NYU

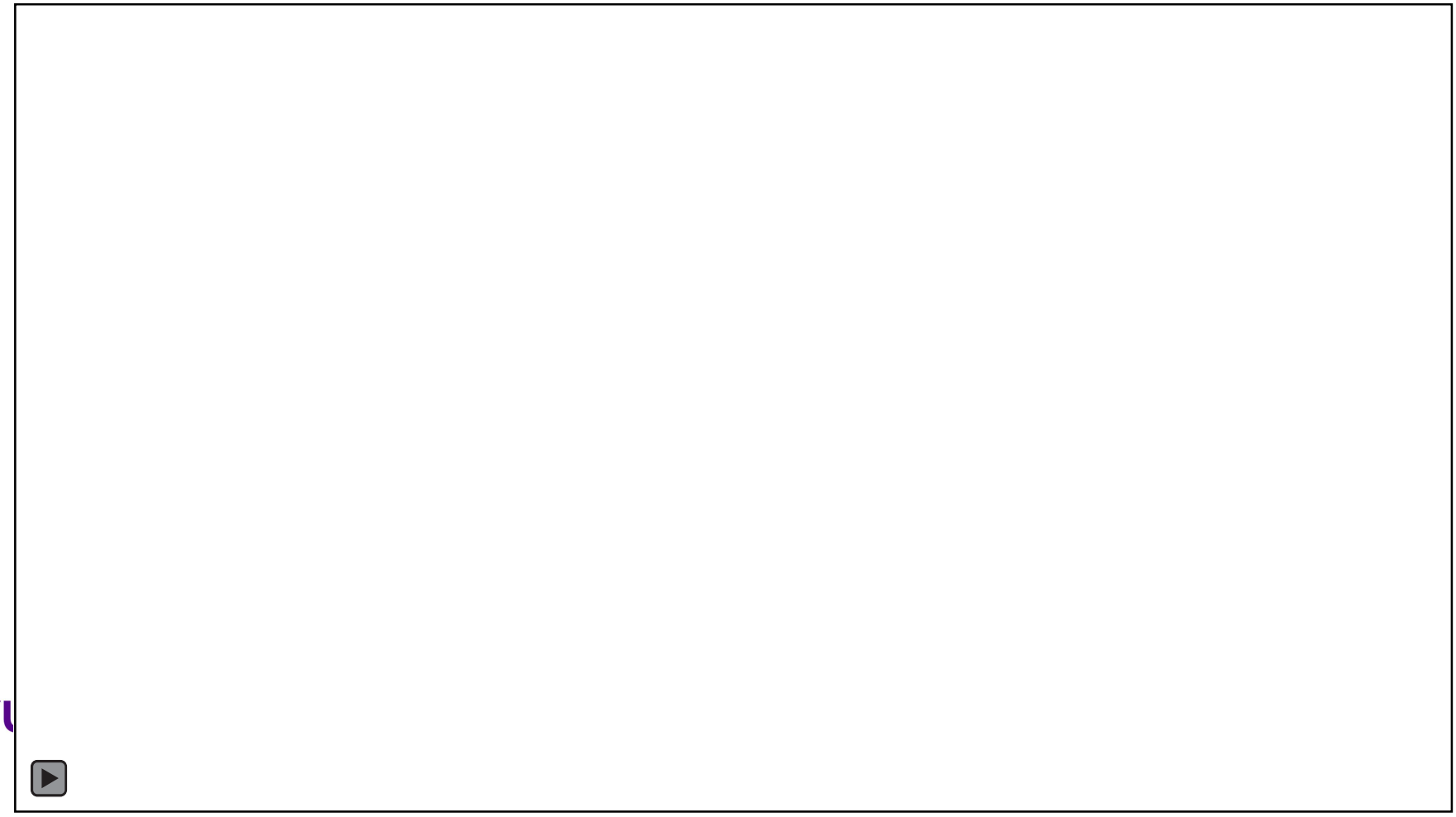
Virtual and Augmented Reality

CS-GY 9223/CUSP-GX 6004

2023 Fall

Prof. Qi Sun
qisun@nyu.edu | www.qisun.me

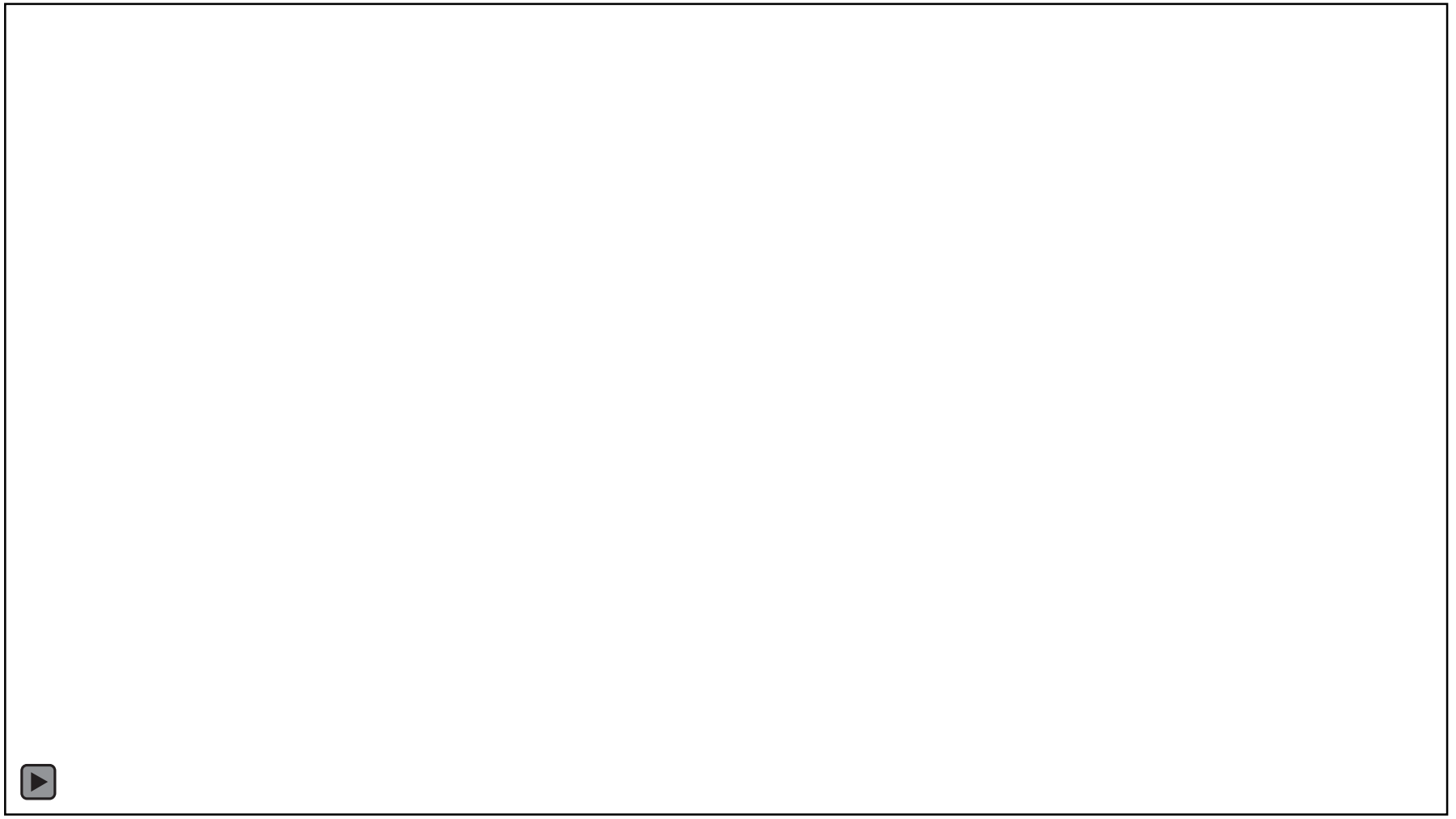
What's Happening in Each Frame of VR/AR?



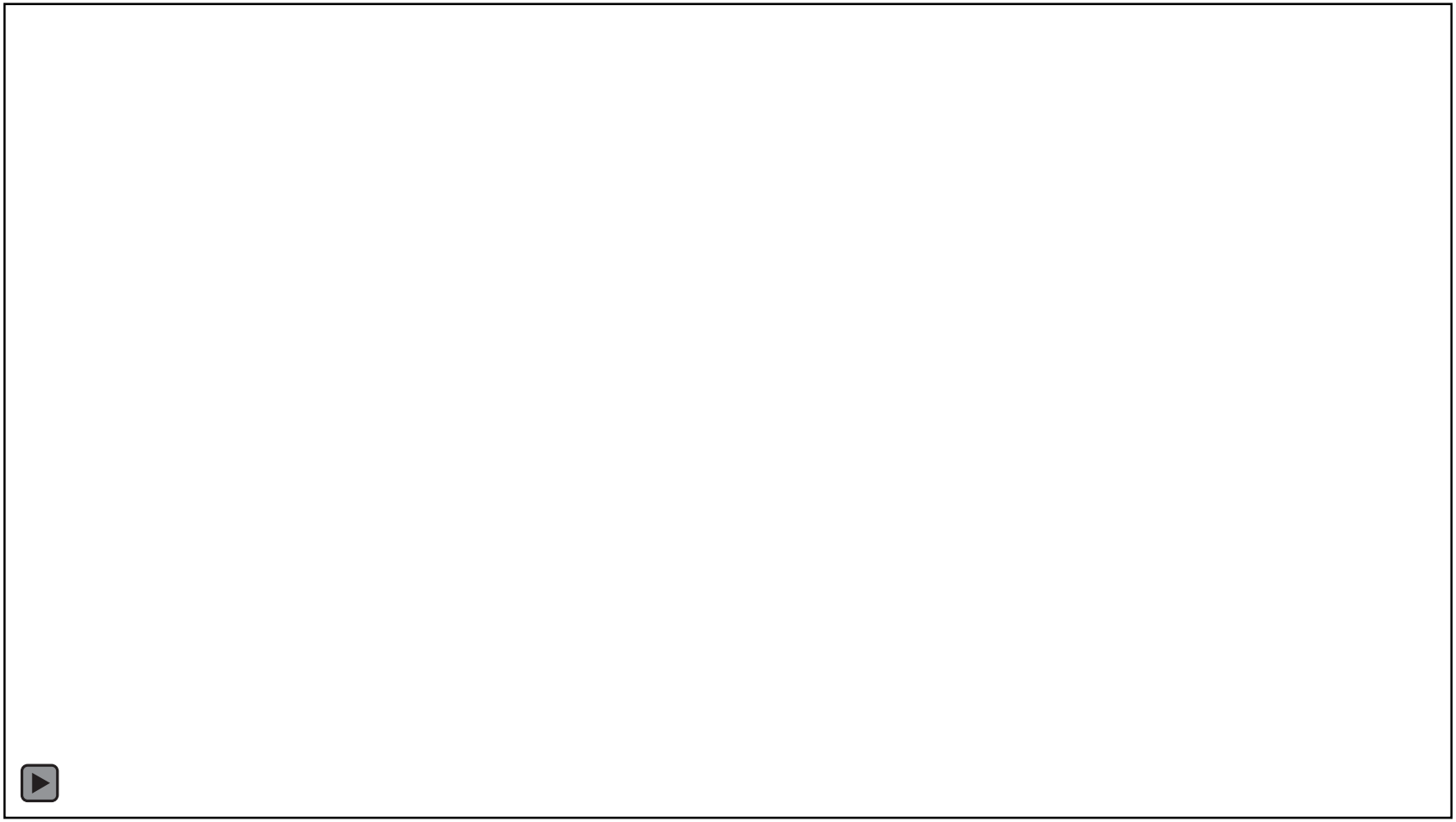
This Lecture

- Transformation and Projection
  - Why transformation?
  - 2D images transformations: rotation, scale, shear
  - Homogeneous coordinates
  - Composing transforms
  - → 3D transformations and projections

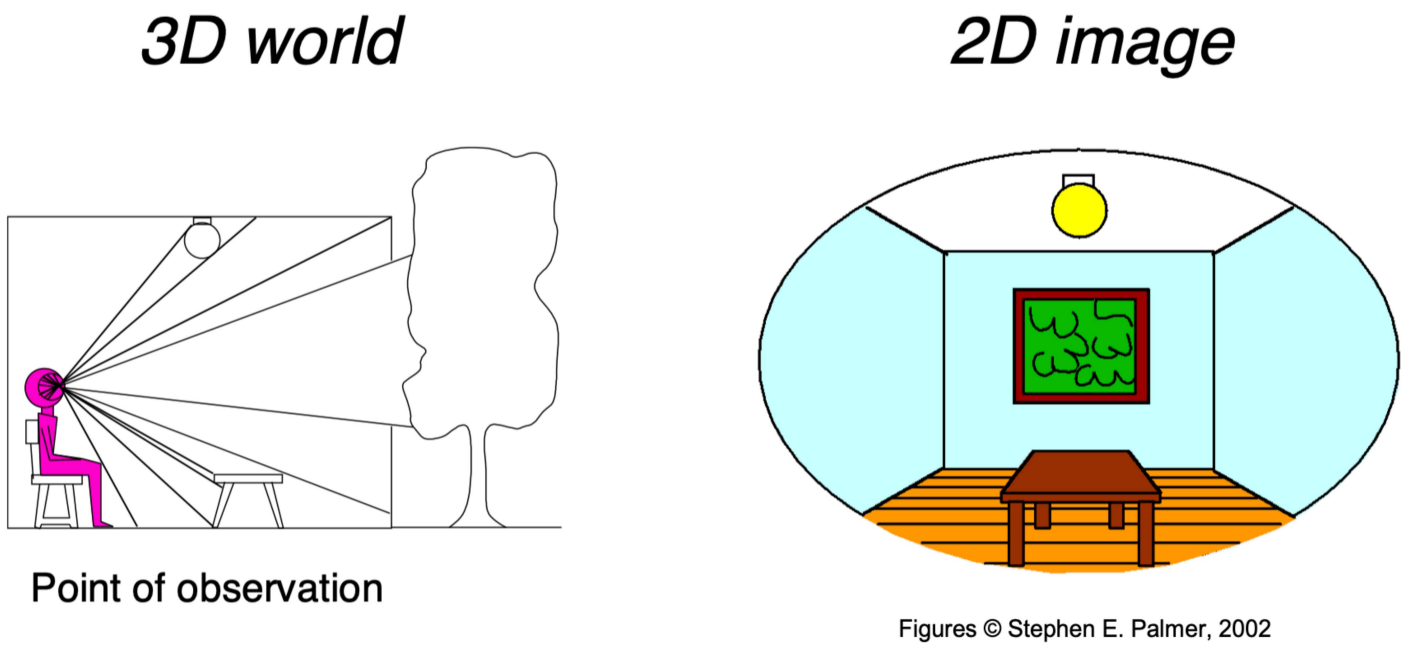
Why Transformation?



Transformation – Translation + Rotation



Transformation



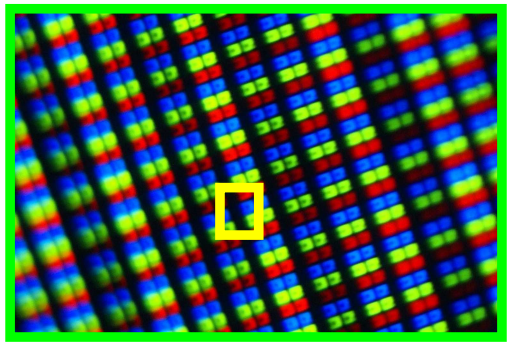
Viewing: (3D to 2D) projection

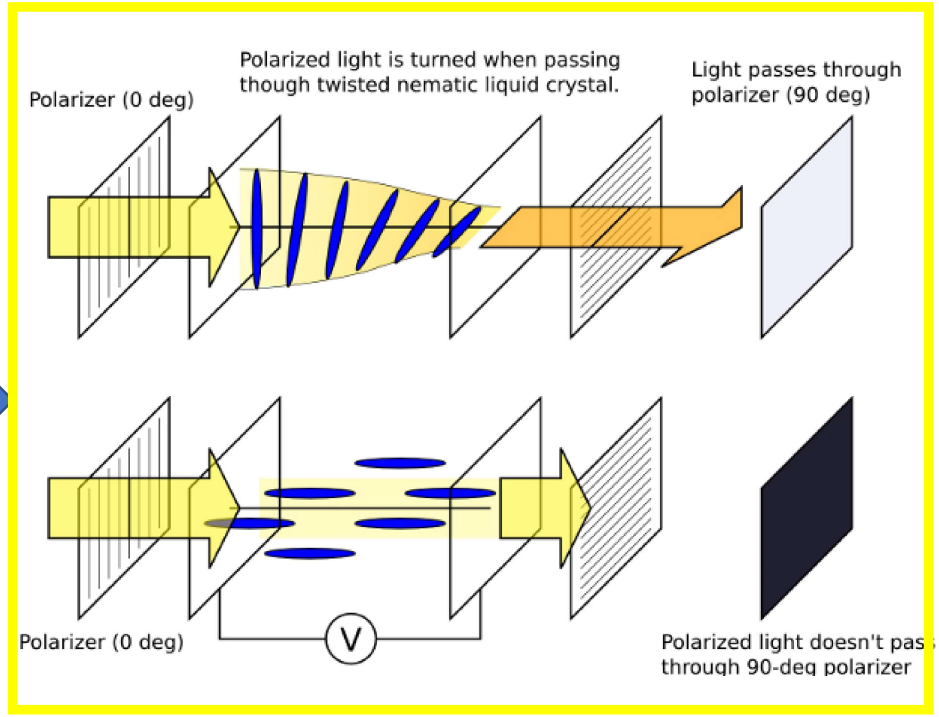
Today

- 2D transformations
  - Representing transformations using matrices
  - Rotation, scale, shear
- Homogeneous coordinates

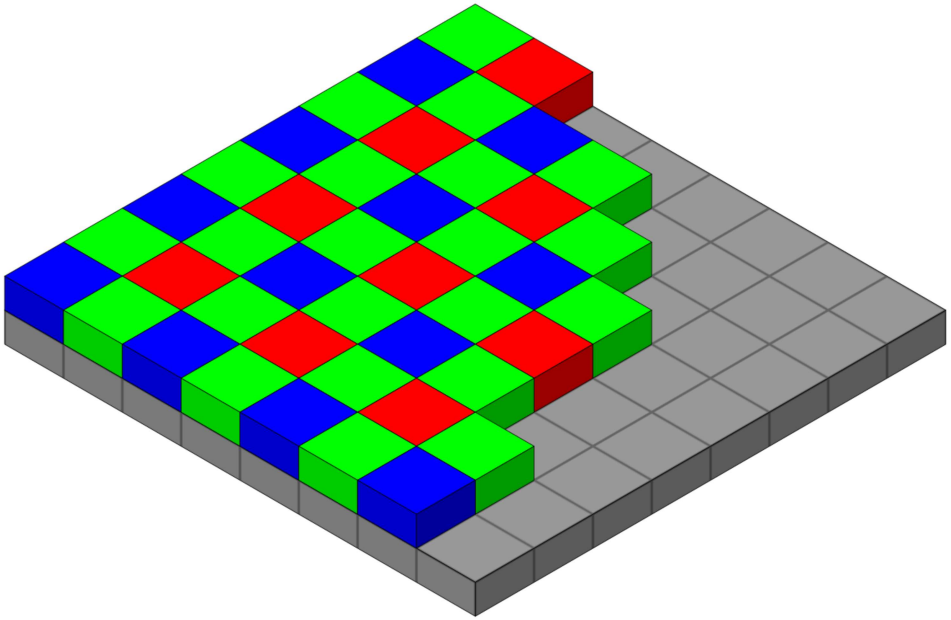
2D Images

- LCD Recap

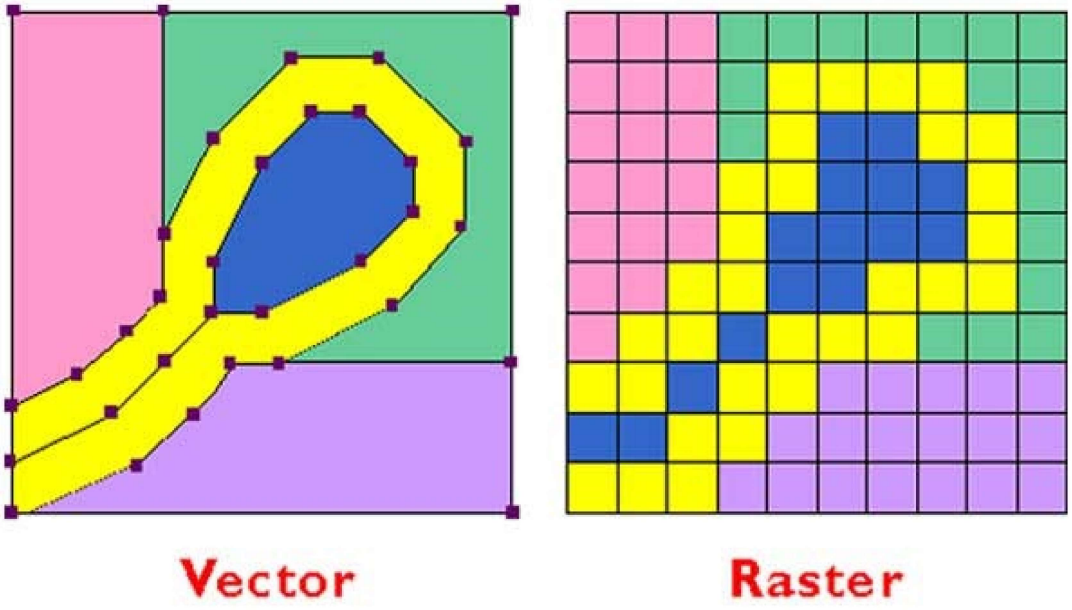




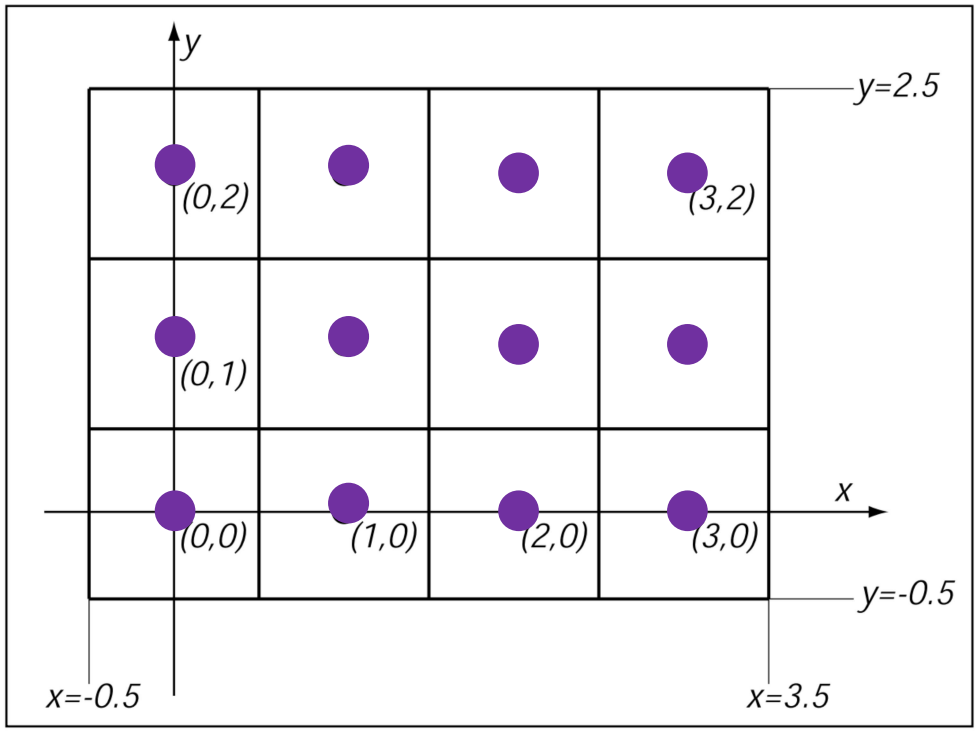
Bayer filter



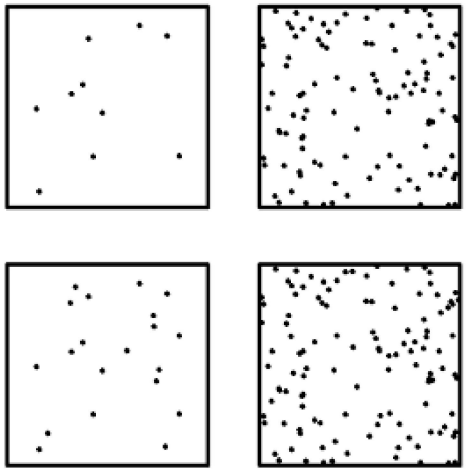
Images – Raster vs. Vector



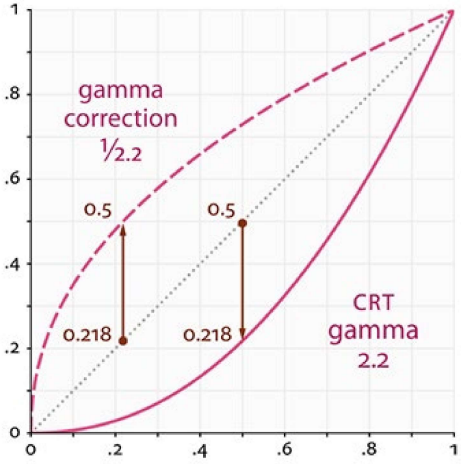
Images – Representation



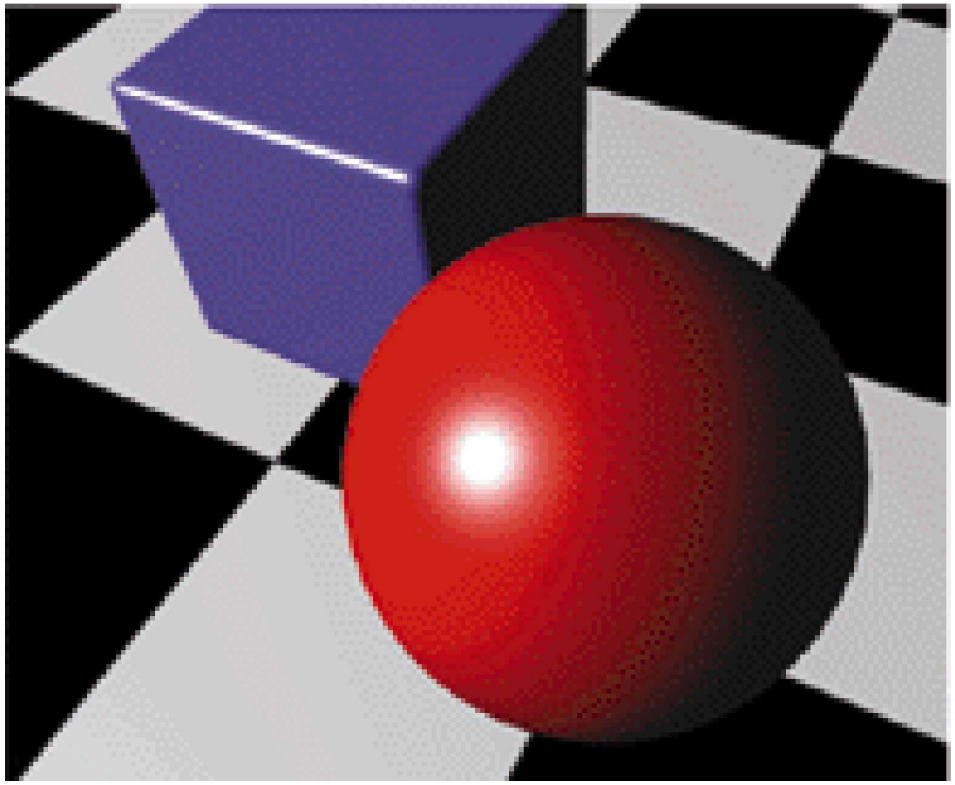
Images – Gamma Correction

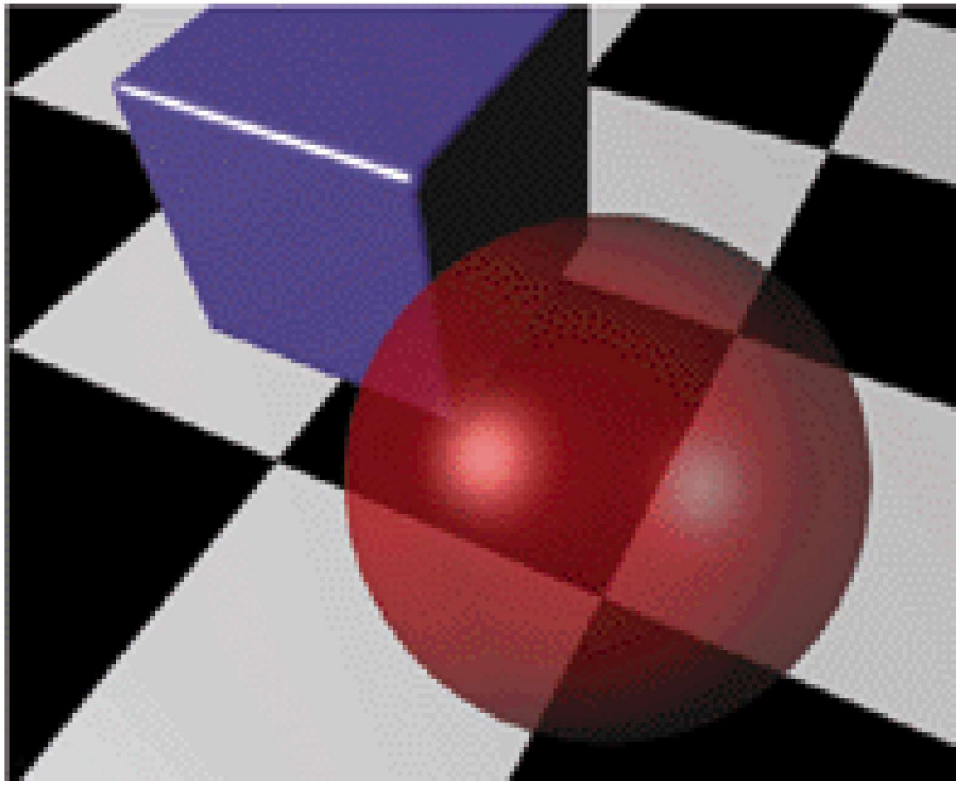


$$S = C \cdot R^{\gamma}$$

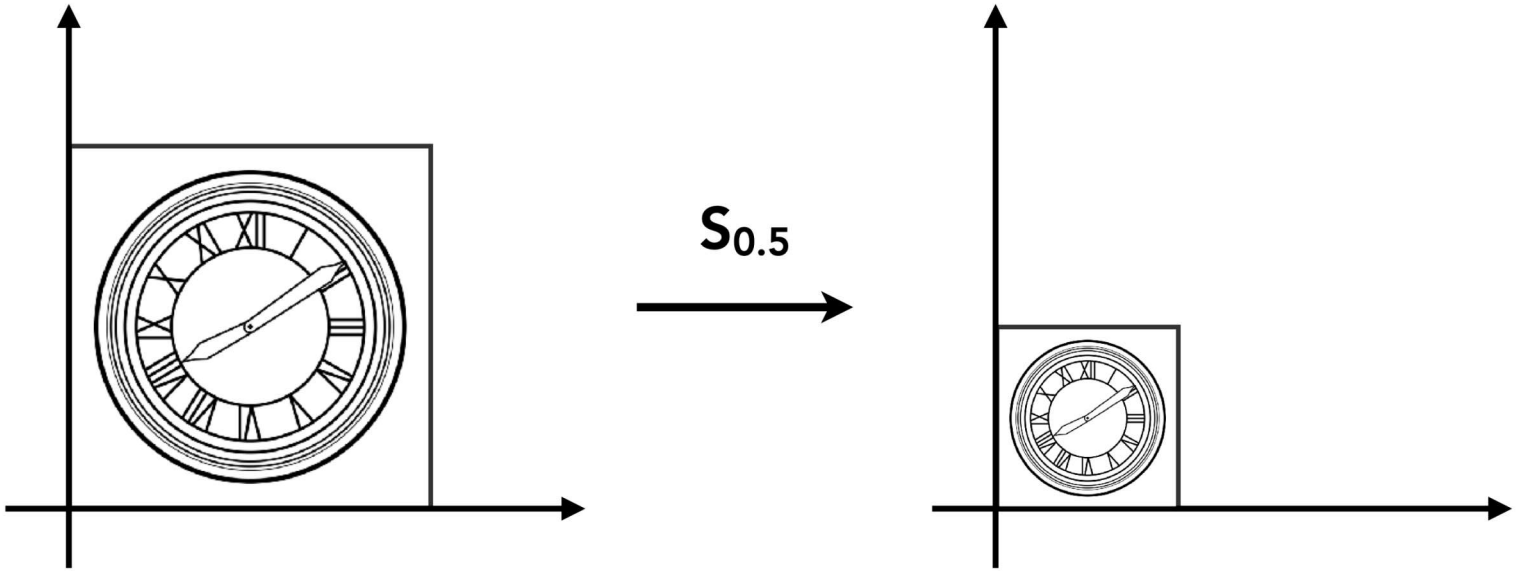


Images – Alpha Blending



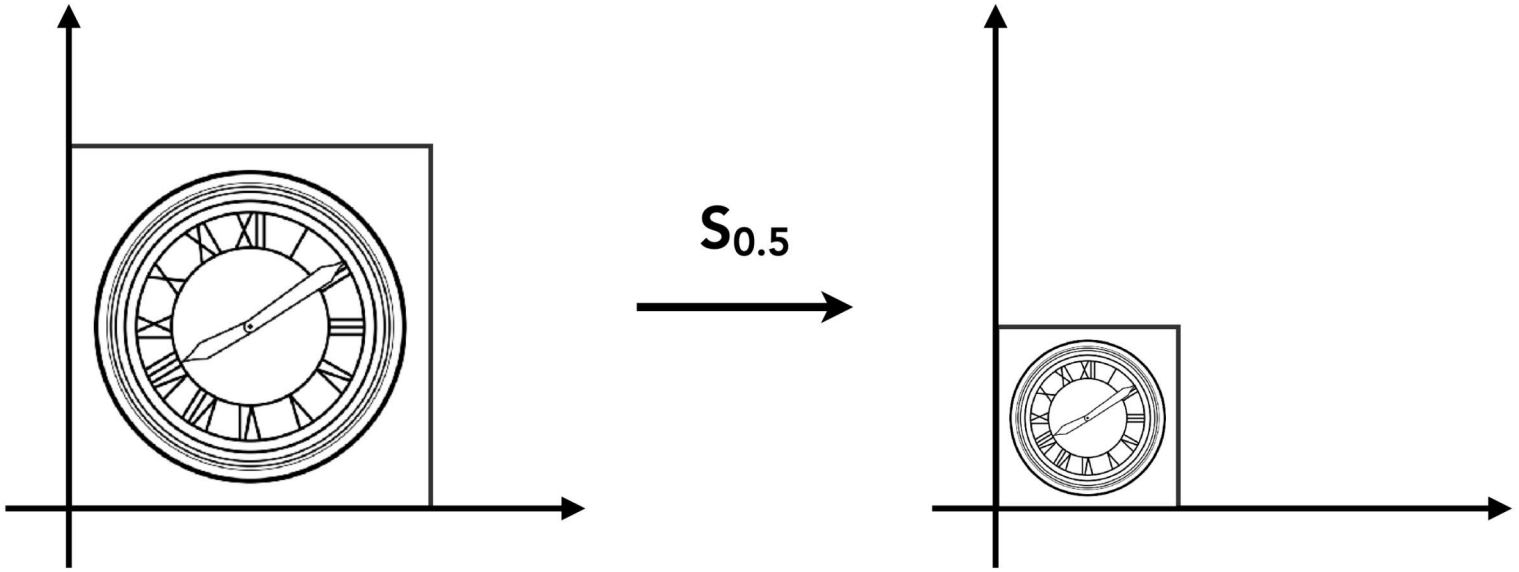


Scale



slide courtesy of Ren Ng

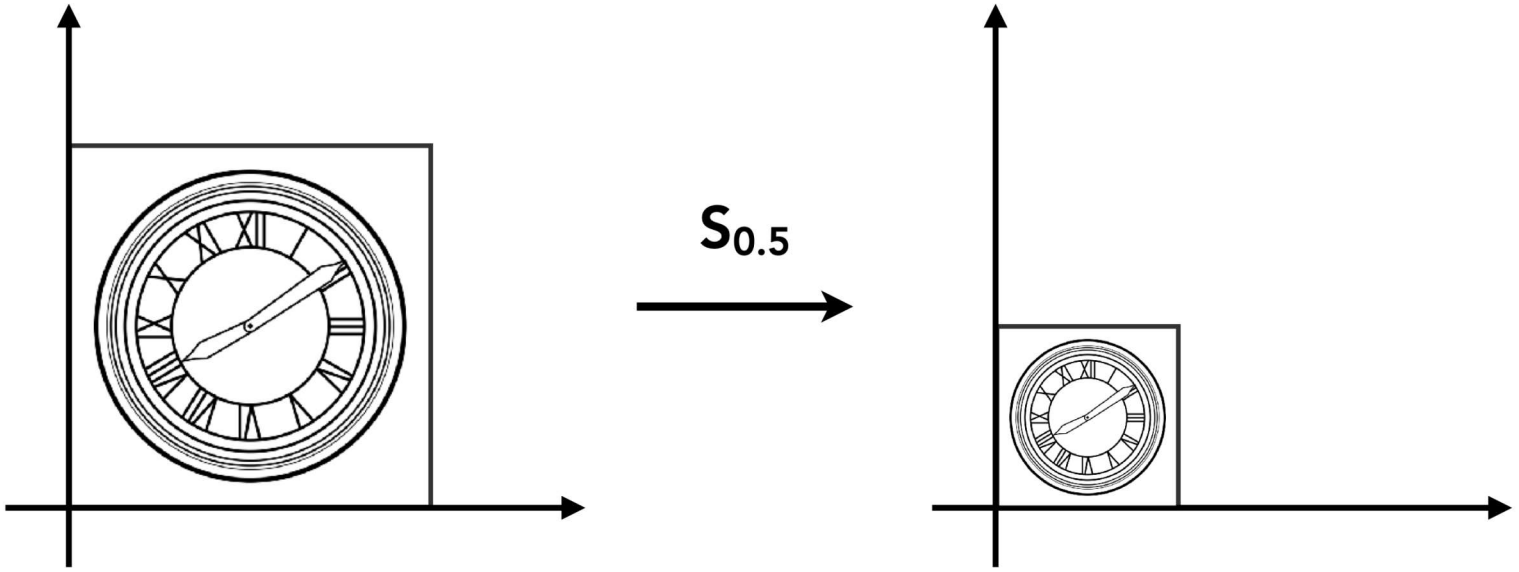
Scale Transform



$$x' = sx$$
$$y' = sy$$

slide courtesy of Ren Ng

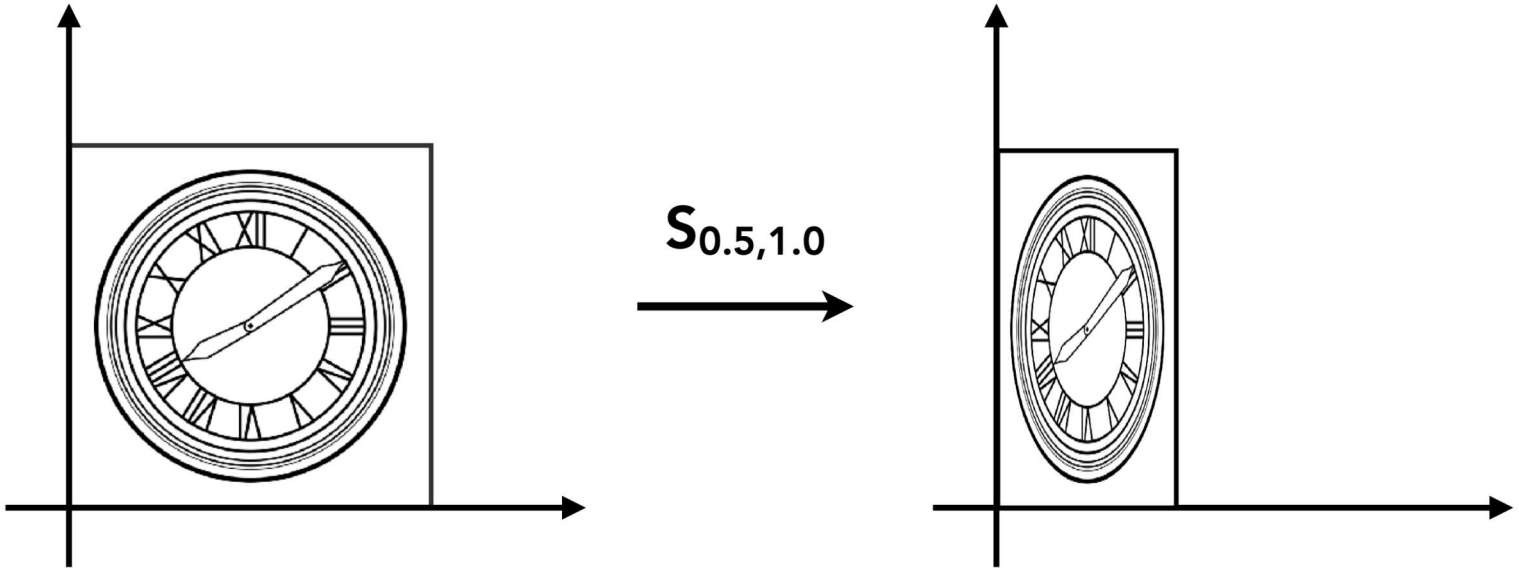
Scale Matrix



$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

slide courtesy of Ren Ng

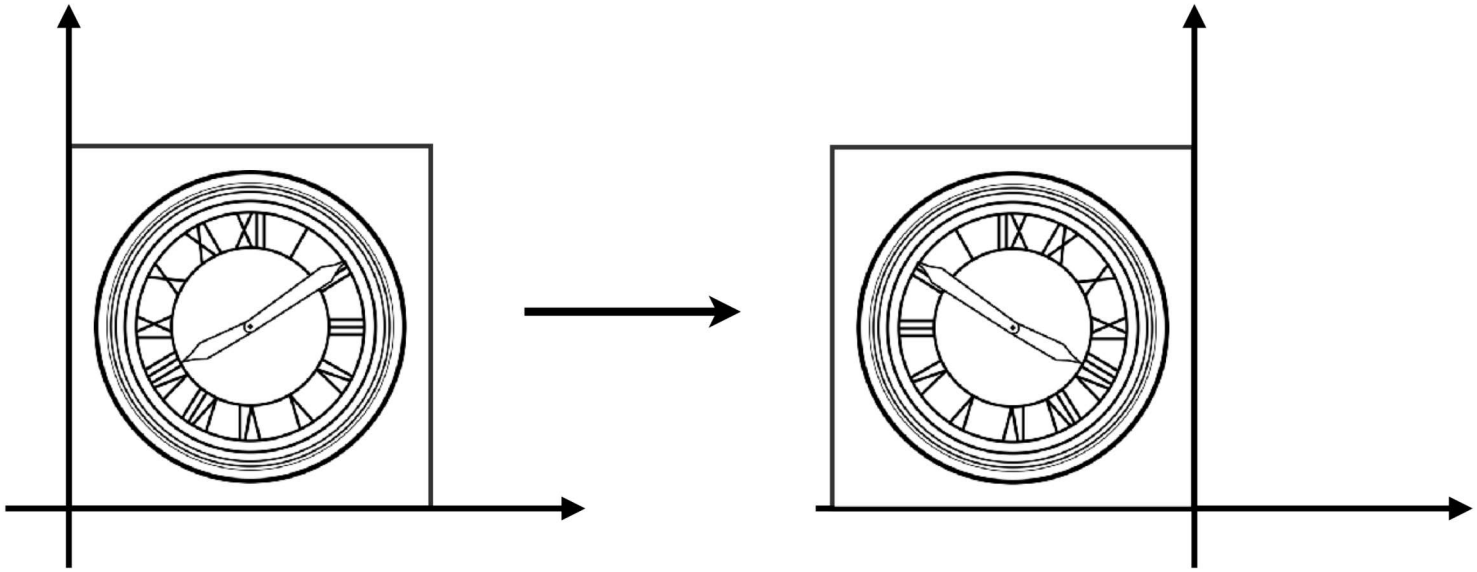
Scale (Non-Uniform)



$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

slide courtesy of Ren Ng

Reflection Matrix



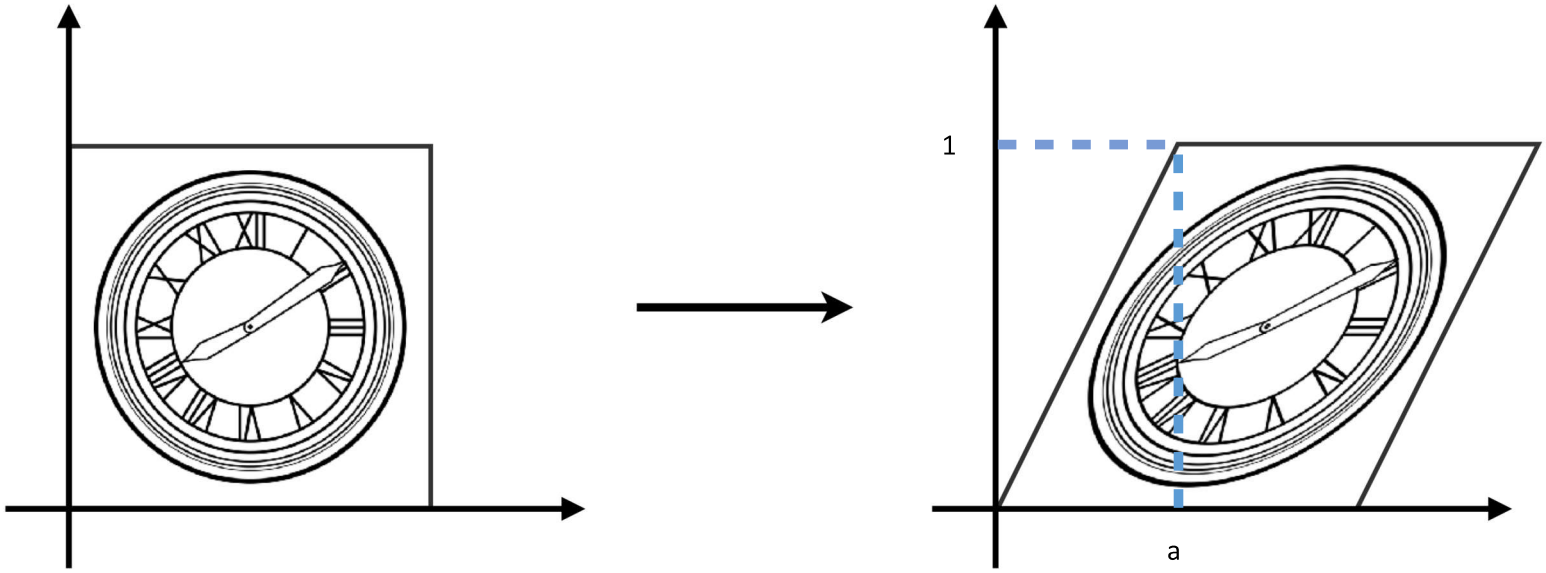
Horizontal reflection:

$$x' = -x$$
$$y' = y$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

slide courtesy of Ren Ng

Shear Matrix



Hints:

Horizontal shift is 0 at y=0

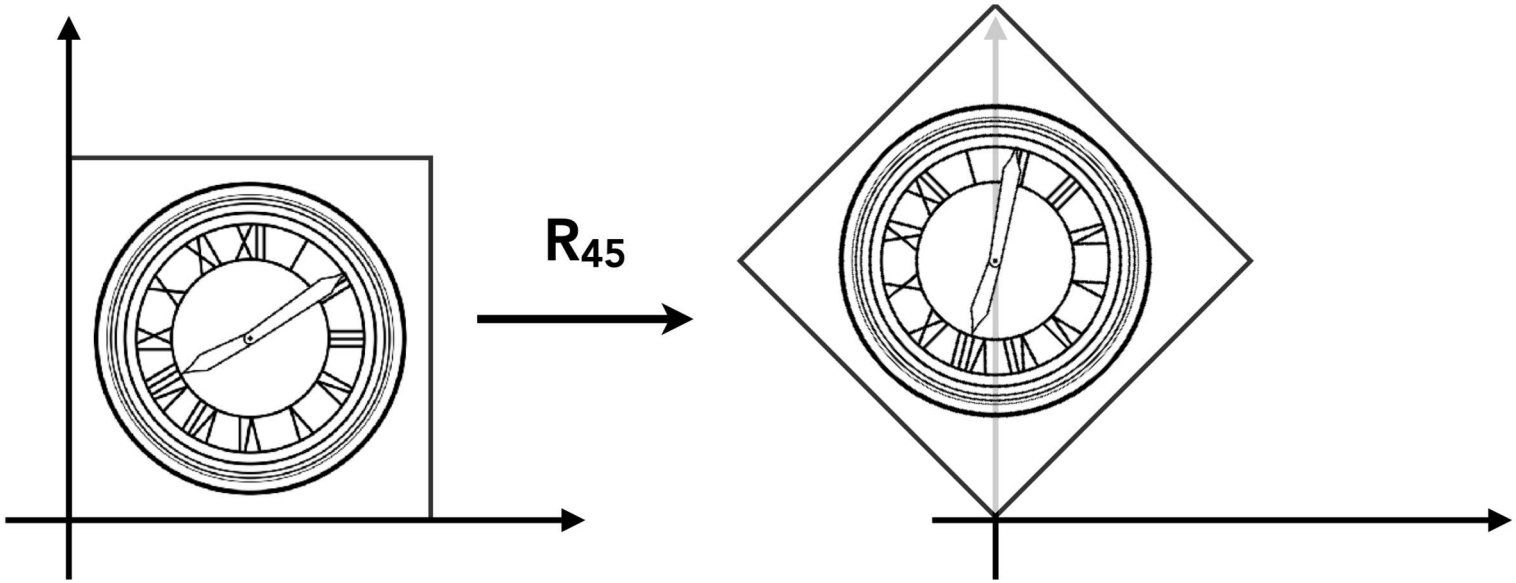
Horizontal shift is a at y=1

Vertical shift is always 0

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & a \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

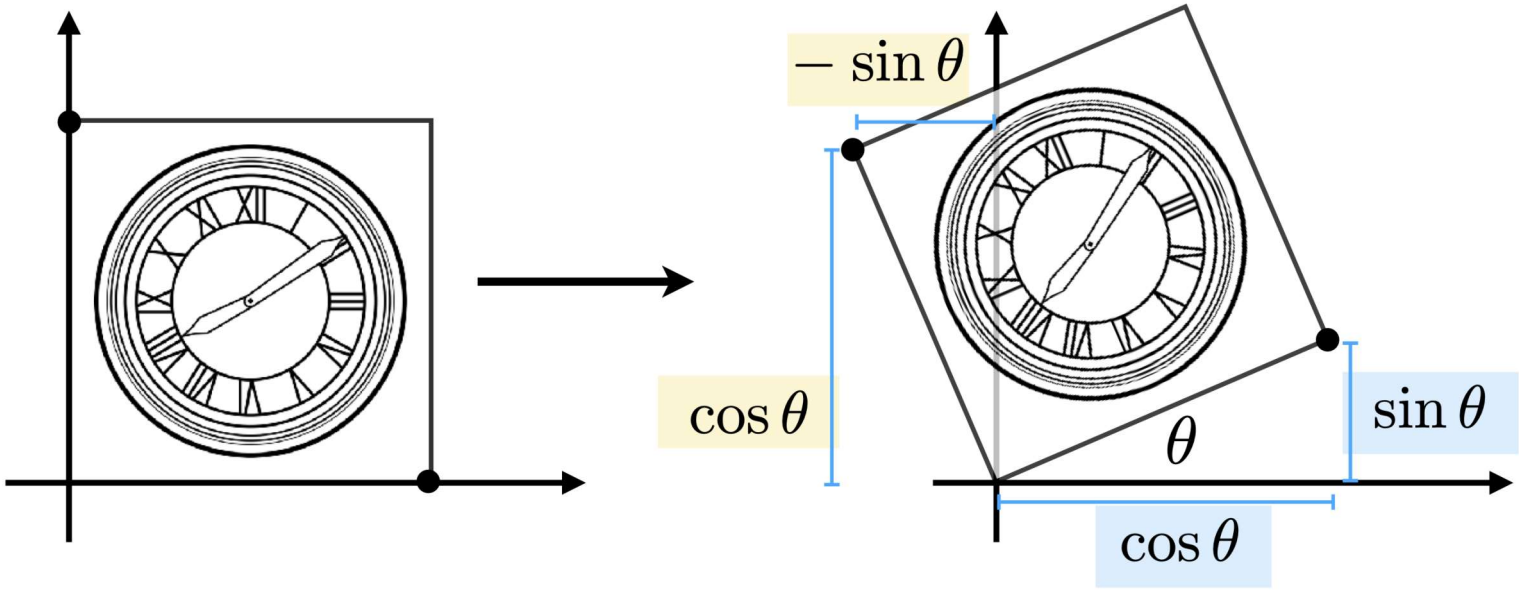
slide courtesy of Ren Ng

Rotate



slide courtesy of Ren Ng

Rotation Matrix



$$\mathbf{R}_\theta = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

slide courtesy of Ren Ng

Linear Transforms = Matrices

(of the same dimension)

$$x' = a\,x + b\,y$$
$$y' = c\,x + d\,y$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

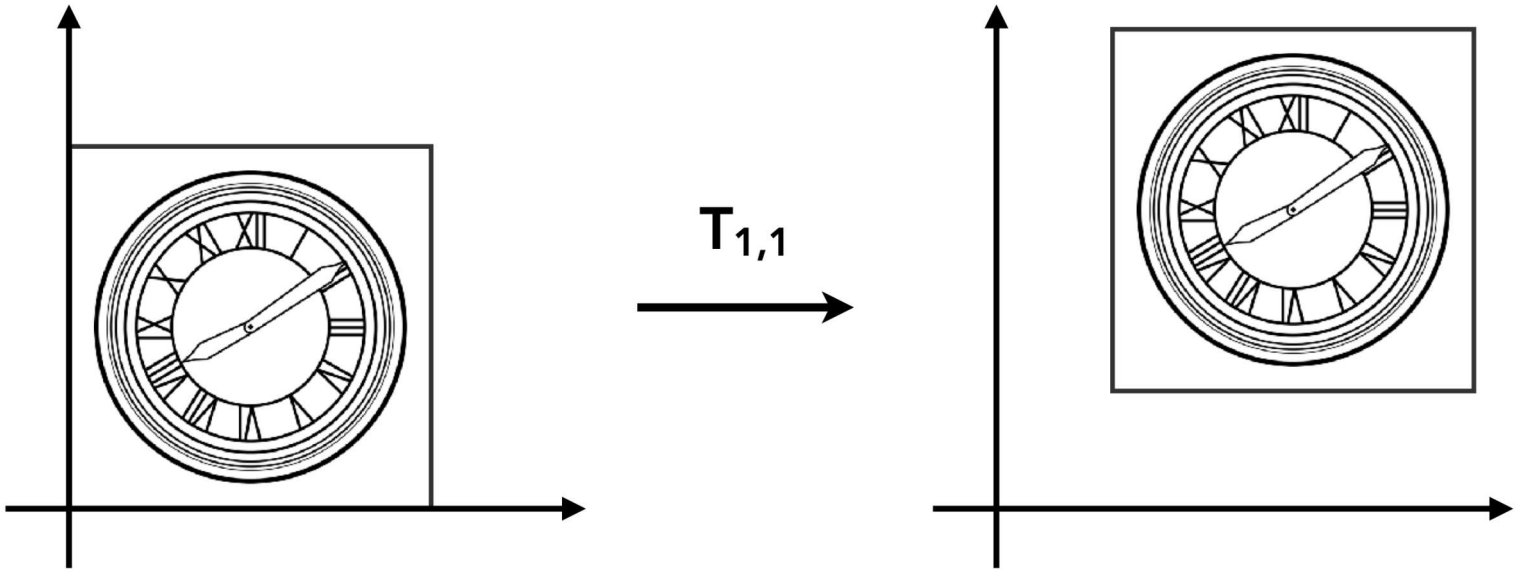
$$\mathbf{x}' = \mathbf{M}\,\mathbf{x}$$

slide courtesy of Ren Ng

Today

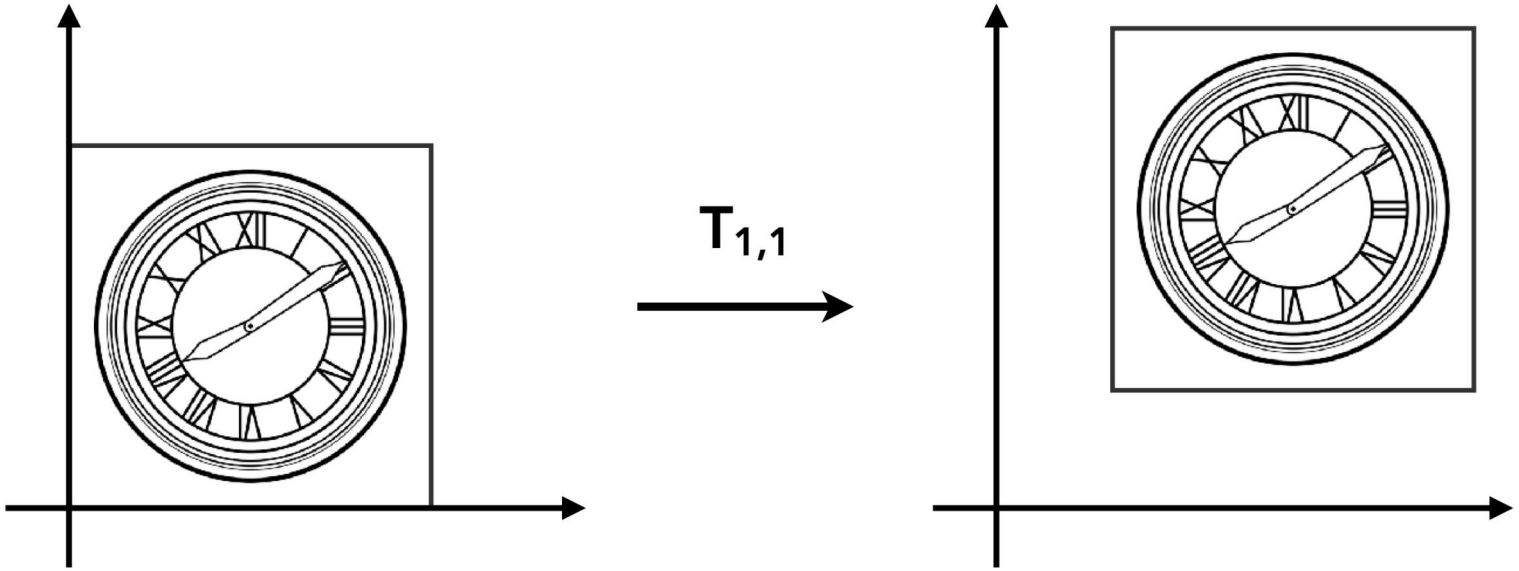
- Recap of matrices
- Why study transformation
- 2D transformations
- Homogeneous coordinates
  - Why homogeneous coordinates
  - Affine transformation

Translate



slide courtesy of Ren Ng

Translation??



$$x' = x + t_x$$
$$y' = y + t_y$$

slide courtesy of Ren Ng

Why Homogeneous Coordinates

- Translation cannot be represented in matrix form

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

(So, translation is NOT linear transform)

- But we don't want translation to be a special case
- Is there a unified way to represent all transformations? (and what's the cost?)

Solution: Homogenous Coordinates

Add a third coordinate (*w-coordinate*)

- 2D point = (x, y, 1)^T
- 2D vector = (x, y, 0)^T

Matrix representation of translations

$$\begin{pmatrix} x' \\ y' \\ w' \end{pmatrix} = \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} x + t_x \\ y + t_y \\ 1 \end{pmatrix}$$

slide courtesy of Ren Ng

Affine Transformations

Affine map = linear map + translation

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$

Using homogenous coordinates:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} a & b & t_x \\ c & d & t_y \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

slide courtesy of Ren Ng

2D Transformations

Scale

$$\mathbf{S}(s_x, s_y) = \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Rotation

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

Translation

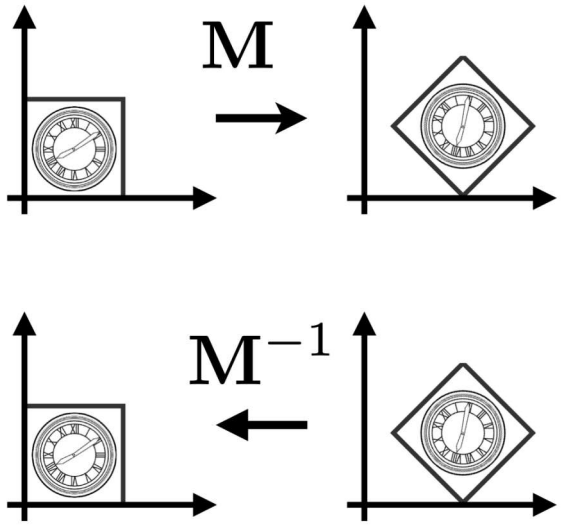
$$\mathbf{T}(t_x, t_y) = \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix}$$

slide courtesy of Ren Ng

Inverse Transform

$$\mathbf{M}^{-1}$$

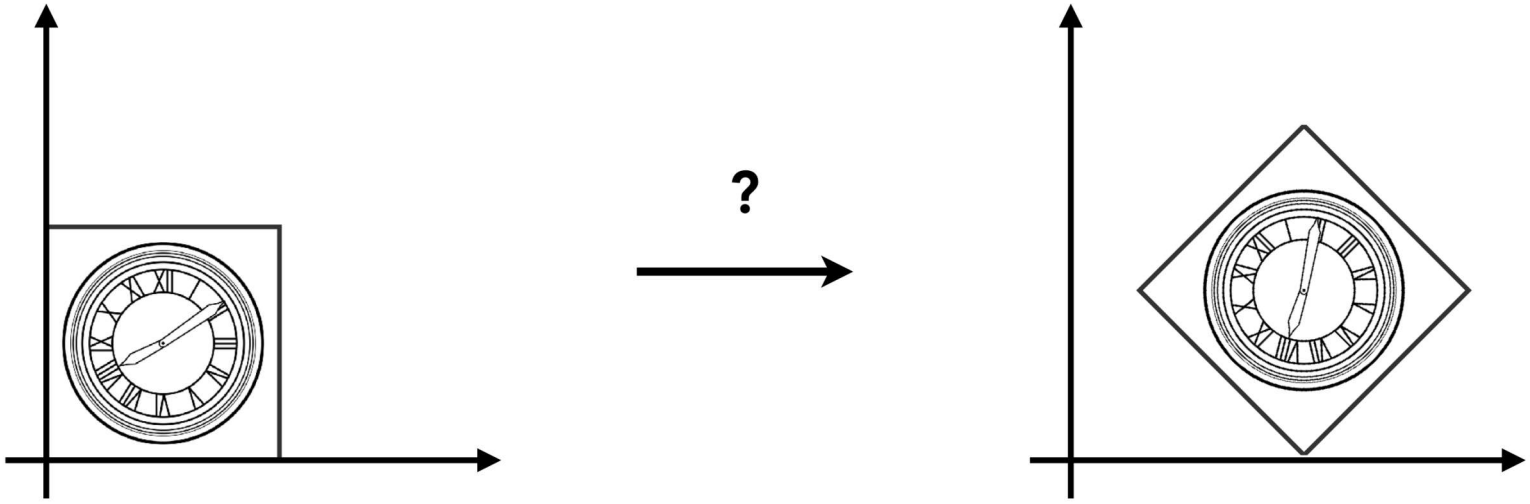
$\mathbf{M}^{-1}$ is the inverse of transform $\mathbf{M}$ in both a matrix and geometric sense



slide courtesy of Ren Ng

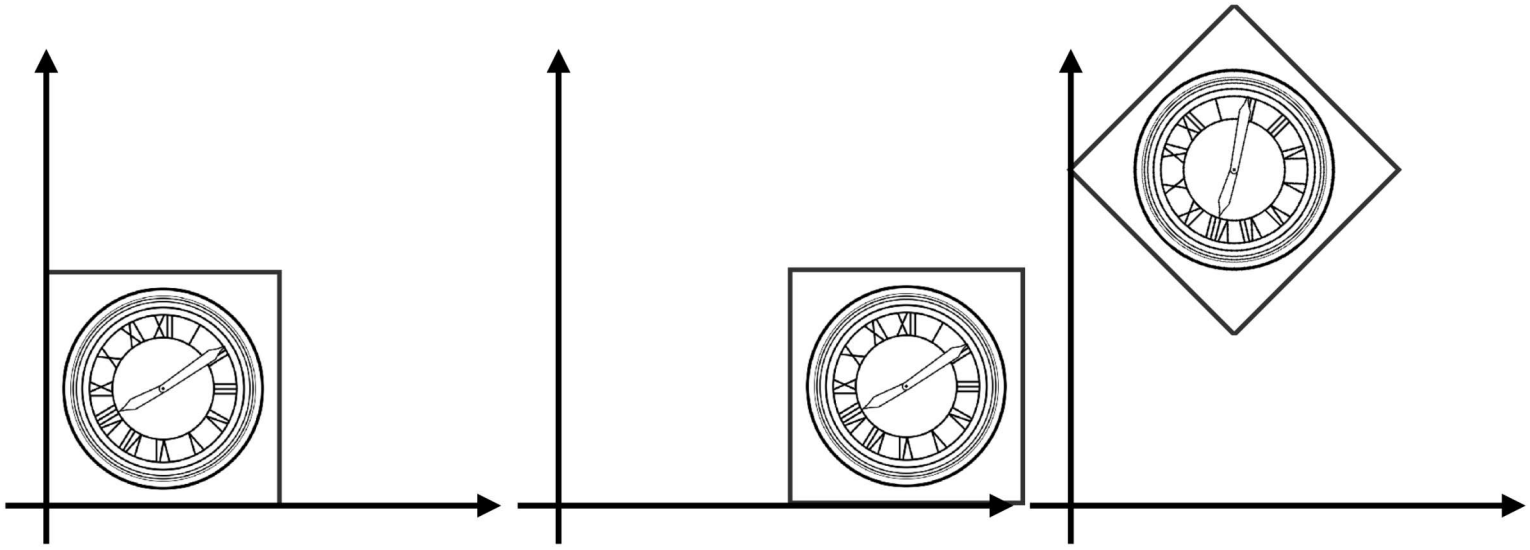
Composing Transforms

Composite Transform



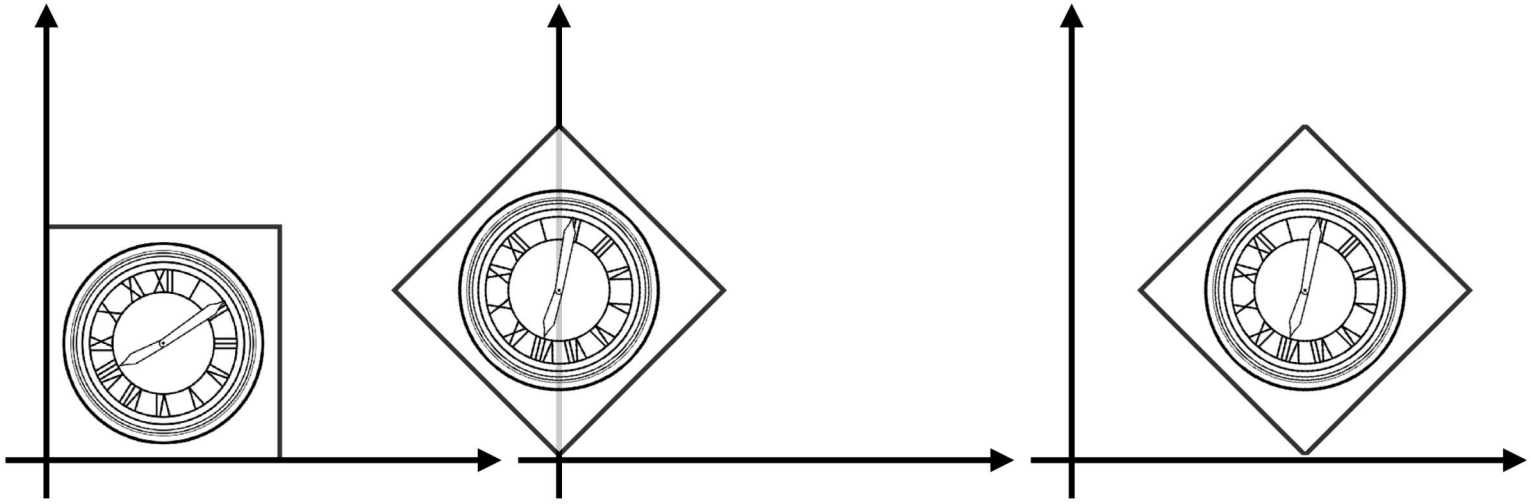
slide courtesy of Ren Ng

Translate Then Rotate?



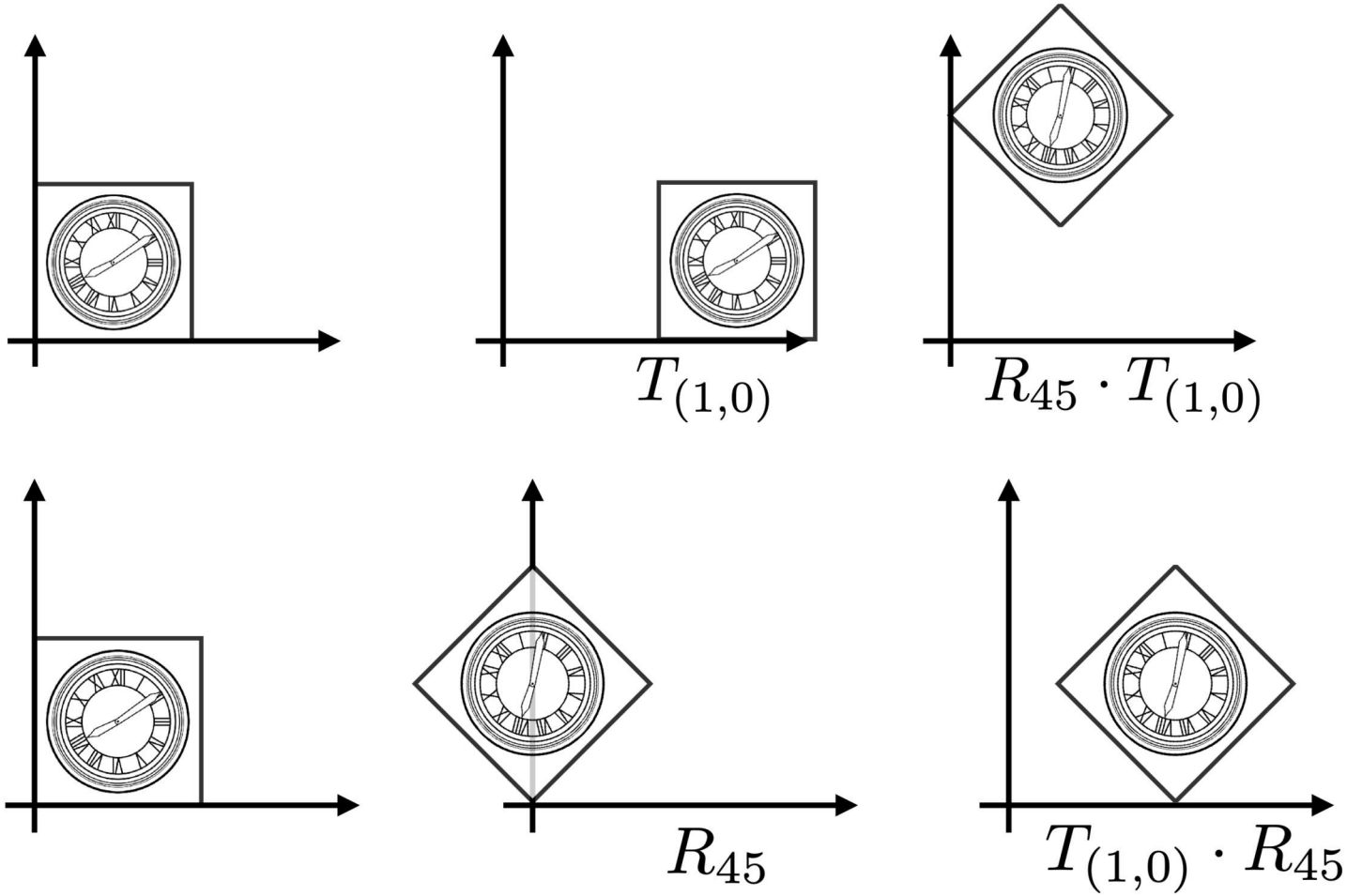
slide courtesy of Ren Ng

Rotate Then Translate



slide courtesy of Ren Ng

Transform Ordering Matters!



slide courtesy of Ren Ng

Transform Ordering Matters!

Matrix multiplication is <u>not</u> commutative

$$R_{45} \cdot T_{(1,0)} \neq T_{(1,0)} \cdot R_{45}$$

Recall the matrix math represented by these symbols:

$$\begin{bmatrix} \cos 45° & -\sin 45° & 0 \\ \sin 45° & \cos 45° & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \neq \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos 45° & -\sin 45° & 0 \\ \sin 45° & \cos 45° & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Note that matrices are applied right to left:

$$T_{(1,0)} \cdot R_{45} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos 45° & -\sin 45° & 0 \\ \sin 45° & \cos 45° & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

slide courtesy of Ren Ng

Composing Transforms

Sequence of affine transforms A_1, A_2, A_3, ...

- Compose by matrix multiplication
  - Very important for performance!

$$A_n(\ldots A_2(A_1(\mathbf{x}))) = \underbrace{\mathbf{A}_n \cdots \mathbf{A}_2 \cdot \mathbf{A}_1}_{} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

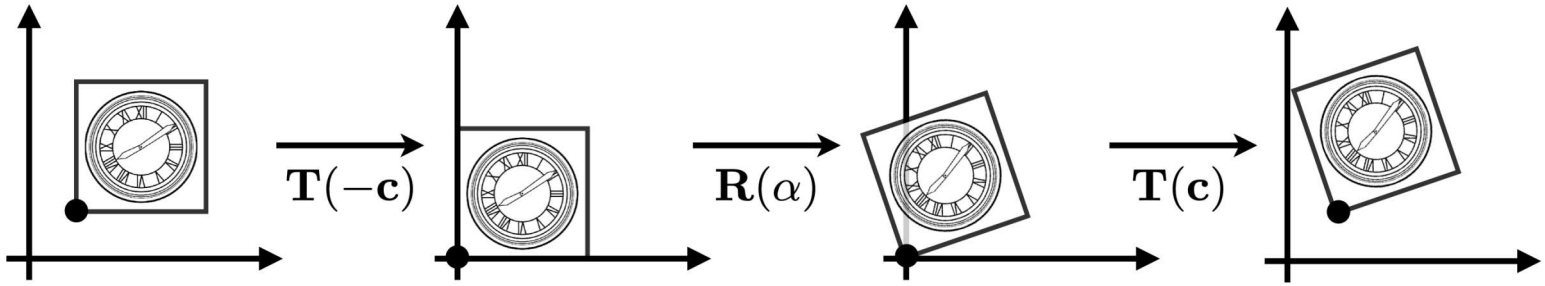
Pre-multiply *n* matrices to obtain a single matrix representing combined transform

slide courtesy of Ren Ng

Decomposing Complex Transforms

How to rotate around a given point c?

1. Translate center to origin
2. Rotate
3. Translate back



Matrix representation?

$$\mathbf{T}(\mathbf{c}) \cdot \mathbf{R}(\alpha) \cdot \mathbf{T}(-\mathbf{c})$$

slide courtesy of Ren Ng