

LECTURE 7

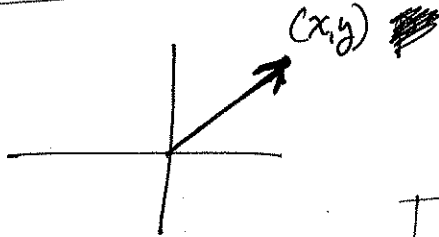
6.1.1-6.1.8
6.2.0-6.2.1
6.3
-6.5

geometric transformations:

- rotation
- translation
- scaling
- projection
- shear

can be accomplished with
transformation matrices

6.1 | 2D linear transformation



a point represented as an offset from the origin.

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$$

2D
linear
transformation

⊗ Linear

$$A(\underline{u} + \underline{v}) = A\underline{u} + A\underline{v}$$

$$A(\alpha \underline{u}) = \alpha A\underline{u}$$

$$A \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a_{11}x + a_{12}y \\ a_{21}x + a_{22}y \end{pmatrix} = \begin{pmatrix} x' \\ y' \end{pmatrix}$$

A maps one 2D vector to another.

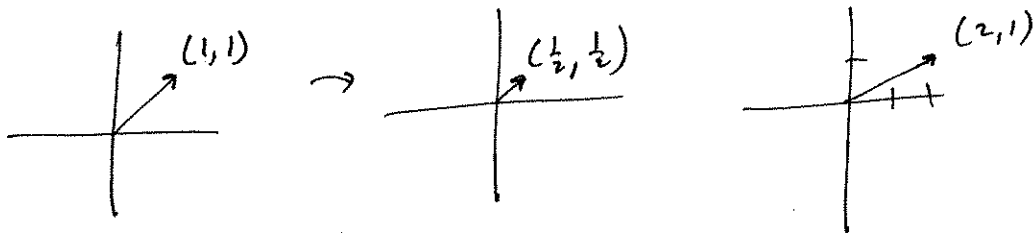
Choice of A entries yield diff. transformations.

6.1.1 | Scaling

$$\begin{pmatrix} S_x & 0 \\ 0 & S_y \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} S_x x \\ S_y y \end{pmatrix}$$

examples of
uniform &
non uniform

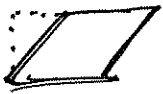
examples



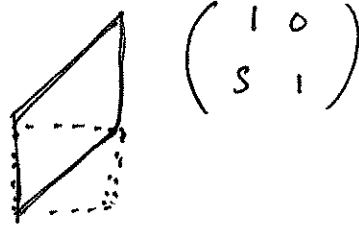
6.1.2 | Shearing.

"deck of cards"

horizontal shear $\begin{pmatrix} 1 & s \\ 0 & 1 \end{pmatrix}$



vertical shear

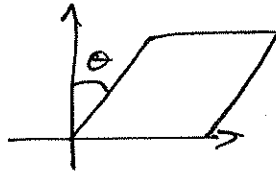


$$\begin{pmatrix} 1 & 0 \\ s & 1 \end{pmatrix}$$

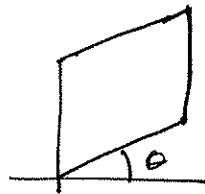
$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} 45^\circ$$

Generally,

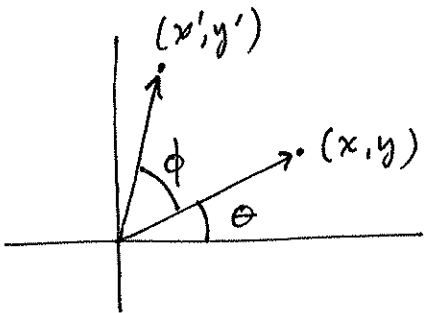
$$\begin{pmatrix} 1 & \tan \theta \\ 0 & 1 \end{pmatrix}$$



$$\begin{pmatrix} 1 & 0 \\ \tan \theta & 1 \end{pmatrix}$$



6.1.3 | Rotation.



rotates
counterclockwise

to rotate clockwise by ϕ ,
rotate c.w. by $-\phi$

$$(x, y) = (\cos \theta, \sin \theta)$$

$$(x', y')^T = \begin{pmatrix} \cos(\theta + \phi) \\ \sin(\theta + \phi) \end{pmatrix} = \begin{matrix} \cos \theta \cos \phi - \sin \theta \sin \phi \\ \cos \theta \sin \phi + \sin \theta \cos \phi \end{matrix}$$

$$= \begin{pmatrix} x \cos \phi - y \sin \phi \\ x \sin \phi + y \cos \phi \end{pmatrix} =$$

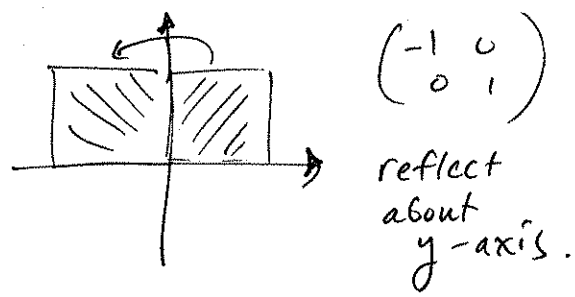
$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos \phi & -\sin \phi \\ \sin \phi & \cos \phi \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

rotate(ϕ)

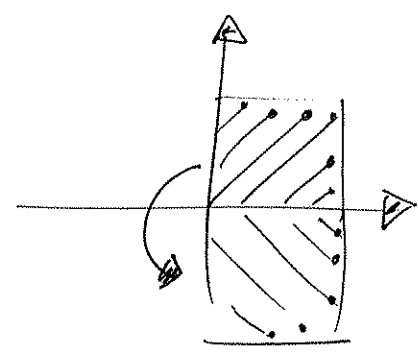
$$R(\phi)$$

⊗ Rotation matrix is an orthogonal matrix. $R(\phi)^T R(\phi) = R(\phi) R(\phi)^T = I$

6.1.4 | Reflection

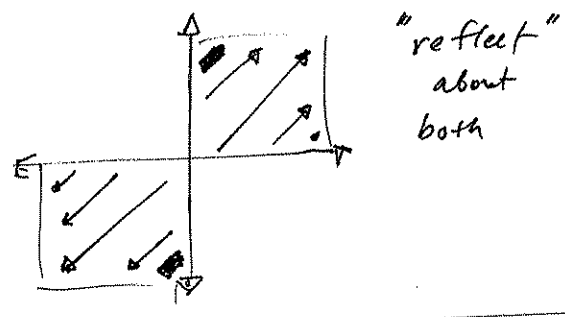


$$\begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$$



$$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

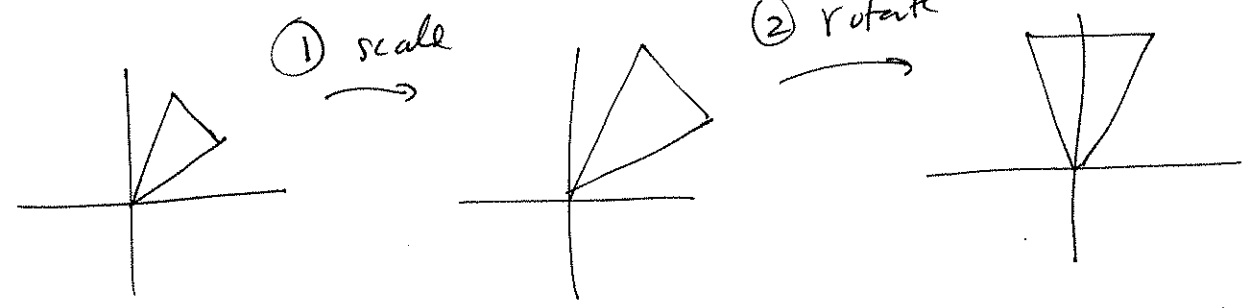
can't achieve without rigidly moving through 3D.



$$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} = \begin{pmatrix} \cos \pi & -\sin \pi \\ \sin \pi & \cos \pi \end{pmatrix} = R(\pi)$$

R^2

6.1.5 | Composition



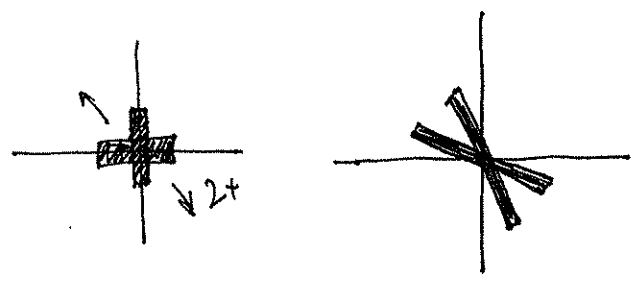
$$\underline{v} = R S \underline{u} = R(S \underline{u}) = (RS) \underline{u}$$

↑ associativity

Composition — Not Commutative!!!

Example: How to do this?

$$R(\frac{\pi}{4}) S(1,2) R(-\frac{\pi}{4})$$



$$R^T S R$$

Q. what if we reversed order?

6.3 Translation and Affine Transformation

Linear transformation always maps $0 \rightarrow 0$.

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$$

$$x' = x + t_x$$

$$y' = y + t_y$$

→ can't do this w/ a 2×2 matrix.

→ use matrix + vector

→ or, use 3×3 matrix + homogeneous coordinates.

$$\begin{pmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$

↑ "homogeneous coordinates"

• vectors that represent directions

$$\begin{pmatrix} x \\ y \\ 0 \end{pmatrix}$$

← don't get translated.

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

position

$$\begin{pmatrix} x \\ y \\ 0 \end{pmatrix}$$

direction

Similar in 3D

$$\begin{pmatrix} 1 & t_x \\ & 1 & t_y \\ & & 1 & t_z \\ & & & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

↑ translator in 3D. ↑ homogen. coord. in 3D

⊗ Example: Windowing transformation

For perspective viewing, homogeneous coord. will take on values other than 0, or 1.