Lecture 2, Sep 5, 2025

Mathematical Fundamentals of Vision

- We need to reason over both the *geometric* (points, lines, shapes) and *photometric* (brightness, contrast, texture, shading) aspects of the scene, which are closely intertwined
- We will use $\underrightarrow{\mathcal{F}}_v$ to denote frame v (vectrix notation) and column vectors like $\boldsymbol{x} = \begin{bmatrix} x \\ y \end{bmatrix}$

2D Transformations

- Points may also be represented in *homogeneous form*: $\tilde{\boldsymbol{x}} = (\tilde{x}, \tilde{y}, \tilde{w}) \in \mathbb{P}^2$
  - $\mathbb{P}^2 = \mathbb{R}^3 \backslash (0, 0, 0)$ is a *projective space*
  - This can be converted back to an inhomogeneous vector by dividing by $\tilde{w}$: $\tilde{\boldsymbol{x}} = \tilde{w}(x, y, 1) = \tilde{w}\bar{\boldsymbol{x}}$
  - $\bar{\boldsymbol{x}} = (x, y, 1)$ is the *augmented vector* (note the bar), with a canonical scale of 1
  - $\tilde{w} = 0$ represent *points at infinity* (aka *ideal points*); hence $(0, 0, 0)$ is undefined and excluded from $\mathbb{P}^2$
- Projective geometry allows us to represent and manipulate objects at infinity, which is necessary for cameras
  - Since they are homogeneous (not affected by scalar multiplication), $\mathbb{P}^2$ is topologically equivalent to the unit sphere
- $\tilde{\boldsymbol{l}} = (a, b, c)$ represents the line $\bar{\boldsymbol{x}} \cdot \tilde{\boldsymbol{l}} = ax + by + c = 0$ in 2D
  - This can be normalized to $\boldsymbol{l} = (\hat{n}_x, \hat{n}_y, d) = (\hat{\boldsymbol{n}}, d)$ where $\hat{\boldsymbol{n}}$ is the unit normal vector and d is the distance to origin
  - The intersection of two lines can be found by taking their cross product
- Define the *skew-symmetric form* $[\boldsymbol{u}]_\times = \begin{bmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{bmatrix}$
  - This is skew-symmetric, i.e. $[\boldsymbol{u}]_\times^T = -[\boldsymbol{u}]_\times$
  - This allows us to write the cross product as $[\boldsymbol{u}]_\times \boldsymbol{v} = \begin{bmatrix} u_2 v_3 - v_2 u_3 \\ u_3 v_1 - v_3 u_1 \\ u_1 v_2 - v_1 u_2 \end{bmatrix}$
- A rigid transformation can be represented as $\boldsymbol{x}' = \begin{bmatrix} \boldsymbol{C} & \boldsymbol{t} \end{bmatrix} \bar{\boldsymbol{x}}$ where $\boldsymbol{C}$ is a rotation matrix, $\boldsymbol{t}$ is a translation vector
  - Note $\det \boldsymbol{C} = 1$ and $\boldsymbol{C}\boldsymbol{C}^T = \boldsymbol{C}^T\boldsymbol{C} = \boldsymbol{I}$
  - We are rotating the vector while keeping the reference frame constant, instead of the other way around
- An *affine transformation* is $\boldsymbol{x}' = \boldsymbol{A}\bar{\boldsymbol{x}}$ where $\boldsymbol{A} \in \mathbb{R}^{2\times 3}$
  - Important to note parallel lines remain parallel after an affine transformation
  - This has 6 degrees of freedom
- A *projective transformation* or *homography* is $\tilde{\boldsymbol{x}}' = \tilde{\boldsymbol{H}}\tilde{\boldsymbol{x}}$
  - Straight lines remain straight, but parallel lines may not be parallel after the transformation
  - This has 8 degrees of freedom: $\tilde{\boldsymbol{H}} \in \mathbb{R}^{3\times 3}$ (note below)
  - Note $\tilde{\boldsymbol{H}}$ is also *homogeneous*, i.e. defined up to scale only
    * This is similar to homogeneous coordinates; if we multiply all 3 components by some scalar, we get the same point back just represented differently

3D Transformations

- Rotations preserve the length and orientating (handedness) of space
- Rotations have the following properties: let a, b, c be rotations, then:
  - Closure: $a \circ b$ is a rotation
  - Associativity: $(a \circ b) \circ c = a \circ (b \circ c)$
  - Invertibility: each rotation has a unique inverse rotation
  - Identity: the identity map is a rotation

Transformation	Matrix	# DoF	Preserves	Icon
translation	$\begin{bmatrix} \boldsymbol{I} \mid \boldsymbol{t} \end{bmatrix}_{2\times 3}$	2	orientation	
rigid (Euclidean)	$\begin{bmatrix} \boldsymbol{R} \mid \boldsymbol{t} \end{bmatrix}_{2\times 3}$	3	lengths	
similarity	$\begin{bmatrix} s\boldsymbol{R} \mid \boldsymbol{t} \end{bmatrix}_{2\times 3}$	4	angles	
affine	$\begin{bmatrix} \boldsymbol{A} \end{bmatrix}_{2\times 3}$	6	parallelism	
projective	$\begin{bmatrix} \tilde{\boldsymbol{H}} \end{bmatrix}_{3\times 3}$	8	straight lines	

Figure 1: Hierarchy of 2D coordinate transformations.

- Therefore rotations form a *group* under composition; this is the *rotation group* or *special orthogonal group* on $\mathbb{R}^3$, denoted $SO(3)$
- Rotations can be represented by matrices, Euler angles, axis/angle or quaternions
  - Euler angles decomposes rotations into the product of 3 elementary rotations about individual frame axes
    * Due to different orders, there are 12 possible rotation sequences
    * Suffers from gimbal lock
  - Axis-angle expresses rotations as angle θ around a unit vector $\hat{\boldsymbol{n}}$
    * Note only the perpendicular component rotates
    * *Rodriguez formula*: $\boldsymbol{C}(\hat{\boldsymbol{n}}, \theta) = \boldsymbol{I}_3 + \sin\theta\,[\hat{\boldsymbol{n}}]_\times + (1-\cos\theta)\,[\hat{\boldsymbol{n}}]_\times^2$
      - This can be derived by decomposing the vector into parallel and perpendicular components and rotating the perpendicular component
  - Quaternions are hyper-complex numbers that take the form $\boldsymbol{q} = q_0 + q_1\hat{\imath} + q_2\hat{\jmath} + q_3\hat{k}$
    * $i^2 = j^2 = k^2 = ijk = -1$
    * $ij = k, jk = i, ki = j, ji = -k, kj = -i, ik = -j$
      - Note i, j, k do not commute
    * The set of quaternions is denoted $\mathbb{H}$ and form a 4D non-commutative division algebra
    * Unit quaternions satisfy $\|\boldsymbol{q}\|^2 = q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1$ and can be mapped to rotations
    * They have a direct relationship with the axis angle form: $\boldsymbol{q} = \begin{bmatrix} q_0 \\ \mathbf{q} \end{bmatrix} = \begin{bmatrix} \cos(\theta/2) \\ \hat{\boldsymbol{u}}\sin(\theta/2) \end{bmatrix}$
      - Using Rodriguez's formula, we have $\boldsymbol{C} = \boldsymbol{I}_3 + 2q_0\,[\mathbf{q}]_\times + 2\,[\mathbf{q}]_\times^2$
      - Explicit form: $\boldsymbol{C}(\boldsymbol{q}) = \begin{bmatrix} 1-2q_2^2-2q_3^2 & 2q_1q_2-2q_0q_3 & 2q_0q_2+2q_1q_3 \\ 2q_0q_3+2q_1q_2 & 1-2q_1^2-2q_3^2 & 2q_2q_3-2q_0q_1 \\ 2q_1q_3-2q_0q_2 & 2q_0q_1+2q_2q_3 & 1-2q_1^2-2q_2^2 \end{bmatrix}$
    * To compose two rotations represented as quaternions, we can multiply them, following the rules of quaternion multiplication (denoted $\otimes$)
      - $\boldsymbol{p} \otimes \boldsymbol{q} = (p_0 + \mathbf{p}) \otimes (q_0 + \mathbf{q})$
        $= p_0q_0 - \mathbf{p}^T\mathbf{q} + p_0\mathbf{q} + q_0\mathbf{p} + \mathbf{p}\times\mathbf{q}$
      - Note order matters!
- In 3D, rigid transformations take the same form of $\boldsymbol{x}' = \begin{bmatrix} \boldsymbol{C} & \boldsymbol{t} \end{bmatrix} \bar{\boldsymbol{x}}$
- Affine transformations are analogous but $\boldsymbol{A} \in \mathbb{R}^{3\times 4}$
- Projective transformations now use $\tilde{\boldsymbol{H}} \in \mathbb{R}^{4\times 4}$

	Euclidean	similarity	affine	projective
Transformations				
rotation	X	X	X	X
translation	X	X	X	X
uniform scaling		X	X	X
nonuniform scaling			X	X
shear			X	X
perspective projection				X
composition of projections				X
Invariants				
length	X			
angle	X	X		
ratio of lengths	X	X		
parallelism	X	X	X	
incidence	X	X	X	X
cross ratio	X	X	X	X

Figure 2: Summary of different geometries, allowed transformations in each, and which quantities are invariant under the allowed transformations.