Rigid Geometric Transformations and the Pinhole Camera Model

COMPSCI 527 — Computer Vision

Outline

Rigid Transformations

- 3D reconstruction: Given corresponding points in two (or more) images taken from different viewpoints, find the relative pose of the two cameras and 3D coordinates of the world points
- The relative motion between a camera and an otherwise static scene is a rigid transformation: rotation + translation
- Reconstruction techniques also require knowing about othogonal projection, cross product, triple product
- *All vectors are in* $\mathbb{R}^3$

The Cross Product

- Geometry: The cross product of two three-dimensional vectors **a** and **b** is a vector **c** orthogonal to both **a** and **b**, oriented so that the triple **a**, **b**, **c** is right-handed, and with magnitude

$$\|\mathbf{c}\| = \|\mathbf{a} \times \mathbf{b}\| = \|\mathbf{a}\| \, \|\mathbf{b}\| \sin\theta$$

  where θ is the smaller angle between **a** and **b**
- The magnitude of $\mathbf{a} \times \mathbf{b}$ is the area of a parallelogram with sides **a** and **b**
- Algebra: $\mathbf{c} = \mathbf{a} \times \mathbf{b} = \begin{vmatrix} a_x & a_y & a_z \\ b_x & b_y & b_z \end{vmatrix}$

  $= (a_y b_z - a_z b_y \,,\; a_z b_x - a_x b_z \,,\; a_x b_y - a_y b_x)^T$
- Easy to check that $\mathbf{a} \times \mathbf{b} = -\mathbf{b} \times \mathbf{a}$

The Cross-Product Matrix

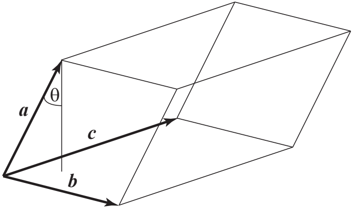
- $\mathbf{c} = (a_y b_z - a_z b_y \, , \; a_z b_x - a_x b_z \, , \; a_x b_y - a_y b_x)^T$ is linear in $\mathbf{b}$
- Therefore, there exists a 3×3 matrix $[\mathbf{a}]_\times$ such that

  $\mathbf{c} = \mathbf{a} \times \mathbf{b} = [\mathbf{a}]_\times \mathbf{b}$

  $$\mathbf{c} = \begin{bmatrix} c_x \\ c_y \\ c_z \end{bmatrix} = \begin{bmatrix} & & \\ & & \\ & & \end{bmatrix} \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix}$$
- The matrix $[\mathbf{a}]_\times$ is skew-symmetric: $[\mathbf{a}]_\times^T = -[\mathbf{a}]_\times$

The Triple Product

- Definition: $\det([\mathbf{a}, \mathbf{b}, \mathbf{c}]) = \mathbf{a}^T(\mathbf{b} \times \mathbf{c})$
  $= a_x(b_y c_z - b_z c_y) - a_y(b_x c_z - b_z c_x) + a_z(b_x c_y - b_y c_x)$
- Signed volume of parallelepiped



- Easy to check: $\mathbf{a}^T(\mathbf{b} \times \mathbf{c}) = \mathbf{b}^T(\mathbf{c} \times \mathbf{a}) = \mathbf{c}^T(\mathbf{a} \times \mathbf{b}) = -\mathbf{a}^T(\mathbf{c} \times \mathbf{b}) = -\mathbf{c}^T(\mathbf{b} \times \mathbf{a}) = -\mathbf{b}^T(\mathbf{a} \times \mathbf{c})$

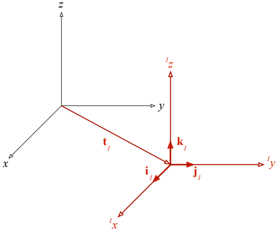
Multiple Reference Systems

- If we associate a reference system to a camera and the camera moves, or we consider multiple cameras, or we consider one camera and the world, we have multiple reference systems
- Point coordinates are x, y, z
- Left superscript denotes which reference system coordinates are expressed in: ${}^1y$
- Subscripts denote which point or reference system we are talking about: x_2
- ${}^2y_3$ is the y coordinate of point 3 in reference system 2

Multiple Reference Systems

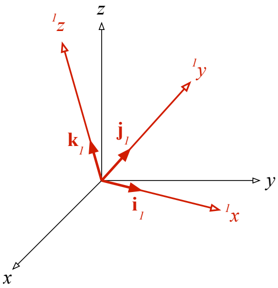
- A zero left superscript can be omitted: ${}^0z = z$
- The origin of a reference system is $\mathbf{t}$ (for "translation")
- We always have ${}^i\mathbf{t}_i = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$
- If $\mathbf{i}$, $\mathbf{j}$, $\mathbf{k}$ are the unit points of a reference system, we always have $\begin{bmatrix} {}^i\mathbf{i}_i & {}^i\mathbf{j}_i & {}^i\mathbf{k}_i \end{bmatrix} = I$, the 3×3 identity matrix

Translations



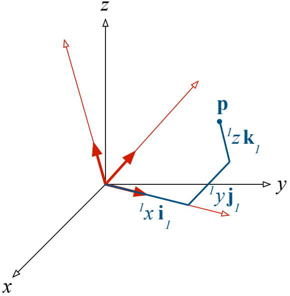
- No rotation: ${}^0R_1 = R_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
- Both systems right-handed
- ${}^0\mathbf{t}_1 = \mathbf{t}_1$ is the origin of reference system 1 expressed in reference system 0
- Given $\mathbf{p} = {}^0\mathbf{p}$, we have ${}^1\mathbf{p} = {}^0\mathbf{p} - {}^0\mathbf{t}_1 = \mathbf{p} - \mathbf{t}_1$

Rotations



- No translation: ${}^{0}\mathbf{t}_1 = \mathbf{t}_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$
- Both systems right-handed
- $\mathbf{i}_1, \mathbf{j}_1, \mathbf{k}_1$ are the unit vectors of reference system 1 expressed in reference system 0
- Given $\mathbf{p} = {}^{0}\mathbf{p}$, what is ${}^{1}\mathbf{p}$?

Rotations



$$\mathbf{p} = {}^1x\ \mathbf{i}_1 + {}^1y\ \mathbf{j}_1 + {}^1z\ \mathbf{k}_1$$

$$ {}^1x = \mathbf{i}_1^T\mathbf{p} \quad , \quad {}^1y = \mathbf{j}_1^T\mathbf{p} \quad , \quad {}^1z = \mathbf{k}_1^T\mathbf{p}$$

$$ {}^1\mathbf{p} = \begin{bmatrix} {}^1x \\ {}^1y \\ {}^1z \end{bmatrix} = \begin{bmatrix} \mathbf{i}_1^T\mathbf{p} \\ \mathbf{j}_1^T\mathbf{p} \\ \mathbf{k}_1^T\mathbf{p} \end{bmatrix} = R_1\,\mathbf{p}$$

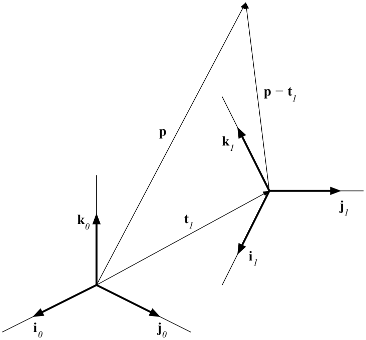
where $R_1 = {}^0R_1 = \begin{bmatrix} \mathbf{i}_1^T \\ \mathbf{j}_1^T \\ \mathbf{k}_1^T \end{bmatrix}$ (unit vectors are the *rows*)

Rotations in General

- More generally, $^{b}\mathbf{p} = {}^{a}R_{b}\,{}^{a}\mathbf{p}$ where ${}^{a}R_{b} = \begin{bmatrix} {}^{a}\mathbf{i}_{b}^{T} \\ {}^{a}\mathbf{j}_{b}^{T} \\ {}^{a}\mathbf{k}_{b}^{T} \end{bmatrix}$
- Rotations are reversible, so there exists ${}^{b}R_{a} = {}^{a}R_{b}^{-1}$
- ${}^{b}R_{a} = {}^{a}R_{b}^{T}$ because ${}^{a}R_{b}$ is orthogonal
- Cross-product is covariant with rotations:

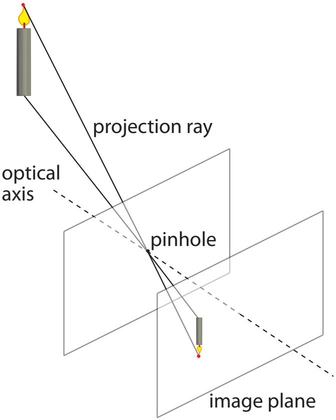
  $(R\mathbf{a}) \times (R\mathbf{b}) = R\,(\mathbf{a} \times \mathbf{b})$

Coordinate Transformation

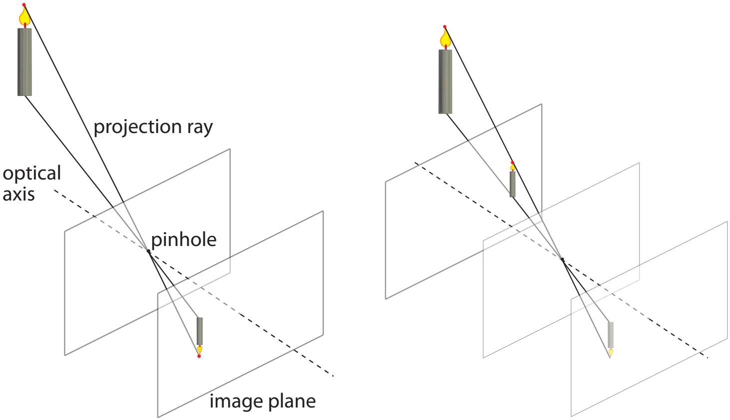


- A.k.a. rigid transformation
- First translate, then rotate: $^1\mathbf{p} = R_1(\mathbf{p} - \mathbf{t}_1)$
- Inverse: $\mathbf{p} = R_1^T\,{}^1\mathbf{p} + \mathbf{t}_1$
- Generally, if $^b\mathbf{p} = {}^aR_b({}^a\mathbf{p} - {}^a\mathbf{t}_b)$ then $^a\mathbf{p} = {}^bR_a({}^b\mathbf{p} - {}^b\mathbf{t}_a)$ where $^bR_a = {}^aR_b^T$ and $^b\mathbf{t}_a = -{}^aR_b{}^a\mathbf{t}_b$

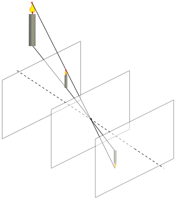
The Pinhole Camera

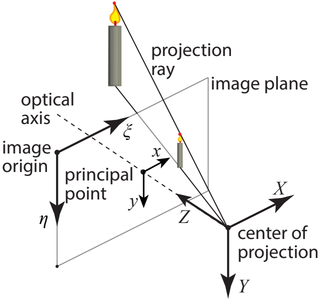


Putting the Image Plane in Front?



In Math, We Can





- *Camera reference system* (X, Y, Z) is right-handed, Z toward scene
- Distance btw center of projection and principal point: *focal distance f*
- *Canonical image reference system* (x, y) has origin at principal point
- *Pixel image reference system* (ξ, η) has origin at top left of sensor
- $\xi = s_x x + \xi_0$ and $\eta = s_y y + \eta_0$ $(s_x, s_y$ in pixels/mm)

The Projection Equations