Computer Vision

Bogdan Alexe

bogdan.alexe@fmi.unibuc.ro

University of Bucharest, 2nd semester, 2023-2024

Project 1 – good to know

- one task to solve (several scenarios) = 4.5p (tasks) + 0.5p (oral presentation) + 0.5p (ex-officio = format) = 5.5p maximum
- deadline: in ~ 4 weeks time, Tuesday, 14th of May 2024, 23:59
  - **hard deadline – no late submission policy**
  - upload your source code
  - on Wednesday, 15th of May 2024 we will release the test set. You will run your (untouched) code on test images and send us your results. Immediately we will published the intermediate grades (based on your results). Final grades for Project 1 will be published after carefully checking your code, your pdf report and listening to your oral presentation (~2 weeks time).

- do not share/copy the solution with/from your colleagues: you + your colleague/s will get 0 points

Project 1 – grading

This project worths 5.5 points. We will grade your project based on the performance achieved by your algorithms on each of 200 test images.

You will receive a test set containing 200 testing images organized in 4 games of 50 moves. The distribution of images in the test data follows the distribution of train data, meaning that the images were acquired in the same conditions and also that the first game will contain images with regular views, the second game will contain images with rotated views, the third game will contain images with perspective view, the fourth game will contain images with mixed views. For each test image you have to output a .txt file containing information similar to the annotation files and for every game a .txt file with scores after each round. Each correctly solved test image worths 0.0175 points. For correctly specifying the position of the added token you receive 0.01 points per image (move), for correctly specifying the token placed at the corresponding position on the board you receive 0.0075 points per image (move) and for correctly specifying the total score of the current player after a turn you receive 0.02 point per turn. You receive 0.5 points from *ex officio* conditioned on the fact that you respect the format of the submitted results, such that our evaluation script works smoothly on your provided results.

The oral presentation of this project (face-to-face or online) will be scheduled in the weeks 14 – 17 or 20 – 24 of May. It will take around 10-15 minutes in which Alexandra or Bogdan will ask question regarding implementation. The oral presentation will count for 0.5 points and is mandatory for each student submitting his solution for this project.

Project 1 – deadlines

Deadlines

Submit a *zip archive* containing your code (python files or Jupyter notebook files), all auxiliary data that you are using (templates, models, etc.) and a pdf file describing your approach until Tuesday, 14^{th} of May using the following link `https://tinyurl.com/CV-2024-PROJECT1-SUBMISSIONS`. Please do not include in your zip archive any unuseful data (like training images, we already have them!!!). Notice that this is a hard deadline, no projects will be accepted after the deadline. Your code should include a README file (see the example in the materials for this project) containing the following information: (i) the libraries required to run the project including the full version of each library; (ii) indications of how to run the solution and where to look for the output file. Students who do not describe their approach (using a pdf file) will incur a penalty of 0.5 points.

On Wednesday 15^{th} of May we will make available the test data. You will have to run your solution on the test images provided by us and upload your results in the same day as a zip archive using the following link `https://tinyurl.com/CV-2024-PROJECT1-RESULTS`.

Project 1 – presentations

- today, at midnight, we will publish on Teams the list of students who have submitted their solution

- based on the number of submission we will offer some intervals this week (Friday) and next week

- use something like Google Forms to gather options tomorrow (Wednesday)

- publish this week (Wednesday or Thursday) a file with proposed schedule of presentations (10 minutes/each – online or f2f)

Course structure

1. **Features and filters: low-level vision**
   Linear filters, color, texture, edge detection, template matching

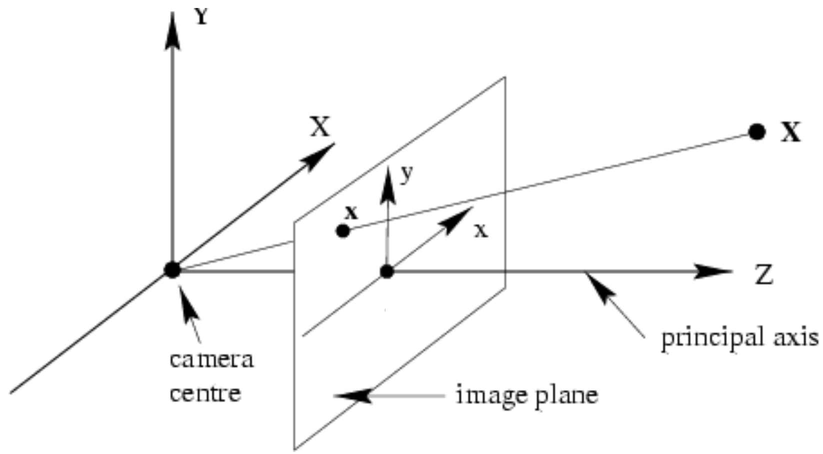
2. **Grouping and fitting: mid-level vision**
   Fitting curves and lines, robust fitting, RANSAC, Hough transform, segmentation

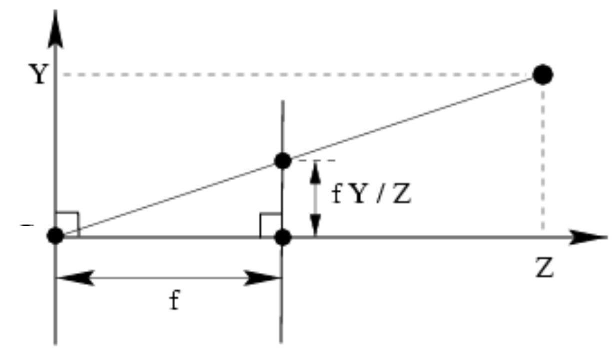
3. **Multiple views**
   Local invariant feature and description, epipolar geometry and stereo object instance recognition

4. **Object Recognition: high – level vision**
   Object classification, object detection, part based models, bovw models

5. **Video understanding**
   Object tracking, background subtraction, motion descriptors, optical flow

Pinhole camera model





$$(X, Y, Z) \mapsto (f X / Z, f Y / Z)$$

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \begin{bmatrix} f & & & 0 \\ & f & & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

$$\mathbf{x} = \mathrm{P}\mathbf{X}$$

x = homogenous coordinates in the image plane

X = homogenous coordinates in the 3D scene

P = 3x4 homogeneous camera projection matrix

Camera coordinate frame

$$\mathbf{x} = \mathrm{PX} = \mathrm{K}[\mathrm{I}|0]\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = K[I\,|\,0]X_{cam}$$

K - calibration matrix

$X_{cam} = (X, Y, Z, 1)^T$ = camera is assumed to be located at the origin of a Euclidean coordinate system with the principal axis of the camera pointing straight down the Z-axis, and the point X_{cam} is expressed in this coordinate system. Such a coordinate system may be called the ***camera coordinate frame.***

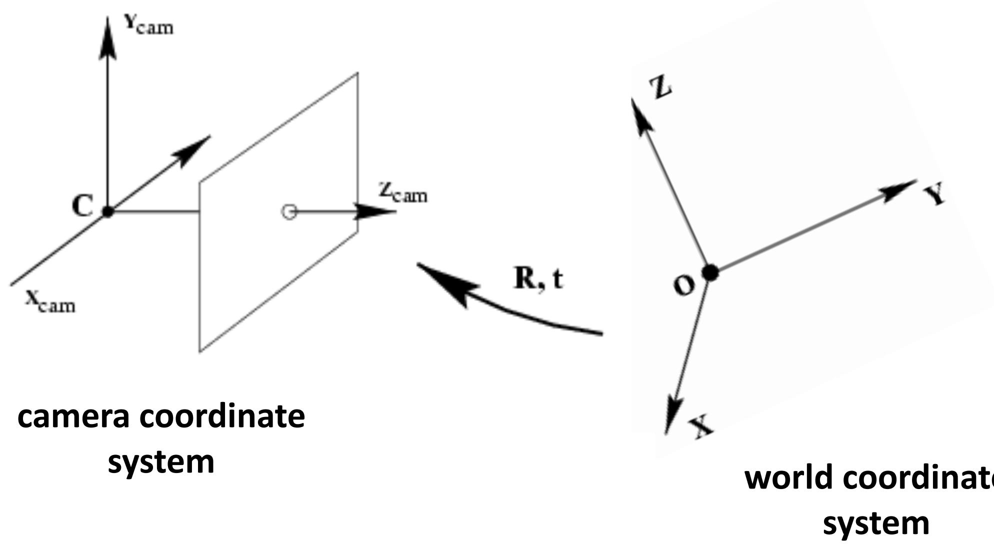
In general, points in space will be expressed in terms of a different Euclidean coordinate frame, known as the ***world coordinate frame***. The two coordinate frames are related via a rotation and a translation.

World coordinate frame

$$\mathbf{x} = \mathrm{P}\mathrm{X} = K[I\,|\,0]X_{cam}$$

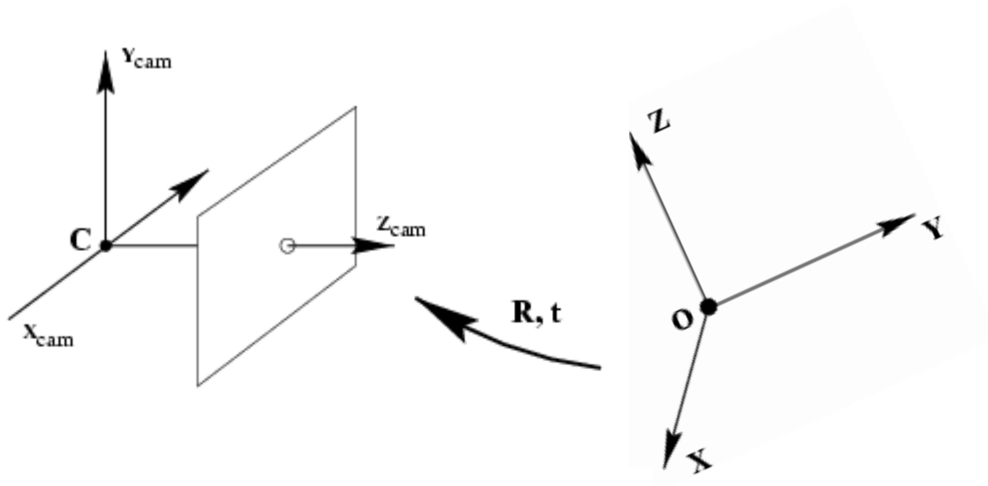
$X_{cam} = (X, Y, Z, 1)^T$ = expressed in the ***camera coordinate frame.***

In general, points in space will be expressed in terms of a different Euclidean coordinate frame, known as the ***world coordinate frame***. The two coordinate frames are related via a rotation **R** and a translation **t**.



Camera calibration: figuring out transformation from *world* coordinate system to *image* coordinate system

Camera rotation and translation



$$\mathrm{x} = \mathrm{PX} = K[I \mid 0]X_{cam}$$

$$\mathrm{x} = \mathrm{K}\left[\mathrm{I} \mid 0\right]\begin{bmatrix} \mathrm{R} & -\mathrm{R}\widetilde{\mathrm{C}} \\ 0 & 1 \end{bmatrix}\mathrm{X}$$

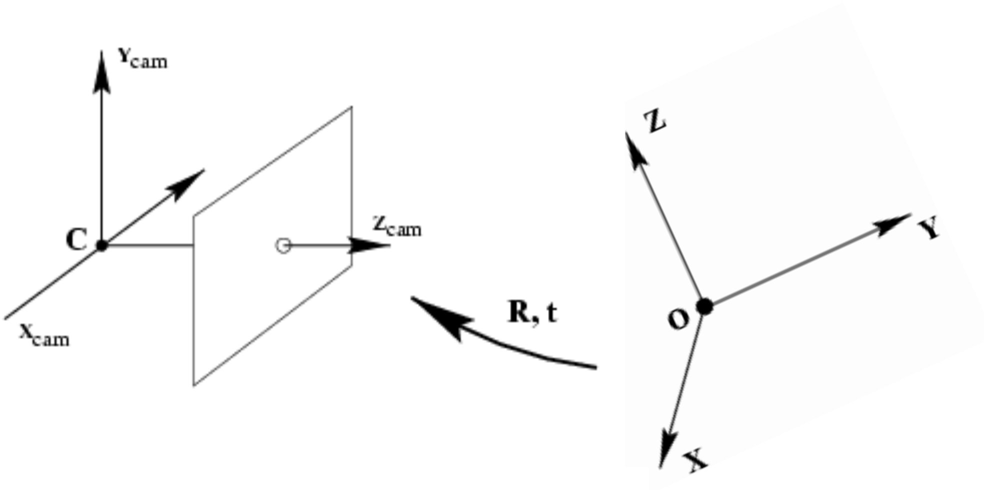
2D transformation matrix (3 x 3)

perspective projection matrix (3 x 4)

3D transformation matrix (4 x 4)

calibration matrix K

Camera rotation and translation



$$\mathrm{x} = \mathrm{K}\left[\mathrm{R} \mid -\mathrm{R}\tilde{\mathrm{C}}\right]\mathrm{X}$$

$$\mathrm{x} = \mathrm{PX}$$

$$\mathbf{P} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}$$

Camera parameters $\mathbf{P} = \mathbf{K}[\mathbf{R} \quad \mathbf{t}]$

- Intrinsic parameters
  - Principal point coordinates
  - Focal length
  - Pixel magnification factors
  - *Skew (non-rectangular pixels)*
  - *Radial distortion*
- Extrinsic parameters
  - Rotation and translation relative to world coordinate system

$$\mathbf{P} = \mathbf{K}[\mathbf{R} \quad -\mathbf{R}\underbrace{\tilde{\mathbf{C}}}_{\text{coords. of camera center in world frame}}]$$

Camera parameters $\mathbf{P} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}$

$$\mathbf{x} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}\mathbf{X}$$

$$\begin{bmatrix} \alpha_x & s & \beta_x \\ 0 & \alpha_y & \beta_y \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix}$$

Degrees of freedom

$$\mathbf{x} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}\mathbf{X}$$

6 (3 for rotation and 3 for translation)

$$\overbrace{\begin{bmatrix} \alpha_x & s & \beta_x \\ 0 & \alpha_y & \beta_y \\ 0 & 0 & 1 \end{bmatrix}}^{5}\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix}$$

5 + 6 = 11 degrees of freedom = 12 (matrix 3 x 4) – 1 (scale)

How to calibrate the camera?

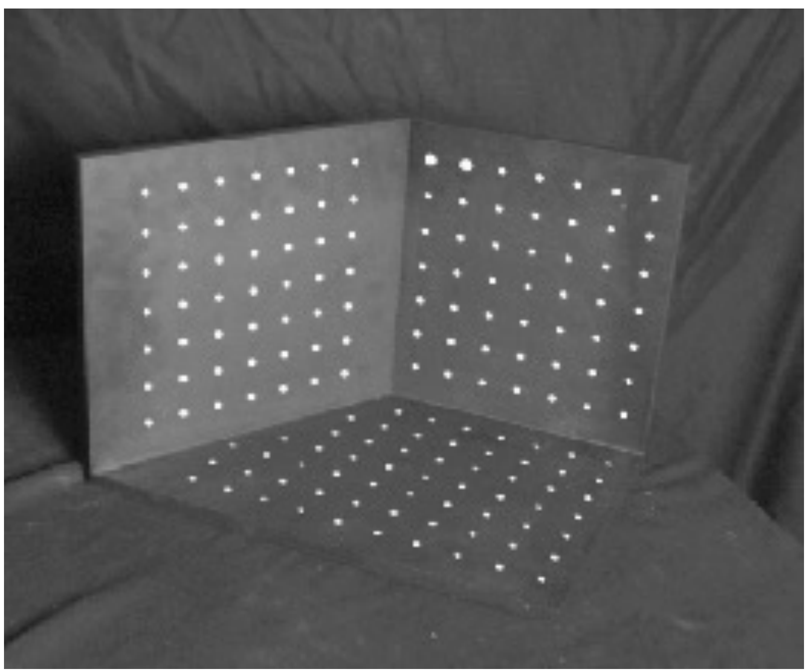
$$\mathbf{x} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}\mathbf{X}$$

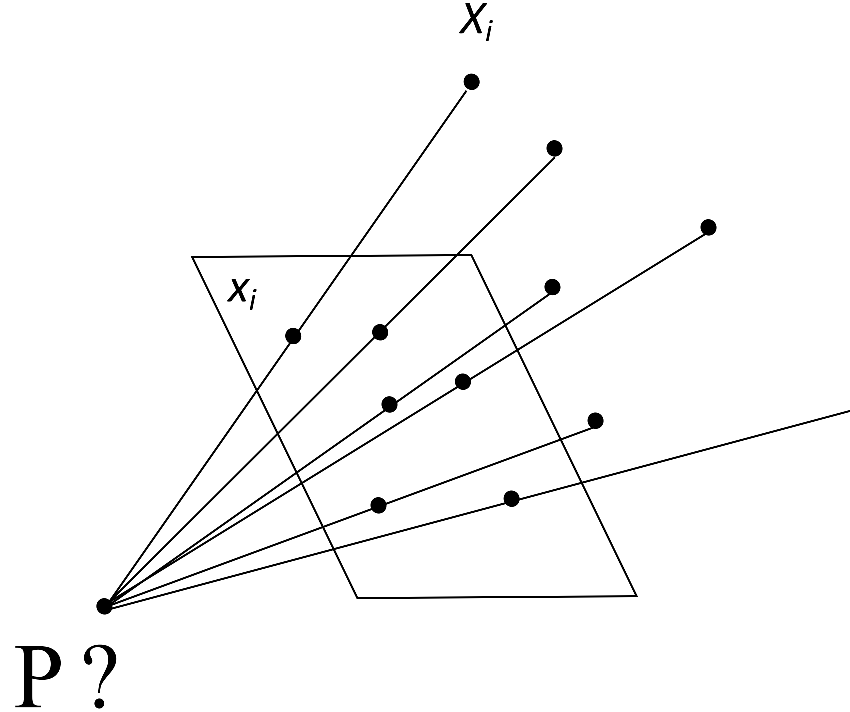
$$\begin{bmatrix} wu \\ wv \\ w \end{bmatrix} = \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Calibrating the Camera

Method 1: Use an object (calibration grid) with known geometry

- Given n points with known 3D coordinates $\boldsymbol{X}_i$ and known image projections $\boldsymbol{x}_i$, estimate the camera parameters



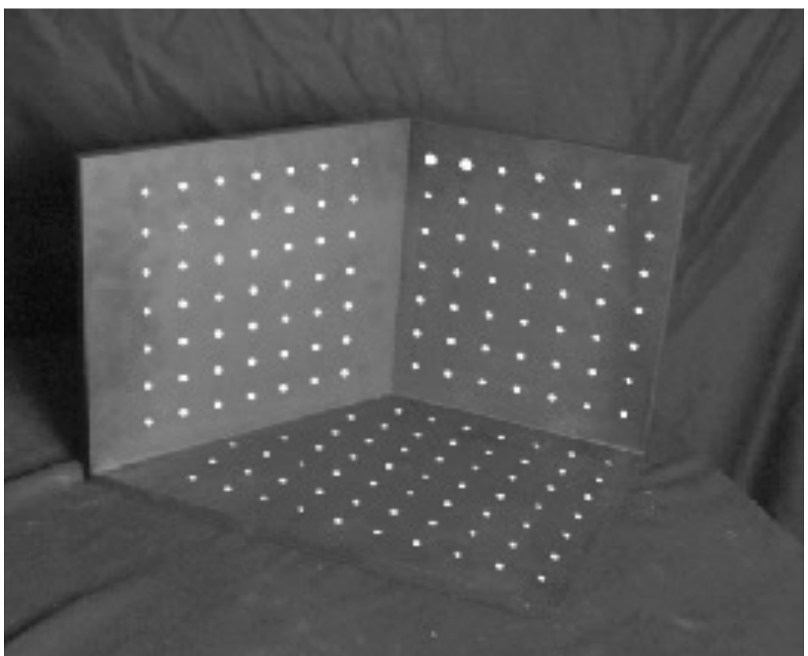


Calibrating the Camera

Method 1: Use an object (calibration grid) with known geometry

– Given n points with known 3D coordinates $\boldsymbol{X}_i$ and known image projections $\boldsymbol{x}_i$, estimate the camera parameters

– Get least squares solution (or non-linear solution)



$$\begin{bmatrix} wu \\ wv \\ w \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Linear method

- Solve using linear least squares

$$\begin{bmatrix} wu \\ wv \\ w \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

- P has 11 degrees of freedom
- One 2D/3D correspondence gives us two linearly independent equations
  - 6 correspondences needed for a minimal solution

Linear method

- Solve using linear least squares

$$\begin{bmatrix} wu \\ wv \\ w \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

(X_i, Y_i, Z_i) – 3D world coordinates

(u_i, v_i) – 2D image coordinates

$$\begin{bmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -u_1X_1 & -u_1Y_1 & -u_1Z_1 & -u_1 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -v_1X_1 & -v_1Y_1 & -v_1Z_1 & -v_1 \\ & & & & & & \vdots & & & & & \\ X_n & Y_n & Z_n & 1 & 0 & 0 & 0 & 0 & -u_nX_n & -u_nY_n & -u_nZ_n & -u_n \\ 0 & 0 & 0 & 0 & X_n & Y_n & Z_n & 1 & -v_nX_n & -v_nY_n & -v_nZ_n & -v_n \end{bmatrix} \begin{bmatrix} m_{11} \\ m_{12} \\ m_{13} \\ m_{14} \\ m_{21} \\ m_{22} \\ m_{23} \\ m_{24} \\ m_{31} \\ m_{32} \\ m_{33} \\ m_{34} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}$$

Ap=0 form

Linear method

- Solve using linear least squares

$$\begin{bmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -u_1X_1 & -u_1Y_1 & -u_1Z_1 & -u_1 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -v_1X_1 & -v_1Y_1 & -v_1Z_1 & -v_1 \\ & & & & & & \vdots & & & & & \\ X_n & Y_n & Z_n & 1 & 0 & 0 & 0 & 0 & -u_nX_n & -u_nY_n & -u_nZ_n & -u_n \\ 0 & 0 & 0 & 0 & X_n & Y_n & Z_n & 1 & -v_nX_n & -v_nY_n & -v_nZ_n & -v_n \end{bmatrix} \begin{bmatrix} m_{11} \\ m_{12} \\ m_{13} \\ m_{14} \\ m_{21} \\ m_{22} \\ m_{23} \\ m_{24} \\ m_{31} \\ m_{32} \\ m_{33} \\ m_{34} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}$$

Ap=0 form

- Homogeneous least squares: find p minimizing $||Ap||^2$
  - Solution given by the eigenvector of A^TA with smallest eigenvalue

Laboratory class 6

The goal of this lab is to introduce you to camera and scene geometry. Specifically we will estimate the camera projection matrix, which maps 3D world coordinates to image coordinates, as well as the fundamental matrix, which relates points in one scene to epipolar lines in another. The camera projection matrix and the fundamental matrix can each be estimated using point correspondences. To estimate the projection matrix (camera calibration), the input is corresponding 3D and 2D points. To estimate the fundamental matrix the input is corresponding 2D points across two images. We start by estimating the projection matrix and the fundamental matrix for a scene with ground truth correspondences. Then we'll move on to estimating the fundamental matrix using point correspondences from SIFT.

Tutorial on epipolar geometry is here: https://docs.opencv.org/master/da/de9/tutorial_py_epipolar_geometry.html or here: https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_calib3d/py_epipolar_geometry/py_epipolar_geometry.html

Data

We provide 2D and 3D ground truth point correspondences for the base image pair (pic_a.jpg and pic_b.jpg), as well as other images which will not have any ground truth dataset.

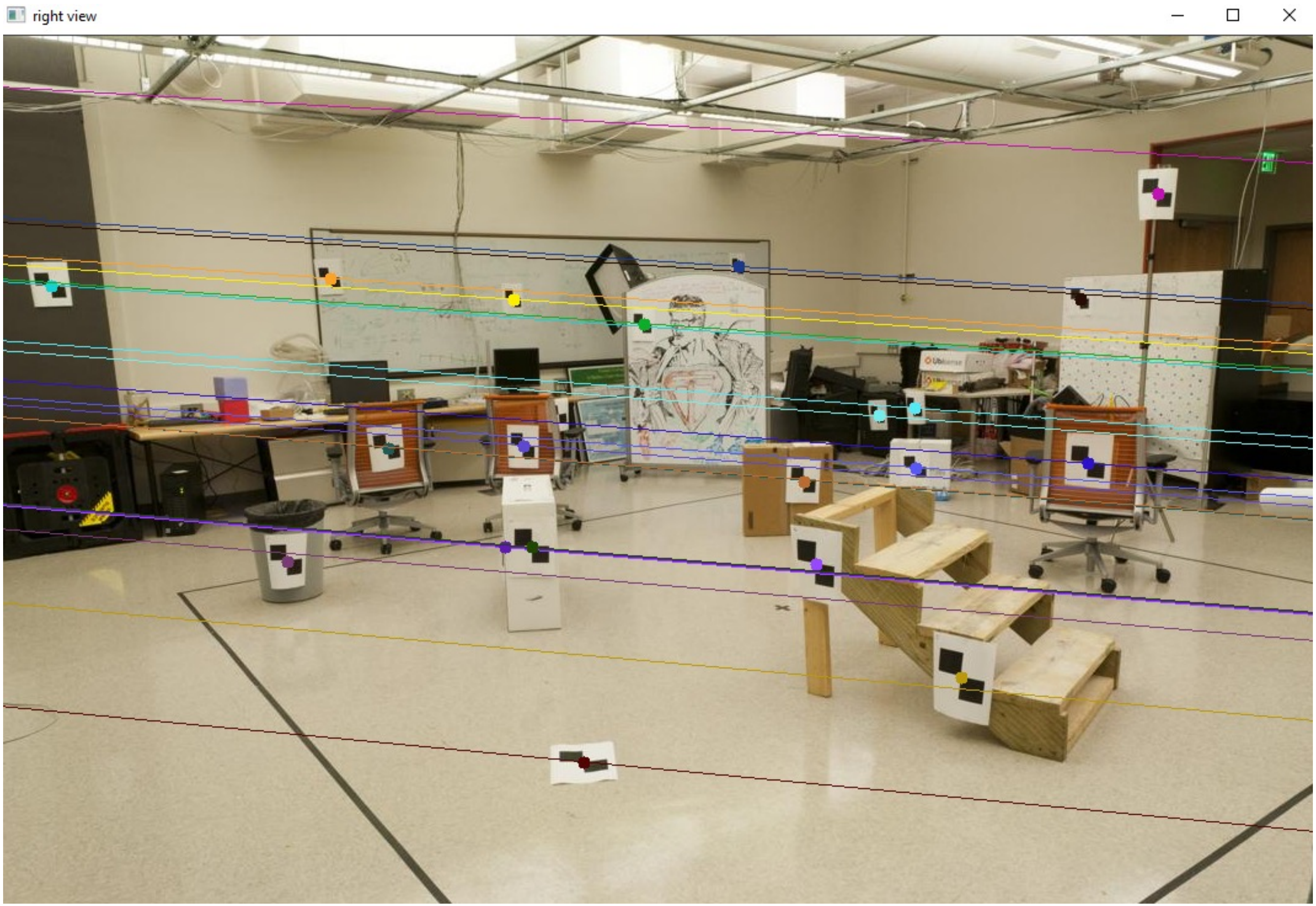
Image 1 of a scene

Image 2 of a scene

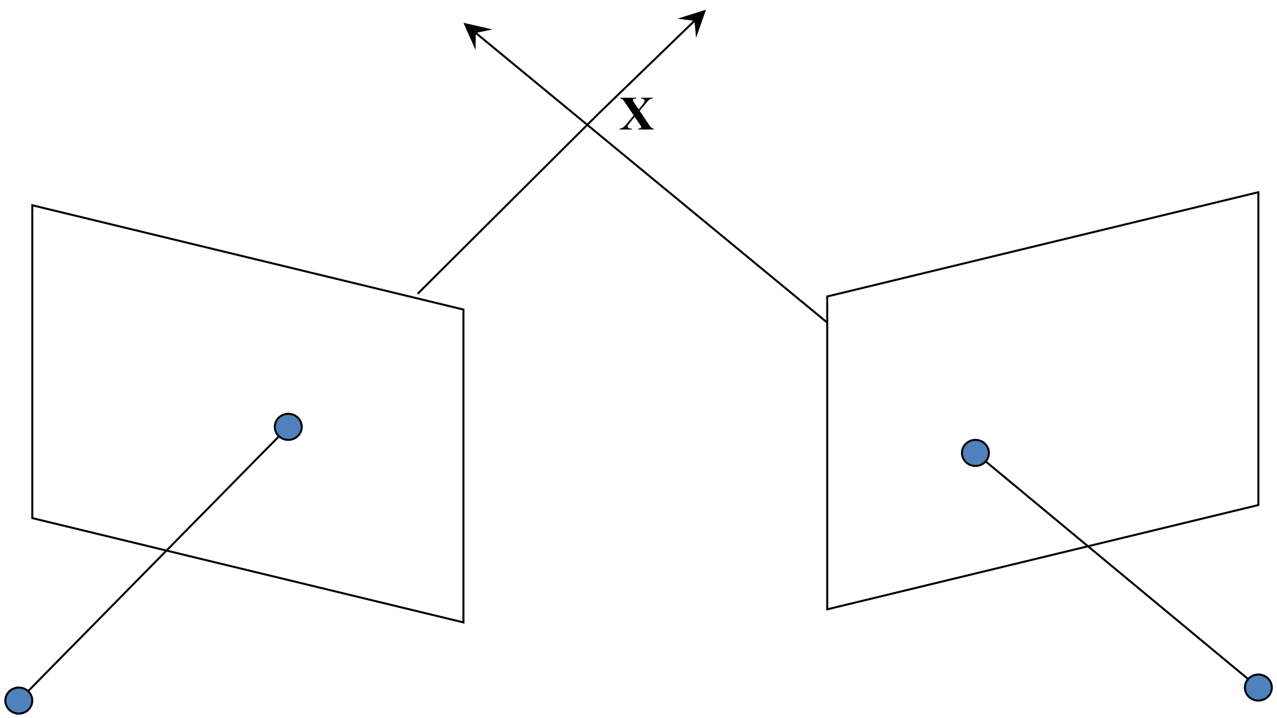
This laboratory consists of three parts: (1) estimating the projection matrix, (2) estimating the fundamental matrix, (3) estimating the fundamental matrix with unreliable SIFT matches using RANSAC.

Laboratory class 6

Laboratory class 6



Two–View Geometry

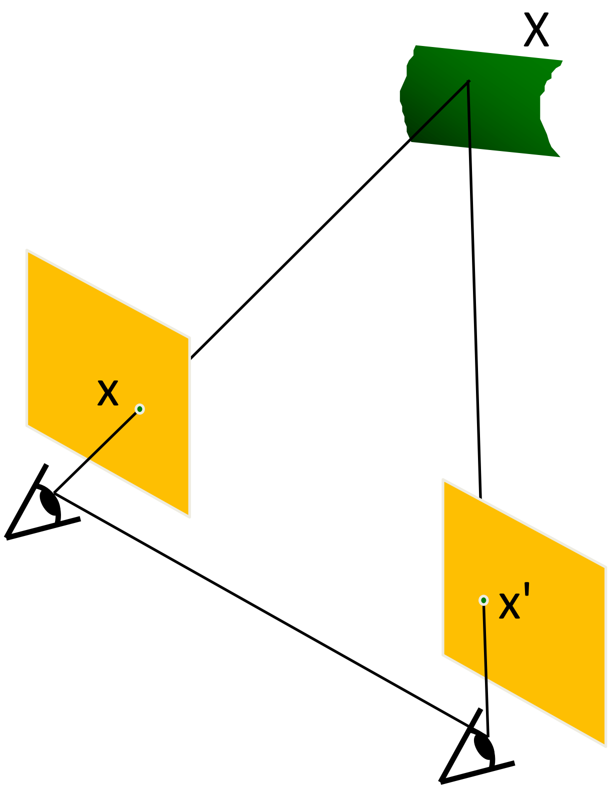


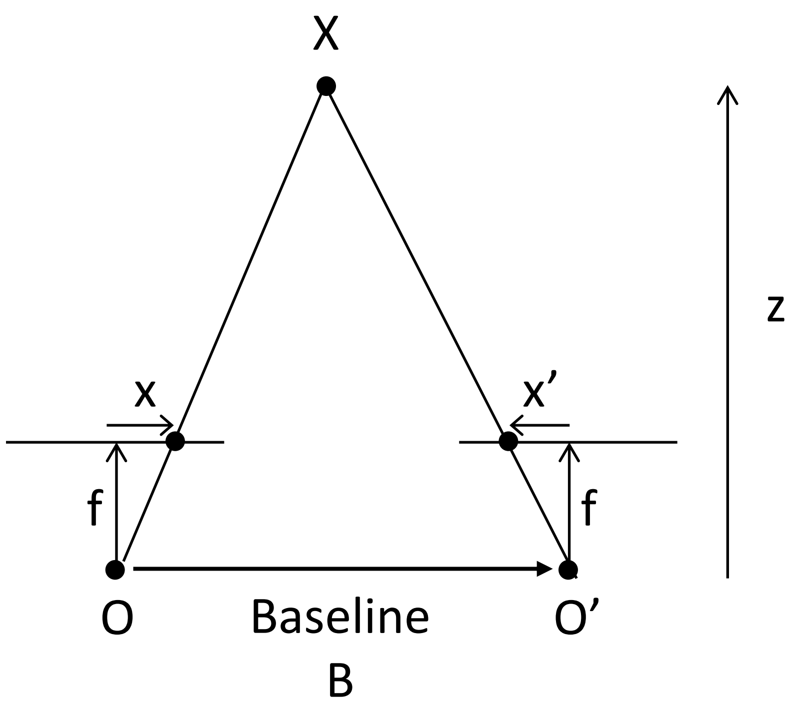
Two-View Geometry

- Epipolar geometry
  - Relates cameras from two positions
- Stereo depth estimation
  - Recover depth from two images

Depth from Stereo

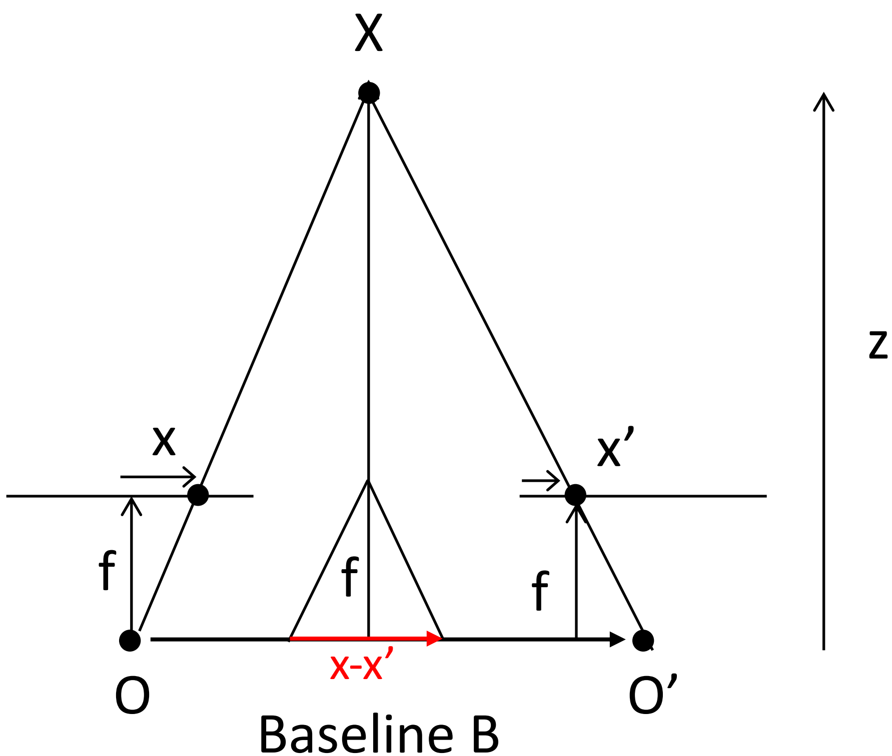
- Goal: recover depth by finding image coordinate x' that corresponds to x





Depth from Stereo

$$\frac{x - x'}{O - O'} = \frac{f}{z}$$

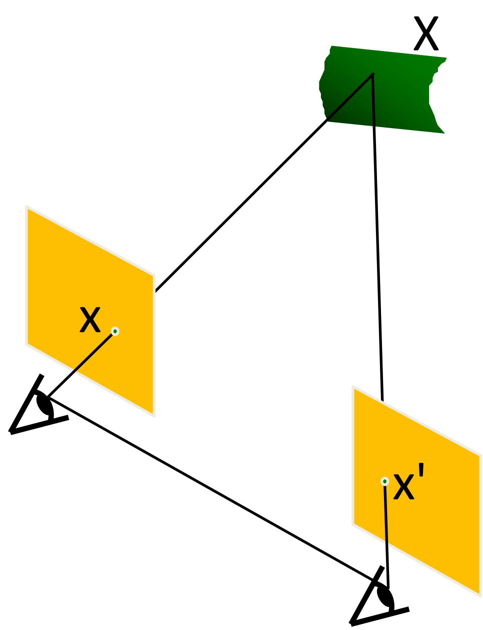


$$disparity = x - x' = \frac{B \cdot f}{z}$$

Disparity is inversely proportional to depth.

Depth from Stereo

- Goal: recover depth by finding image coordinate x' that corresponds to x
- Sub-Problems
  1. Calibration: How do we recover the relation of the cameras (if not already known)?
  2. Correspondence: How do we search for the matching point x'?

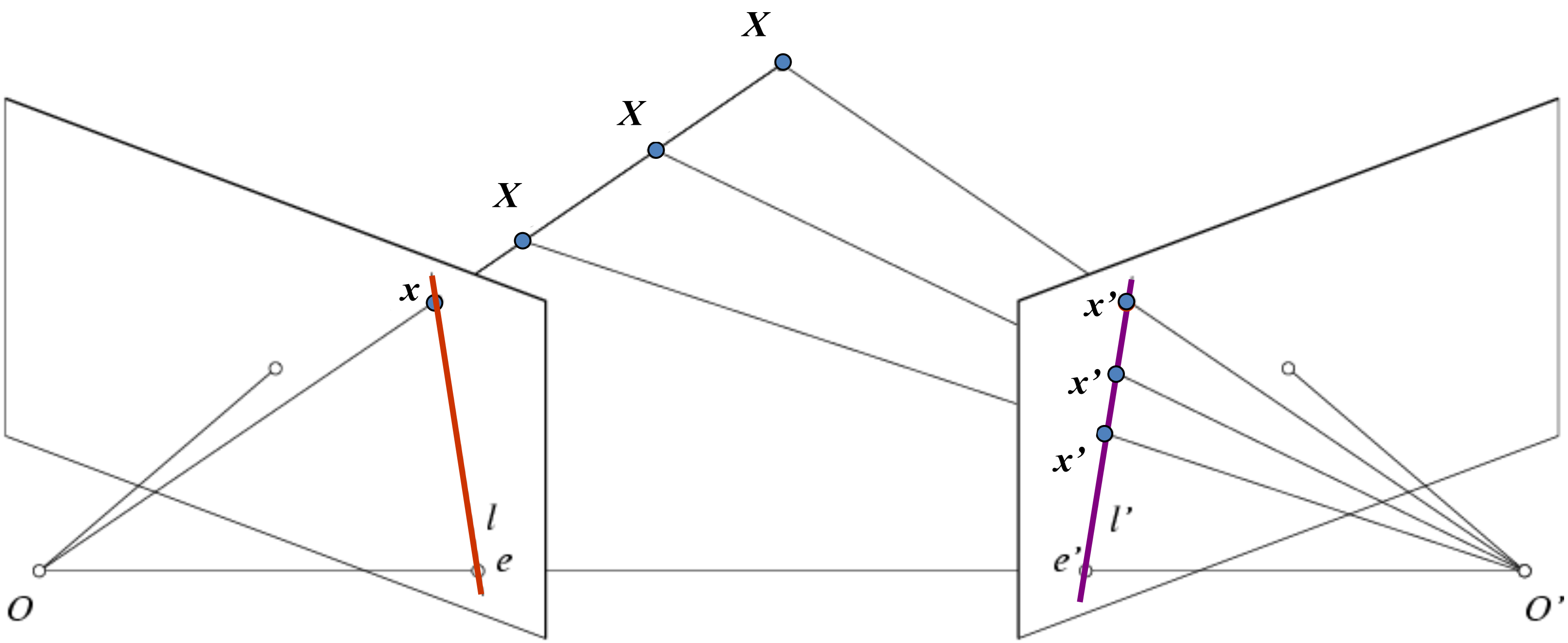


Correspondence Problem

- We have two images taken from cameras with different intrinsic and extrinsic parameters

- How do we match a point in the first image to a point in the second? How can we constrain our search?

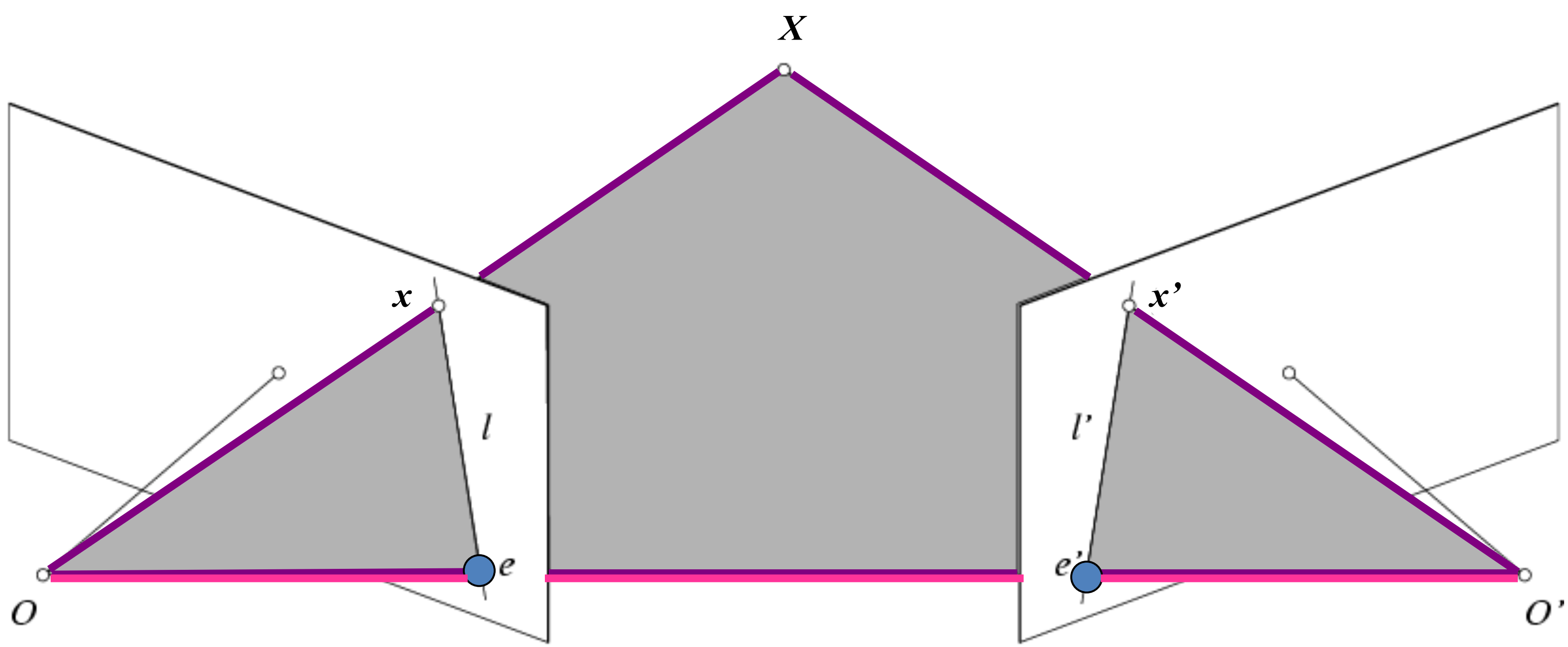
Key idea: Epipolar constraint



Potential matches for *x* have to lie on the corresponding line *l'*.

Potential matches for *x'* have to lie on the corresponding line *l*.

Epipolar geometry: notation

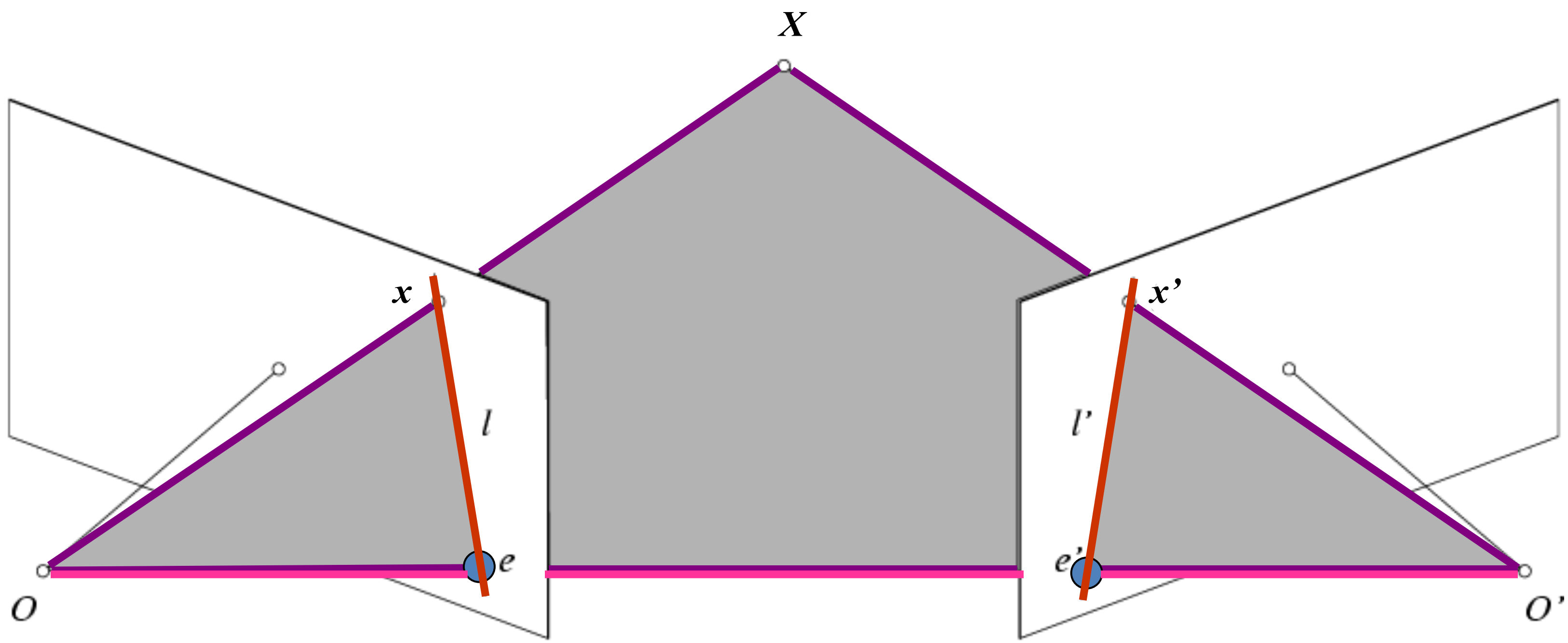


- **Baseline** – line connecting the two camera centers
- **Epipoles**

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

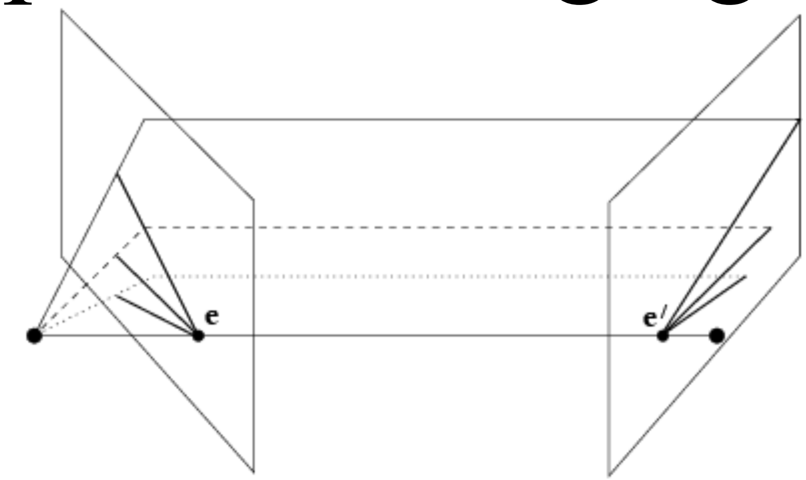
- **Epipolar Plane** – plane containing baseline (1D family)

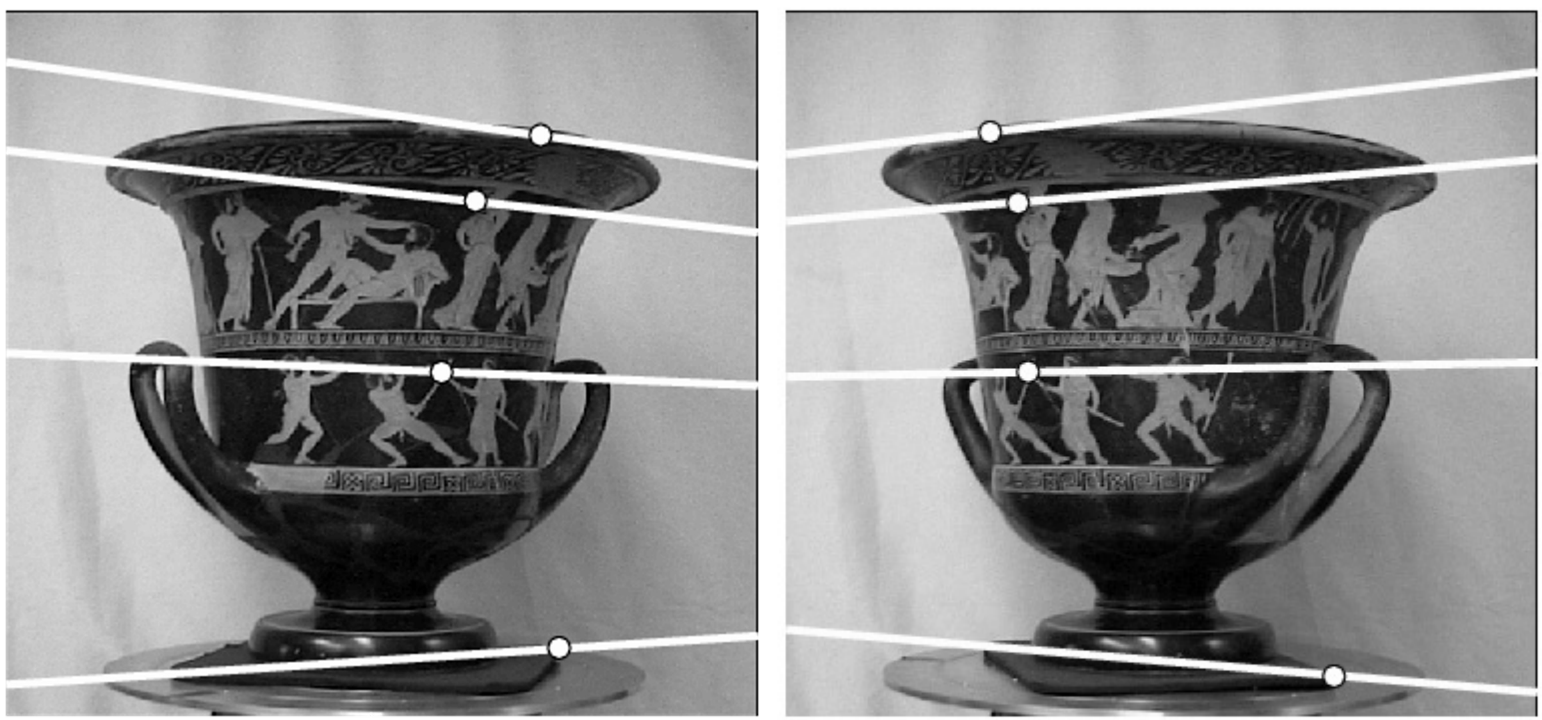
Epipolar geometry: notation



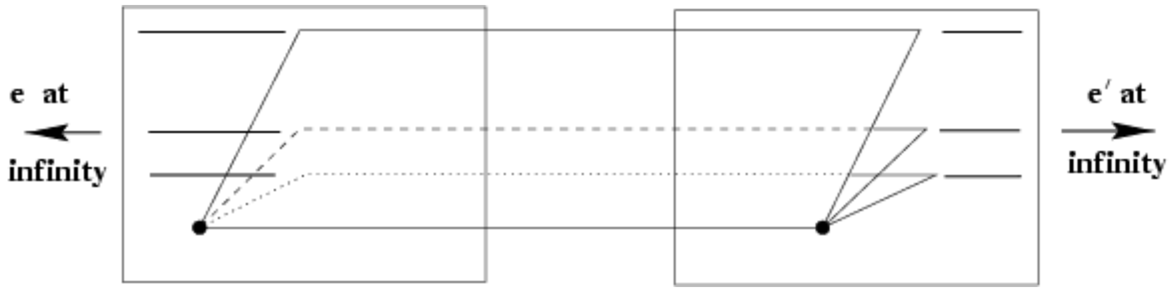
- **Baseline** – line connecting the two camera centers
- **Epipoles**
  = intersections of baseline with image planes
  = projections of the other camera center
- **Epipolar Plane** – plane containing baseline (1D family)
- **Epipolar Lines** - intersections of epipolar plane with image planes (always come in corresponding pairs)

Example: Converging cameras





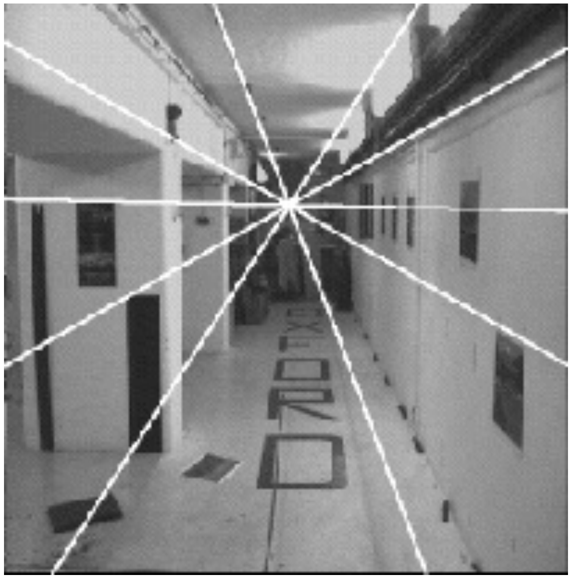
Example: Parallel cameras

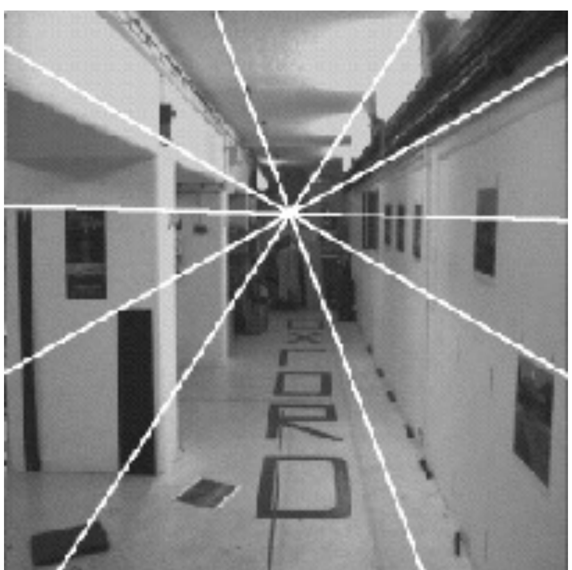


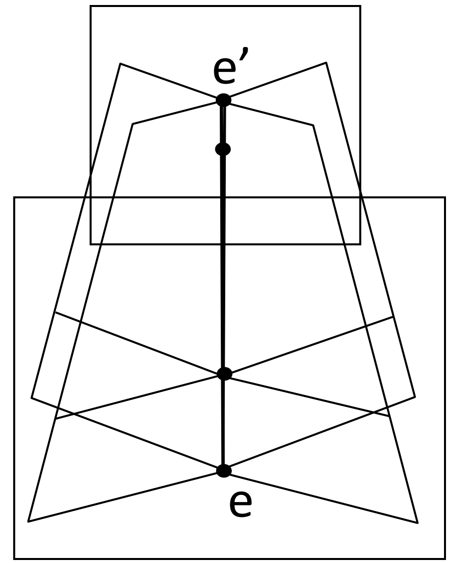
Example: Forward motion

What would the epipolar lines look like if the camera moves directly forward?

Example: Forward motion







Epipole has same coordinates in both images.
Points move along lines radiating from e: “Focus of expansion”

Camera parameters $\mathbf{P} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}$

- Intrinsic parameters
  - Principal point coordinates
  - Focal length
  - Pixel magnification factors
  - *Skew (non-rectangular pixels)*
  - *Radial distortion*

$$\mathbf{K} = \begin{bmatrix} m_x & & \\ & m_y & \\ & & 1 \end{bmatrix}\begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix} = \begin{bmatrix} \alpha_x & s & \beta_x \\ & \alpha_y & \beta_y \\ & & 1 \end{bmatrix}$$

- Extrinsic parameters
  - Rotation and translation relative to world coordinate system
  - What is the projection of the camera center?

$$\mathbf{P} = \mathbf{K}\begin{bmatrix}\mathbf{R} & -\mathbf{R}\widetilde{\mathbf{C}}\end{bmatrix}$$

coords. of camera center in world frame

$$\mathbf{PC} = \mathbf{K}\begin{bmatrix}\mathbf{R} & -\mathbf{R}\widetilde{\mathbf{C}}\end{bmatrix}\begin{bmatrix}\widetilde{\mathbf{C}} \\ 1\end{bmatrix} = 0$$

The camera center is the *null space* of the projection matrix!

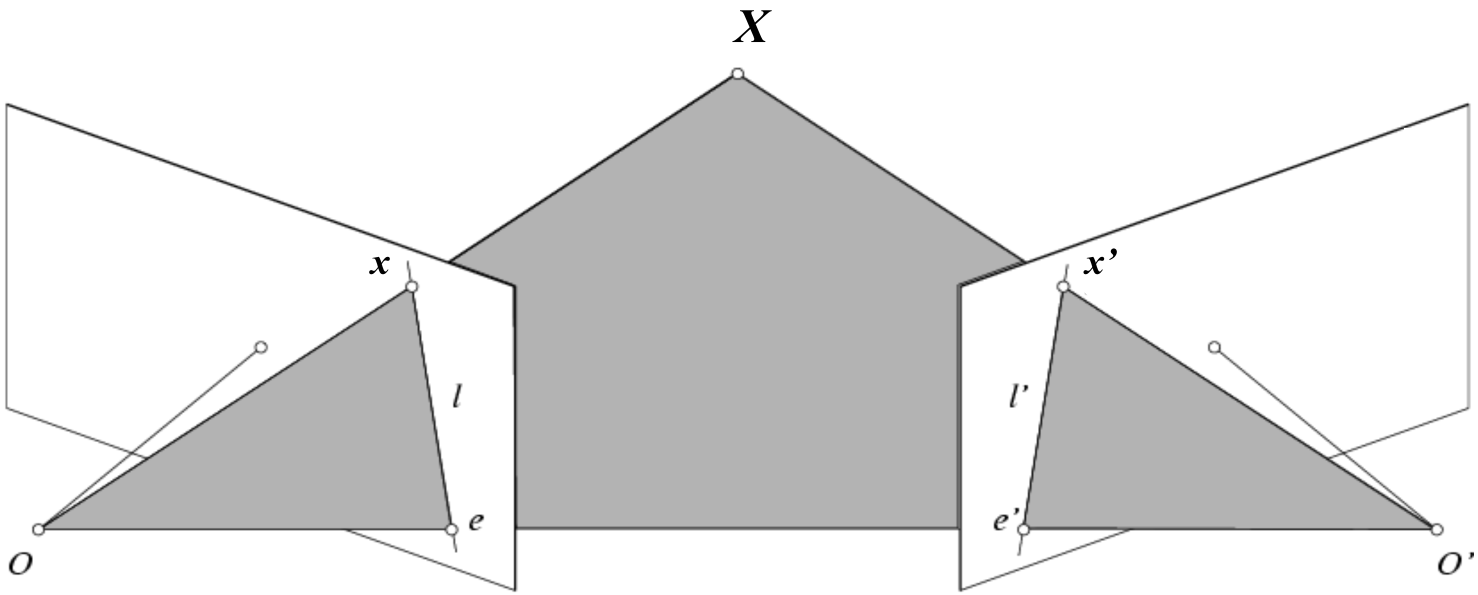
Camera parameters $\mathbf{P} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}$

$$\mathbf{x} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}\mathbf{X}$$

$$\downarrow$$

$$\begin{bmatrix} \alpha_x & s & \beta_x \\ 0 & \alpha_y & \beta_y \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix}$$

Epipolar constraint: Calibrated case



Given the intrinsic parameters of the cameras:

1. Convert to normalized coordinates by pre-multiplying all points with the inverse of the calibration matrix; set first camera's coordinate system to world coordinates

for some scale factor

$$\hat{x} = K^{-1}x = X$$

$$\hat{x}' = K'^{-1}x' = X'$$

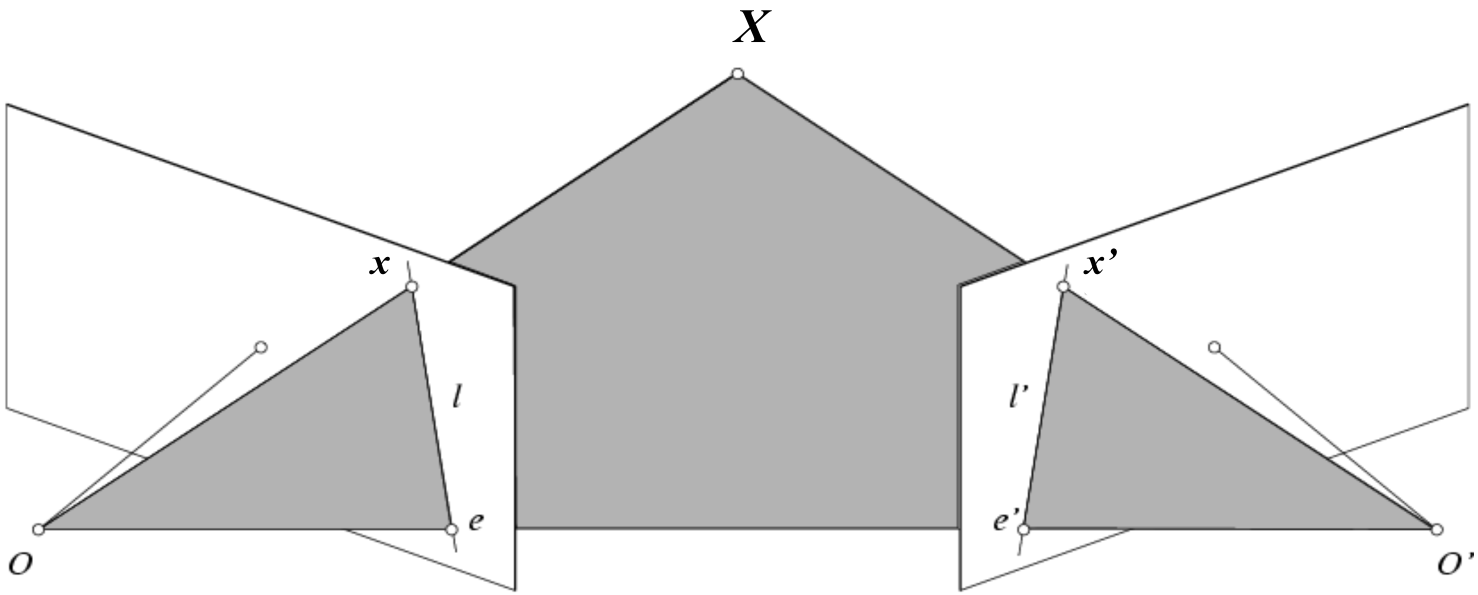
Homogeneous 2d point (3D ray towards X)

2D pixel coordinate (homogeneous)

3D scene point

3D scene point in 2nd camera's 3D coordinates

Epipolar constraint: Calibrated case



Given the intrinsic parameters of the cameras:

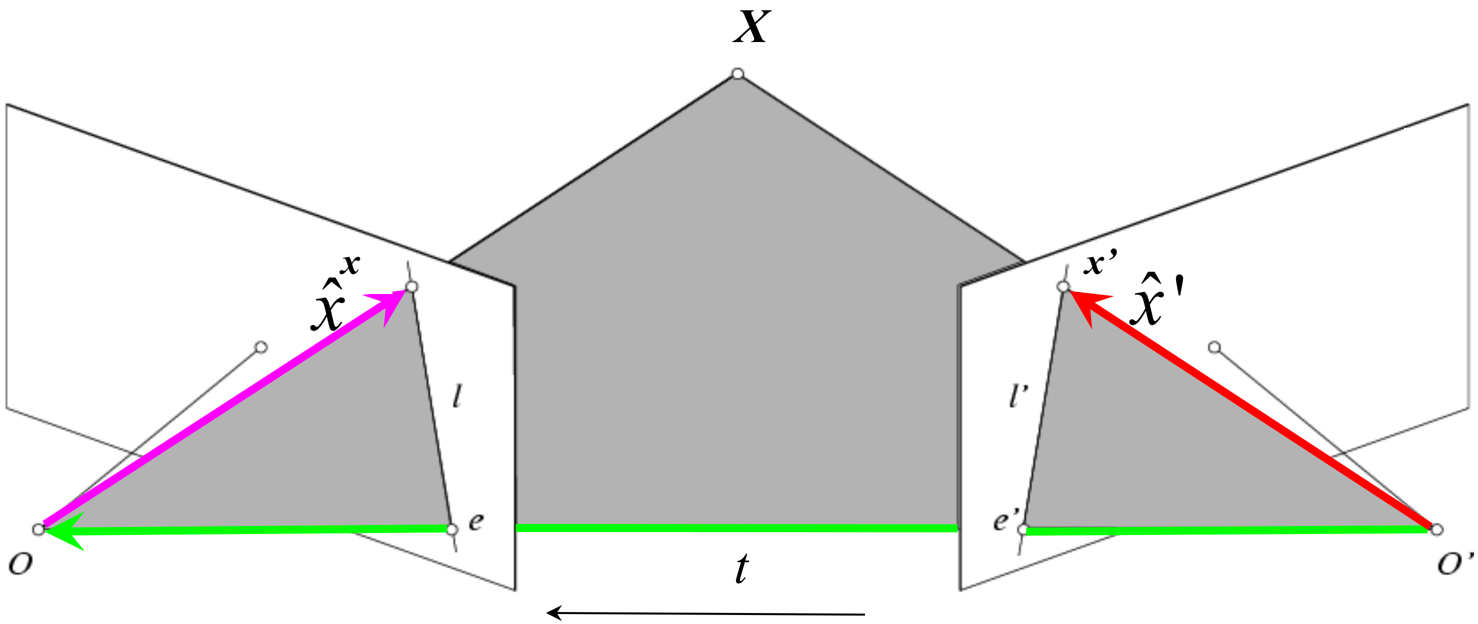
1. Convert to normalized coordinates by pre-multiplying all points with the inverse of the calibration matrix; set first camera's coordinate system to world coordinates
2. Define some *R* and *t* that relate X to X' as below

for some scale factor

$$\hat{x} = K^{-1} x = X \qquad \hat{x}' = K'^{-1} x' = X'$$

$$\hat{x} = R\hat{x}' + t$$

Epipolar constraint: Calibrated case

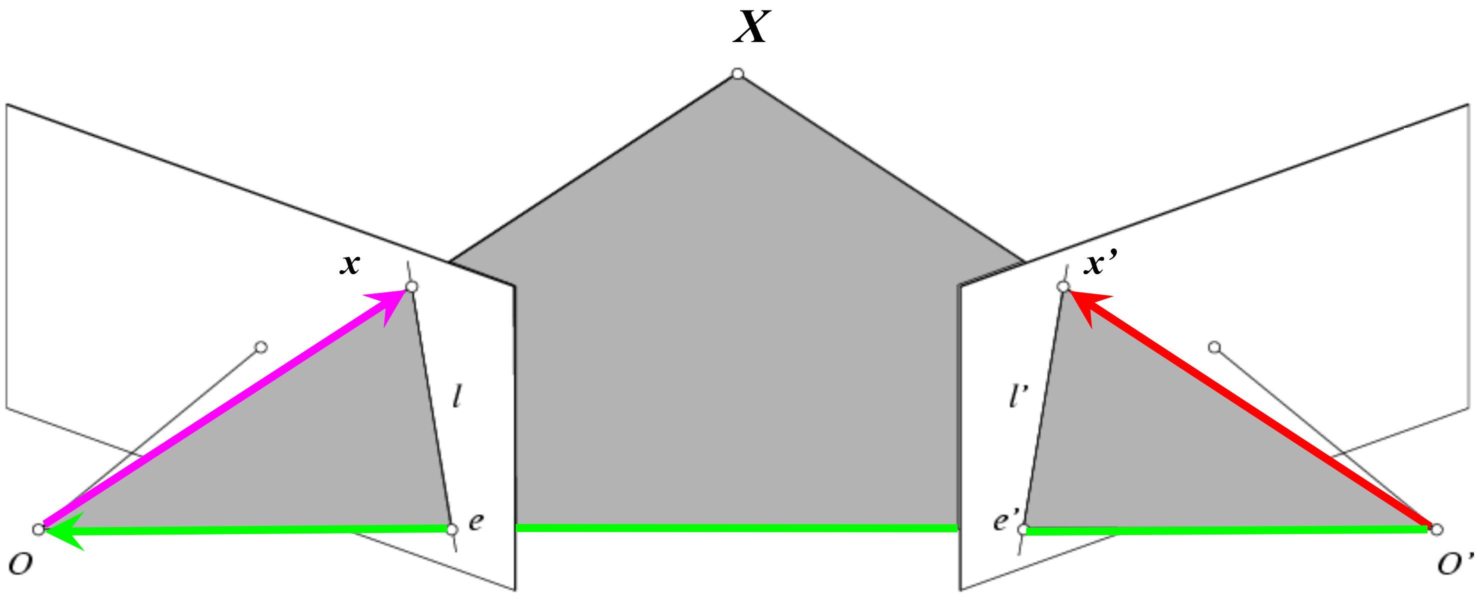


$$\hat{x} = K^{-1}x = X \qquad \hat{x}' = K'^{-1}x' = X'$$

$$\hat{x} = R\hat{x}' + t \quad \Rightarrow \quad \hat{x} \cdot [t \times (R\hat{x}')] = 0$$

(because x, $R\hat{x}'$ and t are co-planar)

Essential matrix



$$\hat{x} \cdot [t \times (R\hat{x}')] = 0$$

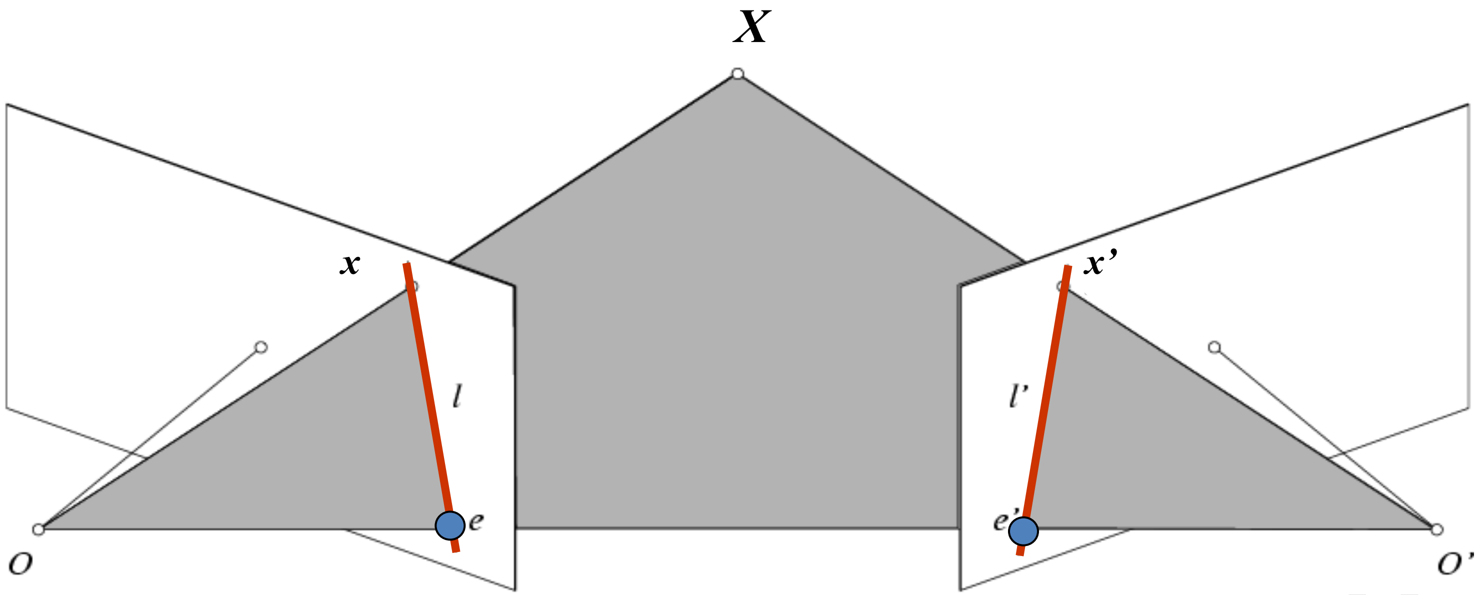
$$\hat{x}^T E \hat{x}' = 0 \quad \text{with} \quad E = [t]_\times R$$

Essential Matrix
(Longuet-Higgins, 1981)

$$[t]_\times = \begin{bmatrix} 0 & -t_3 & t_2 \\ t_3 & 0 & -t_1 \\ -t_2 & t_1 & 0 \end{bmatrix}$$

$[t]_\times$ is the skew symmetric matrix of $t=(t_1,t_2,t_3)$

Properties of the Essential matrix



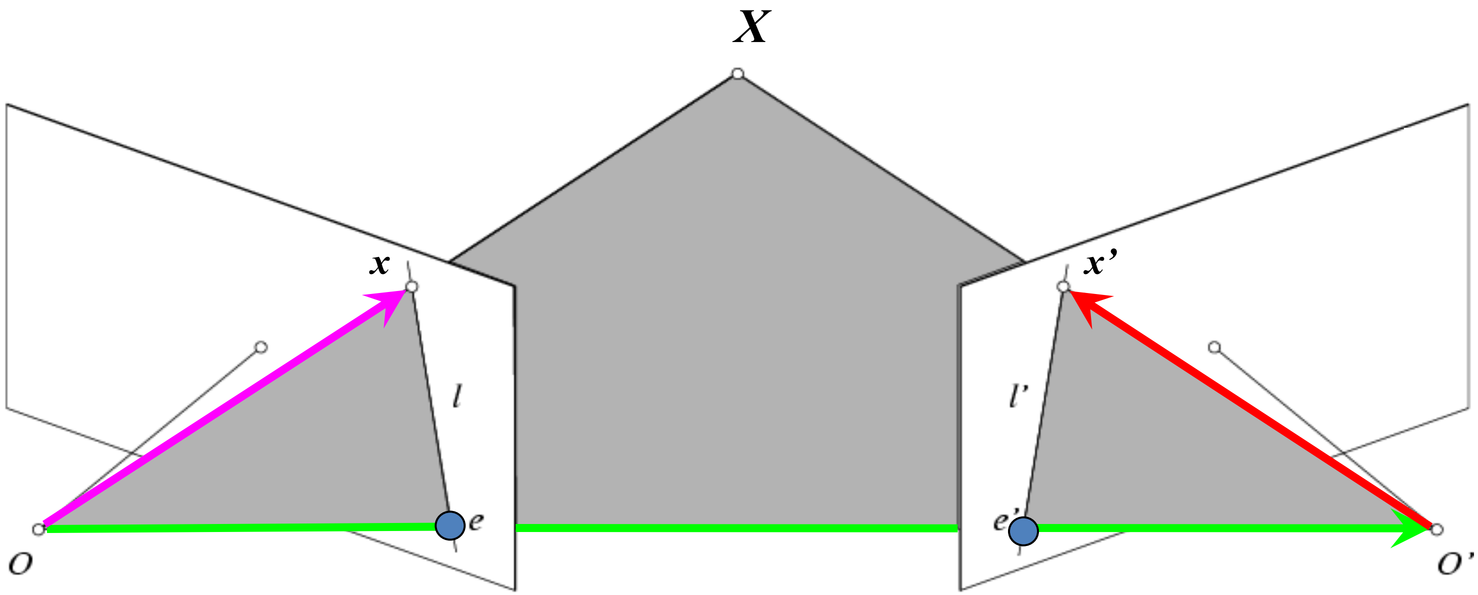
$$\hat{x}\cdot[t\times(R\hat{x}')]=0 \quad \Rightarrow \quad \hat{x}^T E\hat{x}'=0 \quad \text{with} \quad E=[t]_\times R$$

Drop ^ below to simplify notation

Skew-symmetric matrix

- $E\,x'$ is the epipolar line associated with x' ($l = E\,x'$)
- $E^T x$ is the epipolar line associated with x ($l' = E^T x$)
- $E\,e' = 0$ and $E^T e = 0$ *(e and e' are on each l and l')*
- E is singular (rank two – is determinant = 0)
- E has five degrees of freedom
  - (3 for R, 2 for t because it's up to a scale)

Epipolar constraint: Uncalibrated case



- If we don't know *K* and *K'*, then we can write the epipolar constraint in terms of *unknown* normalized coordinates:

$$\hat{x}^T E \hat{x}' = 0 \qquad\qquad x = K\hat{x}, \quad x' = K'\hat{x}'$$

The Fundamental Matrix

Without knowing K and K', we can define a similar relation using *unknown* normalized coordinates

$$\hat{x}^T E \hat{x}' = 0$$

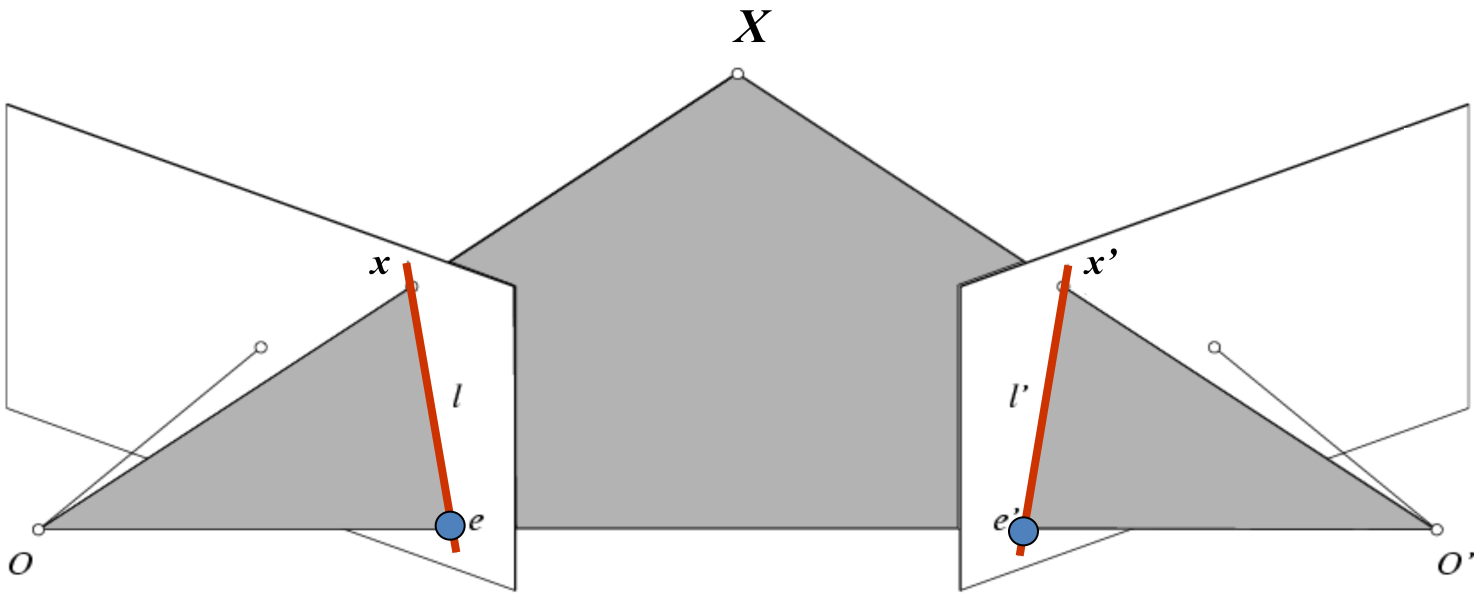
$$\hat{x} = K^{-1} x$$

$$\hat{x}' = K'^{-1} x'$$

$$x^T F x' = 0 \quad \text{with} \quad F = K^{-T} E K'^{-1}$$

Fundamental Matrix
(Faugeras and Luong, 1992)

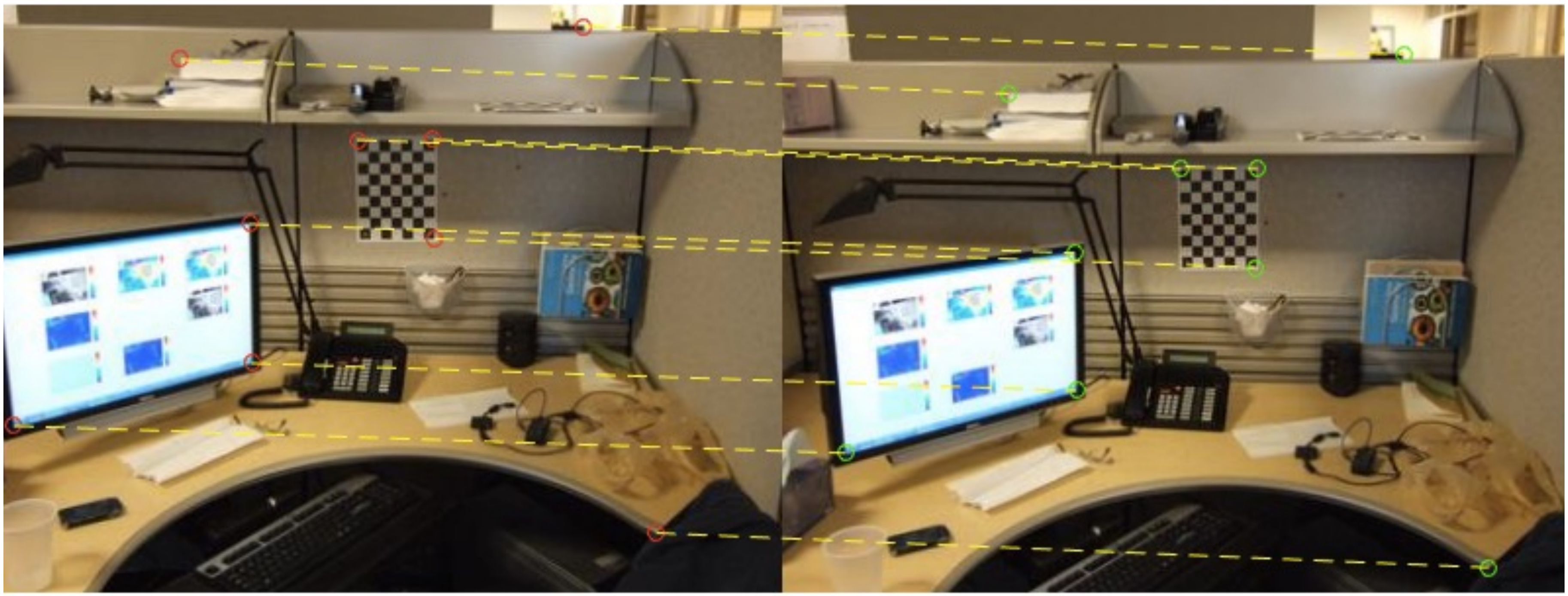
Properties of the Fundamental matrix



$$x^T F x' = 0 \quad \text{with} \quad F = K^{-T} E K'^{-1}$$

- $F x'$ is the epipolar line associated with x' $(l = F x')$
- $F^T x$ is the epipolar line associated with x $(l' = F^T x)$
- $F e' = 0$ and $F^T e = 0$
- F is singular (rank two): det(F)=0
- F has seven degrees of freedom: 9 entries but defined up to scale, det(F)=0

Estimating the fundamental matrix



The eight-point algorithm

$$\boldsymbol{x} = (u, v, 1)^T, \quad \boldsymbol{x}' = (u', v', 1) \qquad x^T F x' = 0$$

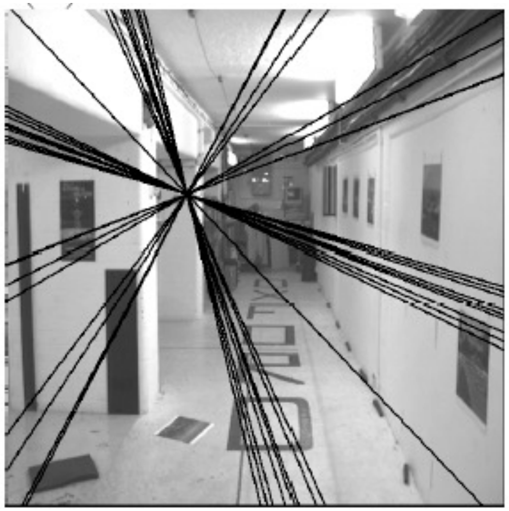
$$\begin{bmatrix} u & v & 1 \end{bmatrix} \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} = 0 \quad \Longrightarrow \quad \begin{bmatrix} u'u & u'v & u' & v'u & v'v & v' & u & v & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = 0$$

Solve homogeneous linear system using eight or more matches

Fundamental matrix has rank 2: det(F) = 0

Enforce rank-2 constraint (take SVD of $\boldsymbol{F}$ and throw out the smallest singular value)

Left: uncorrected F – epipolar lines are not coincident



Right: epipolar lines from corrected F

The eight-point algorithm

$$\begin{bmatrix} u'u & u'v & u' & v'u & v'v & v' & u & v \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \end{bmatrix} = -1$$

Problem with eight-point algorithm

250906.36	183269.57	921.81	200931.10	146766.13	738.21	272.19	198.81
2692.28	131633.03	176.27	6196.73	302975.59	405.71	15.27	746.79
416374.23	871684.30	935.47	408110.89	854384.92	916.90	445.10	931.81
191183.60	171759.40	410.27	416435.62	374125.90	893.65	465.99	418.65
48988.86	30401.76	57.89	298604.57	185309.58	352.87	846.22	525.15
164786.04	546559.67	813.17	1998.37	6628.15	9.86	202.65	672.14
116407.01	2727.75	138.89	169941.27	3982.21	202.77	838.12	19.64
135384.58	75411.13	198.72	411350.03	229127.78	603.79	681.28	379.48

$$\begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \end{bmatrix} = -1$$

- Poor numerical conditioning
- Can be fixed by rescaling the data

The normalized eight-point algorithm

(Hartley, 1995)

- Center the image data at the origin, and scale it so the mean squared distance between the origin and the data points is 2 pixels
- Use the eight-point algorithm to compute $\boldsymbol{F}$ from the normalized points
- Enforce the rank-2 constraint (for example, take SVD of $\boldsymbol{F}$ and throw out the smallest singular value)
- Transform fundamental matrix back to original units: if $\boldsymbol{T}$ and $\boldsymbol{T}'$ are the normalizing transformations in the two images, than the fundamental matrix in original coordinates is $\boldsymbol{T}'^{T} \boldsymbol{F} \boldsymbol{T}$

The (normalized) eight-point algorithm

1. Solve a system of homogeneous linear equations (for normalized points data)
   a. Write down the system of equations
   b. Solve **f** from A**f**=**0** using SVD

      Matlab:
      ```
      [U, S, V] = svd(A); %A = U*S*V'
      f = V(:, end);
      F = reshape(f, [3 3])';
      ```

2. Resolve det(F) = 0 constraint using SVD

   Matlab:
   ```
   [U, S, V] = svd(F);
   S(3,3) = 0;
   F = U*S*V';
   ```

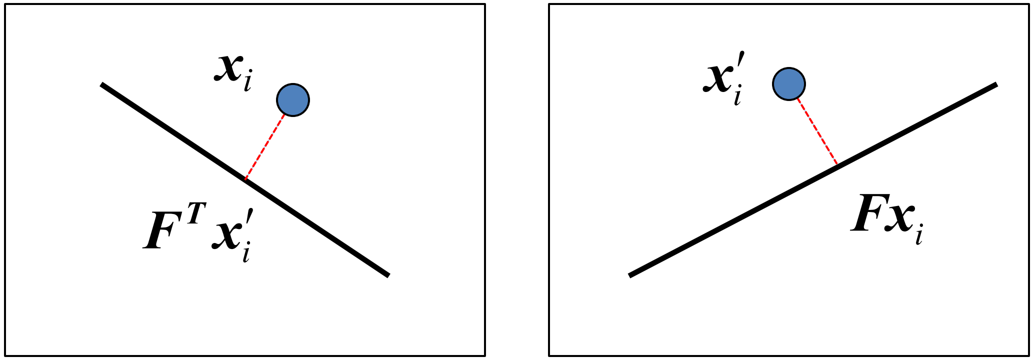
Nonlinear estimation

- Linear estimation minimizes the sum of squared *algebraic* distances between points $\boldsymbol{x}'_i$ and epipolar lines $\boldsymbol{F}\,\boldsymbol{x}_i$ (or points $\boldsymbol{x}_i$ and epipolar lines $\boldsymbol{F}^T\boldsymbol{x}'_i$):

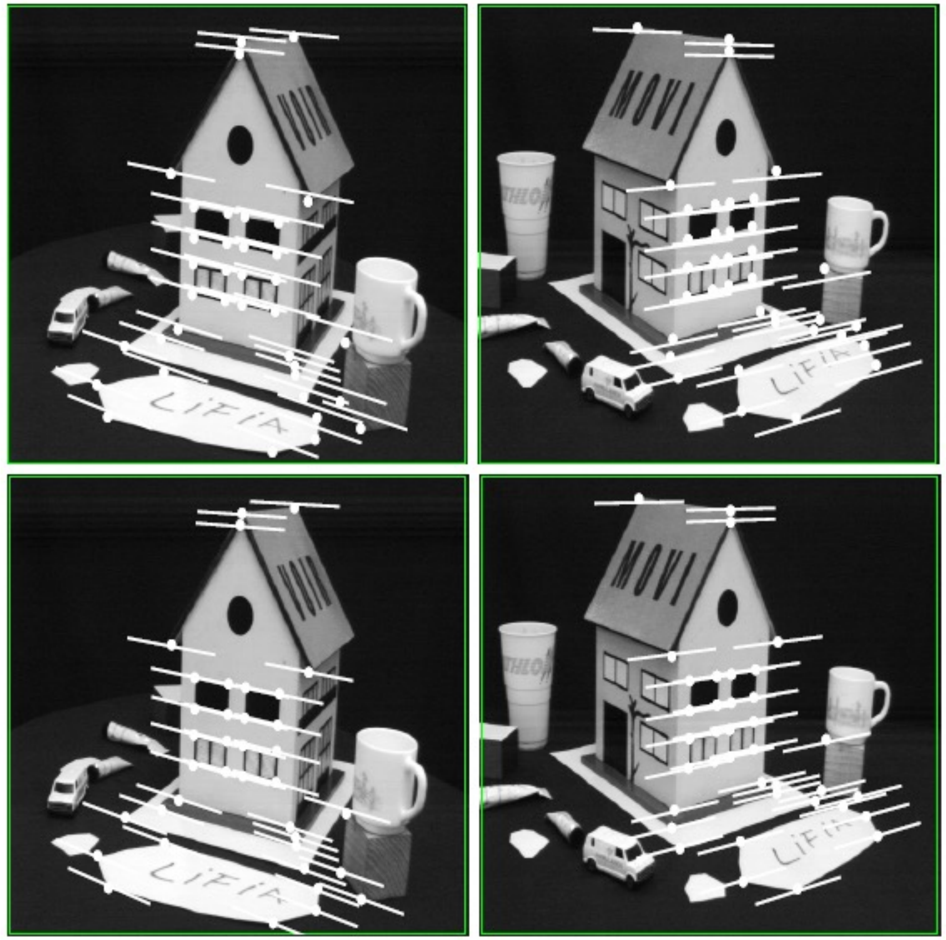
$$\sum_{i=1}^{N} (\boldsymbol{x}_i'^T \boldsymbol{F}\,\boldsymbol{x}_i)^2$$

- Nonlinear approach: minimize sum of squared *geometric* distances

$$\sum_{i=1}^{N} \left[\mathrm{d}^2(\boldsymbol{x}_i', \boldsymbol{F}\,\boldsymbol{x}_i) + \mathrm{d}^2(\boldsymbol{x}_i, \boldsymbol{F}^T \boldsymbol{x}_i') \right]$$

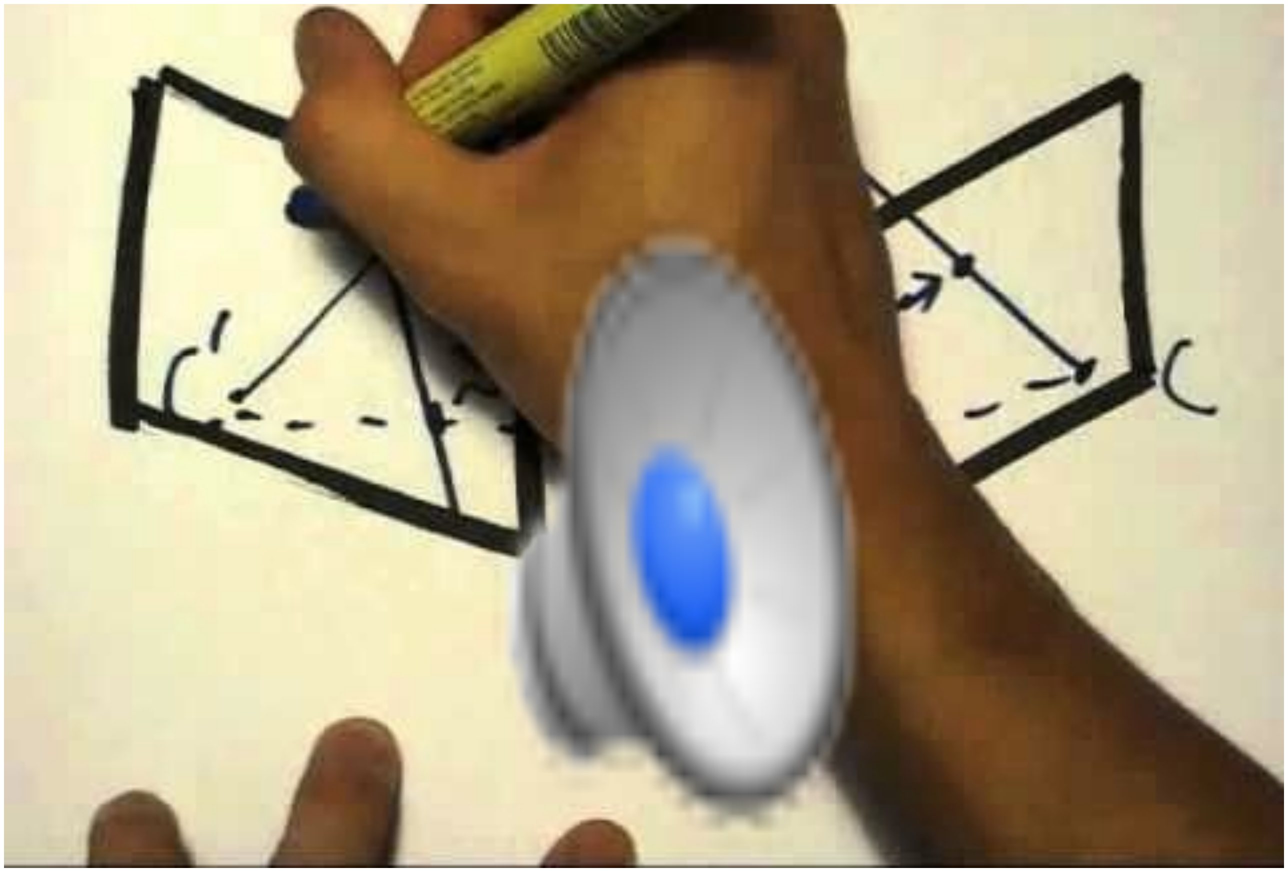


Comparison of estimation algorithms



	8-point	Normalized 8-point	Nonlinear least squares
Av. Dist. 1	2.33 pixels	0.92 pixel	0.86 pixel
Av. Dist. 2	2.18 pixels	0.85 pixel	0.80 pixel

The Fundamental Matrix Song



http://danielwedge.com/fmatrix/

Laboratory class 6

The goal of this lab is to introduce you to camera and scene geometry. Specifically we will estimate the camera projection matrix, which maps 3D world coordinates to image coordinates, as well as the fundamental matrix, which relates points in one scene to epipolar lines in another. The camera projection matrix and the fundamental matrix can each be estimated using point correspondences. To estimate the projection matrix (camera calibration), the input is corresponding 3D and 2D points. To estimate the fundamental matrix the input is corresponding 2D points across two images. We start by estimating the projection matrix and the fundamental matrix for a scene with ground truth correspondences. Then we'll move on to estimating the fundamental matrix using point correspondences from SIFT.

Tutorial on epipolar geometry is here: https://docs.opencv.org/master/da/de9/tutorial_py_epipolar_geometry.html or here: https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_calib3d/py_epipolar_geometry/py_epipolar_geometry.html

Data

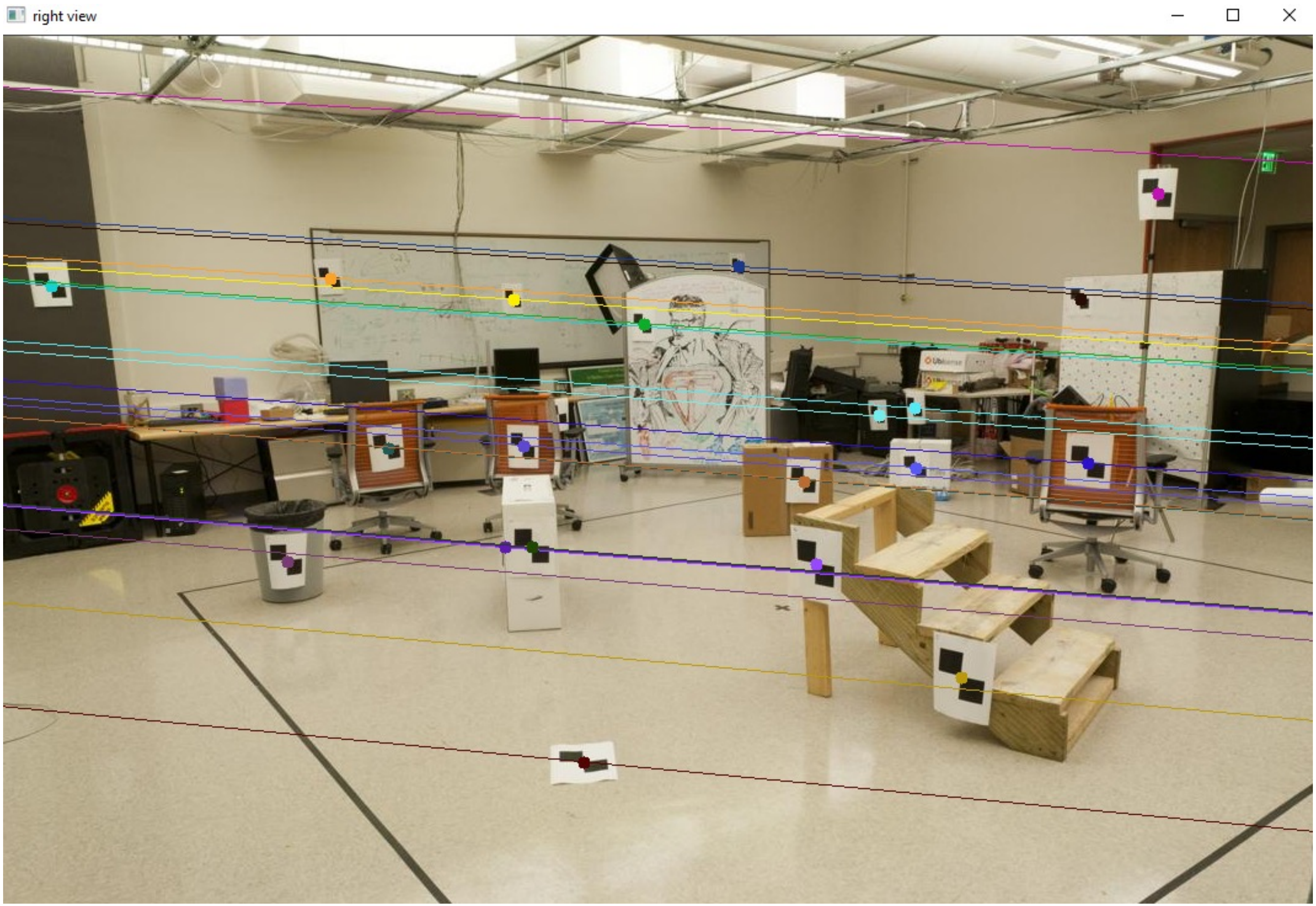
We provide 2D and 3D ground truth point correspondences for the base image pair (pic_a.jpg and pic_b.jpg), as well as other images which will not have any ground truth dataset.

Image 1 of a scene	Image 2 of a scene
	

This laboratory consists of three parts: (1) estimating the projection matrix, (2) estimating the fundamental matrix, (3) estimating the fundamental matrix with unreliable SIFT matches using RANSAC.

Laboratory class 6

Laboratory class 6



Moving on to stereo…

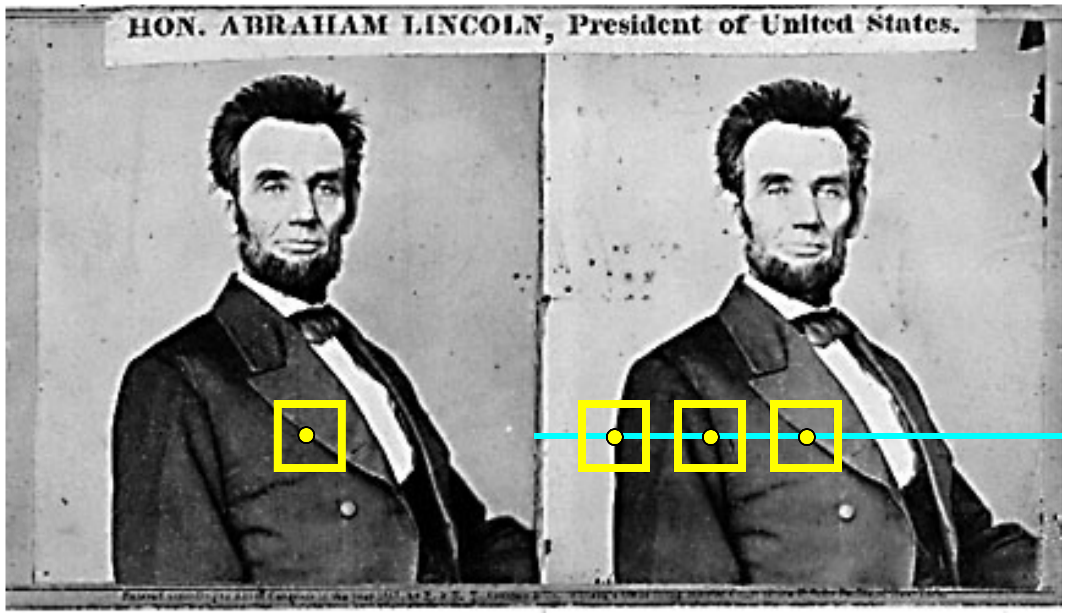
Fuse a calibrated binocular stereo pair to produce a depth image

image 1

image 2

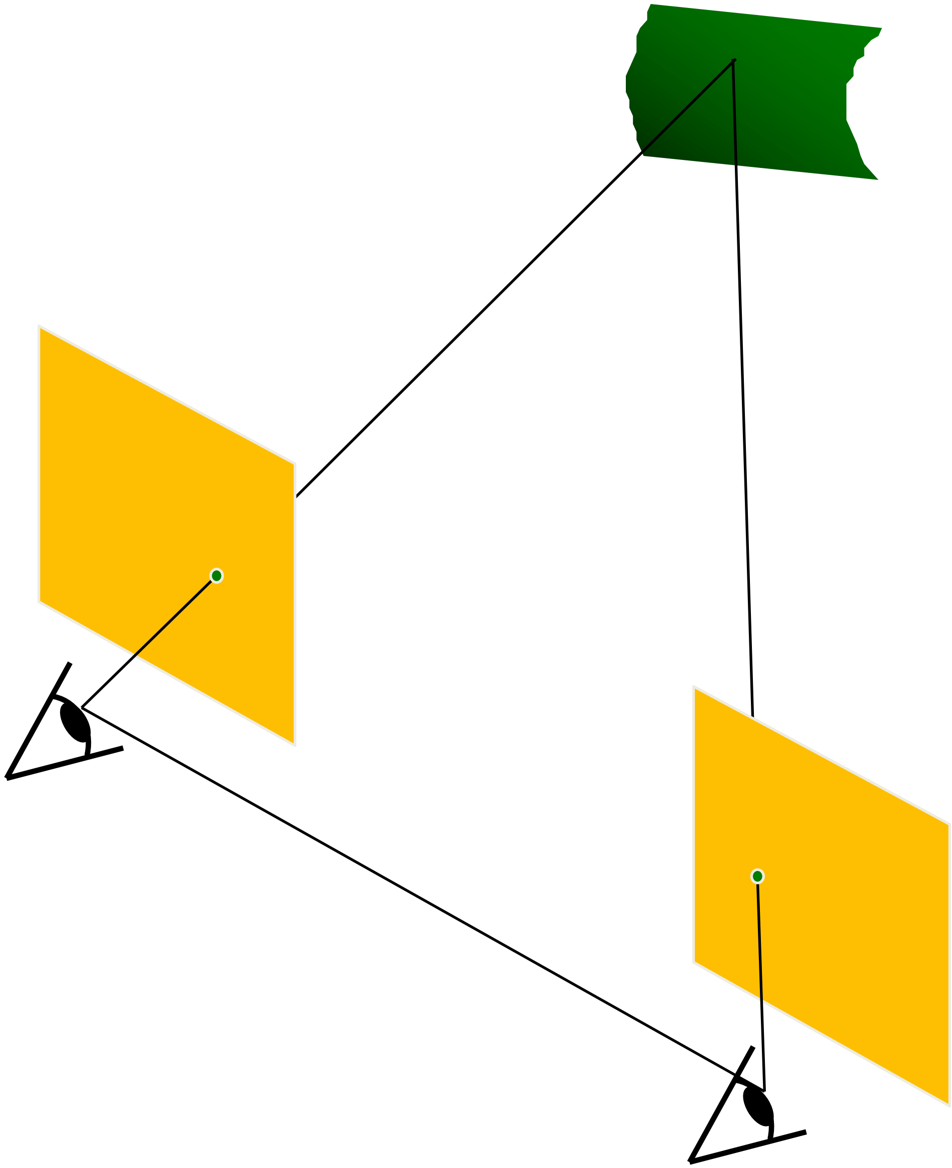
Dense depth map

Basic stereo matching algorithm



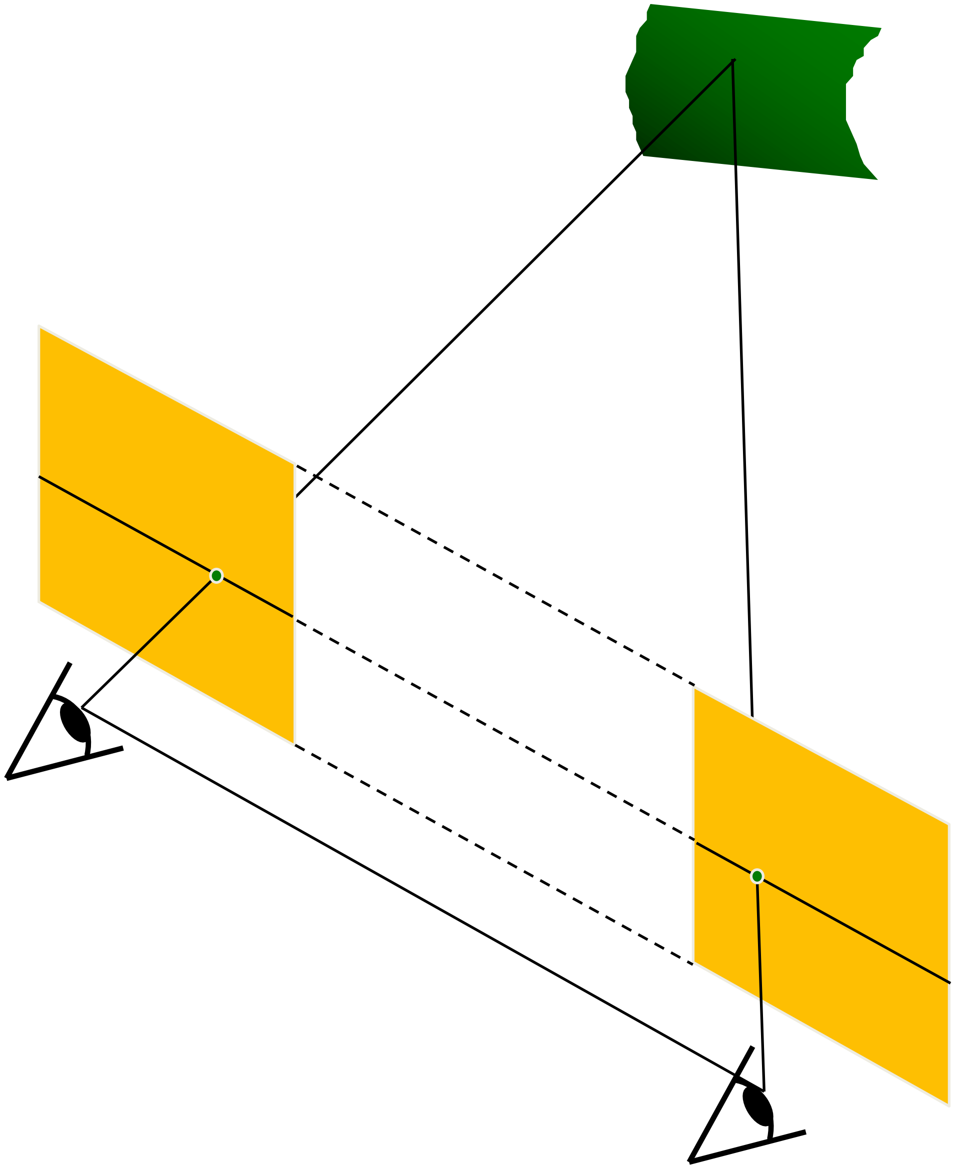
- For each pixel in the first image
  - Find corresponding epipolar line in the right image
  - Search along epipolar line and pick the best match
  - Triangulate the matches to get depth information
- Simplest case: epipolar lines are scanlines
  - When does this happen?

Simplest Case: Parallel images



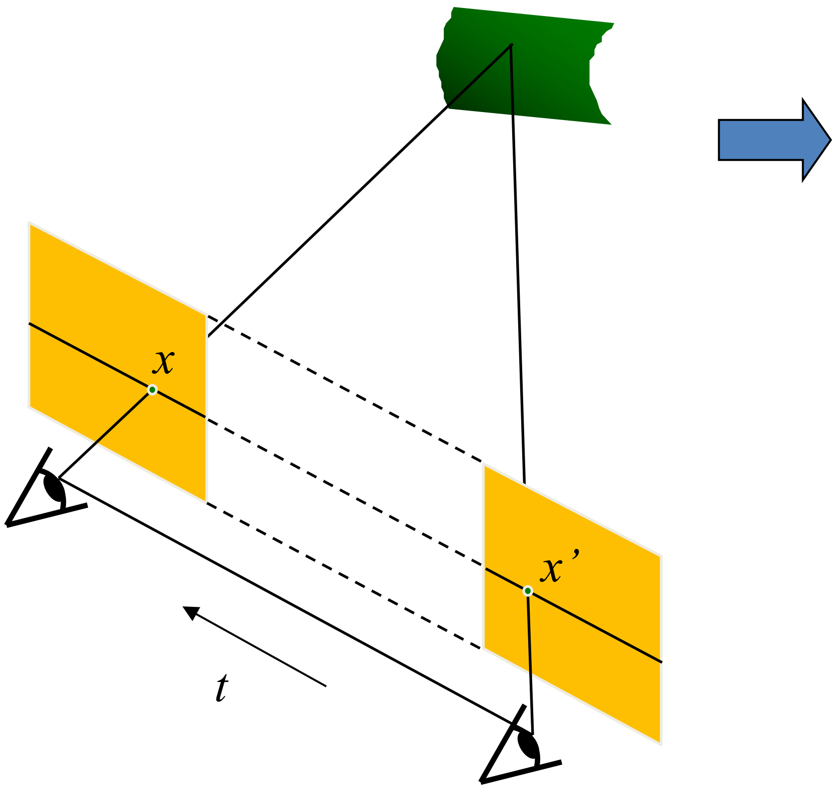
- Image planes of cameras are parallel to each other and to the baseline
- Camera centers are at same height
- Focal lengths are the same

Simplest Case: Parallel images



- Image planes of cameras are parallel to each other and to the baseline
- Camera centers are at same height
- Focal lengths are the same
- Then, epipolar lines fall along the horizontal scan lines of the images

Simplest Case: Parallel images



Epipolar constraint:

$$\hat{x}^T E \hat{x}' = 0 \quad \text{with} \quad E = [t]_\times R$$

$$R = I \qquad t = (T, 0, 0)$$

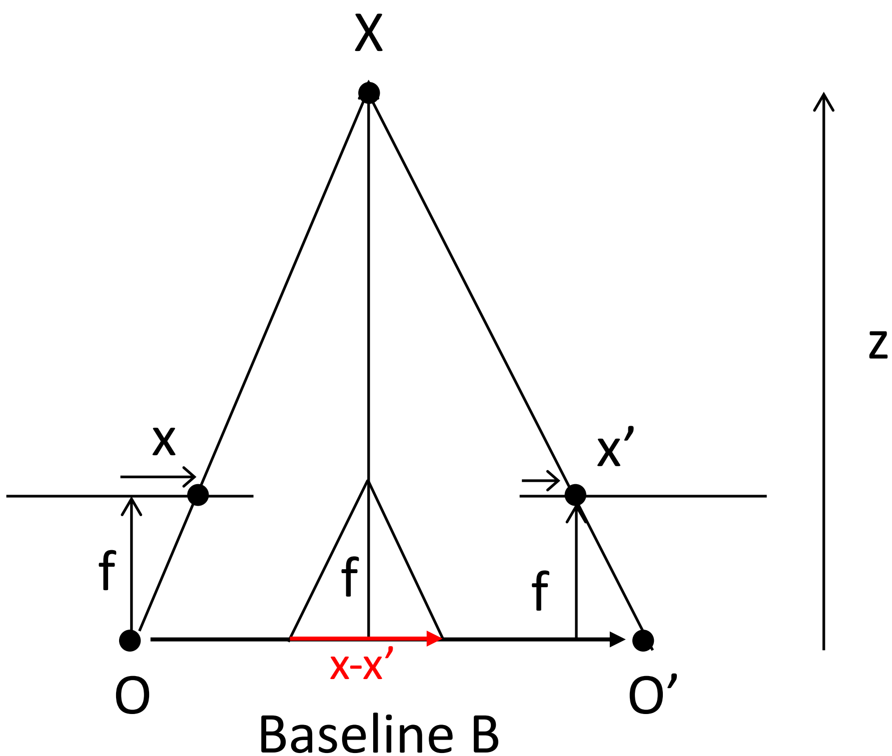
$$[t]_\times = \begin{bmatrix} 0 & -t_3 & t_2 \\ t_3 & 0 & -t_1 \\ -t_2 & t_1 & 0 \end{bmatrix}$$

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

The y-coordinates of corresponding points are the same

Depth from disparity

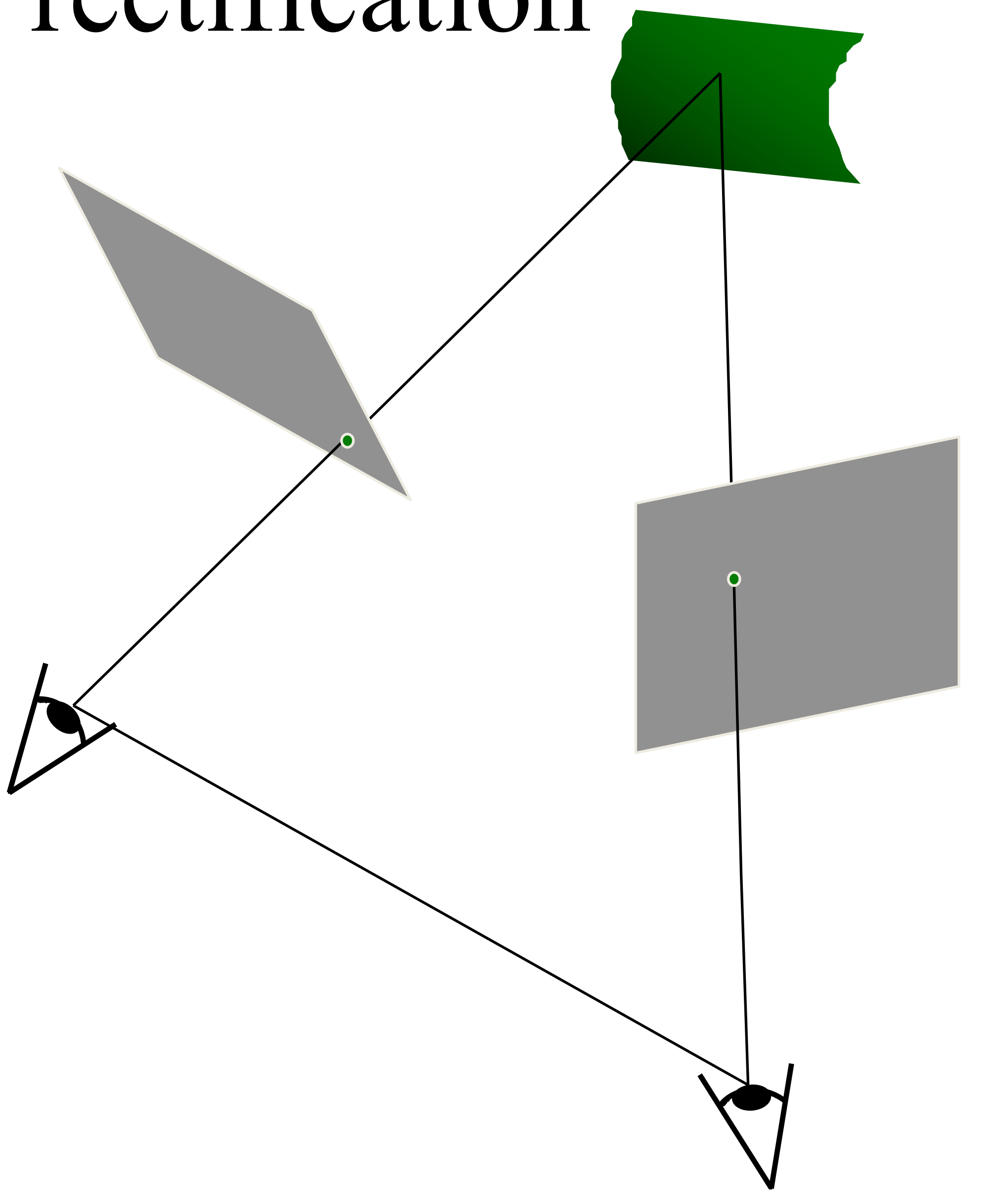
$$\frac{x - x'}{O - O'} = \frac{f}{z}$$



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

Disparity is inversely proportional to depth.

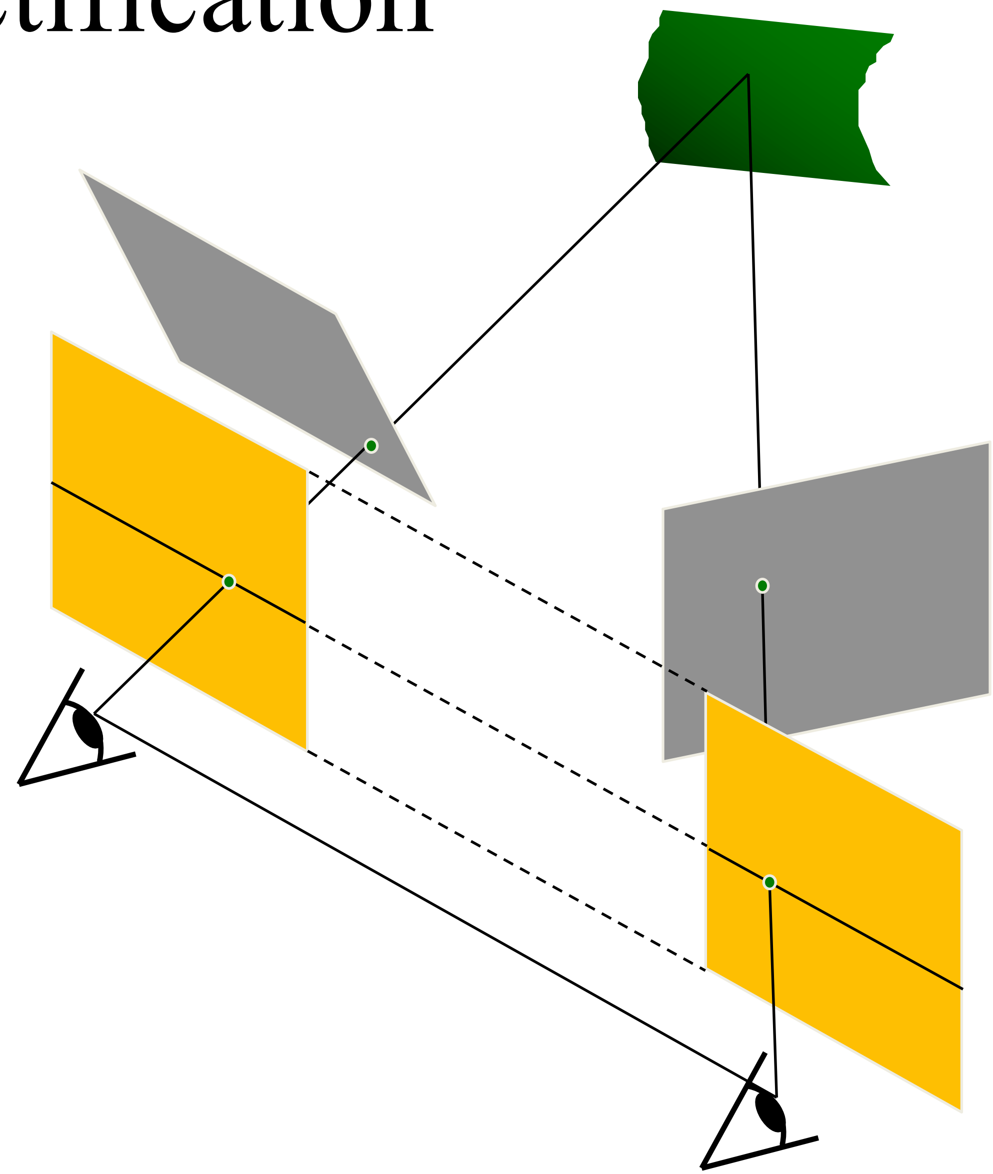
Stereo image rectification



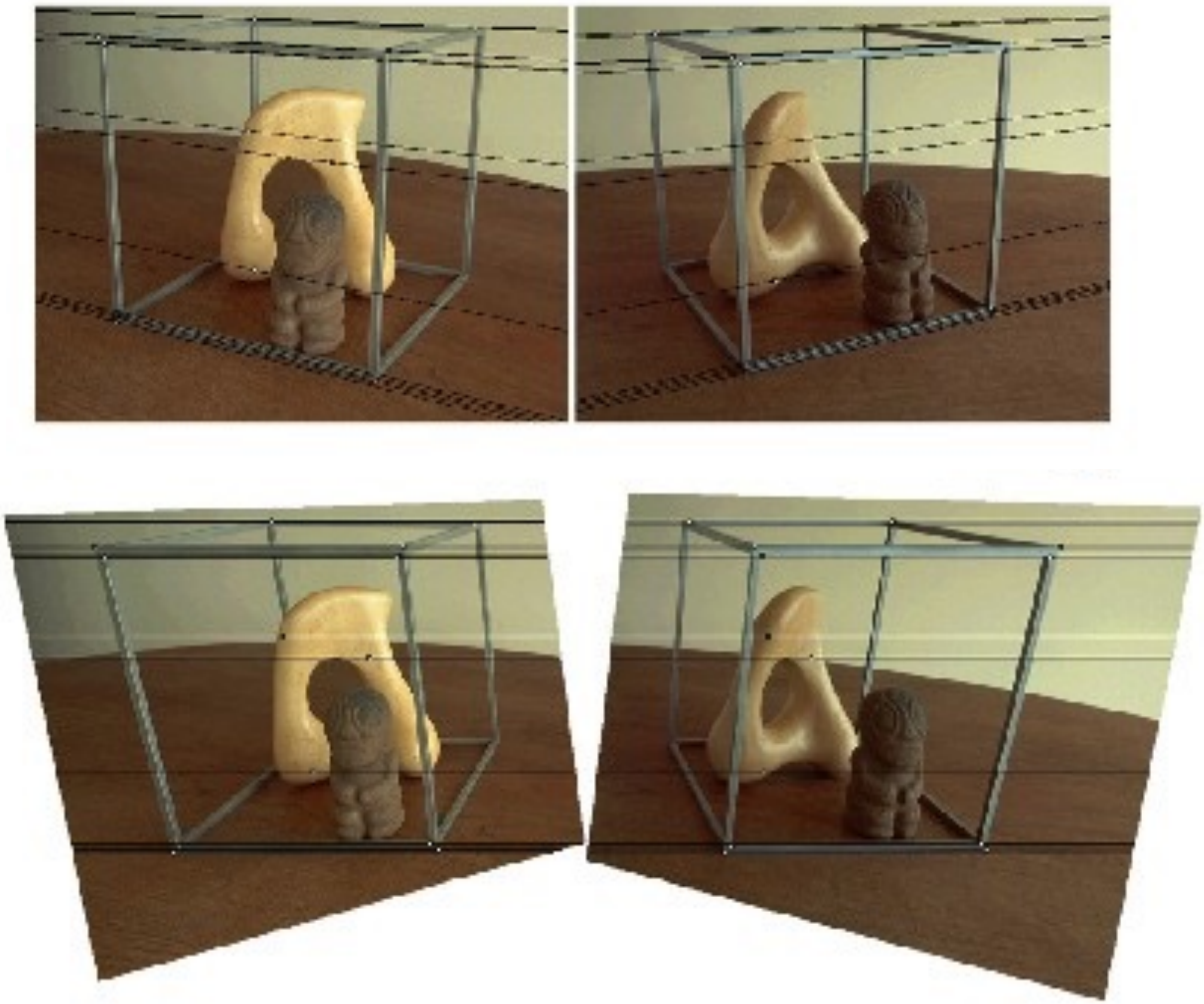
Stereo image rectification

- Reproject image planes onto a common plane parallel to the line between camera centers
- Pixel motion is horizontal after this transformation
- Two homographies (3x3 transform), one for each input image reprojection

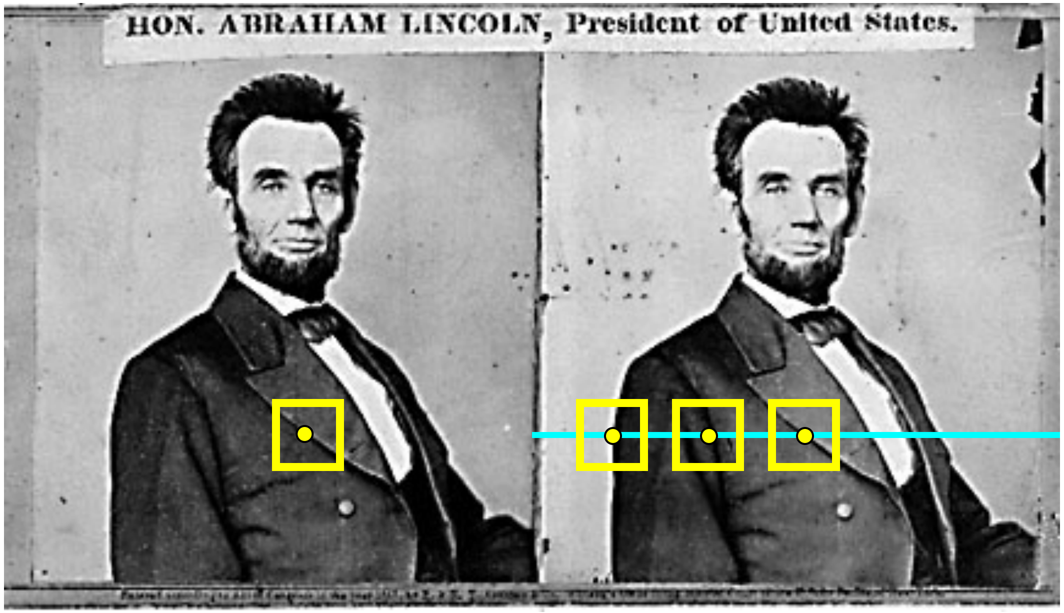
➢ C. Loop and Z. Zhang. [Computing Rectifying Homographies for Stereo Vision](). IEEE Conf. Computer Vision and Pattern Recognition, 1999.



Rectification example

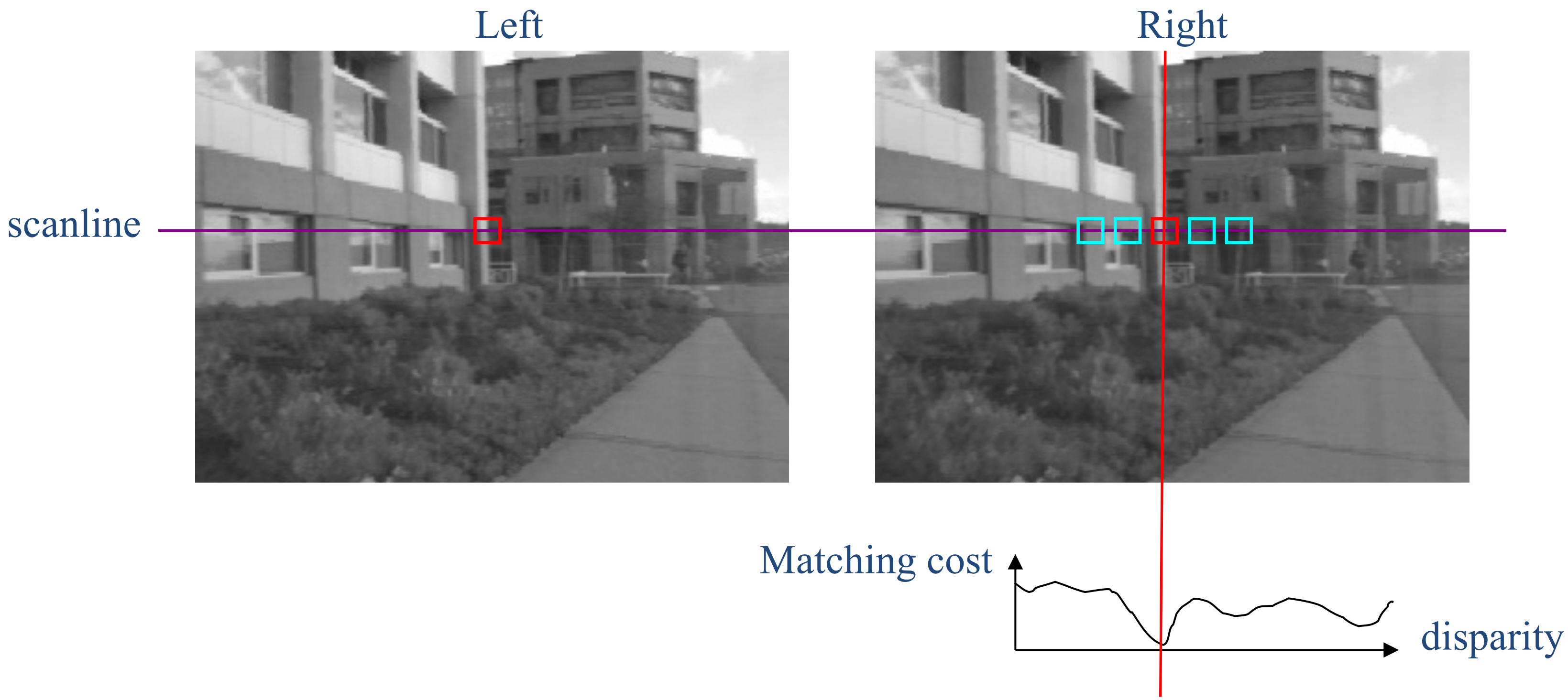


Basic stereo matching algorithm



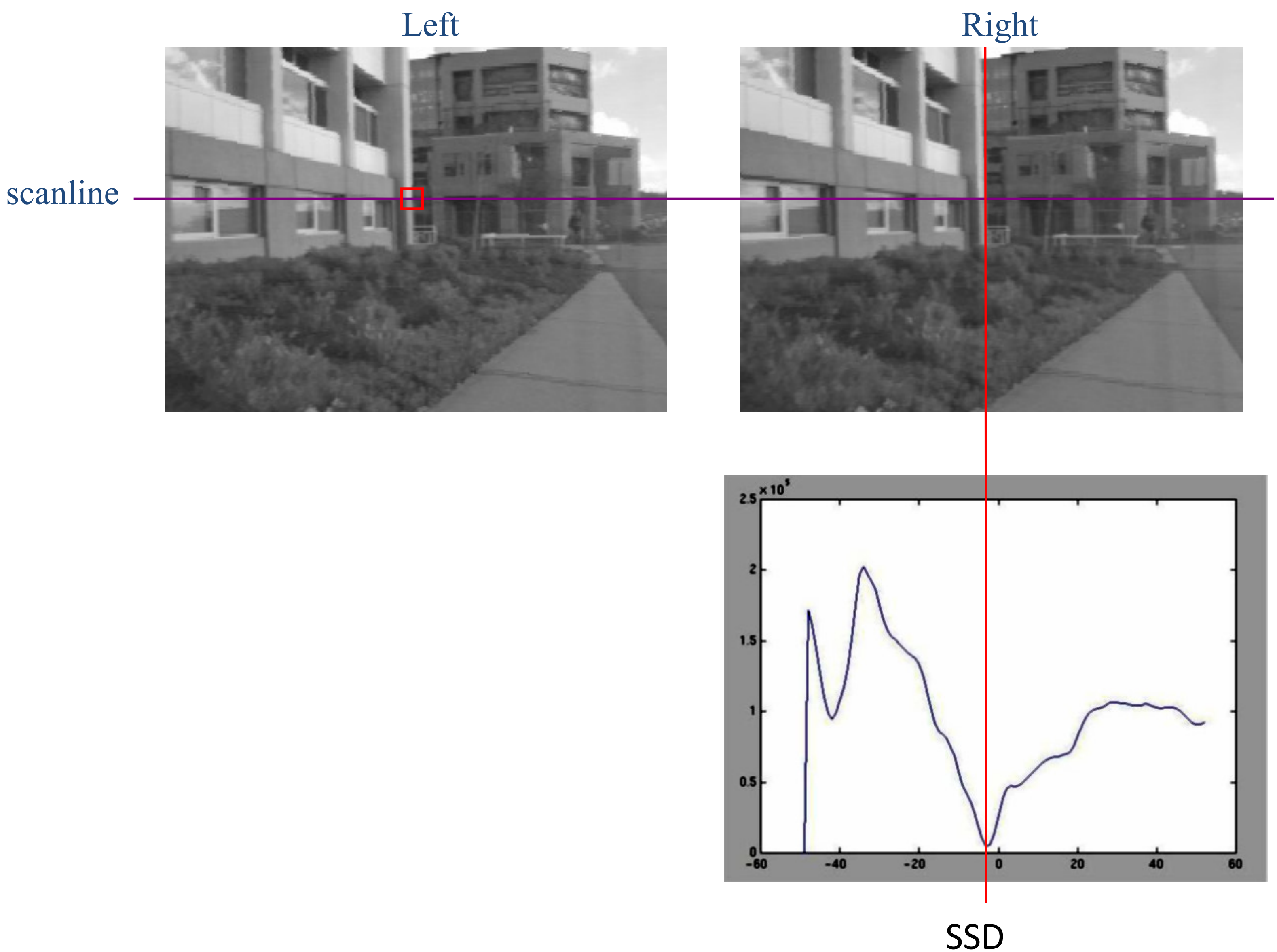
- If necessary, rectify the two stereo images to transform epipolar lines into scanlines
- For each pixel x in the first image
  - Find corresponding epipolar scanline in the right image
  - Search the scanline and pick the best match x'
  - Compute disparity x-x' and set depth(x) = fB/(x-x')

Correspondence search

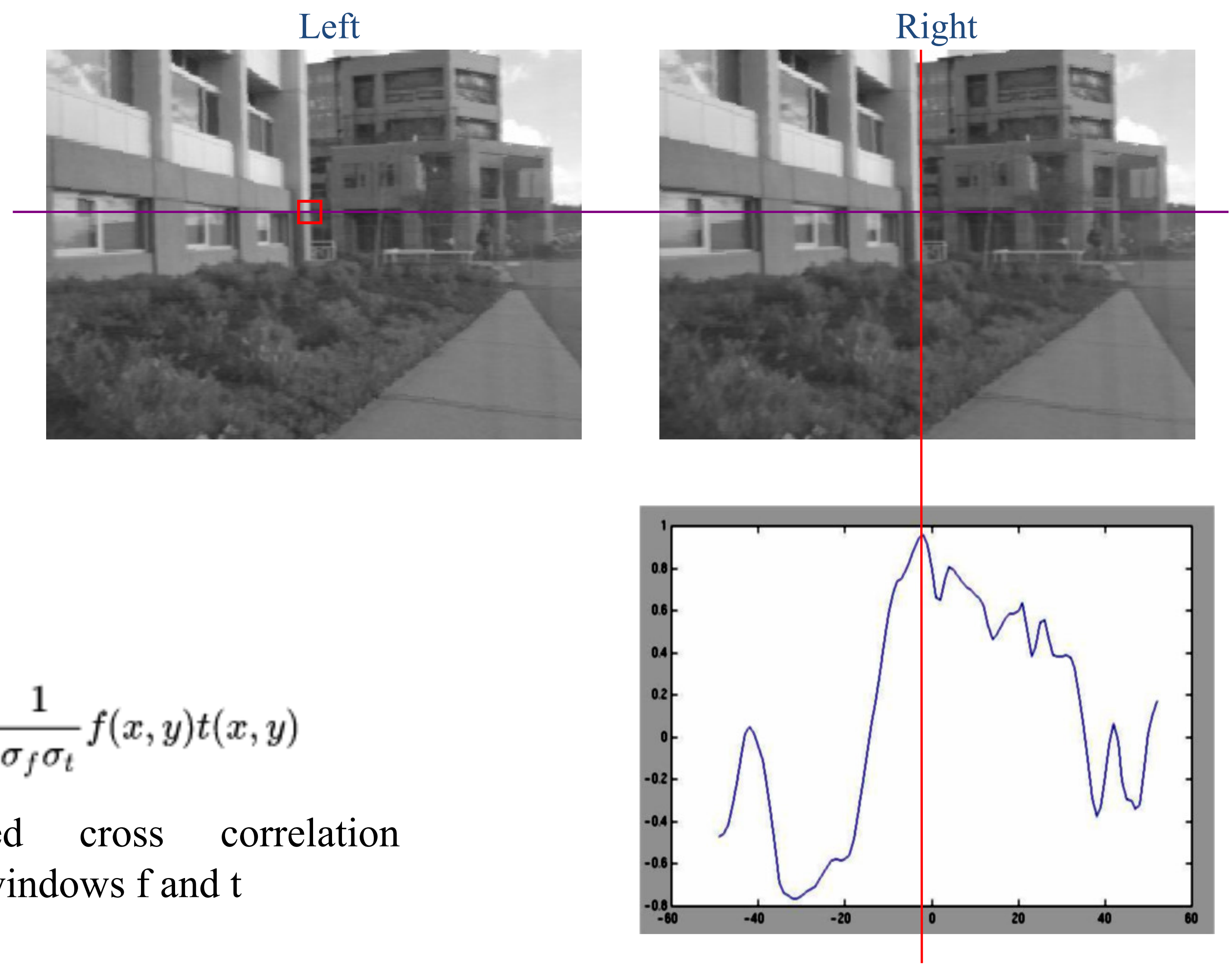


- Slide a window along the right scanline and compare contents of that window with the reference window in the left image
- Matching cost: SSD or normalized correlation

Correspondence search



Correspondence search



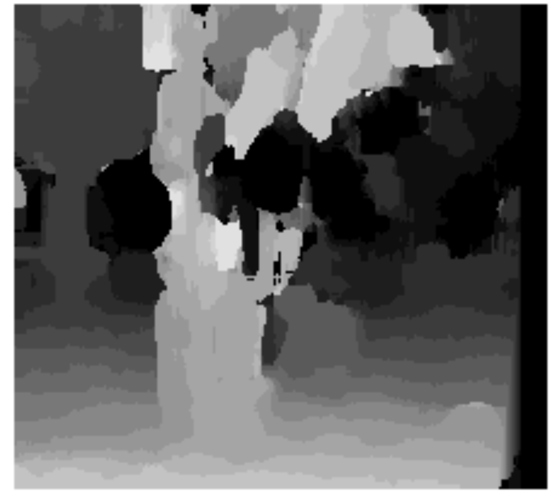
Norm. corr

$$\frac{1}{n}\sum_{x,y}\frac{1}{\sigma_f \sigma_t} f(x,y)t(x,y)$$

Normalized cross correlation between windows f and t

Effect of window size

W = 3

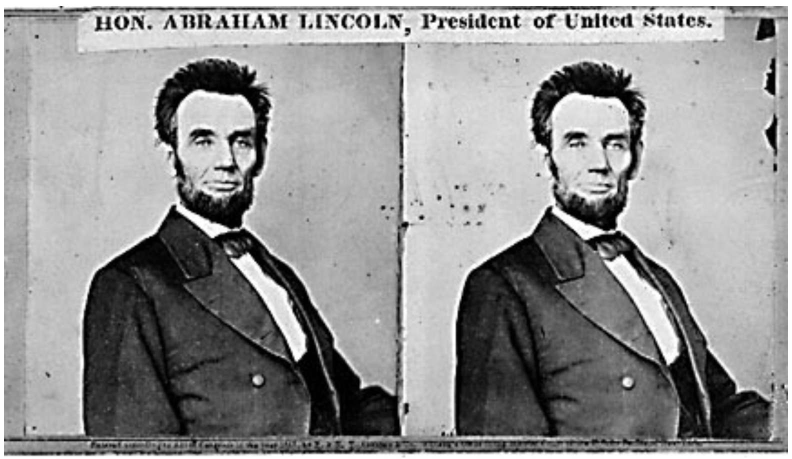


W = 20

- Smaller window
  + More detail
  – More noise

- Larger window
  + Smoother disparity maps
  – Less detail
  – Fails near boundaries

Failures of correspondence search



Textureless surfaces

Occlusions, repetition

Non-Lambertian surfaces, specularities

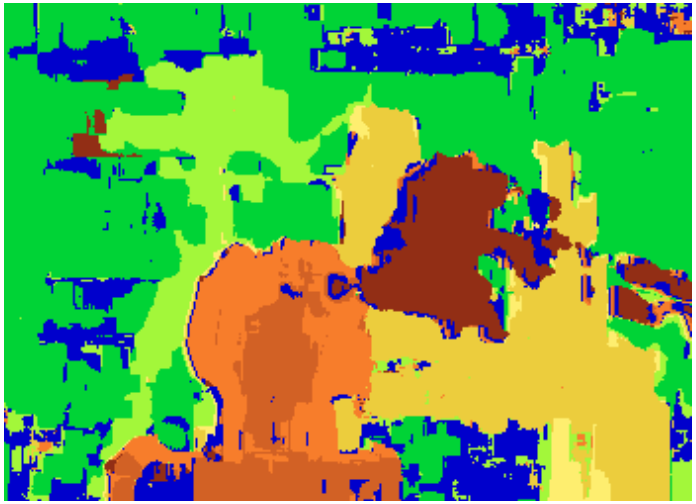
Left view

Right view

Results with window search

Data

Window-based matching



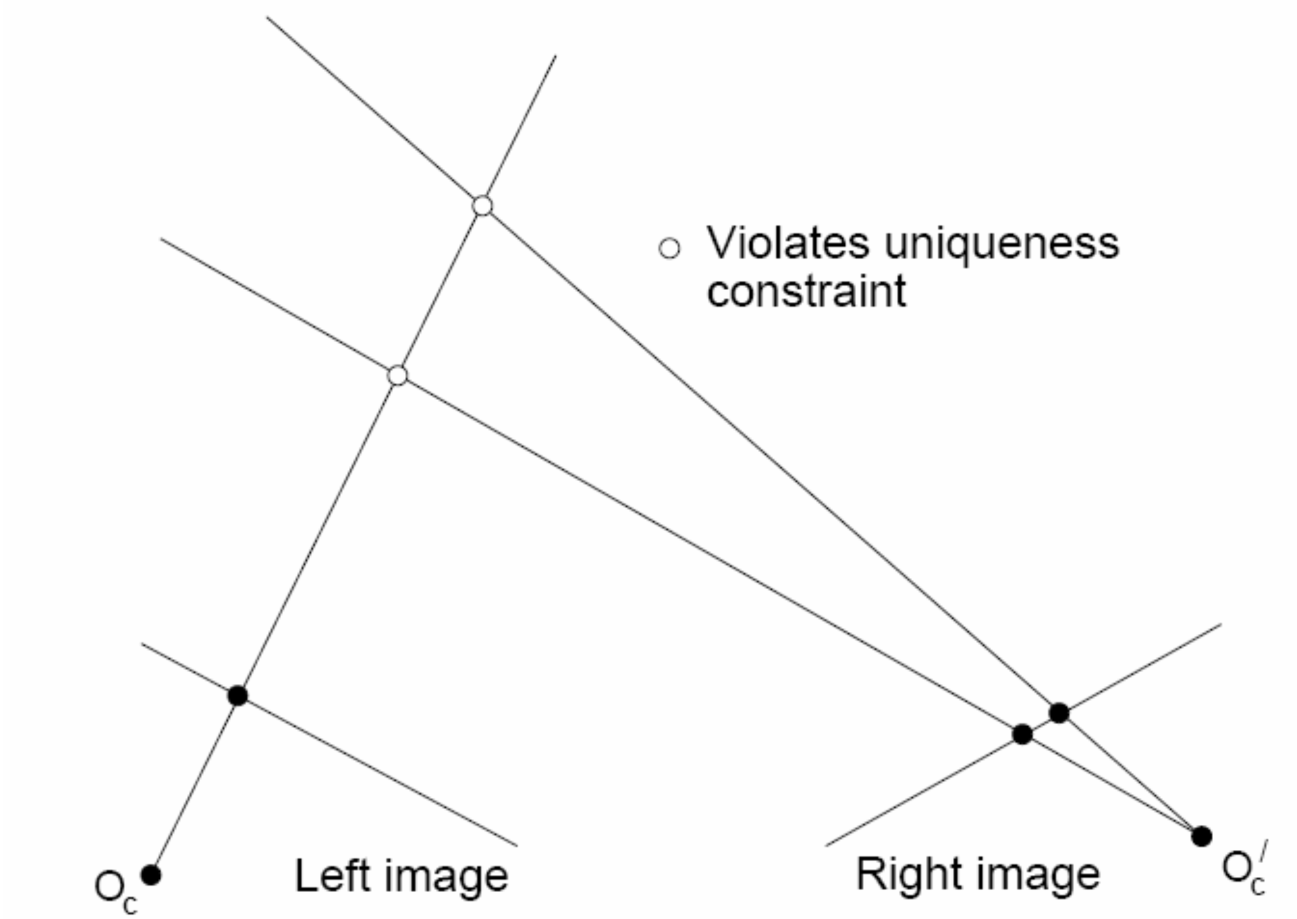
Ground truth

How can we improve window-based matching?

- So far, matches are independent for each point
- What constraints or priors can we add?

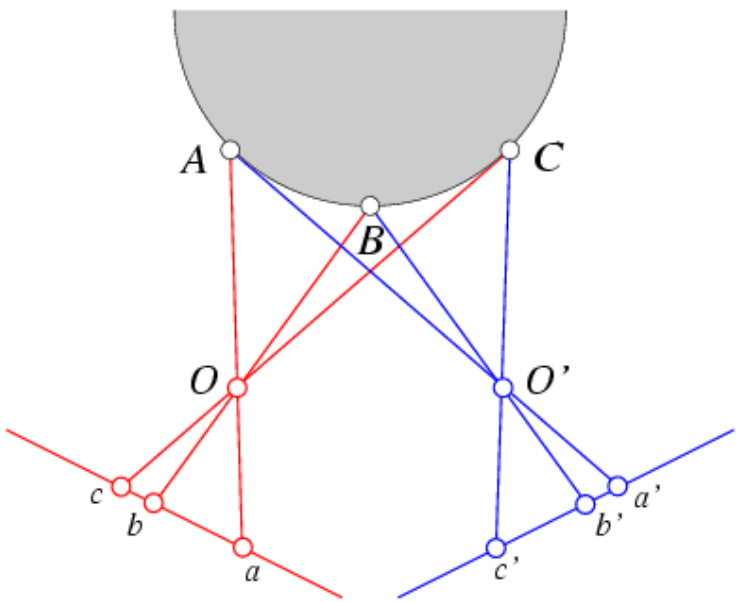
Stereo constraints/priors

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



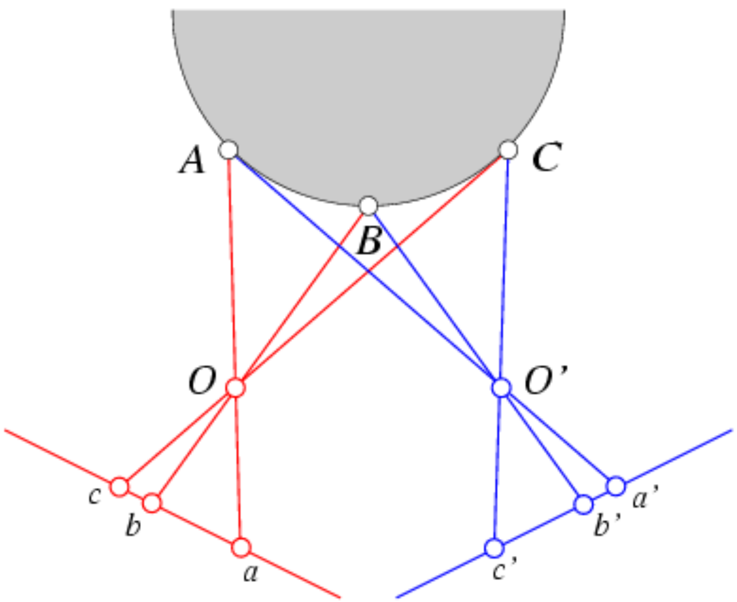
Stereo constraints/priors

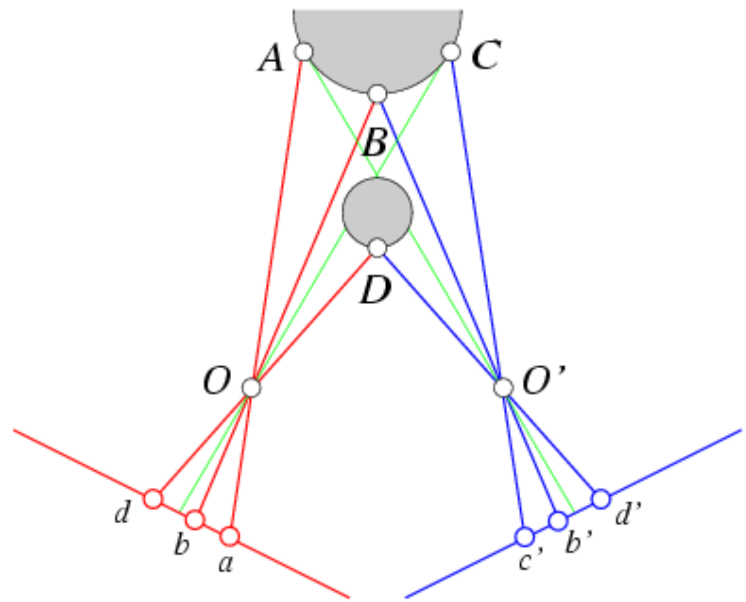
- Uniqueness
  - For any point in one image, there should be at most one matching point in the other image
- Ordering
  - Corresponding points should be in the same order in both views



Stereo constraints/priors

- 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



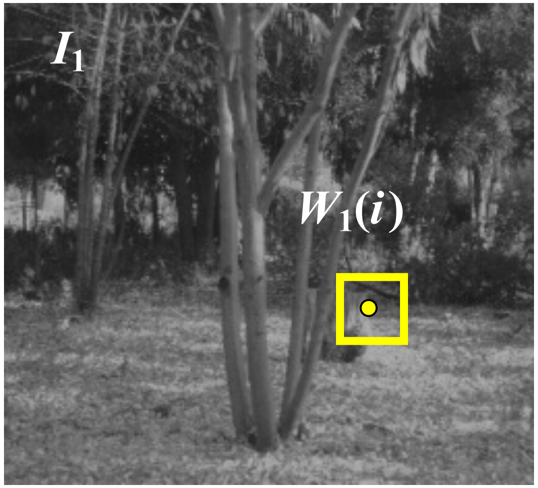


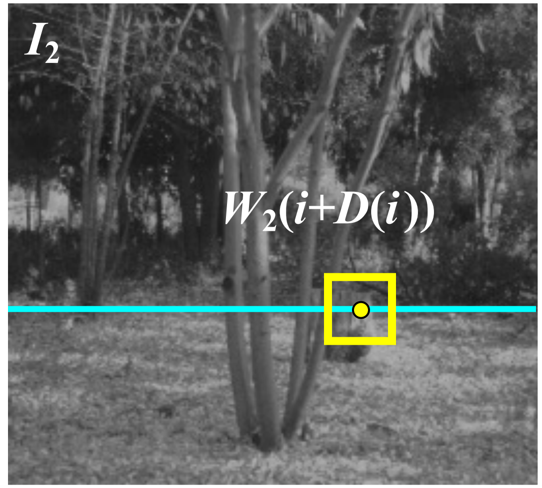
Ordering constraint doesn't hold

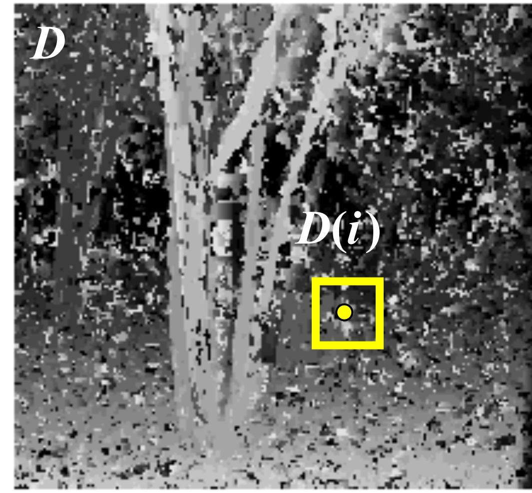
Priors and constraints

- 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
  - We expect disparity values to usually change slowly

Stereo matching as energy minimization







$$E = E_{\text{data}}(D; I_1, I_2) + \beta E_{\text{smooth}}(D)$$ D – disparity map

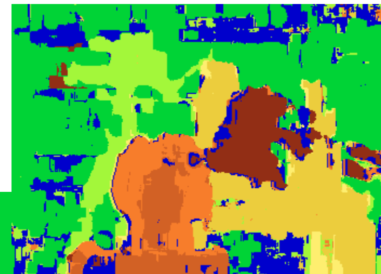
$$E_{\text{data}} = \sum_i \left(W_1(i) - W_2(i + D(i)) \right)^2 \qquad E_{\text{smooth}} = \sum_{\text{neighbors } (i,j)} \left\| D(i) - D(j) \right\|^2$$

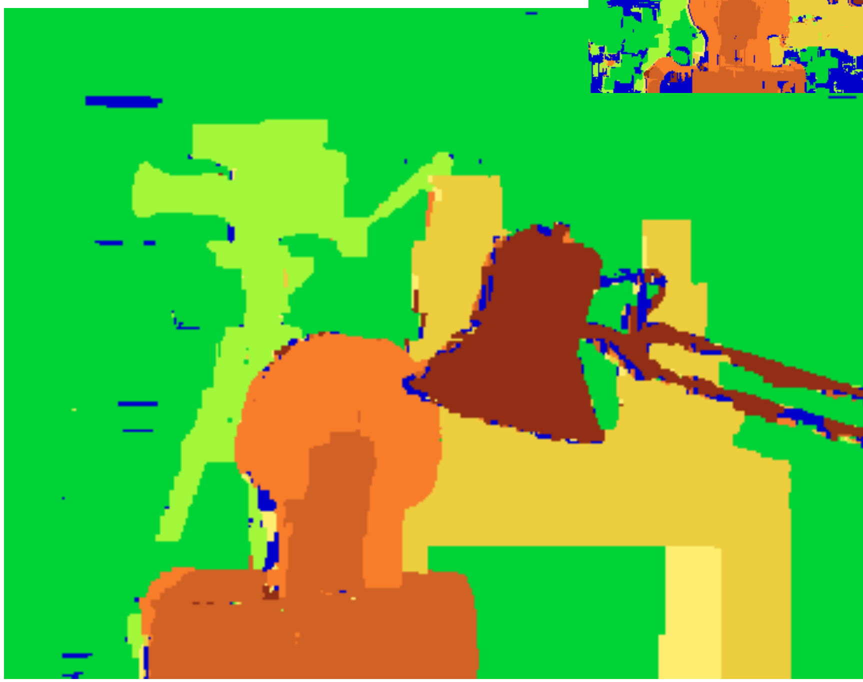
- Energy functions of this form can be minimized using *graph cuts*

Y. Boykov, O. Veksler, and R. Zabih, [Fast Approximate Energy Minimization via Graph Cuts](), PAMI 2001

Many of these constraints can be encoded in an energy function and solved using graph cuts

Before





Graph cuts

Ground truth

Y. Boykov, O. Veksler, and R. Zabih, [Fast Approximate Energy Minimization via Graph Cuts](), PAMI 2001

For the latest and greatest: [http://www.middlebury.edu/stereo/]()