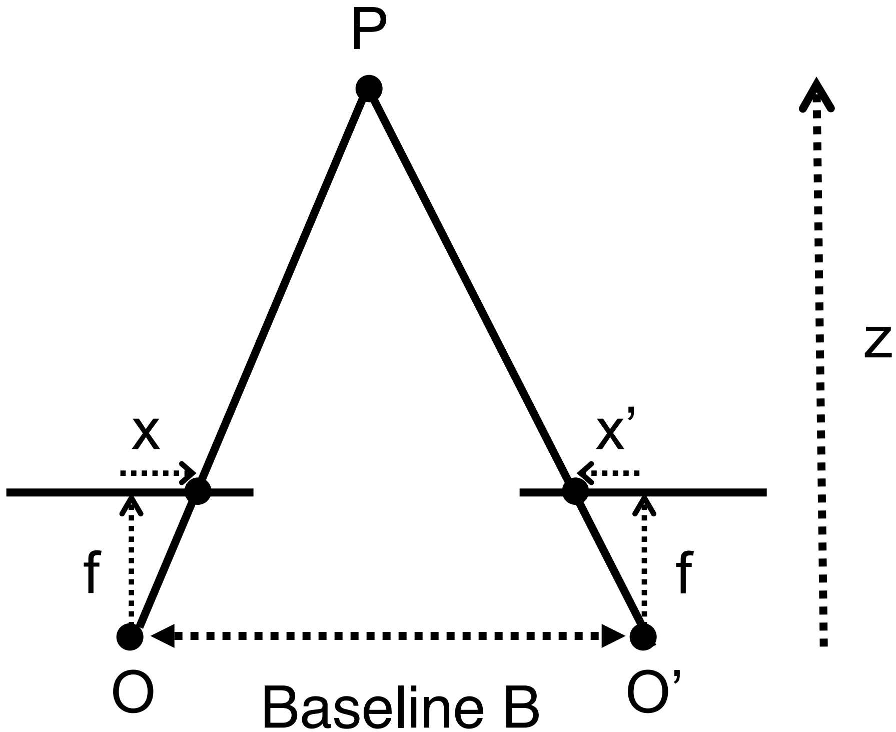
Slide source: S. Savarese.

# Computing Depth




Slide source: S. Savarese.

# Computing Depth

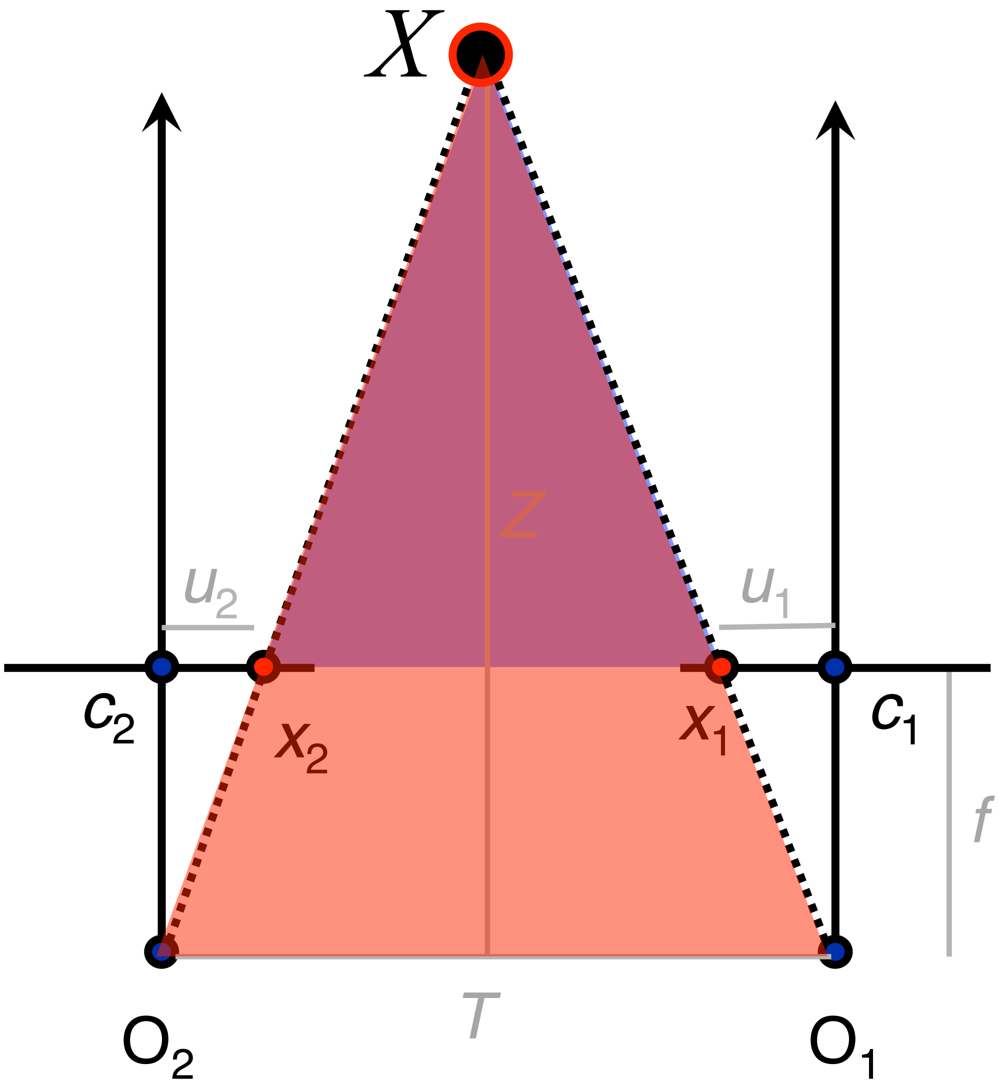



$$x - x' = \frac{B \cdot f}{z} = \text{disparity}$$

Note: Disparity is inversely proportional to depth

Slide source: S. Savarese.

# Computing Depth




$$\frac{T + u_2 - u_1}{Z - f} = \frac{T}{Z}$$

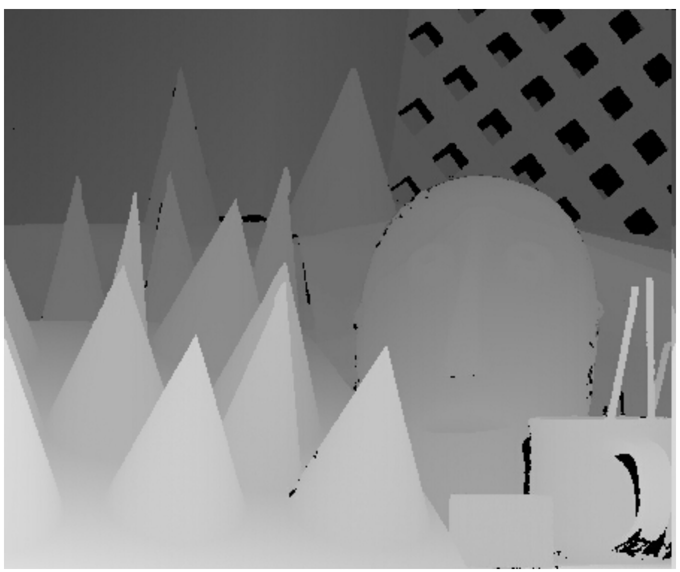
$$Z = f\frac{T}{u_1 - u_2}$$

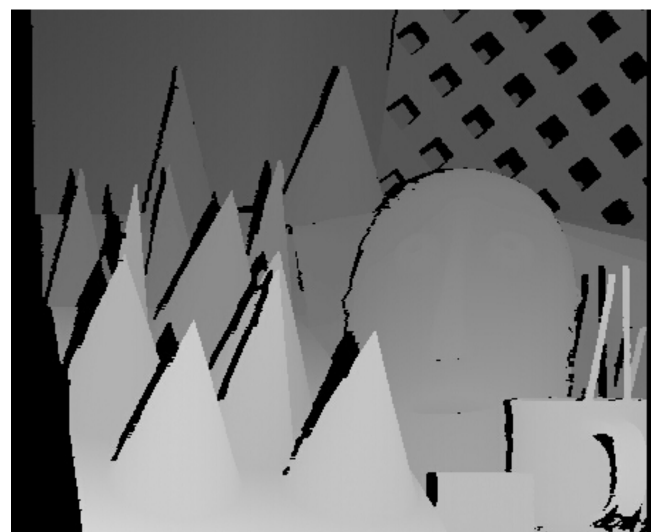
# Disparity Maps



$$Z = f \frac{T}{u_1 - u_2}$$

Stereo pair





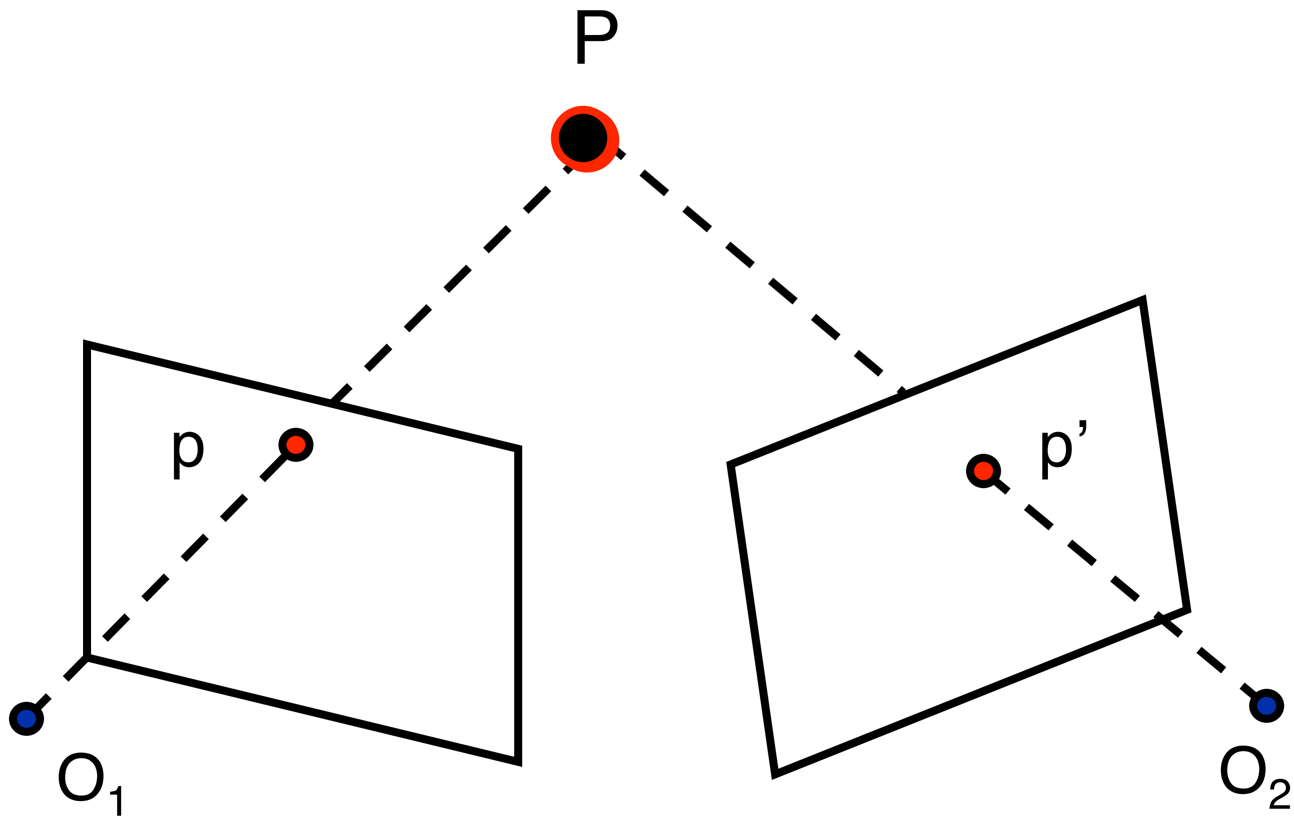
Disparity map / depth map

Disparity map  with occlusions

Slide source: S. Savarese.

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

# Stereo Vision

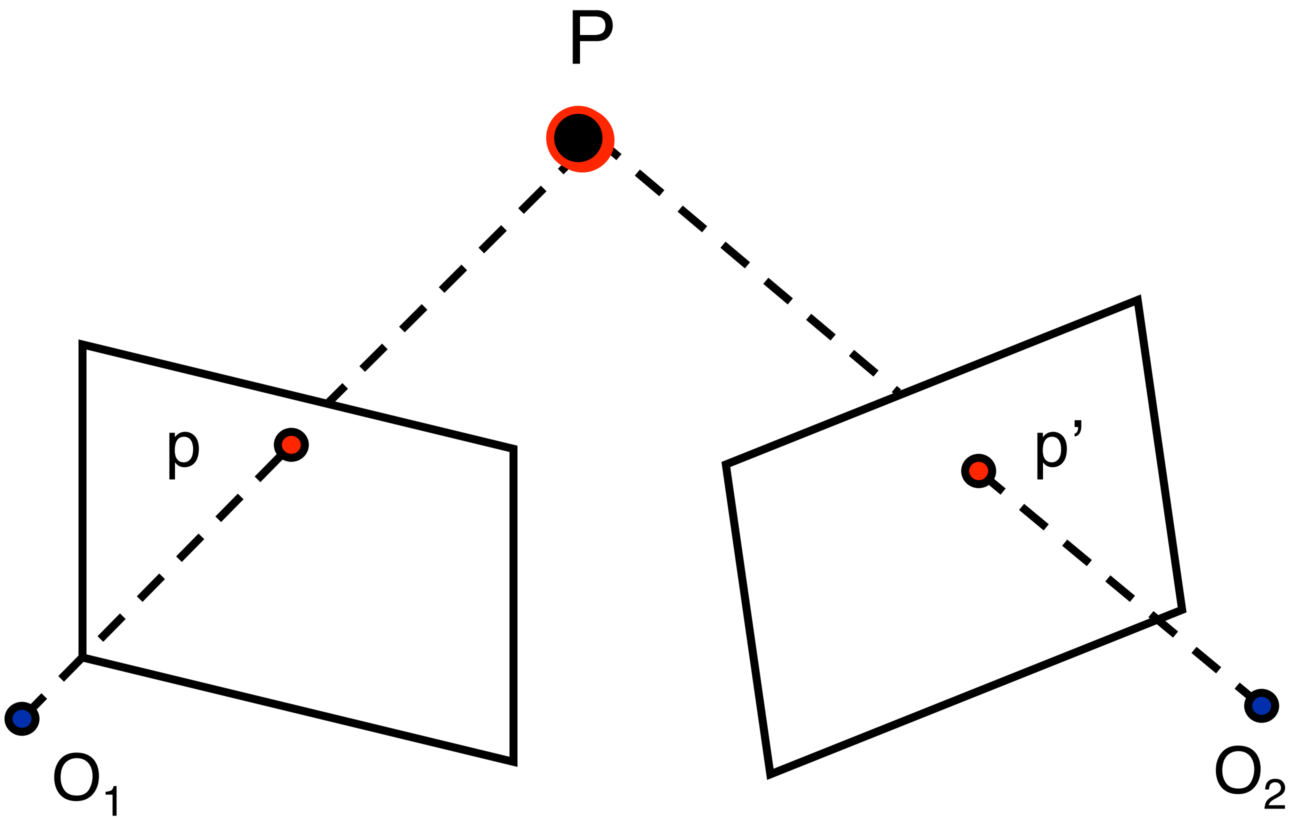



Subgoals:

- Solve the correspondence problem
- Use corresponding observations to triangulate

Slide source: S. Savarese.

# The Correspondence Problem




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

Slide source: S. Savarese.

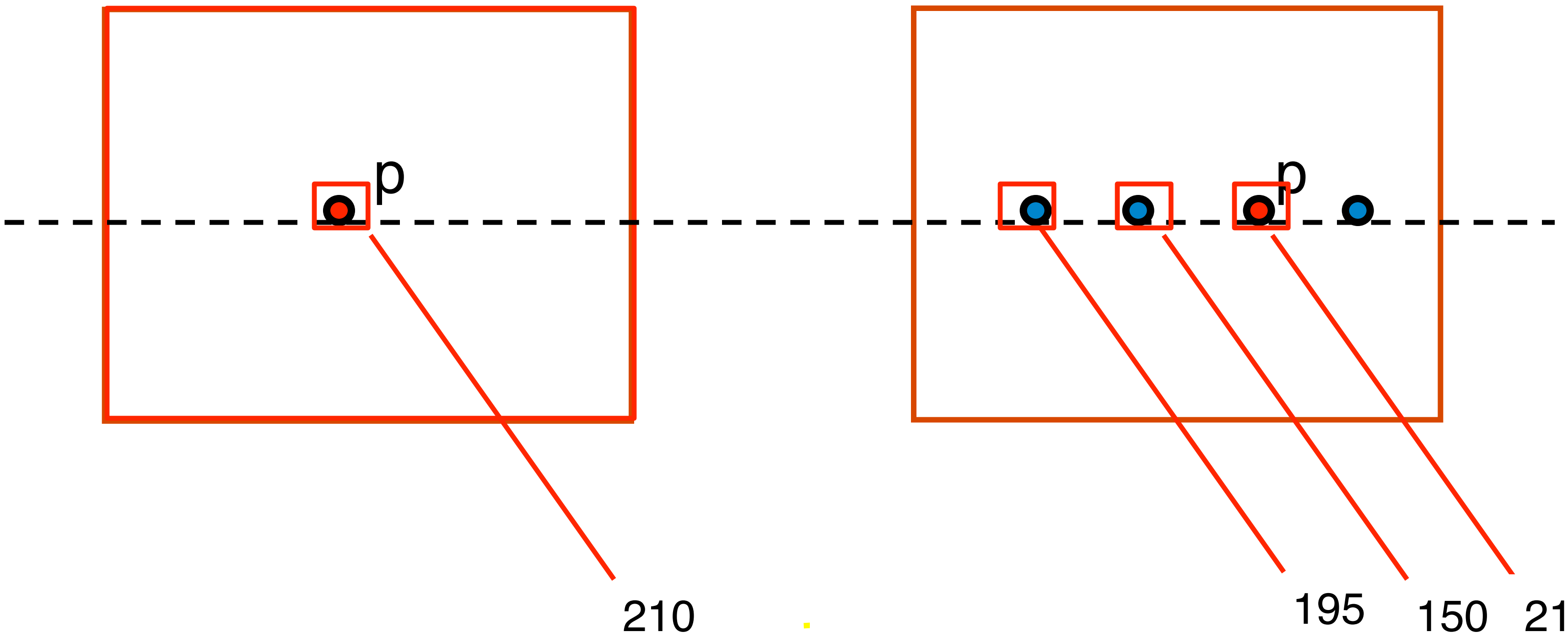
# The Correspondence Problem

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

[FP] Chapters: 8

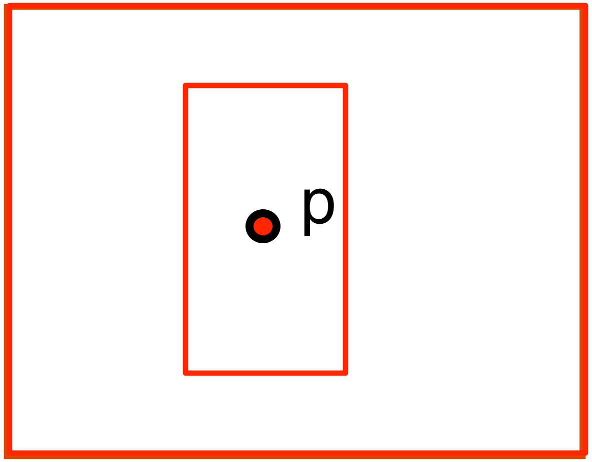
Slide source: S. Savarese.

# Correlation Methods (1970–)



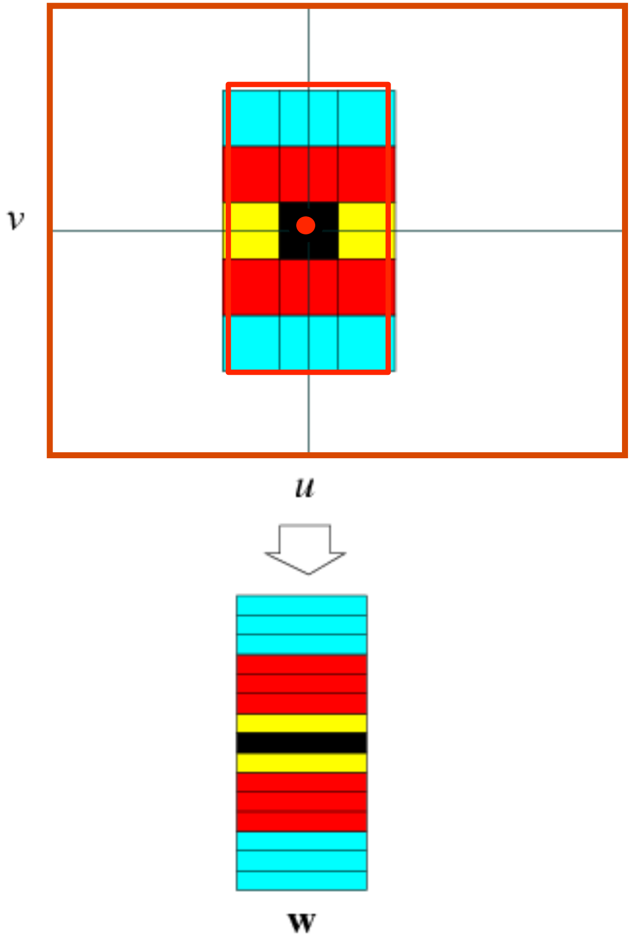

Slide source: S. Savarese.

# Correlation Methods (1970–)




- Pick up a window around p(u,v)

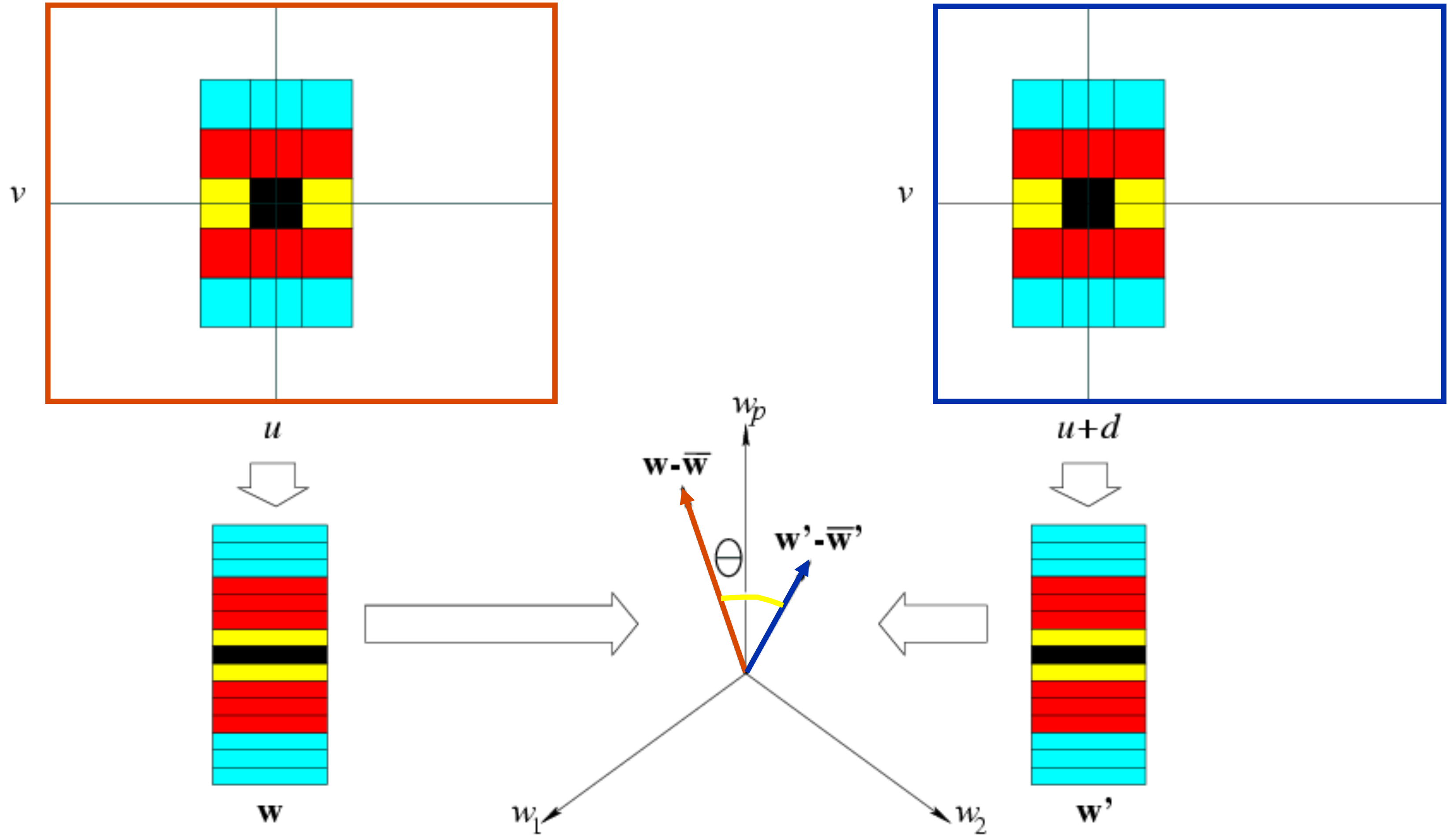
# Correlation Methods (1970–)




- Pick up a window around p(u,v)
- Build vector W
- Slide the window along v line in image 2 and compute w’
- Keep sliding until w·w’ is maximized.

Slide source: S. Savarese.

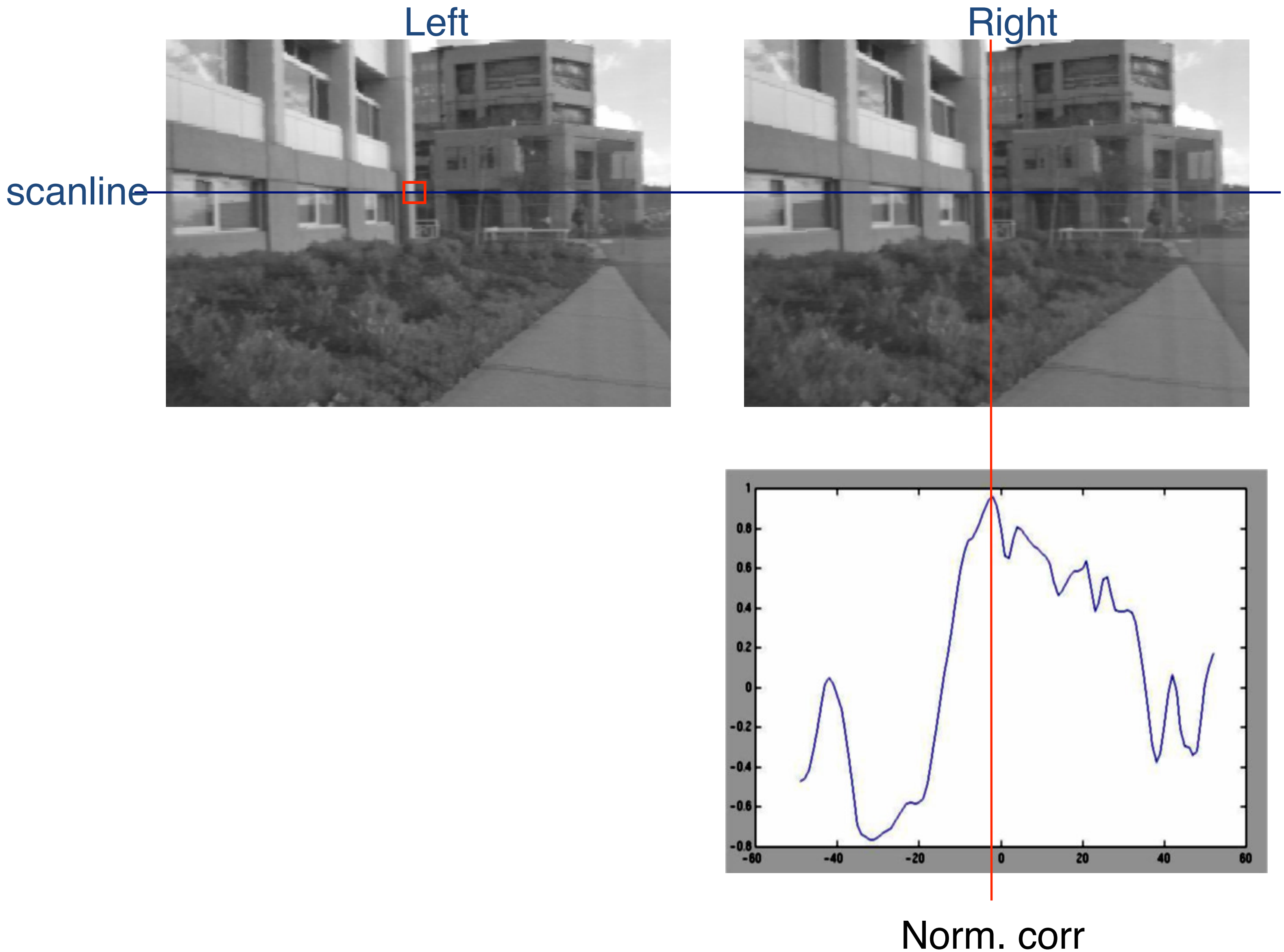
# Correlation Methods (1970–)




Normalized Correlation; minimize:

$$\frac{(\mathrm{w}-\overline{\mathrm{w}})(\mathrm{w}'-\overline{\mathrm{w}}')}{\left\|(\mathrm{w}-\overline{\mathrm{w}})(\mathrm{w}'-\overline{\mathrm{w}}')\right\|}$$

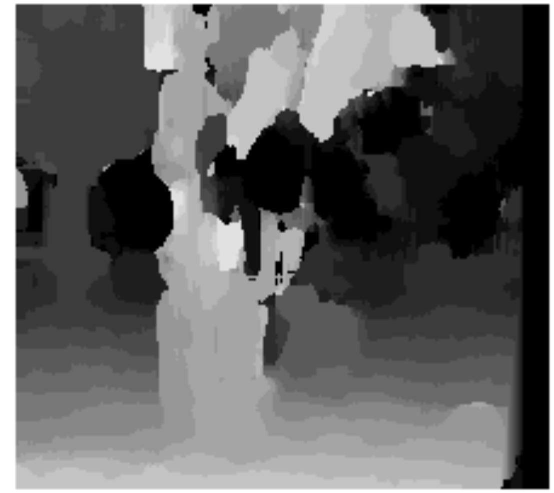
Slide source: S. Savarese.




Credit slide S. Lazebnik

# Correlation Methods

Window size = 3



Window size = 20

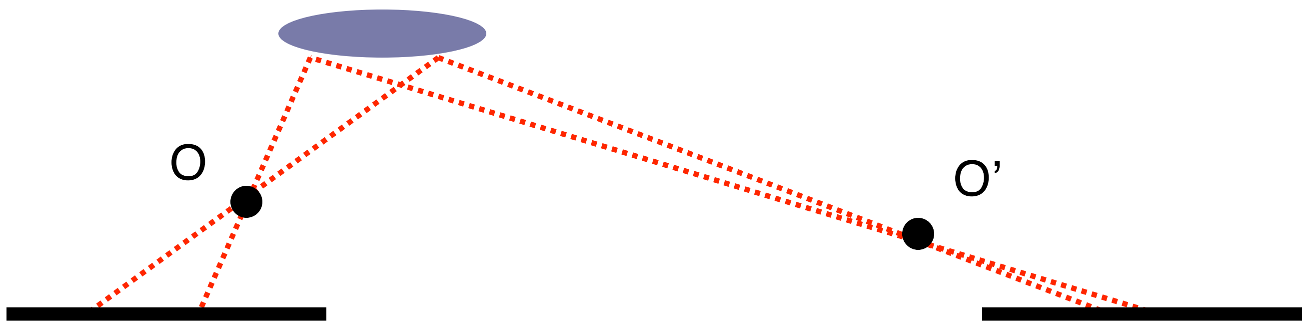
– Smaller window
  - More detail
  - More noise

– Larger window
  - Smoother disparity maps
  - Less prone to noise

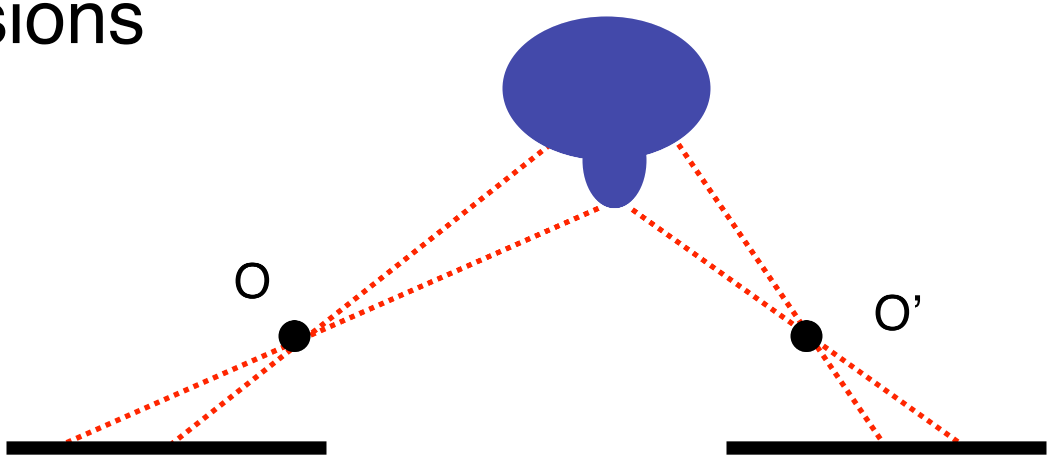
Credit slide S. Lazebnik

# Issues

- Fore shortening effect



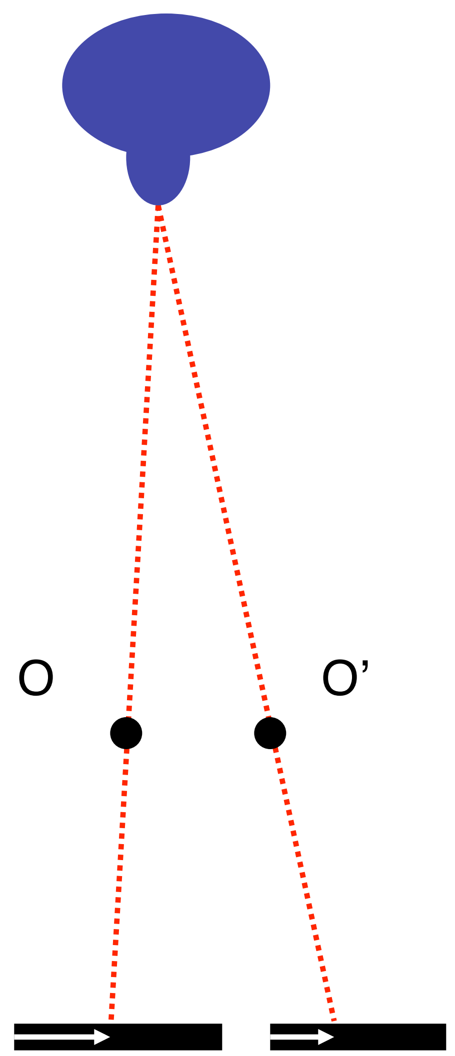

- Occlusions




Slide source: S. Savarese.

# Issues

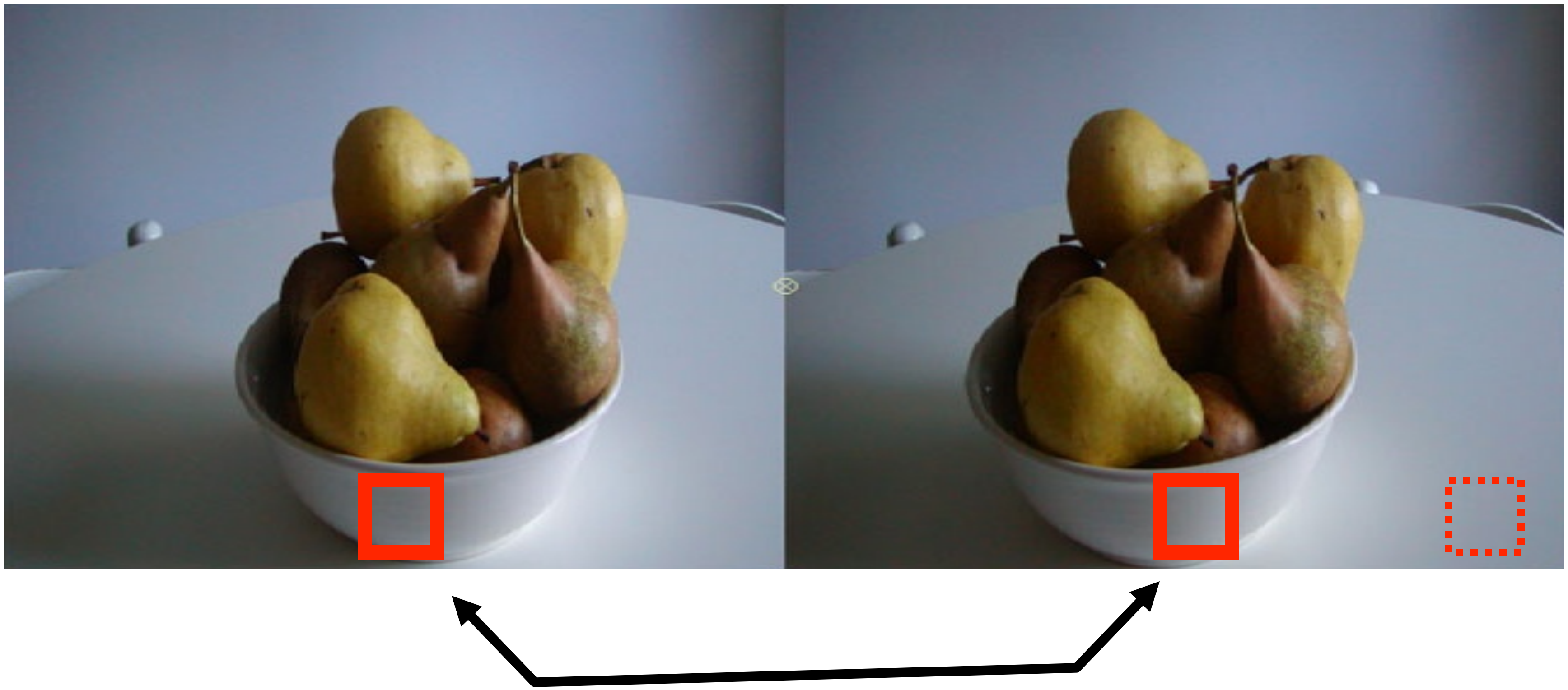
- It is desirable to have small T/z ratio!
- Small error in measurements implies large error in estimating depth




Slide source: S. Savarese.

# Issues

- Homogeneous regions



Hard to match pixels in these regions

Slide source: S. Savarese.

# Issues

- Repetitive patterns

Slide source: S. Savarese.

# The Correspondence Problem is Difficult

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

Apply non-local constraints to help enforce the correspondences

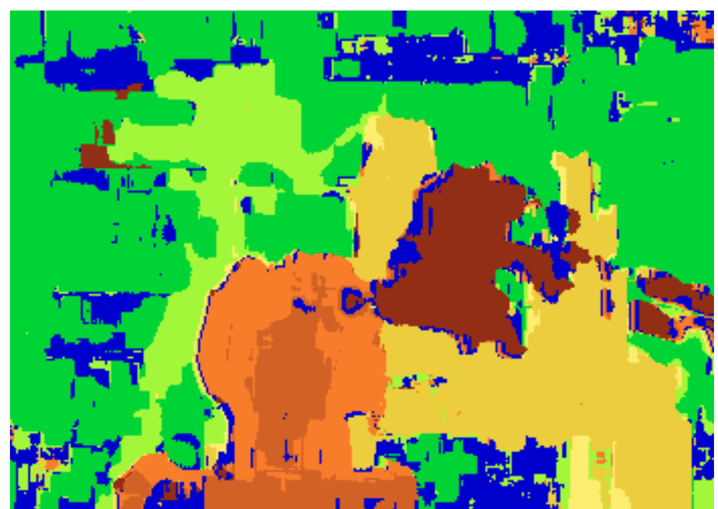
Slide source: S. Savarese.

# Results with window search

Data

Ground truth

Window-based matching



Credit slide S. Lazebnik

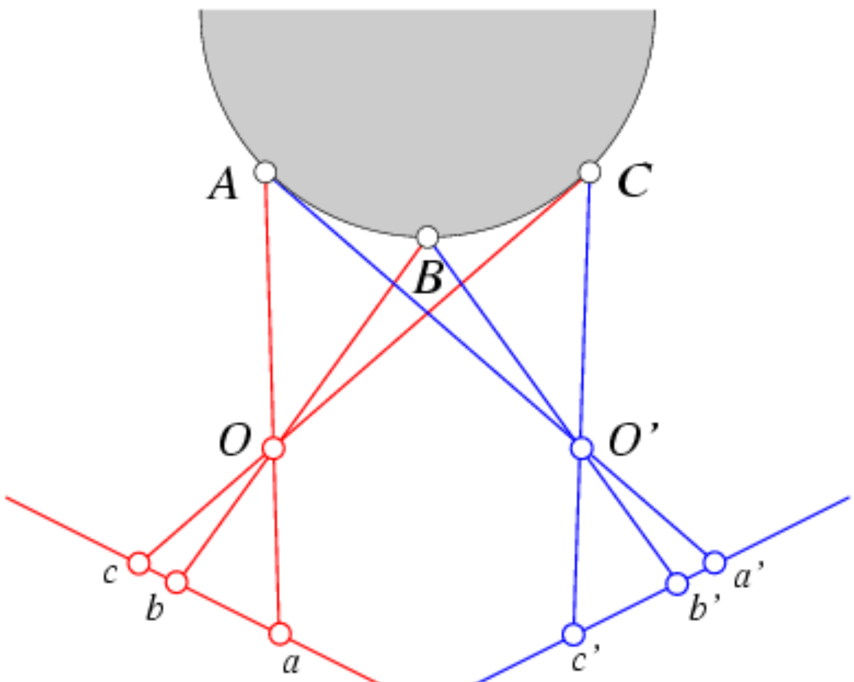
# Improving correspondence: Non-local constraints

- Uniqueness
  - For any point in one image, there should be at most one matching point in the other image

Slide source: S. Savarese.

# Improving correspondence: 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

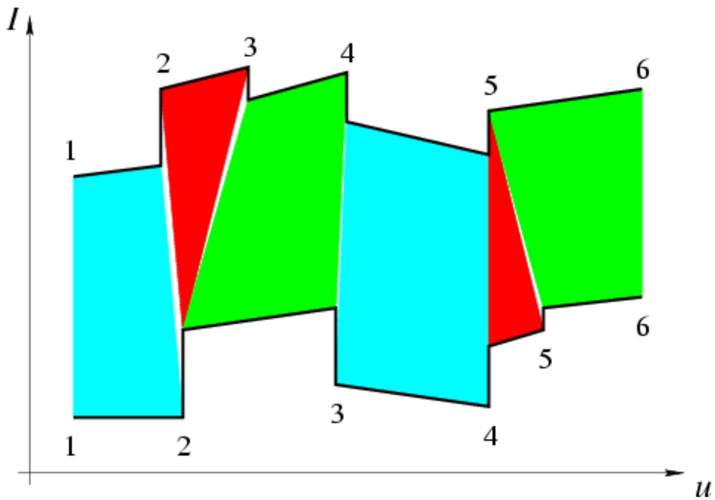


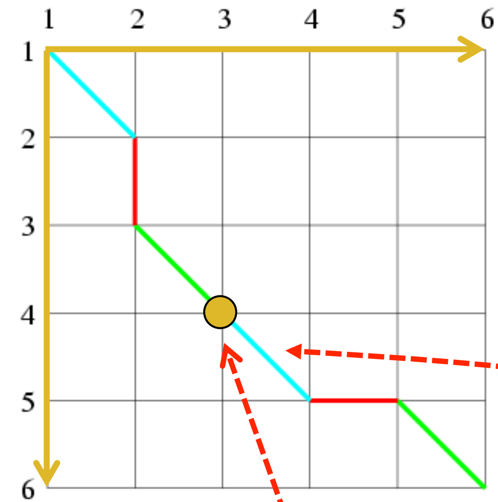

Not always in presence of occlusions!

courtesy slide to J. Ponce

# Dynamic Programming (Baker and Binford, 1981)

[Uses ordering constraint]



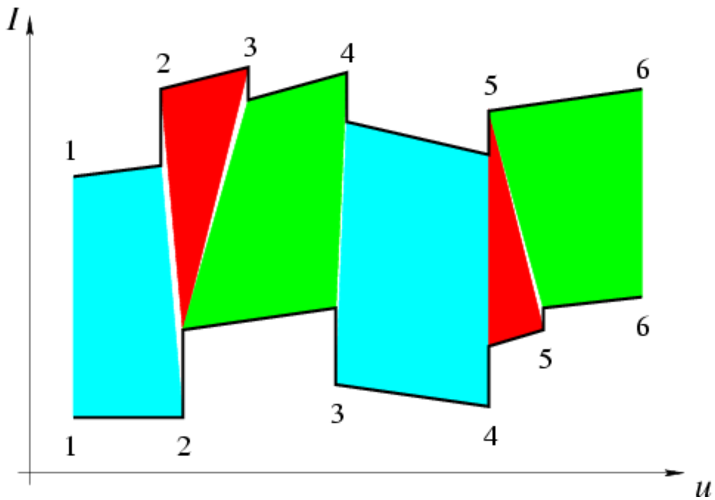



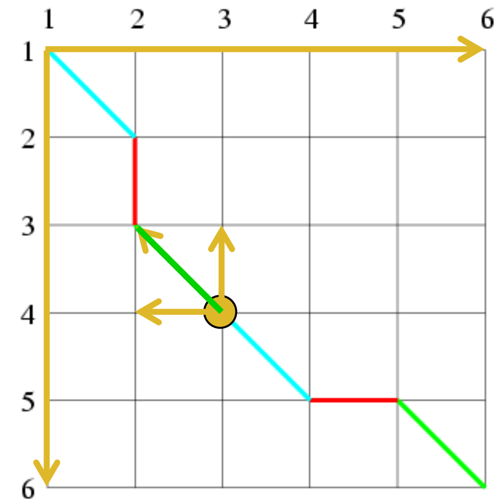

- Nodes = matched feature points (e.g., edge points).
- Arcs = matched intervals along the epipolar lines.
- Arc cost = discrepancy between intervals.

Find the minimum-cost path going monotonically down and right from the top-left corner of the graph to its bottom-right corner.

courtesy slide to J. Ponce

# Dynamic Programming (Baker and Binford, 1981)

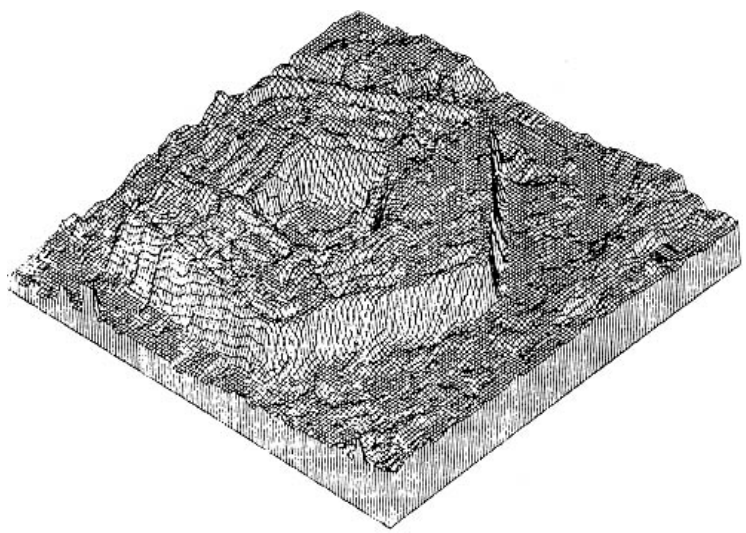



```
% Loop over all nodes (k, l) in ascending order.
for k = 1 to m do
  for l = 1 to n do
    % Initialize optimal cost C(k, l) and backward pointer B(k, l).
    C(k, l) ← +∞; B(k, l) ← nil;
    % Loop over all inferior neighbors (i, j) of (k, l).
    for (i, j) ∈ Inferior − Neighbors(k, l) do
      % Compute new path cost and update backward pointer if necessary.
      d ← C(i, j) + Arc − Cost(i, j, k, l);
      if d < C(k, l) then C(k, l) ← d; B(k, l) ← (i, j) endif;
      endfor;
    endfor;
  endfor;
% Construct optimal path by following backward pointers from (m, n).
P ← {(m, n)}; (i, j) ← (m, n);
while B(i, j) ≠ nil do (i, j) ← B(i, j); P ← {(i, j)} ∪ P endwhile.
```

courtesy slide to J. Ponce

# Dynamic Programming (Ohta and Kanade, 1985)



Reprinted from “Stereo by Intra- and Intet-Scanline Search,” by Y. Ohta and T. Kanade, IEEE Trans. on Pattern Analysis and Machine Intelligence, 7(2):139-154 (1985). 

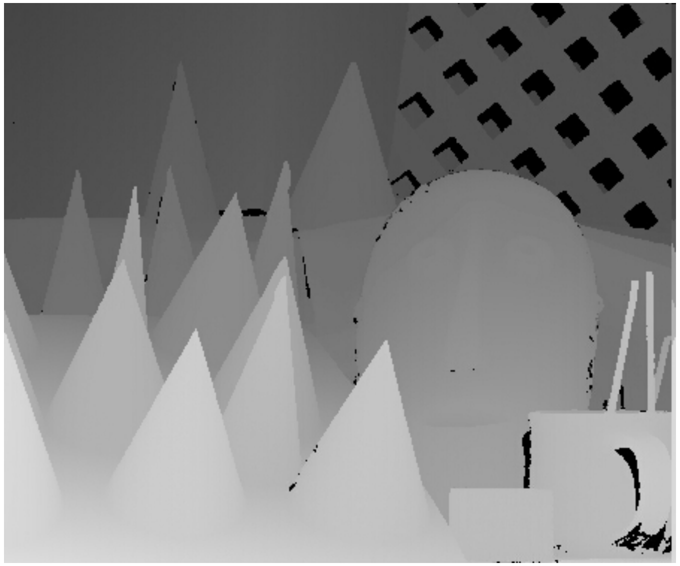
# Improving correspondence: Non-local constraints

- 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 (expect in occluding boundaries)

Slide source: S. Savarese.

# Smoothness

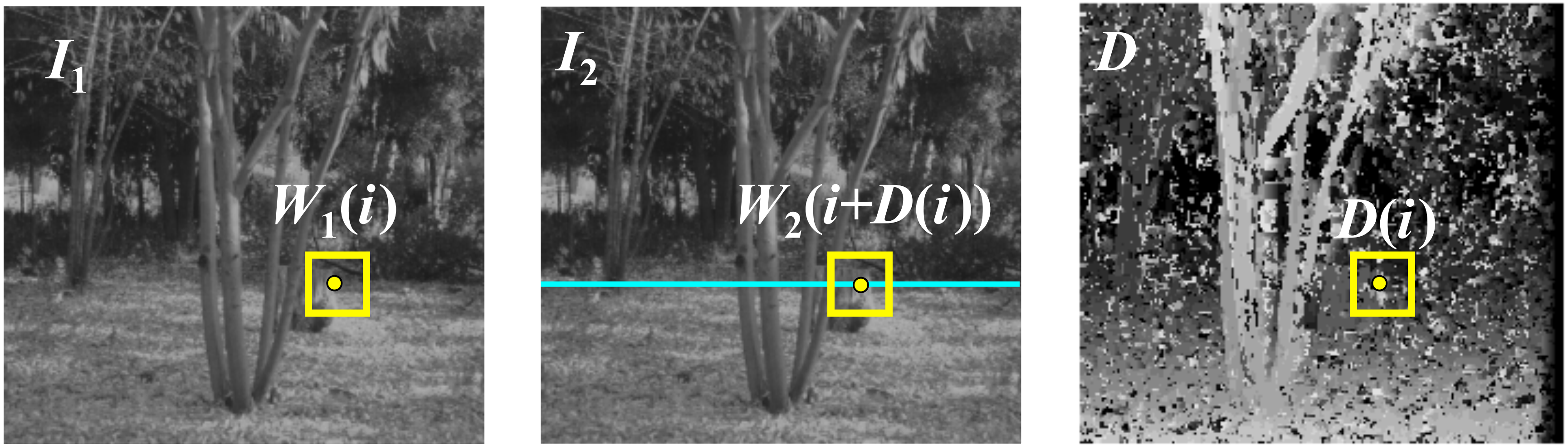




Slide source: S. Savarese.

# Stereo Matching as Energy Minimization

Y. Boykov, O. Veksler, and R. Zabih, Fast Approximate Energy Minimization via Graph Cuts, PAMI 01




$$E = \alpha E_{\text{data}}(I_1, I_2, D) + \beta E_{\text{smooth}}(D)$$

$$E_{\text{data}} = \sum_i \left(W_1(i) - W_2(i + D(i))\right)^2$$

$$E_{\text{smooth}} = \sum_{\text{neighbors } i,j} \rho\left(D(i) - D(j)\right)$$

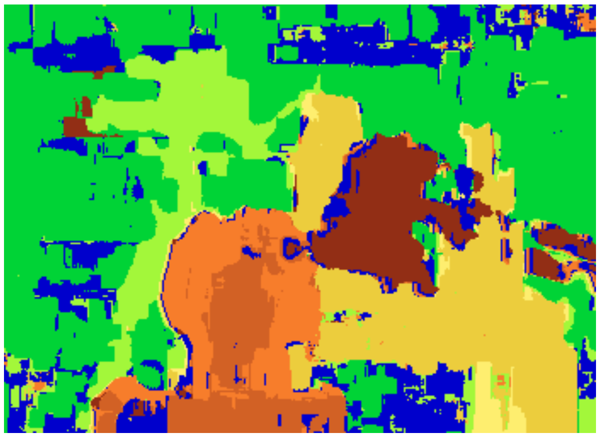
- Energy functions of this form can be minimized using *graph cuts*

Credit slide S. Lazebnik

# Stereo Matching as Energy Minimization

Y. Boykov, O. Veksler, and R. Zabih, Fast Approximate Energy Minimization via Graph Cuts, PAMI 01

Ground truth



Window-based matching

Graph cuts

Slide source: S. Savarese.

# *Stereo SDK* stereo vision software development kit.

**A. Criminisi, A. Blake and D. Robertson**

# Application: Foreground/Background Segmentation

V. Kolmogorov, A. Criminisi, A. Blake, G. Cross and C. Rother.
**Bi-layer segmentation of binocular stereo video** CVPR 2005

http://research.microsoft.com/~antcrim/demos/ACriminisi_Recognition_CowDemo.wmv

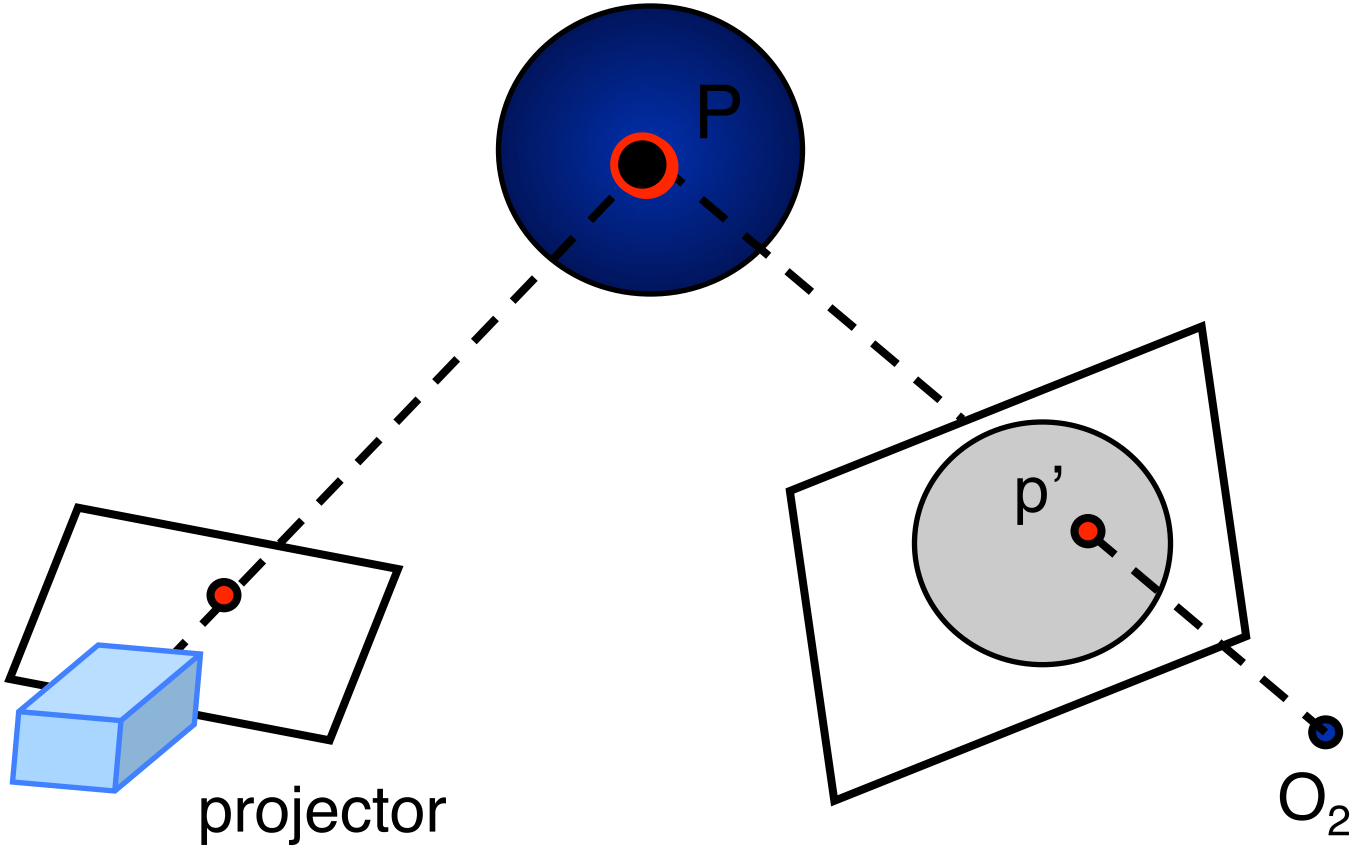
# Application: 3D Urban Scene Modeling

3D Urban Scene Modeling Integrating Recognition and Reconstruction,
N. Cornelis, B. Leibe, K. Cornelis, L. Van Gool, IJCV 08.

Link to movie.

http://www.vision.ee.ethz.ch/showroom/index.en.html#

# Active Stereo: Point

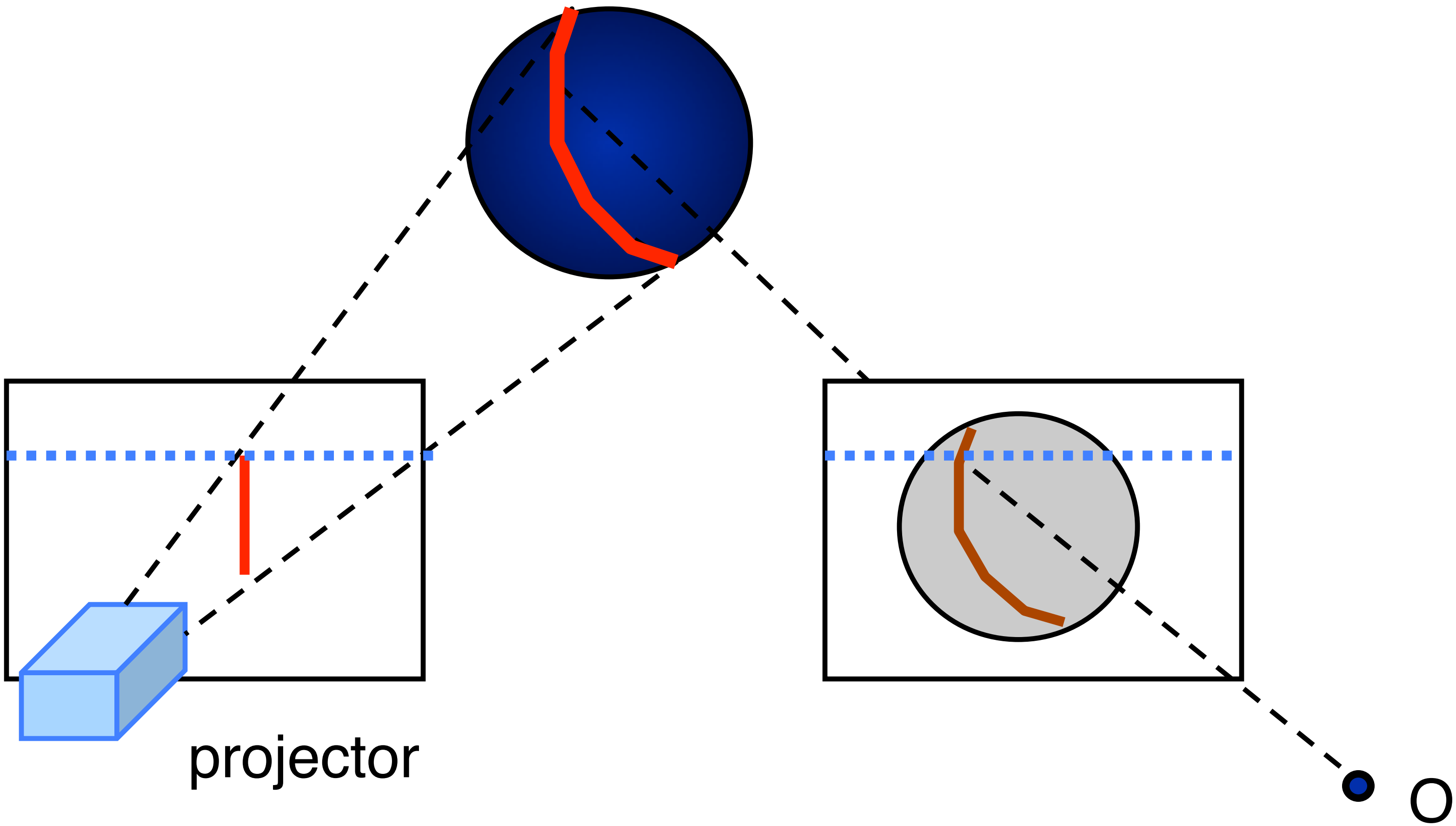



Replace one of the two cameras by a projector

- Single camera
- Projector geometry calibrated
- What's the advantage of having the projector? Correspondence problem solved!

Slide source: S. Savarese.

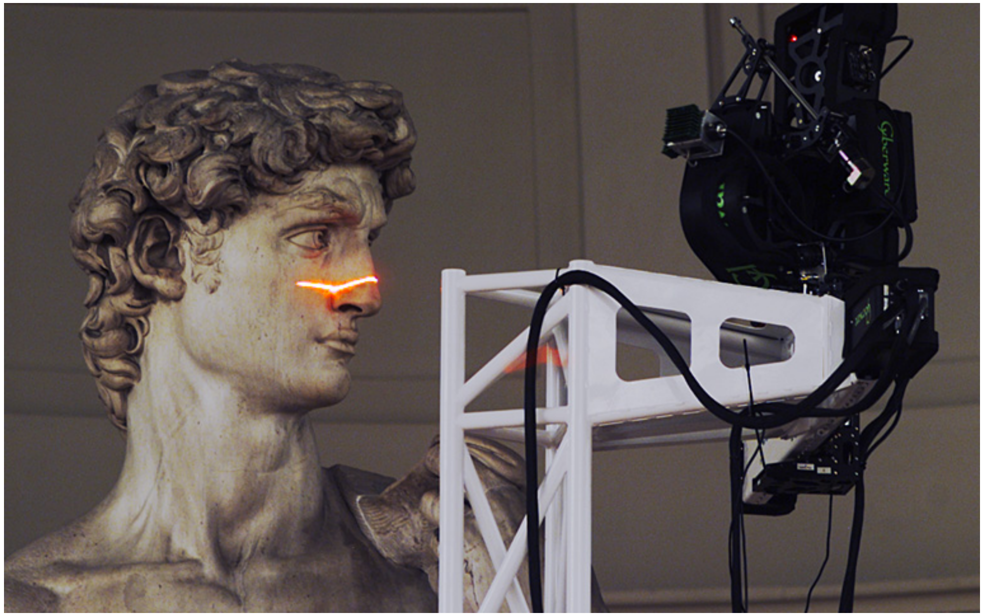
# Active Stereo: Stripe




-Projector and camera are parallel
- Correspondence problem solved!

Slide source: S. Savarese.

# Laser scanning



Digital Michelangelo Project
http://graphics.stanford.edu/projects/mich/

- Optical triangulation
  - Project a single stripe of laser light
  - Scan it across the surface of the object
  - This is a very precise version of structured light scanning

Source: S. Seitz

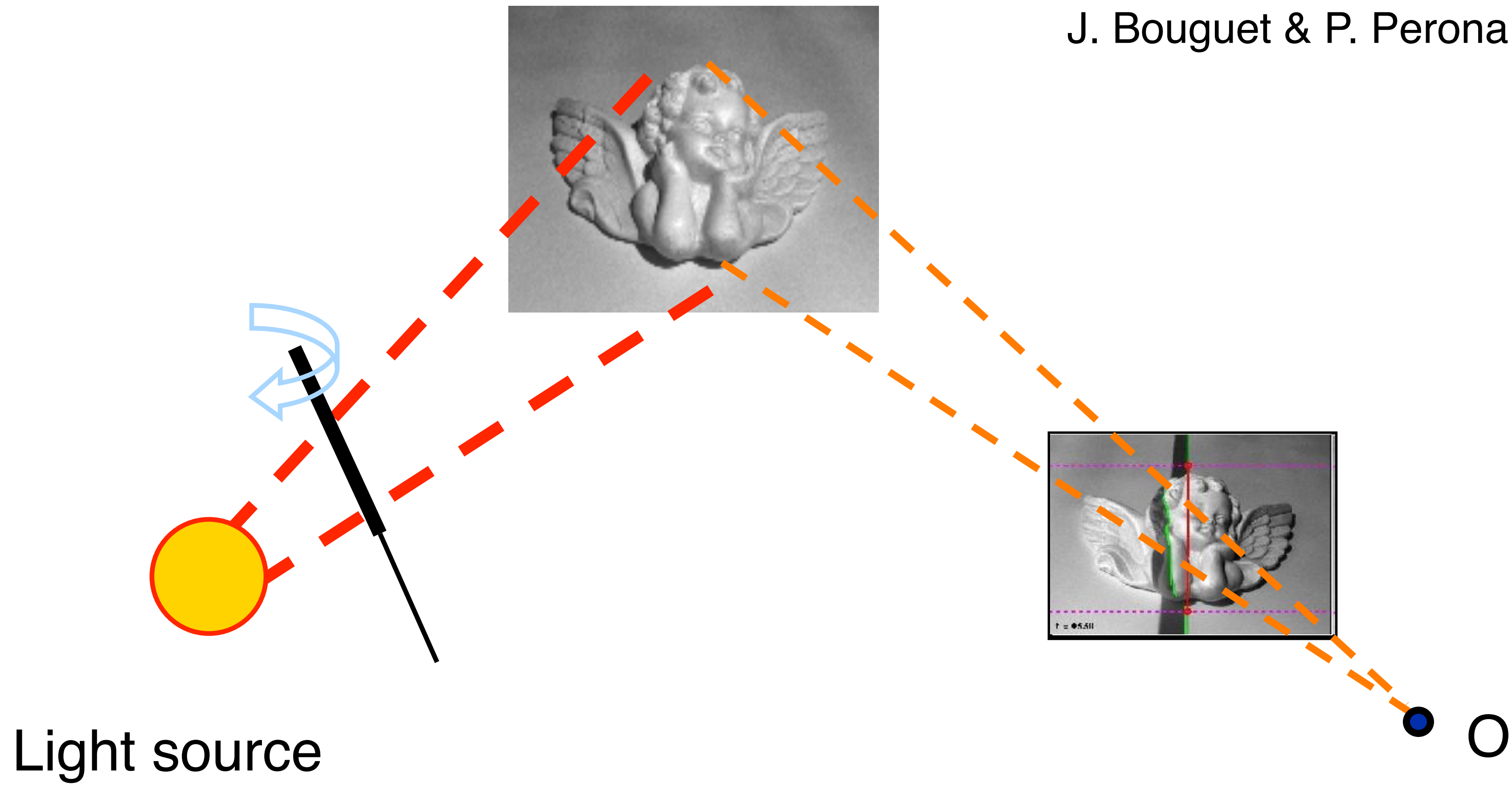
# Laser scanning

The Digital Michelangelo Project, Levoy et al.

Source: S. Seitz

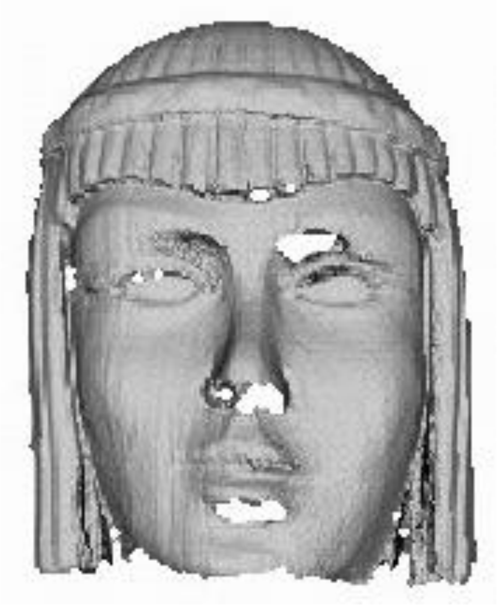
# Active Stereo: Shadows

J. Bouguet & P. Perona, 99




- 1 camera,1 light source
- very cheap setup
- calibrated the light source

# Active Stereo: Shadows

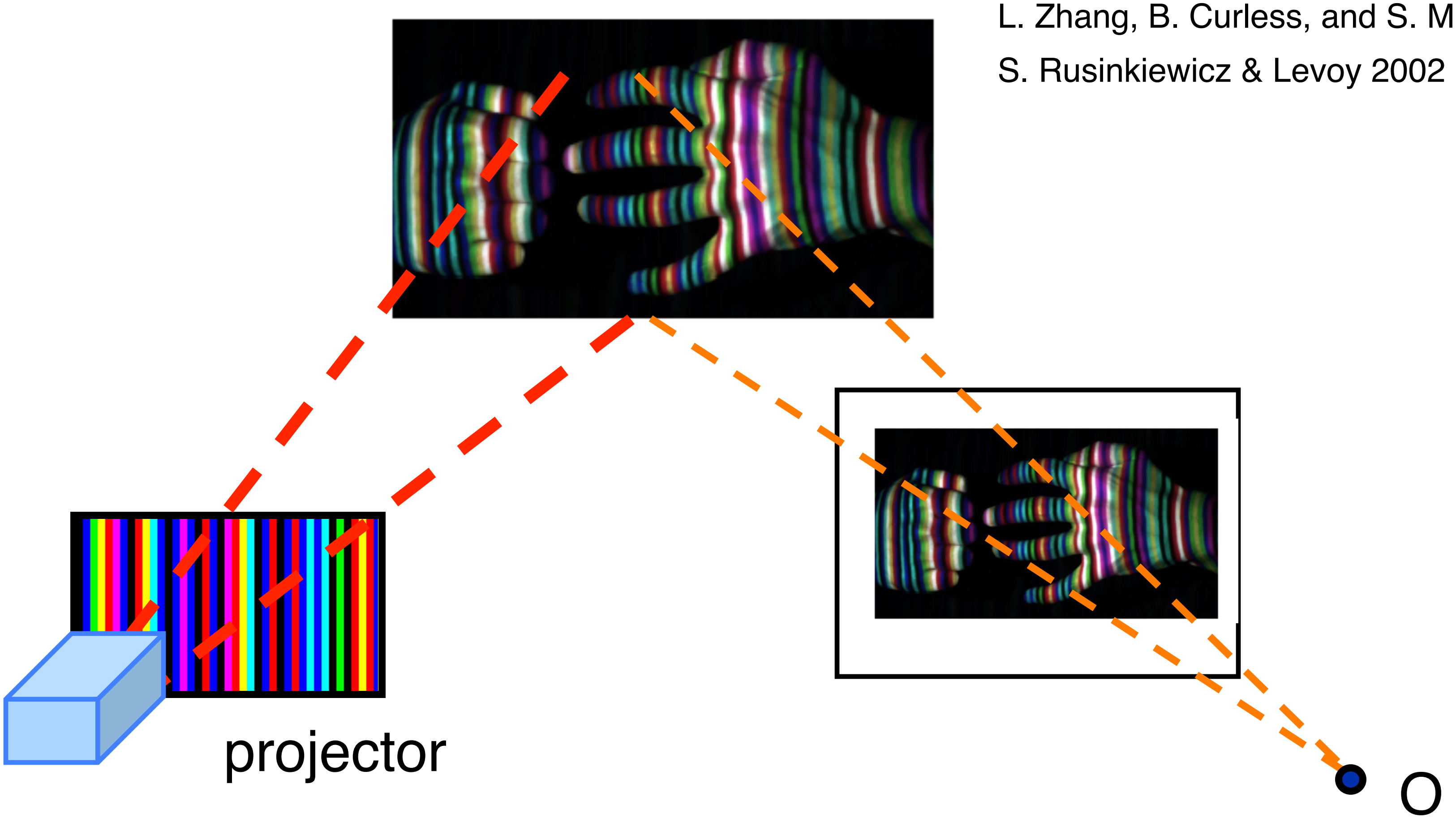


Slide source: S. Savarese.

# Active Stereo: Color-Coded Stripes

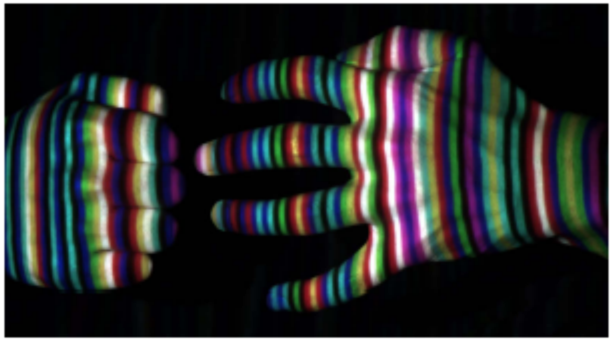
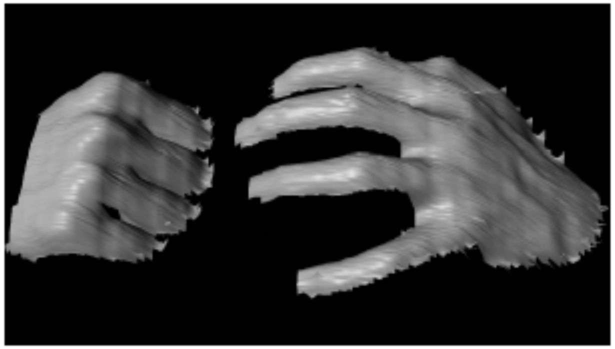
L. Zhang, B. Curless, and S. M. Seitz 2002

S. Rusinkiewicz & Levoy 2002



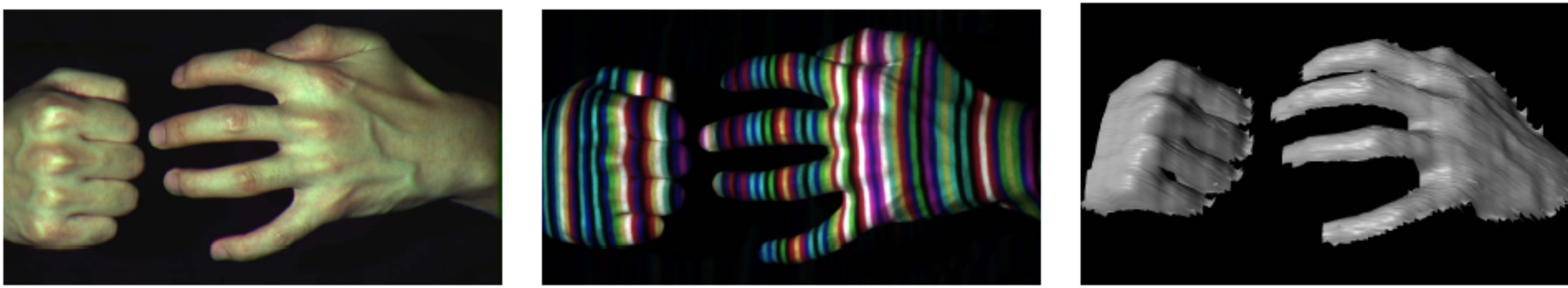

- Dense reconstruction
- Correspondence problem again
- Get around it by using color codes

Slide source: S. Savarese.

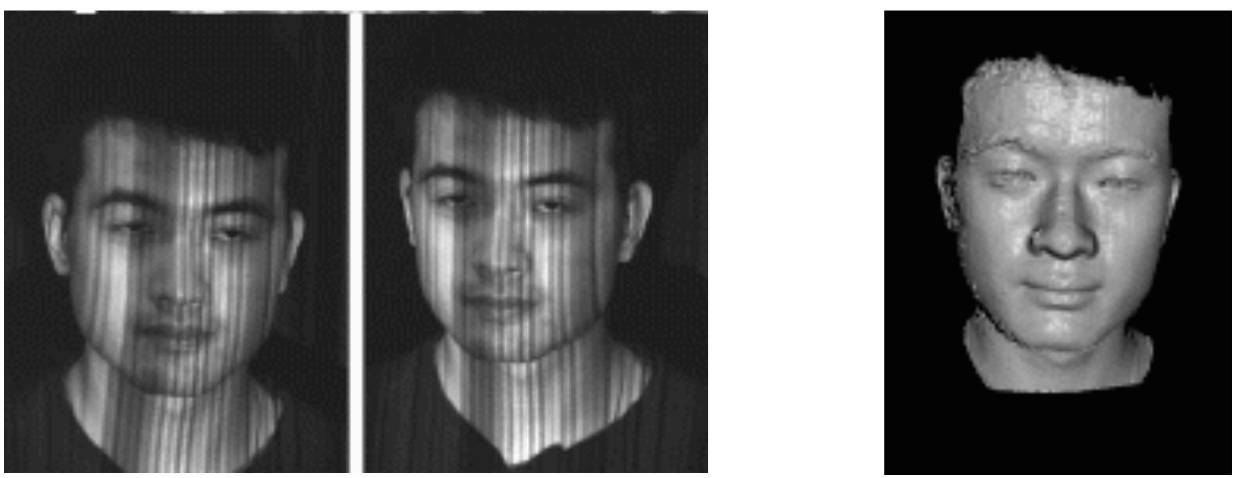
L. Zhang, B. Curless, and S. M. Seitz. Rapid Shape Acquisition Using Color Structured Light and Multi-pass Dynamic Programming. *3DPVT* 2002

Slide source: S. Savarese.



L. Zhang, B. Curless, and S. M. Seitz. Rapid Shape Acquisition Using Color Structured Light and Multi-pass Dynamic Programming. *3DPVT* 2002

# Rapid shape acquisition: Projector + stereo cameras



# Next Lecture: Affine Structure from Motion

- Readings: FP 8.1-2; SZ 7