

Image Formation and Cameras

Introduction to Computer Vision
CSE 152
Lecture 4

CSE152, Spr 2012

Intro Computer Vision

Announcements

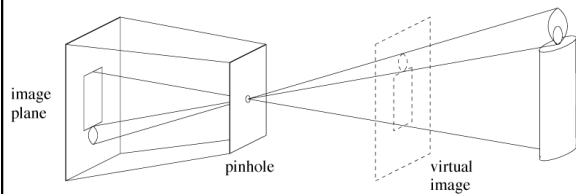
- Read Trucco & Verri: pp. 22-40
- Irfanview: <http://www.irfanview.com/> is a good Windows utility for manipulating images. Try xv for linux.
- Assignment 0: due today

CSE152, Spr 2012

Intro Computer Vision

Pinhole Camera: Perspective projection

- Abstract camera model - box with a small hole in it

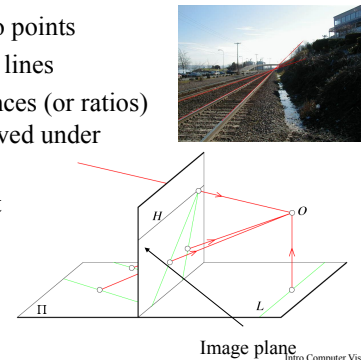


CSE152, Spr 2012 Forsyth&Ponce

Intro Computer Vision

Geometric Aspects of Perspective Projection

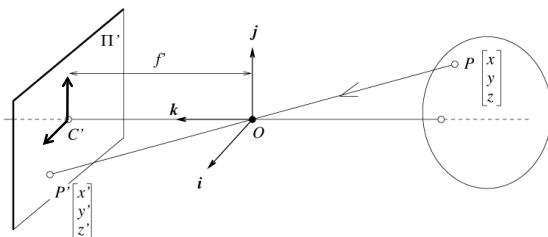
- Points project to points
- Lines project to lines
- Angles & distances (or ratios) are NOT preserved under perspective
- Vanishing point



CSE152, Spr 2012

Intro Computer Vision

Equation of Perspective Projection



Cartesian coordinates:

- We have, by similar triangles, that $(x, y, z) \rightarrow (f' x/z, f' y/z, f')$
- Establishing an image plane coordinate system at C' aligned with i and j , we get $(x, y, z) \rightarrow (f' \frac{x}{z}, f' \frac{y}{z})$

CSE152, Spr 2012

Intro Computer Vision

Euclidean $\rightarrow$ Homogenous $\rightarrow$ Euclidean

In 2-D

- Euclidean $\rightarrow$ Homogenous: $(x, y) \rightarrow \lambda (x, y, 1)$
(can just take $\lambda = 1$)
- Homogenous $\rightarrow$ Euclidean: $(x, y, z) \rightarrow (x/z, y/z)$

In 3-D

- Euclidean $\rightarrow$ Homogenous: $(x, y, z) \rightarrow \lambda (x, y, z, 1)$
(can just take $\lambda = 1$)
- Homogenous $\rightarrow$ Euclidean: $(x, y, z, w) \rightarrow (x/w, y/w, z/w)$

CSE152, Spr 2012

Intro Computer Vision

The camera matrix

Turn

$$(x, y, z) \rightarrow \left(f \frac{x}{z}, f \frac{y}{z}\right)$$

into homogenous
coordinates

- HC's for 3D point are $(X, Y, Z, 1)$
- HC's for point in image are (U, V, W)

$$\begin{pmatrix} U \\ V \\ W \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/f & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

CSE152, Spr 2012

Intro Computer Vision

Affine Camera Model

- Take Perspective projection equation, and perform Taylor Series Expansion about some point (x_0, y_0, z_0) .
- Drop terms of higher order than linear.
- Resulting expression is called affine camera model.

$$\begin{bmatrix} u \\ v \end{bmatrix} \approx \frac{1}{z_0} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + \begin{bmatrix} 1/z_0 & 0 & -x_0/z_0^2 \\ 0 & 1/z_0 & -y_0/z_0^2 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \mathbf{A}\mathbf{p} + \mathbf{b}$$

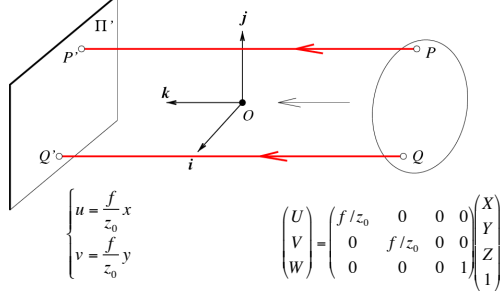
- Properties
 - Points map to points
 - Lines map to lines
 - Parallel lines map to parallel lines (no vanishing point – at infinity)
 - Ratios of distance/angles preserved

CSE152, Spr 2012

Intro Computer Vision

Orthographic projection

Start with affine camera model, and take Taylor series about $(x_0, y_0, z_0) = (0, 0, z_0)$ – a point on optical axis



Depth (z) is lost

CSE152, Spr 2012

Intro Computer Vision

Three projection models

- Perspective – all depths
- Affine – points near point of Taylor series expansion. E.g., when depth and size variation is small, and camera is far from point relative to size.
- Orthographic – when object is near optical axis, and depth variation of object is small compared to distance to the object.

CSE152, Spr 2012

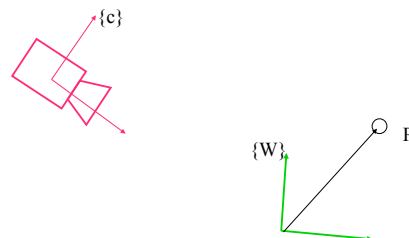
Intro Computer Vision

- Trombone
- <http://www.youtube.com/watch?hl=en&gl=US&client=mv-google&v=Yd9gaOU6zFY&rl=yes&feature=related>
- Trombone Stabilized
- <http://www.youtube.com/watch?hl=en&gl=US&client=mv-google&v=Yd9gaOU6zFY&rl=yes&feature=related>

CSE152, Spr 2012

Intro Computer Vision

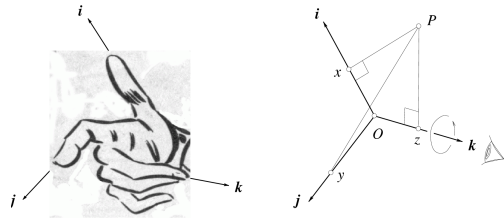
What if camera coordinate system differs from object coordinate system



CSE152, Spr 2012

Intro Computer Vision

Euclidean Coordinate Systems

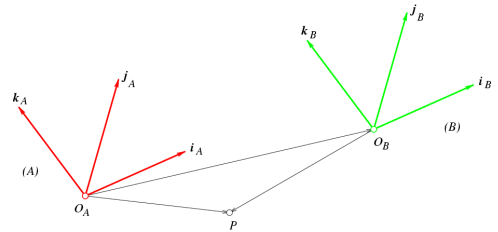


$$\begin{cases} x = \overrightarrow{OP} \cdot \mathbf{i} \\ y = \overrightarrow{OP} \cdot \mathbf{j} \\ z = \overrightarrow{OP} \cdot \mathbf{k} \end{cases} \Leftrightarrow \overrightarrow{OP} = x\mathbf{i} + y\mathbf{j} + z\mathbf{k} \Leftrightarrow \mathbf{P} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

CSE152, Spr 2012

Intro Computer Vision

Coordinate Changes: Pure Translations No rotation (e.g., $\mathbf{i}_A = \mathbf{i}_B$ etc)



$$\overrightarrow{O_B P} = \overrightarrow{O_B O_A} + \overrightarrow{O_A P}, \quad {}^B P = {}^A P + {}^B O_A$$

CSE152, Spr 2012

Intro Computer Vision

A convenient notation

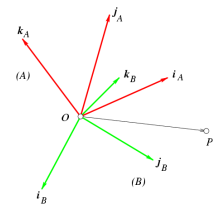
$${}^B P = {}^B R {}^A P + {}^B O_A$$

- Points: ${}^A P_i$
 - Leading superscript indicates the coordinate system that the coordinates are with respect to
 - Subscript – an identifier
- Rotation Matrices ${}^B R_A$
 - Lower left (Going from this system)
 - Upper left (Going to this system)
- To add vectors, coordinate systems (leading superscript) must agree
- To rotate a vector, points coordinate system must agree with lower left of rotation matrix

CS252A, Fall 2011

Computer Vision I

Coordinate Changes: Pure Rotations



$$\overrightarrow{OP} = \begin{bmatrix} \mathbf{i}_A & \mathbf{j}_A & \mathbf{k}_A \end{bmatrix} \begin{bmatrix} {}^A x \\ {}^A y \\ {}^A z \end{bmatrix} = \begin{bmatrix} \mathbf{i}_B & \mathbf{j}_B & \mathbf{k}_B \end{bmatrix} \begin{bmatrix} {}^B x \\ {}^B y \\ {}^B z \end{bmatrix}$$

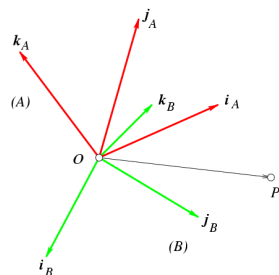
$$\begin{bmatrix} {}^B x \\ {}^B y \\ {}^B z \end{bmatrix} = \begin{bmatrix} \mathbf{i}_B & \mathbf{j}_B & \mathbf{k}_B \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{i}_A & \mathbf{j}_A & \mathbf{k}_A \end{bmatrix} \begin{bmatrix} {}^A x \\ {}^A y \\ {}^A z \end{bmatrix} = \begin{bmatrix} \mathbf{i}_B & \mathbf{j}_B & \mathbf{k}_B \end{bmatrix}^T \begin{bmatrix} \mathbf{i}_A & \mathbf{j}_A & \mathbf{k}_A \end{bmatrix} \begin{bmatrix} {}^A x \\ {}^A y \\ {}^A z \end{bmatrix}$$

$$\Rightarrow {}^B P = {}^B R {}^A P$$

CSE152, Spr 2012

Intro Computer Vision

Rotation Matrix

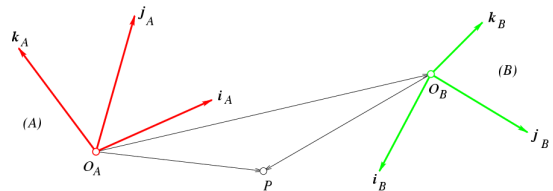


$${}^B R_A = \begin{bmatrix} \mathbf{i}_A \cdot \mathbf{i}_B & \mathbf{j}_A \cdot \mathbf{i}_B & \mathbf{k}_A \cdot \mathbf{i}_B \\ \mathbf{i}_A \cdot \mathbf{j}_B & \mathbf{j}_A \cdot \mathbf{j}_B & \mathbf{k}_A \cdot \mathbf{j}_B \\ \mathbf{i}_A \cdot \mathbf{k}_B & \mathbf{j}_A \cdot \mathbf{k}_B & \mathbf{k}_A \cdot \mathbf{k}_B \end{bmatrix} = \begin{bmatrix} {}^A \mathbf{i}_B^T \\ {}^A \mathbf{j}_B^T \\ {}^A \mathbf{k}_B^T \end{bmatrix} = \begin{bmatrix} {}^B \mathbf{i}_A & {}^B \mathbf{j}_A & {}^B \mathbf{k}_A \end{bmatrix}$$

CSE152, Spr 2012

Intro Computer Vision

Coordinate Changes: Rigid Transformations Rotation + Translation



$${}^B P = {}^B R {}^A P + {}^B O_A$$

CSE152, Spr 2012

Intro Computer Vision

More about rotations matrices

A rotation matrix R has the following properties:

- Its inverse is equal to its transpose $R^{-1} = R^T$ or $R^T R = I$
- Its determinant is equal to 1: $\det(R) = 1$.

Or equivalently:

- Rows (or columns) of R form a right-handed orthonormal coordinate system.

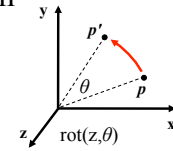
Even though a rotation matrix is 3x3 with nine numbers, it only has three degrees of freedom can be parameterized with three numbers. There are many parameterization.

CSE152, Spr 2012

Intro Computer Vision

Rotation

- About z axis



$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Note: z coordinate doesn't change after rotation

CSE152, Spr 2012

Intro Computer Vision

Rotation

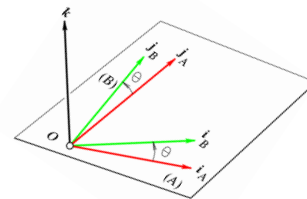
- About x axis: $\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$
- About y axis: $\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$

CSE152, Spr 2012

Intro Computer Vision

Roll-Pitch-Yaw

$$R = \text{rot}(\hat{i}, \alpha) \text{rot}(\hat{j}, \beta) \text{rot}(\hat{k}, \varphi)$$



Euler Angles

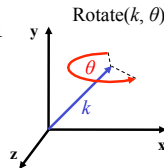
$$R = \text{rot}(\hat{k}'', \alpha) \text{rot}(\hat{j}', \beta) \text{rot}(\hat{k}, \varphi)$$

CSE152, Spr 2012

Intro Computer Vision

Rotation

- Rotation by angle θ about (k_x, k_y, k_z) , a unit vector
- (Rodrigues Formula)



$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} k_x k_x (1-c) + c & k_x k_y (1-c) - k_z s & k_x k_z (1-c) + k_y s \\ k_y k_x (1-c) + k_z s & k_y k_y (1-c) + c & k_y k_z (1-c) - k_x s \\ k_z k_x (1-c) - k_y s & k_z k_y (1-c) - k_x s & k_z k_z (1-c) + c \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

where $c = \cos \theta$ & $s = \sin \theta$

CSE152, Spr 2012

Intro Computer Vision

Homogeneous Representation of Rigid Transformations

$$\begin{bmatrix} {}^B P \\ 1 \end{bmatrix} = \begin{bmatrix} {}^B R & {}^B O_A \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} {}^A P \\ 1 \end{bmatrix} = \begin{bmatrix} {}^B R & {}^B O_A \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} {}^A P \\ 1 \end{bmatrix} = {}^B T_A \begin{bmatrix} {}^A P \\ 1 \end{bmatrix}$$

Transformation represented by 4 by 4 Matrix

Block Matrix Multiplication

Given

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \quad B = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}$$

What is AB ?

$$AB = \begin{bmatrix} A_{11}B_{11} + A_{12}B_{21} & A_{11}B_{12} + A_{12}B_{22} \\ A_{21}B_{11} + A_{22}B_{21} & A_{21}B_{12} + A_{22}B_{22} \end{bmatrix}$$

CSE152, Spr 2012

Intro Computer Vision

3x3 homogenous matrix

Focal length:

Principal Point: C'

Units (e.g. pixels)

Orientation and position of image coordinate system

Pixel Aspect ratio

$$\begin{pmatrix} U \\ V \\ W \end{pmatrix} = \begin{pmatrix} \text{Transformation} \\ \text{represented by} \\ \text{intrinsic parameters} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \text{Rigid Transformation} \\ \text{represented by} \\ \text{extrinsic parameters} \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ T \end{pmatrix}$$

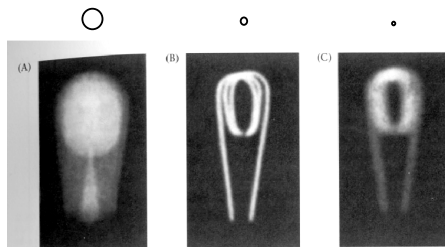
3×3 4×4

CSE152, Spr 2012 Intro Computer Vision

CSE152, Spr 2012 Intro Computer Vision

CSE152, Spr 2012 Intro Computer Vision

Limits for pinhole cameras

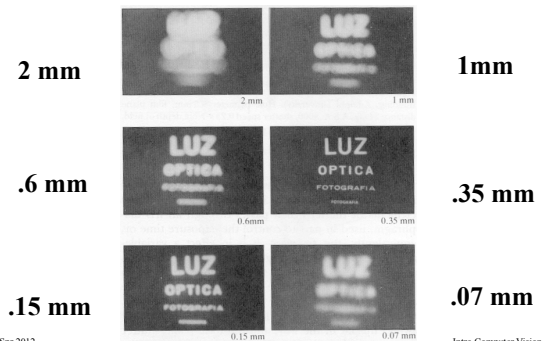


2.18 DIFFRACTION LIMITS THE QUALITY OF PINHOLE OPTICS. These three images of a bulb filament were made using pinholes with decreasing size. (A) When the pinhole is relatively large, the image rays are not properly converged, and the image is blurred. (B) Reducing the size of the pinhole improves the focus. (C) Reducing the size of the pinhole further worsens the focus, due to diffraction. From Ruechardt, 1958.

CSE152, Spr 2012

Intro Computer Vision

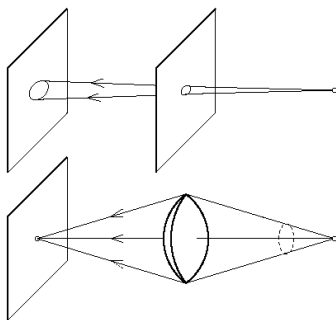
Pinhole Camera Images with Variable Aperture



CSE152, Spr 2012

Intro Computer Vision

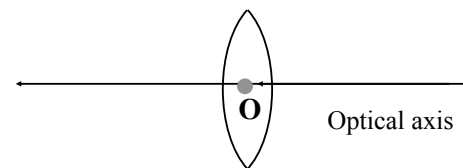
The reason for lenses



CSE152, Spr 2012

Intro Computer Vision

Thin Lens

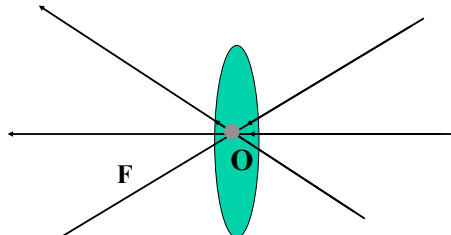


- Rotationally symmetric about optical axis.
- Spherical interfaces.

CSE152, Spr 2012

Intro Computer Vision

Thin Lens: Center

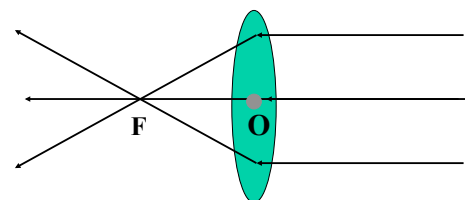


- All rays that enter lens along line pointing at **O** emerge in same direction.

CSE152, Spr 2012

Intro Computer Vision

Thin Lens: Focus



Incoming light rays parallel to the optical axis pass through the focus, **F**

CSE152, Spr 2012

Intro Computer Vision

