Calibrated Stereo (Part 1)

Computer Vision I

CSE 252A

Lecture 7

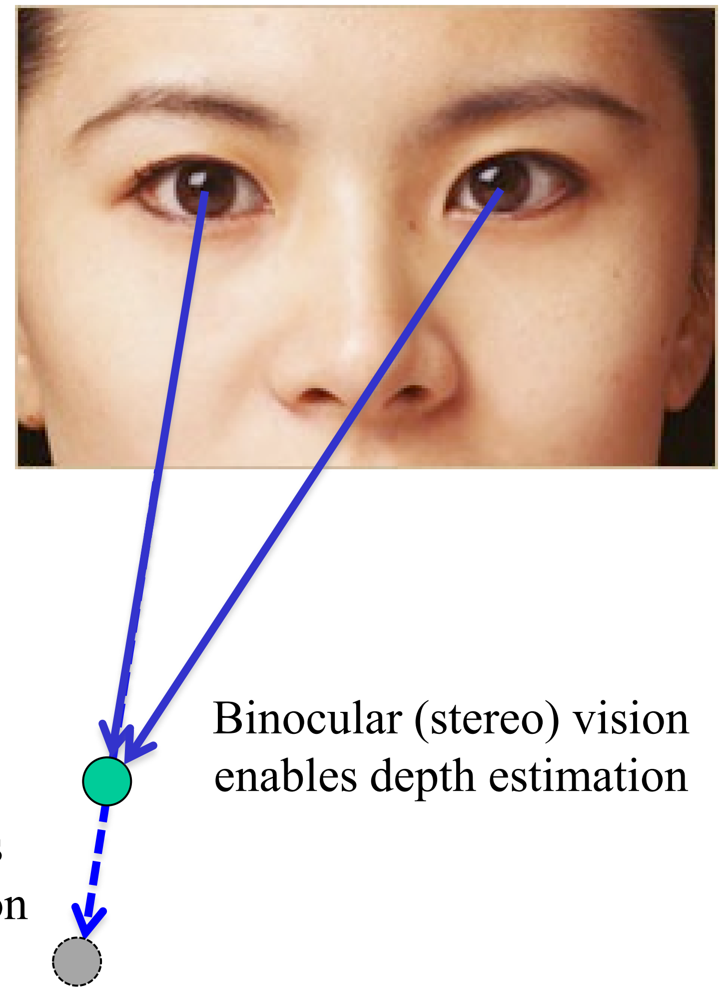
Announcements

- Assignment 1 is due Oct 25, 11:59 PM
- Assignment 2 will be released Oct 25
  - Due Nov 8, 11:59 PM

Why Do We Have Two Eyes?

1. Redundancy – If we lose one, we're not blind
2. Larger field of view
3. Ability to recover depth for some points

Why Do We Have Two Eyes?



Binocular (stereo) vision enables depth estimation

Depth information is lost in image formation

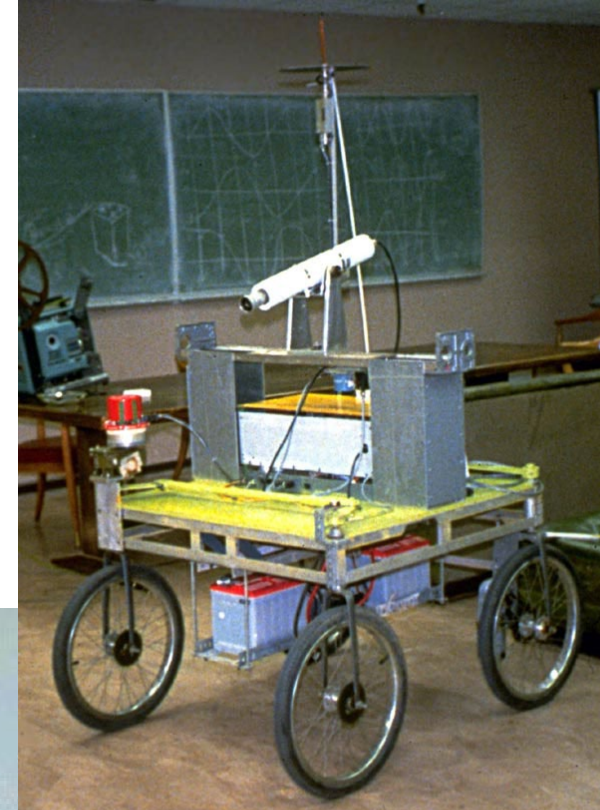
Stereo Vision

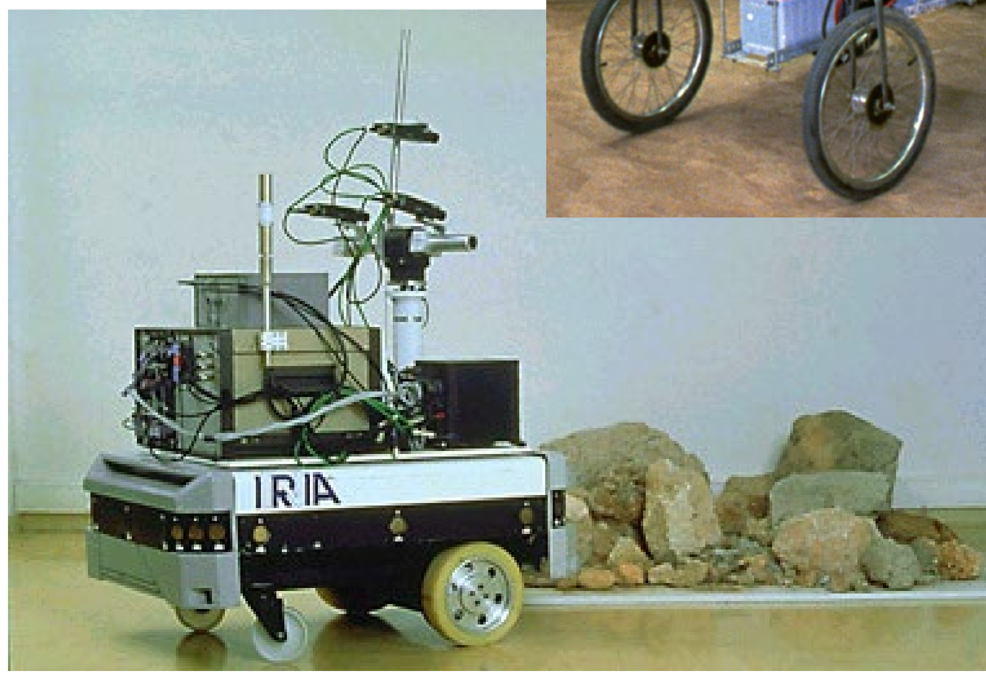
Holmes Stereoscope

An Application: Mobile Robot Navigation

Mobi, Stanford, 1987

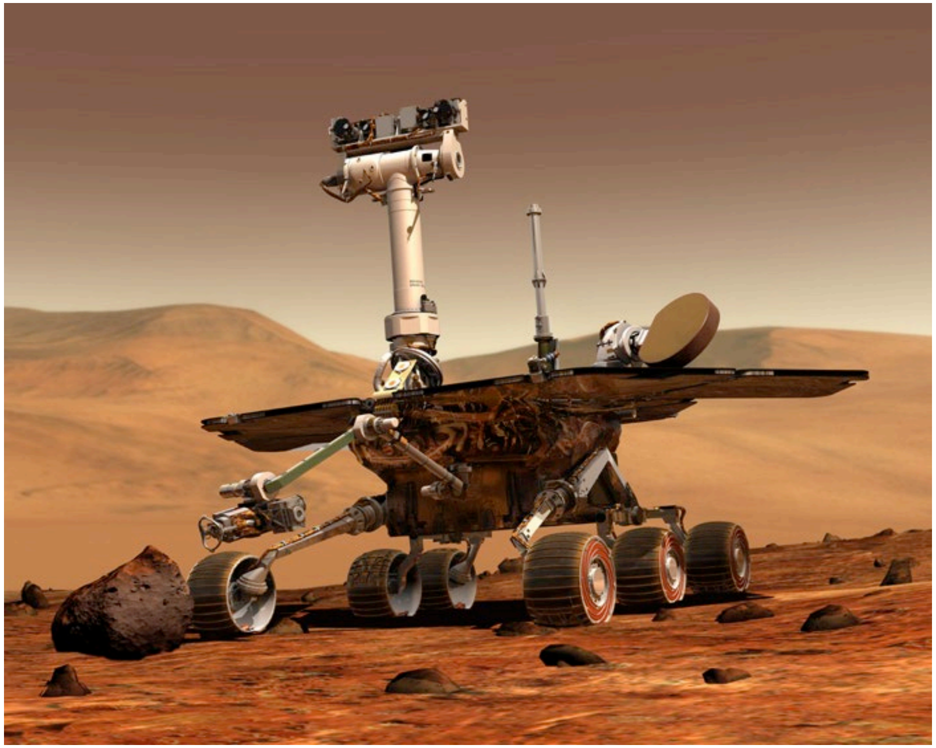
The Stanford Cart,
H. Moravec, 1979

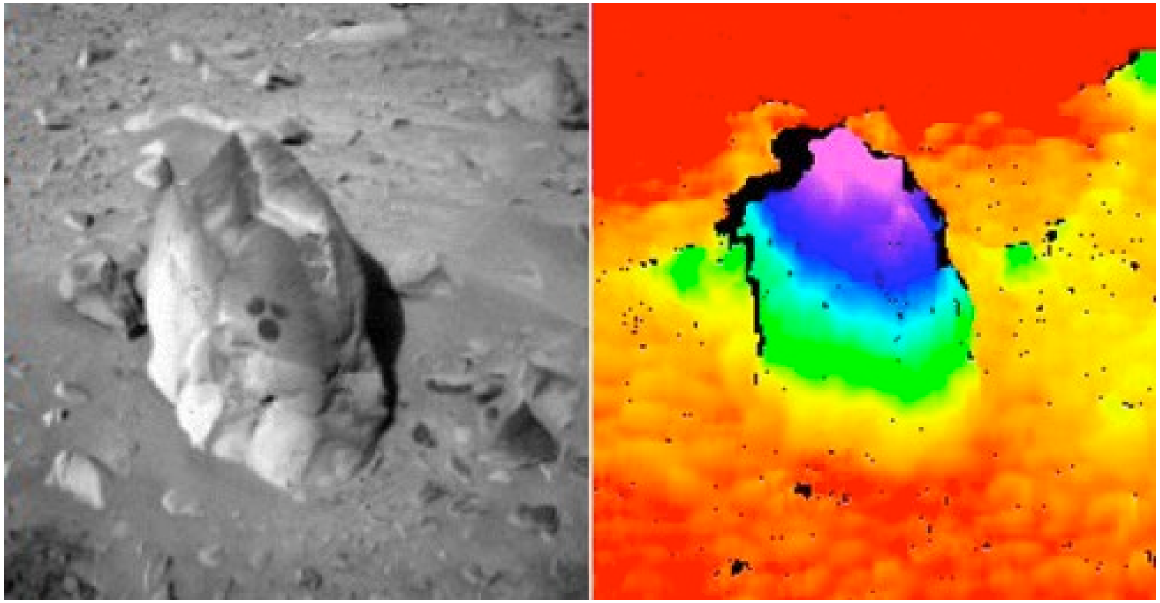




INRIA Mobile
Robot 1990

Mars Exploratory Rovers: Spirit and Opportunity, 2004

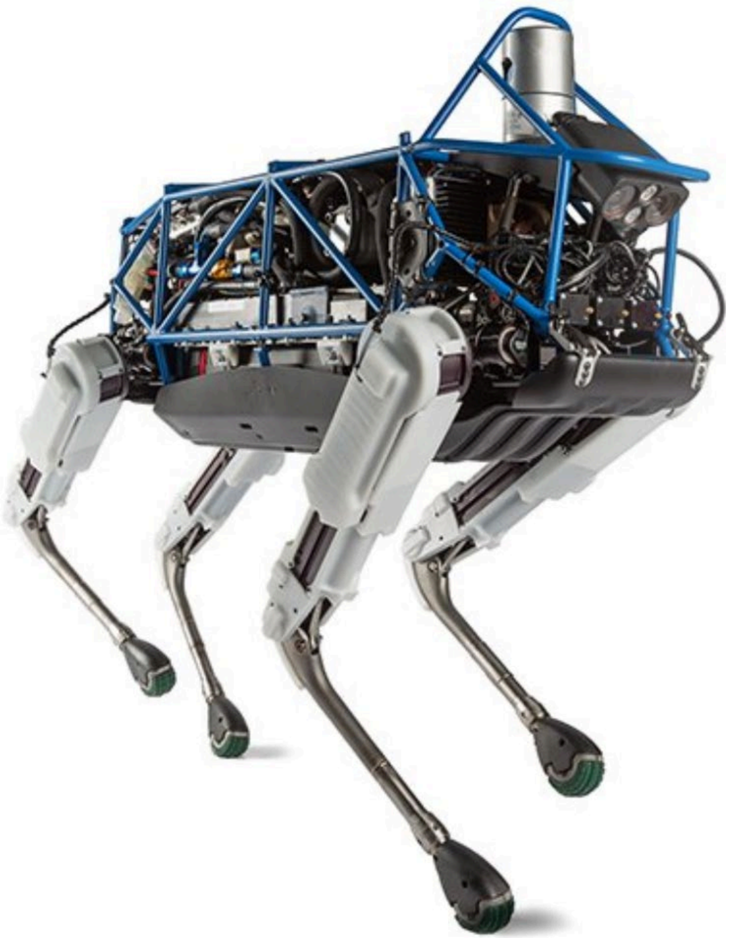




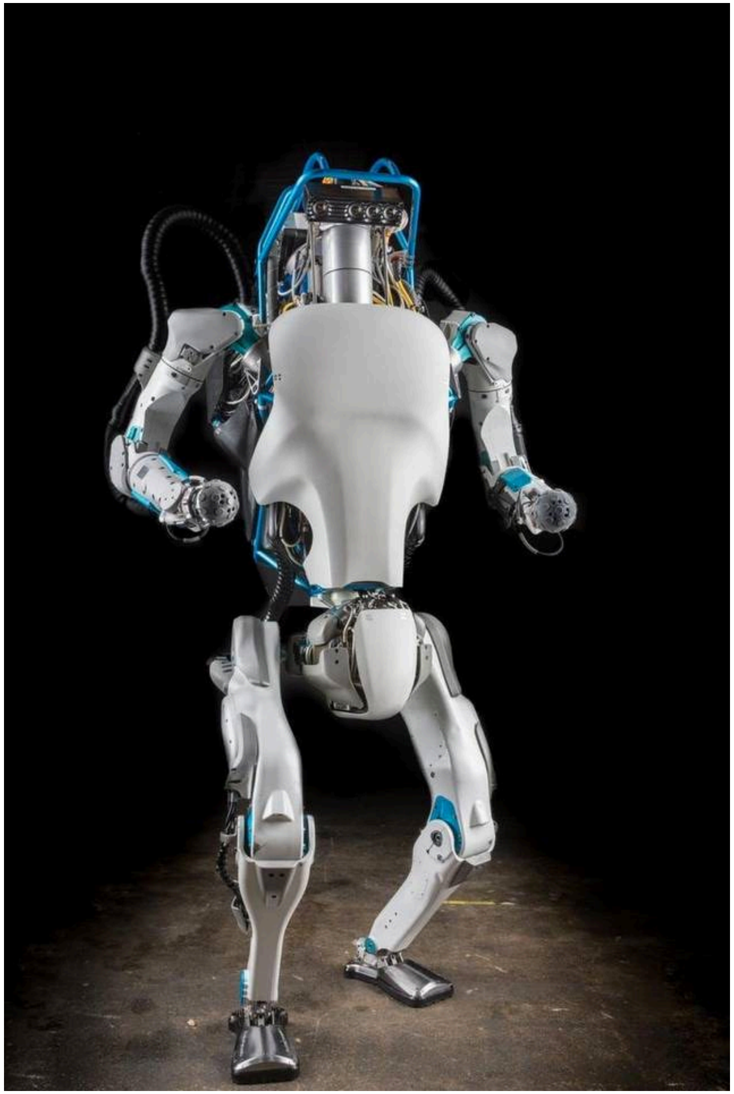
Curiosity Rover (2012)

- Navigation cameras (Navcams) B&W, 45^o field of view
- Hazard avoidance cameras (hazcams), 4 pairs, 120^o field of view

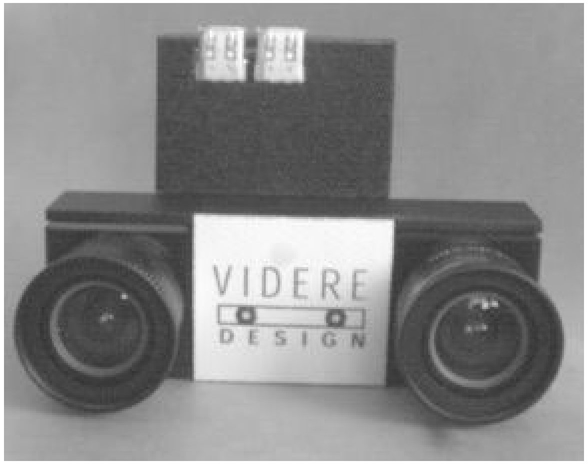
Boston Dynamics



Stereo + Lidar

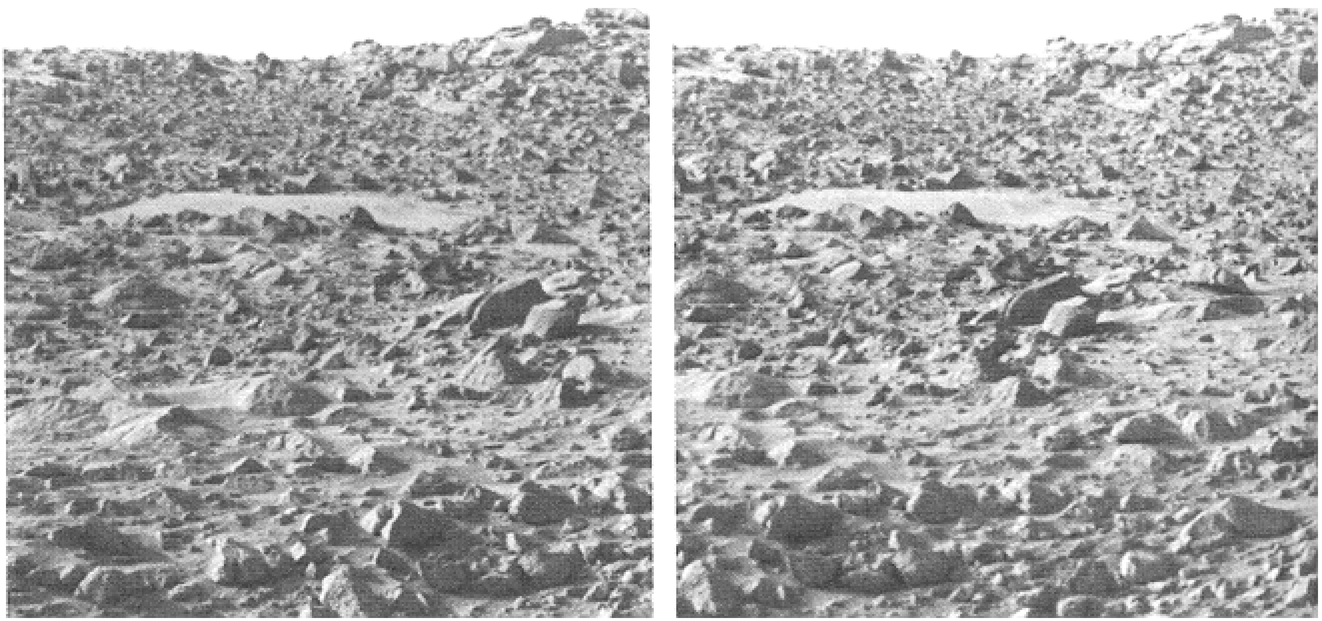


Commercial Stereo Heads



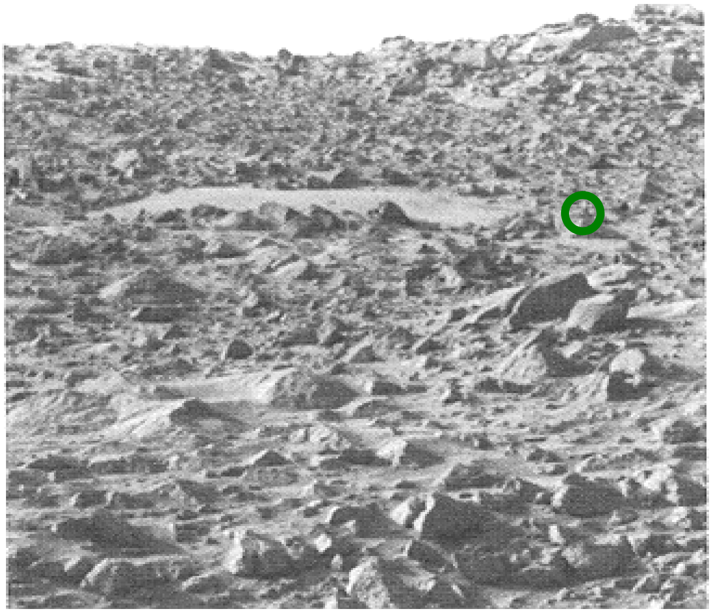
Binocular Stereopsis: Mars

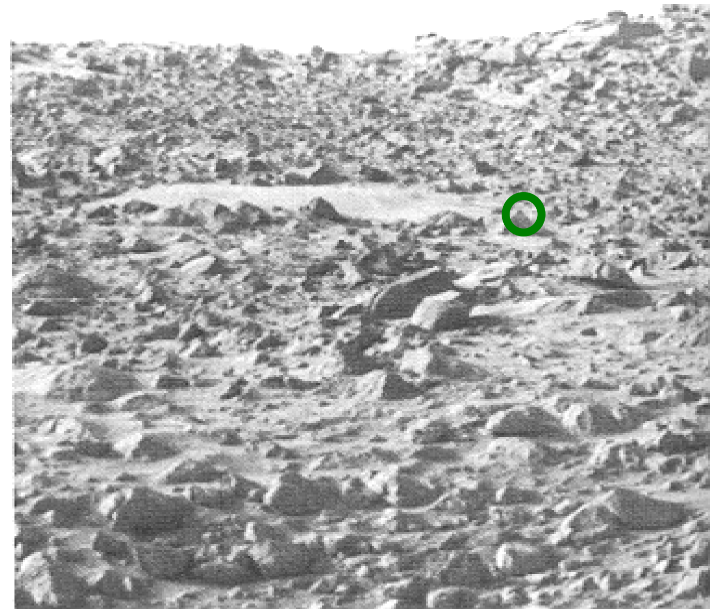
Given two images of a scene where relative locations of cameras are known, estimate depth of all common scene points.



Two images of Mars (Viking Lander)

Matching complexity (naïve)





Input: two images that are n x n pixels

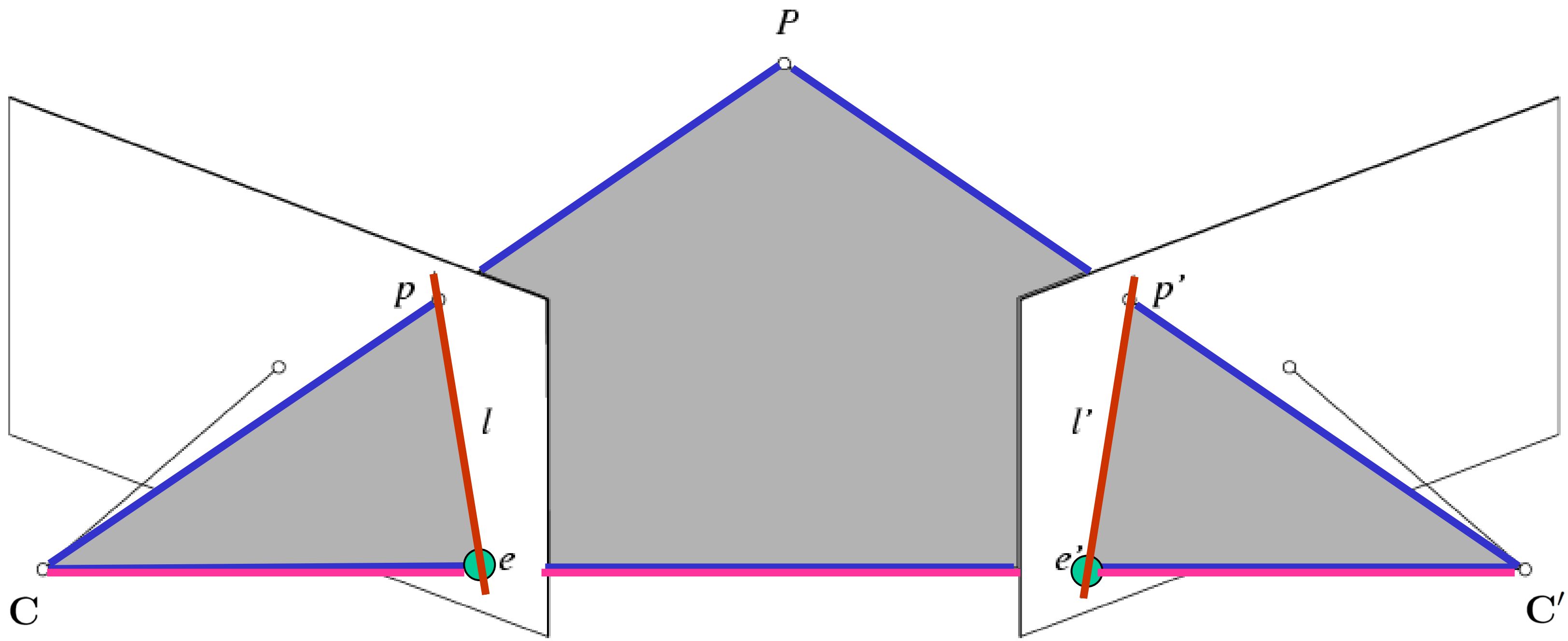
For a given point in the left image, where do we look in the right image?

For each point in left mage, there are $O(n^2)$ possible matching points in right image.

With n^2 pixels in left image, complexity of matching is $O(n^4)$

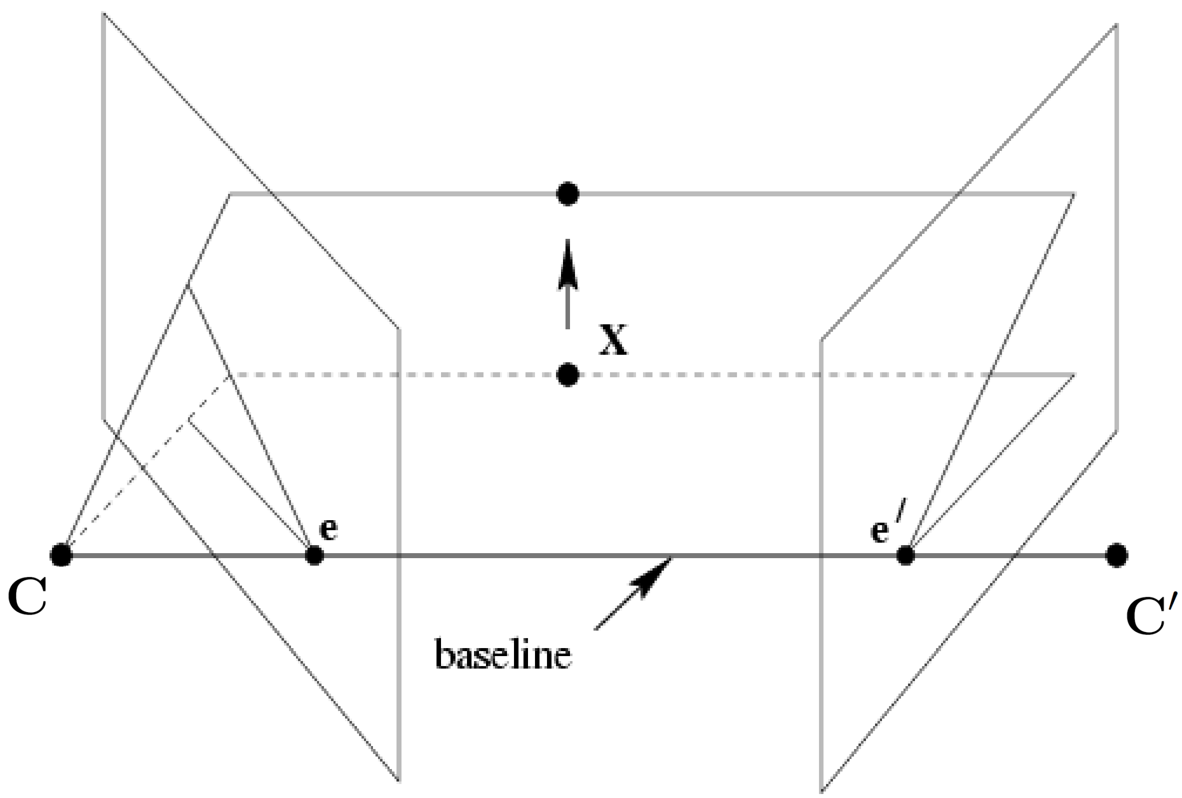
Can we do better?

Epipolar Geometry Terminology



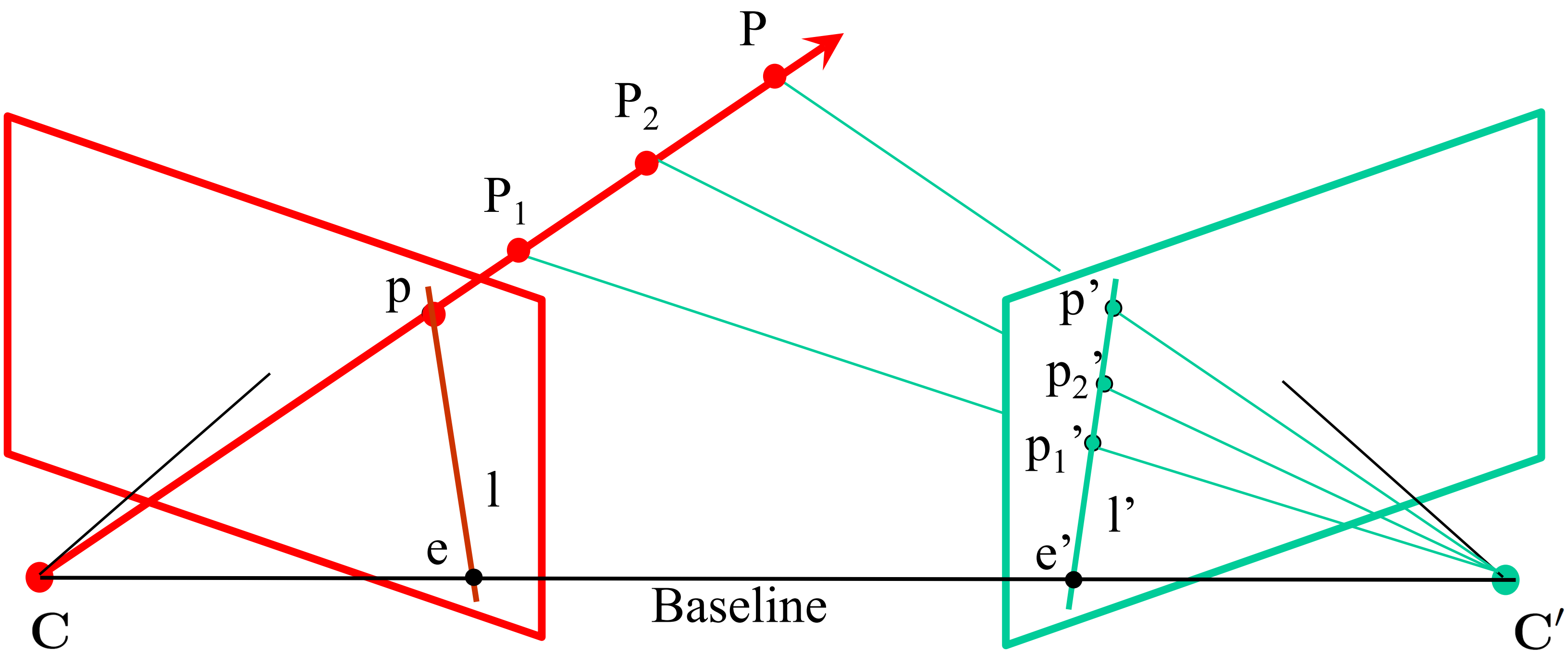
- **Baseline:** line connecting camera centers (of projection) **C** and **C'**
- **Epipoles (e, e'):** Two intersection points of baseline with image planes
- **Epipolar Plane:** Any plane that contains the baseline
- **Epipolar Lines (l, l'):** Pair of lines from intersection of an epipolar plane with the two image planes

Family of Epipolar Planes



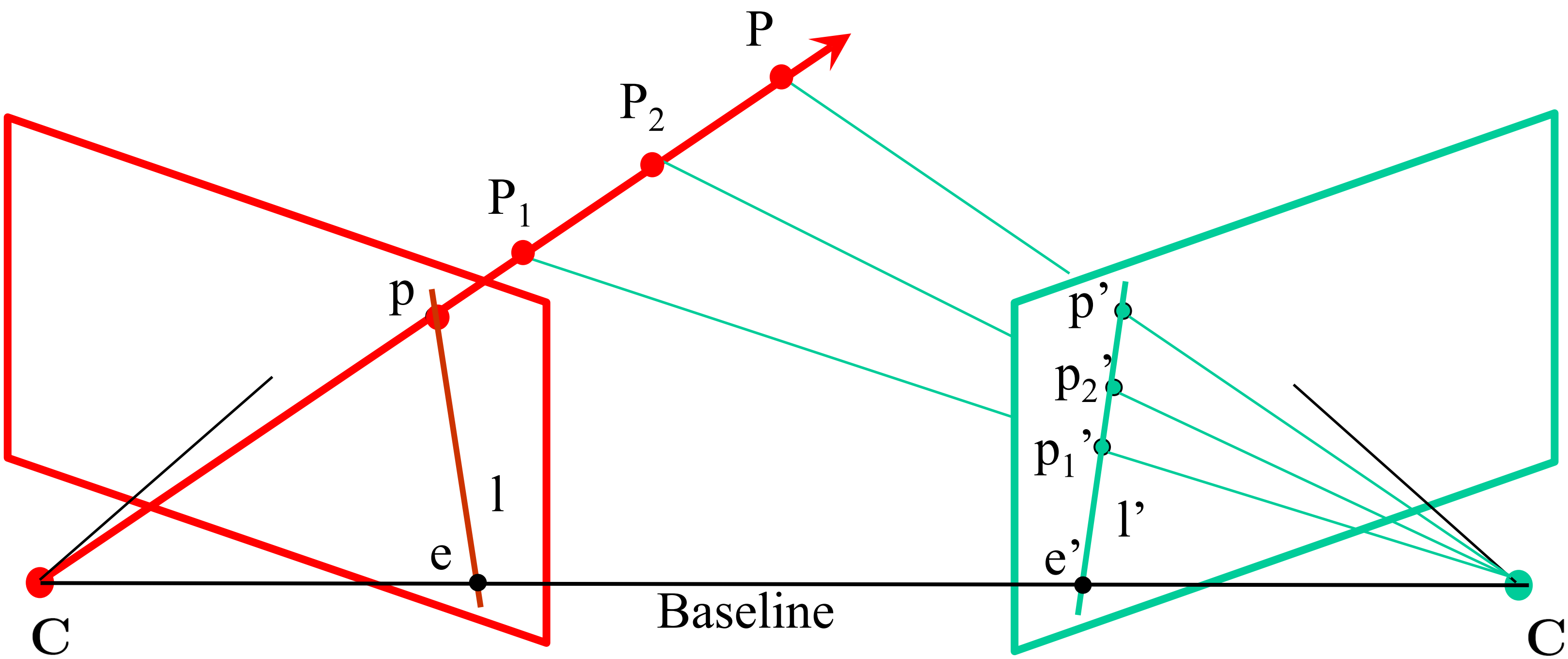
- Epipolar Plane: Any plane that contains the baseline
- The set of epipolar planes is a family of all planes passing through the baseline and can be parameterized by the angle about baseline

Epipolar matching



- Potential matches for **p** have to lie on the corresponding epipolar line **l'**
- Epipolar line **l'** passes through epipole **e'**, the intersection of the baseline with the image plane
- Potential matches for **p'** have to lie on the corresponding epipolar line **l**

Epipolar matching complexity



Using epipolar matching, complexity is reduced from $O(n^4)$ to $O(n^3)$. Why?

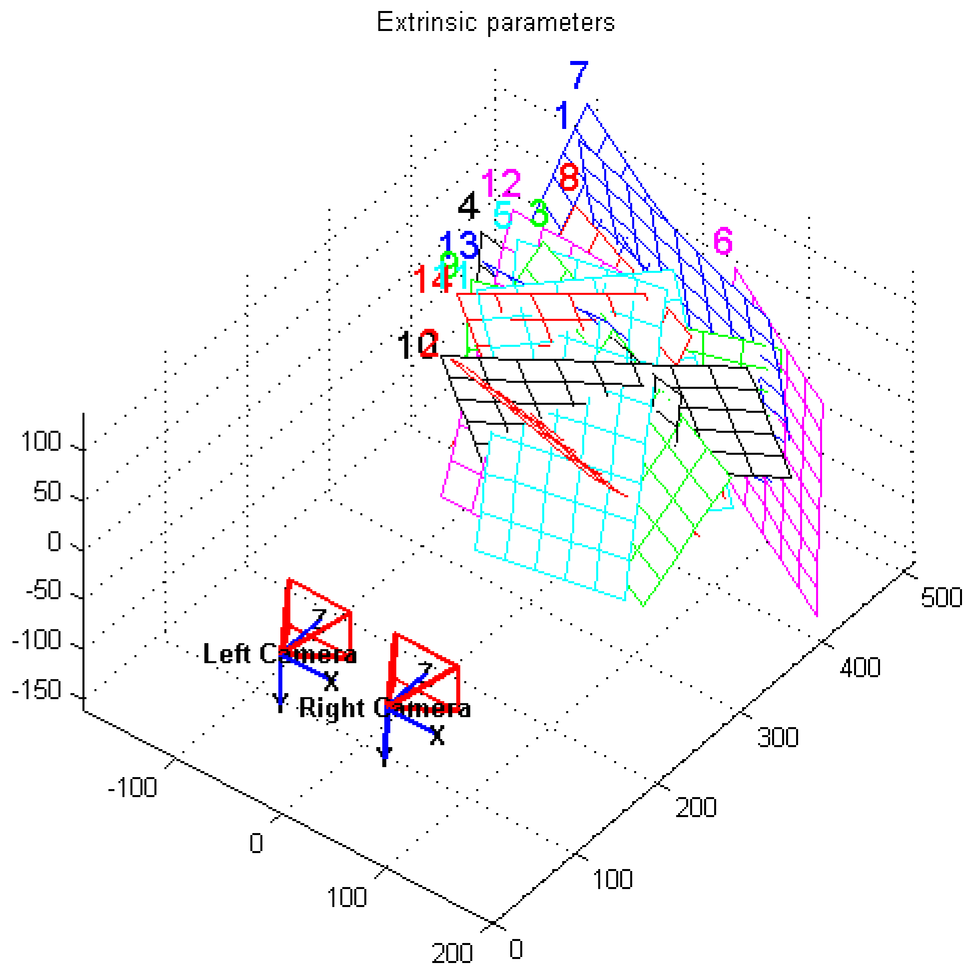
- There are n^2 points in the left image
- For each point in the left image, all candidate matches are on an epipolar line in the right image, and the length of the epipolar line is $O(n)$
- Therefore, match complexity is $O(n^2 * n) = O(n^3)$

Stereo Vision Outline

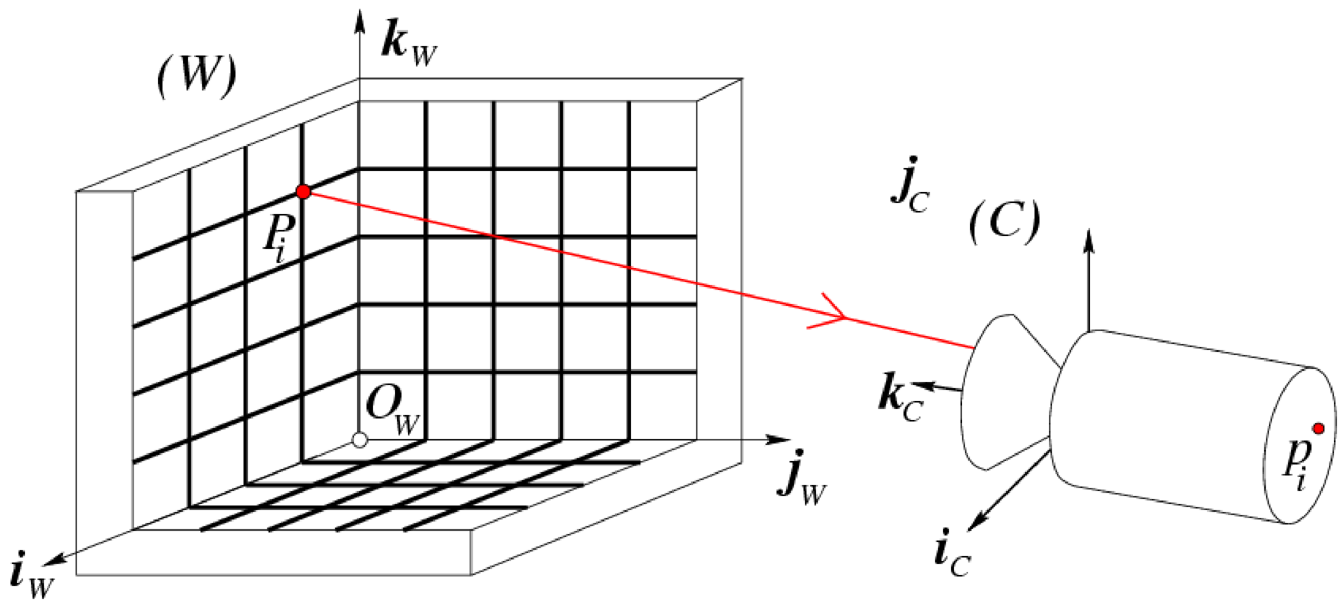
- Offline
  – Calibration of stereo cameras
- Online
  1. Acquire stereo images
  2. Epipolar rectify stereo images
  3. Establish correspondence
  4. Estimate depth

Calibration of stereo cameras

1. From images of known calibration fixture, determine intrinsic parameters $\mathbf{K}_1$, $\mathbf{K}_2$ and extrinsic relation of two cameras $\mathbf{R}_1$, $\mathbf{t}_1$ and $\mathbf{R}_2$, $\mathbf{t}_2$
2. Compute the relative rotation $\mathbf{R}$ and translation $\mathbf{t}$ of the two cameras from $\mathbf{R}_1$, $\mathbf{t}_1$ and $\mathbf{R}_2$, $\mathbf{t}_2$
3. Compute the essential matrix $\mathbf{E}$



Camera calibration



- Given n points $\mathbf{P}_1,\ldots, \mathbf{P}_n$ with known 3-D position and their pixel coordinates $\mathbf{x}_1,\ldots, \mathbf{x}_n$, estimate intrinsic **K** and extrinsic camera parameters and lens distortion parameters
- See textbook for details
- Camera Calibration Toolbox for Matlab (Bouguet) Archived

http://www.vision.caltech.edu/bouguetj/calib_doc/

Compute the rotation and translation of the second camera relative to the first one

$$\mathbf{x}_1 = \mathtt{K}_1[\mathtt{I} \,|\, \mathbf{0}] \begin{bmatrix} \mathtt{R}_1 & \mathbf{t}_1 \\ \mathbf{0}^\top & 1 \end{bmatrix} \mathbf{X}$$

$$\mathbf{x}_1 = \mathtt{K}_1[\mathtt{I} \,|\, \mathbf{0}] \mathbf{X}_{\text{cam},1}$$

$$\text{where } \mathbf{X}_{\text{cam},1} = \begin{bmatrix} \mathtt{R}_1 & \mathbf{t}_1 \\ \mathbf{0}^\top & 1 \end{bmatrix} \mathbf{X}$$

$$\begin{bmatrix} \mathtt{R}_1 & \mathbf{t}_1 \\ \mathbf{0}^\top & 1 \end{bmatrix}^{-1} \mathbf{X}_{\text{cam},1} = \mathbf{X}$$

$$\mathbf{x}_2 = \mathtt{K}_2[\mathtt{I} \,|\, \mathbf{0}] \begin{bmatrix} \mathtt{R}_2 & \mathbf{t}_2 \\ \mathbf{0}^\top & 1 \end{bmatrix} \mathbf{X}$$

$$\mathbf{x}_2 = \mathtt{K}_2[\mathtt{I} \,|\, \mathbf{0}] \begin{bmatrix} \mathtt{R}_2 & \mathbf{t}_2 \\ \mathbf{0}^\top & 1 \end{bmatrix} \begin{bmatrix} \mathtt{R}_1 & \mathbf{t}_1 \\ \mathbf{0}^\top & 1 \end{bmatrix}^{-1} \mathbf{X}_{\text{cam},1}$$

$$\mathbf{x}_2 = \mathtt{K}_2[\mathtt{I} \,|\, \mathbf{0}] \begin{bmatrix} \mathtt{R}_2 & \mathbf{t}_2 \\ \mathbf{0}^\top & 1 \end{bmatrix} \begin{bmatrix} \mathtt{R}_1^\top & -\mathtt{R}_1^\top \mathbf{t}_1 \\ \mathbf{0}^\top & 1 \end{bmatrix} \mathbf{X}_{\text{cam},1}$$

$$\mathbf{x}_2 = \mathtt{K}_2[\mathtt{I} \,|\, \mathbf{0}] \begin{bmatrix} \mathtt{R}_2\mathtt{R}_1^\top & \mathbf{t}_2 - \mathtt{R}_2\mathtt{R}_1^\top \mathbf{t}_1 \\ \mathbf{0}^\top & 1 \end{bmatrix} \mathbf{X}_{\text{cam},1}$$

$$\mathbf{x}_2 = \mathtt{K}_2[\mathtt{I} \,|\, \mathbf{0}] \begin{bmatrix} \mathtt{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \mathbf{X}_{\text{cam},1}$$

where $\mathtt{R} = \mathtt{R}_2\mathtt{R}_1^\top$ and

$\mathbf{t} = \mathbf{t}_2 - \mathtt{R}_2\mathtt{R}_1^\top \mathbf{t}_1$

Image points

- Image points in pixel coordinates

$$\mathbf{x} = \mathtt{K}[\mathtt{I} \mid \mathbf{0}] \begin{bmatrix} \mathtt{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \mathbf{X}$$

$$\mathbf{x} = \mathtt{K}[\mathtt{R} \mid \mathbf{t}]\mathbf{X}$$

- Image points in normalized coordinates

$$\mathbf{x} = \mathtt{K}[\mathtt{R} \mid \mathbf{t}]\mathbf{X}$$

$$\mathtt{K}^{-1}\mathbf{x} = [\mathtt{R} \mid \mathbf{t}]\mathbf{X}$$

$$\hat{\mathbf{x}} = [\mathtt{R} \mid \mathbf{t}]\mathbf{X} \text{ where } \hat{\mathbf{x}} = \mathtt{K}^{-1}\mathbf{x}$$

Image points in normalized coordinates

$$\mathbf{x} = \mathtt{K}[\mathtt{I} \,|\, \mathbf{0}] \begin{bmatrix} \mathtt{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \mathbf{X}$$

$$\mathtt{K}^{-1}\mathbf{x} = [\mathtt{I} \,|\, \mathbf{0}] \begin{bmatrix} \mathtt{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \mathbf{X}$$

$$\hat{\mathbf{x}} = [\mathtt{I} \,|\, \mathbf{0}] \begin{bmatrix} \mathtt{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \mathbf{X} \text{ where } \hat{\mathbf{x}} = \mathtt{K}^{-1}\mathbf{x}$$

$$\hat{\mathbf{x}} = [\mathtt{I} \,|\, \mathbf{0}]\mathbf{X}_{\text{cam}}$$

$$\hat{\mathbf{x}} = [\mathtt{I} \,|\, \mathbf{0}] \begin{bmatrix} \tilde{\mathbf{X}}_{\text{cam}} \\ 1 \end{bmatrix}$$

$$\hat{\mathbf{x}} = \tilde{\mathbf{X}}_{\text{cam}} \text{ (up to nonzero scale)}$$

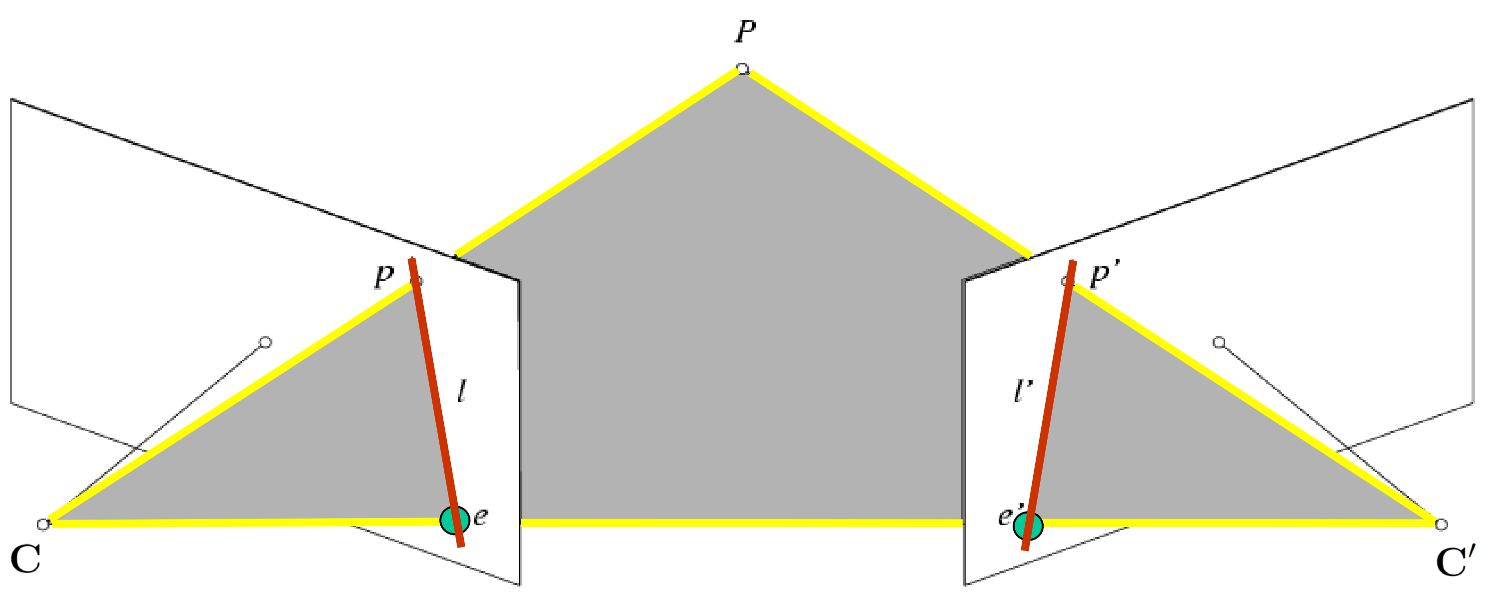
The essential matrix

3D point $\tilde{\mathbf{X}}' = \lambda' \hat{\mathbf{x}}'$ in the second camera coordinate frame.
3D point $\tilde{\mathbf{X}} = \lambda \hat{\mathbf{x}}$ in the first camera coordinate frame.
Map $\tilde{\mathbf{X}}$ from first camera coordinate frame to second camera coordinate frame.

$$\tilde{\mathbf{X}}' = \mathtt{R}\tilde{\mathbf{X}} + \mathbf{t}$$

$$\tilde{\mathbf{X}}' = \mathtt{R}(\lambda \hat{\mathbf{x}}) + \mathbf{t}$$

$$\tilde{\mathbf{X}}' = \lambda \mathtt{R}\hat{\mathbf{x}} + \mathbf{t}$$



The essential matrix

3D point $\tilde{\mathbf{X}}' = \lambda' \hat{\mathbf{x}}'$ in the second camera coordinate frame.
3D point $\tilde{\mathbf{X}} = \lambda \hat{\mathbf{x}}$ in the first camera coordinate frame.
Map $\tilde{\mathbf{X}}$ from first camera coordinate frame to second camera coordinate frame.

$$\tilde{\mathbf{X}}' = \mathtt{R}\tilde{\mathbf{X}} + \mathbf{t}$$
$$\tilde{\mathbf{X}}' = \mathtt{R}(\lambda \hat{\mathbf{x}}) + \mathbf{t}$$
$$\tilde{\mathbf{X}}' = \lambda \mathtt{R} \hat{\mathbf{x}} + \mathbf{t}$$

$$\lambda \mathtt{R} \hat{\mathbf{x}} + \mathbf{t} = \lambda' \hat{\mathbf{x}}'$$
$$[\mathbf{t}]_\times (\lambda \mathtt{R} \hat{\mathbf{x}} + \mathbf{t}) = [\mathbf{t}]_\times (\lambda' \hat{\mathbf{x}}')$$
$$\lambda [\mathbf{t}]_\times \mathtt{R} \hat{\mathbf{x}} = \lambda' [\mathbf{t}]_\times \hat{\mathbf{x}}'$$
$$\hat{\mathbf{x}}'^\top (\lambda [\mathbf{t}]_\times \mathtt{R} \hat{\mathbf{x}}) = \hat{\mathbf{x}}'^\top (\lambda' [\mathbf{t}]_\times \hat{\mathbf{x}}')$$
$$\lambda \hat{\mathbf{x}}'^\top [\mathbf{t}]_\times \mathtt{R} \hat{\mathbf{x}} = 0$$
$$\hat{\mathbf{x}}'^\top [\mathbf{t}]_\times \mathtt{R} \hat{\mathbf{x}} = 0$$

The epipolar constraint $\quad \hat{\mathbf{x}}'^\top \mathtt{E} \hat{\mathbf{x}} = 0$ where $\mathtt{E} = [\mathbf{t}]_\times \mathtt{R}$

Essential Matrix
(Longuet-Higgins, 1981)

Cross product using a skew symmetric matrix

- The cross product $\mathbf{a} \times \mathbf{b}$ of two 3-vectors $\mathbf{a} = (a_1, a_2, a_3)^\top$ and $\mathbf{b} = (b_1, b_2, b_3)^\top$ can be expressed a matrix-vector product $[\mathbf{a}]_\times \mathbf{b}$, where $[\mathbf{a}]_\times$ is the 3x3 skew symmetric matrix

$$[\mathbf{a}]_\times = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}$$

- A matrix **S** is skew symmetric if and only if $\mathbf{S} = -\mathbf{S}^{T}$
- The determinant of a skew symmetric matrix is 0

The essential matrix

- Maps a point (in normalized coordinates) in the first image to its corresponding epipolar line (in normalized coordinates) in the second image

$$\hat{\boldsymbol{\ell}}' = \mathtt{E}\hat{\mathbf{x}}$$

- The epipolar line passes through the corresponding point in the second image

$$\hat{\mathbf{x}}'^\top \hat{\boldsymbol{\ell}}' = 0$$

$$\hat{\mathbf{x}}'^\top \mathtt{E}\hat{\mathbf{x}} = 0 \quad \text{The epipolar constraint}$$

- Every epipolar line passes through the epipole

$$\hat{\mathbf{e}}'^\top \hat{\boldsymbol{\ell}}' = 0$$

The essential matrix

- Maps a point (in normalized coordinates) in the second image to its corresponding epipolar line (in normalized coordinates) in the first image

$$\hat{\boldsymbol{\ell}} = \mathtt{E}^\top \hat{\mathbf{x}}'$$

$$\hat{\boldsymbol{\ell}}^\top = \hat{\mathbf{x}}'^\top \mathtt{E}$$

- The epipolar line passes through the corresponding point in the first image

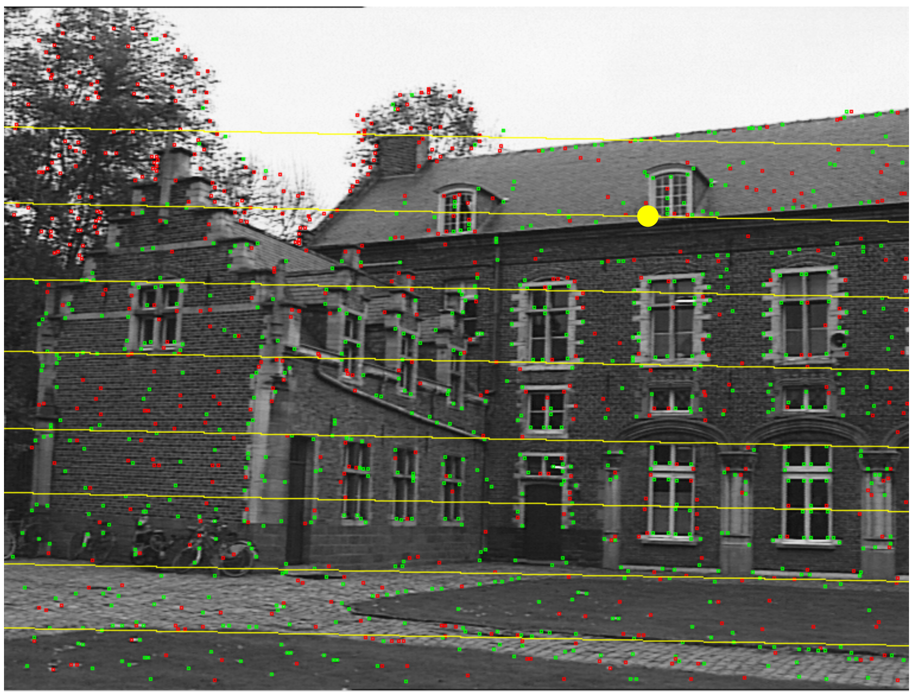
$$\hat{\boldsymbol{\ell}}^\top \hat{\mathbf{x}} = 0$$

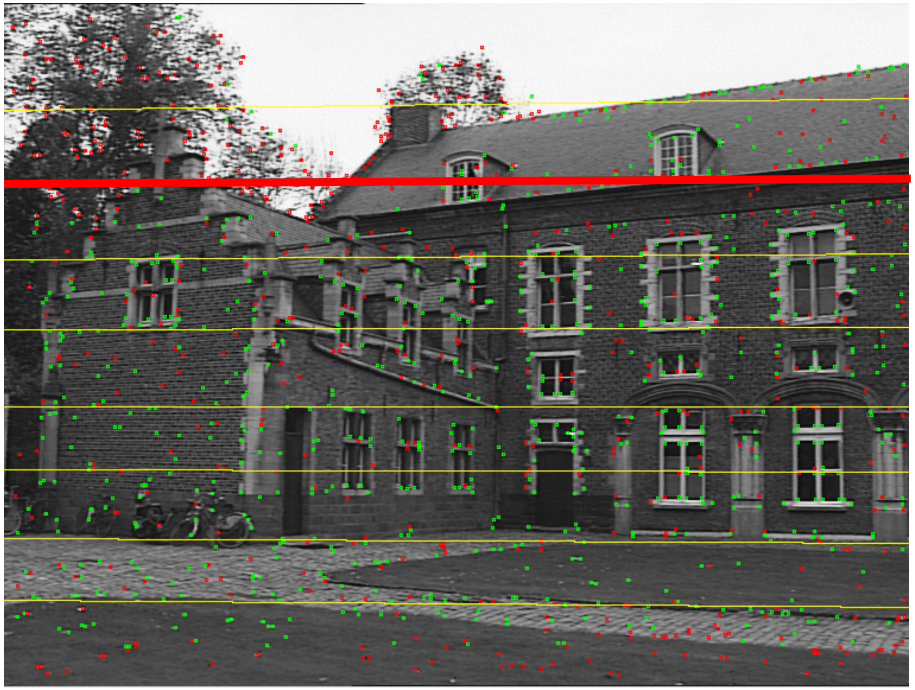
$$\hat{\mathbf{x}}'^\top \mathtt{E} \hat{\mathbf{x}} = 0$$ The epipolar constraint

- Every epipolar line passes through the epipole

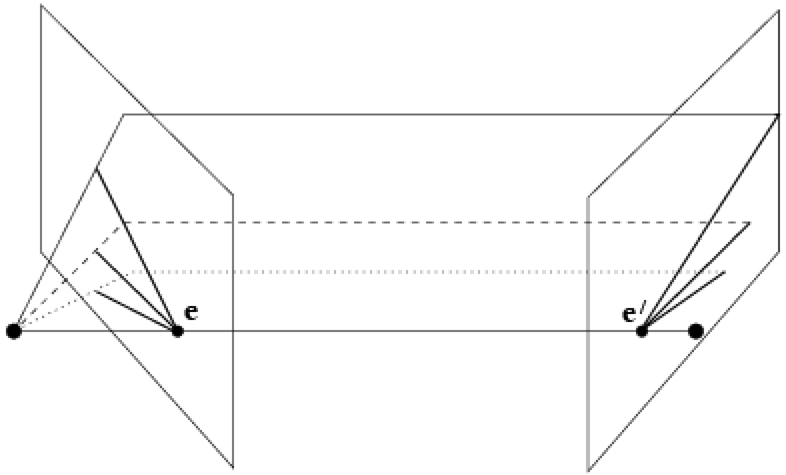
$$\hat{\boldsymbol{\ell}}^\top \hat{\mathbf{e}} = 0$$

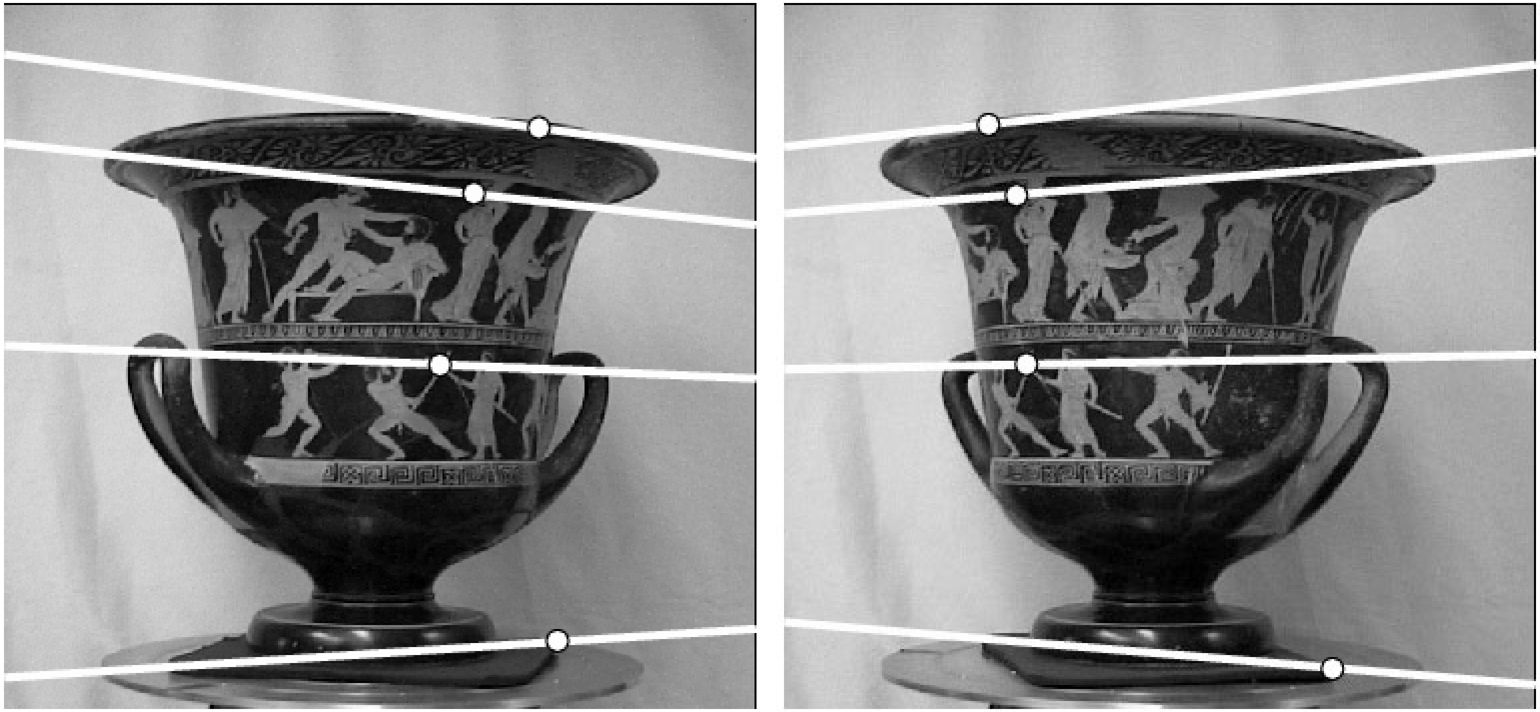
Example of using the essential matrix





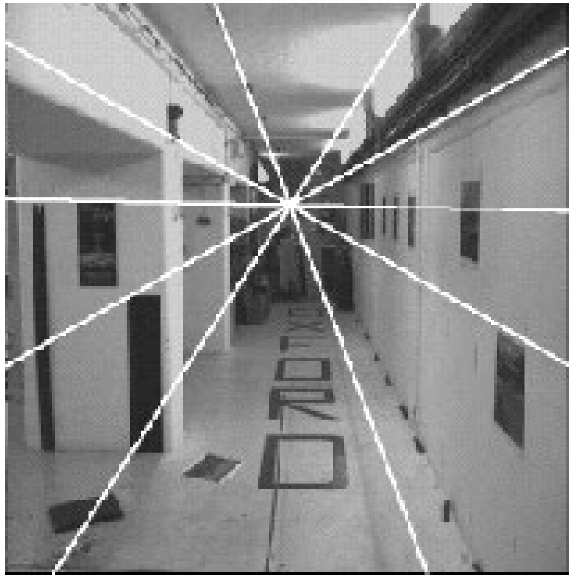
Another example: converging cameras

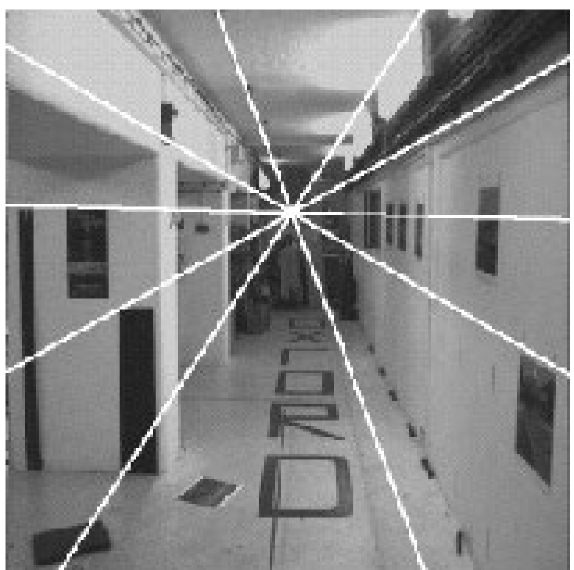


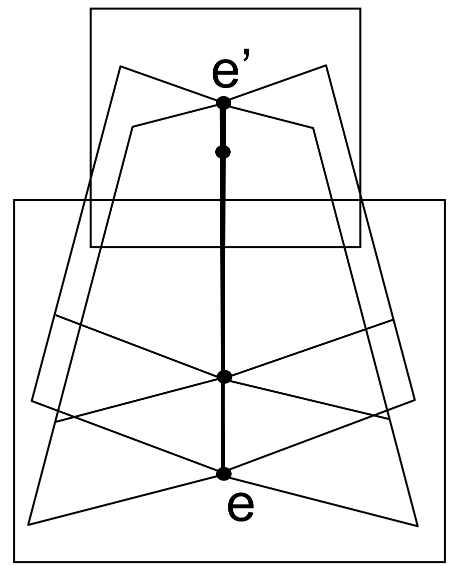


courtesy of Andrew Zisserman

Another example: second camera in front of first camera



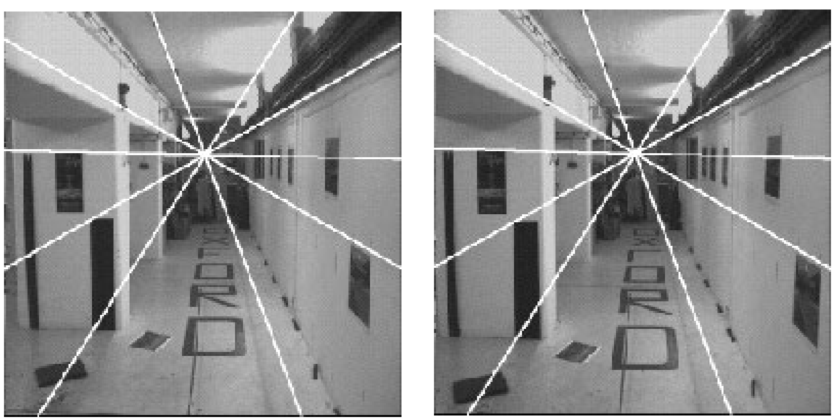




courtesy of Andrew Zisserman

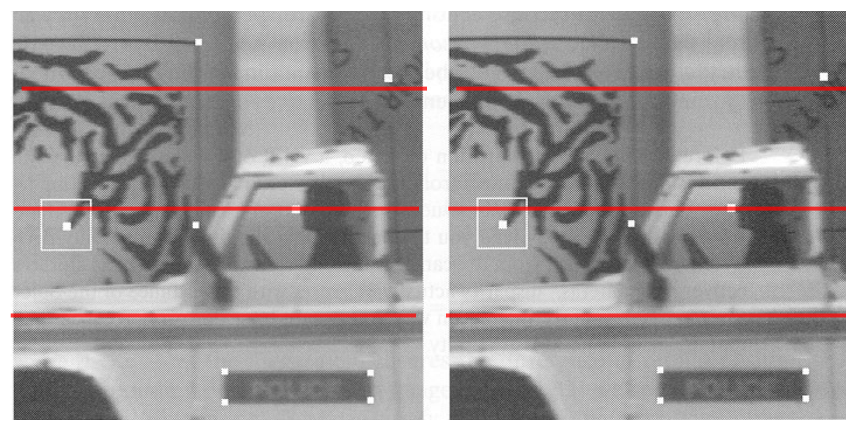
Epipolar rectify stereo images

- Epipolar geometry reduces matching complexity from $O(n^4)$ to $O(n^3)$
- But matching requires comparing points across pairs of epipolar lines which may have arbitrary orientation. That can be costly to index.
- Is there a more convenient epipolar geometry?



Slanted epipolar lines

vs

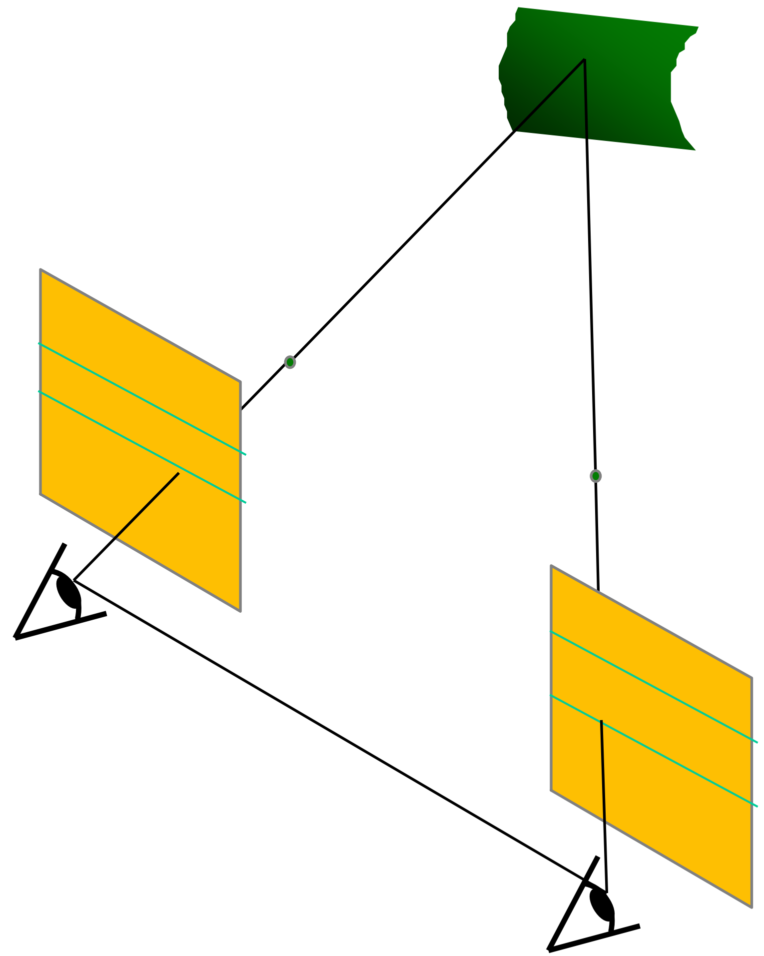


Horizontal, row aligned epipolar lines

Cameras with a convenient epipolar geometry

- When two cameras have parallel optical axes and these axis are orthogonal to the baseline, the epipolar lines are parallel

- When rows of the two images are parallel to the baseline, the epipolar lines are horizontal rows of the two images



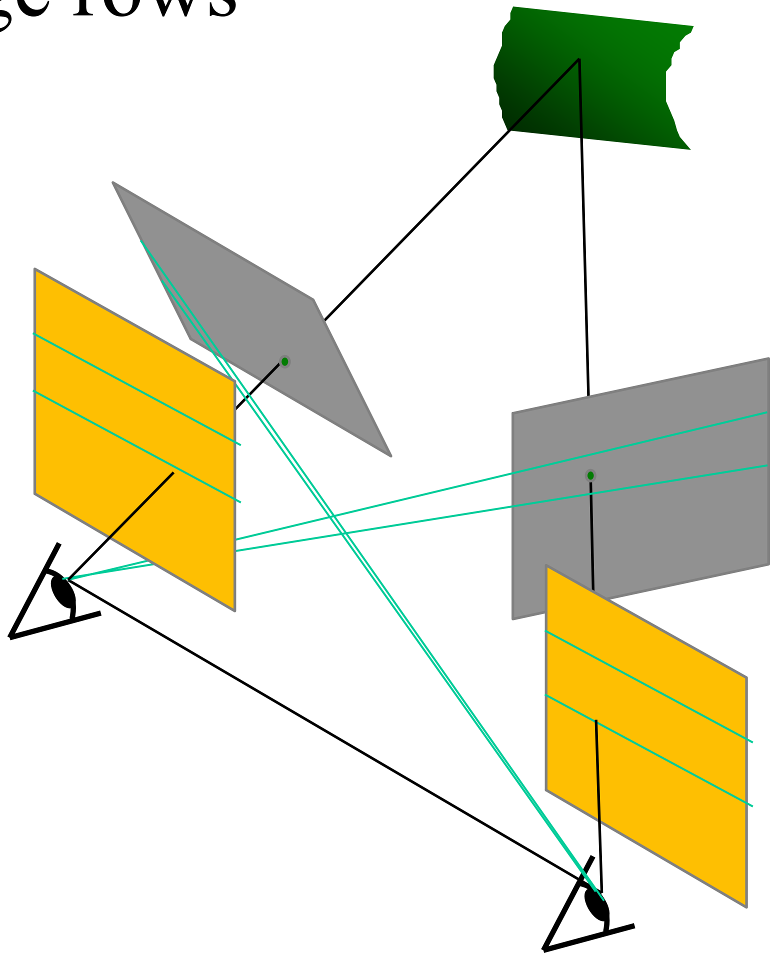
Cameras with a convenient epipolar geometry

- When two cameras have parallel optical axes and these axis are orthogonal to the baseline, the epipolar lines are parallel

- When rows of the two images are parallel to the baseline, the epipolar lines are horizontal rows of the two images

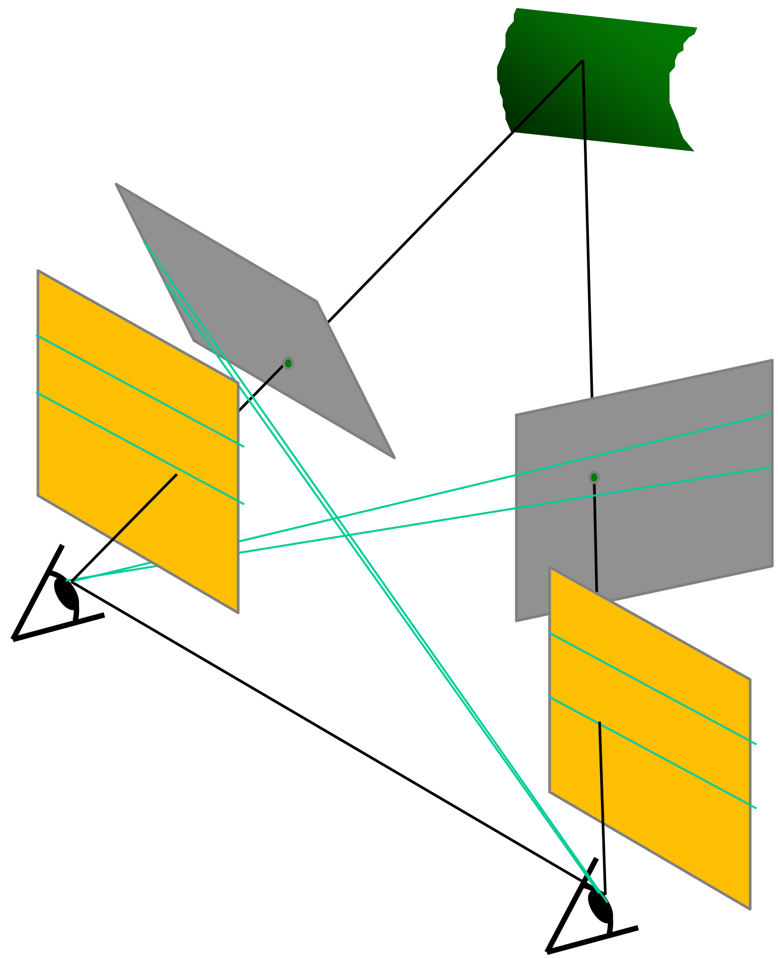
What if stereo geometry is not convenient?

Rectification: Given a pair of images, transform both images so that epipolar lines are image rows



Epipolar rectification

- Given a pair of images, transform both images so that epipolar lines are image rows
- Create pair of virtual cameras
  - The virtual cameras have the same camera centers as real cameras
  - Both virtual cameras have the same:
    - Camera rotation matrix R
    - Camera calibration matrix K



Epipolar rectification

- Given calibrated stereo cameras (i.e., $\mathbf{K}_1$, $\mathbf{R}_1$, $\mathbf{t}_1$, $\mathbf{K}_2$, $\mathbf{R}_2$, $\mathbf{t}_2$) determine the (same) rotation matrix $\mathbf{R}$ and (same) calibration matrix $\mathbf{K}$ of the virtual cameras
- To minimize image distortion:
  - For the calibration matrix
    - For principal point, set x_0 and y_0 to average of input x_0 and y_0 values, respectively
    - Set two focal length parameters to average of all four input focal length parameters (results in square pixels)
    - Set skew to 0
  - For the rotation matrix $\mathbf{R}$, interpolate halfway between the two 3D rotations embodied by $\mathbf{R}_1$ and $\mathbf{R}_2$

Epipolar rectification

- For the rotation matrix $\mathbf{R}$, interpolate halfway between the two 3D rotations embodied by $\mathbf{R}_1$ and $\mathbf{R}_2$
  – Rotating a camera does not change its center $\mathbf{C}$, but does change its translation $\mathbf{t}$

$$\tilde{\mathbf{0}}_{\text{cam}} = \mathtt{R}\tilde{\mathbf{C}} + \mathbf{t}$$
$$-\mathtt{R}\tilde{\mathbf{C}} = \mathbf{t}$$
$$\tilde{\mathbf{C}} = -\mathtt{R}^\top \mathbf{t}$$

$$\mathbf{x} = \mathtt{K}[\mathtt{R} \mid \mathbf{t}]\mathbf{X}$$
$$\mathbf{x} = \mathtt{K}[\mathtt{R} \mid -\mathtt{R}\tilde{\mathbf{C}}]\mathbf{X}$$
$$\mathbf{x} = \mathtt{KR}[\mathtt{I} \mid -\tilde{\mathbf{C}}]\mathbf{X}$$

$$\tilde{\mathbf{C}} = -\mathtt{R}_{\text{virtual}}^\top \mathbf{t}_{\text{virtual}}$$
$$\tilde{\mathbf{C}} = -\mathtt{R}_{\text{real}}^\top \mathbf{t}_{\text{real}}$$
$$-\mathtt{R}_{\text{virtual}}^\top \mathbf{t}_{\text{virtual}} = -\mathtt{R}_{\text{real}}^\top \mathbf{t}_{\text{real}}$$
$$\mathbf{t}_{\text{virtual}} = \mathtt{R}_{\text{virtual}}\mathtt{R}_{\text{real}}^\top \mathbf{t}_{\text{real}}$$

Rectification transformation matrices

- Transformation from image acquired by real camera to image acquired by virtual camera

$$\mathbf{x}_{\text{real}} = \mathtt{K}_{\text{real}}\mathtt{R}_{\text{real}}[\mathtt{I} \mid -\tilde{\mathbf{C}}]\mathbf{X}$$

$$\mathtt{R}_{\text{real}}^{\top}\mathtt{K}_{\text{real}}^{-1}\mathbf{x}_{\text{real}} = [\mathtt{I} \mid -\tilde{\mathbf{C}}]\mathbf{X}$$

$$\mathbf{x}_{\text{virtual}} = \mathtt{K}_{\text{virtual}}\mathtt{R}_{\text{virtual}}[\mathtt{I} \mid -\tilde{\mathbf{C}}]\mathbf{X}$$

$$\mathbf{x}_{\text{virtual}} = \mathtt{K}_{\text{virtual}}\mathtt{R}_{\text{virtual}}\mathtt{R}_{\text{real}}^{\top}\mathtt{K}_{\text{real}}^{-1}\mathbf{x}_{\text{real}}$$

$$\mathbf{x}_{\text{virtual}} = \mathtt{H}\mathbf{x}_{\text{real}}, \text{ where } \mathtt{H} = \mathtt{K}_{\text{virtual}}\mathtt{R}_{\text{virtual}}\mathtt{R}_{\text{real}}^{\top}\mathtt{K}_{\text{real}}^{-1}$$

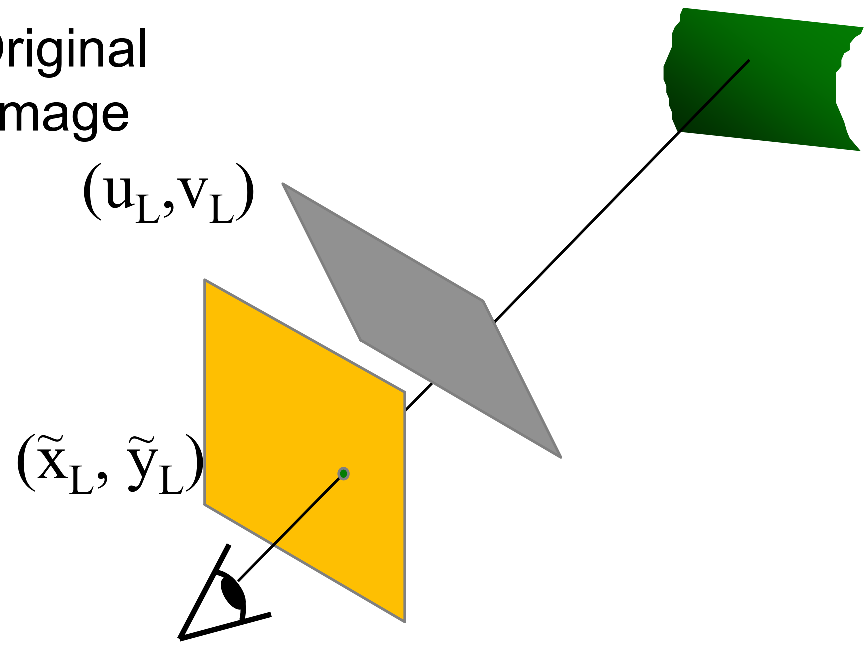
$$\begin{bmatrix} x_{\text{virtual}} \\ y_{\text{virtual}} \\ w_{\text{virtual}} \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x_{\text{real}} \\ y_{\text{real}} \\ w_{\text{real}} \end{bmatrix}$$

Rectification

Under perspective projection, the mapping from a plane to a plane is given by a 2D projective transformation (homography)

Rectified Image → Original Image

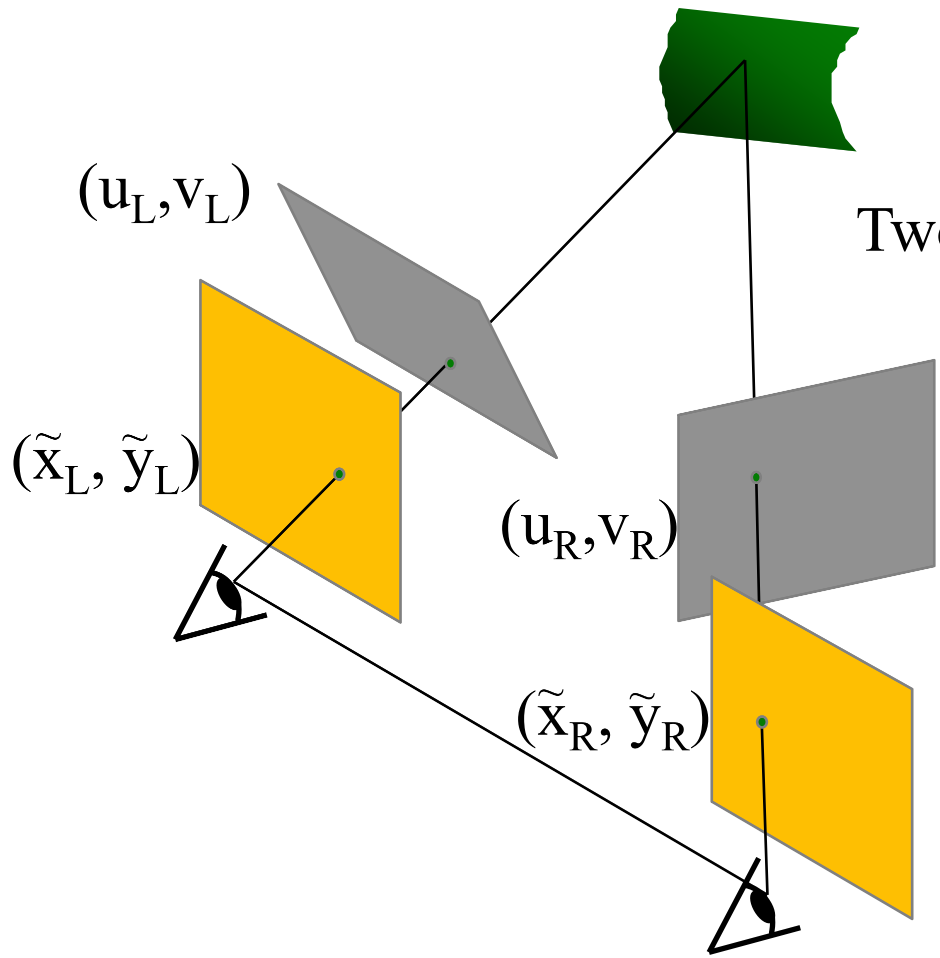
$$\begin{bmatrix} x_L \\ y_L \\ w_L \end{bmatrix} = H_L \begin{bmatrix} u_L \\ v_L \\ 1 \end{bmatrix}$$



Rectification

Under perspective projection, the mapping from a plane to a plane is given by a 2D projective transformation (homography)

$$\begin{bmatrix} x_L \\ y_L \\ w_L \end{bmatrix} = H_L \begin{bmatrix} u_L \\ v_L \\ 1 \end{bmatrix}$$

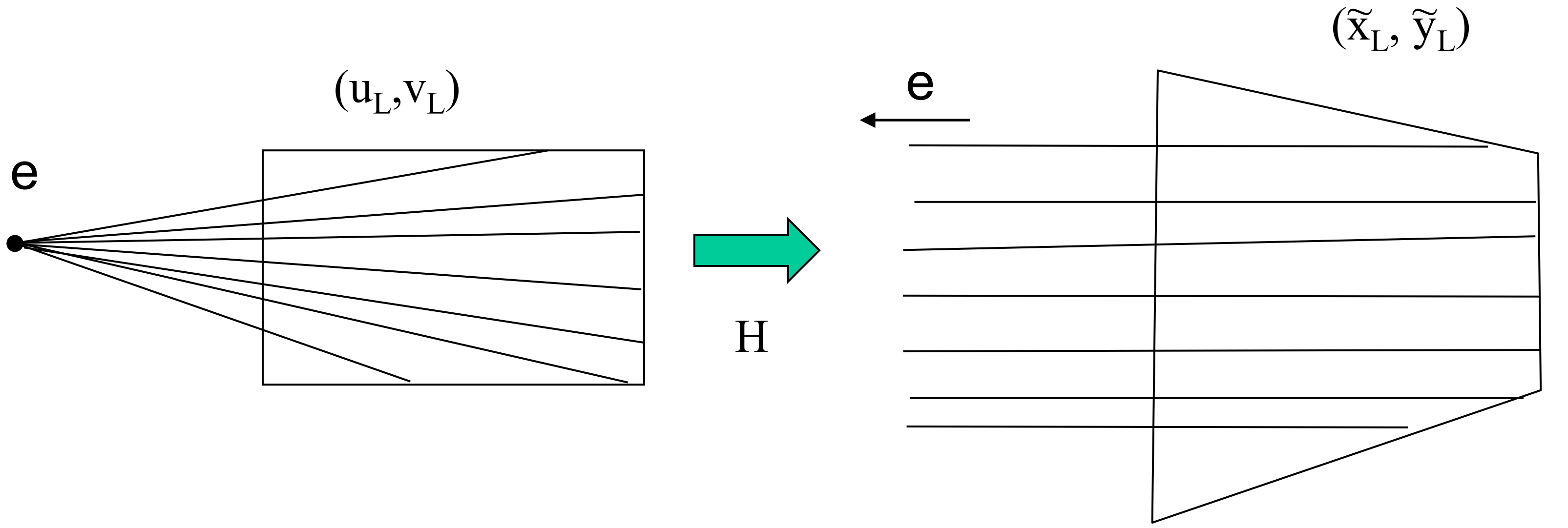


Two images
Two homographies
H_L, H_R

$$\begin{bmatrix} x_R \\ y_R \\ w_R \end{bmatrix} = H_R \begin{bmatrix} u_R \\ v_R \\ 1 \end{bmatrix}$$

Image pair rectification

Apply projective transformation so that epipolar lines correspond to horizontal scanlines



H should map epipole e to (1,0,0), a point at infinity on the x-axis

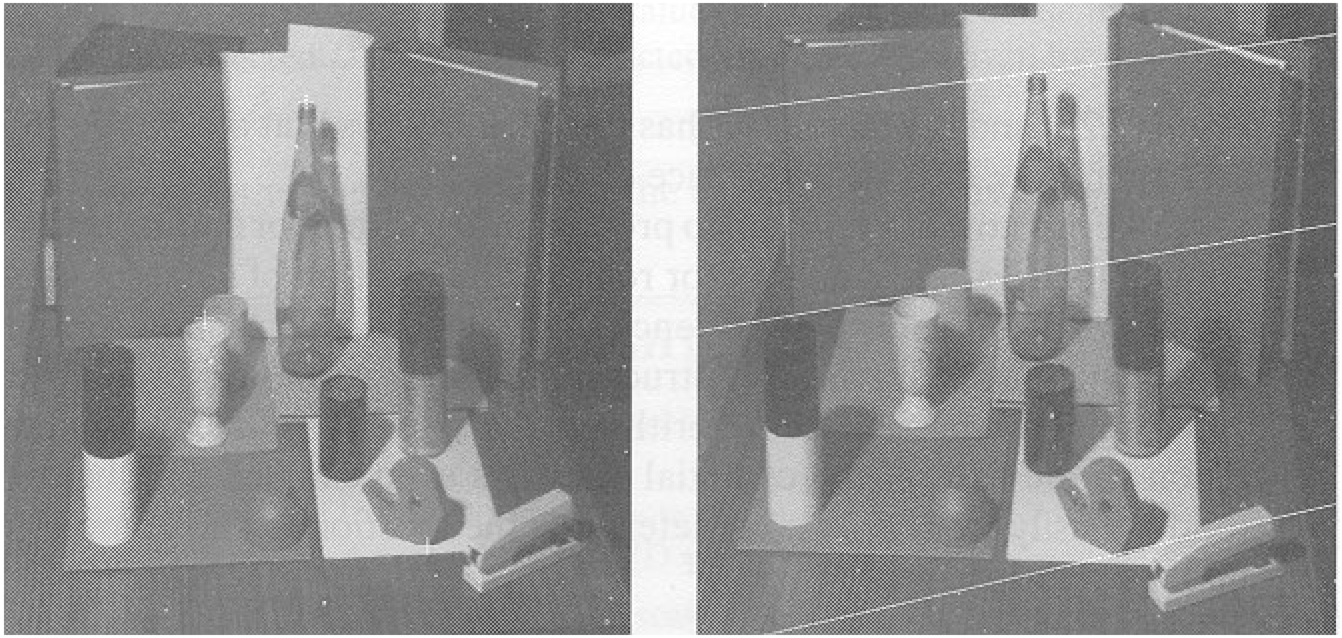
H should minimize image distortion

$$\begin{bmatrix}1\\0\\0\end{bmatrix} = He$$

Note that rectified images are usually not rectangular

Rectification

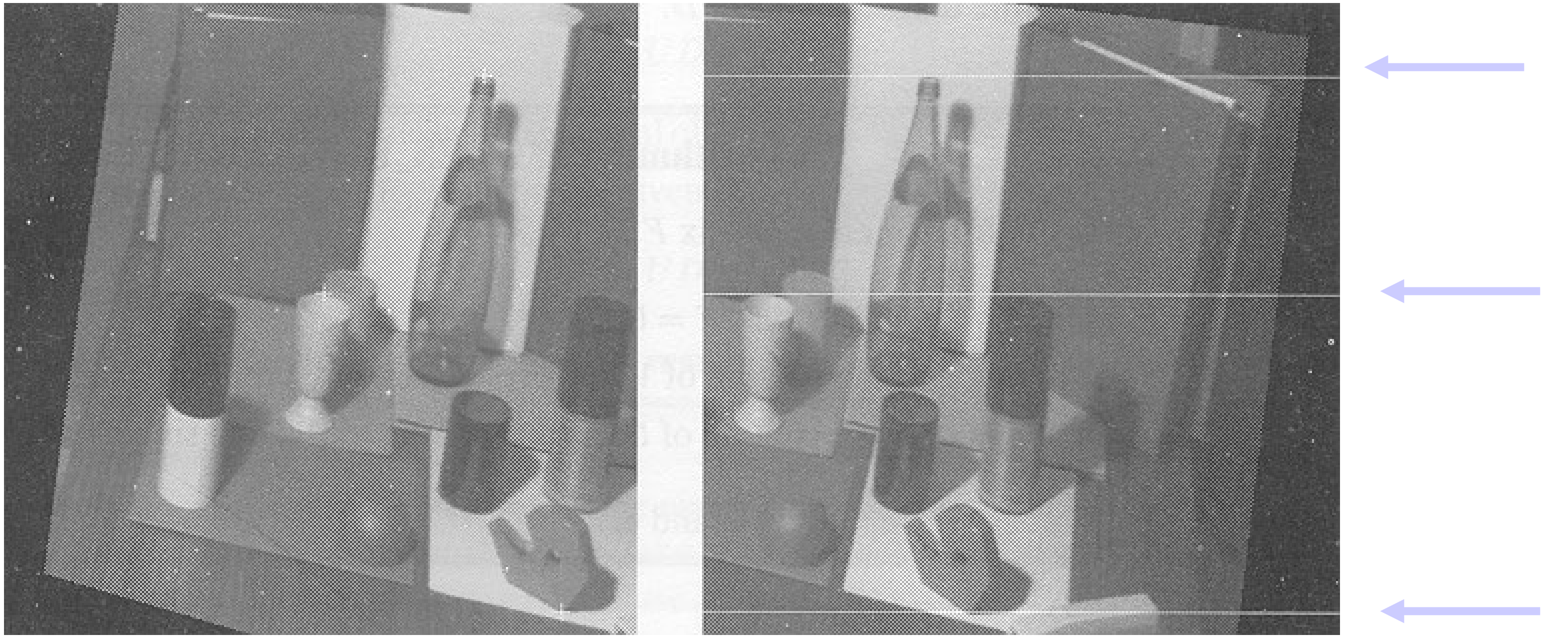
Given a pair of images, transform both images so that epipolar lines are scan lines.



Input Images

Rectification

Given a pair of images, transform both images so that epipolar lines are scan lines.



Rectified Images

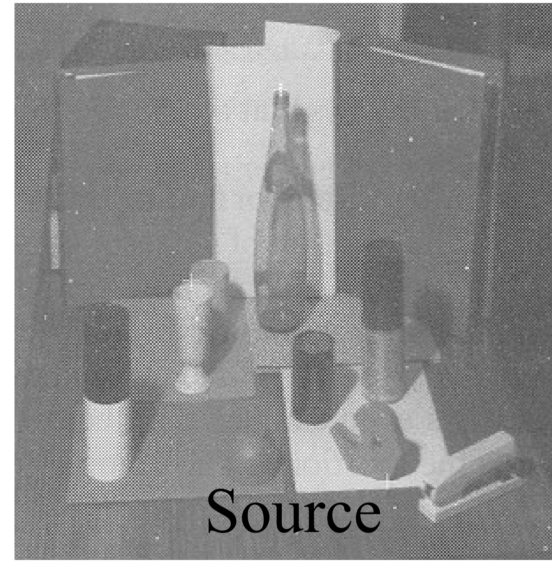
epipolar lines run parallel with the *x*-axis and are aligned between two views (no *y* disparity)

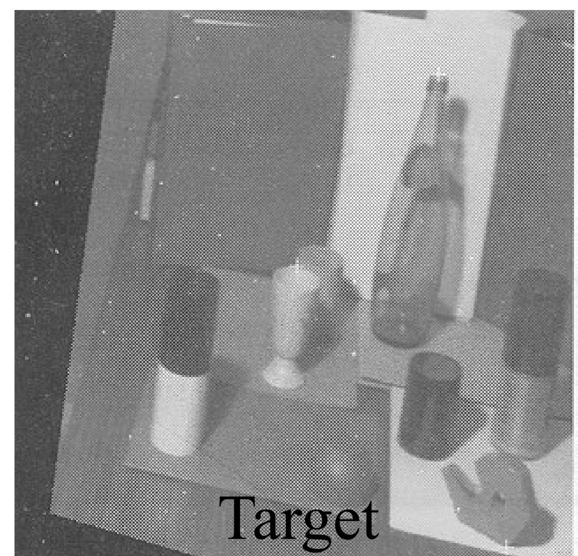
Epipolar rectification

- Input: Source image I and
  Rectification matrix H
- Forward "warping" method
  - For each pixel coordinate $\mathbf{x}_{\text{real}}$ in source image, map it to pixel coordinate $\mathbf{x}_{\text{virtual}}$ in target image

$$\mathbf{x}_{\text{virtual}} = \mathtt{H}\mathbf{x}_{\text{real}}$$

  - Problem: there is no guarantee that every pixel in target Image will be written to
    - If target image is larger than source image or target image is highly stretched, then there may be missing points that appear as speckles or lines





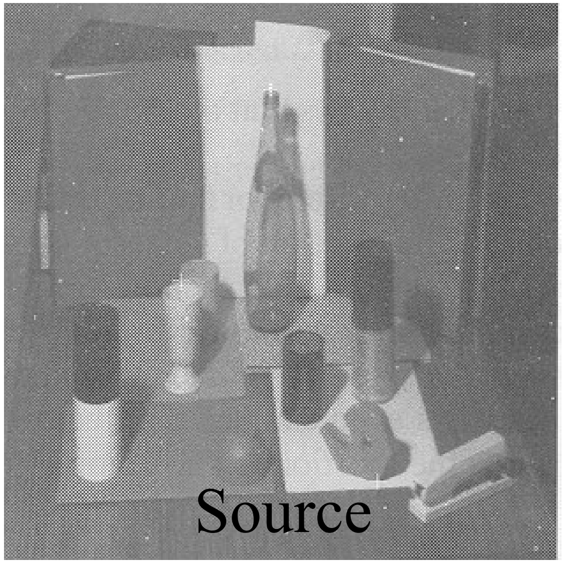
Epipolar rectification

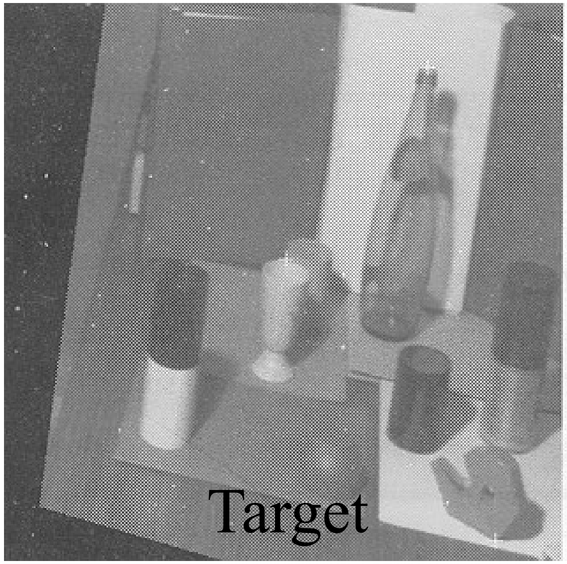
- Input: Source image I and
  Rectification matrix H
- Backward "warping" method
  - For each pixel coordinate $\mathbf{x}_{\text{virtual}}$ in target image, map it to pixel coordinate $\mathbf{x}_{\text{real}}$ in source image

$$\mathbf{x}_{\text{real}} = \mathtt{H}^{-1}\mathbf{x}_{\text{virtual}}$$

    - Interpolate pixel values in source image to determine pixel value in destination image
  - Problem: some of the pixel coordinates in the target image map to pixel coordinates outside of the source image
    - You'll solve this problem in a homework assignment by determining where the pixel coordinates of the corners of the source image map to pixel coordinates in the destination image using

$$\mathbf{x}_{\text{virtual}} = \mathtt{H}\mathbf{x}_{\text{real}}$$

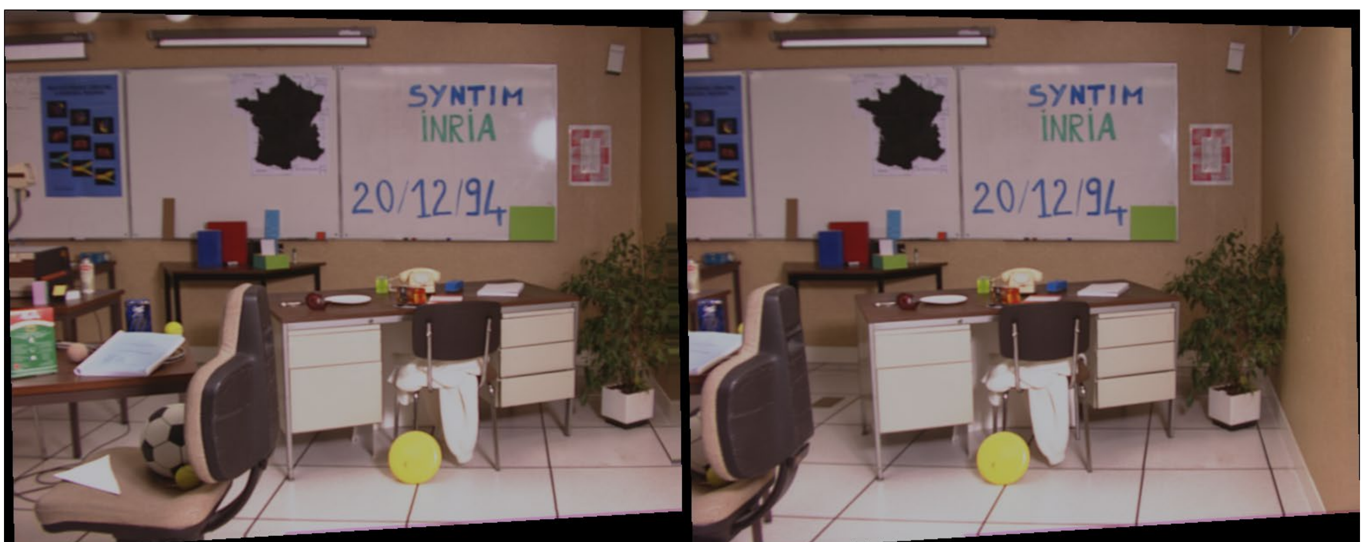




Rectification

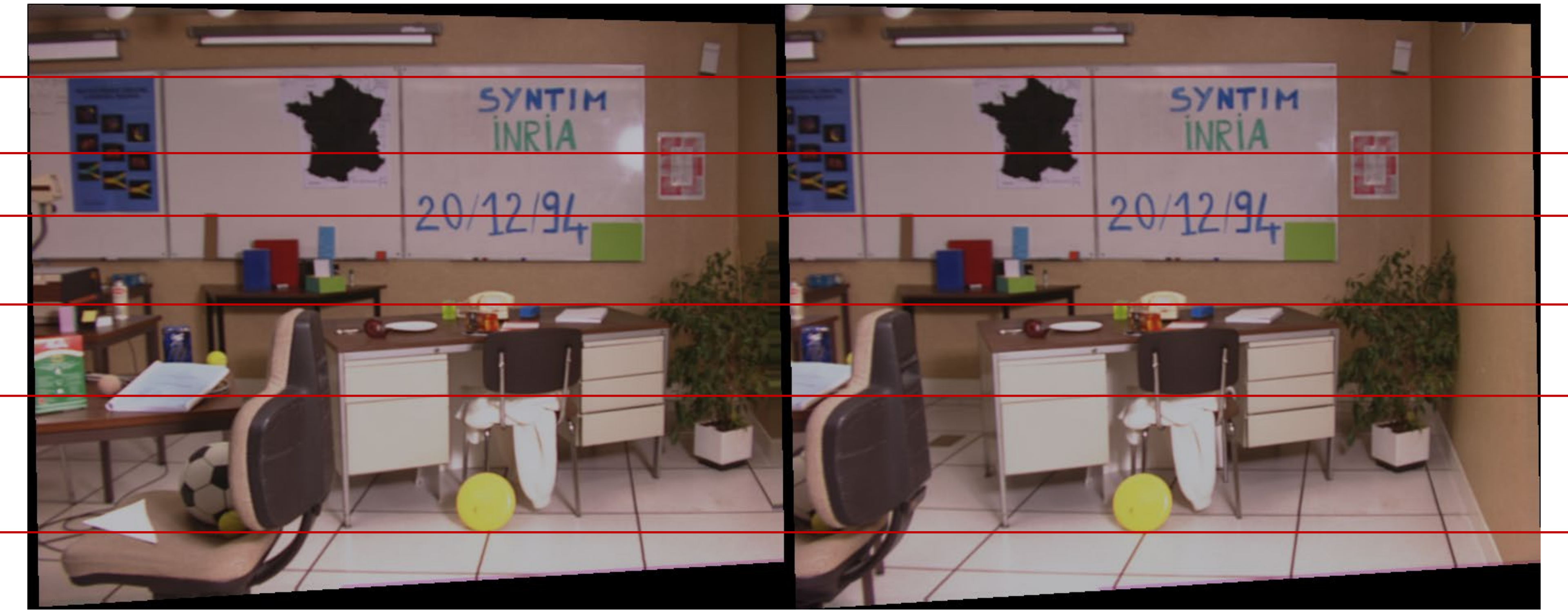
Original

Rectified



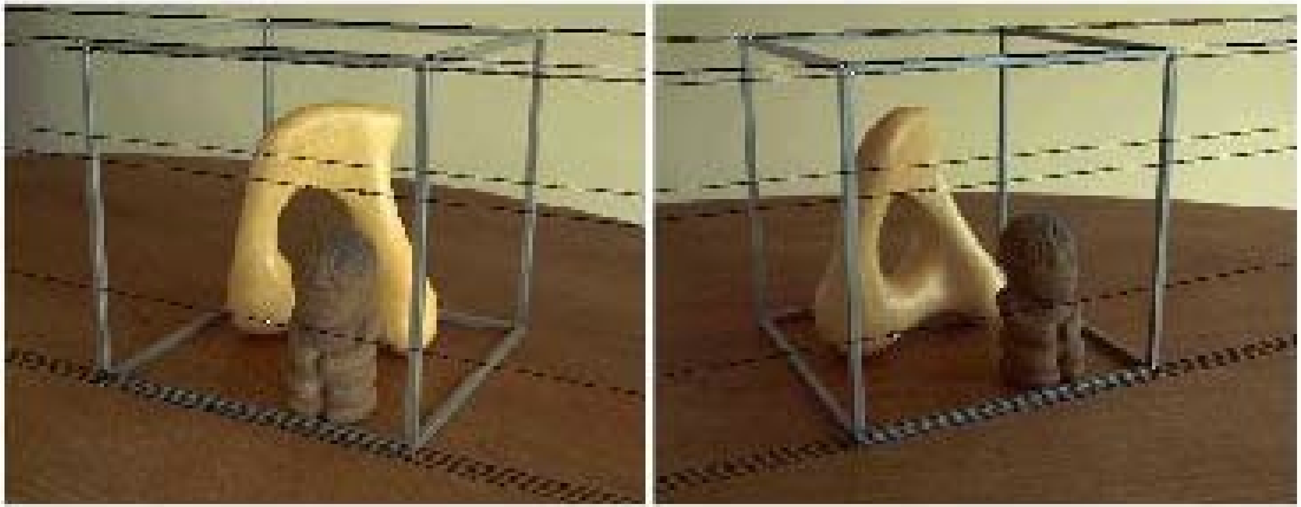
Rectification

- Epipolar lines

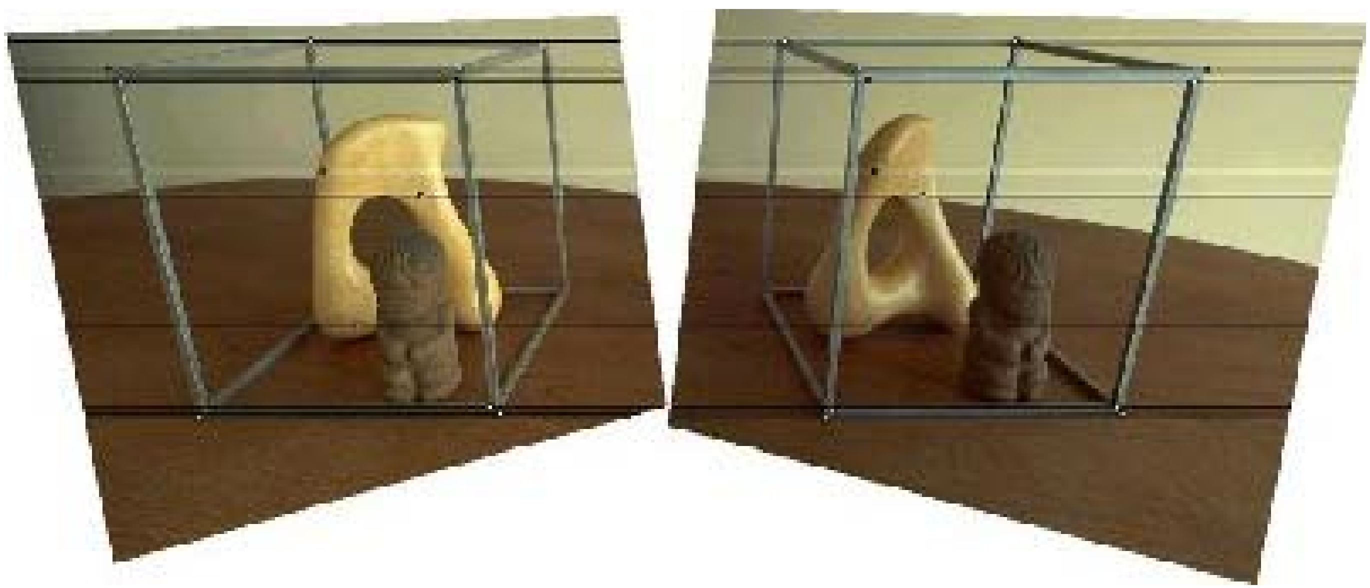


Rectification

Original



Rectified

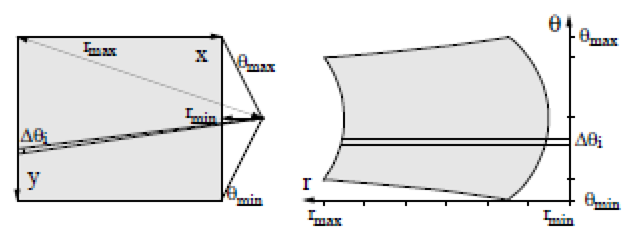


Polar Rectification

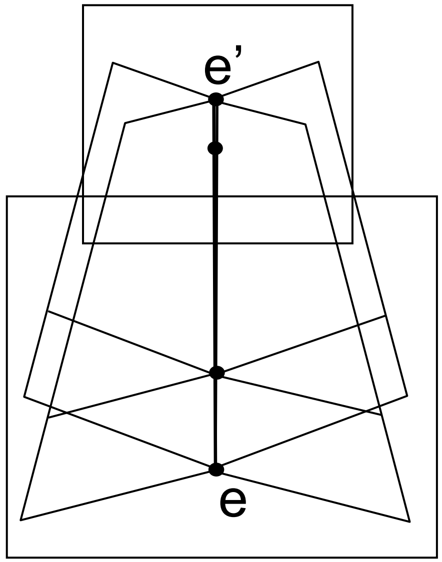
Homography-based Rectification

Polar Rectification

Alternative epipolar rectification method that minimizes pixel distortion

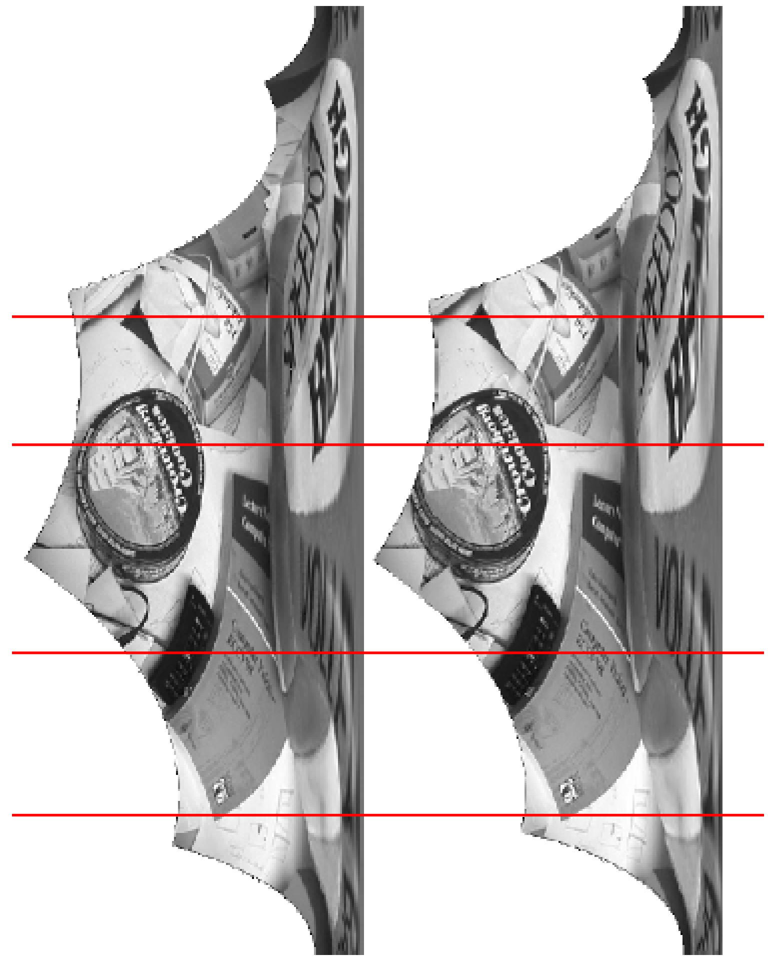


Polar Rectification



Epipoles are in images
(white dot on ball)

Homography-based rectification
is not possible



Features on same epipolar line

Next Lecture

- Calibrated stereo and feature matching