17.0/14.0

L17 - Transformations 3 (old L14) Outline

- Review: so far (Slides)

  2x2 linear T_x
  Homogeneous coords.
  Affine t_x

- Points vs Directions in homogeneous coords (Whiteboard)

  ex: plane = point + normal
  apply t_x, what goes wrong?

$$\vec{p} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad \vec{n} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$T = \begin{pmatrix} I_{2\times 2} & \begin{matrix} 1 \\ 0 \end{matrix} \\ 0 \; 0 & 1 \end{pmatrix}$$

  Try this!

- Similarity T_x (revisit change-of-frame view) - (Slides)

- Inverses (whiteboard)

- 3D Tx - (Slides)

- Rotations in 3D -
  - euler angles (slides → whiteboard)
  - axis-angle (whiteboard)
  - mention quaternions

- Transforming normals (whiteboard)

Linear algebra facts:

$$(Ax)^T = x^T A^T$$

$$A A^{-1} = I$$

$$\vec{n}^T \vec{t} = 0$$

$$(N \vec{n})^T (M \vec{t}) = 0$$

$$\vec{n}^T \underbrace{N^T M}_{I} \vec{t} = 0$$

$$N^T M = I$$

$$N^T M M^{-1} = M^{-1}$$

$$N^T = M^{-1}$$

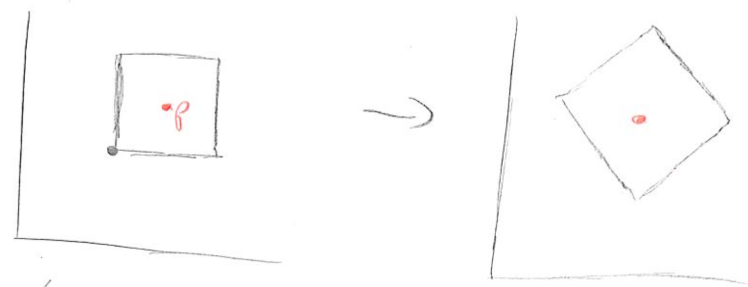
$$N = M^{-T}$$

14.1 ~~13.3~~

Composing 2D Affine Transformations

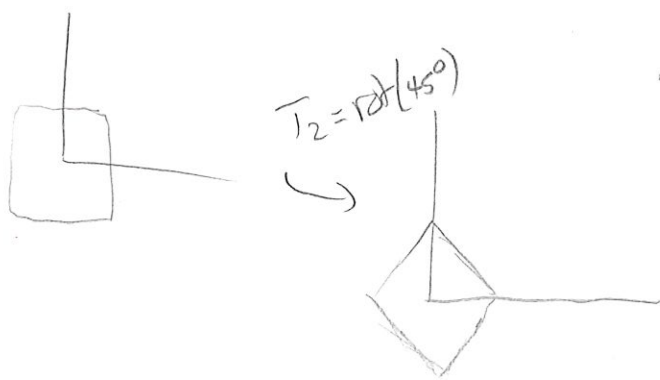
Matrix mutiplication - Just Works

Example: Rotate around a non-origin point P



$T_1 = \text{translate}(-p)$

$T_3 = \text{translate}(p)$



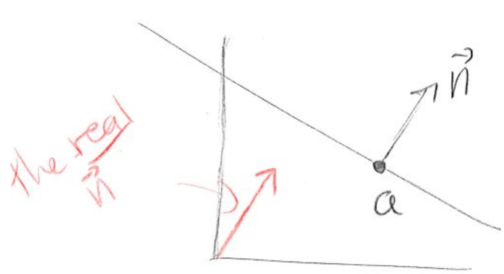
$$\boxed{T = T_3 T_2 T_1}$$

right-to-left!

Points vs Directions

13.3

Both 3-vectors, but one is a place, the other is a direction.



Plane through pt $\vec{a}$ w/ normal vector $\vec{n}$

Points get moved by translations

Directions don't

$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_x \\ a_y \\ 1 \end{bmatrix} = \begin{bmatrix} a_x + 1 \\ a_y + 1 \\ 1 \end{bmatrix}$$

$$\vec{v} = \vec{p} - \vec{q}$$

$$\begin{bmatrix} v_x \\ v_y \\ 0 \end{bmatrix} = \begin{bmatrix} p_x \\ p_y \\ 1 \end{bmatrix} - \begin{bmatrix} q_x \\ q_y \\ 1 \end{bmatrix}$$ neat!

$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} n_x \\ n_y \\ 0 \end{bmatrix} = \begin{bmatrix} n_x \\ n_y \\ 0 \end{bmatrix}$$

turn off translation for vectors that indicate directions

14.2 ~~13.4~~

Affine Composition - Similarity Transforms:

1. Transform to a frame where it's easy
2. Do the easy thing
3. Transform back.

Let $F = \begin{bmatrix} \vec{u} & \vec{v} & \vec{p} \\ 0 & 0 & 1 \end{bmatrix}$, frame where desired transformation T is easy to write.

this is frame-to-canonical!

Canonical-to-frame is F^{-1}, so whole transform is

$$\overset{3}{F}\ \overset{2}{T}\ \overset{1}{F^{-1}}$$

Inverses?

Can compute algebraically, but sometimes it's easy:

- $\text{translate}(\vec{t})^{-1} = \text{translate}(-\vec{t})$
- $\text{rotate}(\theta)^{-1} = \text{rotate}(-\theta)$
- $\text{scale}(s_x, s_y)^{-1} = \text{scale}(1/s_x, 1/s_y)$
- inverse of frame w/ orthonormal basis (rigid tx)

$$\begin{array}{cc} 2\times2 & 2\times1 \end{array}$$

$$\begin{bmatrix} Q & \vec{u} \\ 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} Q^T & -Q^T\vec{u} \\ 0 & 1 \end{bmatrix}$$

$$\begin{array}{cc} 1\times2 & 1\times1 \end{array}$$

(Math fact:) Inverse of an orthogonal matrix is its transpose.

Rotations in 3D

Euler angles - composition of 3 axis-aligned rotations.

Example $\quad R_z R_x R_z$

- 12 possible sequences of 3 axis rotations!
- Also: can choose intrinsic vs extrinsic

intrinsic: axes of rotation stick with object as it rotates

extrinsic: axes of rotation are fixed to the global coordinate system

Yikes! See wikipedia. Pick a convention and stick with it.

Axis-angle

Rotate θ about axis $\vec{a}$

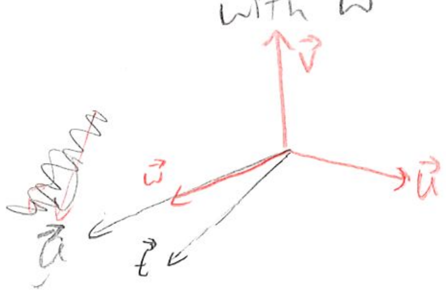
How? Change coordinate frame so $\vec{a}$ is the "z" axis, rotate around z, change coordinate frame back. — Similarity transform!

Basis for frame: $\quad \vec{w} \leftarrow \dfrac{\vec{a}}{\|\vec{a}\|}$

Pick a $\vec{t}$ not collinear with $\vec{w}$ $\quad \vec{t} = w$, but with smallest component set to 1

$u \leftarrow \dfrac{\vec{t} \times \vec{w}}{\|\vec{t} \times \vec{w}\|}$

$v \leftarrow \vec{w} \times \vec{u}$

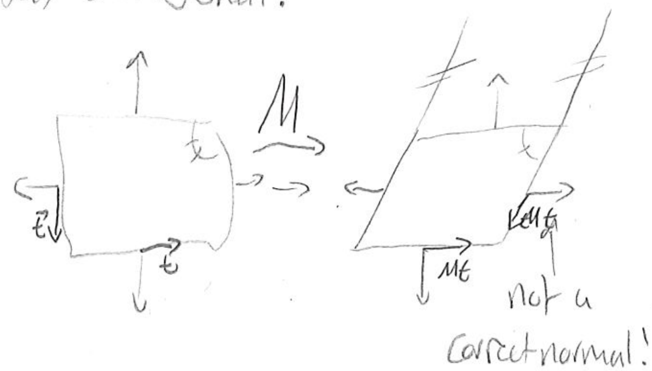


15.1 ~~14.4~~

Transforming Normals

Affine: parallel stays parallel

orthogonal does not stay orthogonal.



If we stored tangent vectors, t

they'd remain correct:

$$Mt$$

Want a matrix N s.t. Nn is orthogonal to Mt

$$\vec{n}^T \vec{t} = 0$$

$N\vec{n}_{orig}$ $\quad$ $M\vec{t}_{orig}$

Let's play some algebra tricks:

$$\vec{n}^T \vec{t} = \vec{n}^T \underbrace{I}_{\text{identity}} \vec{t}$$

$$= \vec{n}^T M^{-1} M \vec{t} = 0$$

$$= \underbrace{(\vec{n}^T M^{-1})}_{\text{these are the normals we want}} (M\vec{t}) = 0$$

$$(N\vec{n})^T (M\vec{t}) = 0$$

$$\vec{n} \cdot \vec{t} = 0$$

$$(\vec{n}^T N^T)(M\vec{t}) = 0$$

$$\vec{n}^T \underbrace{N^T M}_{I} \vec{t} = 0$$

$$N^T M = I$$

$$N^T M M^{-1} = M^{-1}$$

$$\boxed{N = M^{-T}}$$

Transpose it: $M^{-T}\vec{n}$ is orthogonal to the transformed tangent vectors.

To get transformed normals, multiply them by the inverse transpose of the transformation matrix.