Mathematics for Computer Vision

COSC450

Lecture 1

Points, Directions, Vectors, Transforms, Matrices

Points and directions in 2D and 3D

- Represented as co-ordinates

$$(u, v) \qquad (x, y, z)$$

- Represented as vectors

$$\mathbf{u} = \begin{bmatrix} u \\ v \end{bmatrix} \qquad \mathbf{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

- How to tell points from directions?

Transformations applied to points

- Translation

$$\begin{bmatrix} u' \\ v' \end{bmatrix} = \begin{bmatrix} u \\ v \end{bmatrix} + \begin{bmatrix} \delta_u \\ \delta_v \end{bmatrix}$$

- Rotation

$$\begin{bmatrix} u' \\ v' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \delta_u \\ \delta_v \end{bmatrix}$$

- Scale

$$\begin{bmatrix} u' \\ v' \end{bmatrix} = s \begin{bmatrix} u \\ v \end{bmatrix}$$

- How to combine transformations?

Homogeneous Co-ordinates

2D (3D) points as 3D (4D) lines

$$(u, v) \to k \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \qquad (x, y, z) \to k \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

- k is any non-zero scalar
- How to convert back from

$$\begin{bmatrix} a & b & c \end{bmatrix}^\mathsf{T}$$

  to a 2D point?
- Why is this a good idea anyway?!?

Directions in homogeneous form

- Difference of two points:

$$\begin{bmatrix} a \\ b \\ 1 \end{bmatrix} - \begin{bmatrix} c \\ d \\ 1 \end{bmatrix} = \begin{bmatrix} a - c \\ b - d \\ 0 \end{bmatrix}$$

- How do we convert back to a 2D point?
- Directions and point are different
- ‘Points at infinity’

Homogeneous Transforms

Translation

$$\begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} \equiv \begin{bmatrix} 1 & 0 & \delta_u \\ 0 & 1 & \delta_v \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

Rotation

$$\begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} \equiv \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

Scale

$$\begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} \equiv \begin{bmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

All have the same form

- Easy to combine – multiply
- Result is the same form
- Transforming directions?
- What does $\equiv$ mean here?

Cameras and Projections

A camera projects 3D to 2D

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

- Basic project matrix:

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

- General projection matrix

$$\mathbf{u} \equiv \mathrm{K} \left[\mathrm{R} | \mathbf{t} \right] \mathbf{x}$$

- R is a 3×3 3D rotation matrix
- $\mathbf{t}$ is a 3×1 3D translation vector
- K is a 3×3 calibration matrix

$$\mathrm{K} = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

- f_x and f_y are focal lengths (often equal)
- (c_x, c_y) is the centre of the projection
- s is the skew of the pixels (often 0)

For Next Time...

- Read the paper *A Flexible New Technique for Camera Calibration* by Zhengyou Zhang
- Don't worry if you don't get all the details, we'll discuss it next lecture
- The maths above is used in that paper
- More mathematics revision material linked on course website