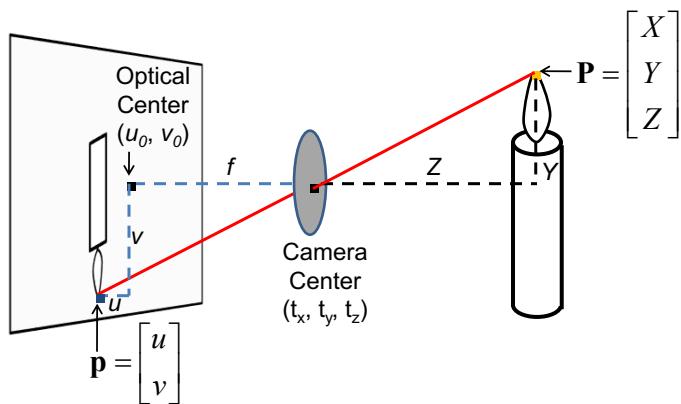


Geometric Camera Models

- Elements of Euclidean Geometry
- The Intrinsic Parameters of a Camera
- The Extrinsic Parameters of a Camera
- The General Form of the Perspective Projection Equation

Projection: world coordinates \rightarrow image coordinates



How do we handle the general case?

Homogeneous coordinates

Conversion

Converting to *homogeneous* coordinates

$$(x, y) \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

homogeneous image coordinates

$$(x, y, z) \Rightarrow \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

homogeneous scene
coordinates

Converting *from* homogeneous coordinates

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} \Rightarrow (x/w, y/w)$$

$$\begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \Rightarrow (x/w, y/w, z/w)$$

Homogeneous coordinates

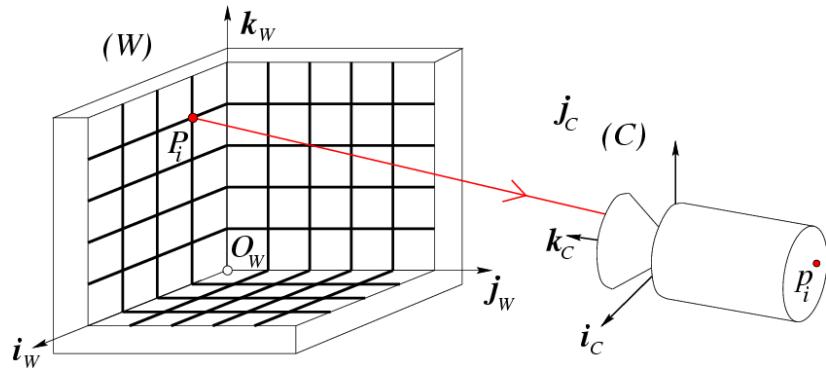
Invariant to scaling

$$k \begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} kx \\ ky \\ kw \end{bmatrix} \Rightarrow \begin{bmatrix} \frac{kx}{kw} \\ \frac{ky}{kw} \end{bmatrix} = \begin{bmatrix} \frac{x}{w} \\ \frac{y}{w} \end{bmatrix}$$

Homogeneous Coordinates Cartesian Coordinates

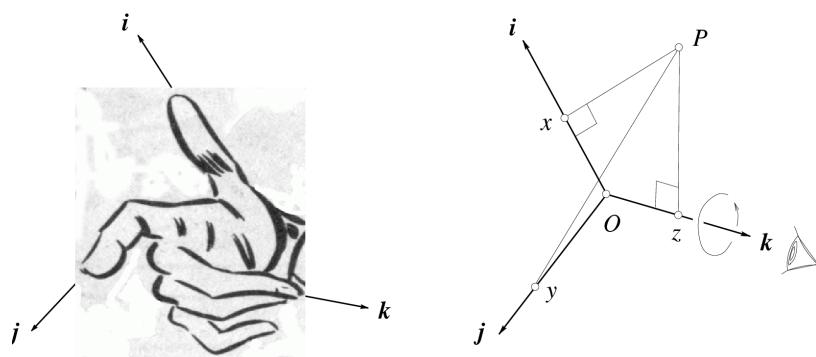
Point in Cartesian is ray in Homogeneous

Quantitative Measurements and Calibration



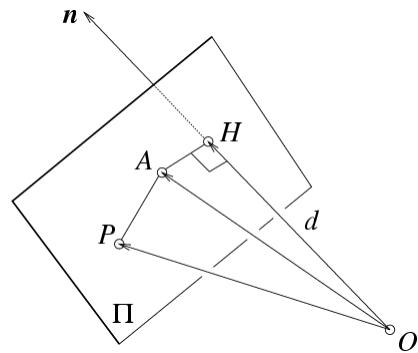
Euclidean Geometry

Euclidean Coordinate Systems



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

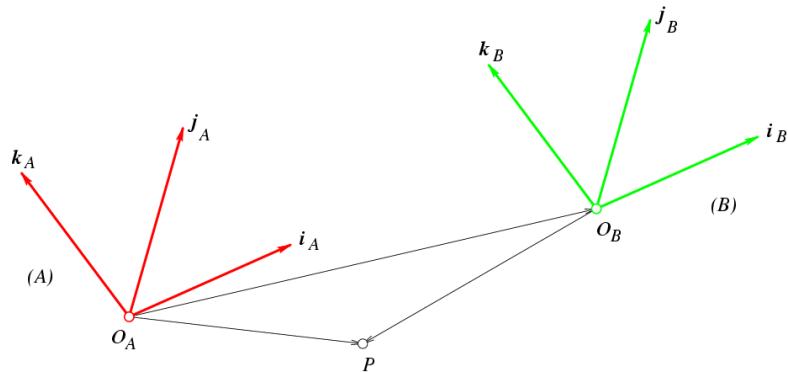
Planes



$$\overrightarrow{AP} \cdot \mathbf{n} = 0 \Leftrightarrow ax + by + cz - d = 0 \Leftrightarrow \mathbf{\Pi} \cdot \mathbf{P} = 0$$

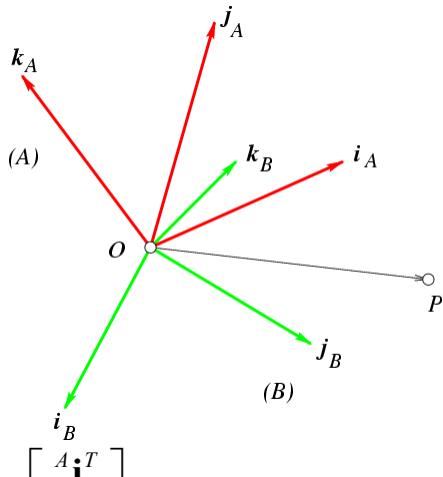
where $\mathbf{\Pi} = \begin{bmatrix} a \\ b \\ c \\ -d \end{bmatrix}$ and $\mathbf{P} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$

Coordinate Changes: Pure Translations



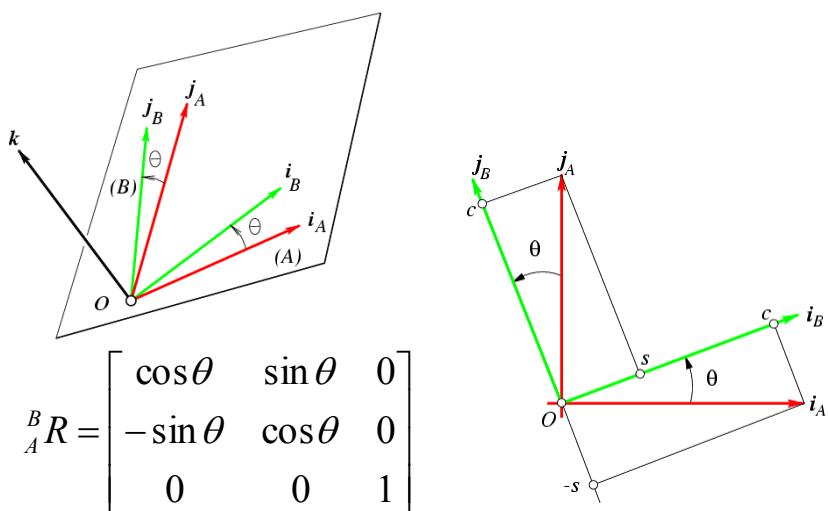
$$\overrightarrow{O_B P} = \overrightarrow{O_B O_A} + \overrightarrow{O_A P}, \quad {}^B P = {}^A P + {}^B O_A$$

Coordinate Changes: Pure Rotations



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

Coordinate Changes: Rotations about the z Axis



$${}^B{}_A R = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

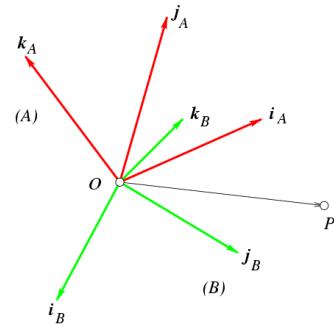
Rotation Matrix Properties

- Its inverse is equal to its transpose, and
- its determinant is equal to 1.

Or equivalently:

- Its rows (or columns) form a right-handed
- orthonormal coordinate system.

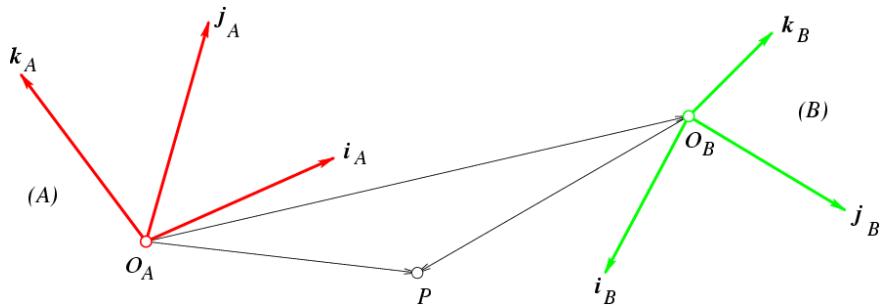
Coordinate Changes: Pure Rotations



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

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

Coordinate Changes: Rigid Transformations



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

Block Matrix Multiplication

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

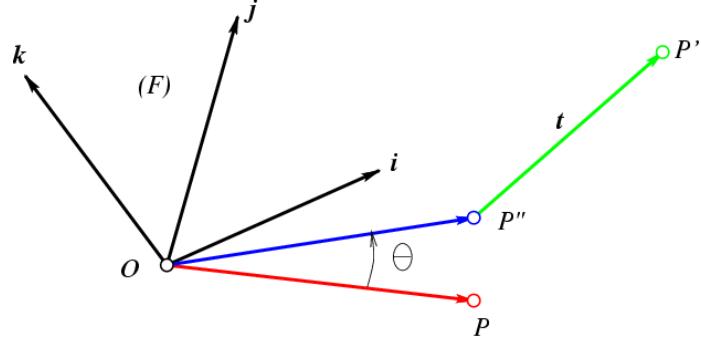
What is AB?

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

Homogeneous Representation of Rigid Transformations

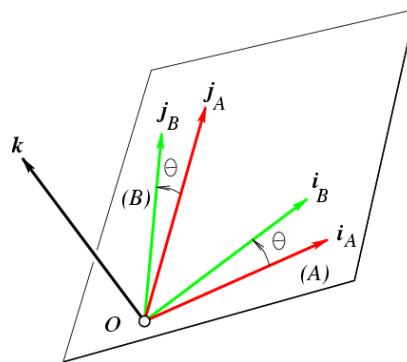
$$\begin{bmatrix} {}^B P \\ 1 \end{bmatrix} = \begin{bmatrix} {}^B R & {}^B O_A \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} {}^A P \\ 1 \end{bmatrix} = \begin{bmatrix} {}^B R {}^A P + {}^B O_A \\ 1 \end{bmatrix} = {}^B T \begin{bmatrix} {}^A P \\ 1 \end{bmatrix}$$

Rigid Transformations as Mappings



$${}^F P' = \mathcal{R} {}^F P + \mathbf{t} \iff \begin{pmatrix} {}^F P' \\ 1 \end{pmatrix} = \begin{pmatrix} \mathcal{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{pmatrix} \begin{pmatrix} {}^F P \\ 1 \end{pmatrix}$$

Rigid Transformations as Mappings: Rotation about the *k* Axis



$${}^F P' = \mathcal{R} {}^F P, \quad \text{where} \quad \mathcal{R} = \begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Basic geometry in homogeneous coordinates

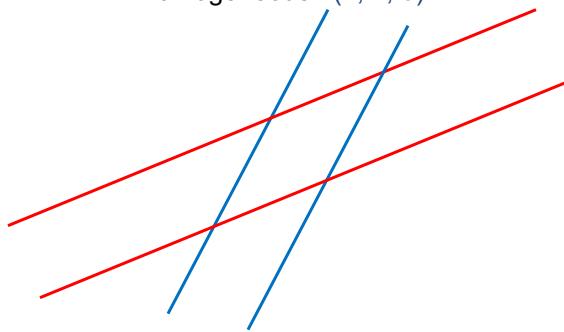
- Line equation: $ax + by + c = 0$ $line_i = \begin{bmatrix} a_i \\ b_i \\ c_i \end{bmatrix}$
- Append 1 to pixel coordinate to get homogeneous coordinate $p_i = \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix}$
- Line given by cross product of two points $line_{ij} = p_i \times p_j$
- Intersection of two lines given by cross product of the lines $q_{ij} = line_i \times line_j$

Another problem solved by homogeneous coordinates

Intersection of parallel lines

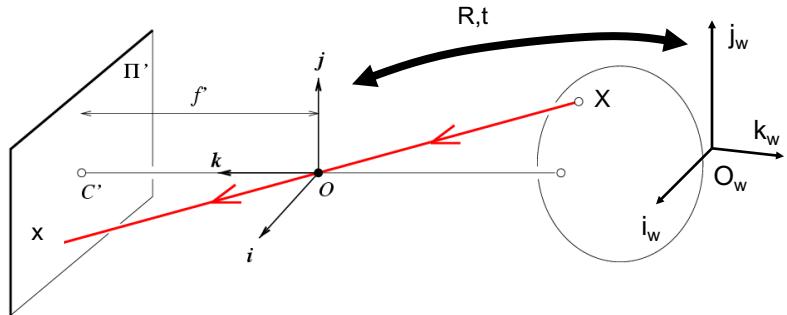
Cartesian: (Inf, Inf)
Homogeneous: (1, 1, 0)

Cartesian: (Inf, Inf)
Homogeneous: (1, 2, 0)



Projection matrix

Slide Credit: Savarese



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

\mathbf{x} : Image Coordinates: $(u, v, 1)$

\mathbf{K} : Intrinsic Matrix (3x3)

\mathbf{R} : Rotation (3x3)

\mathbf{t} : Translation (3x1)

\mathbf{X} : World Coordinates: $(X, Y, Z, 1)$

Object Recognition (CVPR 2006)



Inserting photographed objects into images (SIGGRAPH 2007)



Original



Created

Pinhole Camera Model

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

\mathbf{x} : Image Coordinates: $(u, v, 1)$

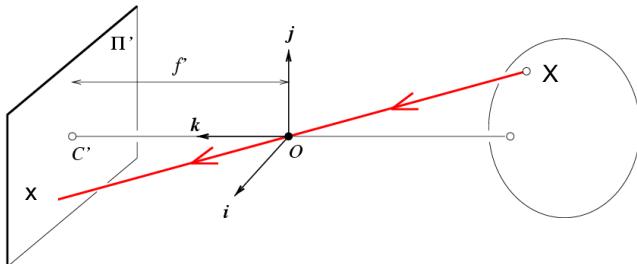
\mathbf{K} : Intrinsic Matrix (3x3)

\mathbf{R} : Rotation (3x3)

\mathbf{t} : Translation (3x1)

\mathbf{X} : World Coordinates: $(X, Y, Z, 1)$

Projection matrix



Intrinsic Assumptions

- Unit aspect ratio
- Optical center at (0,0)
- No skew

Extrinsic Assumptions

- No rotation
- Camera at (0,0,0)

$$\mathbf{x} = \mathbf{K} [\mathbf{I} \quad \mathbf{0}] \mathbf{X} \rightarrow w \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

\mathbf{K}

Slide Credit: Savarese

Remove assumption: known optical center

Intrinsic Assumptions

- Unit aspect ratio
- No skew

Extrinsic Assumptions

- No rotation
- Camera at (0,0,0)

$$\mathbf{x} = \mathbf{K} [\mathbf{I} \quad \mathbf{0}] \mathbf{X} \rightarrow w \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & u_0 & 0 \\ 0 & f & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Remove assumption: square pixels

Intrinsic Assumptions Extrinsic Assumptions

- No skew
- No rotation
- Camera at (0,0,0)

$$\mathbf{x} = \mathbf{K}[\mathbf{I} \quad \mathbf{0}] \mathbf{X} \rightarrow w \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha & 0 & u_0 & 0 \\ 0 & \beta & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Remove assumption: non-skewed pixels

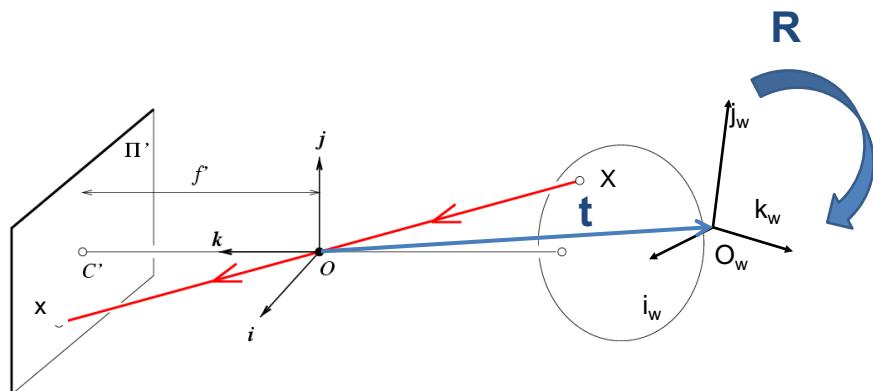
Intrinsic Assumptions Extrinsic Assumptions

- No rotation
- Camera at (0,0,0)

$$\mathbf{x} = \mathbf{K}[\mathbf{I} \quad \mathbf{0}] \mathbf{X} \rightarrow w \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha & s & u_0 & 0 \\ 0 & \beta & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Note: different books use different notation for parameters

Oriented and Translated Camera



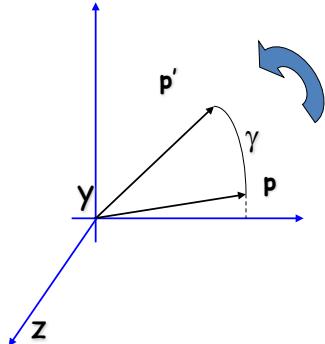
Allow camera translation

Intrinsic Assumptions Extrinsic Assumptions
• No rotation

$$\mathbf{x} = \mathbf{K}[\mathbf{I} \quad \mathbf{t}] \mathbf{X} \rightarrow w \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha & 0 & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

3D Rotation of Points

Rotation around the coordinate axes, counter-clockwise:



$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{bmatrix}$$

$$R_y(\beta) = \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{bmatrix}$$

$$R_z(\gamma) = \begin{bmatrix} \cos \gamma & -\sin \gamma & 0 \\ \sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Allow camera rotation

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



$$w \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha & s & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Degrees of freedom

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



$$w \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha & s & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

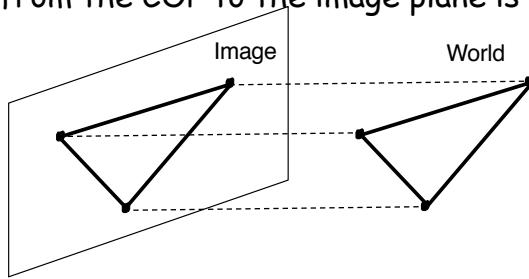
Vanishing Point = Projection from Infinity

$$\mathbf{p} = \mathbf{K}[\mathbf{R} \quad \mathbf{t}] \begin{bmatrix} x \\ y \\ z \\ 0 \end{bmatrix} \Rightarrow \mathbf{p} = \mathbf{K}\mathbf{R} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \Rightarrow \mathbf{p} = \mathbf{K} \begin{bmatrix} x_R \\ y_R \\ z_R \end{bmatrix}$$

$$w \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_R \\ y_R \\ z_R \end{bmatrix} \Rightarrow \begin{aligned} u &= \frac{fx_R}{z_R} + u_0 \\ v &= \frac{fy_R}{z_R} + v_0 \end{aligned}$$

Orthographic Projection

- Special case of perspective projection
 - Distance from the COP to the image plane is infinite



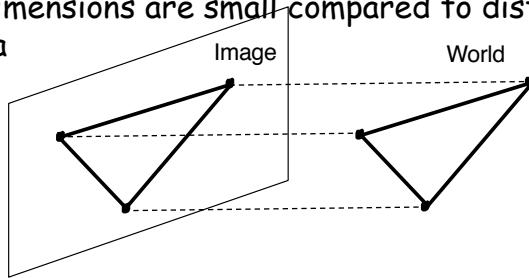
- Also called "parallel projection"
- What's the projection matrix?

$$w \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Slide by Steve Seitz

Scaled Orthographic Projection

- Special case of perspective projection
 - Object dimensions are small compared to distance to camera



- Also called "weak perspective"
- What's the projection matrix?

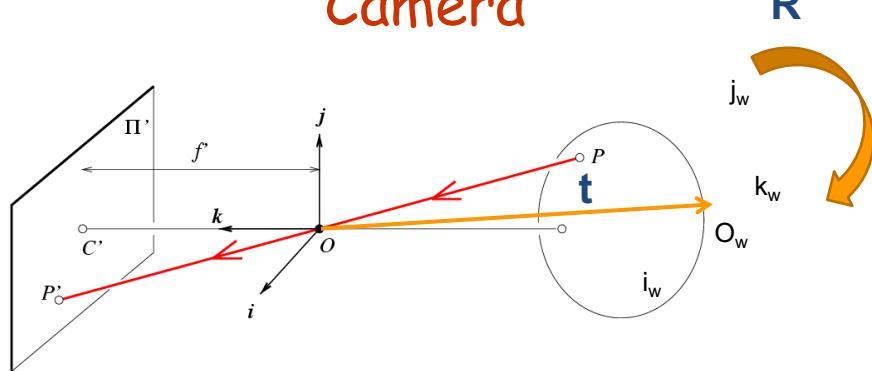
$$w \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 0 & s \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Slide by Steve Seitz

Summary of camera parameters

- Extrinsic
 - Translation vector T
 - Rotation Matrix R
- Intrinsic
 - Focal length
 - Image center
 - Pixel size
 - Radial distortion coefficients

Oriented and Translated Camera



Degrees of freedom

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



$$w \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} 5 & & \\ \alpha & s & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 6 & & \\ r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

How to calibrate the camera?

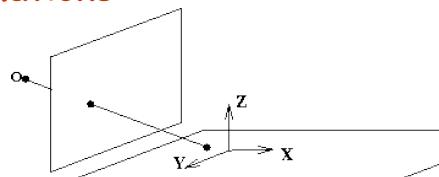
$$\mathbf{x} = \mathbf{K}[\mathbf{R} \quad \mathbf{t}] \mathbf{X}$$

$$\begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Camera calibration

- Issues:
 - what are intrinsic parameters of the camera?
 - what is the camera matrix? (intrinsic+extrinsic)
- General strategy:
 - view calibration object
 - identify image points
 - obtain camera matrix by minimizing error
 - obtain intrinsic parameters from camera matrix
- Error minimization:
 - Linear least squares
 - easy problem numerically
 - solution can be rather bad
 - Minimize image distance
 - more difficult numerical problem
 - solution usually rather good,
 - start with linear least squares
 - Numerical scaling is an issue

Imaging Geometry: Plane projective transformations



$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{12} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{14} \\ p_{21} & p_{12} & p_{24} \\ p_{31} & p_{32} & p_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

Image Geometry: Computing Plane Projective Transform 1

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \sim \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{12} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

or, equivalently:

$$\begin{bmatrix} x' \\ y' \\ \lambda \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{12} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Image Geometry: Computing Plane Projective Transform 2

$$x' = \frac{h_{11}x + h_{12}x + h_{13}}{h_{31}x + h_{32}y + h_{33}}, y' = \frac{h_{21}x + h_{22}x + h_{23}}{h_{31}x + h_{32}y + h_{33}}$$

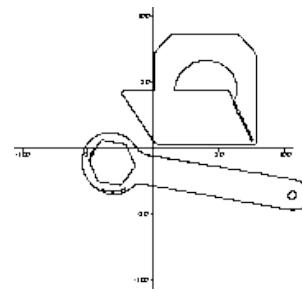
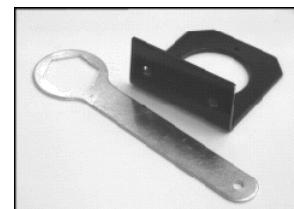
$$(h_{31}x + h_{32}y + h_{33})x' = h_{11}x + h_{12}x + h_{13}$$
$$(h_{31}x + h_{32}y + h_{33})y' = h_{21}x + h_{22}x + h_{23}$$

Image Geometry: Computing Plane Projective Transform 3

$$\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x'_1x_1 & -x'_1y_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -y'_1x_1 & -y'_1y_1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x'_2x_2 & -x'_2y_2 \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -y'_2x_2 & -y'_2y_2 \\ x_3 & y_3 & 1 & 0 & 0 & 0 & -x'_3x_3 & -x'_3y_3 \\ 0 & 0 & 0 & x_3 & y_3 & 1 & -y'_3x_3 & -y'_3y_3 \\ x_4 & y_4 & 1 & 0 & 0 & 0 & -x'_4x_4 & -x'_4y_4 \\ 0 & 0 & 0 & x_4 & y_4 & 1 & -y'_4x_4 & -y'_4y_4 \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \end{bmatrix} = \begin{bmatrix} x'_1 \\ y'_1 \\ x'_2 \\ y'_2 \\ x'_3 \\ y'_3 \\ x'_4 \\ y'_4 \end{bmatrix}$$

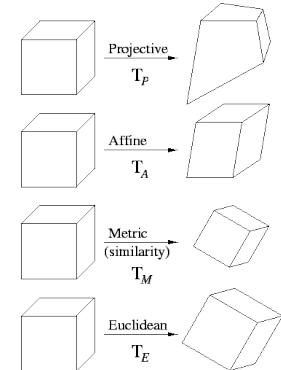
Removing Perspective Distortion

1. Have coordinates of four points on the object plane
2. Solve for H in $x' = Hx$ from the and corresponding image coordinates.
3. Then $x = H^{-1}x'$
4. (E.g.) inspect the part, checking distances or angle



Taxonomy of planar projective

group	matrix	properties
perspective 8 DOF	$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{12} & h_{22} & h_{23} \\ h_{13} & h_{23} & h_{33} \end{bmatrix}$	concurrency, collinearity, incidence, tangency, inflection
affine 6 DOF	$\begin{bmatrix} a_{11} & a_{12} & t_{13} \\ a_{12} & a_{22} & t_{23} \\ 0 & 0 & 1 \end{bmatrix}$	parallelism, ratio of areas, ratio of lengths on collinear or parallel segments
similarity 4 DOF	$\begin{bmatrix} sr_{11} & sr_{12} & t_{13} \\ sr_{12} & sr_{22} & t_{23} \\ 0 & 0 & 1 \end{bmatrix}$	angles, ratio of lengths
Euclidean 3 DOF	$\begin{bmatrix} r_{11} & r_{12} & t_{13} \\ r_{12} & r_{22} & t_{23} \\ 0 & 0 & 1 \end{bmatrix}$	length, area



Notes:

- Properties of the more general transforms are inherited by transformations lower in the table

$$\cdot R = [r_{ij}] \text{ is a rotation matrix, i.e. } R^T R = I, \text{ also } R = \begin{bmatrix} \sin(\theta) & \cos(\theta) \\ -\cos(\theta) & \sin(\theta) \end{bmatrix}$$

Taxonomy of planar projective transforms I

- In many circumstances, we know from the imaging set-up, that the image-to-image transformation is simpler than homography or can be well approximated by a transformation with a lower number of degrees of freedom.
- Three types of transforms are commonly encountered:
 - *Euclidean* (shifted and rotated, e.g. two flatbed scans of the same image)
 - *Similarity* (shift, rotation, isotropic scaling, e.g. two photos from the same spot with different zoom)
 - *Affine transformation*