CE7453: Photogrammetric Computer Vision

Lecture 5

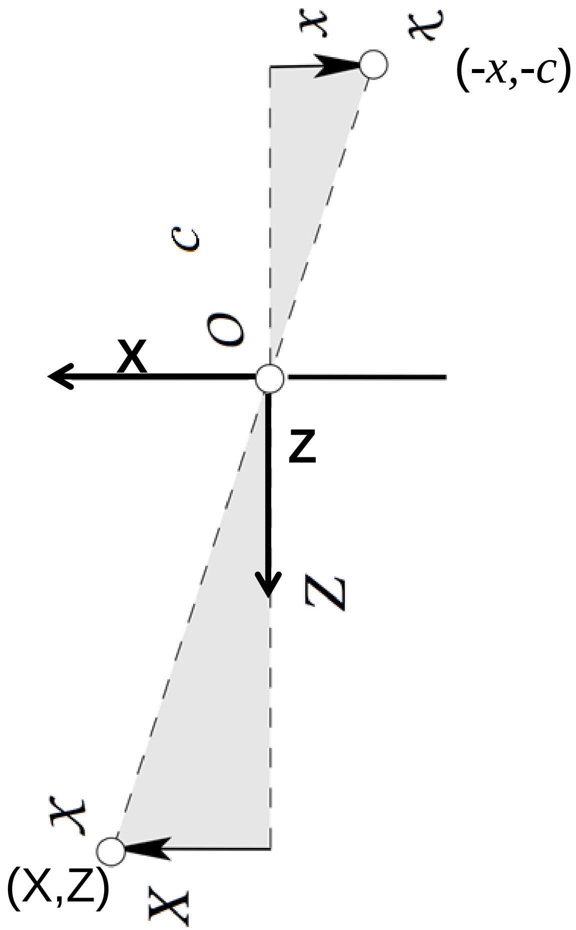
Central Perspective Model
Homogenous Coordinate System
Camera Matrix, Projection Matrix

Acknowledgements: Ping Tan, Yung-Yu Chuang. part of the materials of the all the lecture notes are from Cyrill Stachniss, Marc Pollefey, Wolfgang Foerstner, Bernhard Wrobel, James Hays, A. Dermanis, Armin Gruen, Alper Yilmaz.

Central Perspective Model

$$\frac{c}{Z} = \frac{x}{X} = m$$

$$x = \frac{c}{Z} X = mX$$



c : camera constant

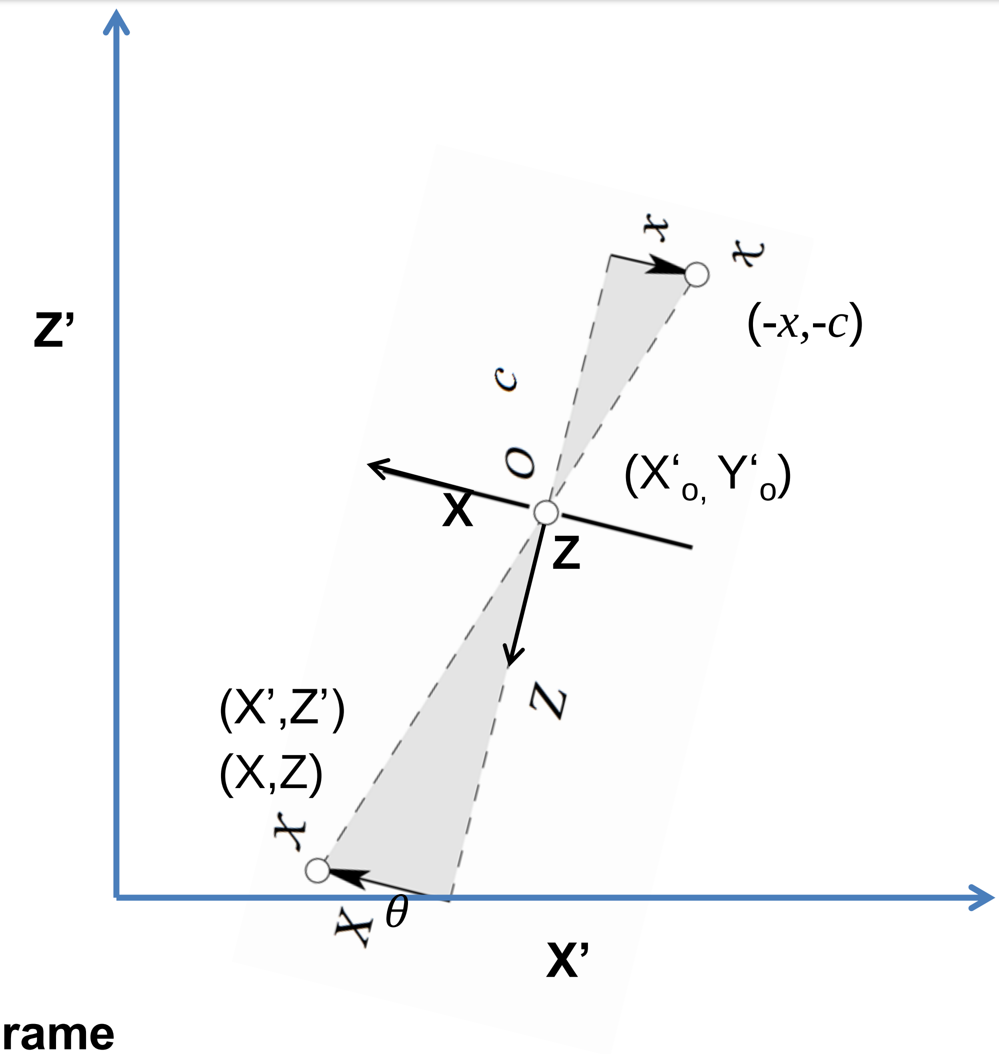
m : map scale

Central Perspective Model

$$\begin{bmatrix} X \\ Z \end{bmatrix} = R \begin{bmatrix} X' - X'_o \\ Z' - Z'_o \end{bmatrix}$$

$$R = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix}$$

$$x = \frac{c}{Z} X$$



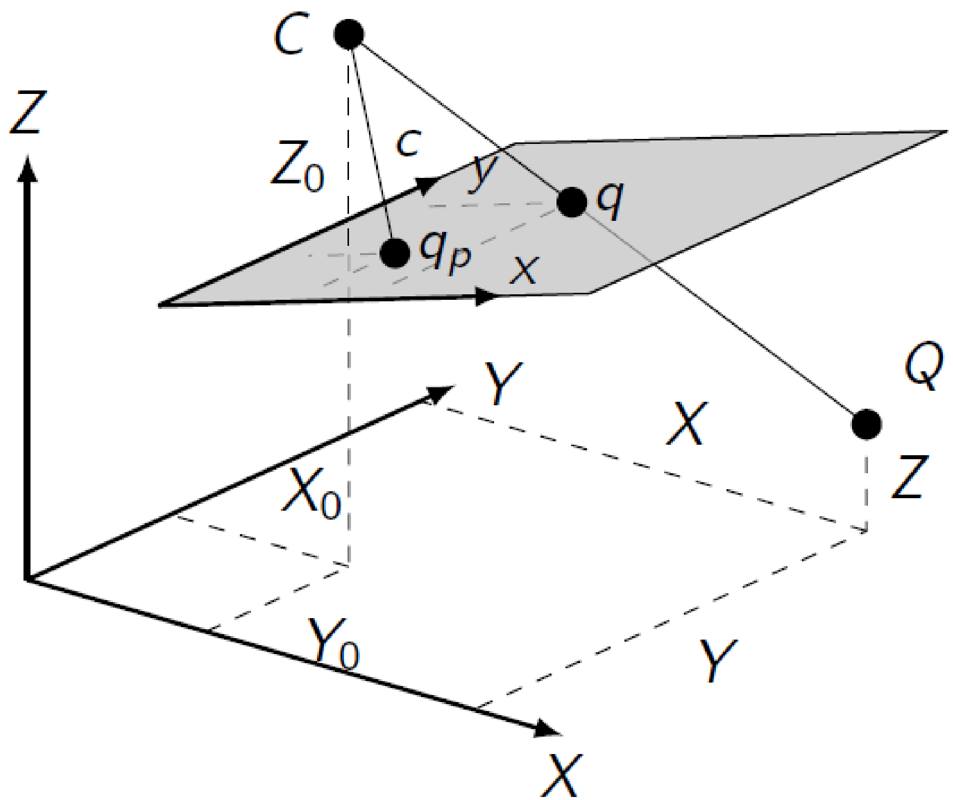
X, Z: Camera Frame
X',Y': World Frame / Geodetic Frame

Co-linearity Equation – Cont.

The *collinearity equations*

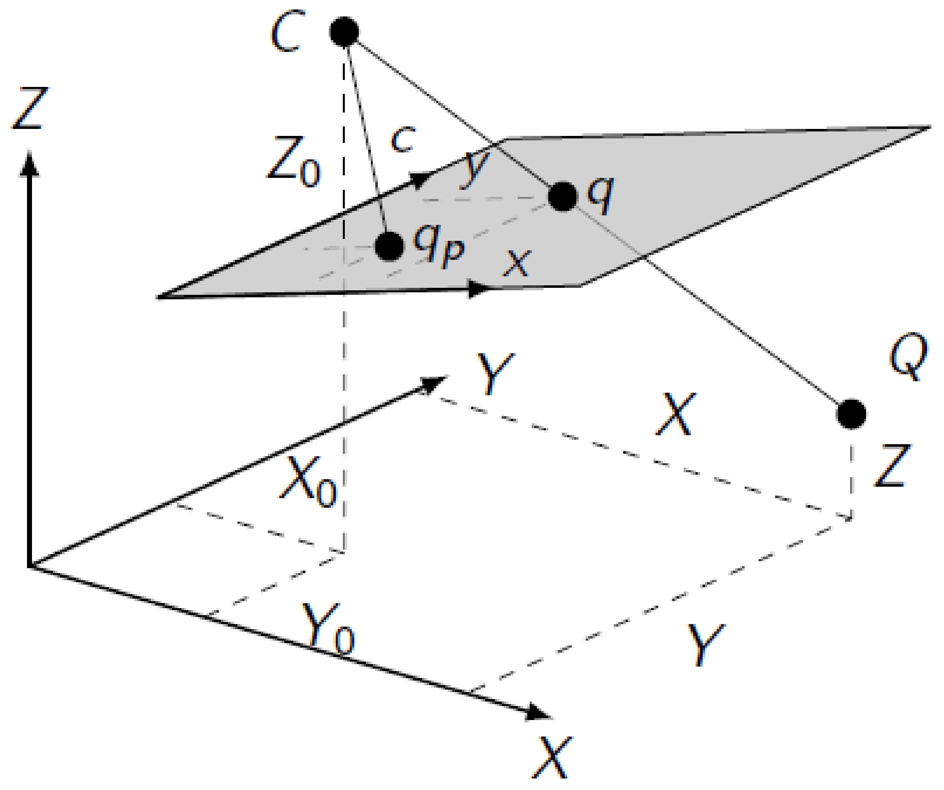
$$\begin{pmatrix} x - x_p \\ y - y_p \\ -c \end{pmatrix} = kR \begin{pmatrix} X - X_0 \\ Y - Y_0 \\ Z - Z_0 \end{pmatrix}$$

describe the relationship between the object point $(X, Y, Z)^T$, the position $C = (X_0, Y_0, Z_0)^T$ of the camera center and the orientation R of the camera.



Co-linearity Equation – Cont.

- The distance c is known as the *principal distance* or *camera constant*.
- The point $q_p = (x_p, y_p)^T$ is called the *principal point*.
- The ray passing through the camera center C and the principal point q_p is called the *principal ray*.



Co-linearity Equation – Cont.

- From

$$\begin{pmatrix} x - x_p \\ y - y_p \\ -c \end{pmatrix} = kR \begin{pmatrix} X - X_0 \\ Y - Y_0 \\ Z - Z_0 \end{pmatrix}, \text{ and } R = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix},$$

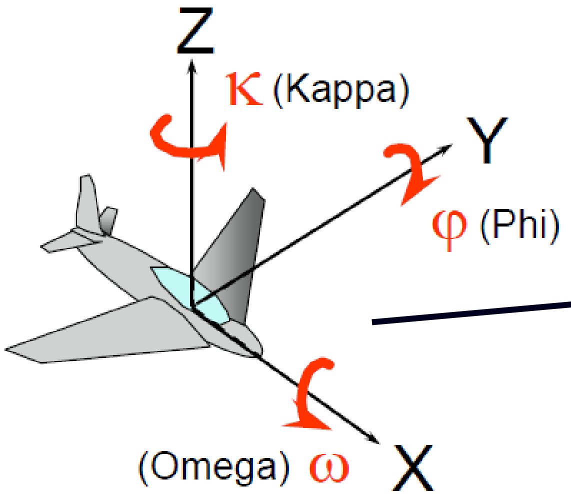
We can get rid of k by ratioing the first and the third, the second and the third equation

$$x = x_p - c\frac{r_{11}(X - X_0) + r_{12}(Y - Y_0) + r_{13}(Z - Z_0)}{r_{31}(X - X_0) + r_{32}(Y - Y_0) + r_{33}(Z - Z_0)},$$

$$y = y_p - c\frac{r_{21}(X - X_0) + r_{22}(Y - Y_0) + r_{23}(Z - Z_0)}{r_{31}(X - X_0) + r_{32}(Y - Y_0) + r_{33}(Z - Z_0)}.$$

Rotation Matrix

angles κ, φ, ω

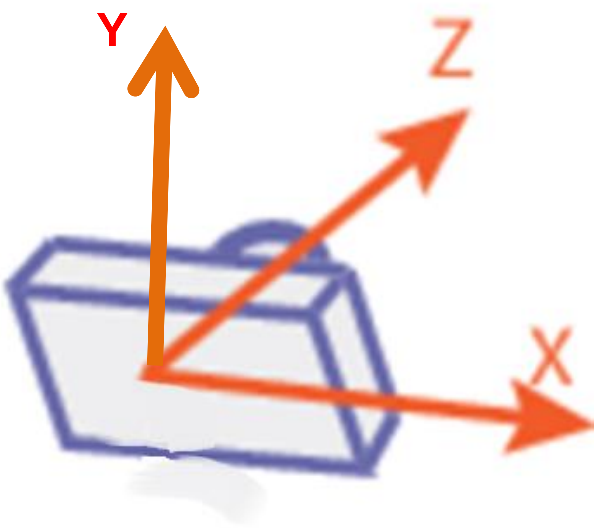


The camera is looking downwards

Camera coordinate frame

– Right handed coordinate system
- Left handed coordinate system (invert Z direction)

Rotation Matrix - Cont.



Computer vision Convention

- Camera coordinate frame
- Left handed system – Z in a different direction as previous one

Rotation Matrix - Cont.

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

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

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

Rotation Matrix - Cont.

$$R = R_z R_y R_x = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

Would changing the sequent result in a different matrix?

Rotation Matrix - Cont.

$$\begin{pmatrix} \cos\phi\cos\kappa & \sin\omega\sin\phi\cos\kappa+\cos\omega\sin\kappa & -\cos\omega\sin\phi\cos\kappa+\sin\omega\sin\kappa \\ -\cos\phi\sin\kappa & -\sin\omega\sin\phi\sin\kappa+\cos\omega\cos\kappa & \cos\omega\sin\phi\sin\kappa+\sin\omega\cos\kappa \\ \sin\phi & -\sin\omega\cos\phi & \cos\omega\cos\phi \end{pmatrix}$$

Question: given a rotation matrix, how to compute the angles? – Any problem you foresee?

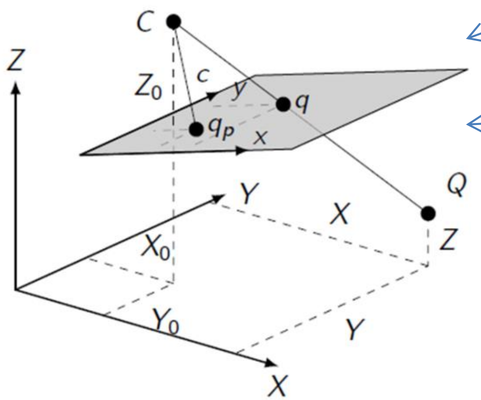
ome = atan(-r32/r 33); kappa = atan(-r21/11)); phi = asin(r13);

Rotation Matrix - Cont.

- Geometric meaning of Rotation Matrix

$$\begin{bmatrix} x - xp \\ y - yp \\ -c \end{bmatrix} = kR \begin{bmatrix} X - X_0 \\ Y - Y_0 \\ Z - Z_0 \end{bmatrix} \longrightarrow R^T \begin{bmatrix} x - xp \\ y - yp \\ -c \end{bmatrix} = k \begin{bmatrix} X - X_0 \\ Y - Y_0 \\ Z - Z_0 \end{bmatrix}$$

$$R^T \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \qquad R^T \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$



$$R^T \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

Where are they in the figure?

Rotation Matrix – Quaternion.

- Representing the rotation using a rotating vector W=(w_1 , w_2 , w_3) and an angle θ.

Rodrigues' rotation formula: the rotation then being

$$R_w(\theta) = I_{3x3} + \sin(\theta) \times S + [1 - \cos(\theta)] \times S^2$$

where
$$S = \begin{bmatrix} 0 & -w_3 & w_2 \\ w_3 & 0 & -w_1 \\ -w_2 & w_1 & 0 \end{bmatrix}$$

Full proof see here:

https://en.wikipedia.org/wiki/Rodrigues%27_rotation_formula

Rotation Matrix – Quaternion – Cont.

- A rotation matrix can be represented by four parameters, θ,w_1,w_2,w_3, where[w_1,w_2,w_3] is a unit vector
- Let's do a little bit of mathematical trick here:
- $q_1 = \sin(\theta/2)w_1, q_2 = \sin(\theta/2)w_2,$
  $q_3 = \sin(\theta/2)w_3,$
- Then

$$q_1 = \mathrm{w}_1 \sin(\theta/2)$$
$$q_2 = \mathrm{w}_2 \sin(\theta/2)$$
$$q_3 = \mathrm{w}_3 \sin(\theta/2)$$

Do a little bit of math by replacing these elements back to the previous *Rodrigues equation*

Rotation Matrix – Quaternion – Cont.

Note:

$$sin(\theta) = 2\sin\left(\frac{\theta}{2}\right)\cos(\frac{\theta}{2}),$$

$$cos(\theta) = \cos^2\left(\frac{\theta}{2}\right) - \sin^2\left(\frac{\theta}{2}\right)$$

$$S^2 = S \times S$$

$$= \begin{bmatrix} -{w_3}^2 - {w_2}^2 & w_1 w_2 & w_1 w_3 \\ w_1 w_2 & -{w_3}^2 - {w_1}^2 & w_2 w_3 \\ w_1 w_3 & w_2 w_3 & -{w_2}^2 - {w_1}^2 \end{bmatrix}$$

Rotation Matrix – Quaternion – Cont.

$$R_w(\theta) = I_{3x3} + \sin(\theta) \times S + [1 - \cos(\theta)] \times S^2$$

$$S \times \sin(\theta) = 2\sin\left(\frac{\theta}{2}\right)\cos\left(\frac{\theta}{2}\right) = 2\sin\left(\frac{\theta}{2}\right)\cos\left(\frac{\theta}{2}\right)\begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{bmatrix} / \sin\left(\frac{\theta}{2}\right)$$

$$S^2 \times [1 - \cos(\theta)] = S^2[1 - cos^2\left(\frac{\theta}{2}\right) + sin^2\left(\frac{\theta}{2}\right)]$$

$$= 2\sin^2\left(\frac{\theta}{2}\right)\begin{bmatrix} -w_3{}^2 - w_2{}^2 & w_1 w_2 & w_1 w_3 \\ w_1 w_2 & -w_3{}^2 - w_1{}^2 & w_2 w_3 \\ w_1 w_3 & w_2 w_3 & -w_2{}^2 - w_1{}^2 \end{bmatrix}$$

$$= 2\begin{bmatrix} -q_3{}^2 - q_2{}^2 & q_1 q_2 & q_1 q_3 \\ q_1 q_2 & -q_3{}^2 - q_1{}^2 & q_2 q_3 \\ q_1 q_3 & q_2 q_3 & -q_2{}^2 - q_1{}^2 \end{bmatrix}$$

Rotation Matrix – Quaternion – Cont.

- Then

$$R_{w(\theta)} = I + 2\cos(\frac{\theta}{2})\begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{bmatrix} +$$

$$2\begin{bmatrix} -{q_3}^2 - {q_2}^2 & q_1 q_2 & q_1 q_3 \\ q_1 q_2 & -{q_3}^2 - {q_1}^2 & q_2 q_3 \\ q_1 q_3 & q_2 q_3 & -{q_2}^2 - {q_1}^2 \end{bmatrix}$$

Let $q_0 = \cos(\frac{\theta}{2})$

Rotation Matrix – Quaternion – Cont.

- Then

$$R_{w(\theta)} = \begin{bmatrix} 1-2{q_3}^2-2{q_2}^2 & 2q_1q_2-2q_0q_3 & 2q_1q_3+2q_0q_2 \\ 2q_1q_2+2q_0q_3 & 1-2{q_3}^2-2{q_1}^2 & 2q_2q_3-2q_0q_1 \\ 2q_1q_3-2q_0q_2 & 2q_2q_3+2q_0q_1 & 1-2{q_2}^2-2{q_1}^2 \end{bmatrix}$$

Where $q_0 = \cos(\frac{\theta}{2})$, $\boldsymbol{q} = [w_1, w_2, w_3] \sin\left(\frac{\theta}{2}\right)$

We have ${q_0}^2+{q_1}^2+{q_2}^2+{q_3}^2=1$

This is just another parameterization of your rotation Matrix! – **Nothing special, you can even use all the elements in the matrix as your parameters**

Rotation Matrix – Quaternion – Cont.

- Given a rotation matrix, how to get **n =** $[w_1, w_2, w_3]$ and θ?
- Tip 1: take the trace of the previous matrix to get θ.
- Tip 2: make combinations of the elements to get **n**

$$\theta = \cos^{-1}\left(\frac{\text{trace}(\mathbf{R}) - 1}{2}\right), \hat{\mathbf{n}} = \frac{1}{2\sin\theta}\begin{pmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{pmatrix}$$

Rotation Matrix – Quaternion – Cont.

- Advantage, easy to formulate when given the rotation axis and angle, this is very common
  - Have vector v_1, want to rotate to v_2
  - Need rotation vector $\hat{r}$, angle θ

$$\theta = \operatorname{acos}(\hat{\mathbf{v}}_1 \bullet \hat{\mathbf{v}}_2)$$
$$\mathbf{r} = \mathbf{v}_1 \times \mathbf{v}_2$$

$$\boldsymbol{n} = \boldsymbol{r}/|\boldsymbol{r}|$$

$q_0 = \cos(\frac{\theta}{2})$, $\boldsymbol{q} = \boldsymbol{n}\sin\left(\frac{\theta}{2}\right)$, Plug back

Rotation Matrix – Comparison.

Euler	Quternion
Advantage: Minimal representation (3 parameters) Easy interpretation Disadvantages: Many “alternative” Euler representations exist (XYZ, ZXZ, ZYX, …) Difficult to concatenate Singularities (gimbal lock)	Advantage: Easy to represent rotating vectors Inverse = easy to compute Disadvantages: One over-parameterization

3D to 2D relationship

$$x = x_p - c\frac{r_{11}(X - X_0) + r_{12}(Y - Y_0) + r_{13}(Z - Z_0)}{r_{31}(X - X_0) + r_{32}(Y - Y_0) + r_{33}(Z - Z_0)},$$

$$y = y_p - c\frac{r_{21}(X - X_0) + r_{22}(Y - Y_0) + r_{23}(Z - Z_0)}{r_{31}(X - X_0) + r_{32}(Y - Y_0) + r_{33}(Z - Z_0)}.$$

r_{ij} can be parameterized by ω, φ, κ or q_0, q_1, q_2, q_3

Given any ground points X,Y,Z, to get its position in the image, you need to know, r_{ij}, X_0, Y_0, Z_0 of this image, and the principal points x_p and y_p;
x,y: image position in the actual films.

3D to 2D relationship

- Then you need to know pixel size, in order to navigate back to its pixel location:

$$x_{pix} = \frac{x}{psz_x}$$

$$y_{pix} = \frac{y}{psz_y} \; or \; imgheight - \frac{y}{psz_y}$$

psz_x and $\mathrm{psz_y}$: pixel size of one cell in CCD

3D to 2D relationship

- Recall

$$\begin{bmatrix} x - xp \\ y - yp \\ -c \end{bmatrix} = kR \begin{bmatrix} X - X_0 \\ Y - Y_0 \\ Z - Z_0 \end{bmatrix}$$

$$\begin{bmatrix} x_{pix} \\ y_{pix} \\ -c \end{bmatrix} = \begin{bmatrix} \frac{1}{psz_x} & & \frac{xp}{(-c)(psz_x)} \\ & \frac{1}{psz_y} & \frac{yp}{(-c)(psz_y)} \\ & & 1 \end{bmatrix} \begin{bmatrix} x - xp \\ y - yp \\ -c \end{bmatrix}$$

3D to 2D relationship

$$\begin{bmatrix} x_{pix} \\ y_{pix} \\ -c \end{bmatrix} = \begin{bmatrix} \frac{1}{psz_x} & & \frac{xp}{(-c)(psz_x)} \\ & \frac{1}{psz_y} & \frac{yp}{(-c)(psz_y)} \\ & & 1 \end{bmatrix} \begin{bmatrix} x - xp \\ y - yp \\ -c \end{bmatrix}$$

$$= \begin{bmatrix} \frac{-c}{psz_x} & & \frac{xp}{(psz_x)} \\ & \frac{-c}{psz_y} & \frac{yp}{(psz_y)} \\ & & -c \end{bmatrix} \begin{bmatrix} (x - xp)/(-c) \\ (y - yp)/(-c) \\ 1 \end{bmatrix}$$

3D to 2D relationship – Camera matrix

$$\begin{bmatrix} x_{pix} \\ y_{pix} \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{-c}{psz_x} & & \frac{xp}{(psz_x)} \\ & \frac{-c}{psz_y} & \frac{yp}{(psz_y)} \\ & & 1 \end{bmatrix} \begin{bmatrix} (x - xp)/(-c) \\ (y - yp)/(-c) \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x_{pix} \\ y_{pix} \\ 1 \end{bmatrix} = \boldsymbol{K} \begin{bmatrix} (x - xp)/(-c) \\ (y - yp)/(-c) \\ 1 \end{bmatrix} = \boldsymbol{K} kR \begin{bmatrix} X - X_0 \\ Y - Y_0 \\ Z - Z_0 \end{bmatrix} /(-c)$$

$$= \boldsymbol{K} \lambda R \begin{bmatrix} X - X_0 \\ Y - Y_0 \\ Z - Z_0 \end{bmatrix}$$

$\boldsymbol{K}$ is called camera matrix in computer vision

3D to 2D relationship – Projection Matrix

- Let’s make it looks even more compact:

$$\begin{bmatrix} x_{pix} \\ y_{pix} \\ 1 \end{bmatrix} = \lambda \boldsymbol{K}[R \quad -RT] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, \boldsymbol{T} = \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \end{bmatrix}$$

Let $\boldsymbol{P} = \lambda \boldsymbol{K}[R \quad -RT]$, **then** $\boldsymbol{P}$ is called Projection Matrix.

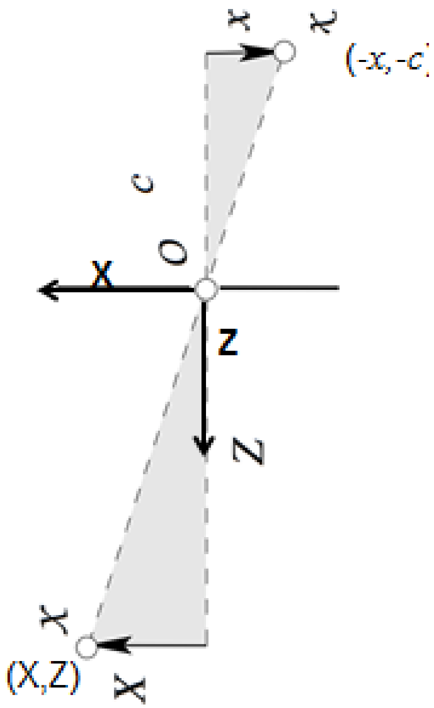
$$\begin{bmatrix} x_{pix} \\ y_{pix} \\ 1 \end{bmatrix} = \boldsymbol{P} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Projection Matrix

$$\begin{bmatrix} x_{pix} \\ y_{pix} \\ 1 \end{bmatrix} = \lambda \boldsymbol{K}[R \quad -RT] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, \boldsymbol{T} = \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \end{bmatrix}$$

$$\boldsymbol{P} = \lambda \boldsymbol{K}[R \quad -RT] - Projection\ Matrix$$

Homogeneous Coordinates



$$\frac{c}{Z} = \frac{x}{X} = m$$

$$x = \frac{c}{Z} X = mX$$

Given X we are getting x through a scale, this is represented by scaling through Z

For any $(\bar{X}, \bar{Z}) = k(X, Z)$, we get the same point x on the image, We want to just represent such points in the space as one point.

Purpose: Easy to represent; a image point x, can be represented in the space by associating to the a scale factor $k[x,1]^T$

$\begin{bmatrix} x \\ 1 \end{bmatrix} = \begin{bmatrix} kx \\ k \end{bmatrix}$, this is defined under the homogeneous coordinate representation

Homogeneous Coordinates – Cont.

Motivation

- Cameras generate a projected image of the world
- **Euclidian geometry is suboptimal to describe the central projection**
- In Euclidian geometry, the math can get difficult
- Projective geometry is an alternative algebraic representation of geometric objects and transformations

Homogeneous Coordinates – Cont.

- Math becomes simpler
- Projective geometry does not change the geometric relations
- Computations can also be done in Euclidian geometry (but more difficult)

Homogeneous Coordinates – Cont.

- H.C. are a system of coordinates used in projective geometry
- Formulas involving H.C. are often simpler than in the Cartesian world
- Points at infinity can be represented using finite coordinates
- A single matrix can represent affine and projective transformations

Homogeneous Coordinates – Cont.

Definition

The representation $\mathbf{x}$ of a geometric object is **homogeneous** if $\mathbf{x}$ and $\lambda\mathbf{x}$ represent the same object for $\lambda \neq 0$

Example

$$\mathbf{x} = \lambda\,\mathbf{x}$$

homogeneous

$$x \neq \lambda\,x$$

Euclidian

Homogeneous Coordinates – Cont.

- H.C. use a n+1 dimensional vector to represent the same (n-dim.) point
- Example for $\mathbb{R}^2/\mathbb{P}^2$

$$\boldsymbol{x} = \begin{bmatrix} x \\ y \end{bmatrix} \Rightarrow \mathbf{x} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$\mathbf{x} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} = w \begin{bmatrix} u/w \\ v/w \\ 1 \end{bmatrix} = \begin{bmatrix} u/w \\ v/w \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Homogeneous Coordinates – Cont.

Definition

The representation $\mathbf{x}$ of a geometric object is **homogeneous** if $\mathbf{x}$ and $\lambda\mathbf{x}$ represent the same object for $\lambda \neq 0$

Example

$$\mathbf{x} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} wx \\ wy \\ w1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \qquad \boldsymbol{x} = \begin{bmatrix} x \\ y \end{bmatrix}$$

homogeneous

Euclidian

Homogeneous Coordinates – Cont.

- Homogeneous Coordinates of a point χ in the plane $\mathbb{R}^2$ is a 3-dim. vector

$$\chi : \quad \mathbf{x} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} \text{ with } |\mathbf{x}|^2 = u^2 + v^2 + w^2 \neq 0$$

- it corresponds to Euclidian coordinates

$$\chi : \quad \boldsymbol{x} = \begin{bmatrix} u/w \\ v/w \end{bmatrix} \text{ with } w \neq 0$$

Homogeneous Coordinates – Cont.

The projective plane $\mathbb{P}^2(\mathbb{R})$ or $\mathbb{P}^2$ contains

- All points $\mathcal{X}$ of the Euclidian plane $\mathbb{R}^2$ with $\boldsymbol{x} = [x, y]^\top$ expressed through the 3-valued vector (e.g., $\mathbf{x} = [x, y, 1]^\top$)
- and all points at infinity, i.e.,

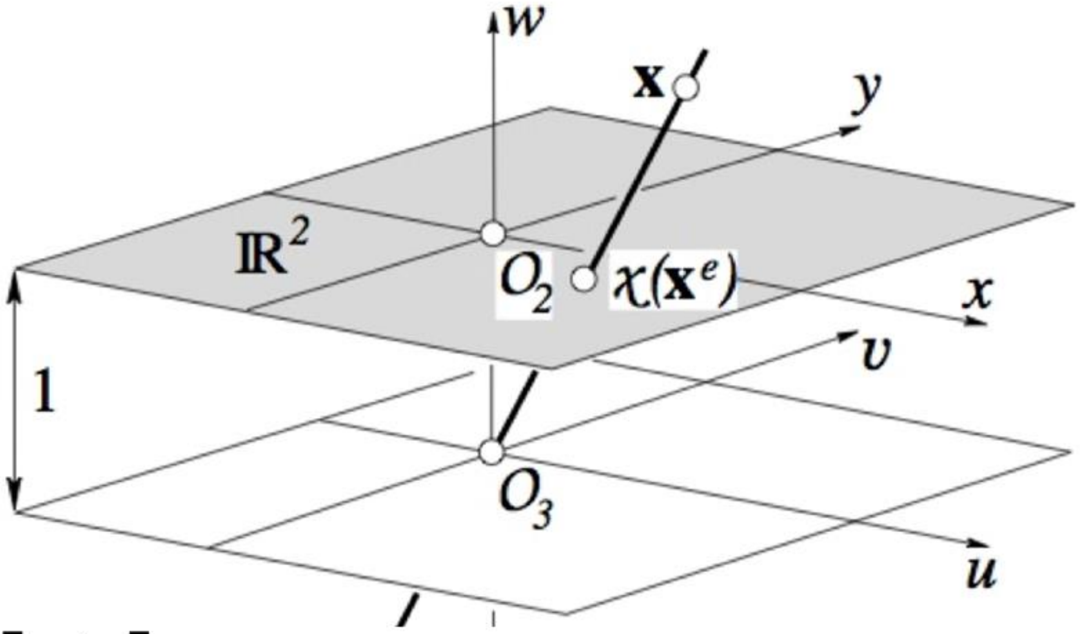
  $\mathbf{x} = [x, y, 0]^\top$
- except $[0, 0, 0]^\top$

Homogeneous Coordinates – Cont.

From Homogeneous to Euclidian Coordinates

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} u/w \\ v/w \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} u/w \\ v/w \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix}$$

From Homogeneous to Euclidian Coordinates



$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} u/w \\ v/w \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} u/w \\ v/w \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix}$$

Image courtesy: Förstner

Homogeneous Coordinates – Cont.

3D Points

Analogous for points in 3D Euclidian space $\mathbb{R}^3$

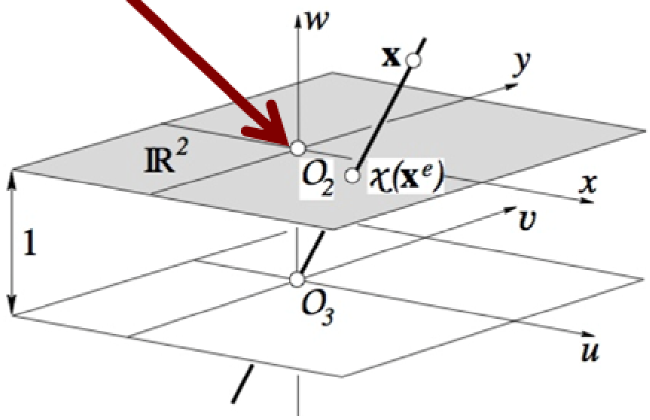
homogeneous Euclidian

$$\mathbf{X} = \begin{bmatrix} U \\ V \\ W \\ T \end{bmatrix} = \begin{bmatrix} U/T \\ V/T \\ W/T \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} U/T \\ V/T \\ W/T \end{bmatrix}$$

Origin of the Euclidian Coordinate System in H.C.

$$\mathbf{O}_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$\mathbf{O}_3 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$



Representations of Lines

- Hesse normal form (angle ϕ, distance d)

$$x \cos \phi + y \sin \phi - d = 0$$

- Intercept form

$$\frac{x}{x_0} + \frac{y}{y_0} = 1 \qquad \text{or} \qquad \frac{x}{x_0} + \frac{y}{y_0} - 1 = 0$$

- Standard form

$$ax + by + c = 0$$

All form linear equations that are equal to zero

Representations of Lines

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

point

$$\mathbf{l} = \begin{bmatrix} \cos\phi \\ \sin\phi \\ -d \end{bmatrix}$$

Hesse

$$\mathbf{l} = \begin{bmatrix} \frac{1}{x_0} \\ \frac{1}{y_0} \\ -1 \end{bmatrix}$$

intercept

$$\mathbf{l} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

standard

$$\boxed{\mathbf{x} \cdot \mathbf{l} = \mathbf{x}^\mathsf{T}\mathbf{l} = \mathbf{l}^\mathsf{T}\mathbf{x} = 0}$$

H.C. – Lines – Cont.

Definition

- Homogeneous Coordinates of a line ℓ in the plane is a 3-dim. vector

$$\ell: \quad \mathbf{l} = \begin{bmatrix} l_1 \\ l_2 \\ l_3 \end{bmatrix} \text{ with } |\mathbf{l}|^2 = l_1^2 + l_2^2 + l_3^2 \neq 0$$

- it corresponds to Euclidian representation

$$l_1 x + l_2 y + l_3 = 0$$

$\mathbf{l} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$ does not corresponds to any line and hence is excluded

H.C. – Lines – Cont.

Test If a Point Lies on a Line

- A point

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- lies on a line

$$\mathbf{l} = \begin{bmatrix} l_1 \\ l_2 \\ l_3 \end{bmatrix}$$

- if $\mathbf{x} \cdot \mathbf{l} = 0$

Intersecting Lines

- The intersection of two lines in H.C. is

$$\boxed{x = \ell \cap m : \quad \mathbf{x} = \mathbf{l} \times \mathbf{m}}$$

- **Simple way for computing the intersection of two lines using H.C.**

H.C. – Lines – Cont.

- Line l between two points $\boldsymbol{x}, \boldsymbol{y}$:

Idea: both points line on that line, meaning:

$$\boldsymbol{x} \cdot \boldsymbol{l} = \mathbf{0}, \boldsymbol{y} \cdot \boldsymbol{l} = \mathbf{0}$$

We know that

$$(\boldsymbol{x} \times \mathbf{y}) \cdot \boldsymbol{x} = \mathbf{0}$$
$$(\boldsymbol{x} \times \mathbf{y}) \cdot \boldsymbol{y} = \mathbf{0}$$

Therefore:

$$\boldsymbol{l} = \boldsymbol{x} \times \mathbf{y}$$

H.C. Lines and points – Duality

- A point lies on a line if

$$\mathbf{x} \cdot \mathbf{l} = 0$$

- Intersection of two lines

$$x = \ell \cap m : \quad \mathbf{x} = \mathbf{l} \times \mathbf{m}$$

- A line through two given points

$$\ell = x \wedge y : \quad \mathbf{l} = \mathbf{x} \times \mathbf{y}$$

Duality

- Without proof we give the definition of duality in the homogenous coordinate system formulation

To any theorem of 2-dimension projective geometry, there corresponds a dual theorem, which may be derived by interchanging the roles of points and lines in the original theorem.

Points at Infinity

- It is possible to **explicitly** model infinitively distant points **with finite coordinates**

$$\chi_\infty : \quad \mathbf{x}_\infty = \begin{bmatrix} u \\ v \\ 0 \end{bmatrix}$$

$\boldsymbol{x}_\infty = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$ (ideal point, all the lines intersect to this infinity point)

- We can **maintain the direction** to that infinitively distant point
- Great tool when working with cameras as they are bearing-only sensors

H.C. – Infinity – Cont.

Intersection at Infinity

- All lines ℓ with $\ell \cdot x_\infty = 0$ pass through x_∞
- This means $[u, v] \cdot [\cos\phi, \sin\phi] = 0$
- This hold for any line $\mathbf{m} = [\cos\phi, \sin\phi, *]^T$ i.e. for any line that is parallel to ℓ

$$\mathbf{l} \times \mathbf{m} = \begin{bmatrix} a \\ b \\ c \end{bmatrix} \times \begin{bmatrix} a \\ b \\ d \end{bmatrix} = \begin{bmatrix} bd - bc \\ ac - ad \\ ab - ab \end{bmatrix} = \begin{bmatrix} bd - bc \\ ac - ad \\ 0 \end{bmatrix}$$

All parallel lines meet at one point at infinity!

H.C. – Infinity – Cont.

Lines at Infinity

$l_\infty = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$, all the points at infinity will lie on this line. (also called ideal line)

i.e. $p_\infty = \begin{bmatrix} x \\ y \\ 0 \end{bmatrix}$

The line at infinity is invariant of affine!

H.C. – Infinity – Cont.

Image Courtesy: J. Jannene

Analogous for 3D Objects

- 3D point

$$\mathbf{X} = \begin{bmatrix} U \\ V \\ W \\ T \end{bmatrix} = \begin{bmatrix} U/T \\ V/T \\ W/T \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} U/T \\ V/T \\ W/T \end{bmatrix}$$

- Plane

$$\mathbf{A} = \begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix}$$

Similar properties in terms of infinity, can be extended

Point on a Plane

- Via the scalar product, we can again test if a point lies on a plane

$$\mathbf{A} \cdot \mathbf{X} = \mathbf{A}^{\mathsf{T}}\mathbf{X} = \mathbf{X}^{\mathsf{T}}\mathbf{A} = 0$$

- which is based on

$$AX + BY + CZ + D = 0$$

H.C. in 3D

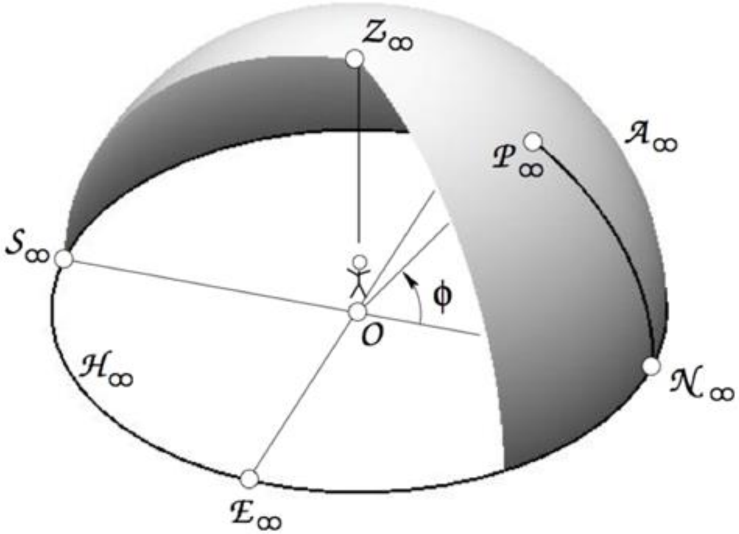
3D Objects at Infinity

- 3D point

$$\mathbf{P}_\infty = \begin{bmatrix} U \\ V \\ W \\ 0 \end{bmatrix}$$

- Plane

$$\mathbf{A}_\infty = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$



Projection Matrix

$$\begin{bmatrix} x_{pix} \\ y_{pix} \\ 1 \end{bmatrix} = \lambda \boldsymbol{K}[R \quad -RT] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, \boldsymbol{T} = \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \end{bmatrix}$$

$$\boldsymbol{P} = \lambda \boldsymbol{K}[R \quad -RT] - Projection\ Matrix$$

$$\begin{bmatrix} x_{pix} \\ y_{pix} \\ 1 \end{bmatrix} = \boldsymbol{K}[R \quad -RT] \boxed{\lambda \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}}$$

or understand as

$$\frac{1}{\lambda} \begin{bmatrix} x_{pix} \\ y_{pix} \\ 1 \end{bmatrix} = \boldsymbol{K}[R \quad -RT] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

This is represented as H.C. for the output, we can get the results by ignoring λ in the definition of $\boldsymbol{P}$

$$\frac{1}{\lambda} \begin{bmatrix} x_{pix} \\ y_{pix} \\ 1 \end{bmatrix} = \boldsymbol{P} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Next Class

- Geometric Transformation
- RANSAC Algorithm
- Panorama – Assignment 2