CSC418/2504 Computer Graphics

Some Slides/Images adapted from Marschner and Shirley

Announcements

- Assignment 5 now due March 1st
- Assignment 6 due also pushed back (March 15th)
- Office hours are now M4-5 and W5-6 in BA5287 until Dave has his baby, then they will be in BA5268 (his office)
- Assignment 4 remark request
  - TA Office Hours will be 12:30 to 1:30 on Friday
- **TA Email Address: *csc418tas@cs.toronto.edu***

Midterm 1

Midterm 1 is right after reading week (Feb 25th in tutorial)

Topics:

- Lectures 1-6
- Assignments 1-5

Format:

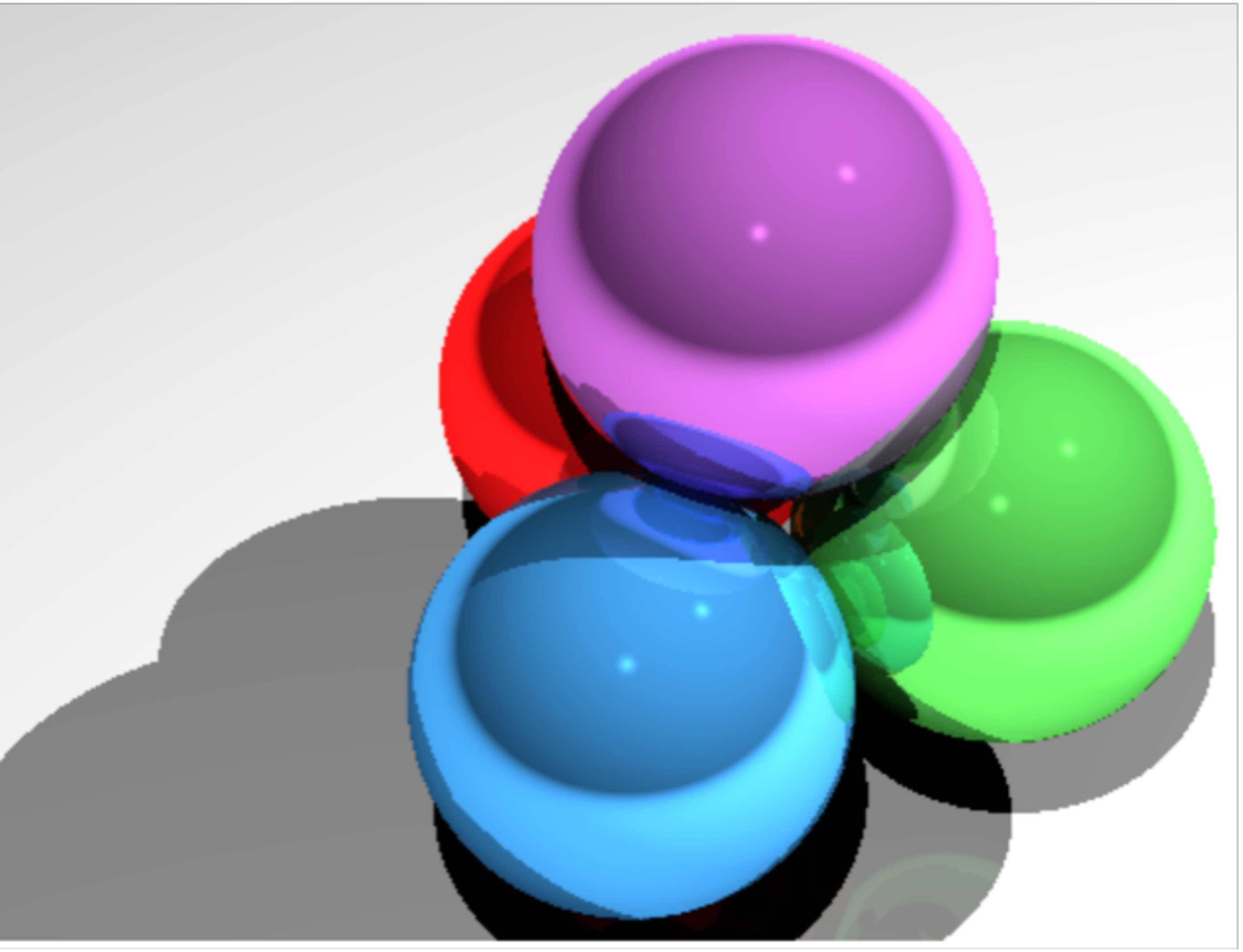
- True/False, multiple choice, short answer, long answer

Any Questions ?

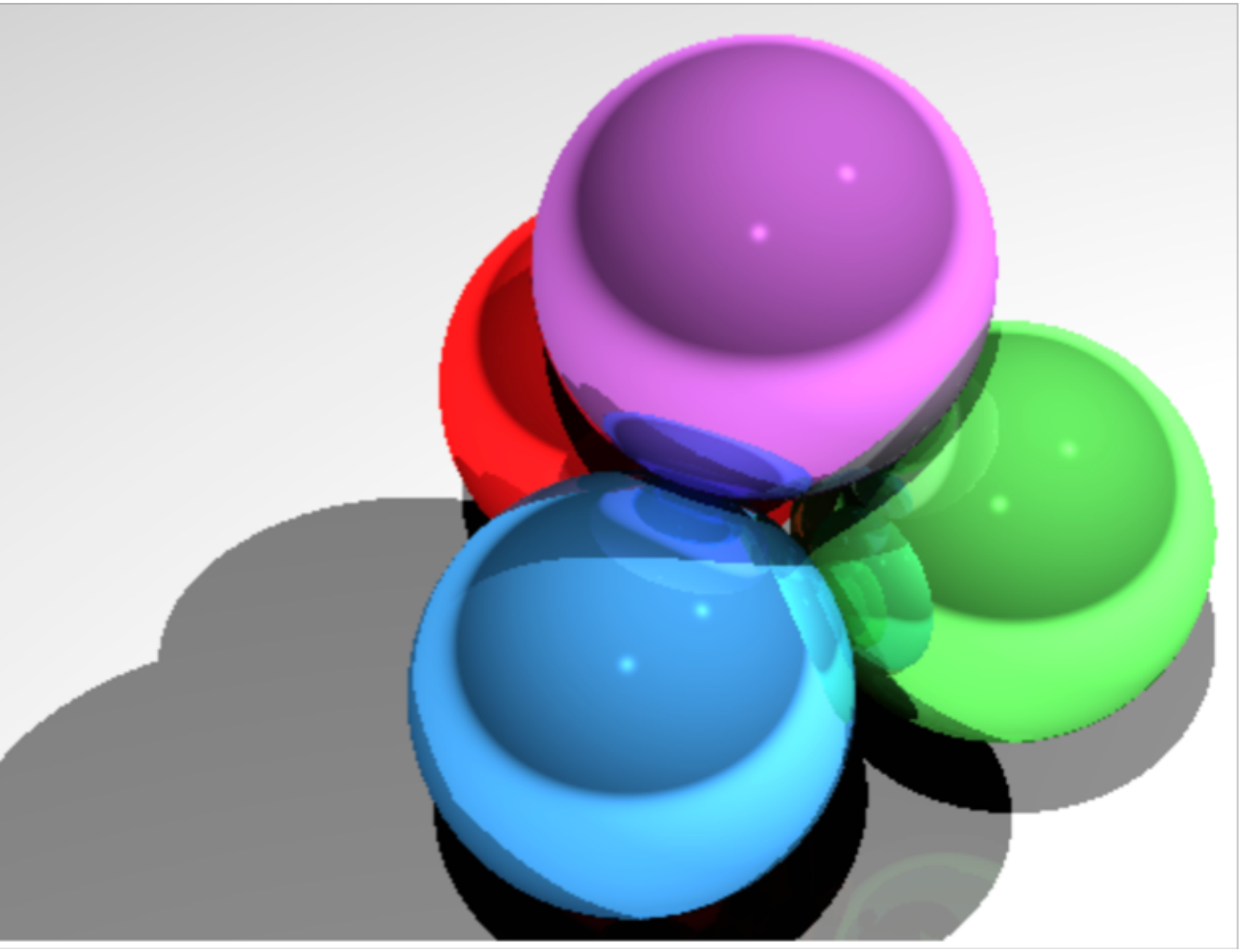
This week: Transformation Matrices

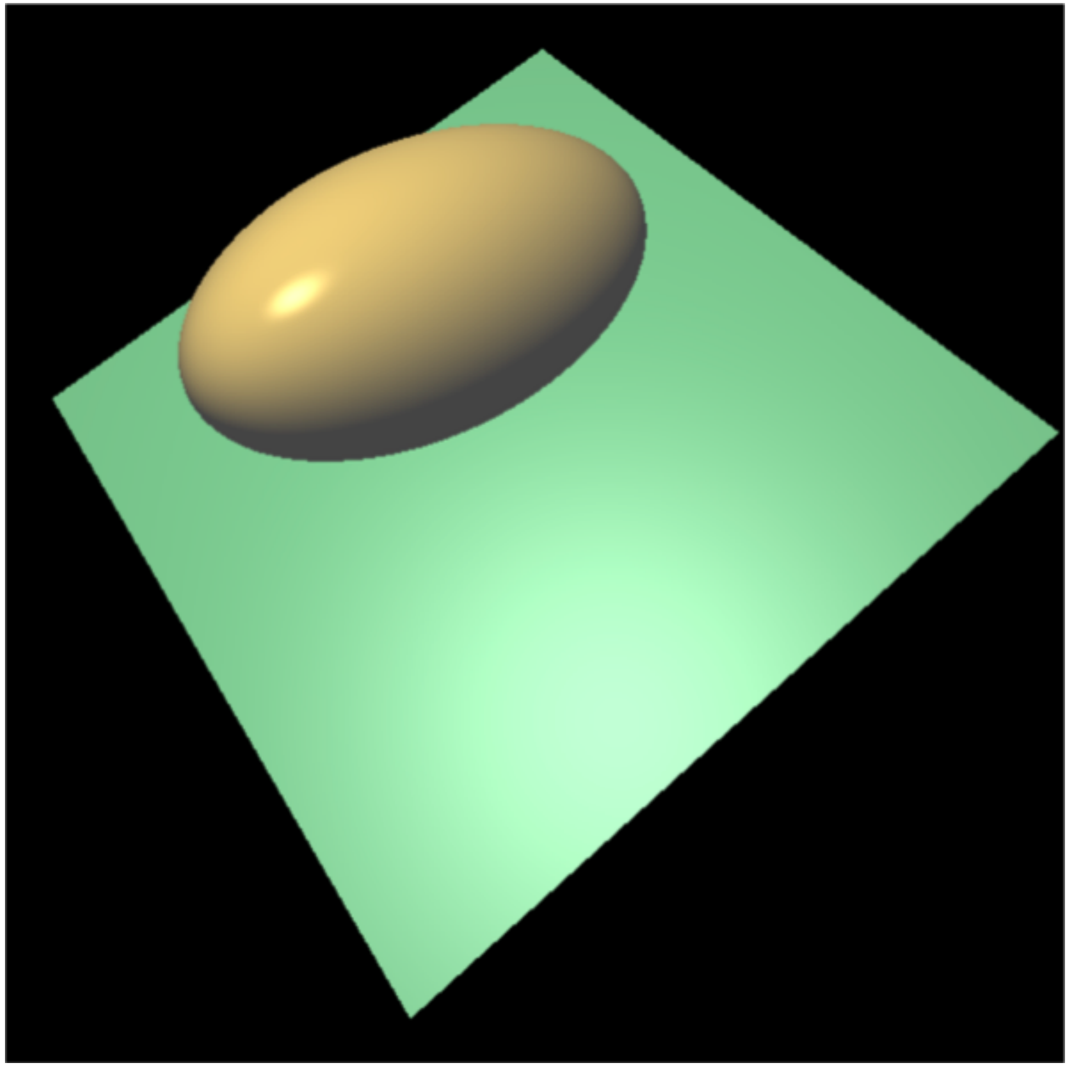
- Motivation
- What is a linear transformation?
- 2D transformations
  - Scale, Shear, Rotate, Reflect, Inverses
- 3D transformations
  - Scale, Shear, Rotate
- Homogeneous Coordinates
  - Affine transformations, translations
- Transforming Normals

Motivation

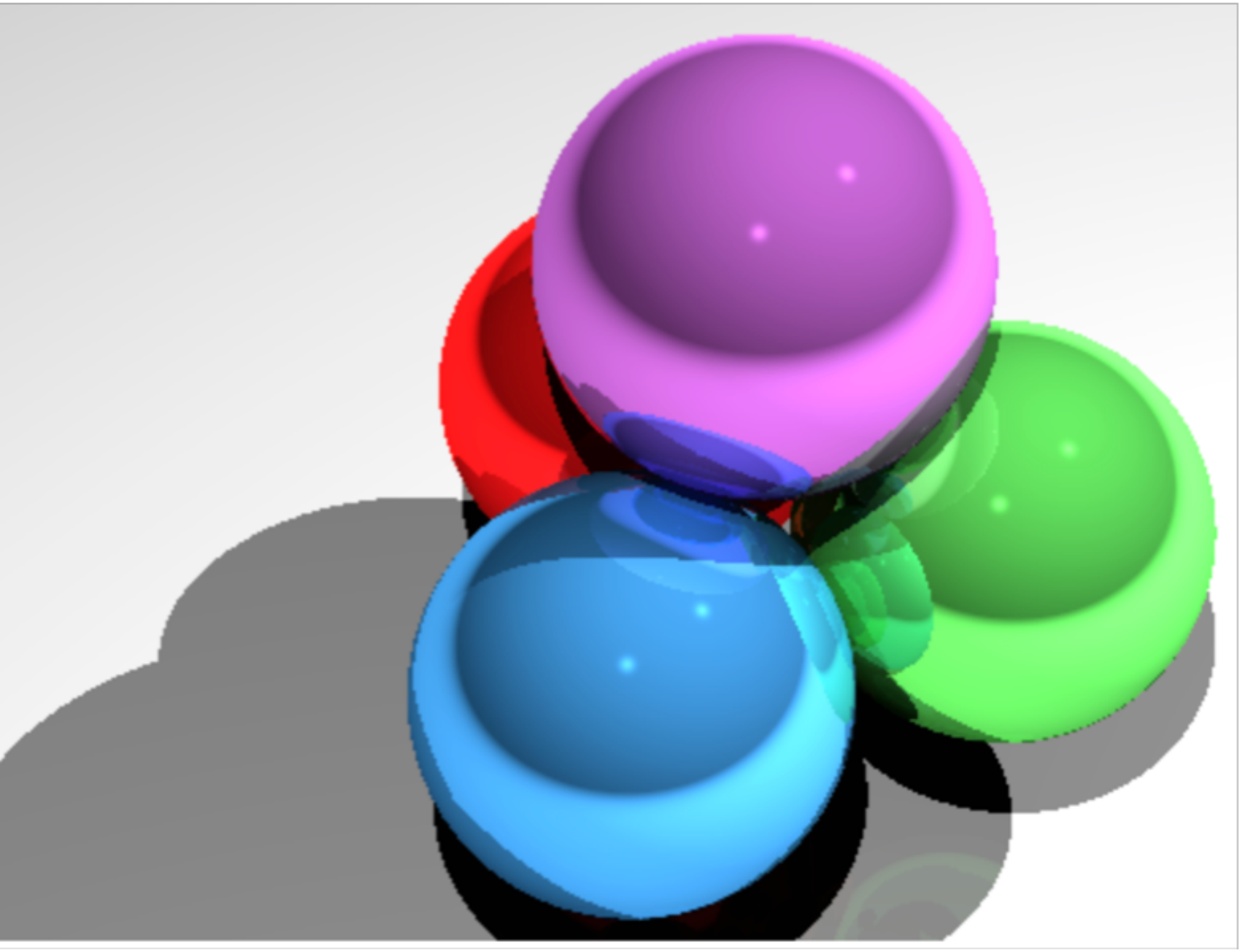


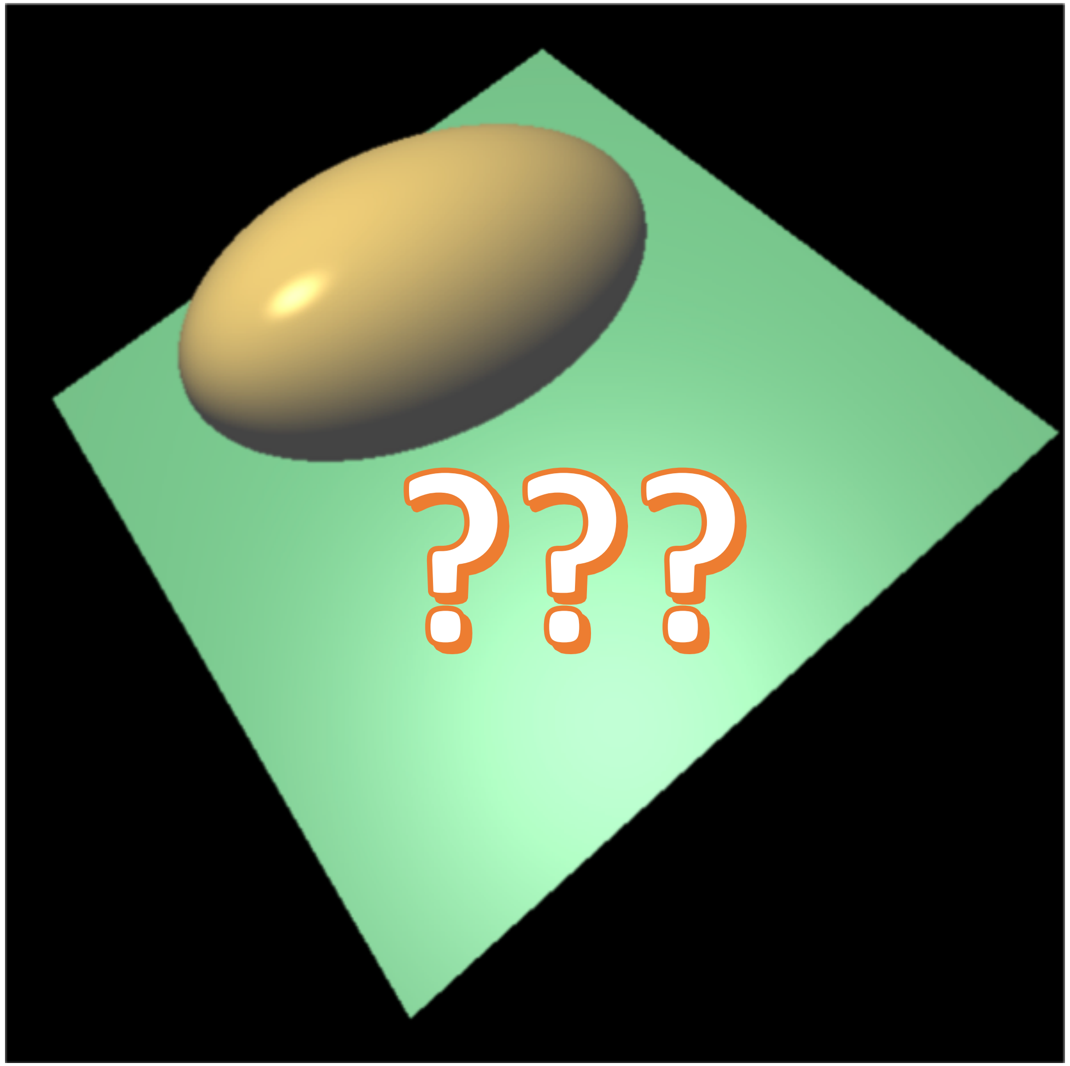
Motivation



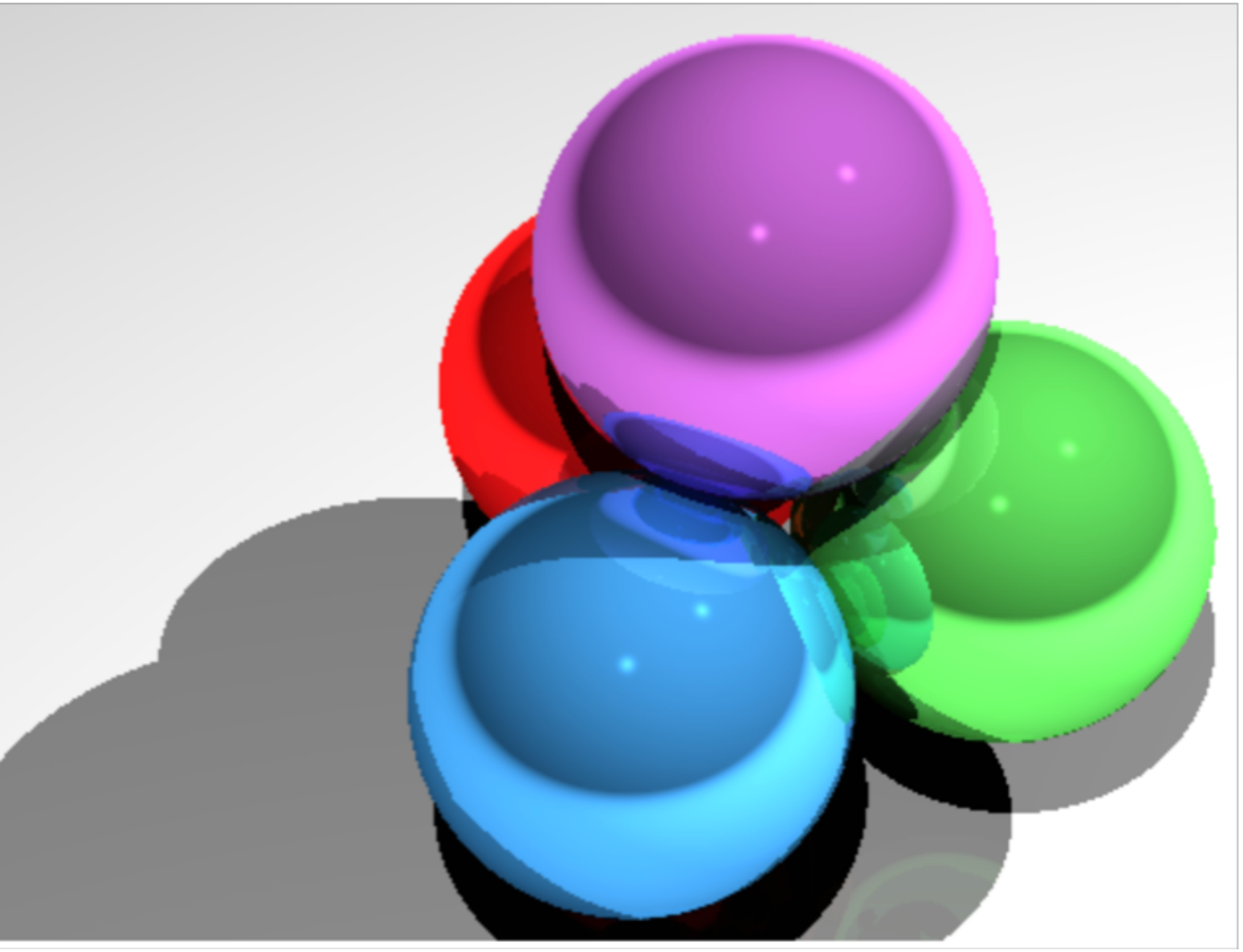


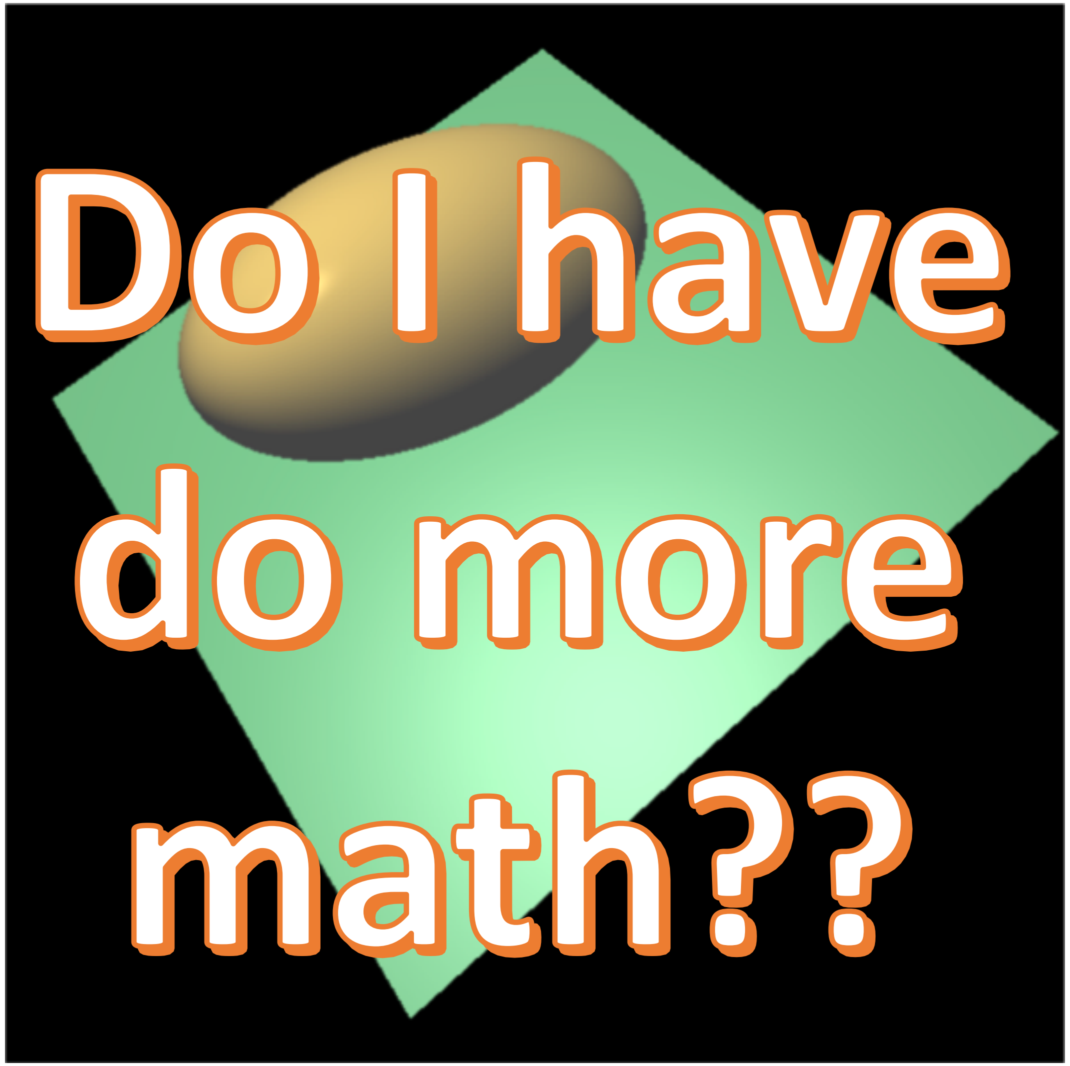
Motivation





Motivation





Motivation

An ellipse is just a “transformed” sphere

Math will help us do less math!!

Motivation

You’ve done ray-tracing up to now, which is an image-order algorithm

Image-Order

for-each ***pixel***

set the ***pixel*** based on the ***objects*** that influence it:

An ode to raytracing

- Raytracing is:

An ode to raytracing

- Raytracing is:
  - Beautiful

An ode to raytracing

- Raytracing is:
  - Beautiful
  - Slow

Motivation

For speed-performance, object-order algorithms are used

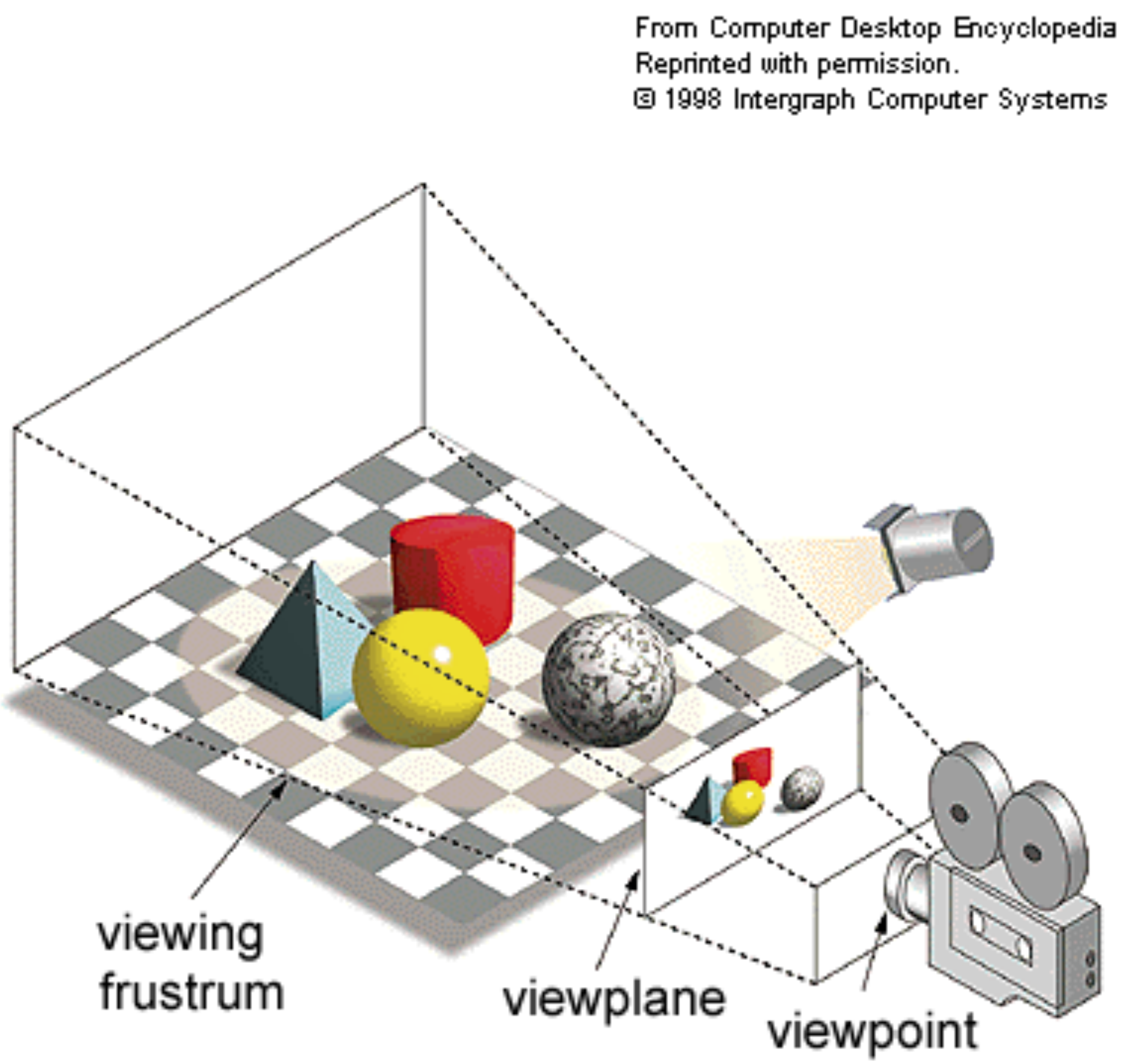
Object-Order

for-each ***object***

update the ***pixels*** the ***object*** influences;

Motivation

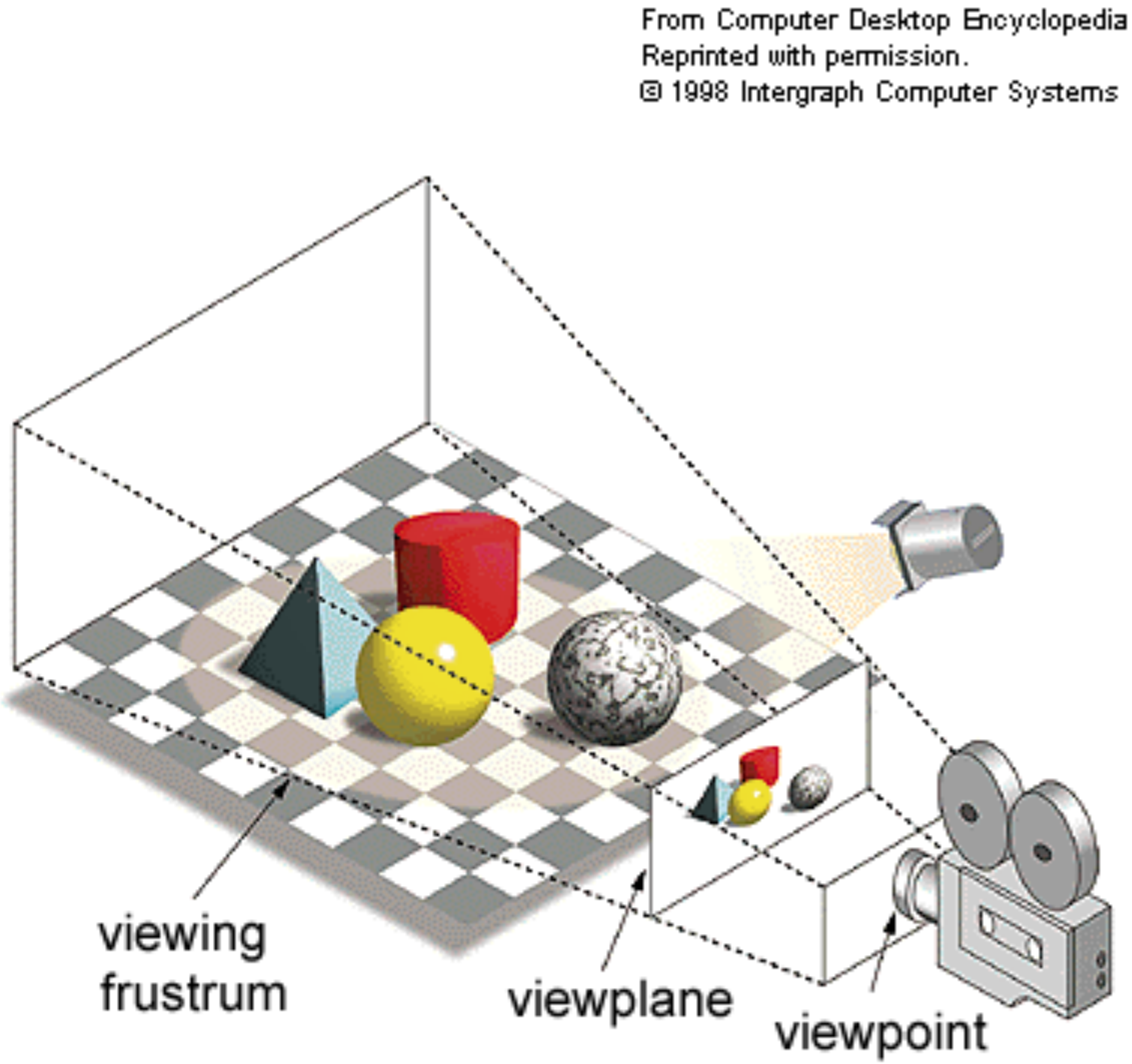
Why are they faster?



Motivation

Why are they faster?

We only need to project to a 2D plane and rasterize each object

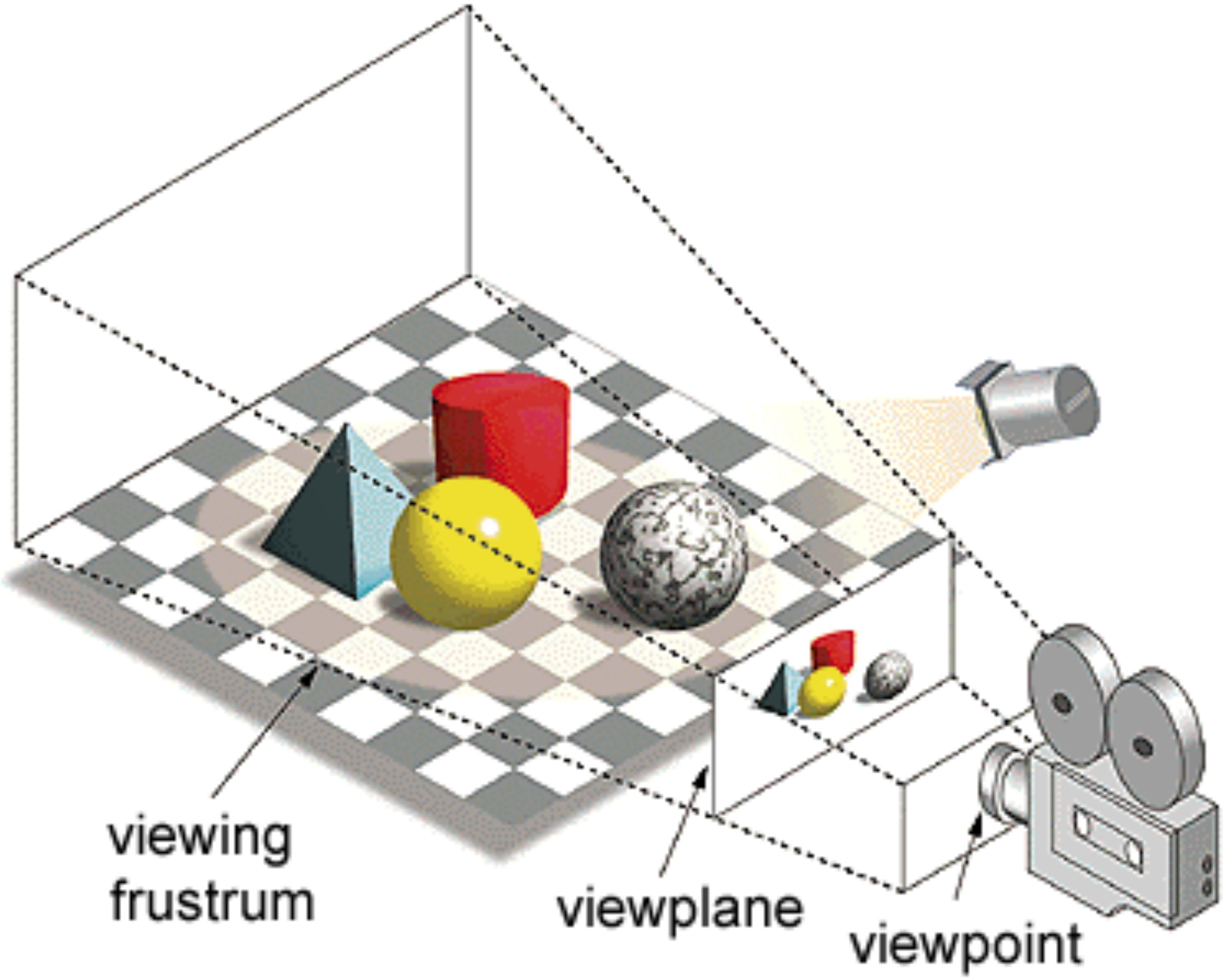


Motivation

Why are they faster?

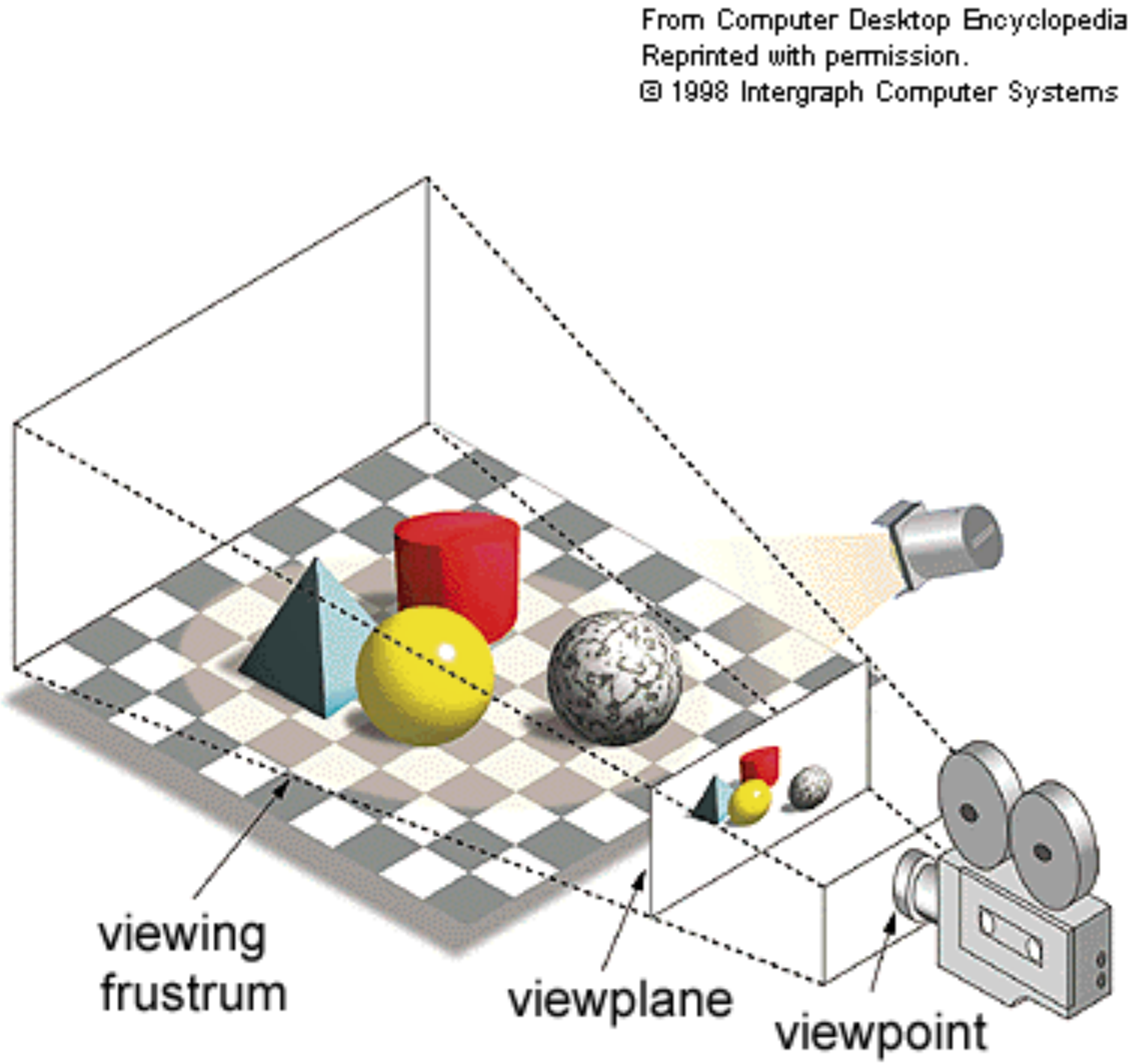
We only need to project to a 2D plane and rasterize each object

Each object tells the pixel what colour to be



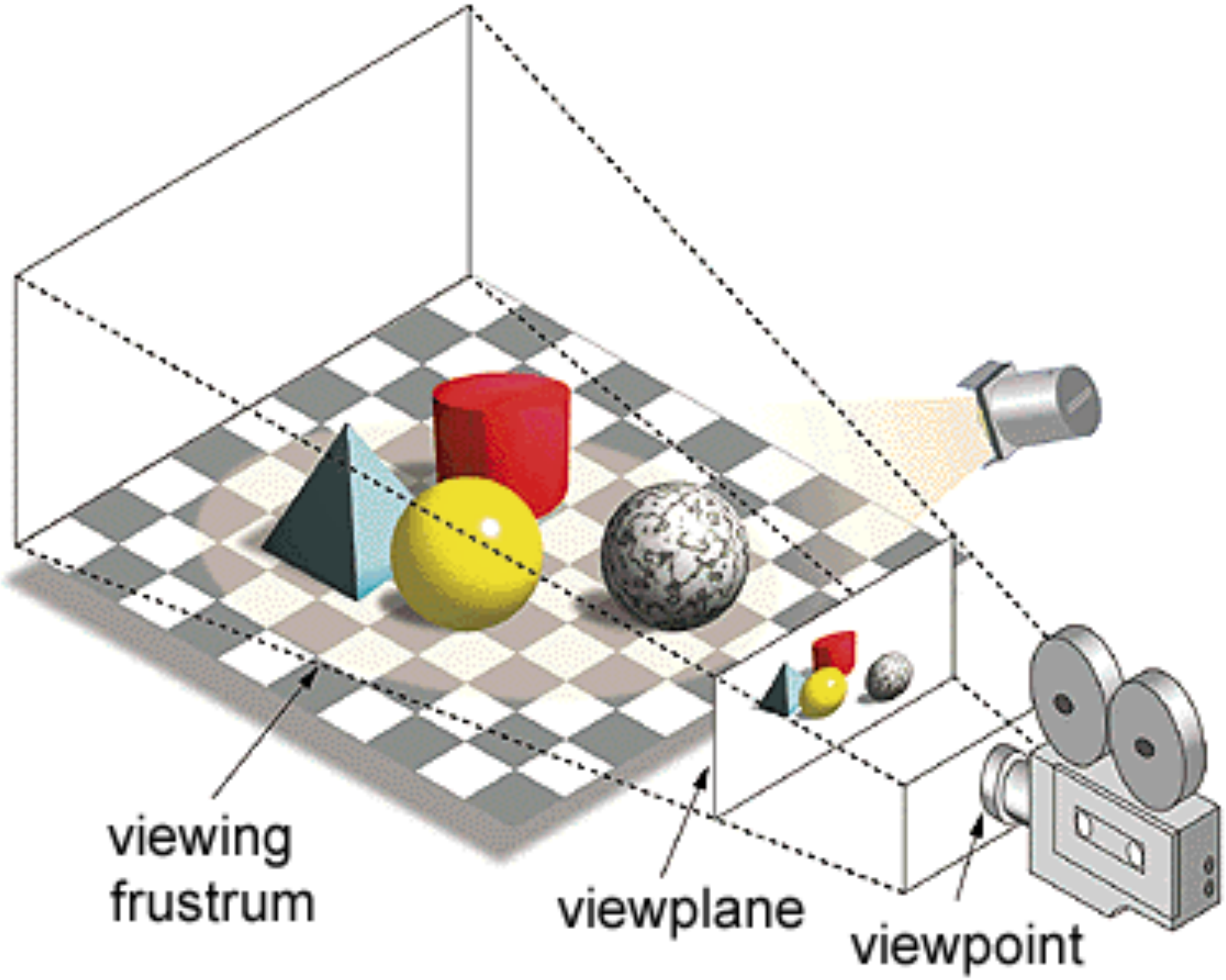
Questions:

- How do we?
  - Get each object into the world efficiently



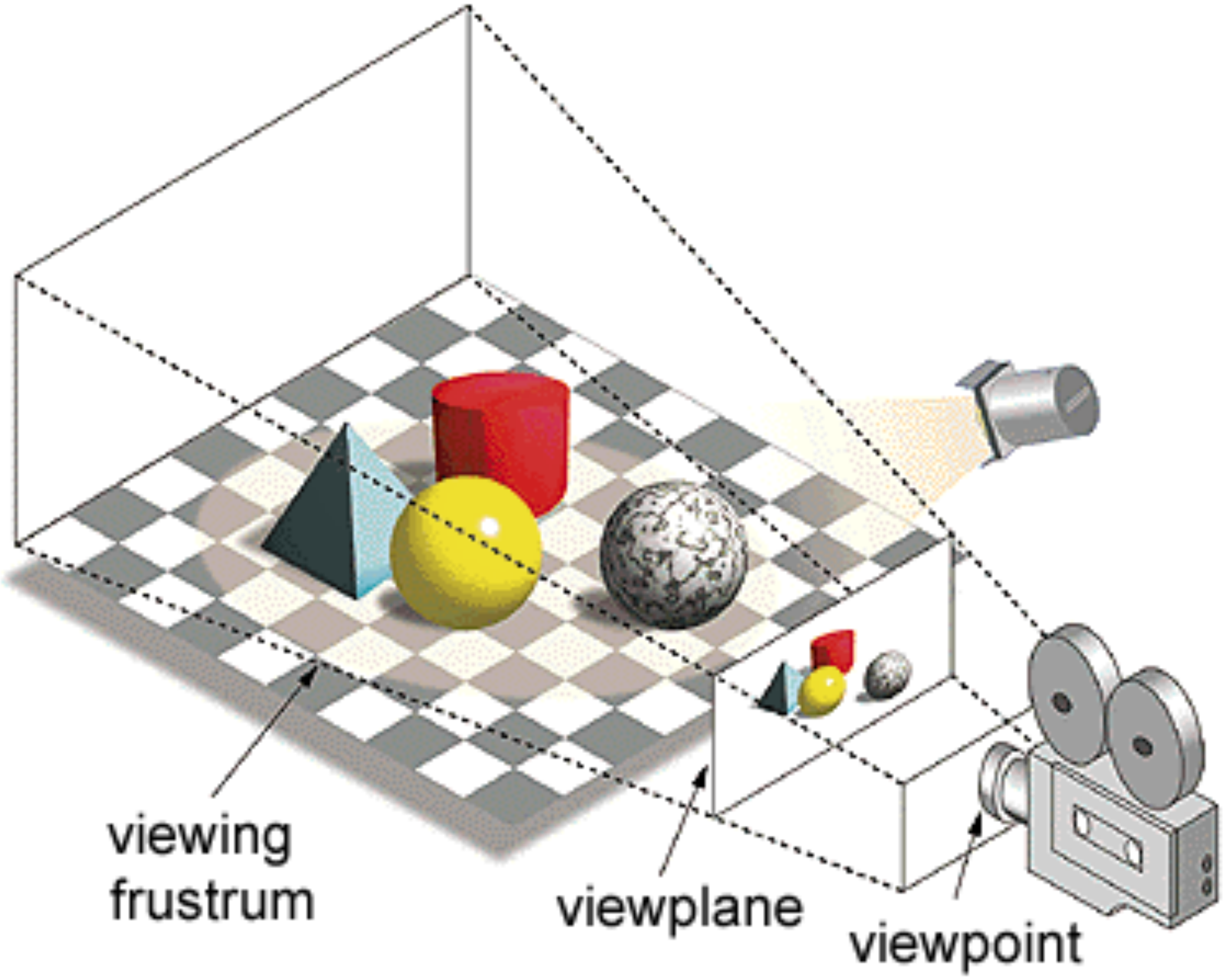
Questions:

- How do we?
  - Get each object into the world efficiently
  - Handle an arbitrary camera (not centered at the origin)



Questions:

- How do we?
  - Get each object into the world efficiently
  - Handle an arbitrary camera (not centered at the origin)
  - Project the scene to the 2D viewplane



The answer to all these questions is:

The answer to all these questions is:

Linear transformations

What is a linear transformation?

A: For vectors, a linear transformation is any operation performed by a matrix

What is a linear transformation?

A: For vectors, a linear transformation is any operation performed by a matrix

$$Ax = b$$

What is a linear transformation?

A: For vectors, a linear transformation is any operation performed by a matrix

$$\underbrace{A}_{n \times n}x = b$$

What is a linear transformation?

A: For vectors, a linear transformation is any operation performed by a matrix

$$A\underbrace{x}_{n \times 1} = b$$

What is a linear transformation?

A: For vectors, a linear transformation is any operation performed by a matrix

$$Ax = \underbrace{b}_{n \times 1}$$

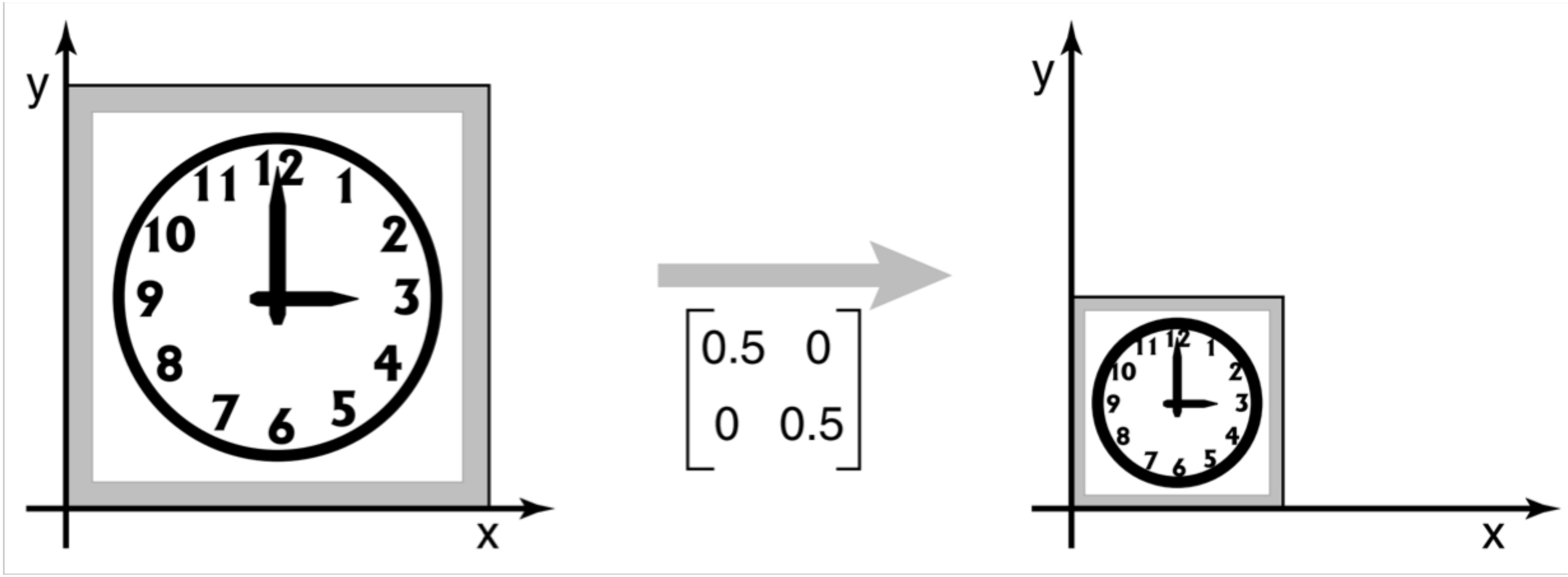
Linear transformations in 2D

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

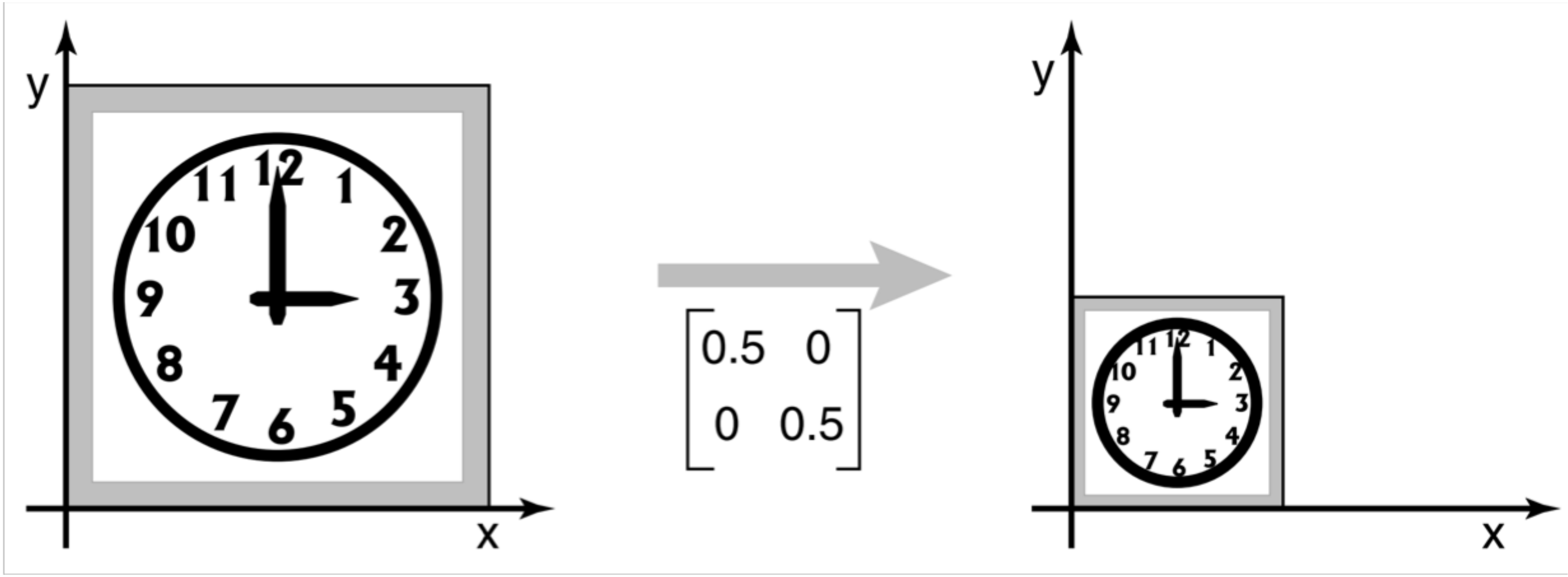
Linear transformations in 2D: Scale

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

Linear transformations in 2D: Scale

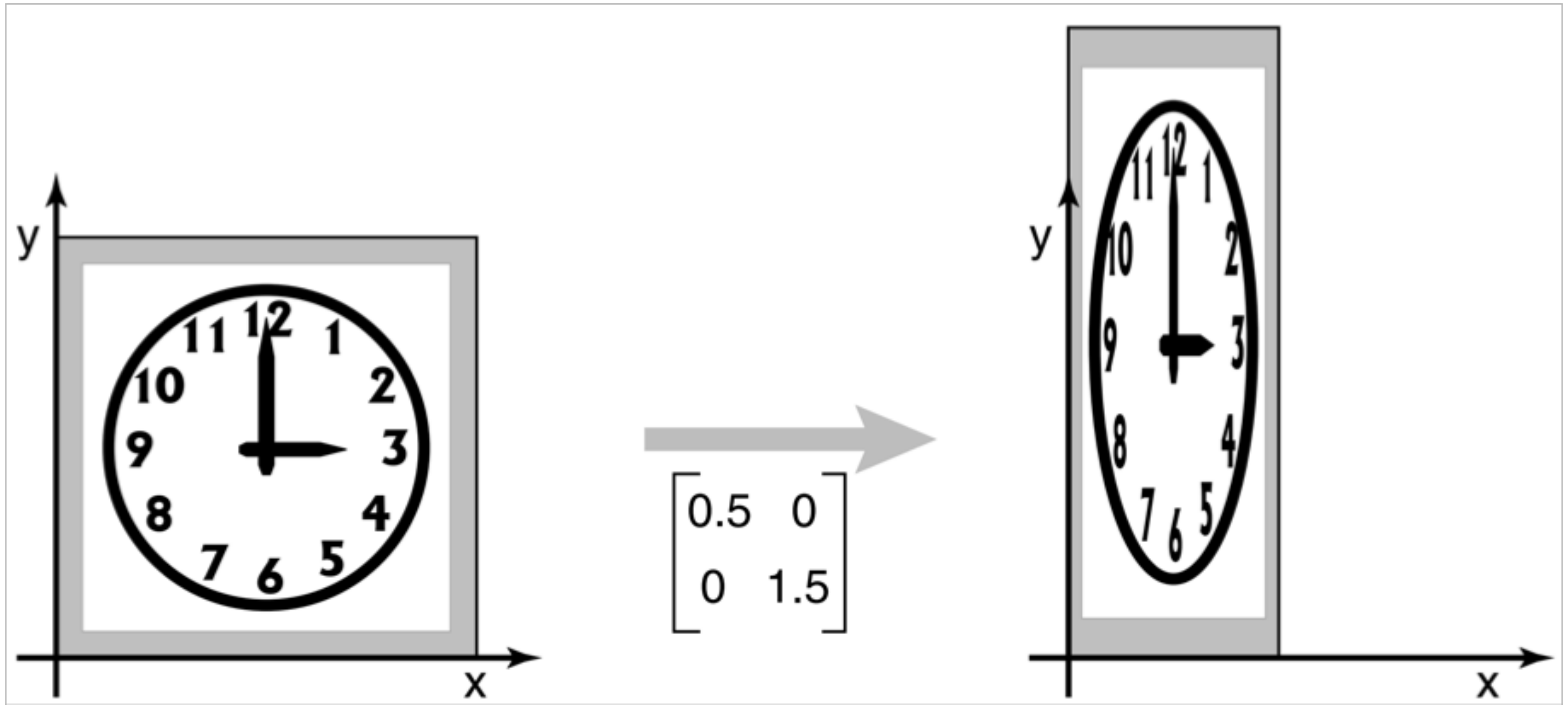


Linear transformations in 2D: Scale



When $s_x = s_y$ the scaling is said to be "Uniform"

Linear transformations in 2D: Scale



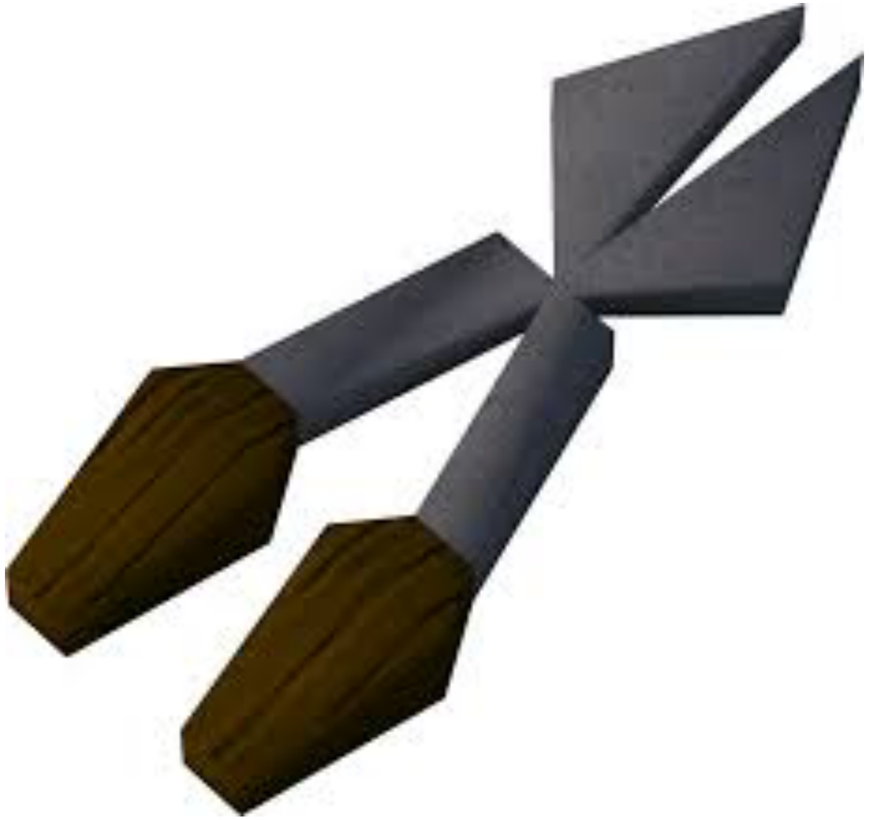
$$\begin{bmatrix} 0.5 & 0 \\ 0 & 1.5 \end{bmatrix}$$

This is an example of non-uniform scaling in x and y

Linear transformations in 2D: Shear

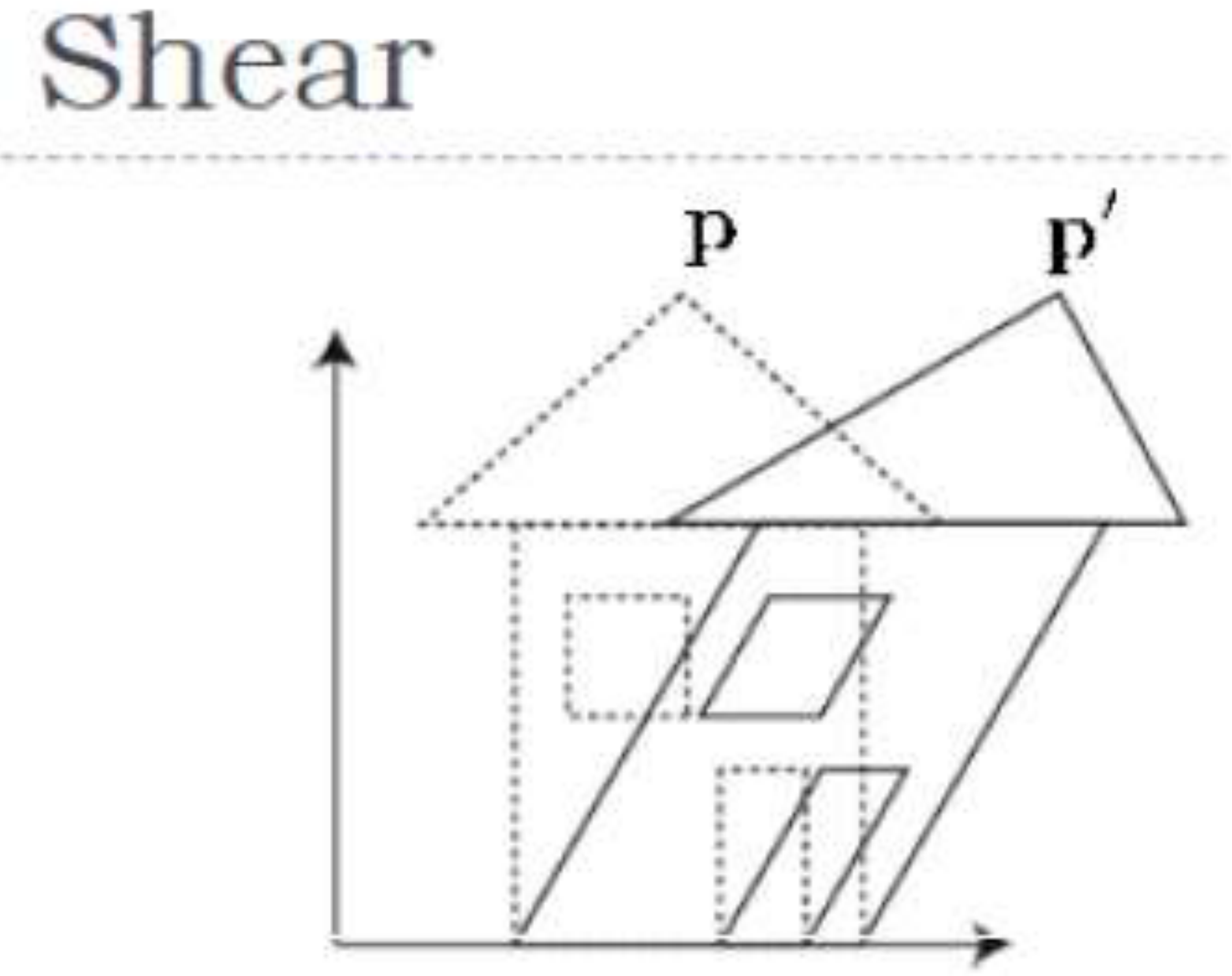
What’s a shear?

What’s a shear?



What’s a shear?

What’s a shear?



https://www.tutorialspoint.com/computer_graphics/3d_transformation.htm

Linear transformations in 2D: Shear

Horizontal Shear

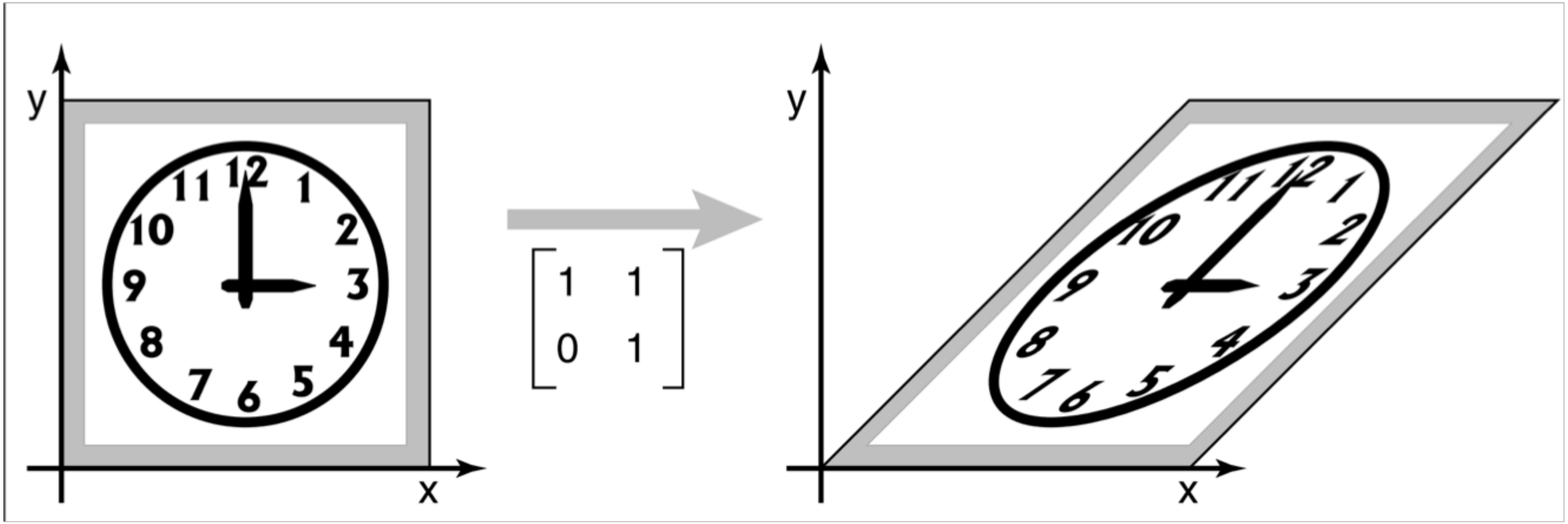
$$\begin{bmatrix} 1 & s \\ 0 & 1 \end{bmatrix}$$

Linear transformations in 2D: Shear

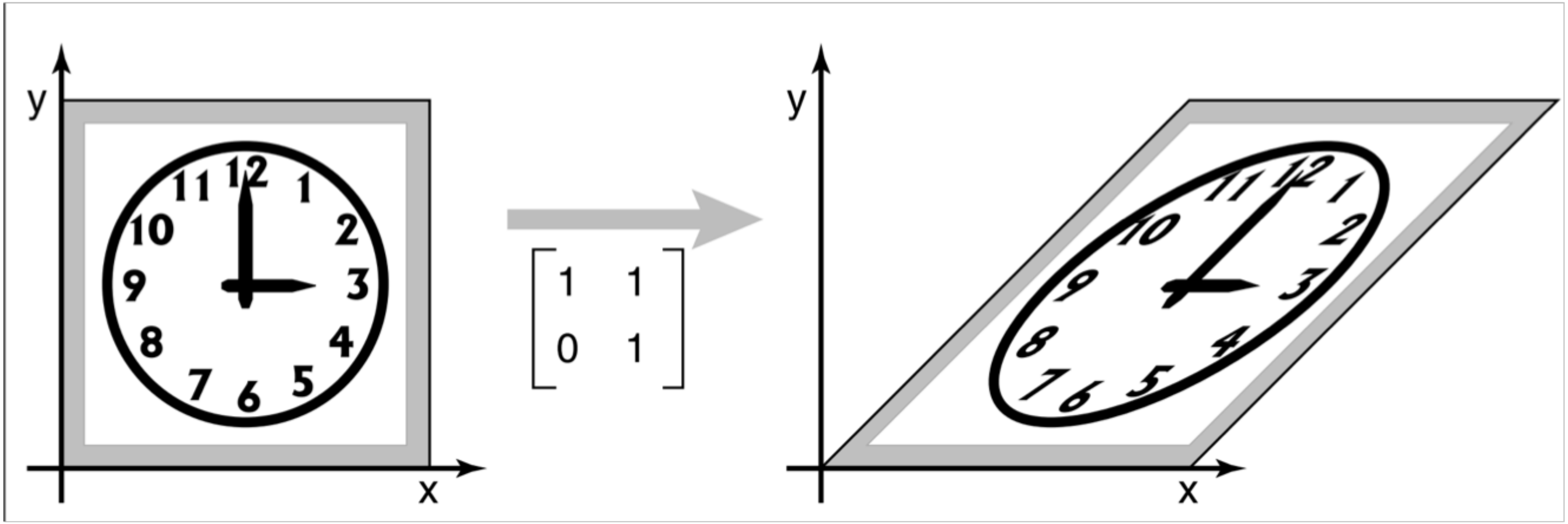
Why horizontal?

$$\begin{bmatrix} 1 & s \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x + sy \\ y \end{bmatrix}$$

Linear transformations in 2D: Shear



Linear transformations in 2D: Shear

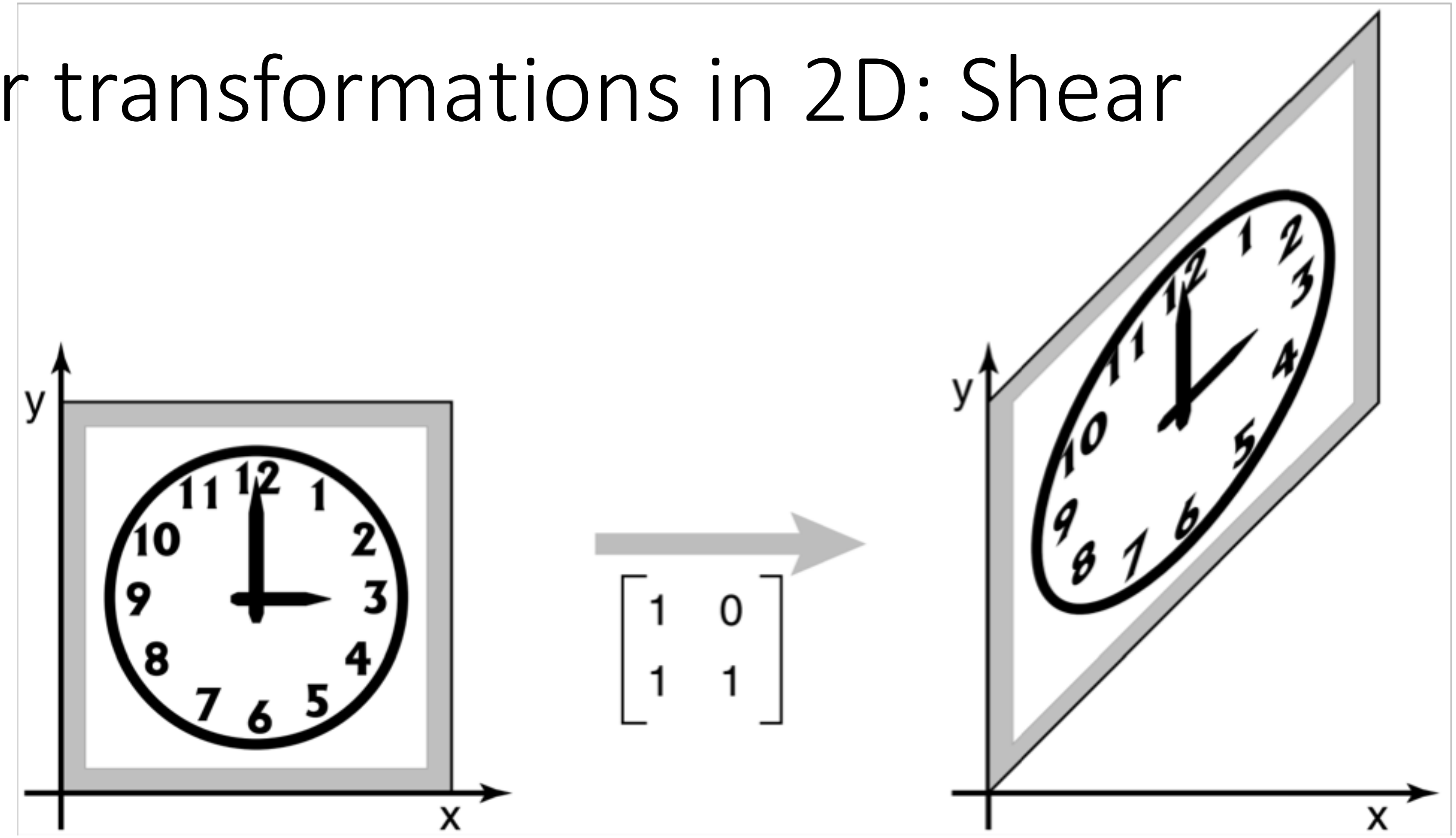


$$\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x + y \\ y \end{bmatrix}$$

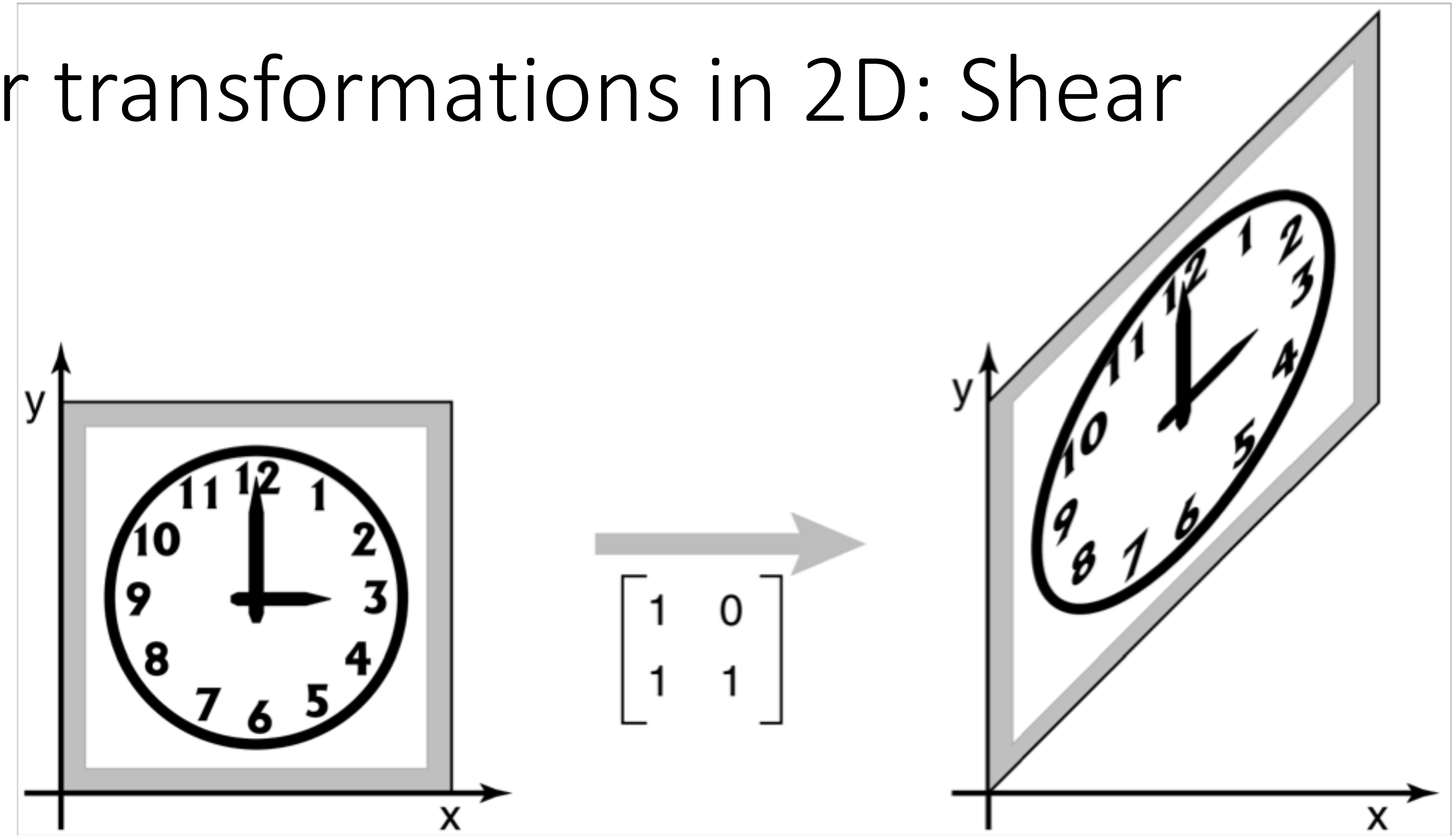
Linear transformations in 2D: Shear

$$\begin{bmatrix} 1 & 0 \\ s & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x \\ y + sx \end{bmatrix}$$

Linear transformations in 2D: Shear

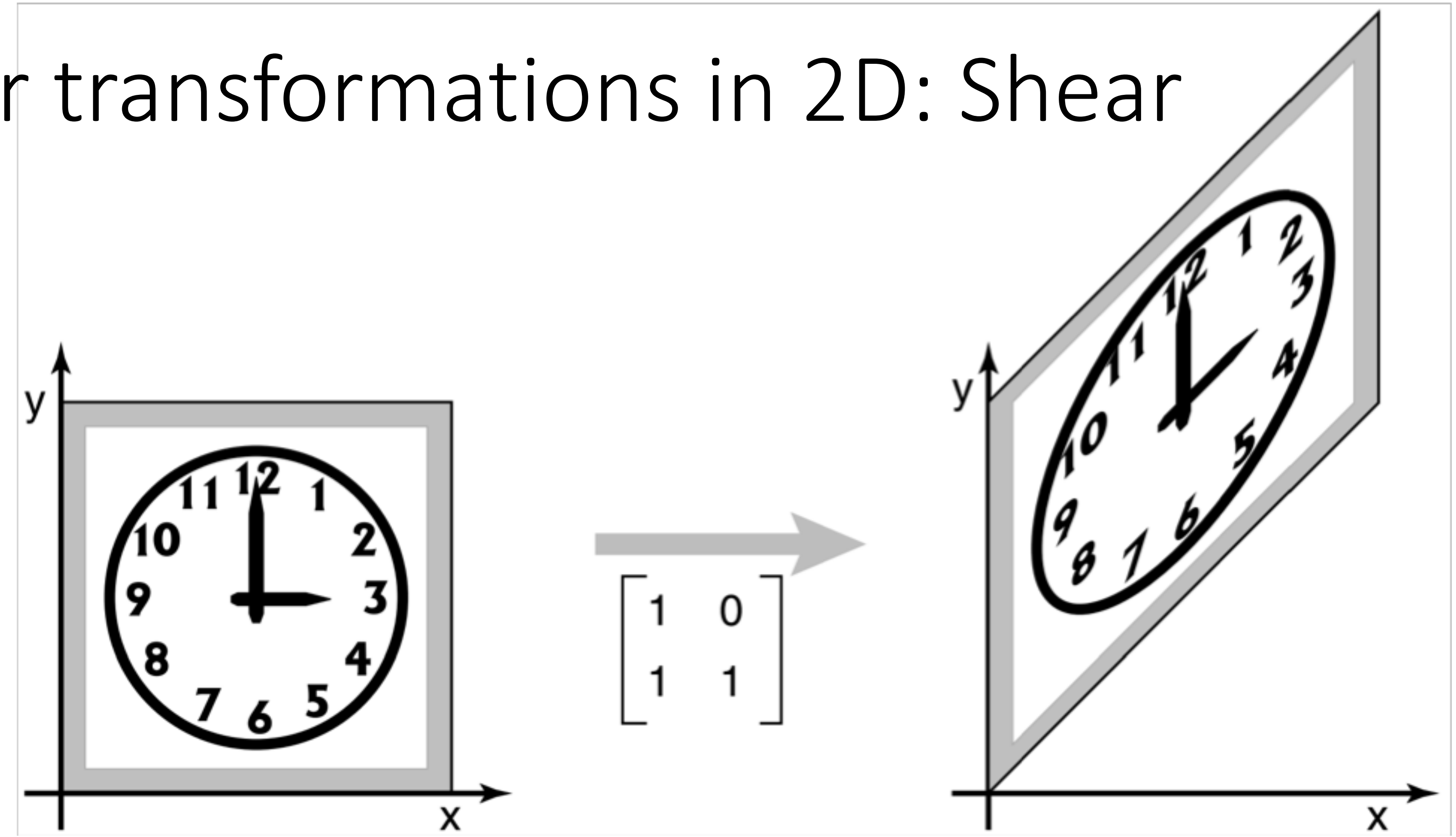


Linear transformations in 2D: Shear



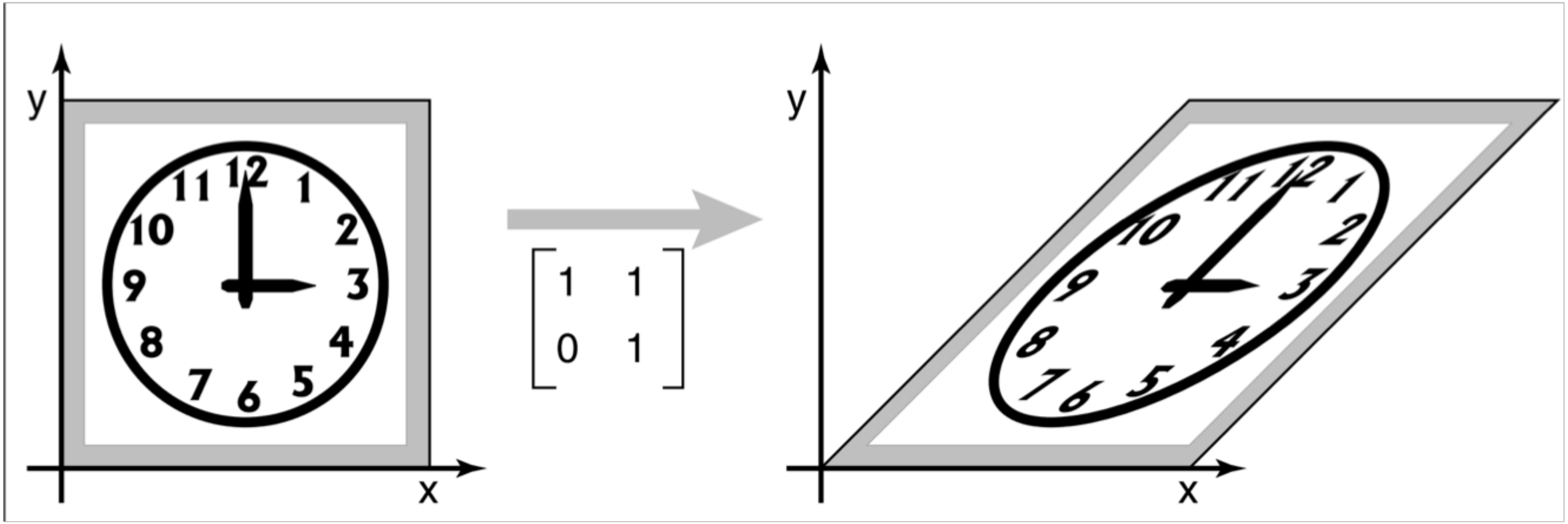
$$\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x \\ y + x \end{bmatrix}$$

Linear transformations in 2D: Shear

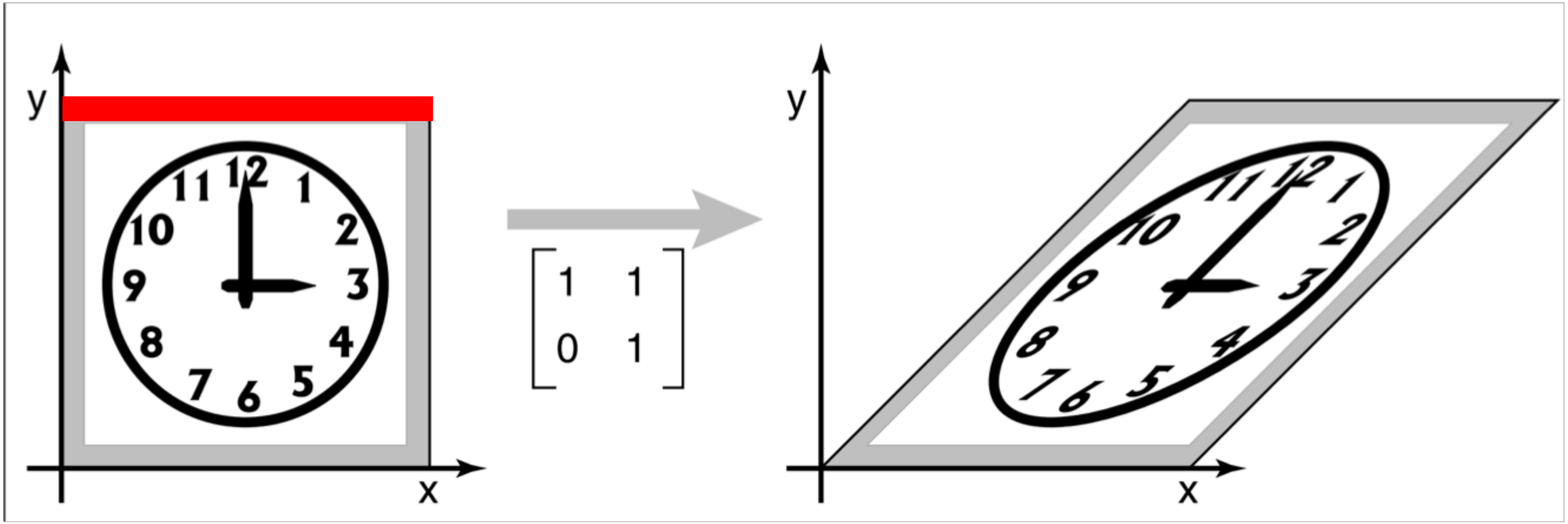


$$\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x \\ y + x \end{bmatrix}$$

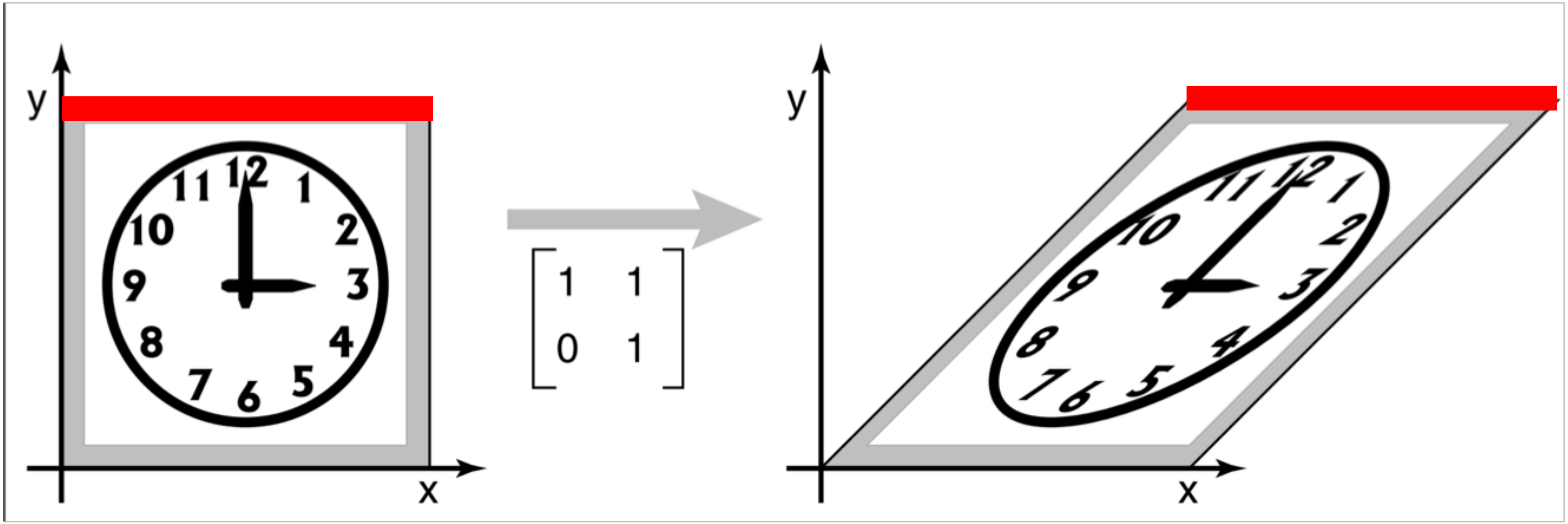
Special note:



Special note:



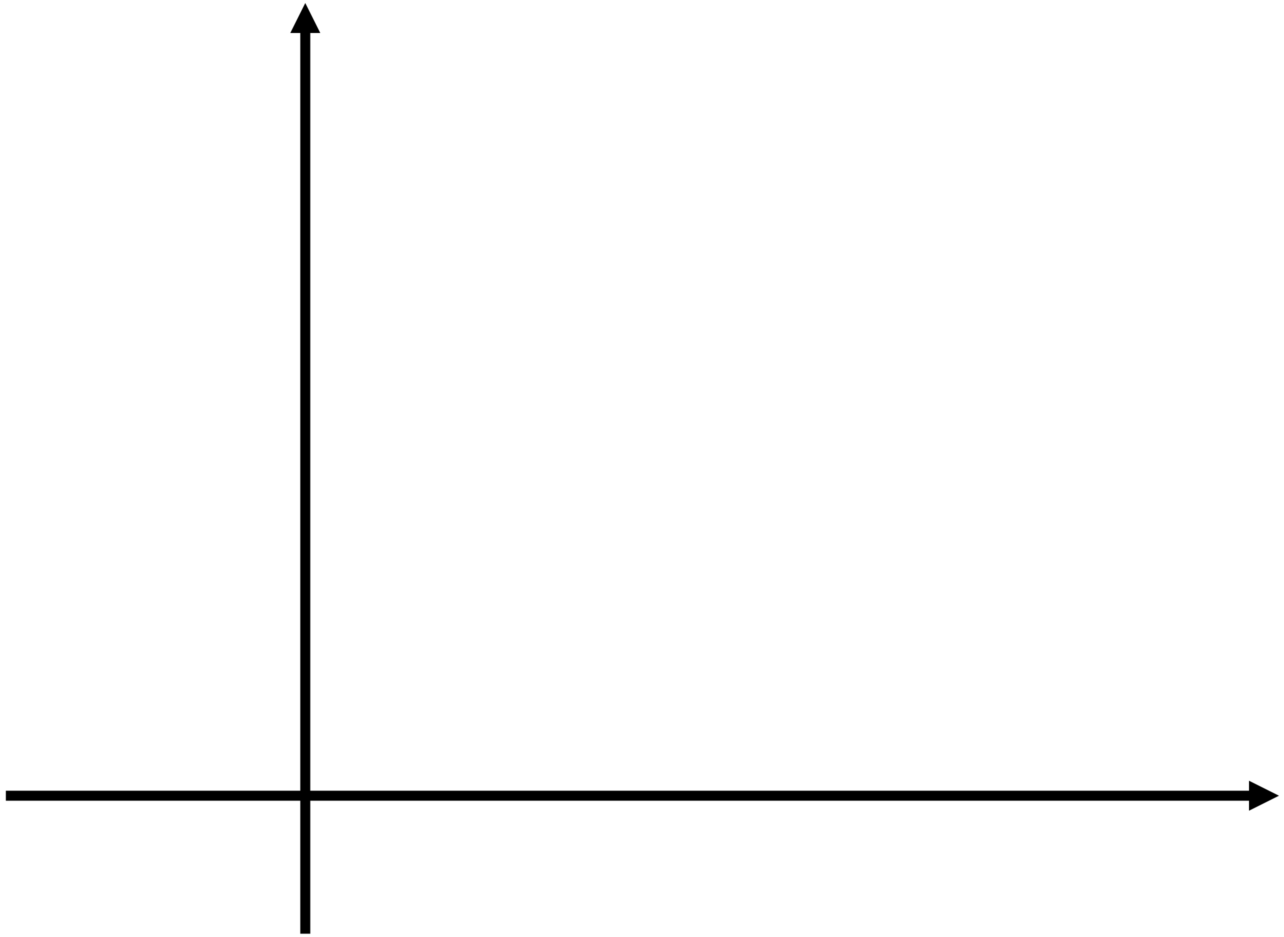
Special note:



These are always the same length

Linear transformations in 2D: Rotate

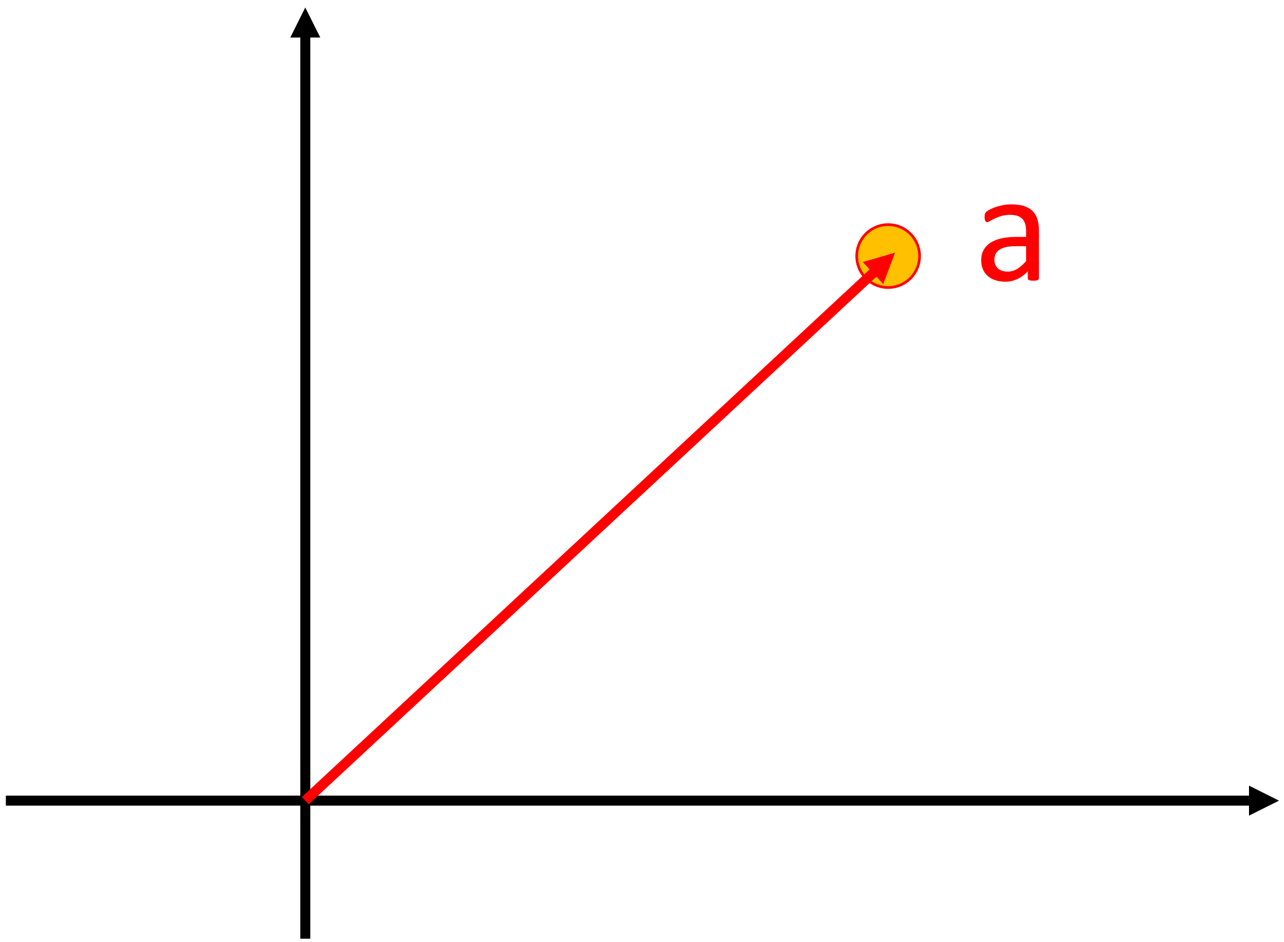
$$x_a = r\cos\alpha$$
$$y_a = r\sin\alpha.$$



Linear transformations in 2D: Rotate

$$x_a = r\cos\alpha$$

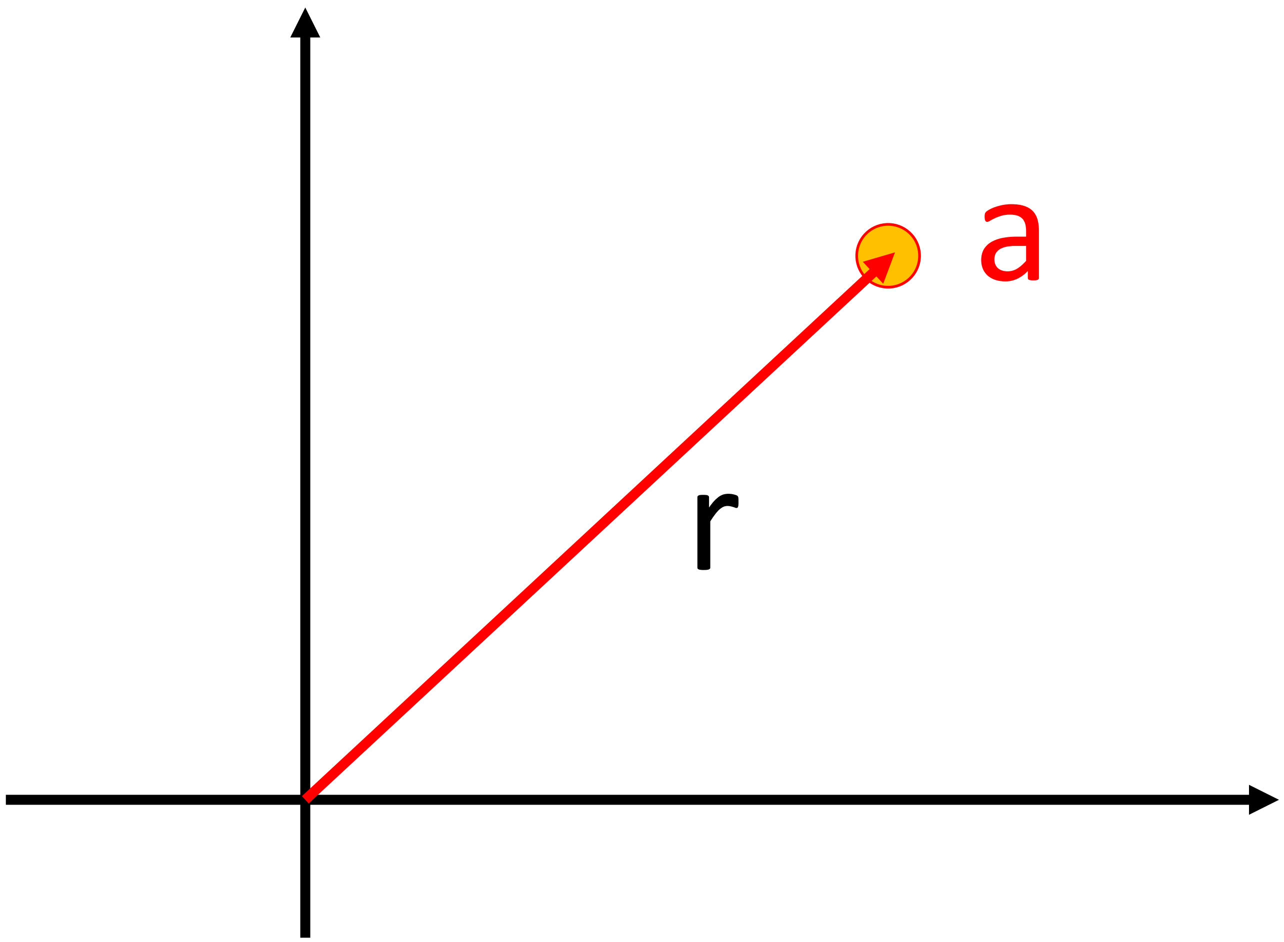
$$y_a = r\sin\alpha$$



Linear transformations in 2D: Rotate

$$x_a = r\cos\alpha$$

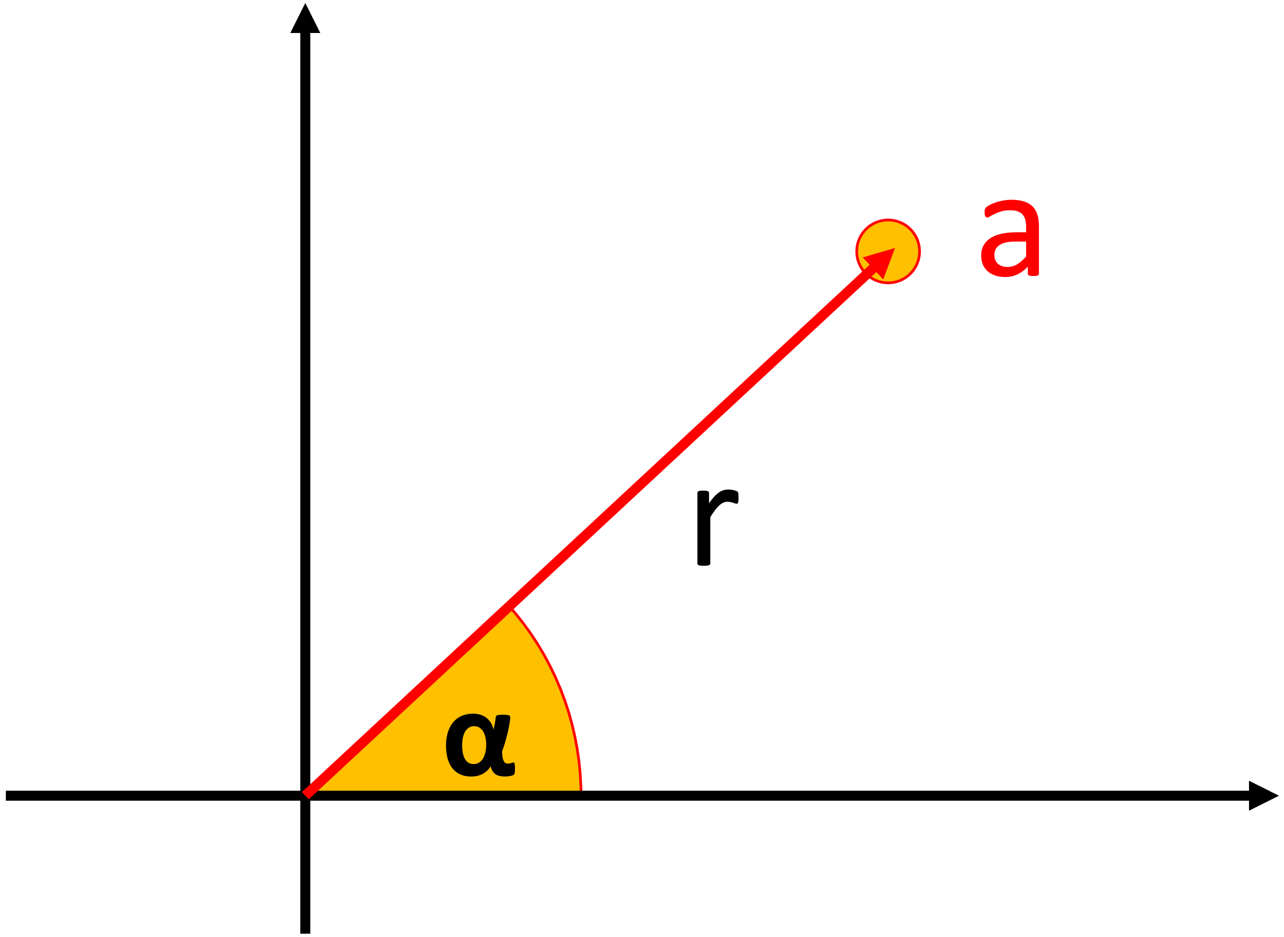
$$y_a = r\sin\alpha$$



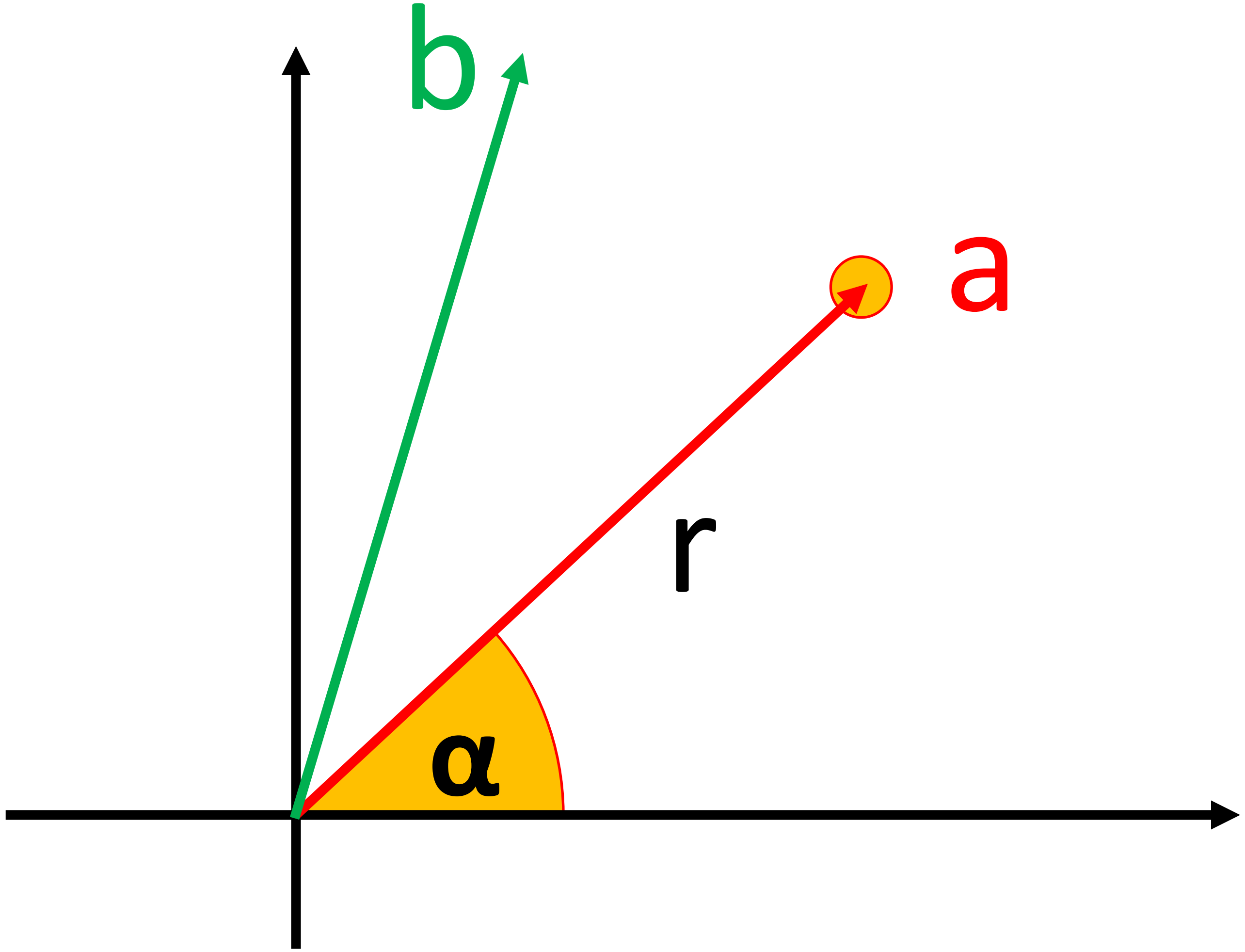
Linear transformations in 2D: Rotate

$$x_a = r\cos\alpha$$

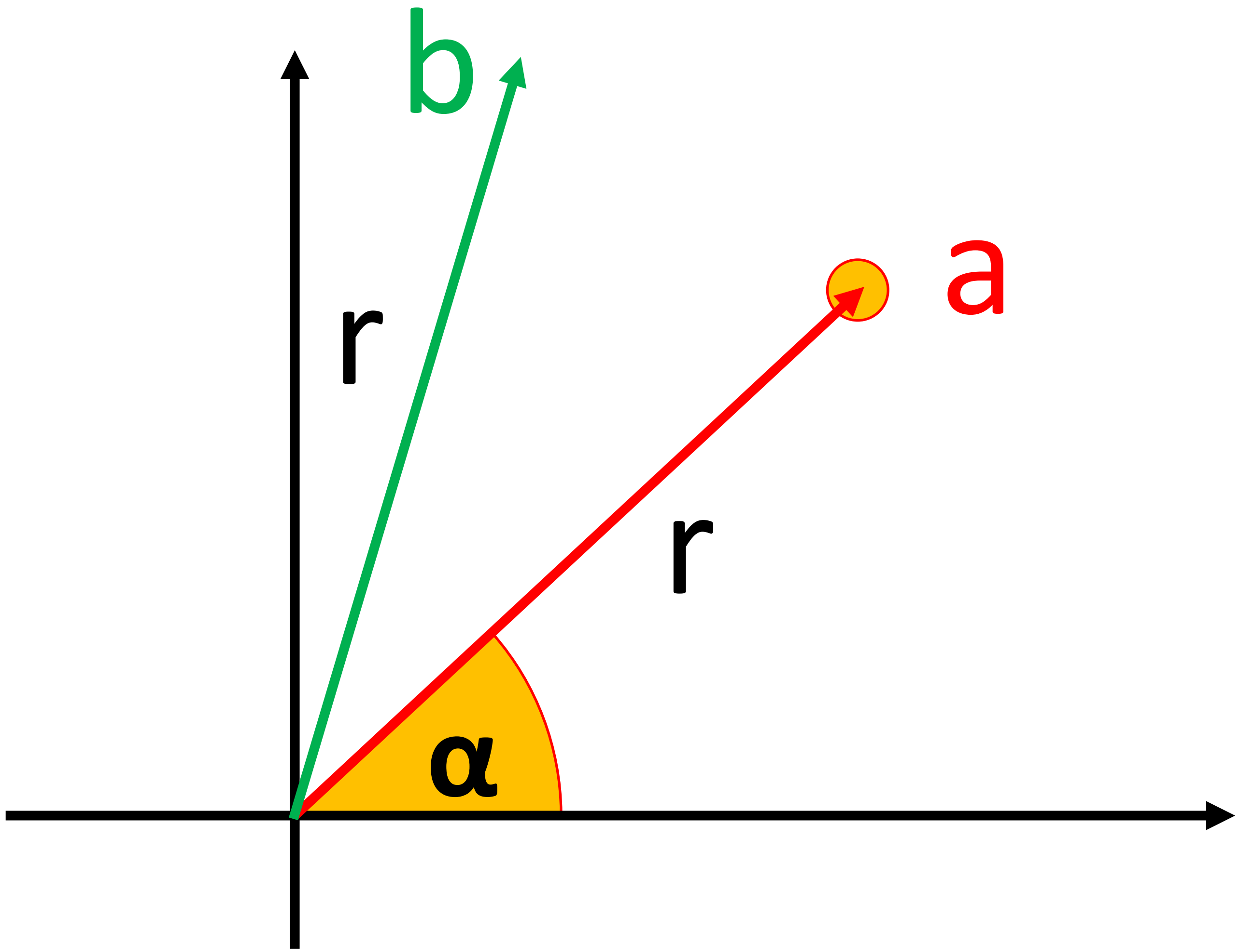
$$y_a = r\sin\alpha$$



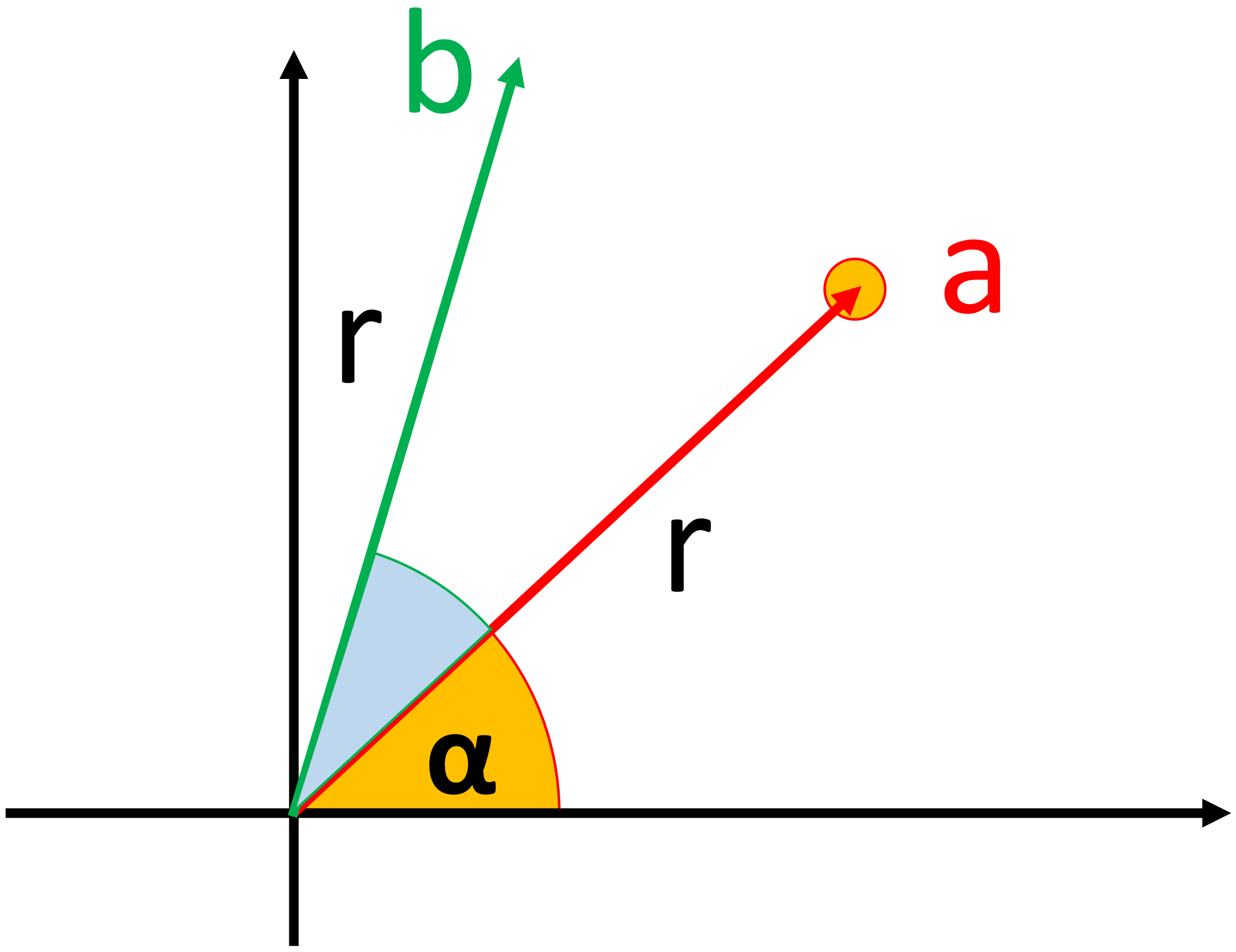
Linear transformations in 2D: Rotate



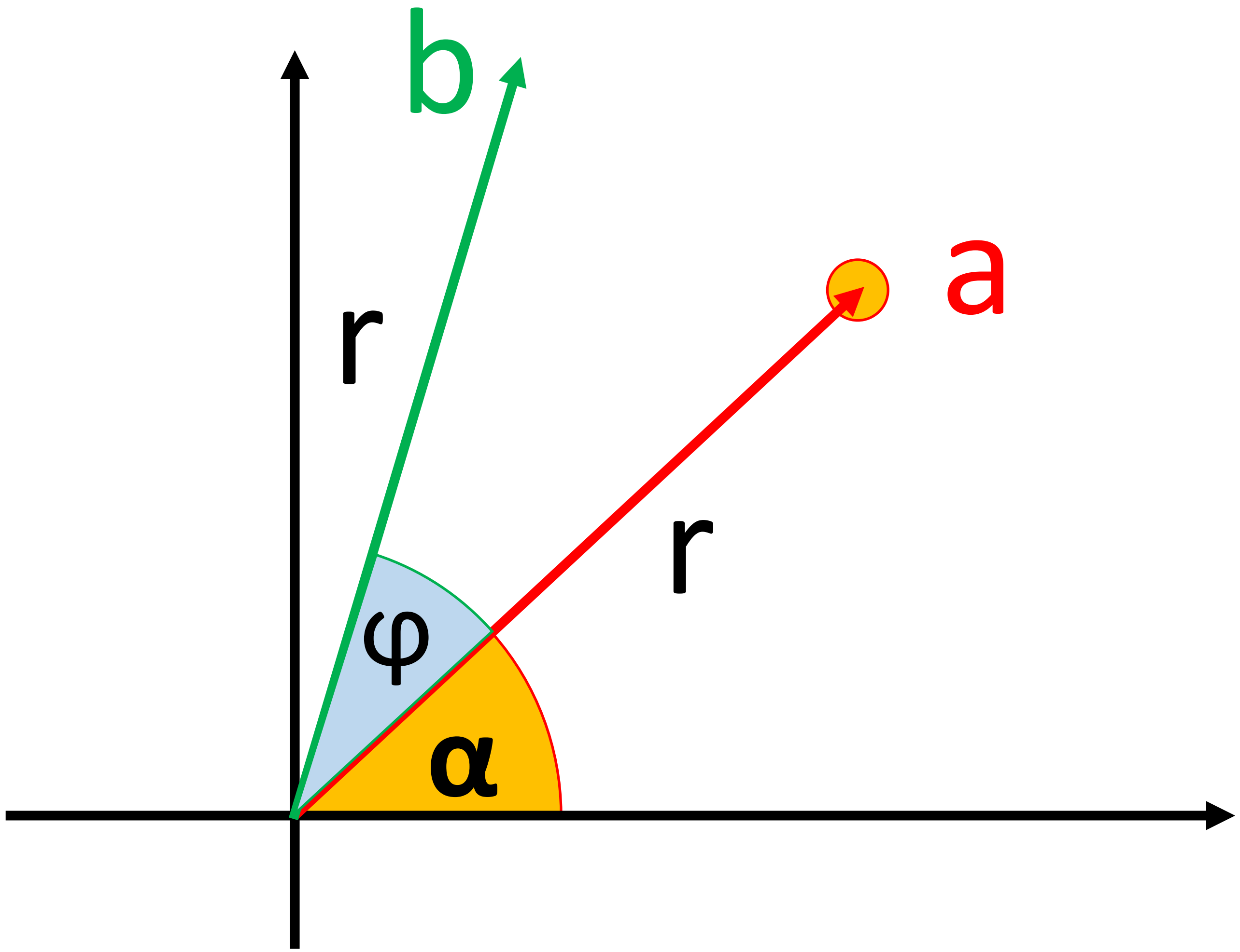
Linear transformations in 2D: Rotate



Linear transformations in 2D: Rotate



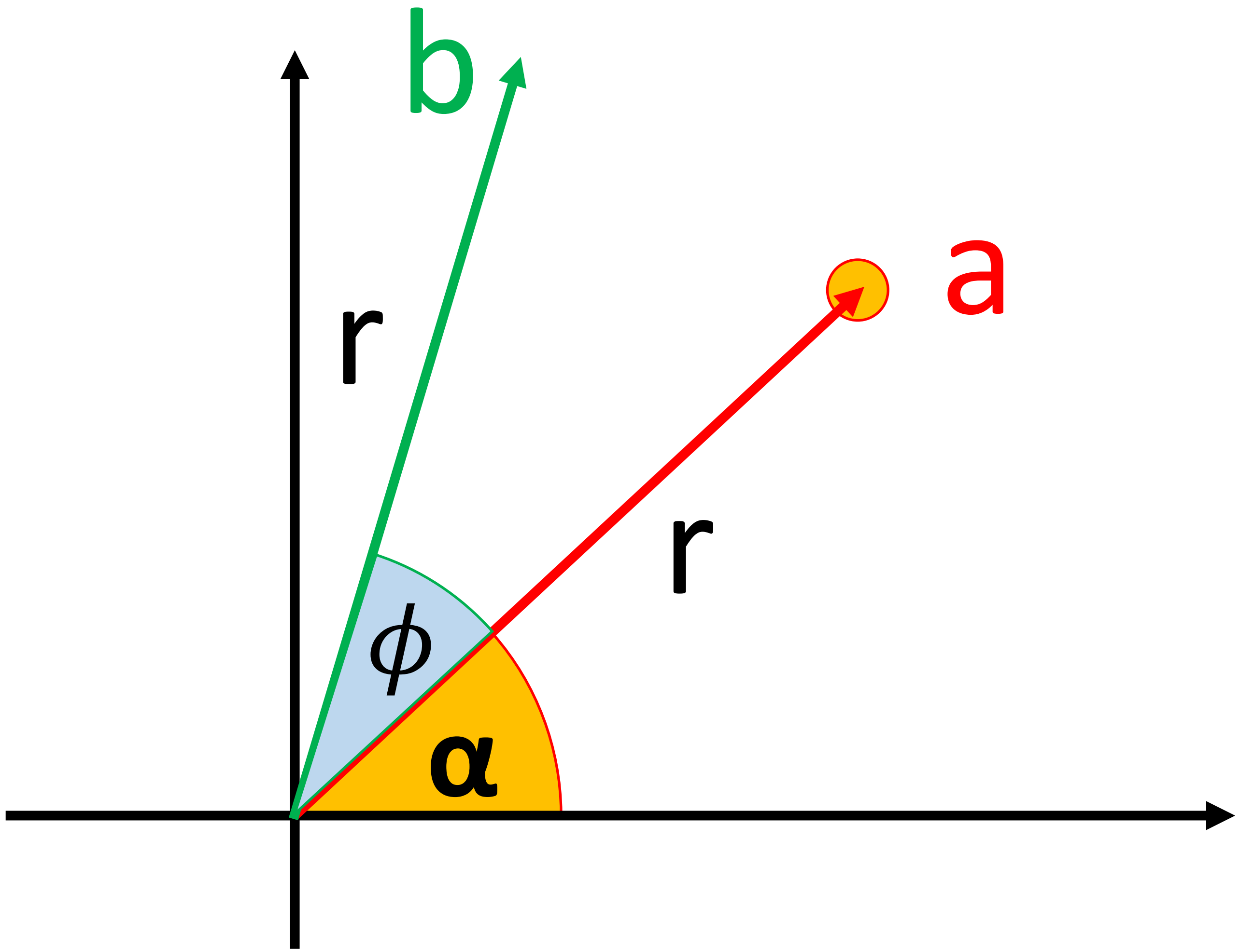
Linear transformations in 2D: Rotate



Linear transformations in 2D: Rotate

$$x_a = r\cos\alpha$$
$$y_a = r\sin\alpha$$

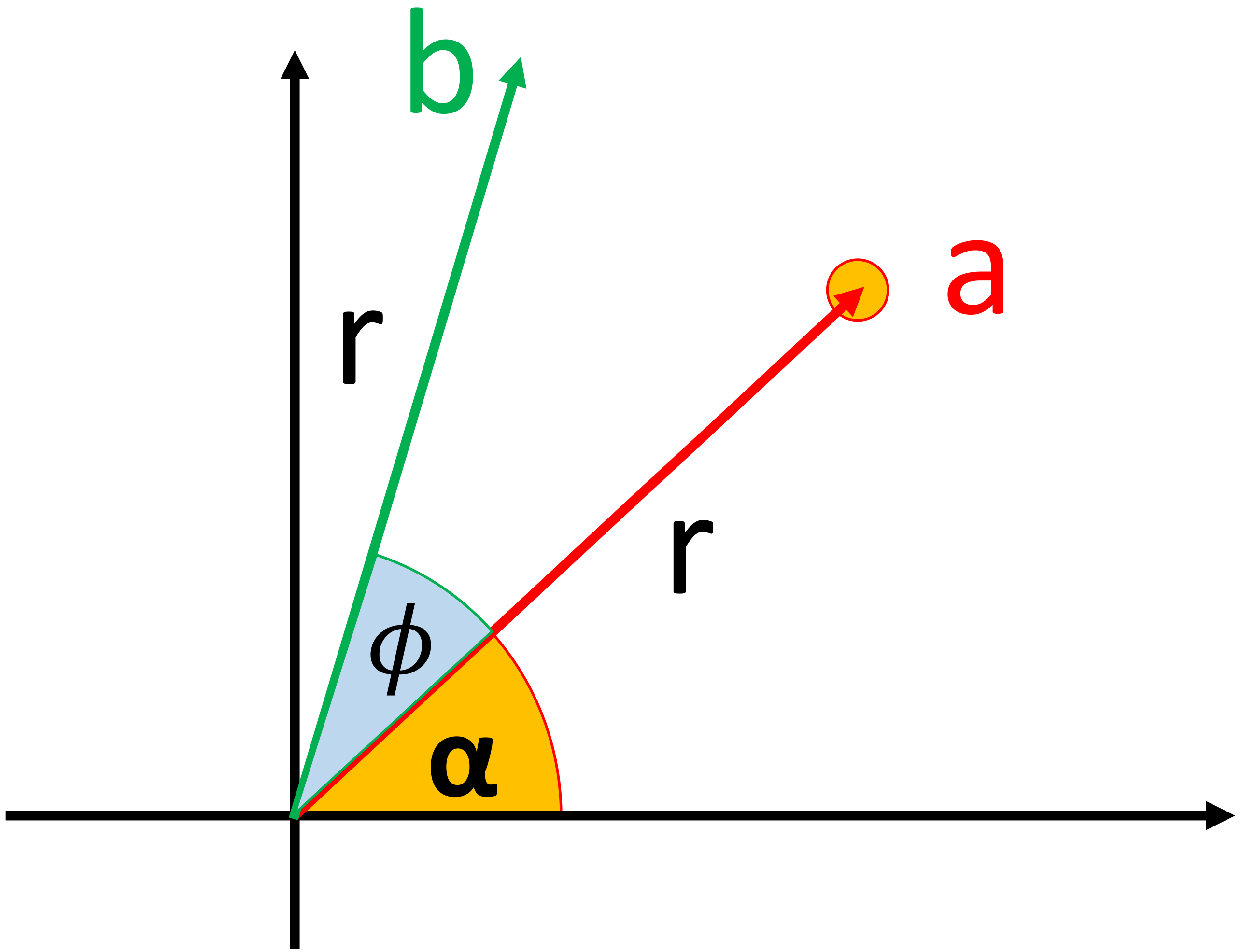
$$x_b = r\cos(\alpha+\phi)$$
$$y_b = r\sin(\alpha+\phi)$$



Linear transformations in 2D: Rotate

$$x_b = r\cos(\alpha + \phi)$$

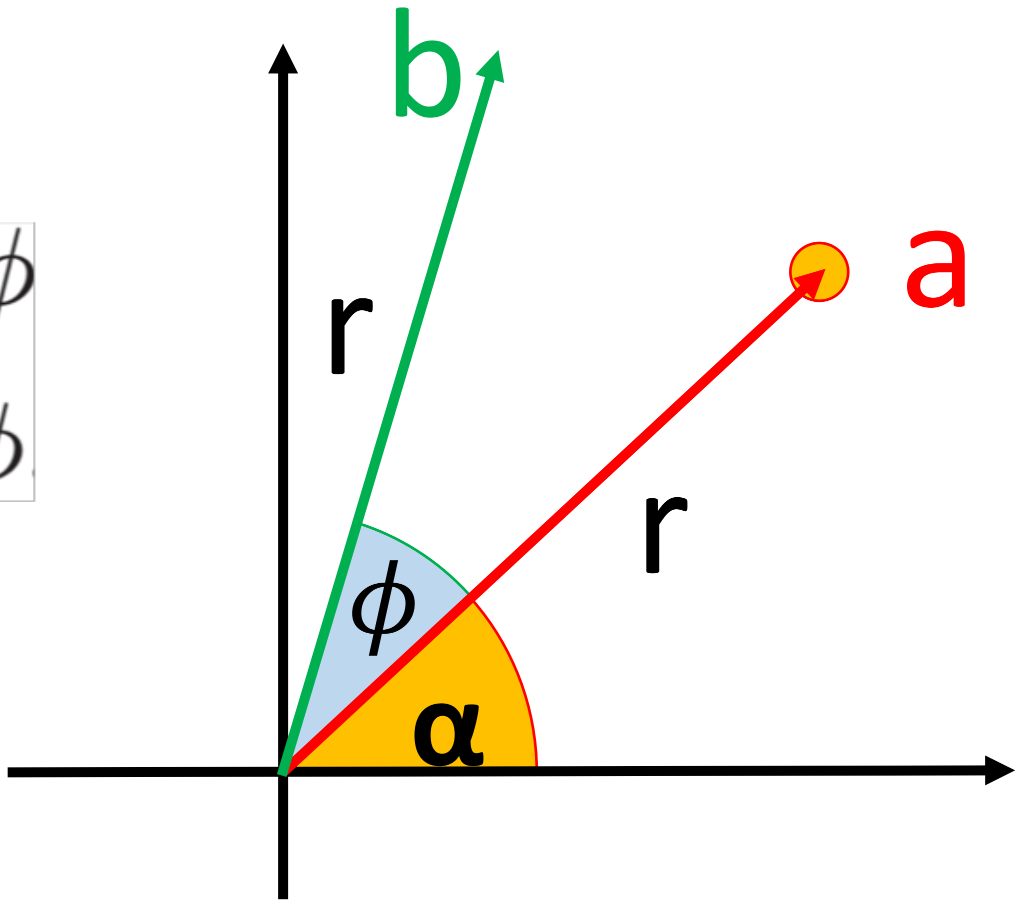
$$y_b = r\sin(\alpha + \phi)$$



Linear transformations in 2D: Rotate

$$x_b = r\cos\alpha\cos\phi - r\sin\alpha\sin\phi$$

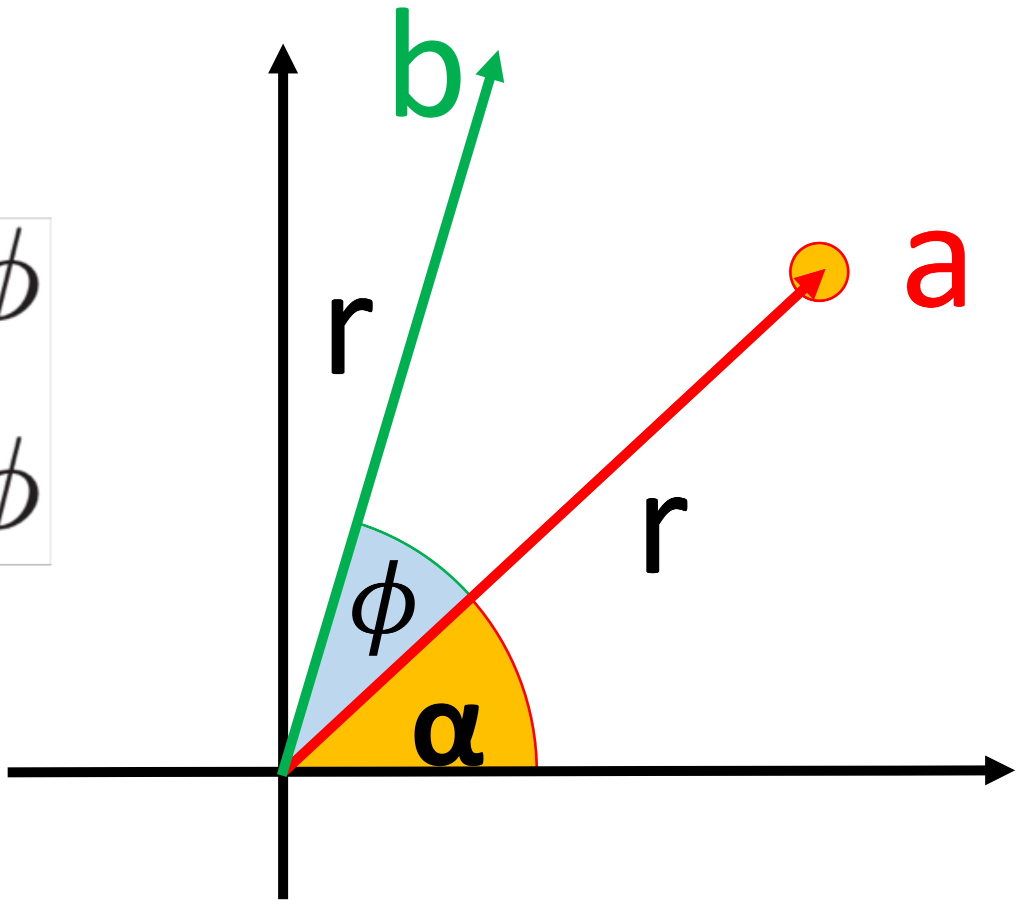
$$y_b = r\sin\alpha\cos\phi + r\cos\alpha\sin\phi$$



Linear transformations in 2D: Rotate

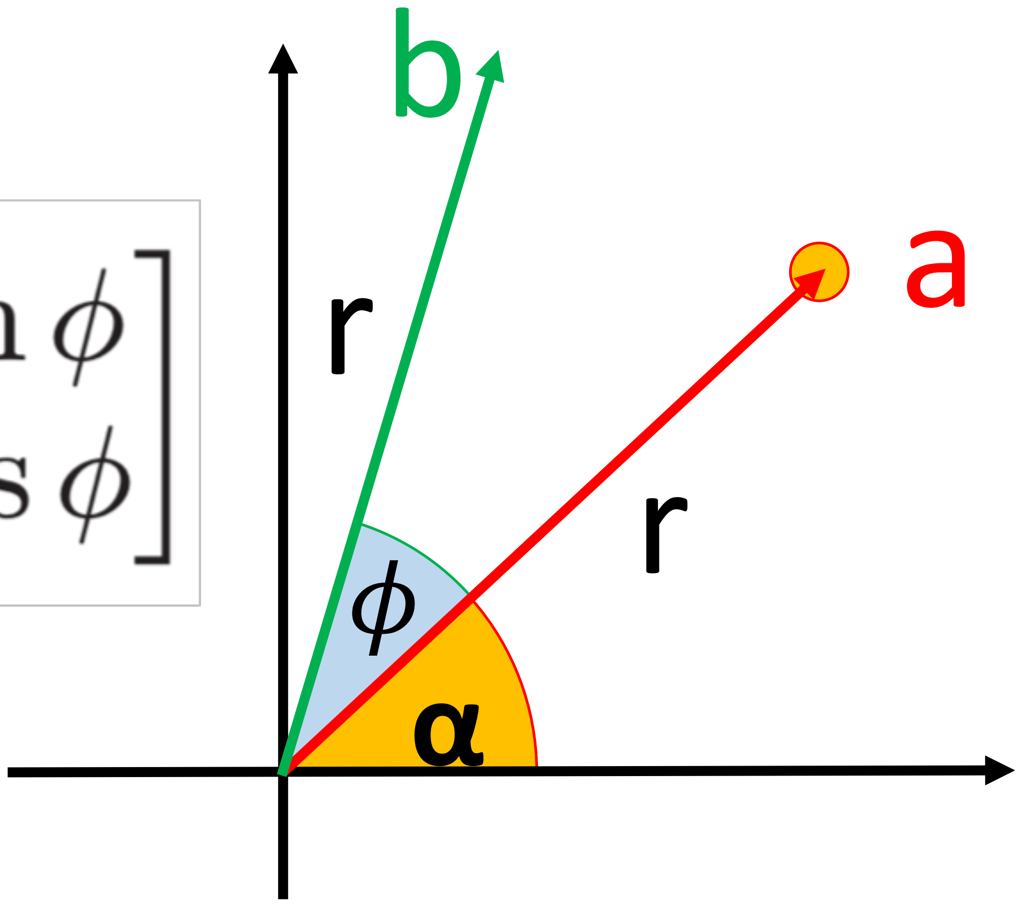
$$x_b = x_a \cos\phi - y_a \sin\phi$$

$$y_b = y_a \cos\phi + x_a \sin\phi$$



Linear transformations in 2D: Rotate

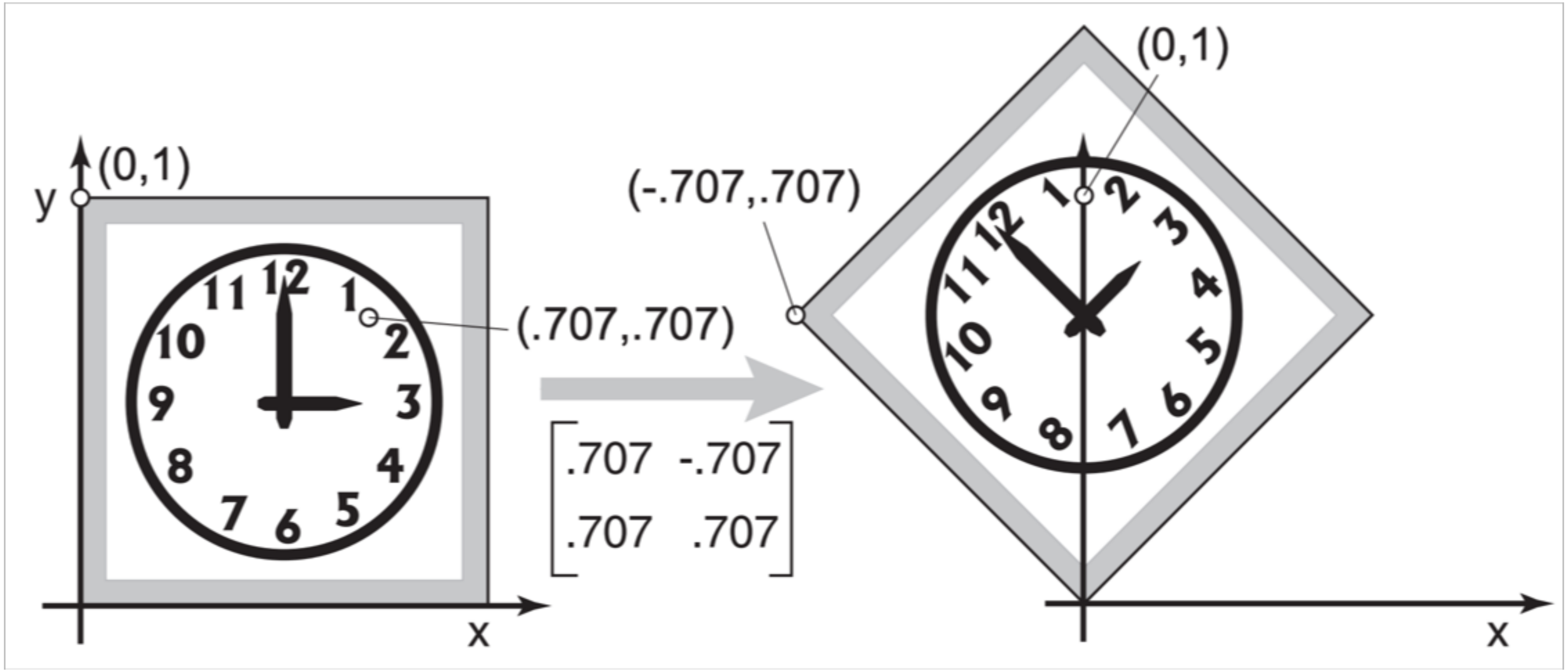
$$\mathrm{rotate}(\phi) = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix}$$



Linear transformations in 2D: Rotate

$$\begin{bmatrix} \cos\frac{\pi}{4} & -\sin\frac{\pi}{4} \\ \sin\frac{\pi}{4} & \cos\frac{\pi}{4} \end{bmatrix} = \begin{bmatrix} 0.707 & -0.707 \\ 0.707 & 0.707 \end{bmatrix}$$

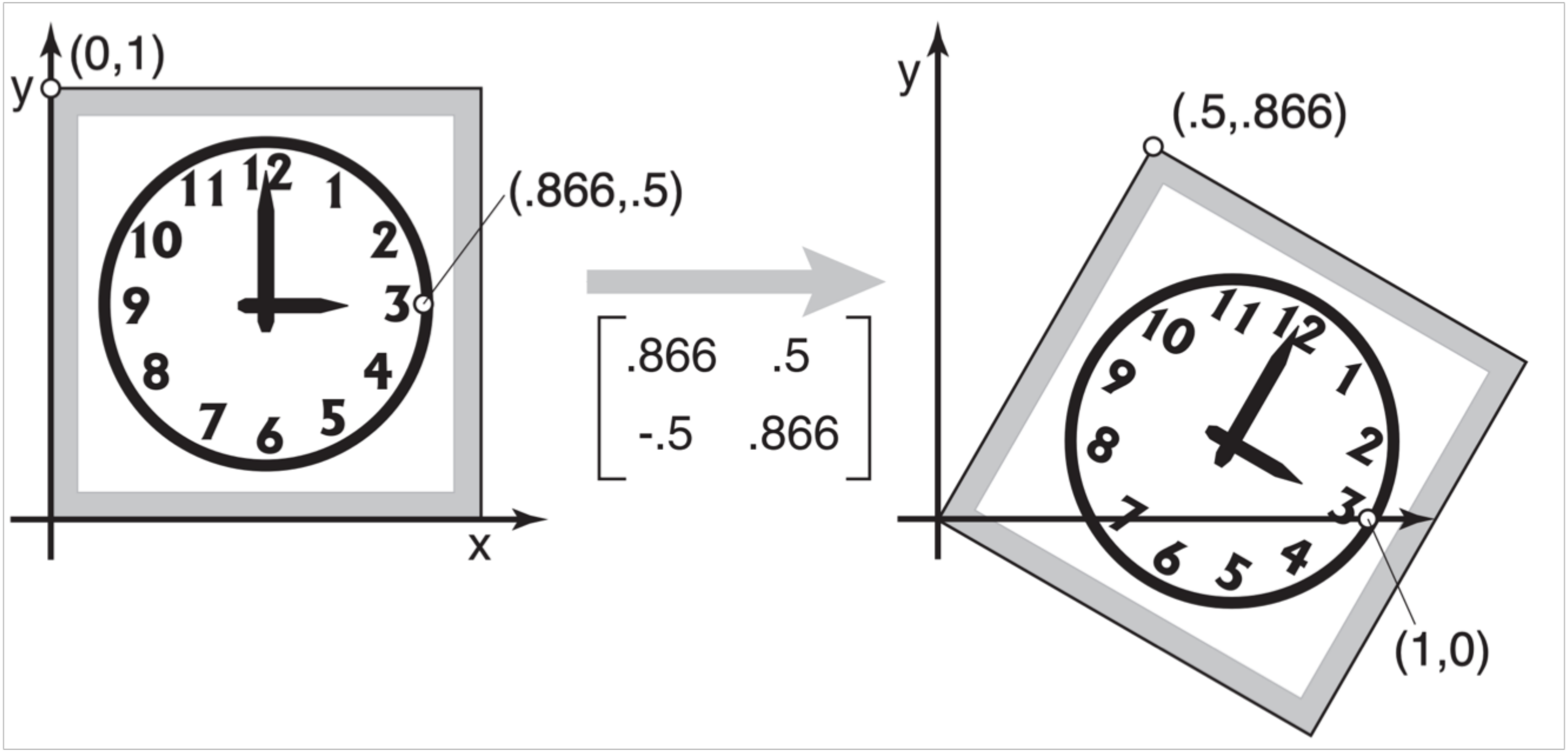
Linear transformations in 2D: Rotate



Linear transformations in 2D: Rotate

$$\begin{bmatrix} \cos\frac{-\pi}{6} & -\sin\frac{-\pi}{6} \\ \sin\frac{-\pi}{6} & \cos\frac{-\pi}{6} \end{bmatrix} = \begin{bmatrix} 0.866 & 0.5 \\ -0.5 & 0.866 \end{bmatrix}$$

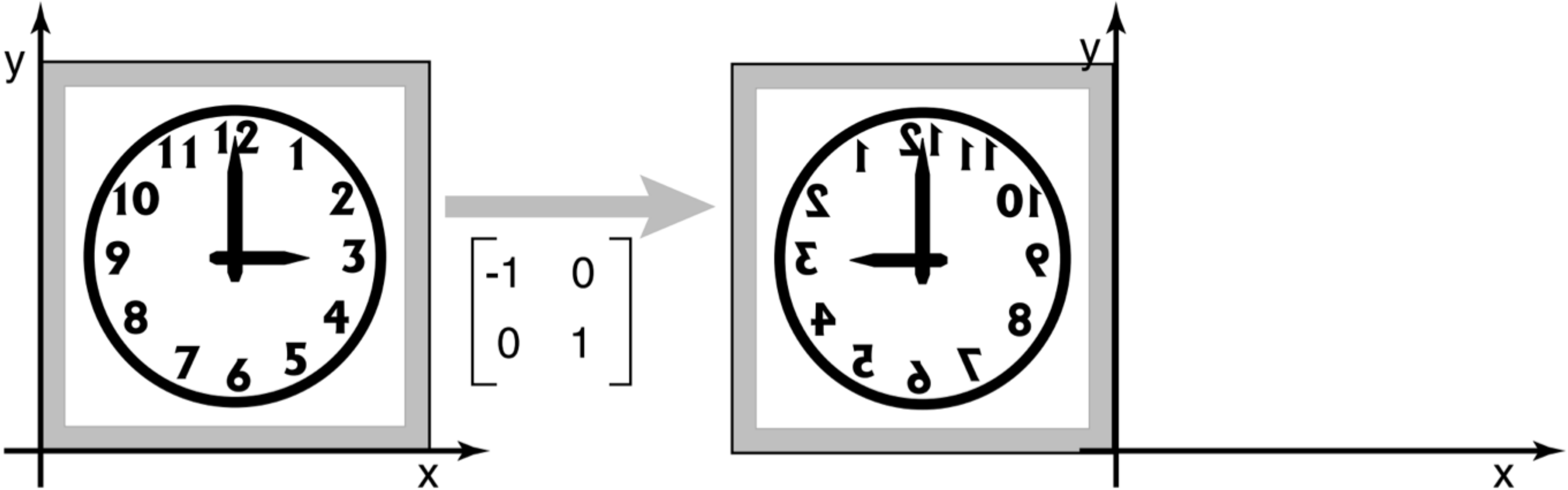
Linear transformations in 2D: Rotate



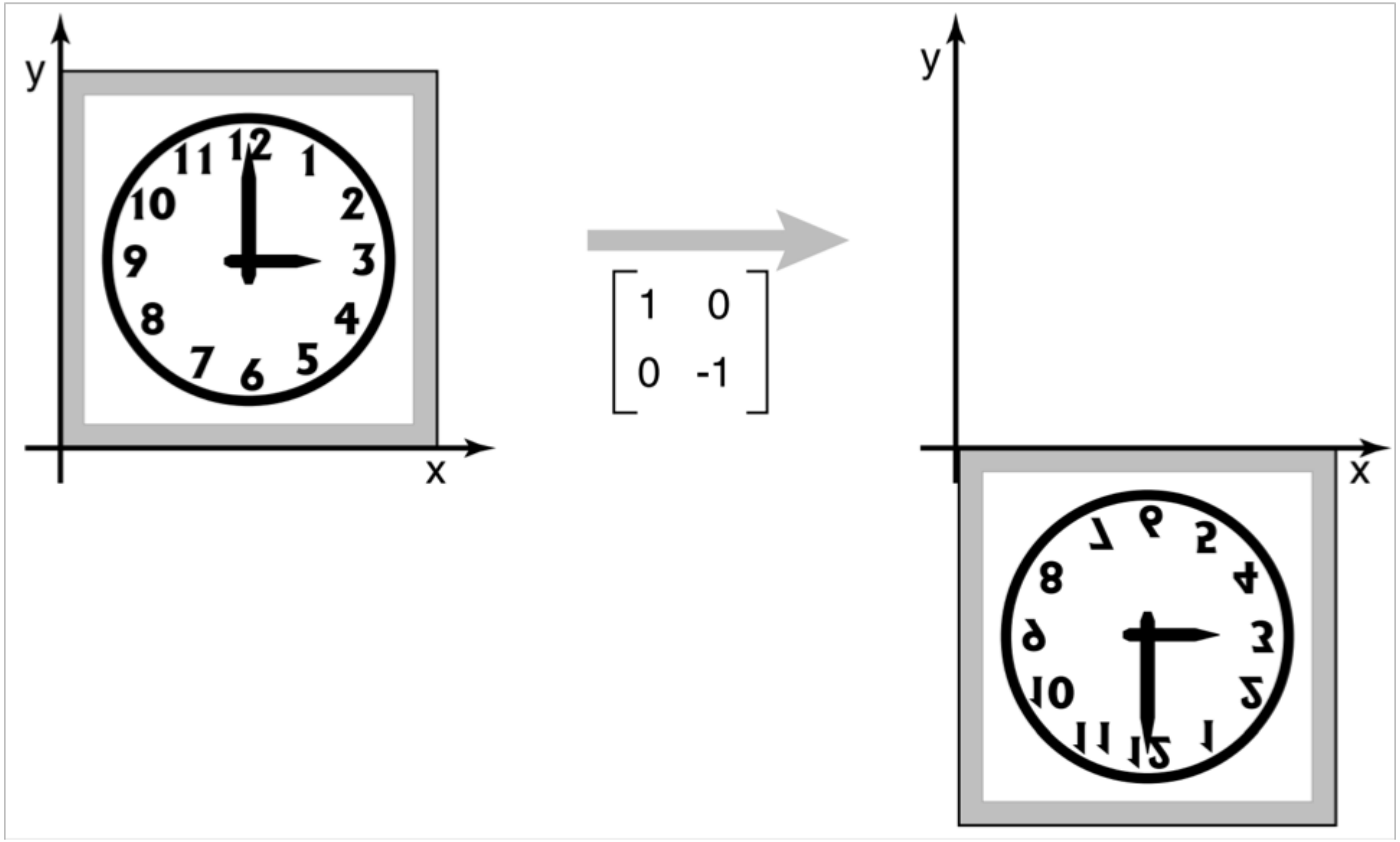
Linear transformations in 2D: Reflect

$$\textbf{reflect-y} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \textbf{reflect-x} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

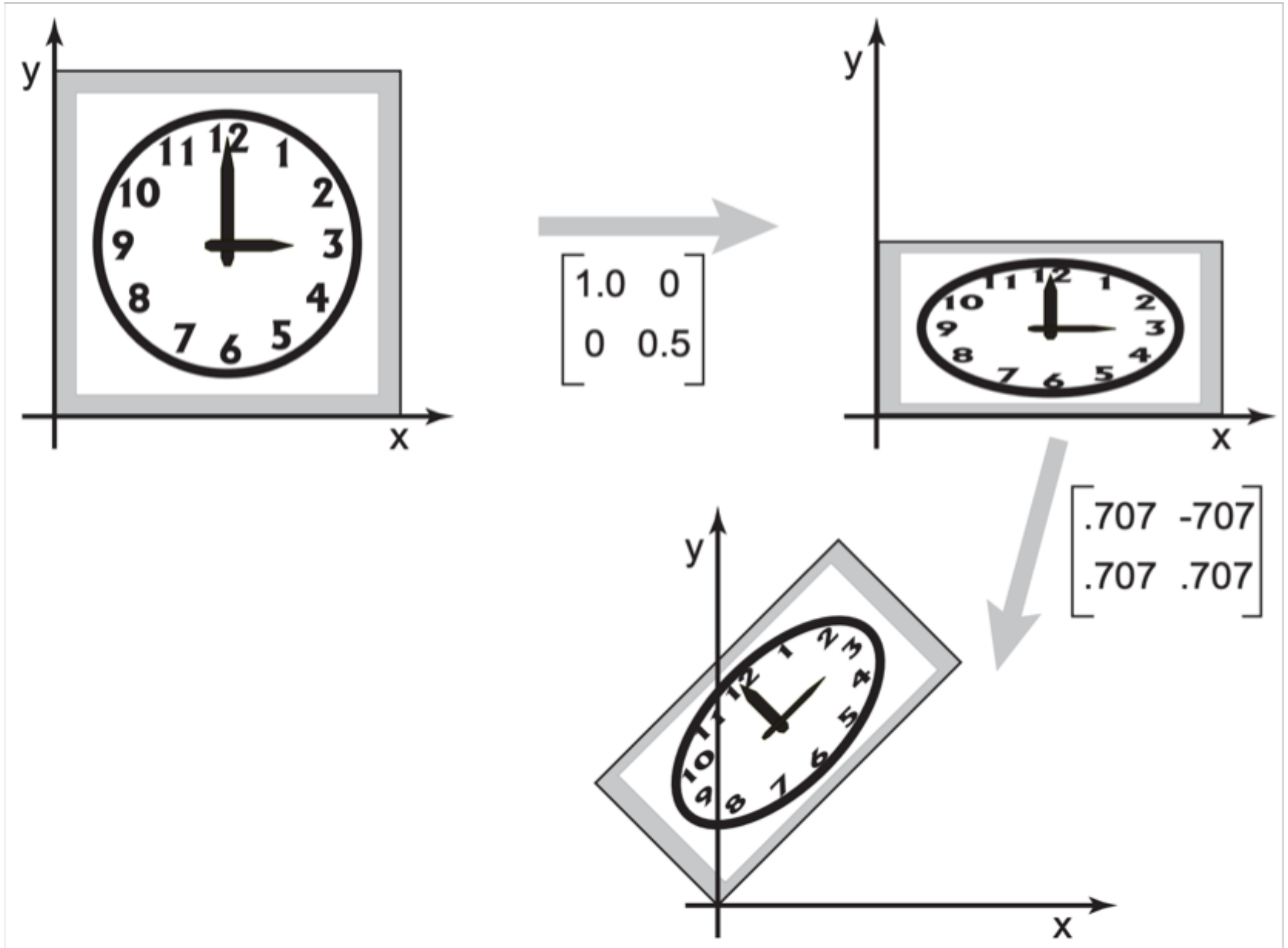
Linear transformations in 2D: Reflect



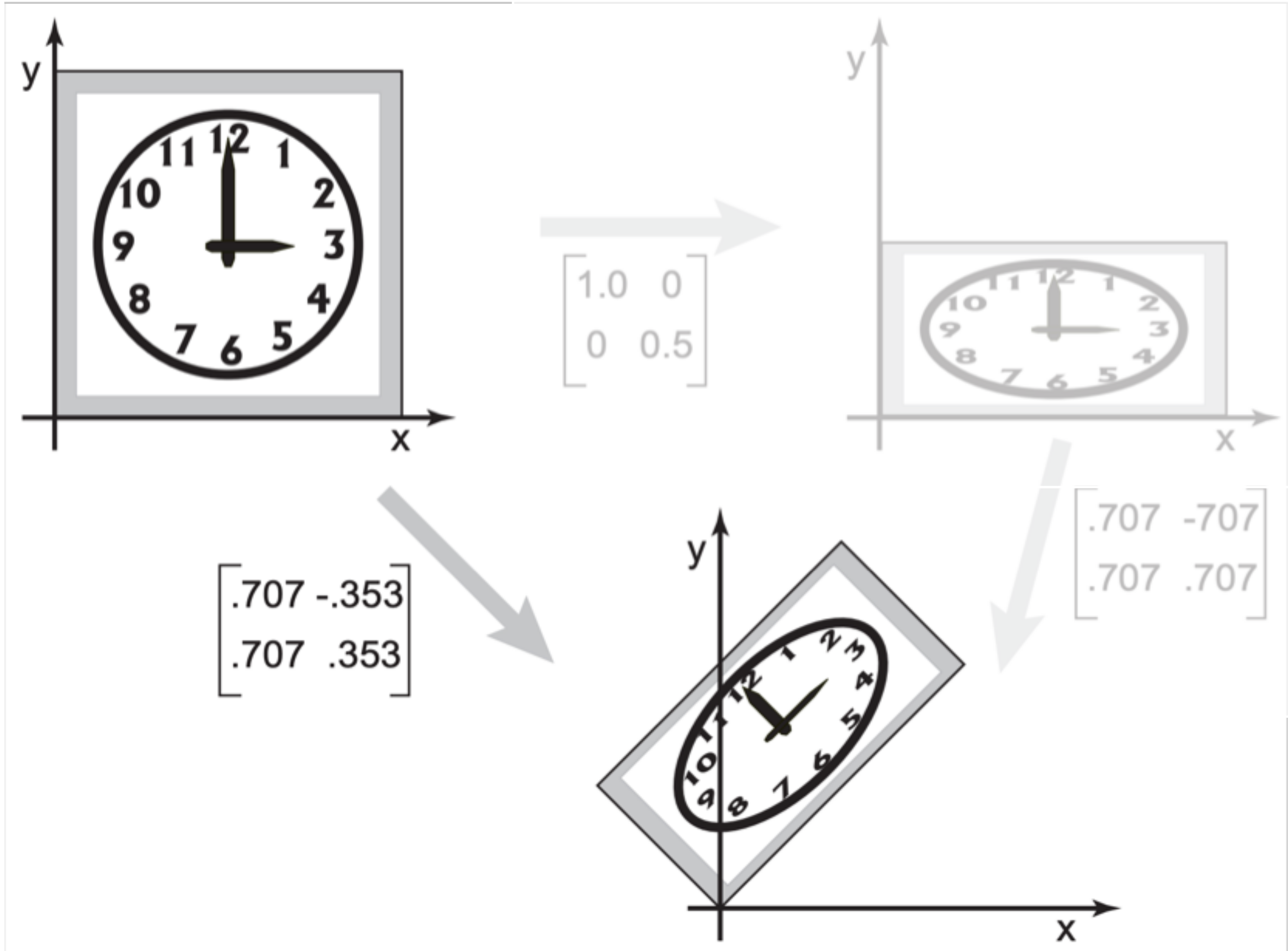
Linear transformations in 2D: Reflect



Composing transformations



Composing transformations



Composing transformations

$$\begin{bmatrix} .707 & -.707 \\ .707 & .707 \end{bmatrix} \begin{bmatrix} 1.0 & 0 \\ 0 & 0.5 \end{bmatrix} = \begin{bmatrix} .707 & -.353 \\ .707 & .353 \end{bmatrix}$$

Composing transformations

$$\begin{bmatrix} .707 & -.707 \\ .707 & .707 \end{bmatrix} \underbrace{\begin{bmatrix} 1.0 & 0 \\ 0 & 0.5 \end{bmatrix}}_{\text{1st transformation}} = \begin{bmatrix} .707 & -.353 \\ .707 & .353 \end{bmatrix}$$

Composing transformations

$$\underbrace{\begin{bmatrix} .707 & -.707 \\ .707 & .707 \end{bmatrix}}_{\text{2nd transformation}} \begin{bmatrix} 1.0 & 0 \\ 0 & 0.5 \end{bmatrix} = \begin{bmatrix} .707 & -.353 \\ .707 & .353 \end{bmatrix}$$

Composing transformations

$$\underbrace{\begin{bmatrix} .707 & -.707 \\ .707 & .707 \end{bmatrix}}_{\text{2nd transformation}} \underbrace{\begin{bmatrix} 1.0 & 0 \\ 0 & 0.5 \end{bmatrix}}_{\text{1st transformation}} = \begin{bmatrix} .707 & -.353 \\ .707 & .353 \end{bmatrix}$$

(someone makes this mistake every year)

Composing transformations

$$\underbrace{\begin{bmatrix} .707 & -.707 \\ .707 & .707 \end{bmatrix}}_{\text{2nd transformation}} \underbrace{\begin{bmatrix} 1.0 & 0 \\ 0 & 0.5 \end{bmatrix}}_{\text{1st transformation}} = \begin{bmatrix} .707 & -.353 \\ .707 & .353 \end{bmatrix}$$

(someone makes this mistake every year)

(I even messed it up the first time I made this slide)

Why that order?

$$Ax = b$$

Matrix multiplication is applied to the left of the vector

Composing transformations

$$\begin{bmatrix} .707 & -.707 \\ .707 & .707 \end{bmatrix} \begin{bmatrix} 1.0 & 0 \\ 0 & 0.5 \end{bmatrix} = \begin{bmatrix} .707 & -.353 \\ .707 & .353 \end{bmatrix}$$

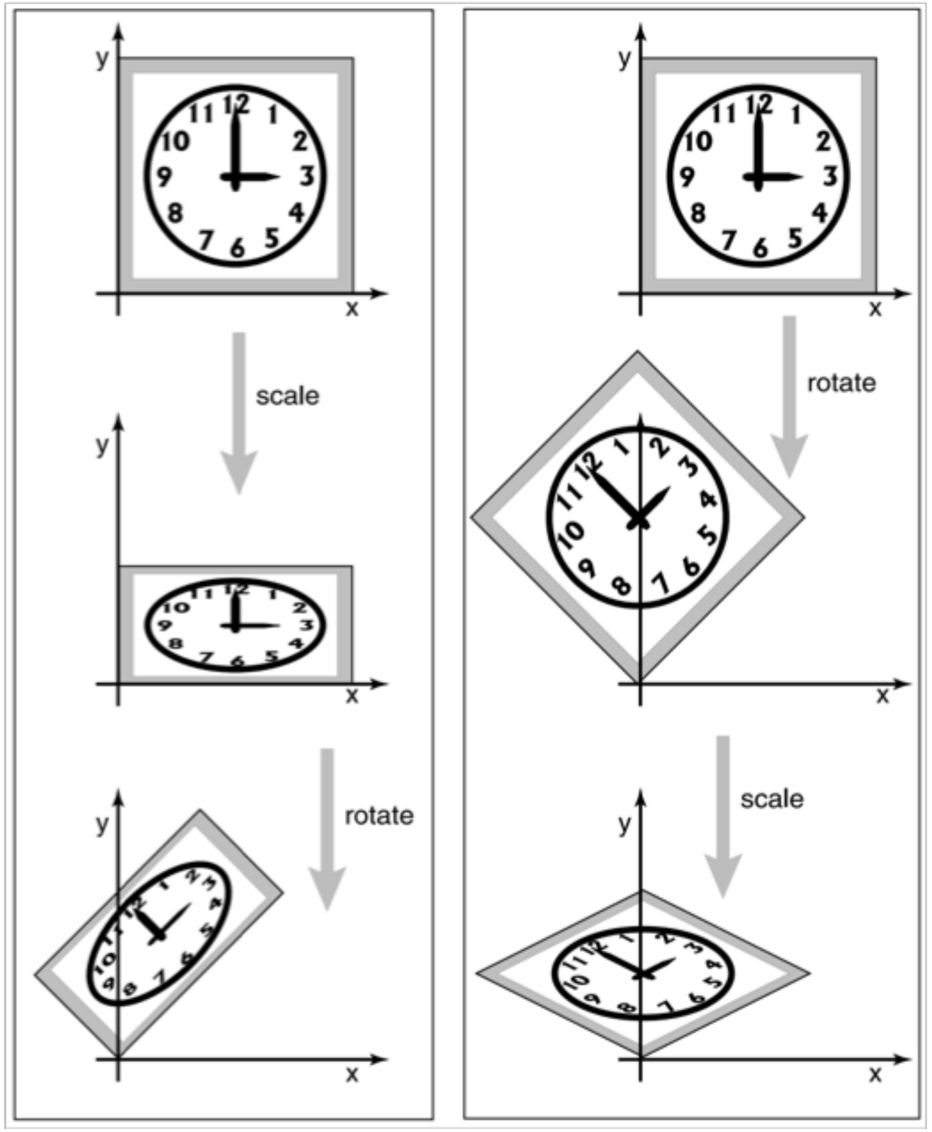
Composing transformations

$$\begin{bmatrix} 1.0 & 0 \\ 0 & 0.5 \end{bmatrix} \begin{bmatrix} .707 & -.707 \\ .707 & .707 \end{bmatrix} \neq \begin{bmatrix} .707 & -.353 \\ .707 & .353 \end{bmatrix}$$

Composing transformations

$$\begin{bmatrix} .707 & -.707 \\ .707 & .707 \end{bmatrix} \begin{bmatrix} 1.0 & 0 \\ 0 & 0.5 \end{bmatrix}$$

$$= \begin{bmatrix} .707 & -.353 \\ .707 & .353 \end{bmatrix}$$



$$\begin{bmatrix} 1.0 & 0 \\ 0 & 0.5 \end{bmatrix} \begin{bmatrix} .707 & -.707 \\ .707 & .707 \end{bmatrix}$$

$$= \begin{bmatrix} .707 & -.707 \\ .353 & .353 \end{bmatrix}$$

Pop quiz!

(Not for marks)

Write down a non-zero rotation and a non-uniform scale that do commute

Example:

Rotation by 180º

Scale in x by 0.5

Scale in y by 2

Linear transformations in 2D: Inverses

Inverse Scaling

$$\begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix}^{-1} = \begin{bmatrix} \frac{1}{s_x} & 0 \\ 0 & \frac{1}{s_y} \end{bmatrix}$$

Linear transformations in 2D: Inverses

Inverse Shearing

$$\begin{bmatrix} 1 & s \\ 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & -s \\ 0 & 1 \end{bmatrix}$$

Linear transformations in 2D: Inverses

Inverse Rotation

$$\begin{bmatrix} cos(\phi) & -sin(\phi) \\ sin(\phi) & cos(\phi) \end{bmatrix}^{-1} = \begin{bmatrix} cos(-\phi) & -sin(-\phi) \\ sin(-\phi) & cos(-\phi) \end{bmatrix}$$

Linear transformations in 2D: Inverses

Inverse Reflection

$$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$

Linear transformations in 3D: Scaling

2D

$$\begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix}$$

Linear transformations in 3D: Scaling

2D

$$\begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix}$$

3D

$$\begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{bmatrix}$$

Linear transformations in 3D: Shearing

3D shear in x coord

Horizontal 2D shear

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

Linear transformations in 3D: Shearing

Horizontal 2D shear

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

3D shear in x coord

$$\begin{bmatrix} 1 & s_y & s_z \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Linear transformations in 3D: Rotation

2D rotation

3D rotation around z axis

$$\begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix}$$

Linear transformations in 3D: Rotation

2D rotation

$$\begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix}$$

3D rotation around z axis

$$\begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Linear transformations in 3D: Rotation

3D rotation around z axis

$$\begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

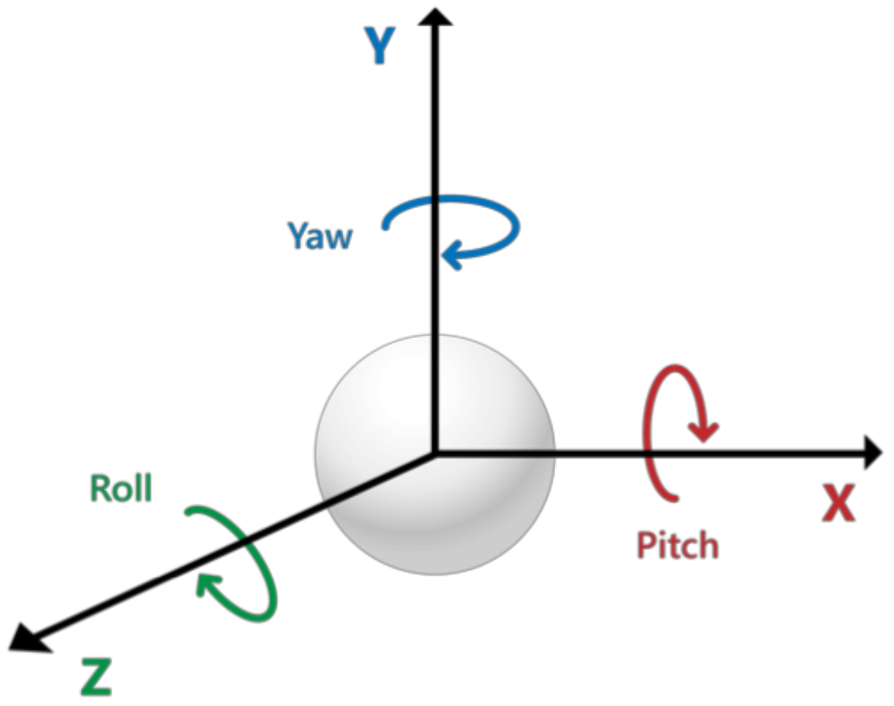


Image curtesy of Vangos Pterneas

https://pterneas.com/2017/05/28/kinect-joint-rotation/

Linear transformations in 3D: Rotation

3D rotation around x axis

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix}$$

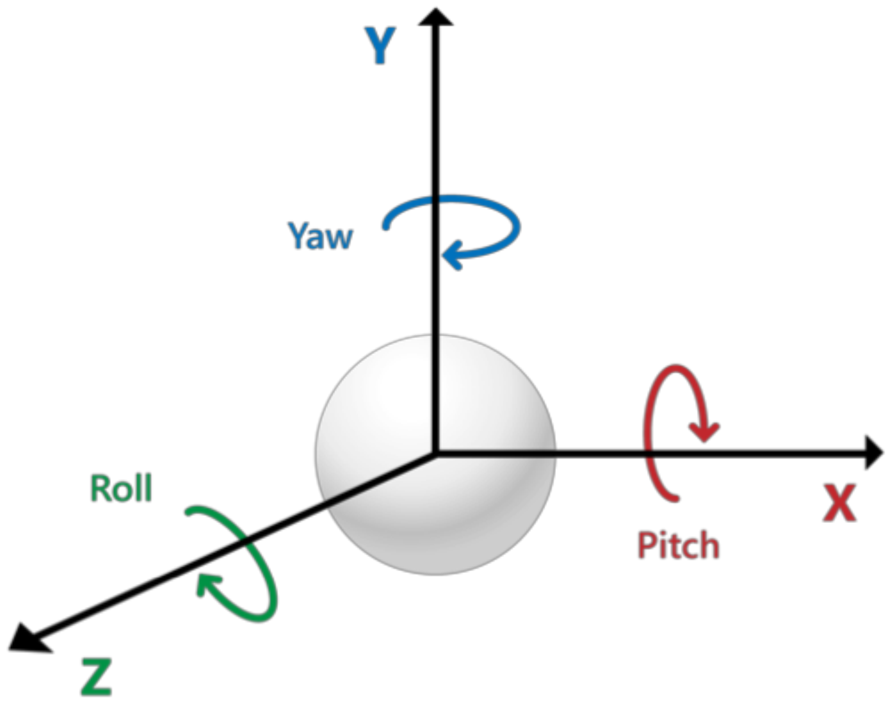


Image curtesy of Vangos Pterneas

https://pterneas.com/2017/05/28/kinect-joint-rotation/

Linear transformations in 3D: Rotation

3D rotation around y axis

$$\begin{bmatrix} \cos\phi & 0 & \sin\phi \\ 0 & 1 & 0 \\ -\sin\phi & 0 & \cos\phi \end{bmatrix}$$

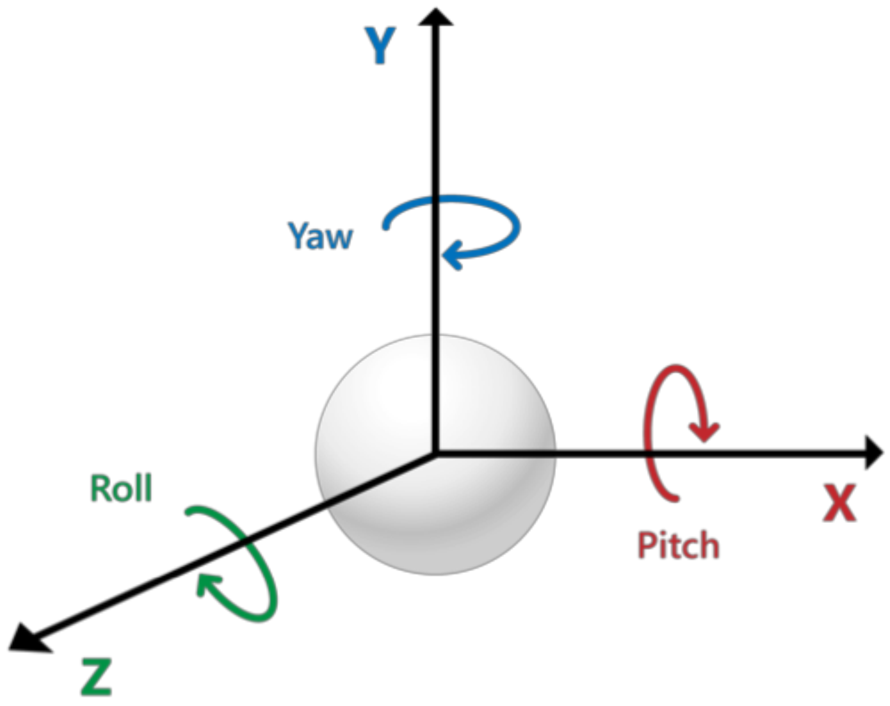


Image curtesy of Vangos Pterneas

https://pterneas.com/2017/05/28/kinect-joint-rotation/

Linear transformations in 3D: Rotation

2D rotation

$$\begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix}$$

3D rotation around y axis

$$\begin{bmatrix} \cos\phi & 0 & \sin\phi \\ 0 & 1 & 0 \\ -\sin\phi & 0 & \cos\phi \end{bmatrix}$$

Linear transformations in 3D: Rotation

2D rotation

$$\begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix}$$

3D rotation around y axis

$$\begin{bmatrix} \cos\phi & 0 & \sin\phi \\ 0 & 1 & 0 \\ -\sin\phi & 0 & \cos\phi \end{bmatrix}$$

Linear transformations in 3D: Rotation

2D rotation

$$\begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix}$$

3D rotation around y axis

$$\begin{bmatrix} \cos\phi & 0 & \sin\phi \\ 0 & 1 & 0 \\ -\sin\phi & 0 & \cos\phi \end{bmatrix}$$

?

Linear transformations in 3D: Rotation

3D rotation around x axis

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\phi) & -sin(\phi) \\ 0 & sin(\phi) & cos(\phi) \end{bmatrix}$$

Linear transformations in 3D: Rotation

3D rotation around x axis

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\phi) & -sin(\phi) \\ 0 & sin(\phi) & cos(\phi) \end{bmatrix}$$

"Cycle" the matrix right

Linear transformations in 3D: Rotation

3D rotation around x axis

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\phi) & -sin(\phi) \\ 0 & sin(\phi) & cos(\phi) \end{bmatrix} \xrightarrow[\text{matrix right}]{\text{“Cycle” the}} \begin{bmatrix} 0 & 1 & 0 \\ -sin(\phi) & 0 & cos(\phi) \\ cos(\phi) & 0 & sin(\phi) \end{bmatrix}$$

Linear transformations in 3D: Rotation

3D rotation around x axis

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\phi) & -sin(\phi) \\ 0 & sin(\phi) & cos(\phi) \end{bmatrix}$$

"Cycle" the matrix right

$$\begin{bmatrix} 0 & 1 & 0 \\ -sin(\phi) & 0 & cos(\phi) \\ cos(\phi) & 0 & sin(\phi) \end{bmatrix}$$

"Cycle" the matrix down

Linear transformations in 3D: Rotation

3D rotation around x axis

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\phi) & -sin(\phi) \\ 0 & sin(\phi) & cos(\phi) \end{bmatrix}$$

"Cycle" the matrix right

$$\begin{bmatrix} 0 & 1 & 0 \\ -sin(\phi) & 0 & cos(\phi) \\ cos(\phi) & 0 & sin(\phi) \end{bmatrix}$$

"Cycle" the matrix down

$$\begin{bmatrix} cos(\phi) & 0 & sin(\phi) \\ 0 & 1 & 0 \\ -sin(\phi) & 0 & cos(\phi) \end{bmatrix}$$

Linear transformations in 3D: Rotation

3D rotation around x axis

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\phi) & -sin(\phi) \\ 0 & sin(\phi) & cos(\phi) \end{bmatrix}$$

"Cycle" the matrix right

$$\begin{bmatrix} 0 & 1 & 0 \\ -sin(\phi) & 0 & cos(\phi) \\ cos(\phi) & 0 & sin(\phi) \end{bmatrix}$$

"Cycle" the matrix down

$$\begin{bmatrix} cos(\phi) & 0 & sin(\phi) \\ 0 & 1 & 0 \\ -sin(\phi) & 0 & cos(\phi) \end{bmatrix}$$

3D rotation around y axis

Cycle order:

x -> y -> z -> x -> ...

Did we miss any important transformations?

Did we miss any important transformations?

Yes, translation!

Did we miss any important transformations?

Yes, translation!

Gavin, what gives? Where’s the translations?

Translations are **not** linear transformations

Translations are **not** linear transformations

Recall, linear transformations can be expressed as a matrix

$$Ax = b$$

Translations are **not** linear transformations

Imagine we had a matrix T
for performing translations

$$T \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x + t_x \\ y + t_y \end{bmatrix}$$

Translations are <u>**not**</u> linear transformations

So then:

$$T \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

Translations are **not** linear transformations

However:

$$T\begin{bmatrix}0\\0\end{bmatrix} = \begin{bmatrix}0\\0\end{bmatrix} \neq \begin{bmatrix}t_x\\t_y\end{bmatrix}$$

Okay, who cares?

Okay, who cares?

Why not just add a vector at the end of the transformation?

$$Ax + t = b$$

Okay, who cares?

Why not just add a vector at the end of the transformation?

$$Ax + t = b$$

$$A_4(A_3(A_2(A_1(x + t_1) + t_2) + t_3) + t_4 = b$$

Okay, who cares?

Why not just add a vector at the end of the transformation?

$$Ax + t = b$$

$$A_4(A_3(A_2(A_1(x + t_1) + t_2) + t_3) + t_4 = b$$
$$A_4A_3A_2A_1x + A_4A_3A_2t_1 + A_4A_3t_2 + A_4t_3 + t_4 = b$$

This is feasible, but the bookkeeping is awkward and the rule for composition is not as simple

Homogeneous Coordinates:

$$\begin{bmatrix} x \\ y \end{bmatrix}$$

Homogeneous Coordinates:

$$\begin{bmatrix} x \\ y \end{bmatrix} \xrightarrow[\text{coordinate}]{\text{Add a 3}^{\text{rd}}} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Homogeneous Coordinates:

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

Homogeneous Coordinates:

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \xrightarrow{\text{Bring to three dimensions, add a translation}} \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

Homogeneous Coordinates:

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \xrightarrow{\text{Bring to three dimensions, add a translation}} \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

$$Ax + t = b$$

Homogeneous Coordinates:

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \xrightarrow{\text{Bring to three dimensions, add a translation}} \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

$$Ax + t = b$$

Homogeneous Coordinates:

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

Homogeneous Coordinates:

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

Homogeneous Coordinates:

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

$$Ax + t = b$$

2D translation:

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

$$Ax + t = b$$

Translation Matrix:

$$\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x + t_x \\ y + t_y \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \end{bmatrix}$$

Translation Matrix:

$$\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x + t_x \\ y + t_y \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \end{bmatrix} \xrightarrow[\text{coordinate}]{\text{Add a 3}^{\text{rd}}} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Translation Matrix:

$$\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x + t_x \\ y + t_y \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \end{bmatrix} \xrightarrow[\text{coordinate}]{\text{Add a 3rd}} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \xrightarrow{\text{Transform}} \begin{bmatrix} x + t_x \\ y + t_y \\ 1 \end{bmatrix}$$

Translation Matrix:

$$\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x + t_x \\ y + t_y \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \end{bmatrix} \xrightarrow[\text{coordinate}]{\text{Add a 3}^{\text{rd}}} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \xrightarrow{\text{Transform}} \begin{bmatrix} x + t_x \\ y + t_y \\ 1 \end{bmatrix} \xrightarrow[\text{coordinate}]{\text{Drop 3}^{\text{rd}}} \begin{bmatrix} x + t_x \\ y + t_y \end{bmatrix}$$

What about the other transformations?

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \xrightarrow{\text{Bring to three dimensions, no translation}} \begin{bmatrix} a_{11} & a_{12} & 0 \\ a_{21} & a_{22} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$Ax = b$$

What about the other transformations?

Example: scaling matrix

$$\begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \rightarrow \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

What about the other transformations?

Example: rotation matrix

$$\begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix} \rightarrow \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Is there a name for this special coordinate?

Is there a name for this special coordinate?

Yes, this 3^{rd} coordinate is called the **homogeneous coordinate**

Is there a name for this special coordinate?

Yes, this 3rd coordinate is called the **homogeneous coordinate**

$$\begin{bmatrix} x \\ y \end{bmatrix} \xrightarrow[\text{homogeneous coordinates}]{\text{Considered as a point in 3D}} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

And what kind of transformation is this if not linear?

And what kind of transformation is this if not linear?

A transformation of the following form is called an **Affine transformation**

$$Ax + t = b$$

Checking for lingo:

An affine transformation can be represented as a linear transformation in homogeneous coordinates.

Checking for lingo:

An affine transformation can be represented as a linear transformation in homogeneous coordinates.

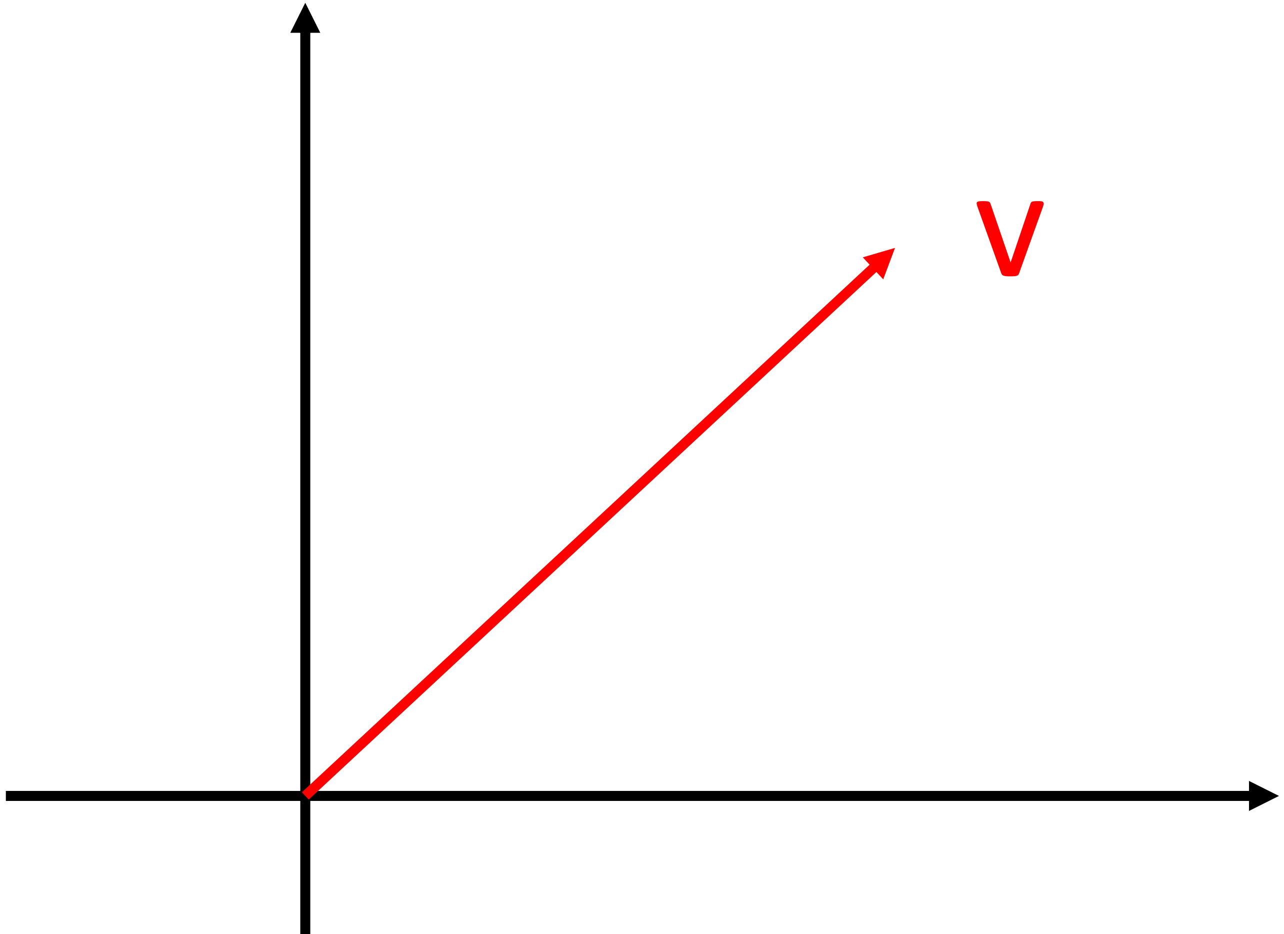
If that makes sense to you, then you know the language of transformations in graphics

What about vectors?

We don’t want to be able to translate a vector

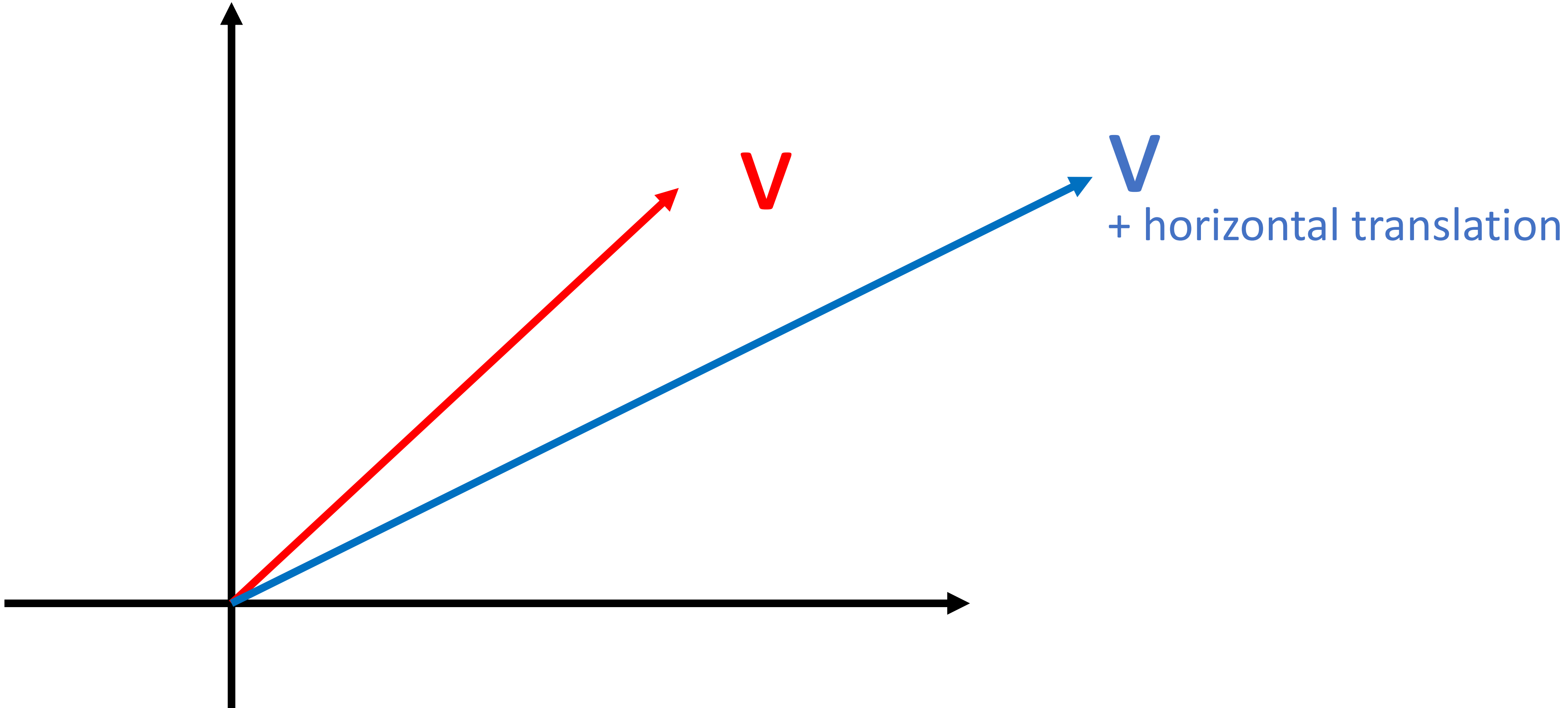
What about vectors?

We don't want to be able to translate a vector



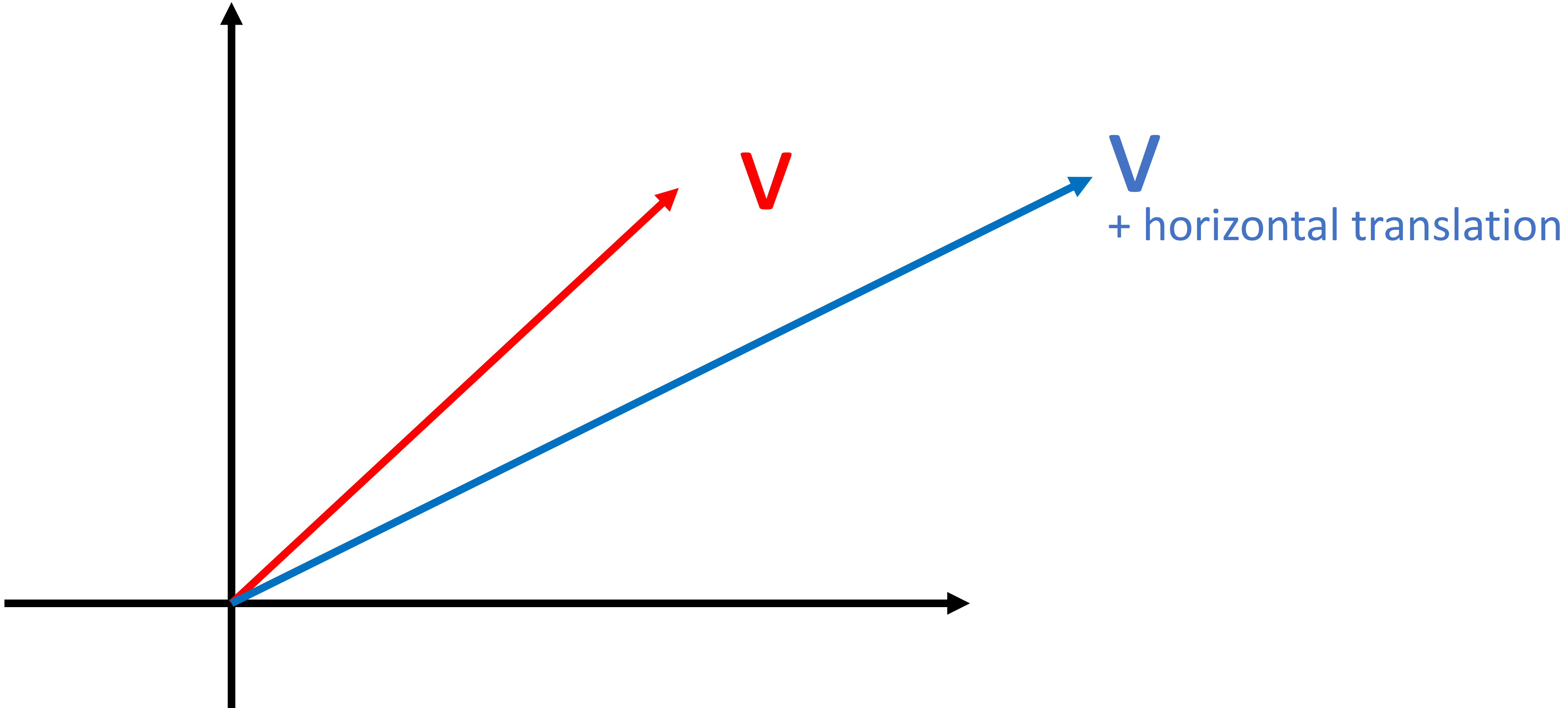
What about vectors?

We don't want to be able to translate a vector



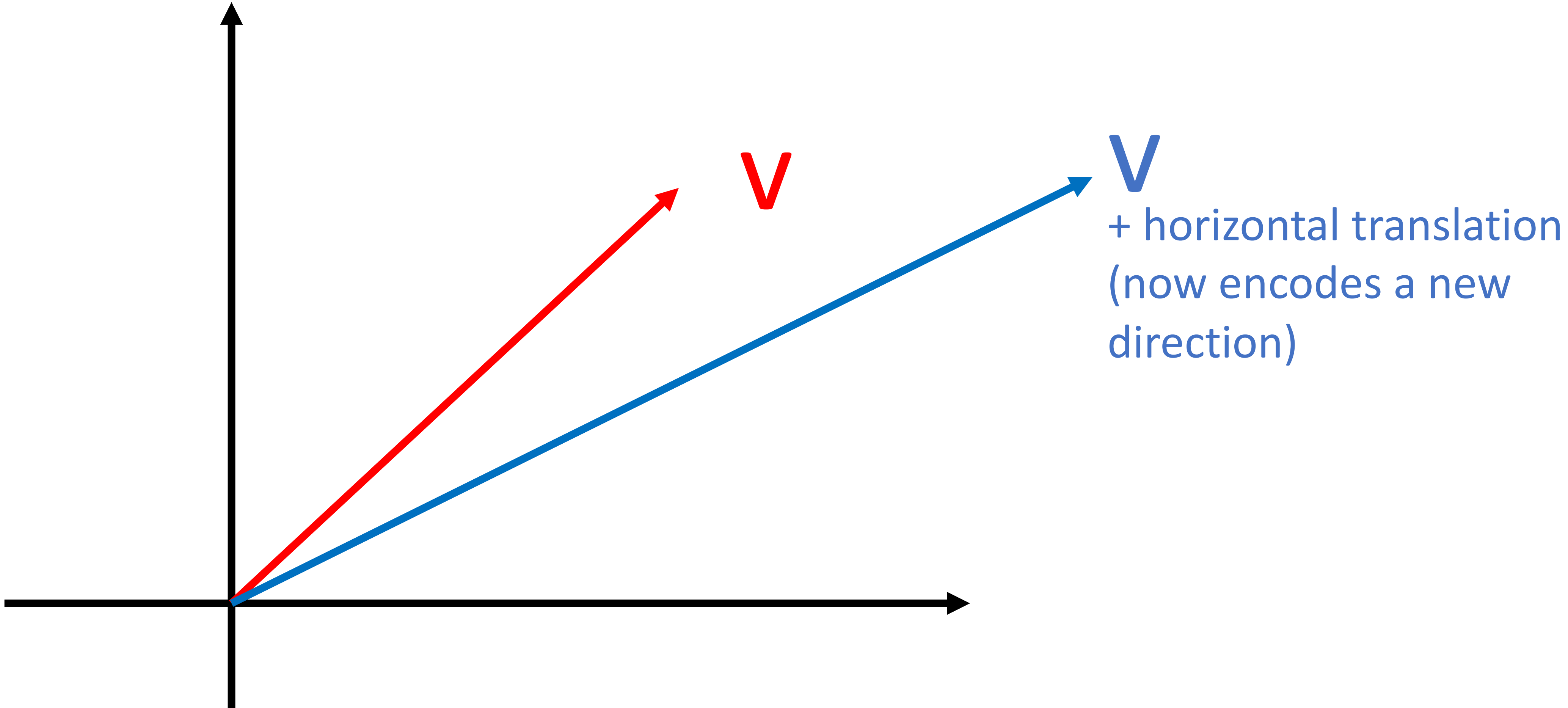
What about vectors?

We don't want to be able to translate a vector



What about vectors?

We don't want to be able to translate a vector



What about vectors?

$$\begin{bmatrix} x \\ y \end{bmatrix} \xrightarrow[\text{homogeneous coordinates}]{\text{Considered as a } \underline{\textbf{\textit{point}}} \text{ in 3D}} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

What about vectors?

$$\begin{bmatrix} x \\ y \end{bmatrix} \xrightarrow[\text{homogeneous coordinates}]{\text{Considered as a } \underline{\textbf{\textit{vector}}} \text{ in 3D}} \begin{bmatrix} x \\ y \\ 0 \end{bmatrix}$$

Why 0 for the homogeneous coordinate?

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

Example:

$\begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix}$ is a location and $\begin{bmatrix} 3 \\ 2 \\ 0 \end{bmatrix}$ is a displacement or direction.

Speaking of vectors, do normals get transformed the same way an object does?

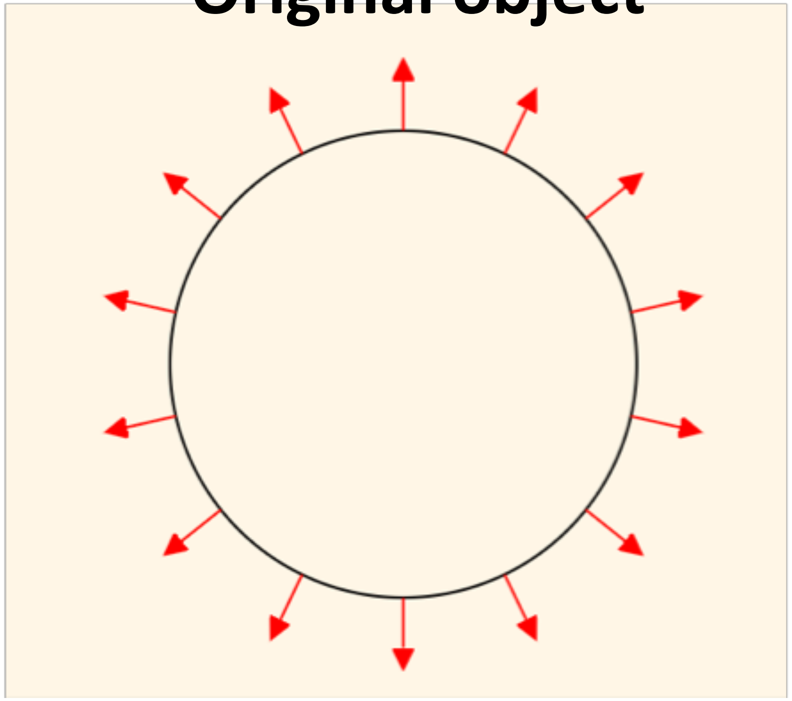
Speaking of vectors, do normals get transformed the same way an object does?

No, thank you for asking.

Speaking of vectors, do normals get transformed the same way an object does?

No, thank you for asking.

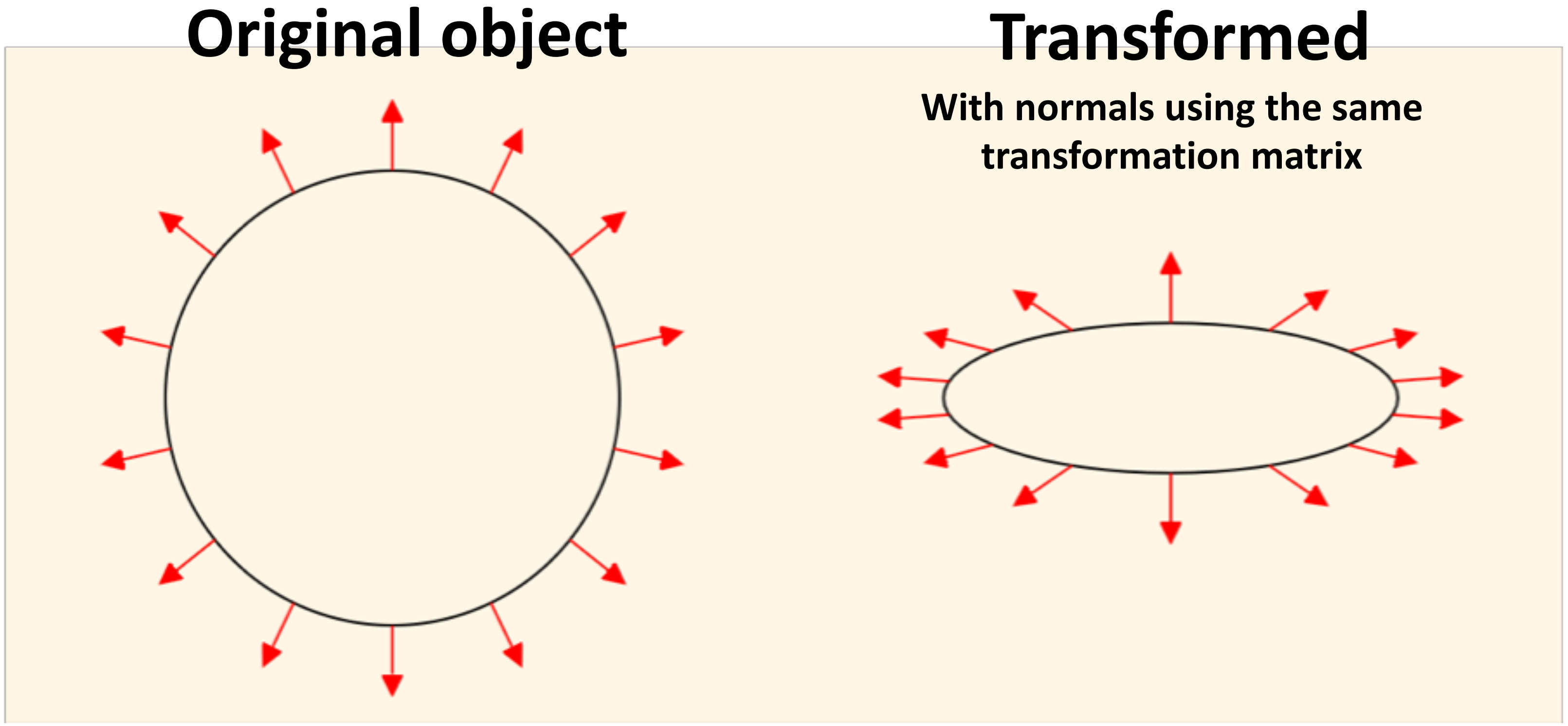
Original object



https://paroj.github.io/gltut/Illumination/Tut09%20Normal%20Transformation.html

Speaking of vectors, do normals get transformed the same way an object does?

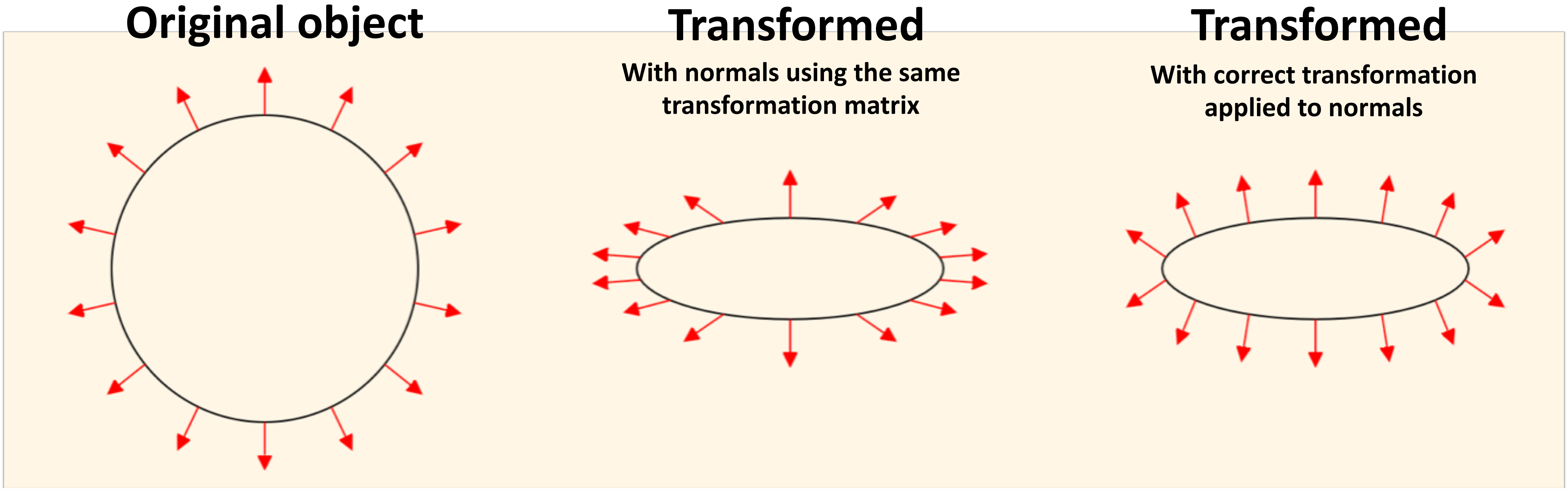
No, thank you for asking.



https://paroj.github.io/gltut/Illumination/Tut09%20Normal%20Transformation.html

Speaking of vectors, do normals get transformed the same way an object does?

No, thank you for asking.



https://paroj.github.io/gltut/Illumination/Tut09%20Normal%20Transformation.html

What’s the right way to transform a normal vector?

- Let’s say we’re transforming an object by an affine transformation M

What's the right way to transform a normal vector?

- Let's say we're transforming an object by an affine transformation M
- Given a tangent vector t on the object, it becomes $t' = Mt$

What’s the right way to transform a normal vector?

- Let’s say we’re transforming an object by an affine transformation M
- Given a tangent vector t on the object, it becomes $t' = Mt$

- Now say we had the normal vector to t called n

What's the right way to transform a normal vector?

- Let's say we're transforming an object by an affine transformation M
- Given a tangent vector t on the object, it becomes $t' = Mt$

- Now say we had the normal vector to t called n
- Since they're orthogonal, $n \cdot t = 0$ or in matrix terms, $n^T t = 0$

What’s the right way to transform a normal vector?

- Let’s say we’re transforming an object by an affine transformation M
- Given a tangent vector t on the object, it becomes $t' = Mt$

- Now say we had the normal vector to t called n
- Since they’re orthogonal, $n \cdot t = 0$ or in matrix terms, $n^T t = 0$

- We want to find a transformation X so that $n' = Xn$ the transformed normal vector is also orthogonal to t', ie $n'^T t' = 0$

What's the right way to transform a normal vector?

$$n'^T t' = 0$$

What’s the right way to transform a normal vector?

$$n'^T t' = 0$$

$$(\mathbf{X}n)^T (Mt) = 0$$

What’s the right way to transform a normal vector?

$$n'^T t' = 0$$

$$(\mathbf{X}n)^T (Mt) = 0$$

$$n^T \mathbf{X}^T Mt = 0$$

What’s the right way to transform a normal vector?

We want $n^T X^T M t = 0$

What's the right way to transform a normal vector?

We want $n^T X^T M t = 0$

So, if $X = (M^{-1})^T$ then,

What’s the right way to transform a normal vector?

We want $n^T X^T M t = 0$

So, if $X = (M^{-1})^T$ then,

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

What’s the right way to transform a normal vector?

Answer: If an object is transformed by M
Then the normal vectors are transformed by $(M^{-1})^T$

Affine transformations in 3D are exactly what you think

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} \xrightarrow[\text{homogeneous coordinates}]{\text{Considered as a \underline{\textbf{\textit{point}}} in 4D}} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Affine transformations in 3D are exactly what you think

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} \xrightarrow[\text{homogeneous coordinates}]{\text{Considered as a } \underline{\textbf{\textit{vector}}} \text{ in 4D}} \begin{bmatrix} x \\ y \\ z \\ 0 \end{bmatrix}$$

Affine transformations in 3D are exactly what you think

$$\begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Translation matrix

Affine transformations in 3D are exactly what you think

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & t_x \\ a_{21} & a_{22} & a_{23} & t_y \\ a_{31} & a_{32} & a_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

General form of

$$Ax + t$$

Example: 3D uniform scale, followed by translation

$$\begin{bmatrix} 2 & 0 & 0 & 1.5 \\ 0 & 2 & 0 & -2 \\ 0 & 0 & 2 & 0.5 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

General form of

$$Ax + t$$

Understanding homogeneous coordinates:

Consider a 3D shear based on the z-coordinate

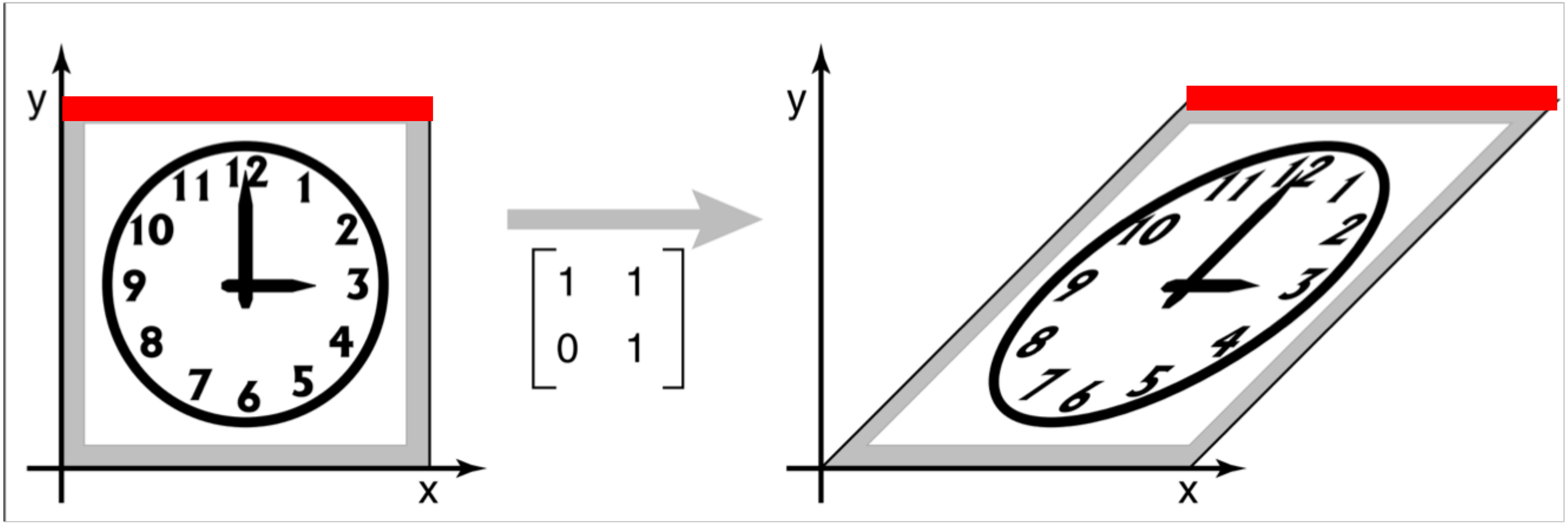
$$\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x + zt_x \\ y + zt_y \\ z \end{bmatrix}$$

Understanding homogeneous coordinates:

If z = 1 we get the same transformation as a 2D translation in homogeneous coordinates

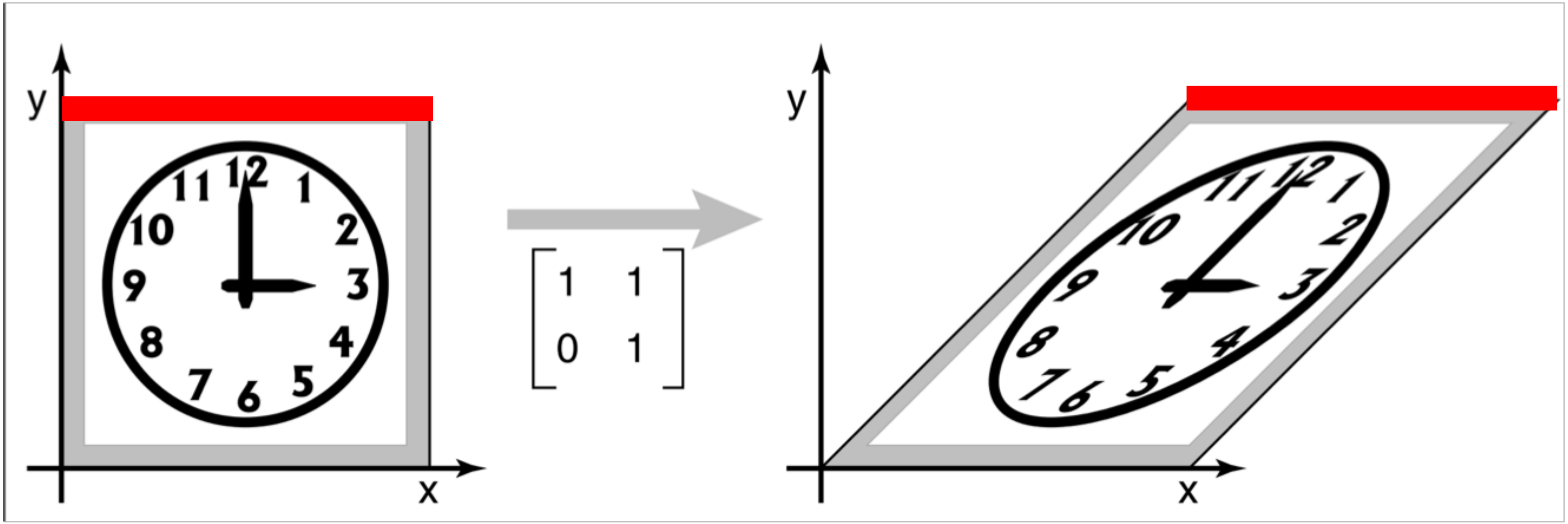
$$\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x + t_x \\ y + t_y \\ 1 \end{bmatrix}$$

Remember my special note about shears?



These are always the same length

Remember my special note about shears?



You can think of this as a 1D translation of the red line in homogeneous coordinates