CSCI 497P/597P: Computer Vision

Scott Wehrwein

Intrinsics, Extrinsics, and Stereo

Reading

- Szeliski: Chapter 2.1.5-6, 6.3

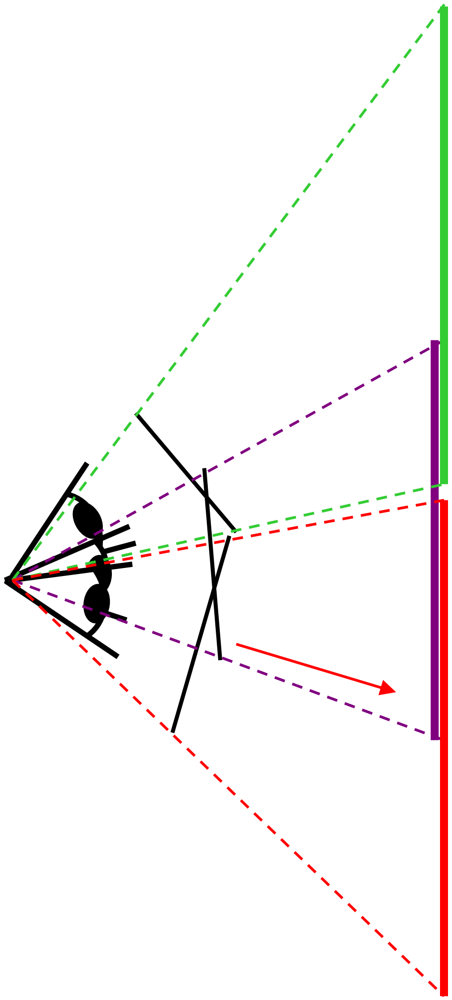
Goals

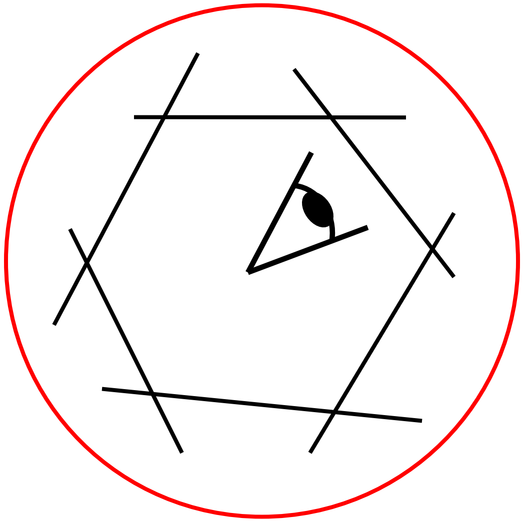
- Know the distinction between intrinsic and extrinsic camera parameters, and how they fit together with the projection matrix to transform world coordinates to pixel coordinates.
- Understand how to calculate depth from disparity in a rectified stereo image pair.

Announcements

- A2 is out, due 2/22 (2 weeks from today)
- Homework 1 out soon (this weekend)
  - Primarily intended as midterm practice
  - Full credit for a faithful attempt at solving the problems.
  - Due TBD, maybe 2/13 or 14

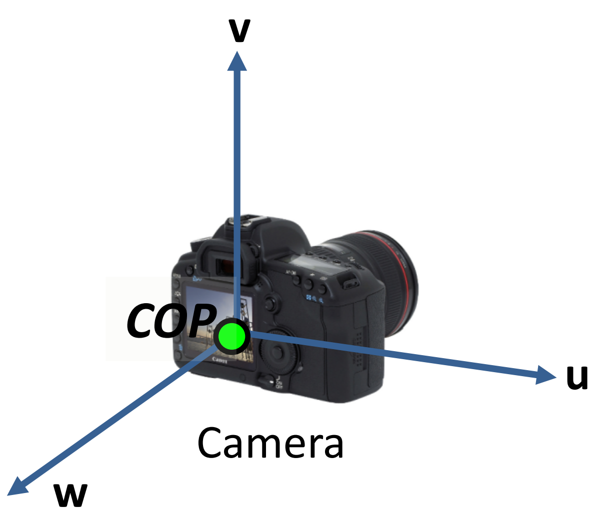
Panoramas require a common COP





Camera(s) without a common COP

- With panoramas, we always assumed a common COP.
- How can we model the geometry of a camera in a separate world coordinate system?

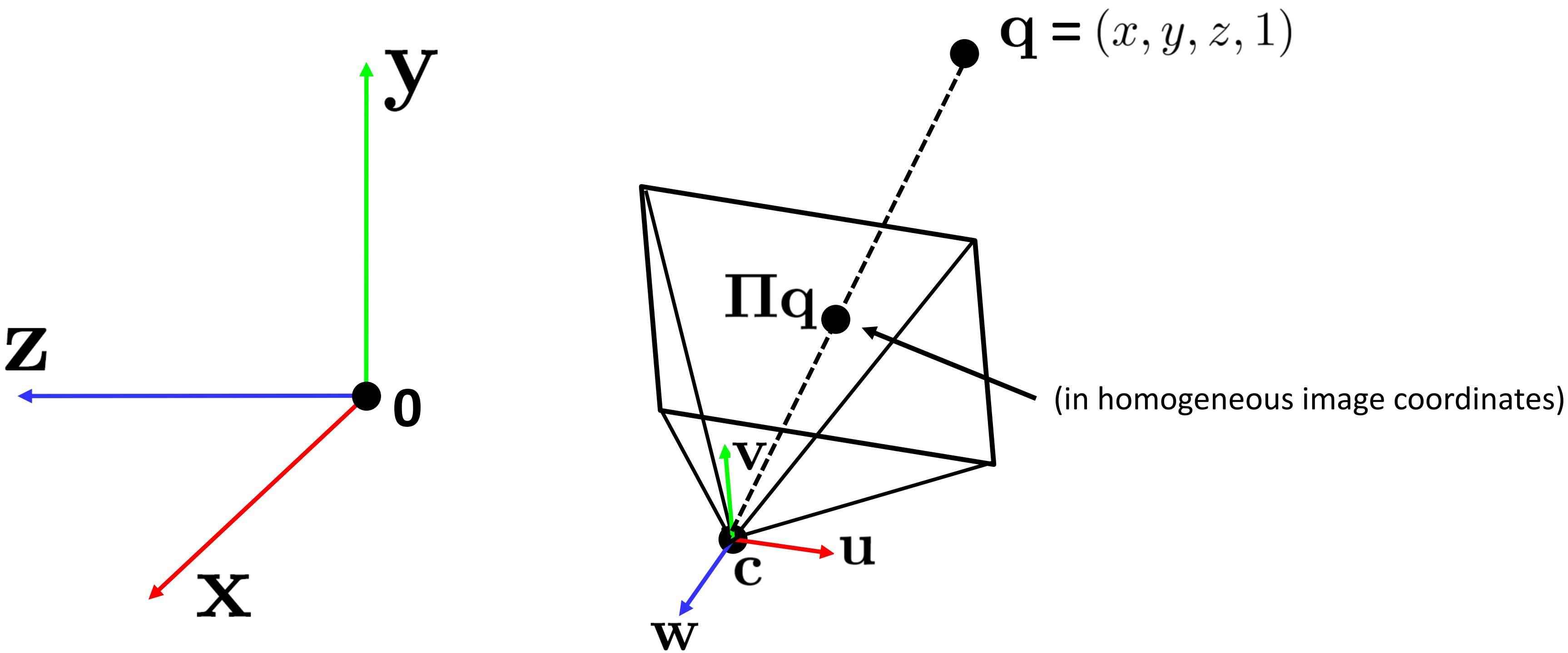


Two important coordinate systems:

1. *World* coordinate system
2. *Camera* coordinate system

How do we project a given point (x, y, z) in world coordinates?

Projection matrix



Intrinsic Camera Parameters

- Everything you need to get from **camera** coordinates to **pixel** coordinates:

$$\underbrace{\begin{bmatrix} -f & 0 & 0 \\ 0 & -f & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{K}} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

(intrinsics)

(converts from 3D rays in camera coordinate system to pixel coordinates)

- Getting more general:

$$\mathbf{K} = \begin{bmatrix} -f & s & c_x \\ 0 & -\alpha f & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

Intrinsic Camera Parameters

Everything you need to get from **camera** coordinates to **pixel** coordinates:

$$\underbrace{\begin{bmatrix} -f & 0 & 0 \\ 0 & -f & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{K}} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$\mathbf{K}$ (intrinsics)

(converts from 3D rays in camera coordinate system to pixel coordinates)

in general, $\mathbf{K} = \begin{bmatrix} -f & s & c_x \\ 0 & -\alpha f & c_y \\ 0 & 0 & 1 \end{bmatrix}$ (upper triangular matrix)

α : **aspect ratio** (1 unless pixels are not square)

s : **skew** (0 unless pixels are shaped like rhombi/parallelograms)

(c_x, c_y) : **principal point** ((0,0) unless optical axis doesn’t intersect projection plane at origin)

Extrinsic Camera Parameters

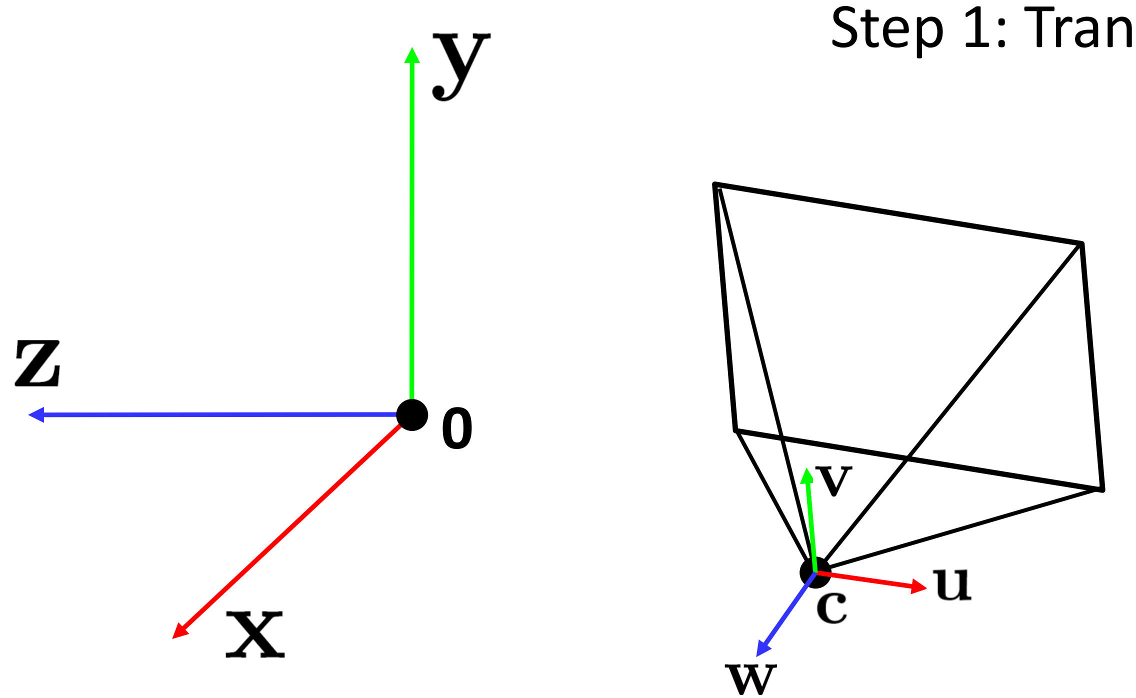
- Everything you need to get from **world** coordinates to **camera** coordinates

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} & \mathbf{R} & & \begin{matrix} 0 \\ 0 \\ 0 \end{matrix} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{I}_{3\times 3} & -\mathbf{c} \\ 0 \quad 0 \quad 0 & 1 \end{bmatrix}$$

Extrinsics

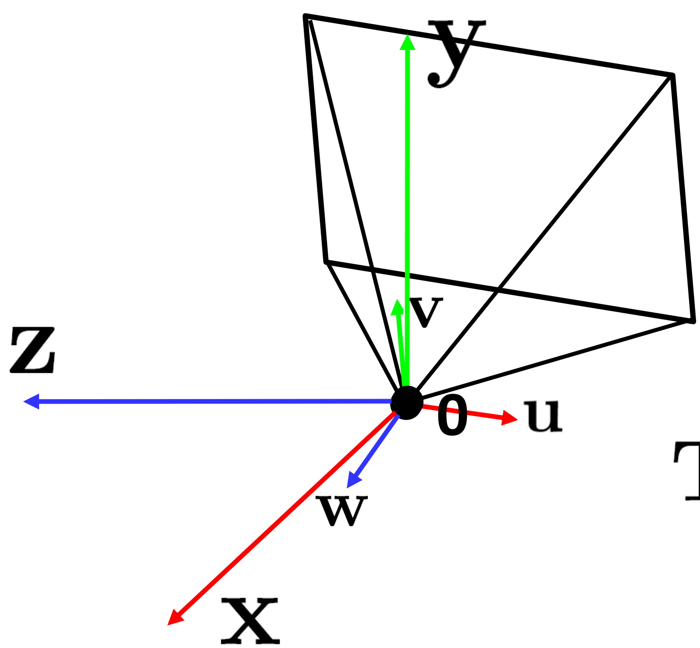
- How do we get the camera to “canonical form”?
  – (Center of projection at the origin, x-axis points right, y-axis points up, z-axis points backwards)

Step 1: Translate by -**c**



Extrinsics

- How do we get the camera to "canonical form"?
  - (Center of projection at the origin, x-axis points right, y-axis points up, z-axis points backwards)



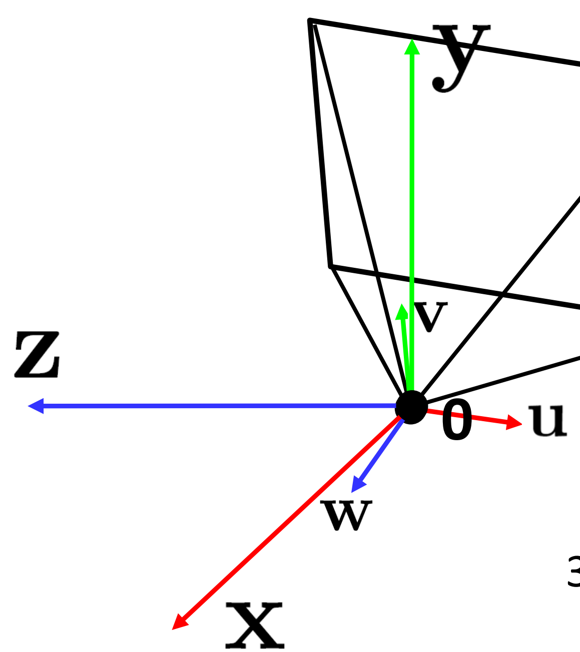
Step 1: Translate by -**c**

How do we represent translation as a matrix multiplication?

$$\mathbf{T} = \begin{bmatrix} \mathbf{I}_{3\times3} & -\mathbf{c} \\ 0 \quad 0 \quad 0 & 1 \end{bmatrix}$$

Extrinsics

- How do we get the camera to "canonical form"?
  – (Center of projection at the origin, x-axis points right, y-axis points up, z-axis points backwards)



Step 1: Translate by -**c**

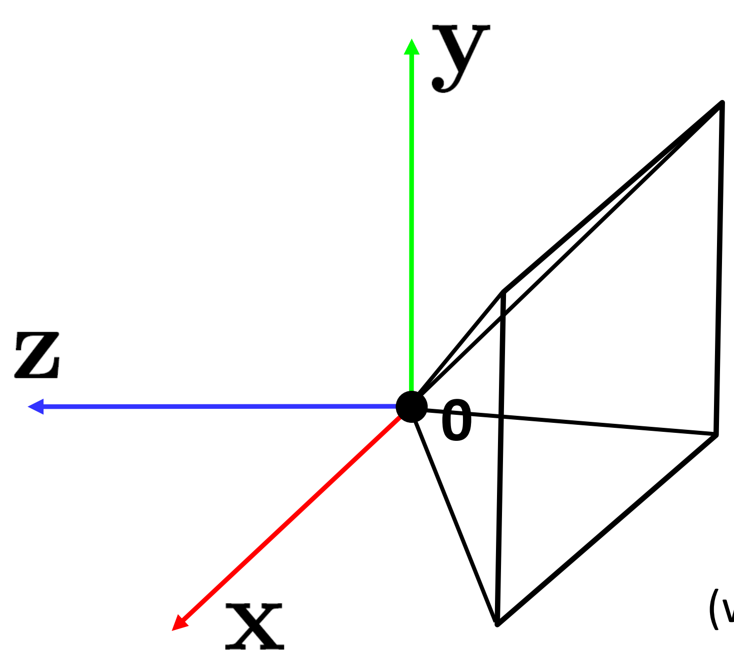
Step 2: Rotate by **R**

$$\mathbf{R} = \begin{bmatrix} \mathbf{u}^T \\ \mathbf{v}^T \\ \mathbf{w}^T \end{bmatrix}$$

3x3 rotation matrix

Extrinsics

- How do we get the camera to "canonical form"?
  - (Center of projection at the origin, x-axis points right, y-axis points up, z-axis points backwards)



Step 1: Translate by -**c**

Step 2: Rotate by **R**

$$\mathbf{R} = \begin{bmatrix} \mathbf{u}^T \\ \mathbf{v}^T \\ \mathbf{w}^T \end{bmatrix}$$

(with extra row/column of [0 0 0 1])

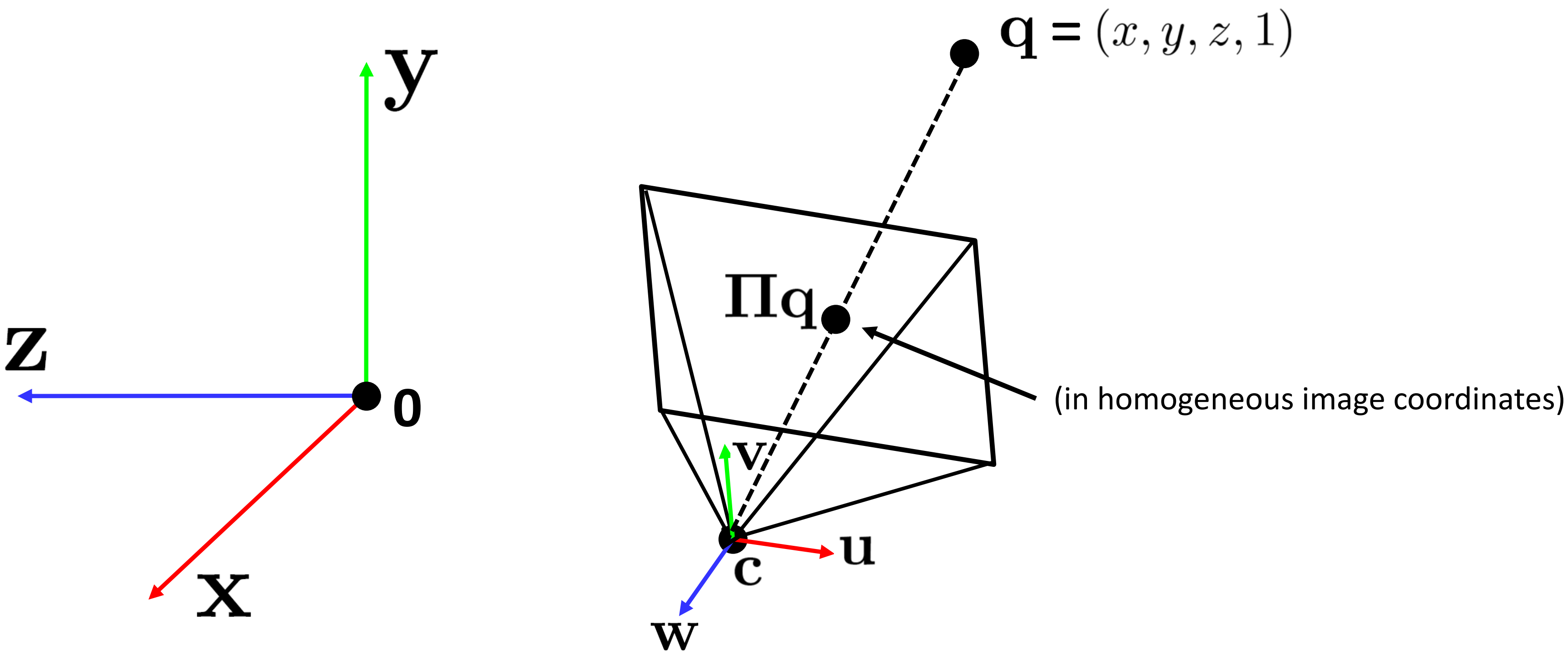
Projection matrix: Putting it all together

$$\mathbf{\Pi} = \underset{\text{intrinsics}}{\mathbf{K}} \underbrace{\underset{\text{projection}}{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}} \underset{\text{rotation}}{\begin{bmatrix} & \mathbf{R} & & \begin{matrix} 0 \\ 0 \\ 0 \end{matrix} \\ 0 & 0 & 0 & 1 \end{bmatrix}} \underset{\text{translation}}{\begin{bmatrix} & \mathbf{I}_{3\times3} & & -\mathbf{c} \\ 0 & 0 & 0 & 1 \end{bmatrix}}}$$

The **K** matrix converts 3D rays in the camera's coordinate system to 2D image points in image (pixel) coordinates.

This part converts 3D points in world coordinates to 3D rays in the camera's coordinate system. There are 6 parameters represented (3 for position/translation, 3 for rotation).

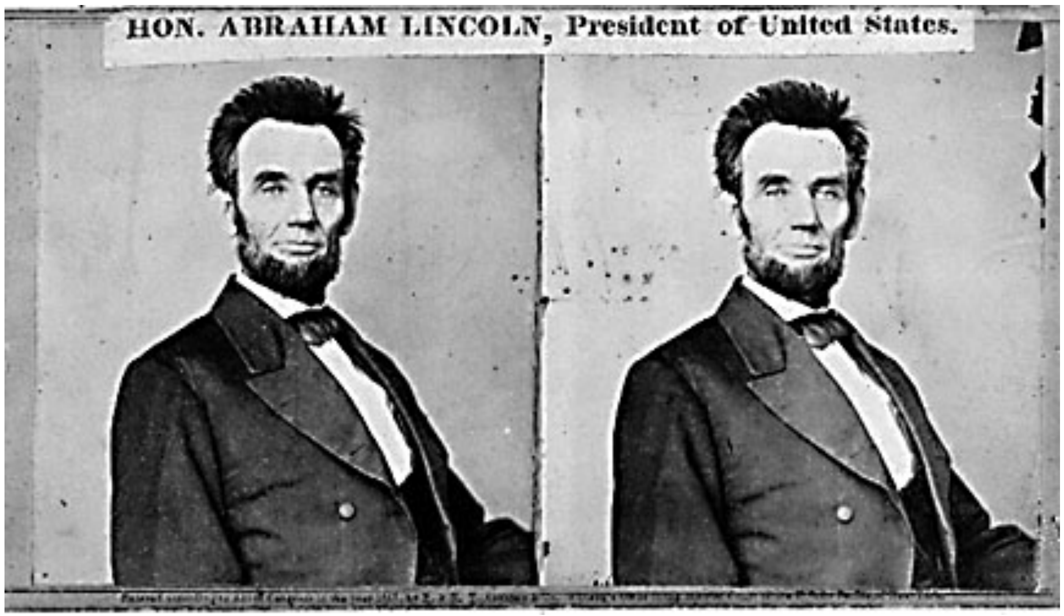
Projection matrix



What happens when cameras have different COPs?

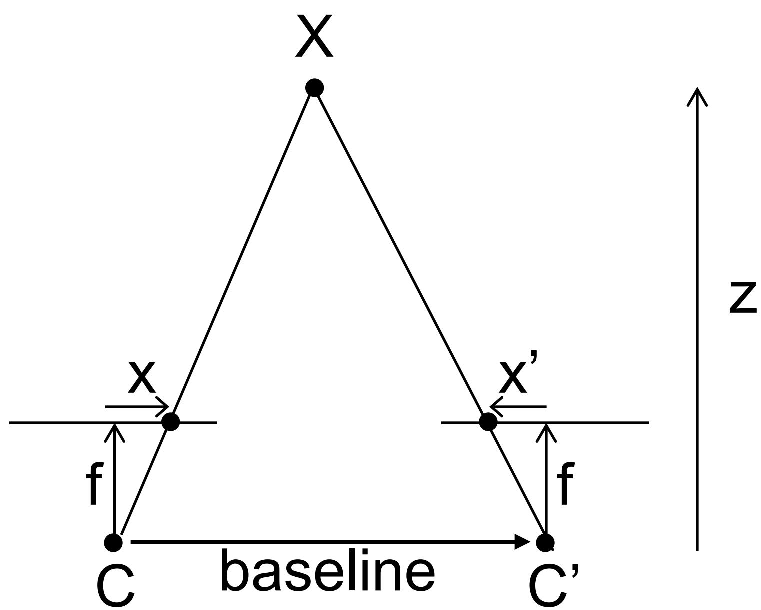
- (on board)

Stereo



- Given two images from different viewpoints
  - How can we compute the depth of each point in the image?
  - Based on *how much each pixel moves* between the two images

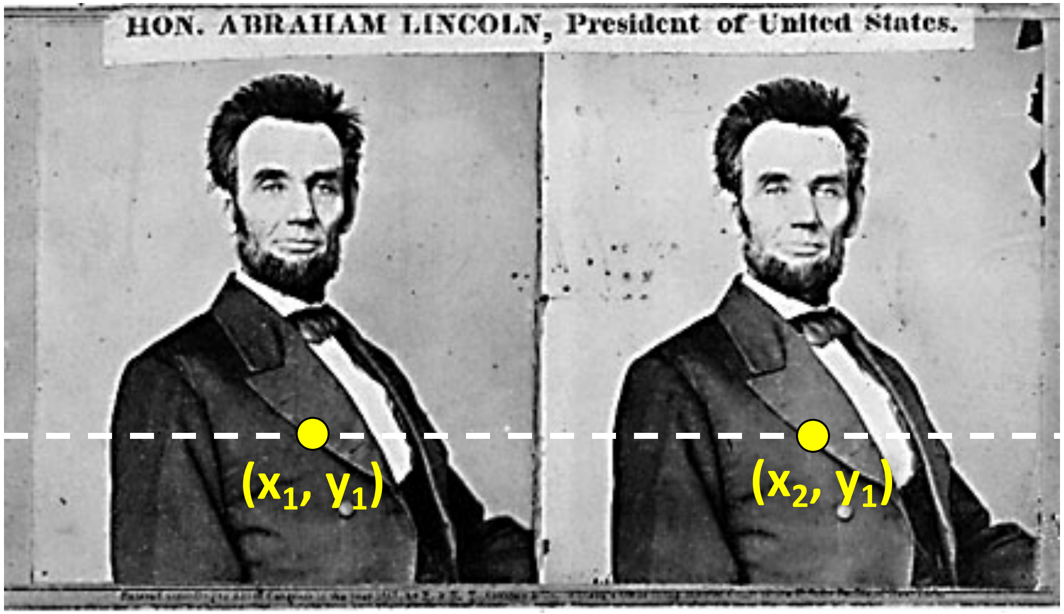
Depth from disparity



$$disparity = x - x' = \frac{baseline * f}{z}$$

Epipolar geometry

epipolar lines



Two images captured by a purely horizontal translating camera (*rectified* stereo pair)

$x_2 - x_1$ = the *disparity* of pixel (x_1, y_1)