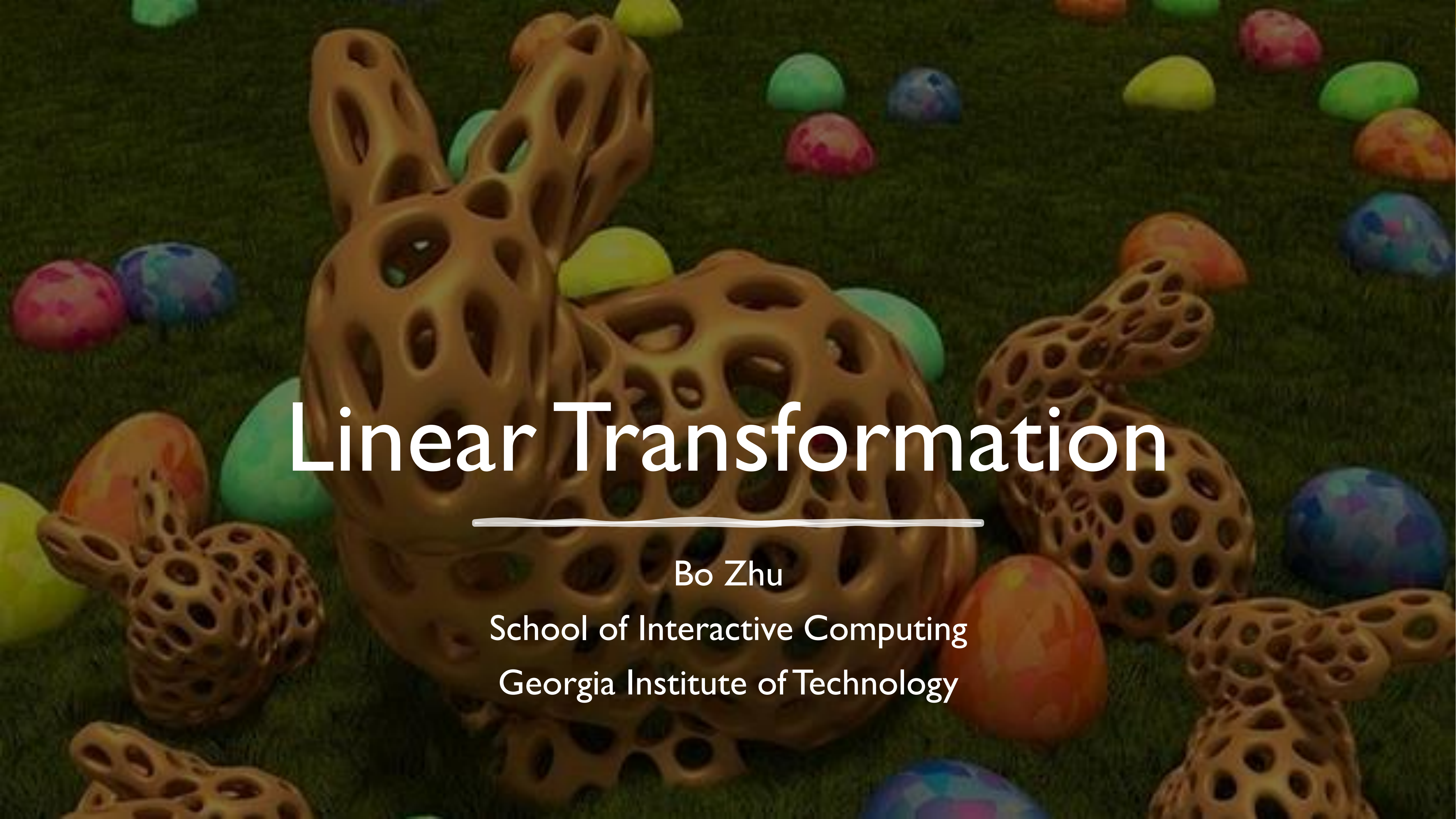
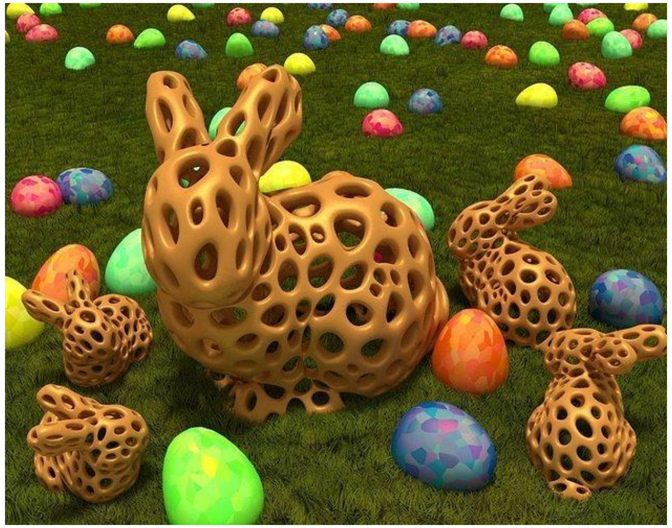


Linear Transformation

Bo Zhu
School of Interactive Computing
Georgia Institute of Technology

Idea: Store Geometry for Once, and Transform it for Many Times



This can be implemented by a simple matrix-vector multiplication

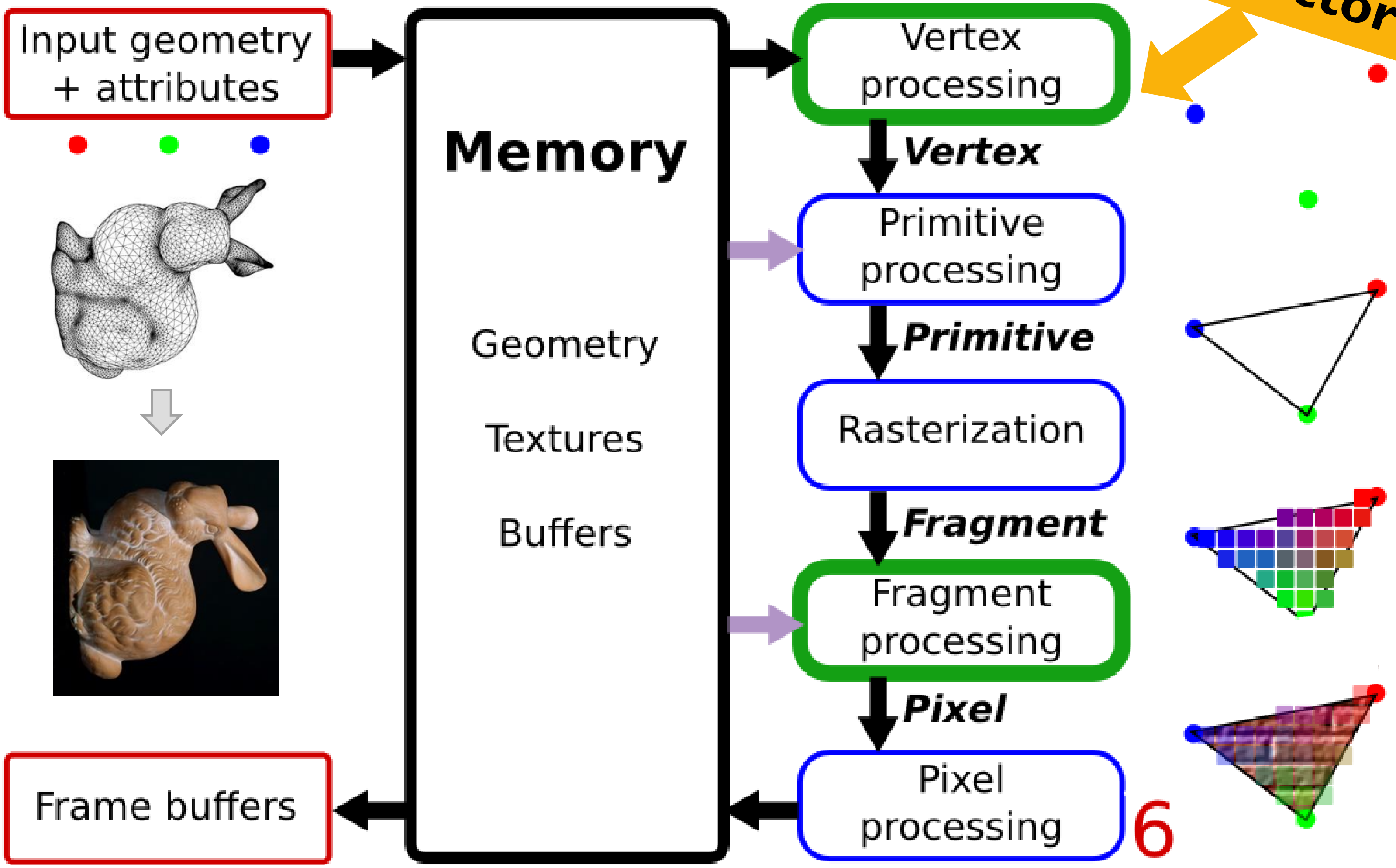
```
//// Simple pseudocode:

uniform mat4 M;
in vec4 vtx;
out vec4 pos;

void main()
{
        pos=M*vtx;
}
```

Recap: Modern Graphics Pipeline

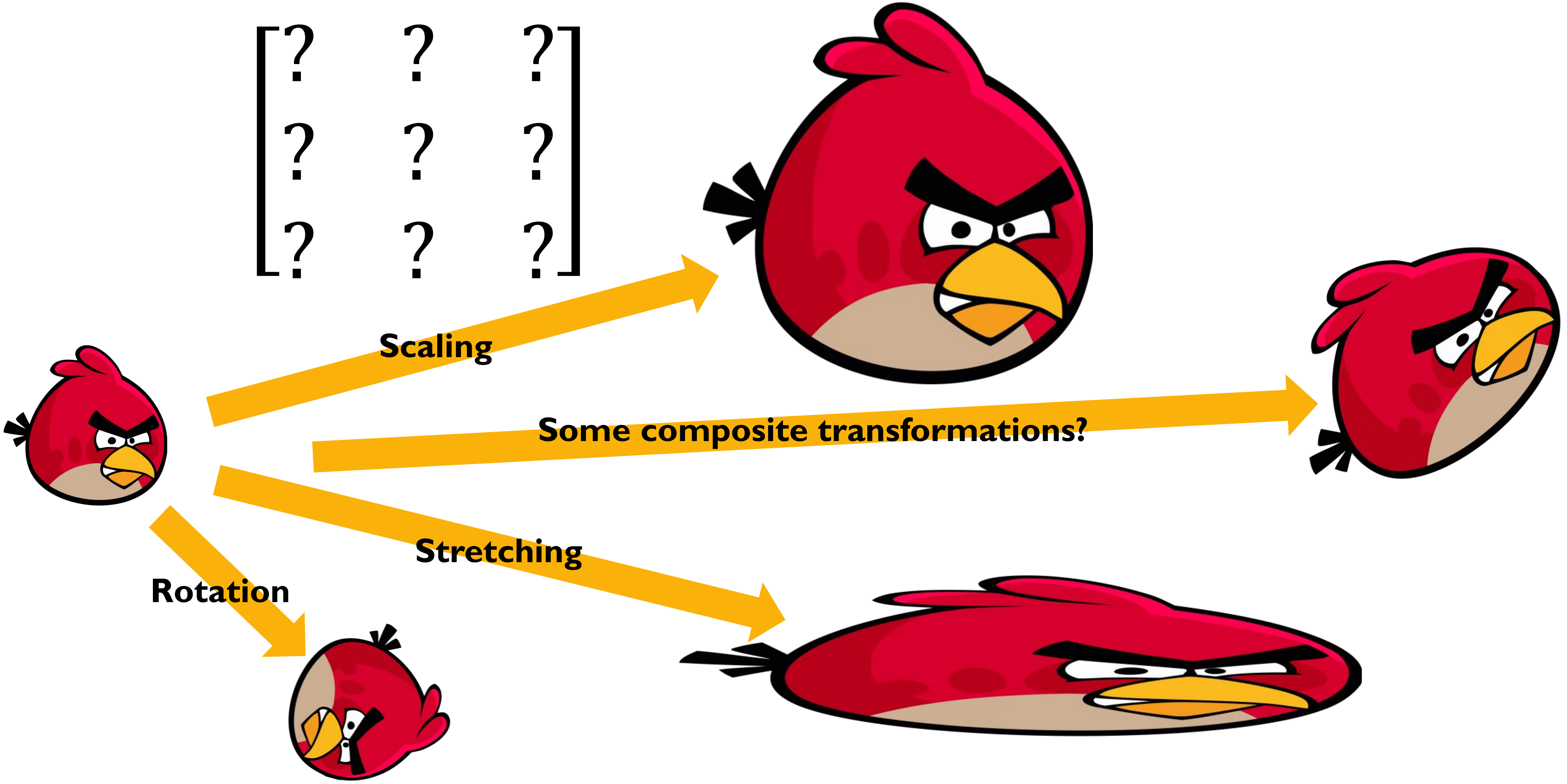
Geometric Transformation
Happens in the vertex shader
as matrix-vector multiplication



IMPORTANT!

How do we decide values of the matrix elements?

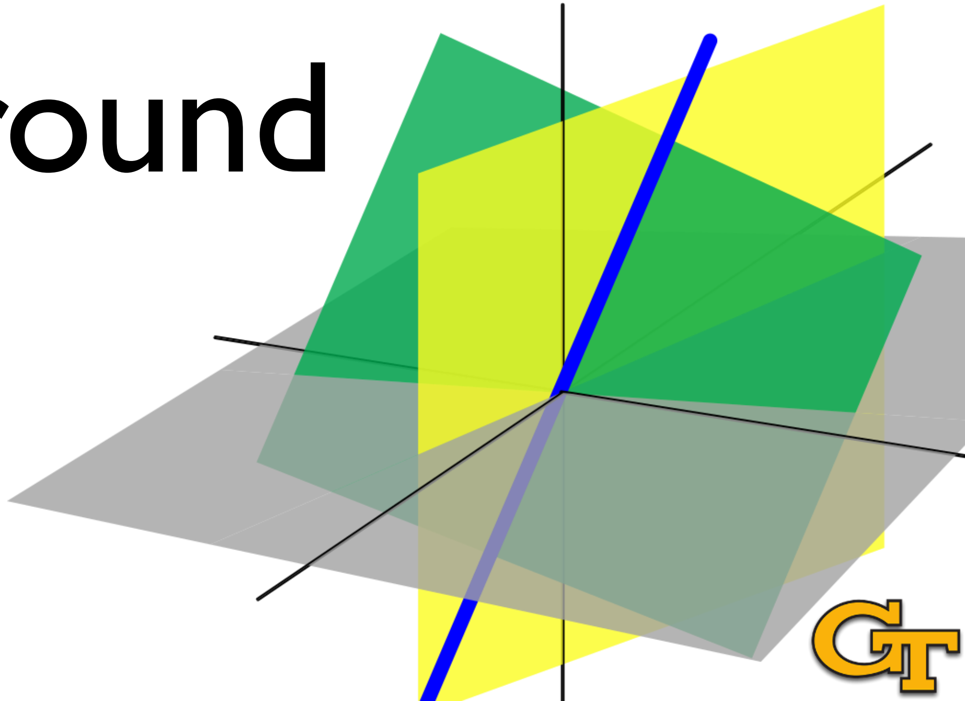
$$\begin{bmatrix} ? & ? & ? \\ ? & ? & ? \\ ? & ? & ? \end{bmatrix}$$



Study Plan

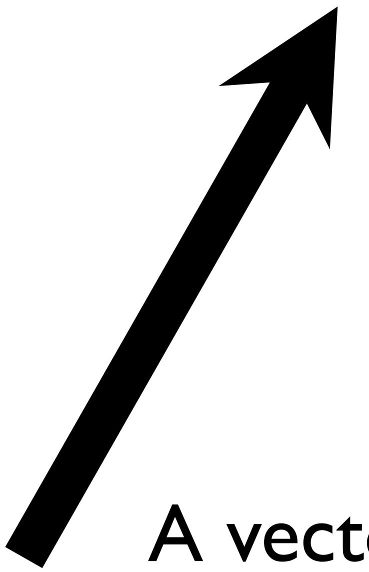
- Linear algebra background
- Linear mapping
- Every linear mapping is a matrix multiplication
- 2D transformations
- Translation, Affine Transform, and homogeneous coordinates
- Composite transformations
- 3D transformations

Linear Algebra Background



Vector

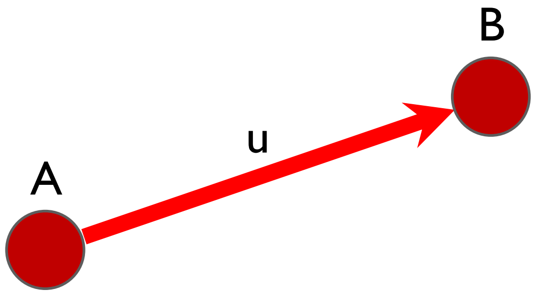
- What is a **vector**?
- Intuitively, a vector is a little arrow with a direction and a magnitude
- Vectors with the same length and magnitude are the same vector
- In a Cartersian coordinate system, a vector is denoted by its coordinates, e.g., [1.2, 3.4], [0.0,1.0,2.3]

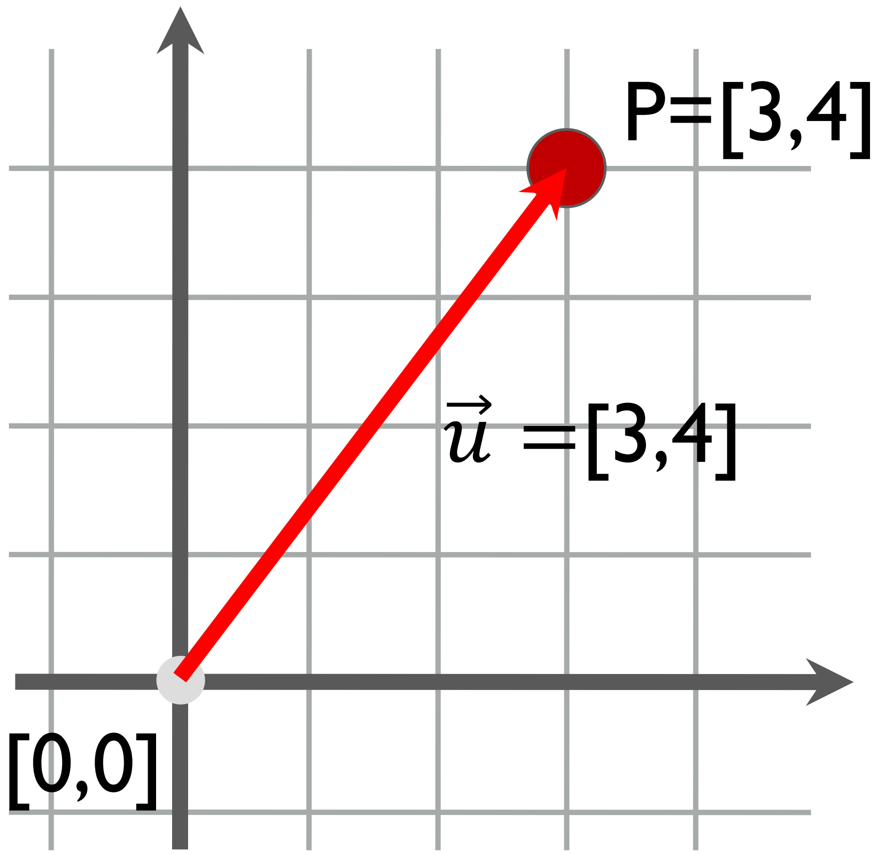


A vector

Vector v.s. Point

- Let's think of a point as **a vector starting from the origin** to its position
- A vector represents the displacement between two points
  - E.g., we have two points A [3,4] and B [1,1], then the vector u=A-B=[3,4]-[1,1]=[2,3] indicates that we will translate by [2,3] to move from A to B





Matrix-vector Multiplication

- **Algebraic view:**
  - Calculate each element by multiplying a matrix row with a vector column
- **Geometric view:**
  - Understand it as a linear combination of columns with the vector elements as weights

Algebraic View

$$\begin{bmatrix} 1 & -1 & 2 \\ 2 & 0 & 1 \\ 1 & 2 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 1\times 1 + (-1)\times 2 + 2\times 3 \\ 2\times 1 + 0\times 2 + 1\times 3 \\ 1\times 1 + 2\times 2 + 1\times 3 \end{bmatrix} = \begin{bmatrix} 5 \\ 5 \\ 8 \end{bmatrix}$$

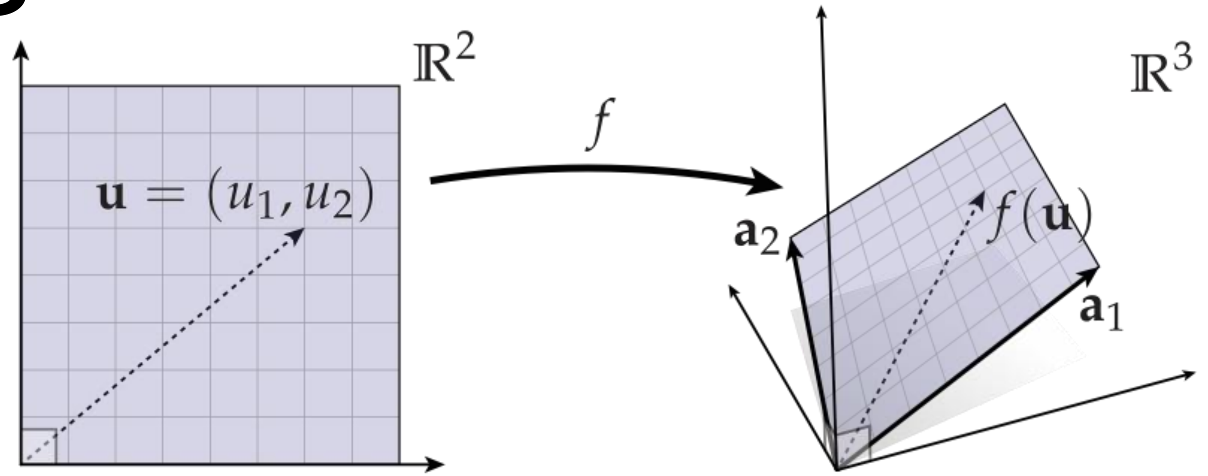
Geometric View

$$\begin{bmatrix} \boxed{\begin{matrix} 1 \\ 2 \\ 1 \end{matrix}} & \boxed{\begin{matrix} -1 \\ 0 \\ 2 \end{matrix}} & \boxed{\begin{matrix} 2 \\ 1 \\ 1 \end{matrix}} \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = 1 \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} + 2 \begin{bmatrix} -1 \\ 0 \\ 2 \end{bmatrix} + 3 \begin{bmatrix} 2 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 5 \\ 5 \\ 8 \end{bmatrix}$$

Algebraic Laws for Matrix Multiplication

- Matrix multiplication is associative, distributive, but NOT commutative!
- Associativity: $ABC = (AB)C = A(BC)$
  - How do you interpret a chain of matrix multiplication: $A_1 A_2 A_3 A_4 u$? Which matrix exerts on u first?
  - Usually we see it as $(A_1(A_2(A_3(A_4 u))))$
  - Sometimes we also see it as $(A_1 A_2 A_3 A_4)u$
- Distributivity: $A(B + C) = AB + AC$
- **NO commutativity**: $AB \neq BA$

Linear Mapping



Definition

- A map f is linear if it maps vectors to vectors, and if for all vectors $\mathbf{u}$,$\mathbf{v}$ and scalars a,b we have:

$$f(\mathbf{u}+\mathbf{v}) = f(\mathbf{u}) + f(\mathbf{v})$$
$$f(a\mathbf{u}) = af(\mathbf{u})$$

$$\Rightarrow \quad f(a\mathbf{u}+b\mathbf{v}) = af(\mathbf{u}) + bf(\mathbf{v})$$

- In other words: doesn't matter if we add the vectors and then apply the map, or apply the map and then add the vectors

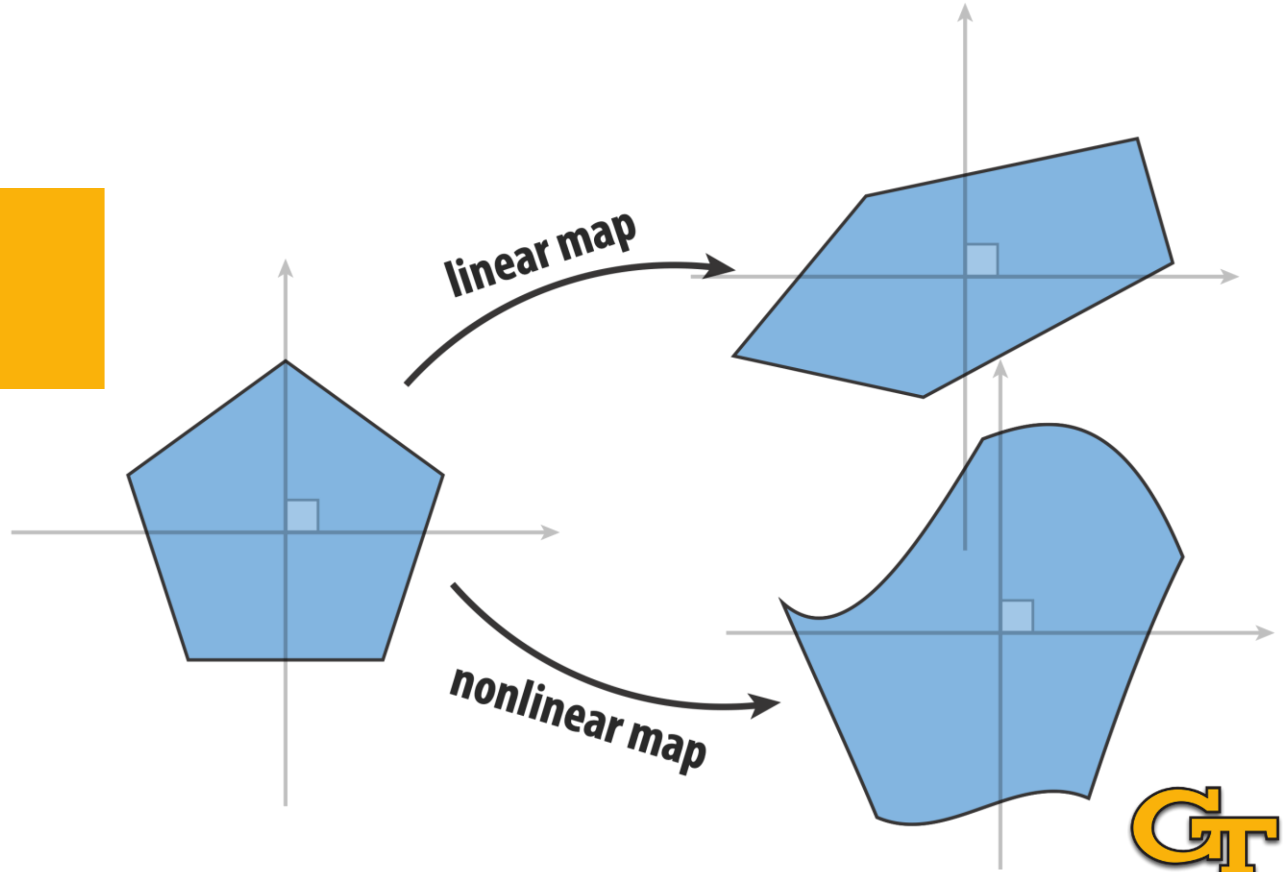
Check if a function is linear or not

- To prove a function f is linear, show $f(\alpha\vec{x} + \beta\vec{y}) = \alpha f(\vec{x}) + \beta f(\vec{y})$ satisfies for any $\alpha, \beta, \vec{x}, \vec{y}$.
- To prove a function f is not linear, give ONE example of $\alpha, \beta, \vec{x}, \vec{y}$ showing $f(\alpha\vec{x} + \beta\vec{y}) \neq \alpha f(\vec{x}) + \beta f(\vec{y})$
  - A quick trick: **showing $\boldsymbol{f(\vec{0})! = \vec{0}}$**
  - E.g.,
    - $f(x) = 2x$ is a linear map because $f(ax + by) = 2(ax + by) = a2x + a2y = a(x) + bf(y)$
    - $f(x) = 2x + 1$ is not a linear map, because $f(0)! = 0$

Geometric Intuition

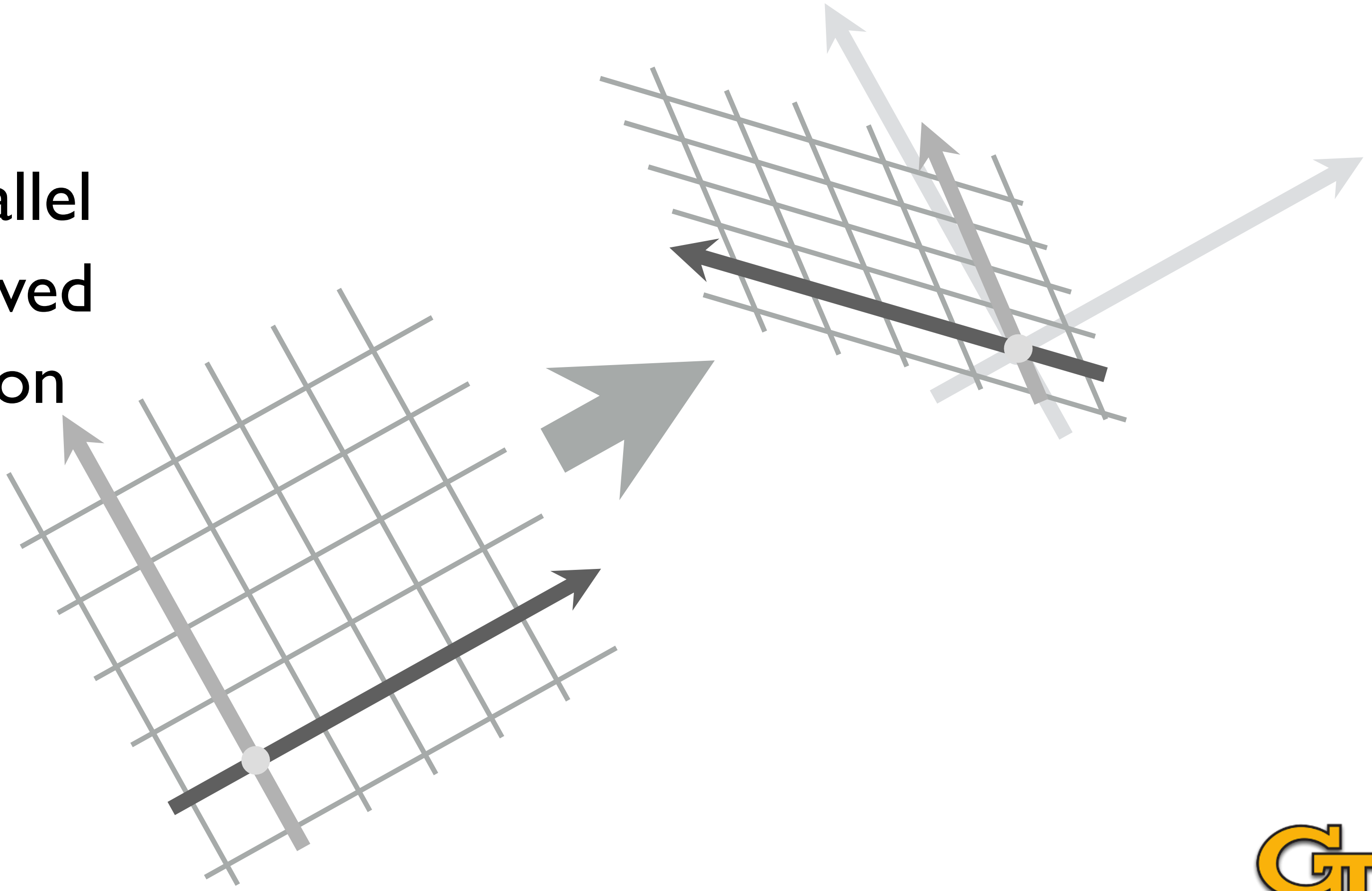
- We can think about the definition of linear map ***visually***.
- Key idea:

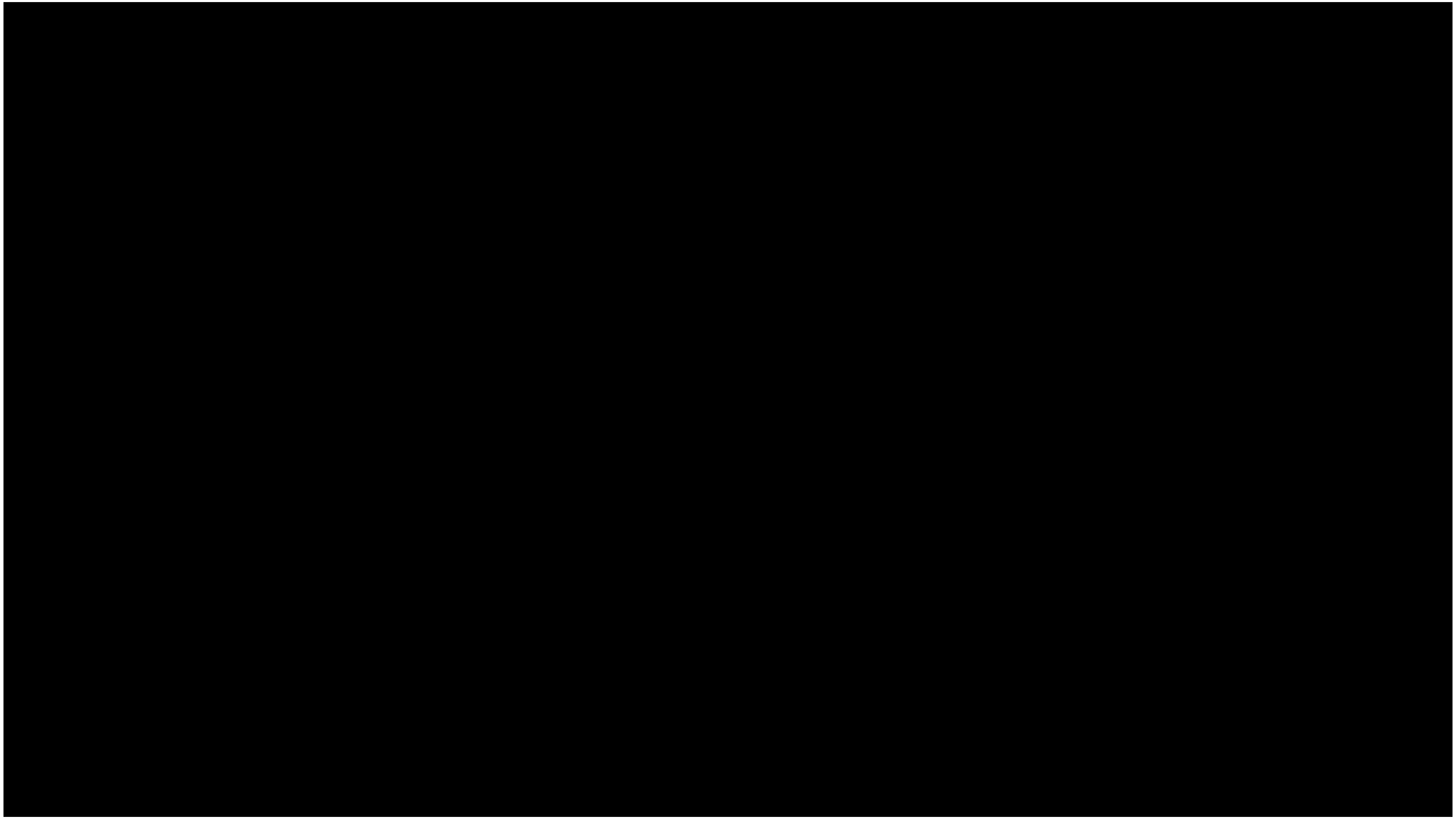
Linear maps take lines to lines,
while keeping origin fixed.



Properties

- **A Linear Map:**
  - **Maps origin to origin**
  - Maps lines to lines
  - Parallel lines remain parallel
  - Length ratios are preserved
  - Closed under composition





3Blue1Brown: Linear Transformation: https://www.youtube.com/watch?v=kYB8IZa5AuE&t=177s

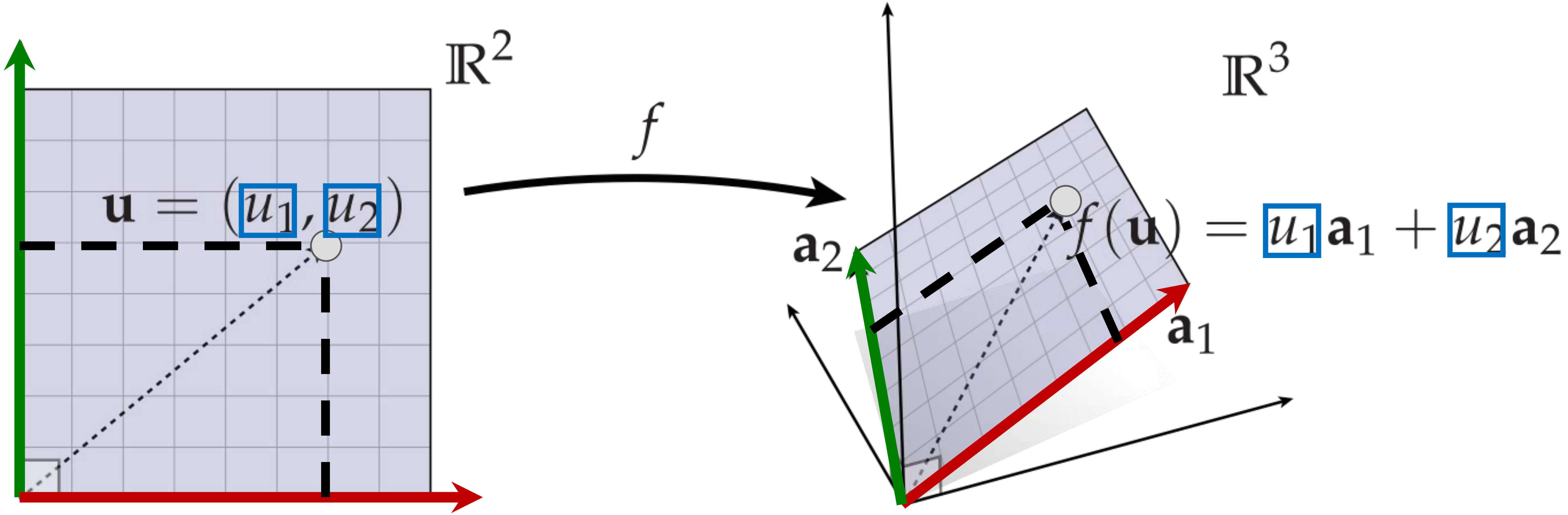
Understand a Linear Map as a Combination of Basis Vectors

- For a linear map, we can give an even more explicit definition.
- A map is linear if it can be expressed as:

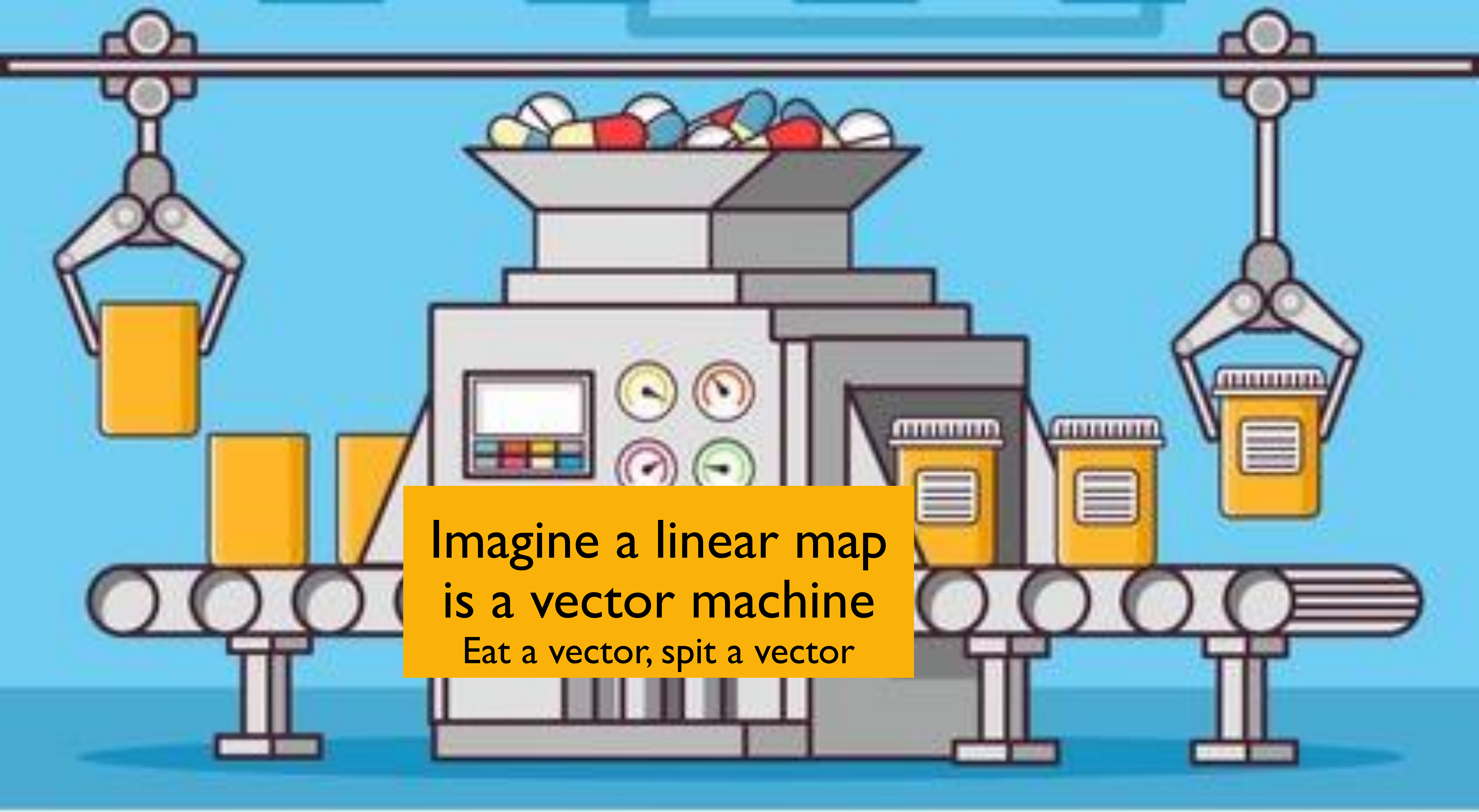
$$f(u_1, \ldots, u_m) = \sum_{i=1}^{m} u_i \mathbf{a}_i$$

- In other words, if it is a linear combination of a fixed set of vectors $\mathbf{a}_i$

A Visual Example: a Linear Map $R^2 \to R^3$



How do we represent a linear map?



Imagine a linear map
is a vector machine

Eat a vector, spit a vector

Every linear mapping is a matrix multiplication!

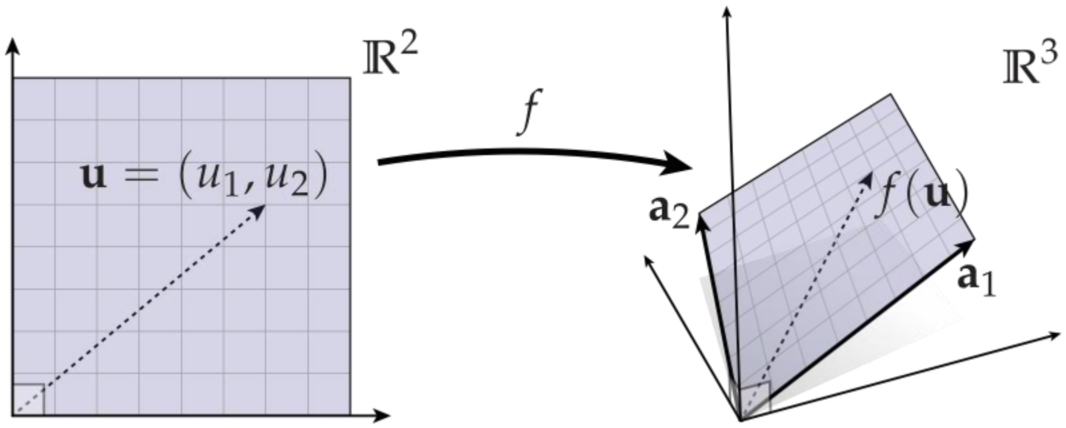
How to write the matrix of a linear mapping?

Representing Linear Maps via Matrices

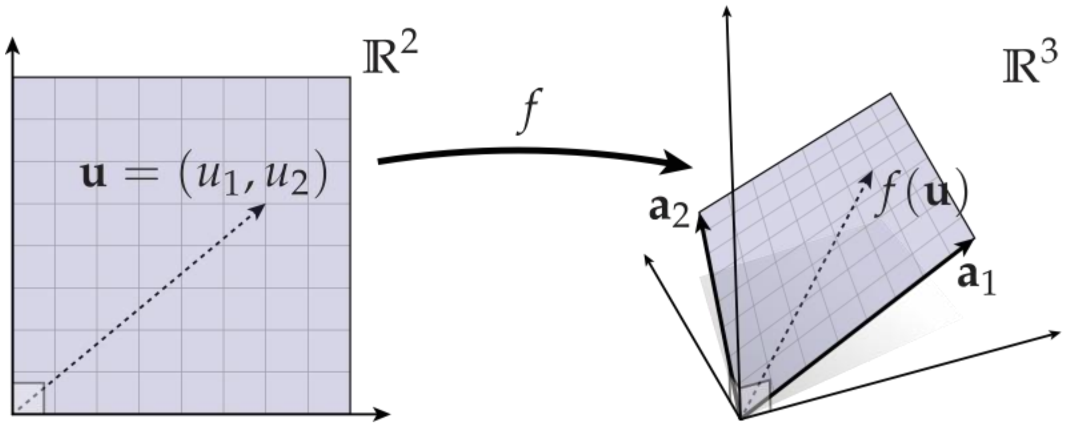
- Suppose I have a linear map

$$f(\mathbf{u}) = u_1\mathbf{a}_1 + u_2\mathbf{a}_2$$

- How do I encode as a matrix?



$$f(\mathbf{u}) = u_1\mathbf{a}_1 + u_2\mathbf{a}_2$$



- How do I encode as a matrix?
- Easy: "a" vectors become matrix columns:

$$\mathbf{A} := \begin{bmatrix} a_{1,x} & a_{2,x} \\ a_{1,y} & a_{2,y} \\ a_{1,z} & a_{2,z} \end{bmatrix}$$

- Now, matrix-vector multiply recovers original map:

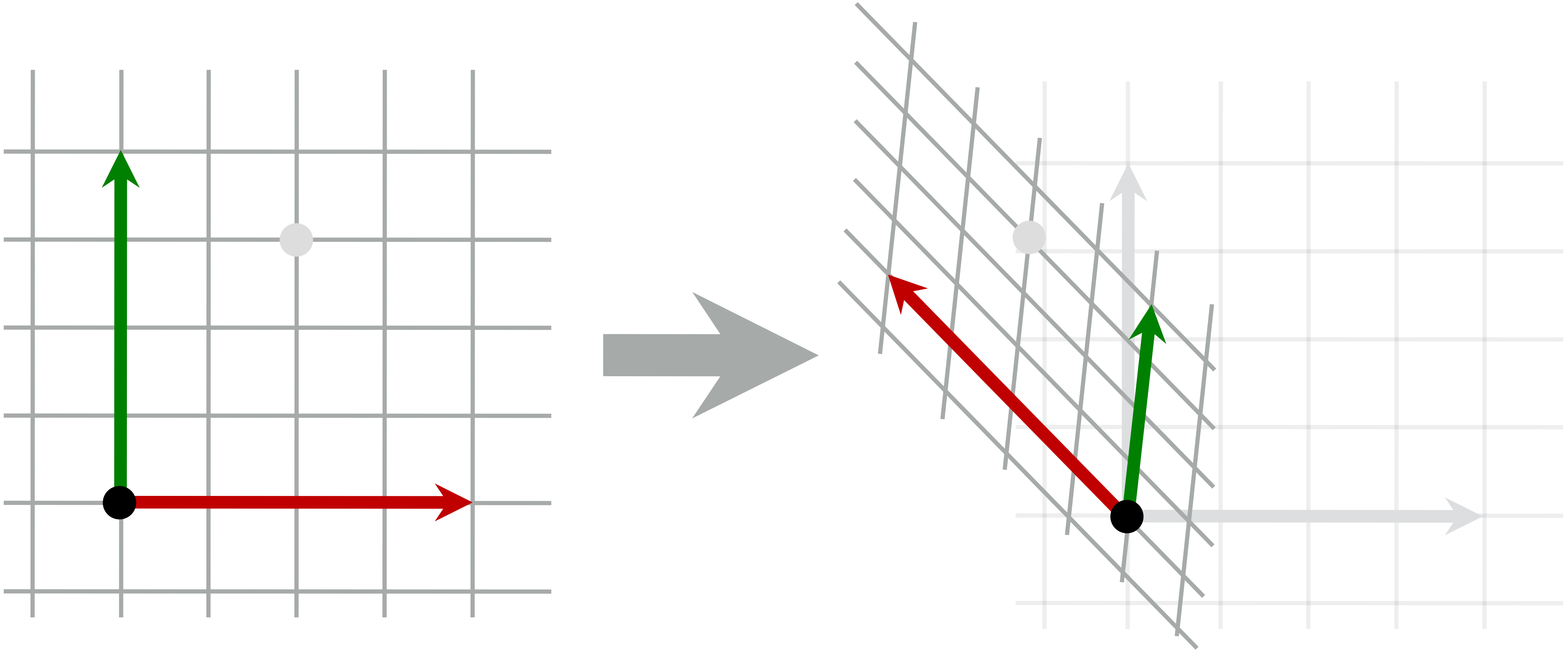
$$f(\mathbf{u}) = A\mathbf{u} = \begin{bmatrix} a_{1,x} & a_{2,x} \\ a_{1,y} & a_{2,y} \\ a_{1,z} & a_{2,z} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} a_{1,x}u_1 + a_{2,x}u_2 \\ a_{1,y}u_1 + a_{2,y}u_2 \\ a_{1,z}u_1 + a_{2,z}u_2 \end{bmatrix} = u_1\mathbf{a}_1 + u_2\mathbf{a}_2$$

Matrix for a Linear Map

Where do these number come from?
What do they mean?

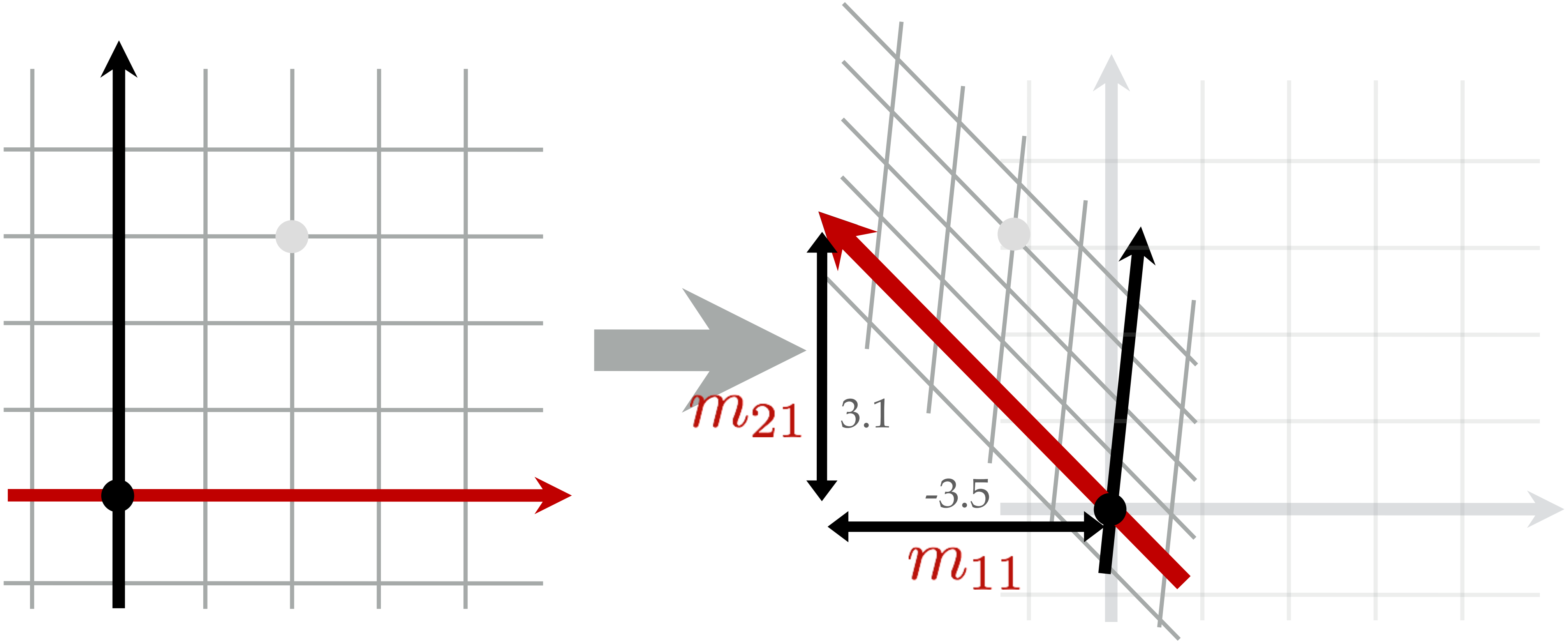
$$f(\mathbf{p}) = M\mathbf{p} = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix} \begin{bmatrix} p_x \\ p_y \end{bmatrix}$$

$$M = \begin{bmatrix} -0.7 & 0.1 \\ 0.7 & 0.6 \end{bmatrix}$$



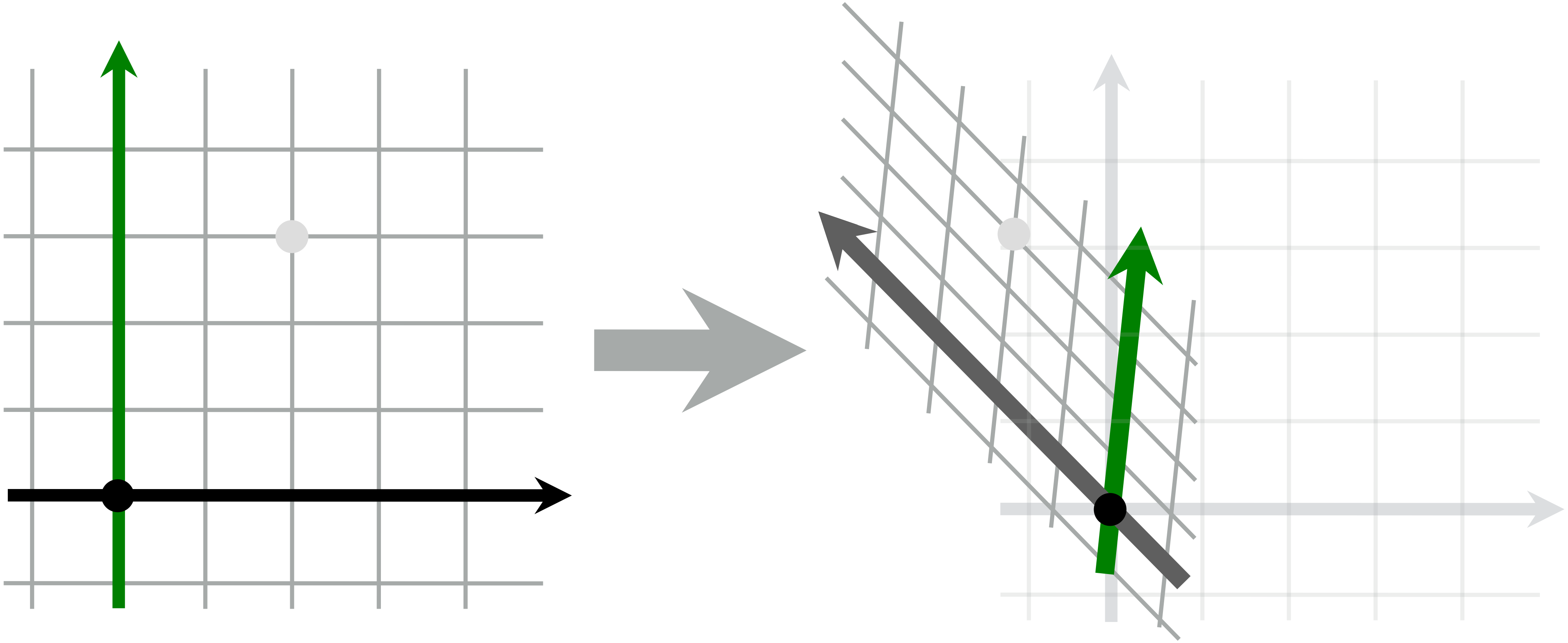
First Column: the Transformed X Axis

$$X(\mathbf{p}) = M\mathbf{p} = \begin{bmatrix} \boxed{\begin{matrix} m_{11} \\ m_{21} \end{matrix}} & \begin{matrix} m_{12} \\ m_{22} \end{matrix} \end{bmatrix} \begin{bmatrix} p_x \\ p_y \end{bmatrix} \qquad M = \begin{bmatrix} \boxed{\begin{matrix} -3.5 \\ 3.1 \end{matrix}} & \begin{matrix} 0.4 \\ 3.1 \end{matrix} \end{bmatrix}$$



Second Column: the Transformed Y Axis

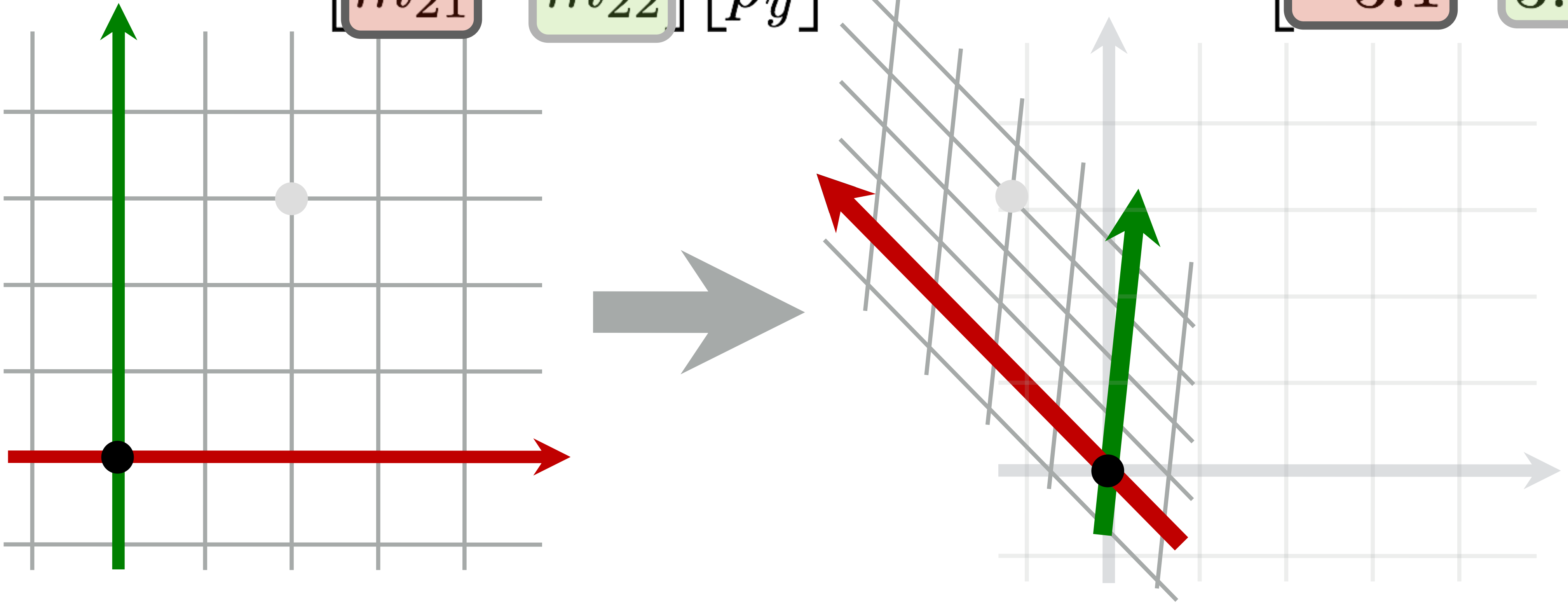
$$X(\mathbf{p}) = M\mathbf{p} = \begin{bmatrix} m_{11} & \boxed{m_{12}} \\ m_{21} & \boxed{m_{22}} \end{bmatrix} \begin{bmatrix} p_x \\ p_y \end{bmatrix} \qquad M = \begin{bmatrix} -3.5 & \boxed{0.4} \\ 3.1 & \boxed{3.1} \end{bmatrix}$$



Summary

$$X(\mathbf{p}) = M\mathbf{p} = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix} \begin{bmatrix} p_x \\ p_y \end{bmatrix}$$

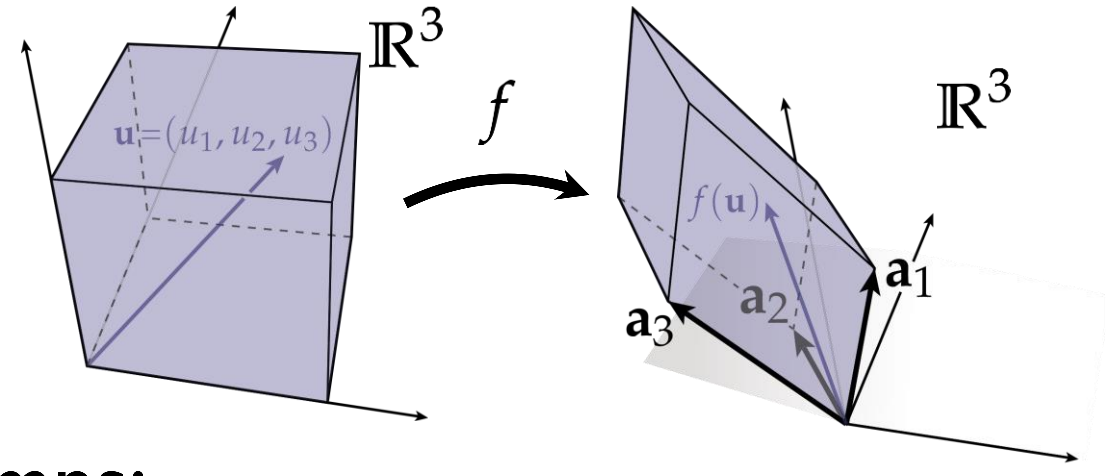
$$M = \begin{bmatrix} -3.5 & 0.4 \\ 3.1 & 3.1 \end{bmatrix}$$



The first column of M is the transformed x axis.

The second column of M is the transformed y axis.

Extend to 3D



- Encode this linear map as a matrix

$$f(\mathbf{u}) = u_1\mathbf{a}_1 + u_2\mathbf{a}_2 + u_3\mathbf{a}_3$$

- Idea: $\mathbf{a}_i$ vectors become matrix columns:

$$A := \begin{bmatrix} | & | & | \\ \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 \\ | & | & | \end{bmatrix} = \begin{bmatrix} a_{1,x} & a_{2,x} & a_{3,x} \\ a_{1,y} & a_{2,y} & a_{3,y} \\ a_{1,z} & a_{2,z} & a_{3,z} \end{bmatrix}$$

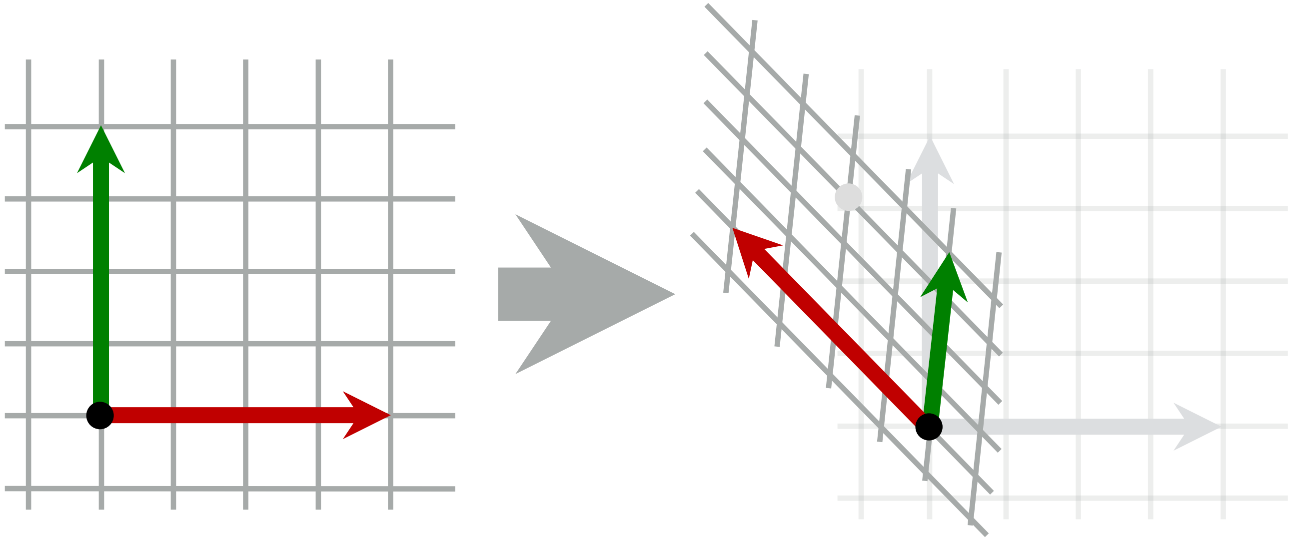
- Now, matrix-vector multiply recovers original map:

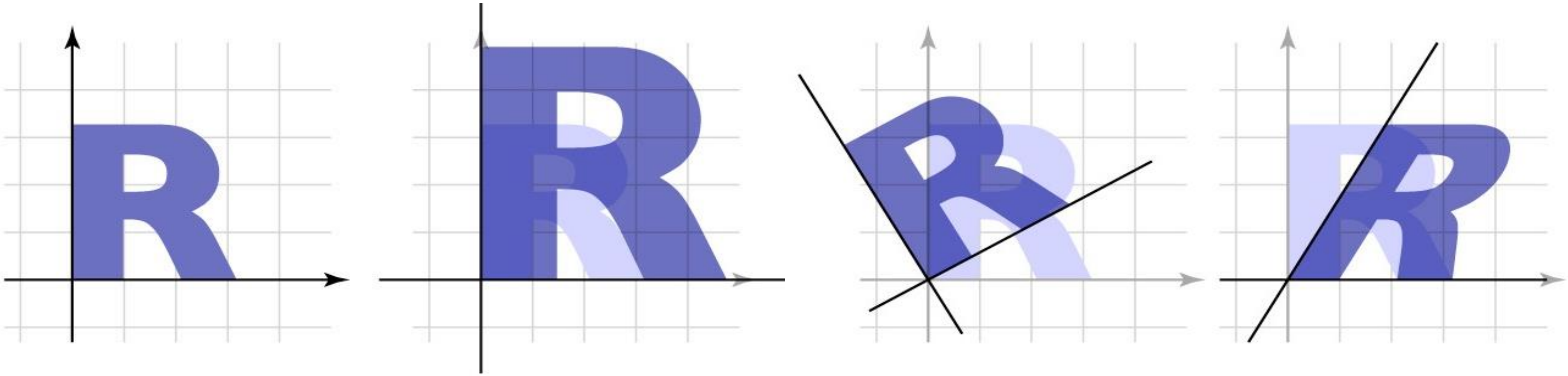
$$A\mathbf{u} = \begin{bmatrix} a_{1,x} & a_{2,x} & a_{3,x} \\ a_{1,y} & a_{2,y} & a_{3,y} \\ a_{1,z} & a_{2,z} & a_{3,z} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} a_{1,x}u_1 + a_{2,x}u_2 + a_{3,x}u_3 \\ a_{1,y}u_1 + a_{2,y}u_2 + a_{3,y}u_3 \\ a_{1,z}u_1 + a_{2,z}u_2 + a_{3,z}u_3 \end{bmatrix}$$

$$= u_1\mathbf{a}_1 + u_2\mathbf{a}_2 + u_3\mathbf{a}_3 = f(\mathbf{u})$$

Key Takeaways of Linear Transformation

- Each column in a transformation matrix is a transformed basis vector.
  - E.g., the first column of M is the transformed basis vector of $[1,0,0]^T$
- The transformed vector is the linear combination of the transformed basis vectors with the same (old) coordinates





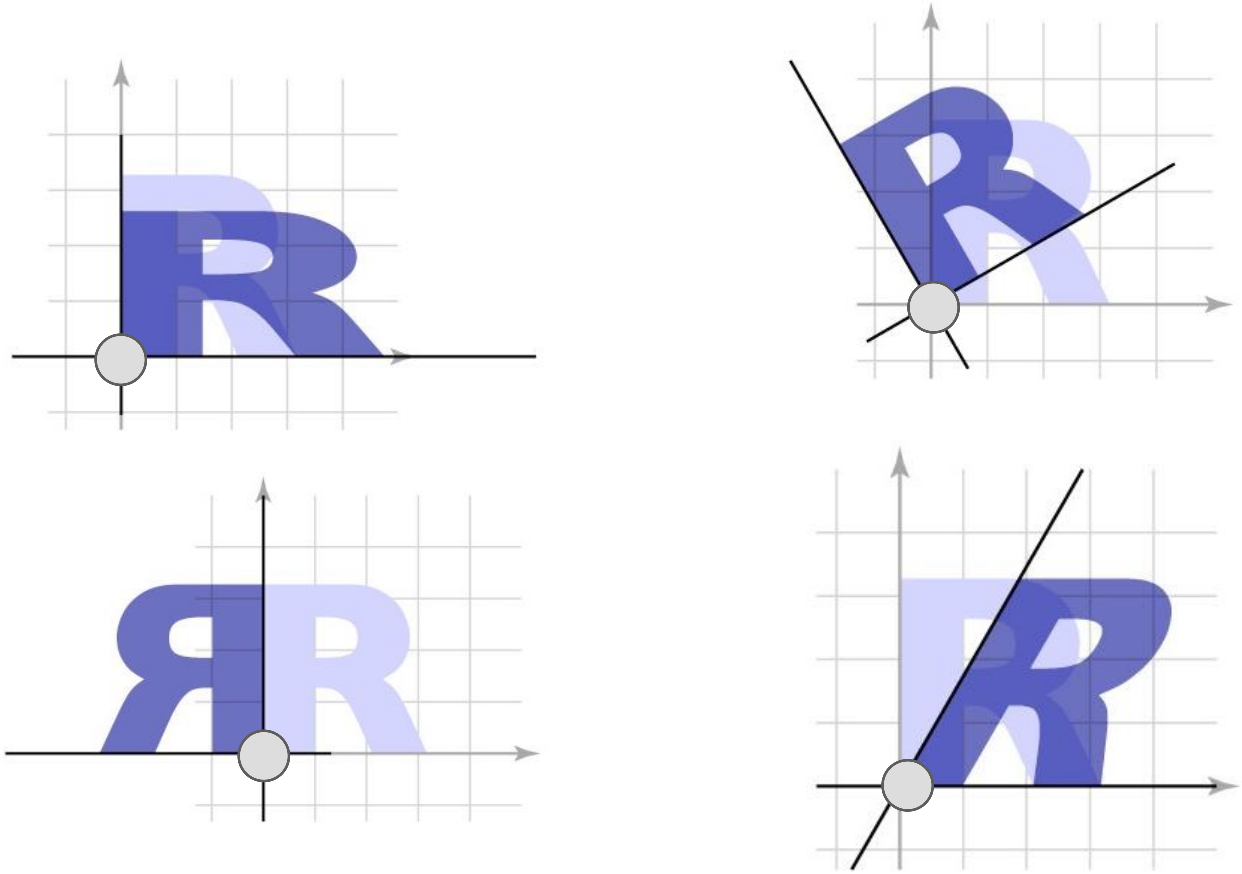
2D Transformations

Task: Understand and memorize these transformation matrices using the geometric picture of linear mapping we just learned!

Geometry of linear transforms

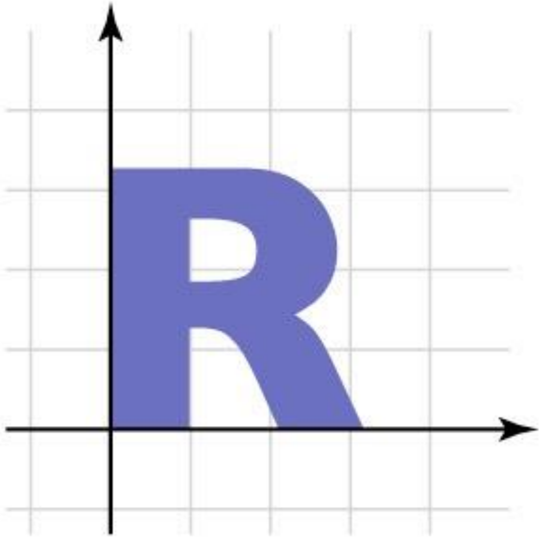
- Scaling
- Reflection
- Rotation
- Shear

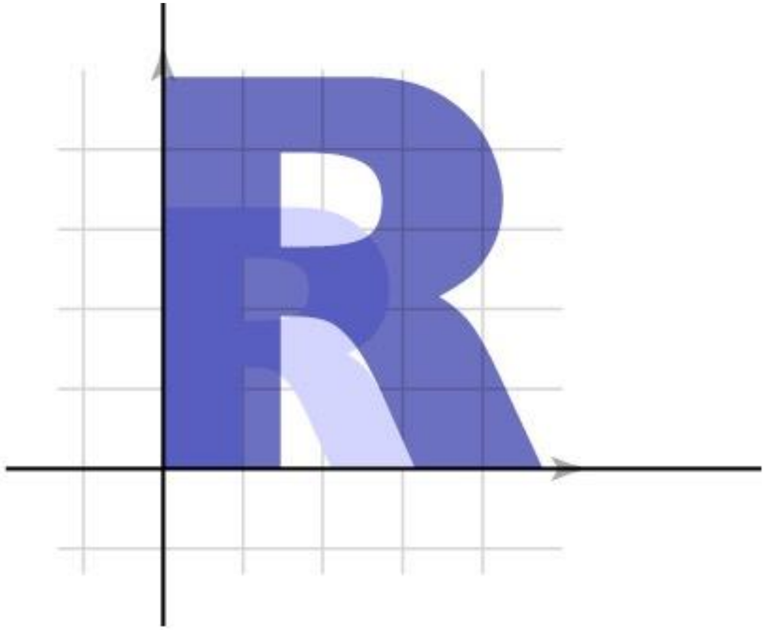
- **Origin does not change**



Uniform Scaling

$$S_s\mathbf{p} = \begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} \begin{bmatrix} p_x \\ p_y \end{bmatrix} = \begin{bmatrix} sp_x \\ sp_y \end{bmatrix}$$

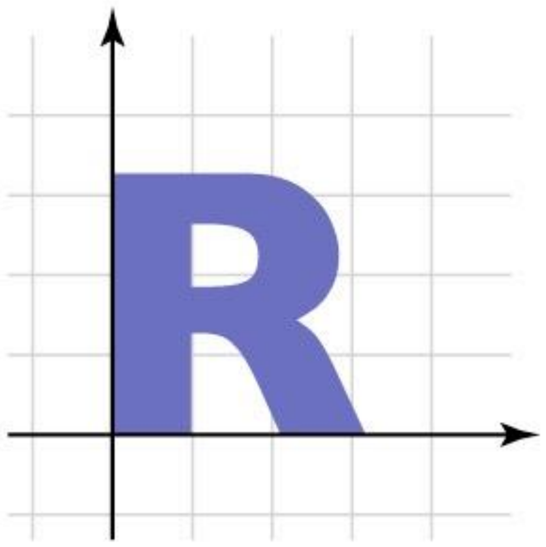


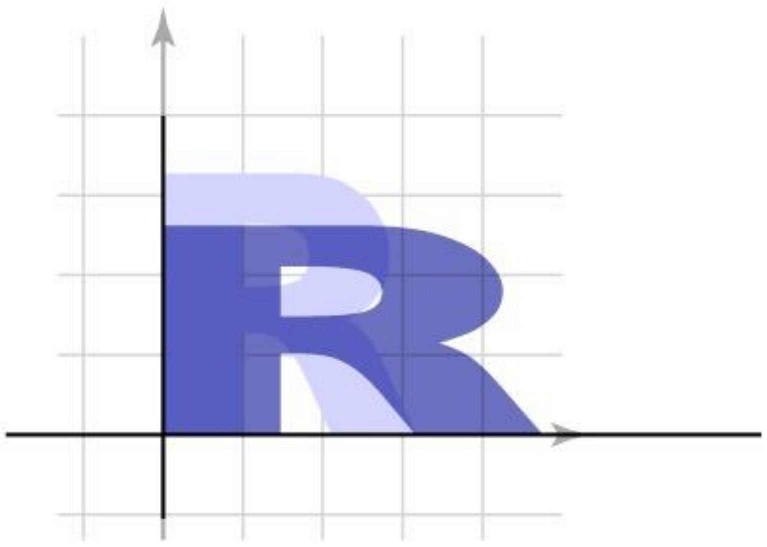


$$\begin{bmatrix} 1.5 & 0 \\ 0 & 1.5 \end{bmatrix}$$

Nonuniform Scaling

$$S_{\mathbf{s}}\mathbf{p} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} p_x \\ p_y \end{bmatrix} = \begin{bmatrix} s_x p_x \\ s_y p_y \end{bmatrix}$$

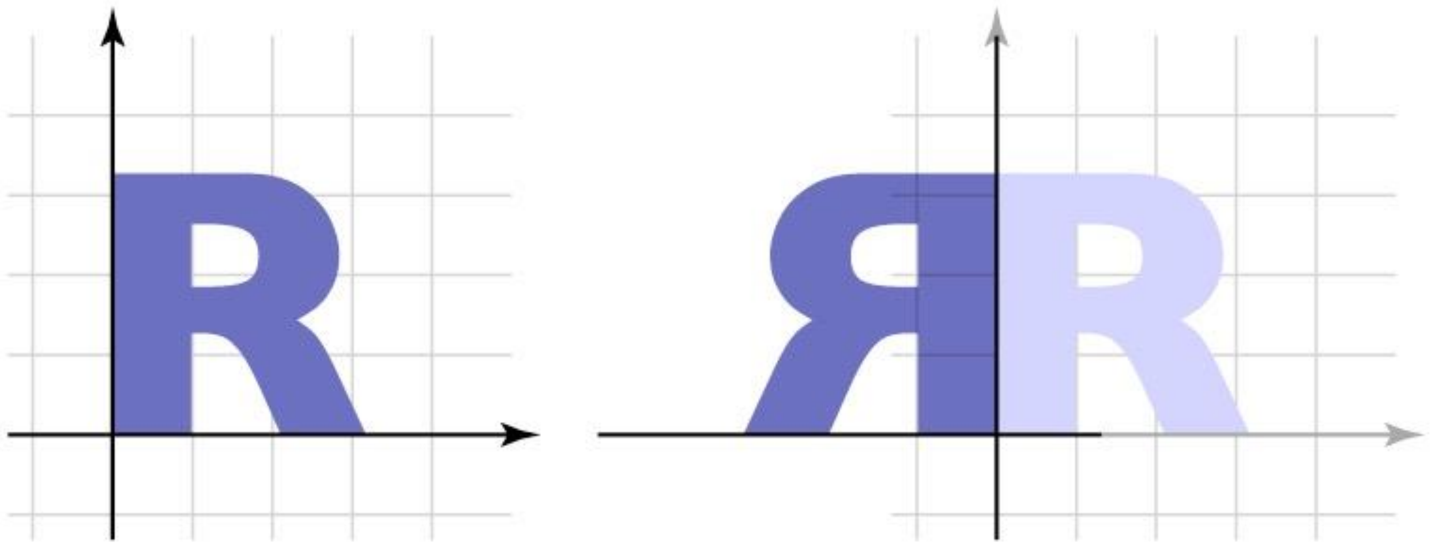




$$\begin{bmatrix} 1.5 & 0 \\ 0 & 0.8 \end{bmatrix}$$

Reflection

- just a special case of nonuniform scale



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

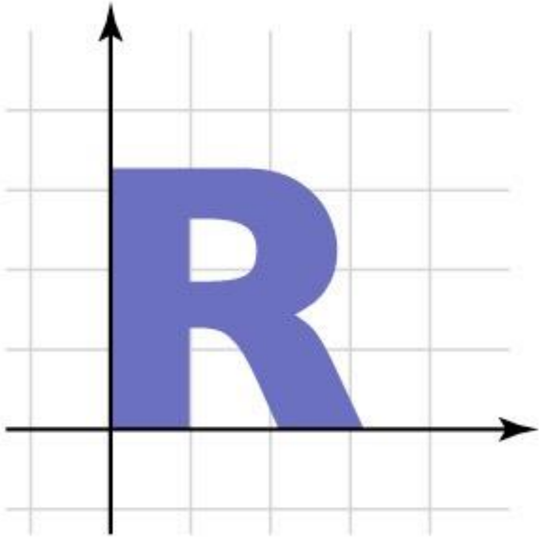
Rotation

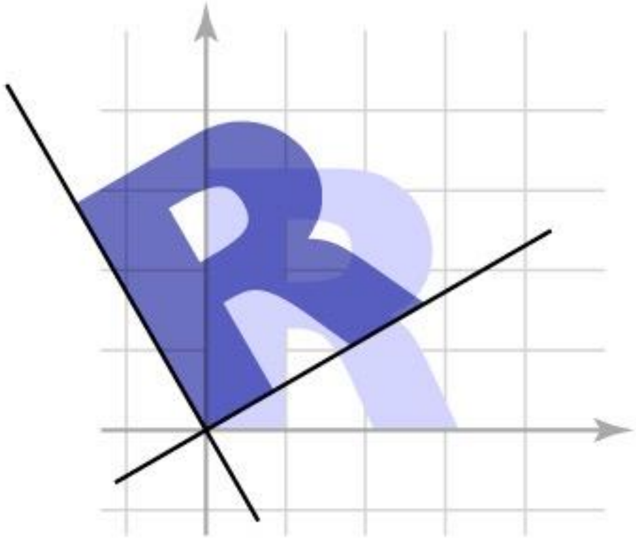
rotate **Counterclockwisely** by angle θ

$$R_\theta \mathbf{p} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} p_x \\ p_y \end{bmatrix} = \begin{bmatrix} p_x\cos\theta - p_y\sin\theta \\ p_x\sin\theta + p_y\cos\theta \end{bmatrix}$$

$$R_\theta^{-1} = R_{-\theta}$$

- A strategy: just memorize this matrix ¯_(ツ)_/¯
- Better strategy: understand why these are the columns

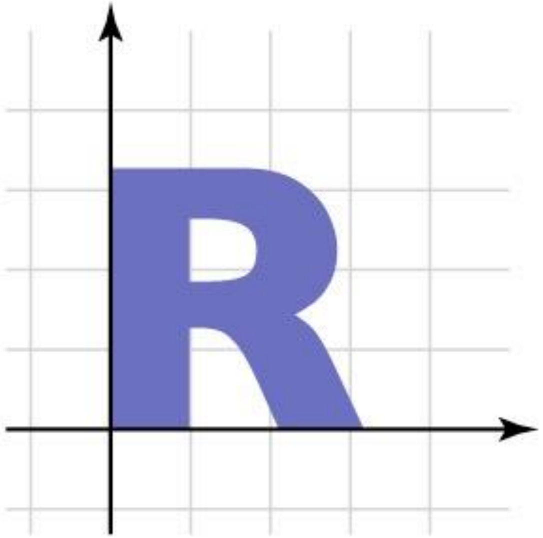


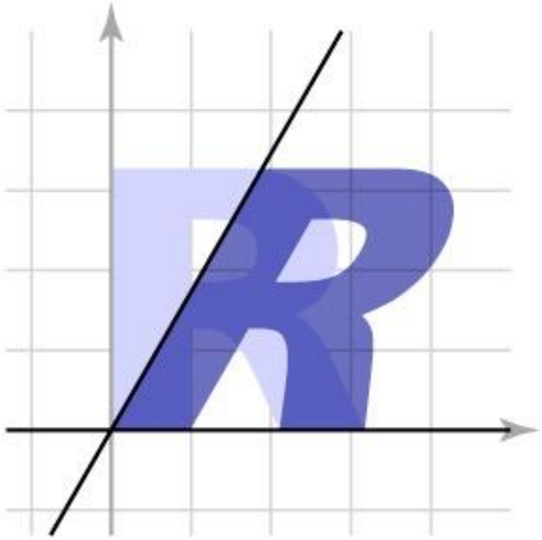


$$\begin{bmatrix} 0.866 & -0.05 \\ 0.5 & 0.866 \end{bmatrix}$$

Shear

$$Sh_{\mathbf{s}}\mathbf{p} = \begin{bmatrix} 1 & s_x \\ s_y & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \end{bmatrix} = \begin{bmatrix} p_x + s_x p_y \\ s_y p_x + p_y \end{bmatrix}$$





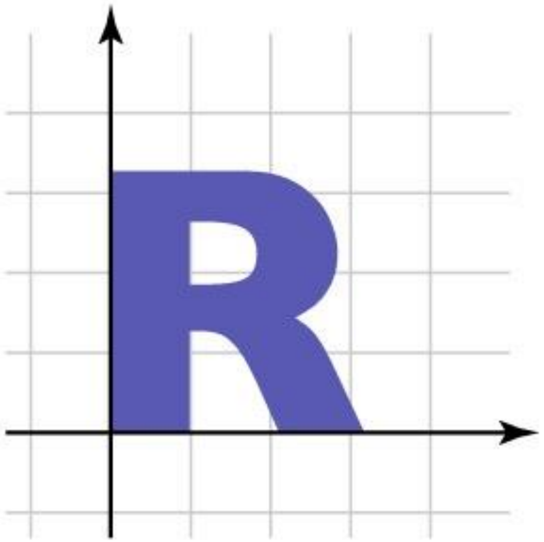
$$\begin{bmatrix} 1 & 0.5 \\ 0 & 1 \end{bmatrix}$$

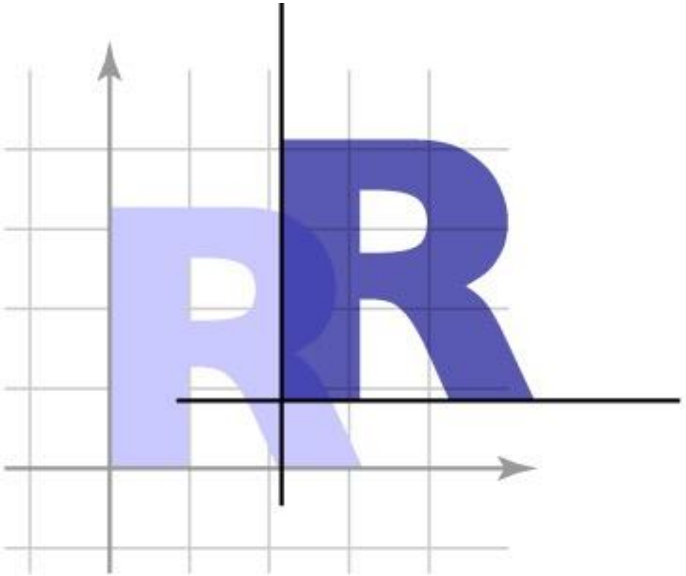
Translation, Affine Transform, and Homogeneous Coordinates

Translation

- Let's consider a simple translation:

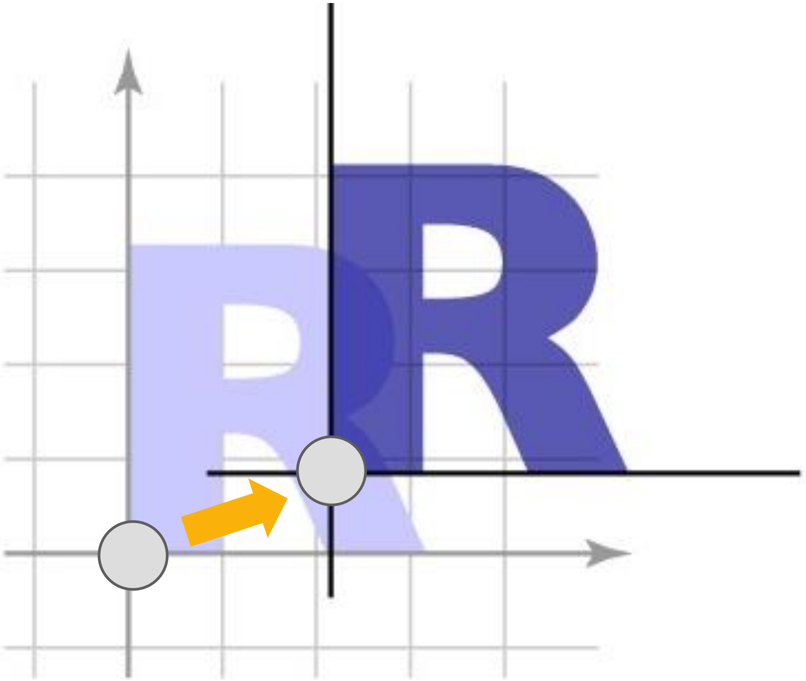
$$T_{\mathbf{t}}(\mathbf{p}) = \mathbf{p} + \mathbf{t}$$





Are translations linear?

- No! They don't keep origin fixed!
  - Let's check this quickly: $T_t(\mathbf{0}) = \mathbf{0} + \boldsymbol{t} = \boldsymbol{t} \mathrel{!}= \mathbf{0}$
- Therefore, we cannot use a matrix to represent translation
- Can we express scale, rotation, translation of objects/shapes using a single common representation?

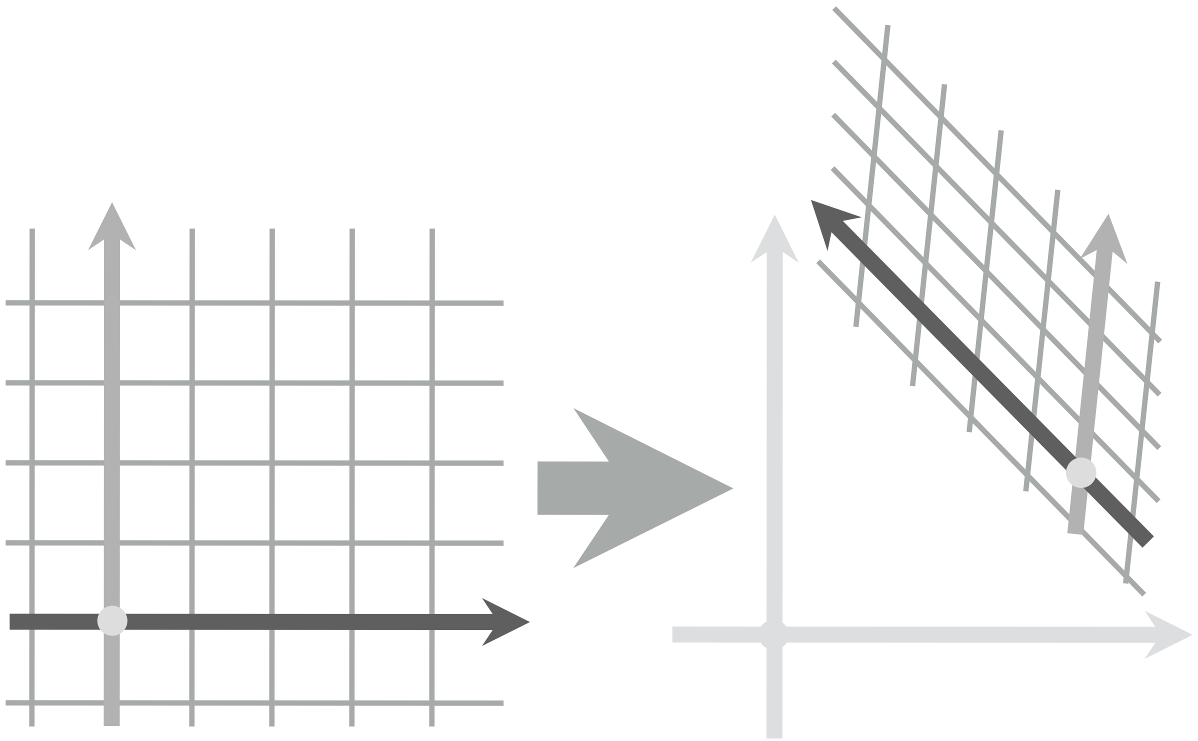


Affine Transforms

- Combine translation with linear transformation

$$T(\mathbf{p}) = M\mathbf{p} + \mathbf{t}$$

- Properties:
  - might **not** map origin to origin, but…
  - maps lines to lines
  - parallel lines remain parallel
  - length ratios are preserved
  - closed under composition



How do we represent an affine transform with **a single matrix multiplication**?

$$\boxed{T(\boldsymbol{p}) = \boldsymbol{M}\boldsymbol{p} + \boldsymbol{t}} \Rightarrow \boxed{T(\boldsymbol{p}) = \boldsymbol{M}'\boldsymbol{p}}$$

Key Idea: Introducing Another Dimension to Represent Translation!

- Represent translation using the extra column

$$T_{\mathbf{t}}\mathbf{p} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ 1 \end{bmatrix} = \begin{bmatrix} p_x + t_x \\ p_y + t_y \\ 1 \end{bmatrix}$$

Homogeneous Coordinates

- Represent translation using the extra column

$$T_{\mathbf{t}}\mathbf{p} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ 1 \end{bmatrix} = \begin{bmatrix} p_x + t_x \\ p_y + t_y \\ 1 \end{bmatrix}$$

- Linear transform occupies the upper-left 2×2 block

$$M\mathbf{p} = \begin{bmatrix} m_{11} & m_{12} & 0 \\ m_{21} & m_{22} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ 1 \end{bmatrix} = \begin{bmatrix} m_{11}p_x + m_{12}p_y \\ m_{21}p_x + m_{22}p_y \\ 1 \end{bmatrix}$$

Homogeneous coordinates

- Put the linear part and the translation part together:

$$T(\mathbf{p}) = M\mathbf{p} + \mathbf{t} = \begin{bmatrix} m_{11} & m_{12} & t_x \\ m_{21} & m_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ 1 \end{bmatrix} = \begin{bmatrix} M & \mathbf{t} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{p} \\ 1 \end{bmatrix}$$

Let's see it in a step-by-step way:
First, we represent a translation

$$\begin{pmatrix} 1 & 0 & a \\ 0 & 1 & b \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} x + a \\ y + b \end{pmatrix}$$

translation matrix

homogenous coordinates

Next, we incorporate the linear transform part into the same matrix

$$\begin{pmatrix} a & b & e \\ c & d & f \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} ax + by + e \\ cx + dy + f \end{pmatrix}$$

The left 2x2 is for linear transform

$$\begin{pmatrix} \boxed{\begin{matrix} a & b \\ c & d \end{matrix}} & \begin{matrix} e \\ f \end{matrix} \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} ax + by + e \\ cx + dy + f \end{pmatrix}$$

linear

The right 1x2 is for translation

$$\begin{pmatrix} a & b & \boxed{e} \\ c & d & \boxed{f} \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} ax + by + e \\ cx + dy + f \end{pmatrix}$$

translation

The last component of the vector is the homogeneous coordinate; but the result is still non-homogeneous (with only two components)!

$$\begin{pmatrix} a & b & e \\ c & d & f \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} ax + by + e \\ cx + dy + f \end{pmatrix}$$

homogenous coordinates

non-homogenous coordinates

Let's extend the 2x3 matrix to 3x3, and append another component to the result as well!

$$\begin{pmatrix} a & b & e \\ c & d & f \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} ax + by + e \\ cx + dy + f \\ 1 \end{pmatrix}$$

affine

homogenous coordinates

homogenous coordinates

Finally, we can represent an affine transform with a single matrix multiplication:

$$T(\mathbf{p}) = M\mathbf{p} + \mathbf{t} = \begin{bmatrix} m_{11} & m_{12} & t_x \\ m_{21} & m_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ 1 \end{bmatrix} = \begin{bmatrix} M & \mathbf{t} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{p} \\ 1 \end{bmatrix}$$

Transforming Points and Vectors

- **Quick recap:** Points and vectors are different entities
  - vectors: encode direction and length (difference of points)
  - points: encode position (origin plus a vector)
- **Vectors**: transform without translation
  - we never translate a vector because vectors with the same orientation and magnitude always represent the same vector!

$$T(\mathbf{v}) = M\mathbf{v}$$

- **Points**: transform with both linear transform and translation

$$T(\mathbf{p}) = M\mathbf{p} + \mathbf{t}$$

How do we represent vector transforms and point transforms with the same expression?

- Again, use the homogeneous coordinate w:
  - set to ***w*=1** for points

$$\mathbf{p} = \begin{bmatrix} p_x \\ p_y \\ p_w \end{bmatrix} = \begin{bmatrix} p_x \\ p_y \\ 1 \end{bmatrix}$$

  - set to ***w*=0** for vectors

$$\mathbf{v} = \begin{bmatrix} v_x \\ v_y \\ 0 \end{bmatrix}$$

Transforming points and vectors

- For vectors, **zero** homogeneous coordinates let us exclude translation
  - just put 0 rather than 1 for w coordinate

$$\begin{bmatrix} M & \mathbf{t} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{p} \\ 1 \end{bmatrix} = \begin{bmatrix} M\mathbf{p} + \mathbf{t} \\ 1 \end{bmatrix} \qquad \begin{bmatrix} M & \mathbf{t} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ 0 \end{bmatrix} = \begin{bmatrix} M\mathbf{v} \\ 0 \end{bmatrix}$$

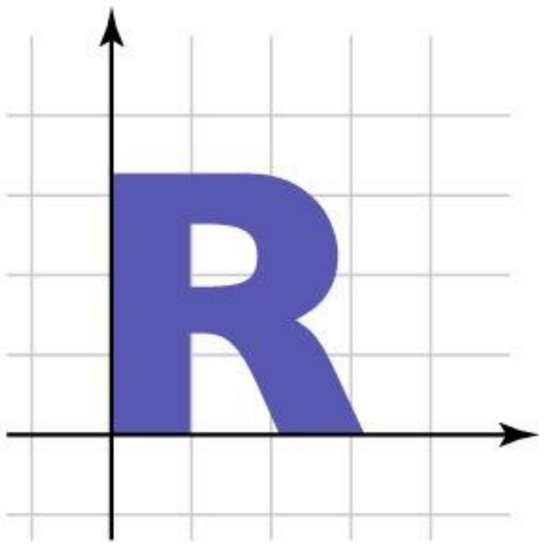
  - and note that subtracting two points cancels the extra coordinate, resulting in a vector!

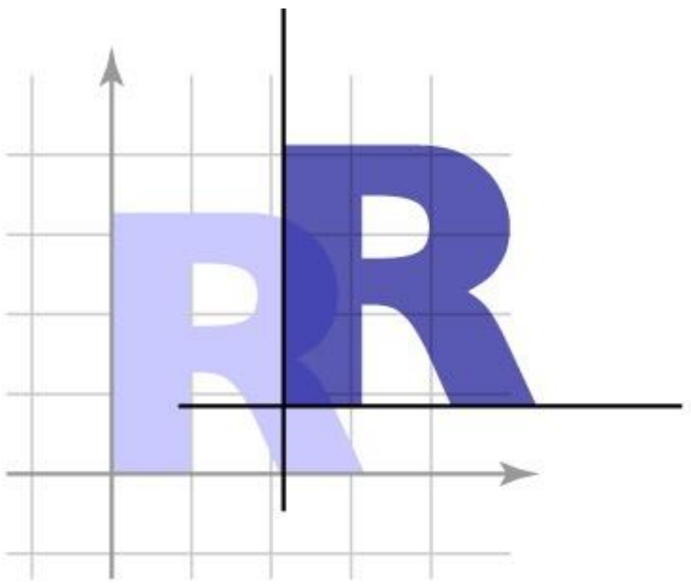
$$\begin{bmatrix} \mathbf{p} \\ 1 \end{bmatrix} - \begin{bmatrix} \mathbf{q} \\ 1 \end{bmatrix} = \begin{bmatrix} p_x - q_x \\ p_y - q_y \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{v} \\ 0 \end{bmatrix}$$

Key takeaways:
(1) Use 1 for the last component if you want to transform a point
(2) Use 0 for the last component if you want to transform a vector
(3) Subtracting two points gives a vector

Rewrite Translation with Homogeneous Coordinates

$$T_{\mathbf{t}}\mathbf{p} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ 1 \end{bmatrix} = \begin{bmatrix} p_x + t_x \\ p_y + t_y \\ 1 \end{bmatrix}$$

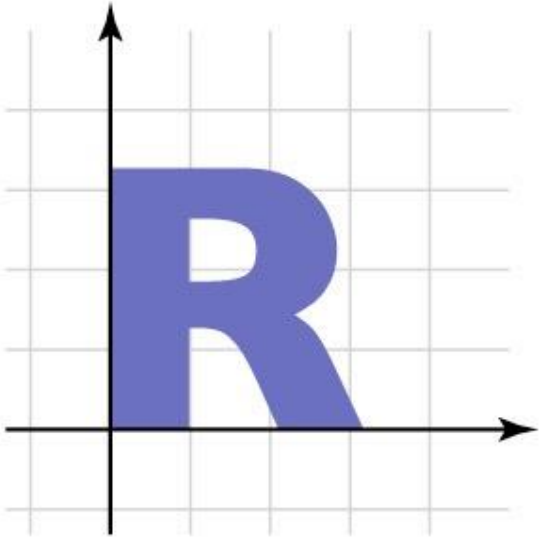


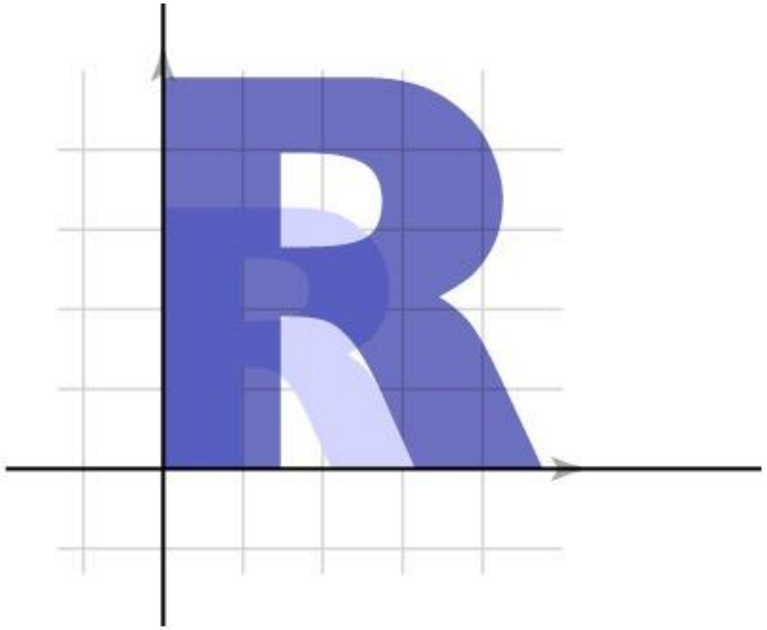


$$\begin{bmatrix} 1 & 0 & 2.15 \\ 0 & 1 & 0.85 \\ 0 & 0 & 1 \end{bmatrix}$$

Rewrite Uniform Scaling with Homogeneous Coordinates

$$S_s\mathbf{p} = \begin{bmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ 1 \end{bmatrix} = \begin{bmatrix} sp_x \\ sp_y \\ 1 \end{bmatrix}$$

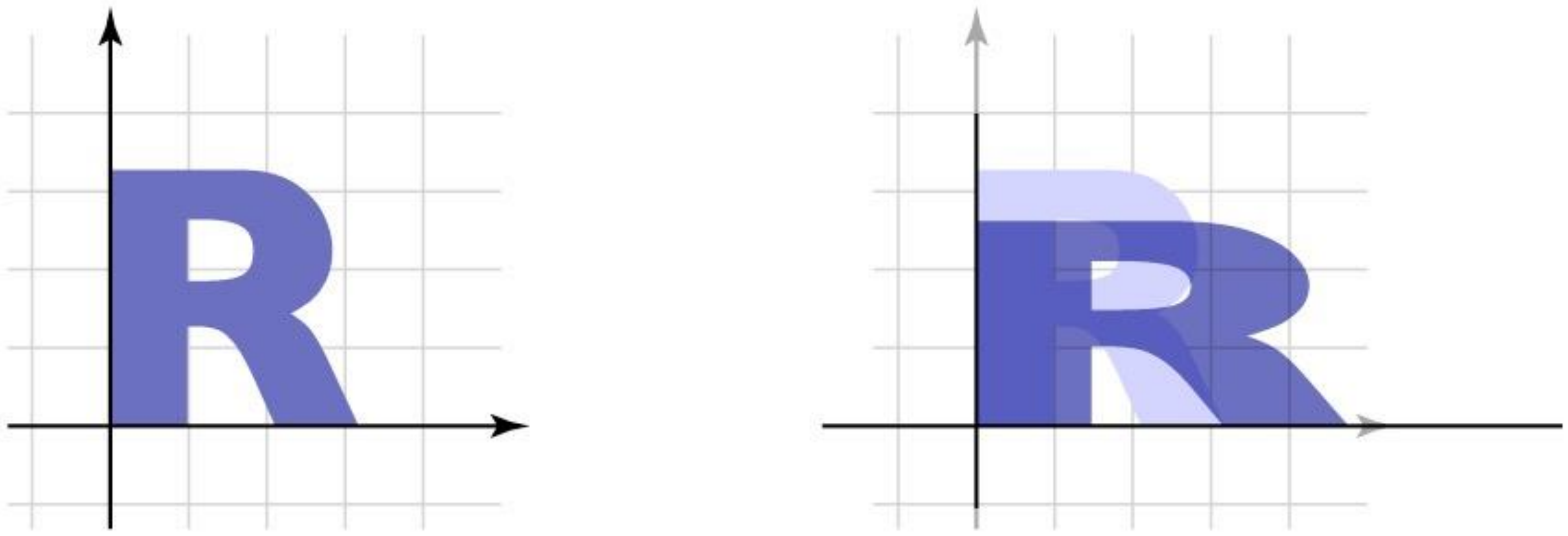




$$\begin{bmatrix} 1.5 & 0 & 0 \\ 0 & 1.5 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Rewrite Nonuniform Scaling with Homogeneous Coordinates

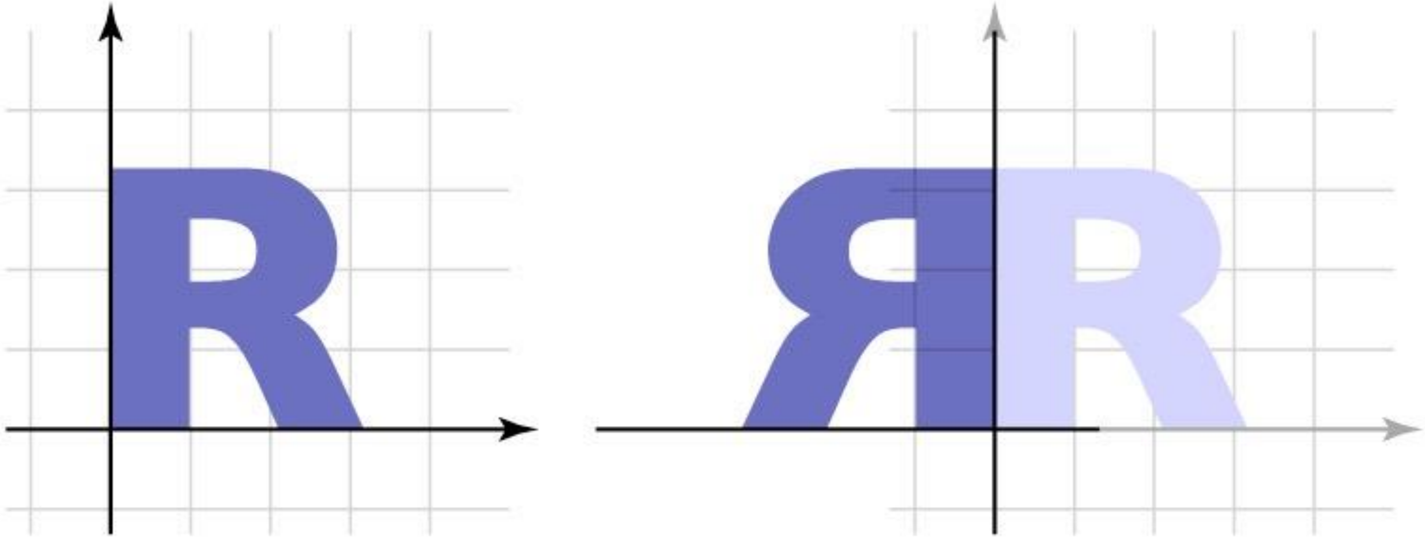
$$S_{\mathbf{s}}\mathbf{p} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ 1 \end{bmatrix} = \begin{bmatrix} s_x p_x \\ s_y p_y \\ 1 \end{bmatrix}$$



$$\begin{bmatrix} 1.5 & 0 & 0 \\ 0 & 0.8 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Rewrite Reflection with Homogeneous Coordinates

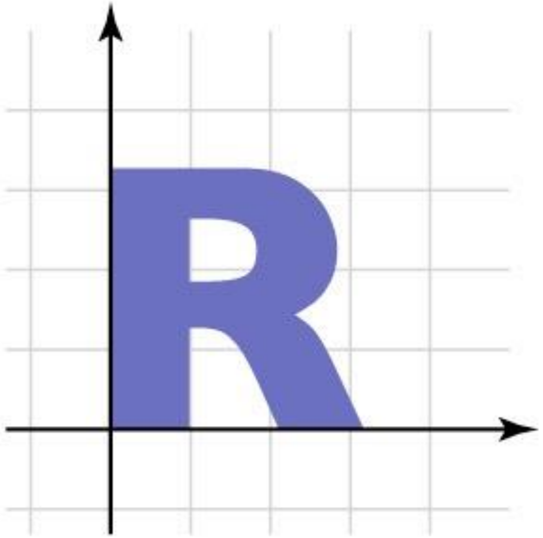
- just a special case of nonuniform scale

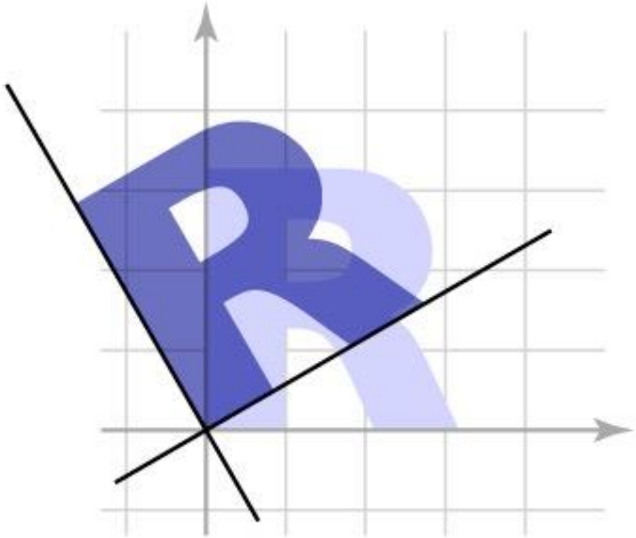


$$\begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Rewrite Rotation with Homogeneous Coordinates

$$R_\theta \mathbf{p} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ 1 \end{bmatrix} = \begin{bmatrix} p_x\cos\theta - p_y\sin\theta \\ p_x\sin\theta + p_y\cos\theta \\ 1 \end{bmatrix}$$

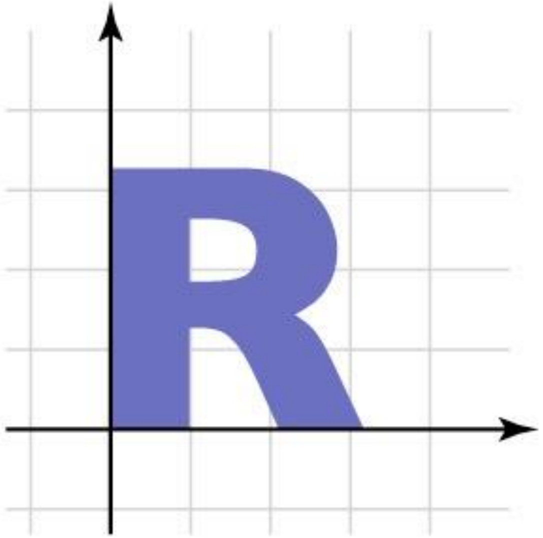


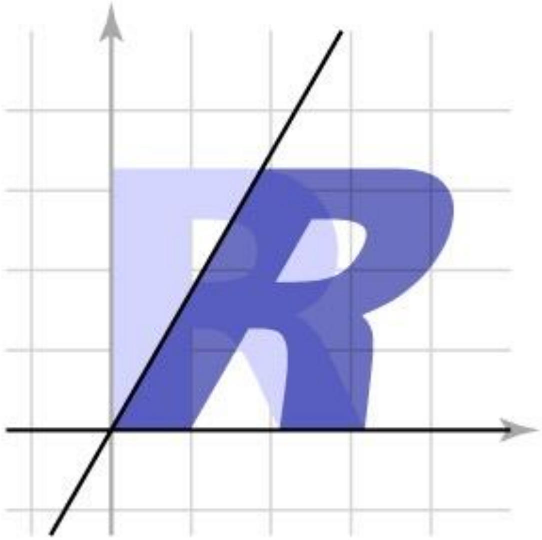


$$\begin{bmatrix} 0.866 & -0.05 & 0 \\ 0.05 & 0.866 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Rewrite Shear with Homogeneous Coordinates

$$Sh_{\mathbf{s}}\mathbf{p} = \begin{bmatrix} 1 & s_x & 0 \\ s_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ 1 \end{bmatrix} = \begin{bmatrix} p_x + s_x p_y \\ s_y p_x + p_y \\ 1 \end{bmatrix}$$





$$\begin{bmatrix} 1 & 0.5 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Composite Transformations

Order of linear and translation parts

- Does the affine transform:

$$\begin{bmatrix} M & \mathbf{t} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{p} \\ 1 \end{bmatrix}$$

e.g. $$\begin{bmatrix} s_x & 0 & t_x \\ 0 & s_y & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ 1 \end{bmatrix}$$

1. transform by M first, and then translate by $\mathbf{t}$, or
2. translate by $\mathbf{t}$ first and then transform by M?

Let's write down the matrix expressions for the two cases and check

Matrix expressions of the two cases

- First translate and then linear transform

$$\begin{bmatrix} M & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p \\ 1 \end{bmatrix} = \begin{bmatrix} M & Mt \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p \\ 1 \end{bmatrix} = \begin{bmatrix} Mp + Mt \\ 1 \end{bmatrix}$$

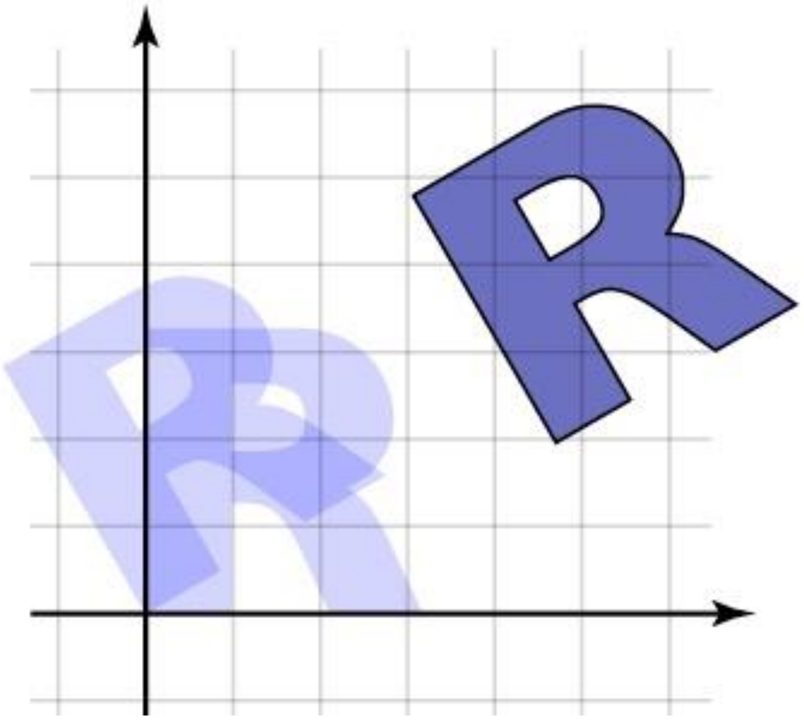
- First linear transform and then translate

$$\begin{bmatrix} I & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} M & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p \\ 1 \end{bmatrix} = \begin{bmatrix} M & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p \\ 1 \end{bmatrix} = \begin{bmatrix} Mp + t \\ 1 \end{bmatrix}$$

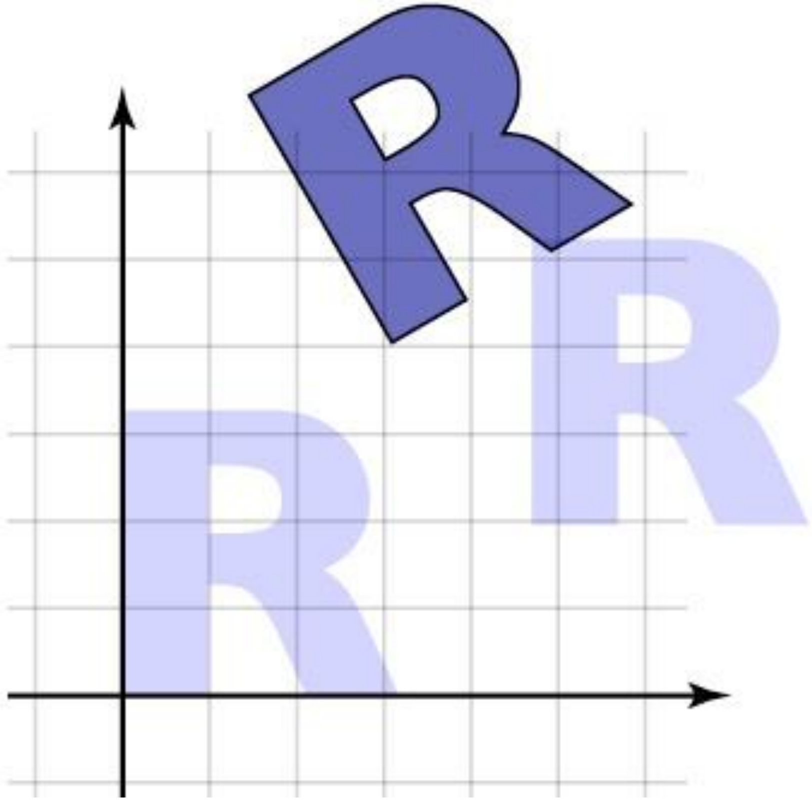
An affine transform exerts the **M** transform first and then the **t** translation.

Composite Affine Transformations

- In general not commutative: **order matters**!



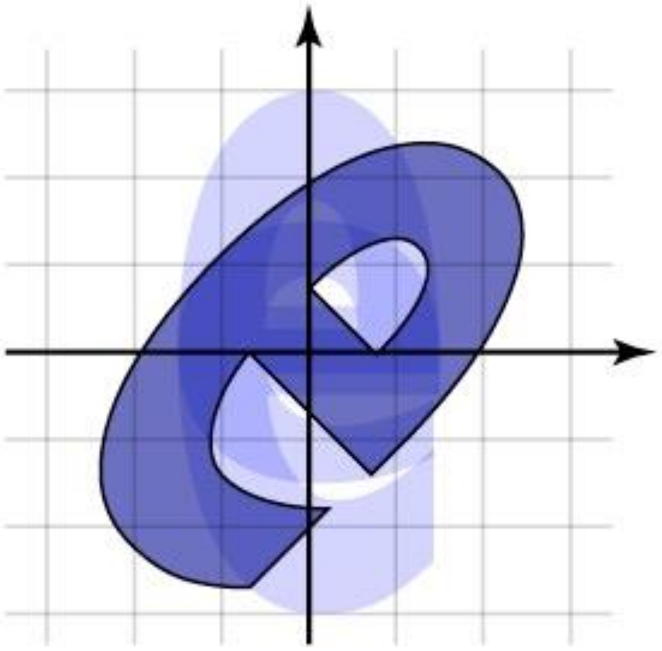
rotate, then translate



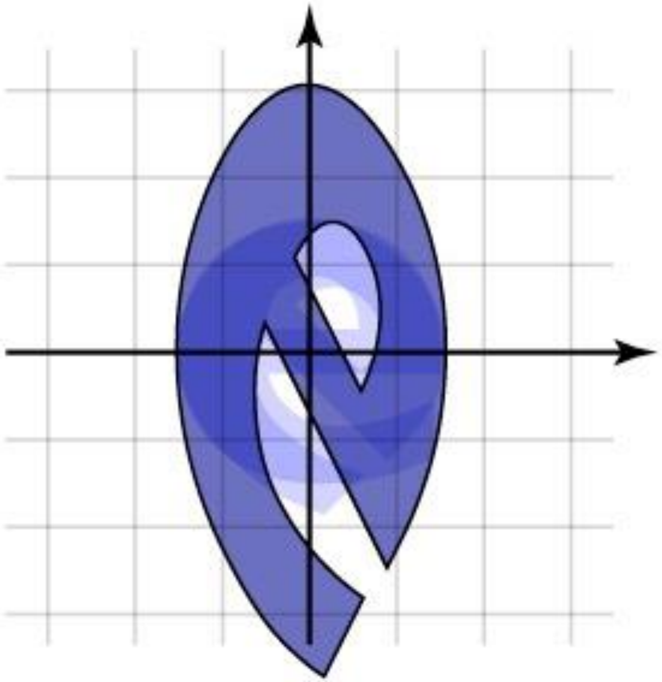
translate, then rotate

Composite affine transformations

- Another example



scale, then rotate

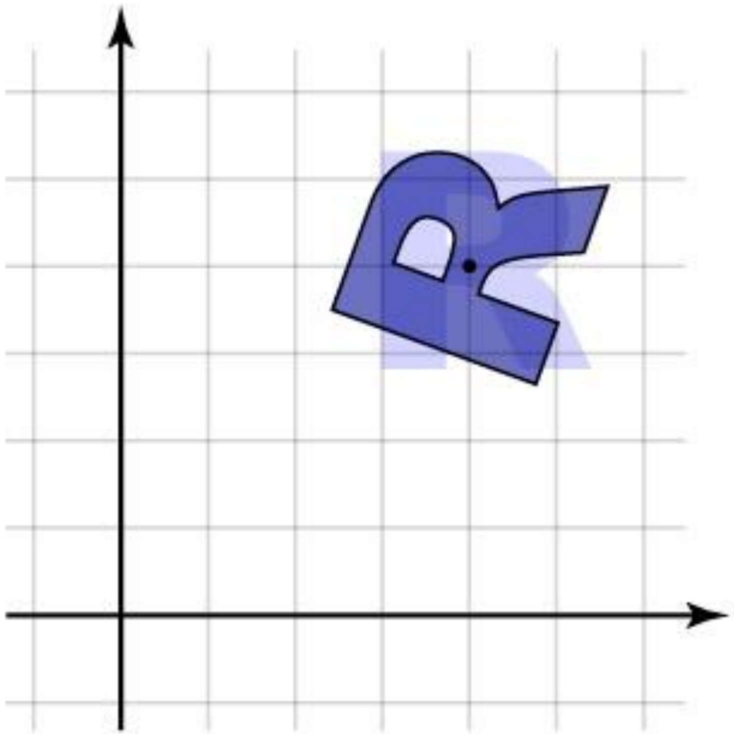


rotate, then scale

Composing to change axes

- Want to rotate about a particular point
- Know how to rotate about the origin
  - so translate that point to the origin first
  - then rotate
  - then translate point back

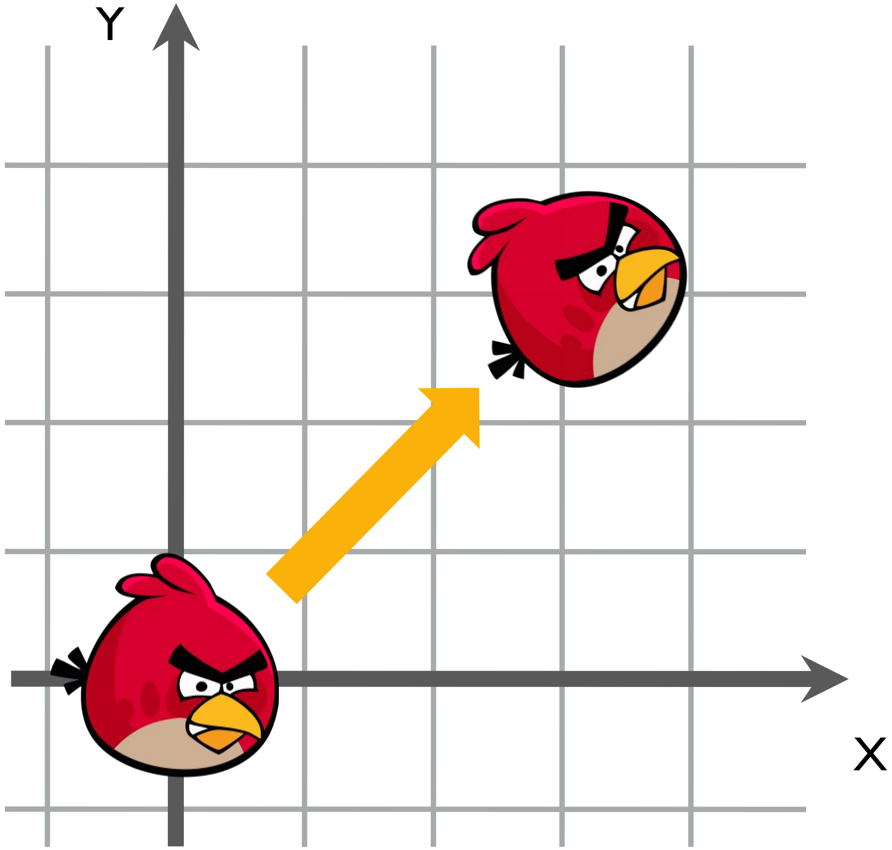
$$M = T^{-1}RT$$



$$M = ?$$

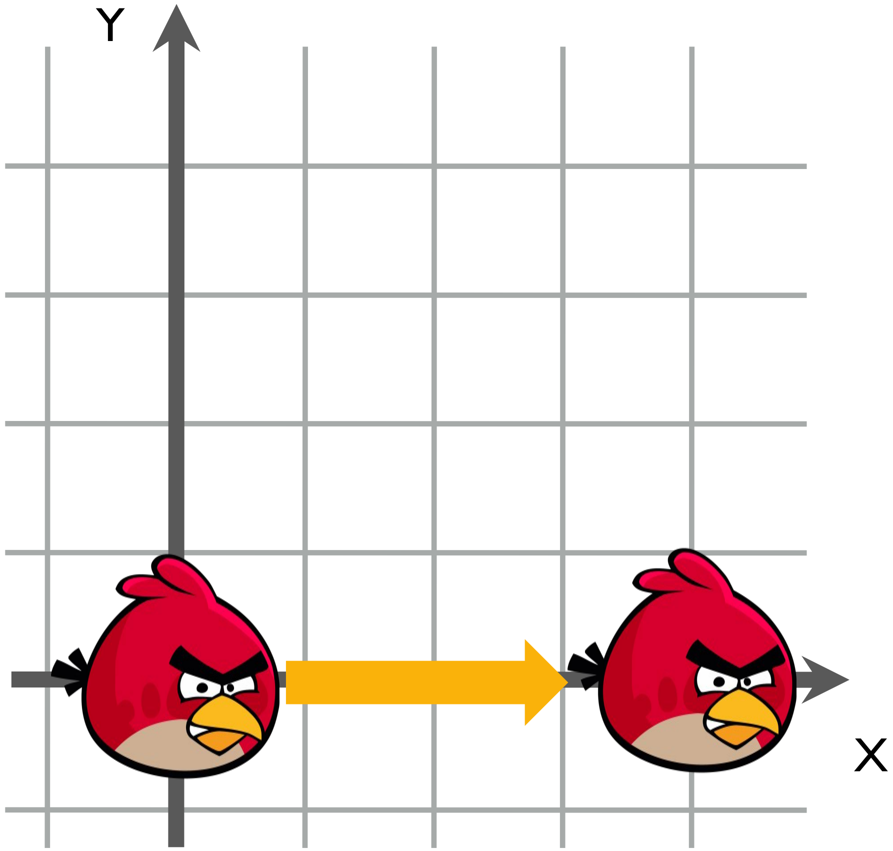
Now let's practice by playing an angry-bird game

- Given a picture, write down its affine transformation matrix



$$\begin{bmatrix} ? & ? & ? \\ ? & ? & ? \\ ? & ? & ? \end{bmatrix}$$

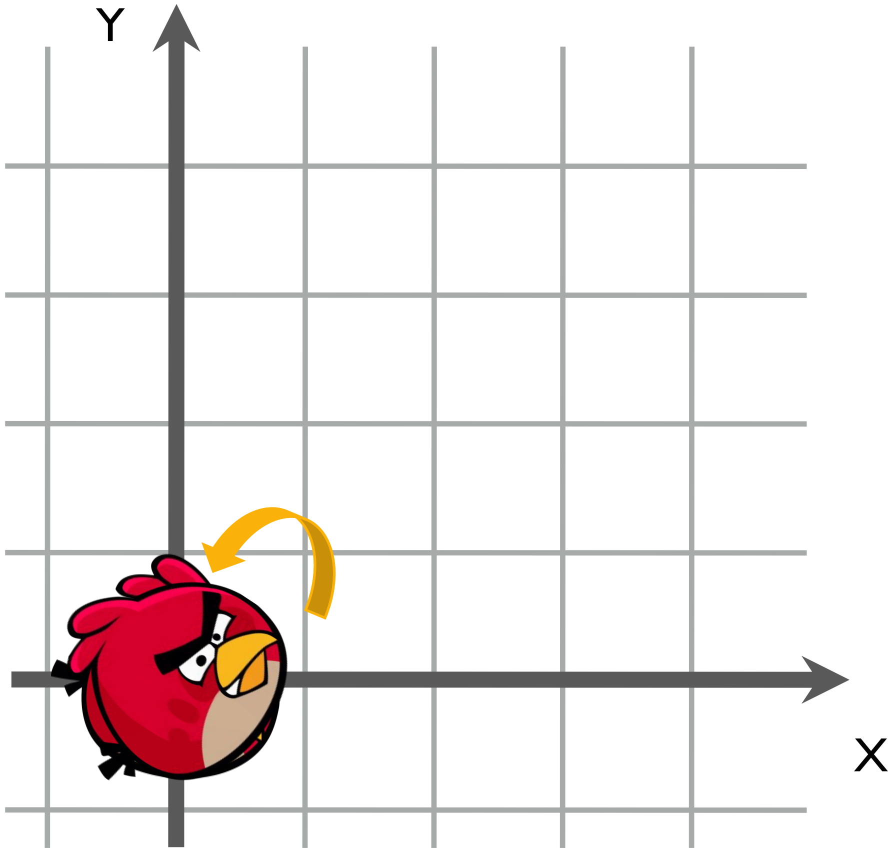
A Warmup Practice



- Translate in X by 4 units

$$\begin{bmatrix} 1 & 0 & 4 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

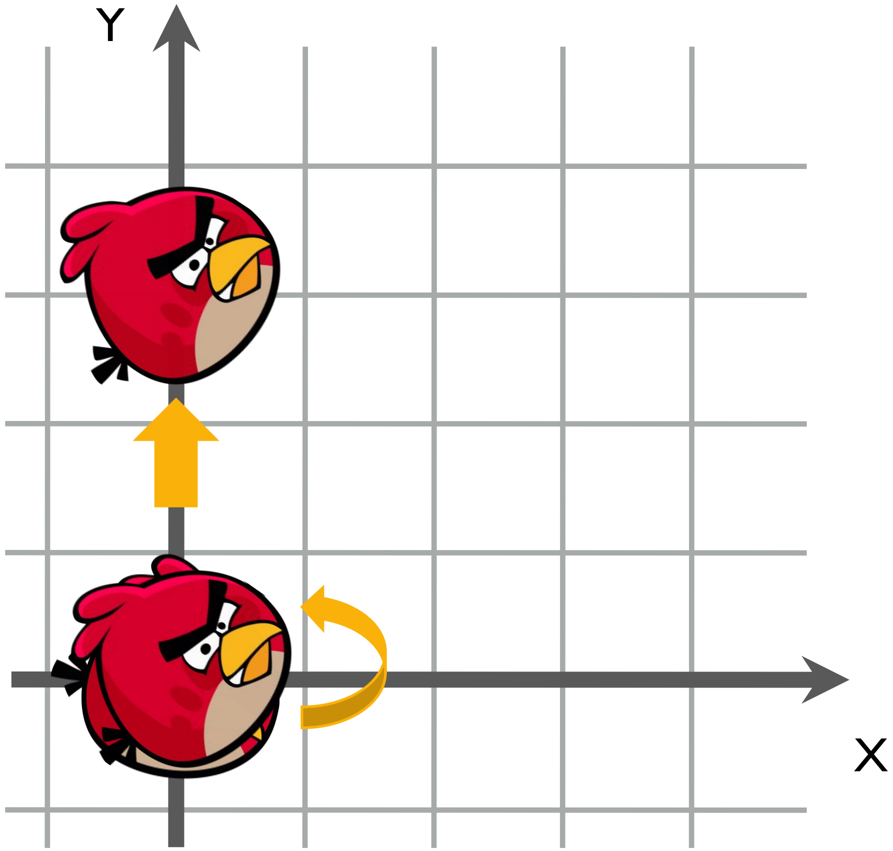
Another Warmup Practice



- Rotate around the origin by 45 degrees

$$\begin{bmatrix} \sqrt{2}/2 & -\sqrt{2}/2 & 0 \\ \sqrt{2}/2 & \sqrt{2}/2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

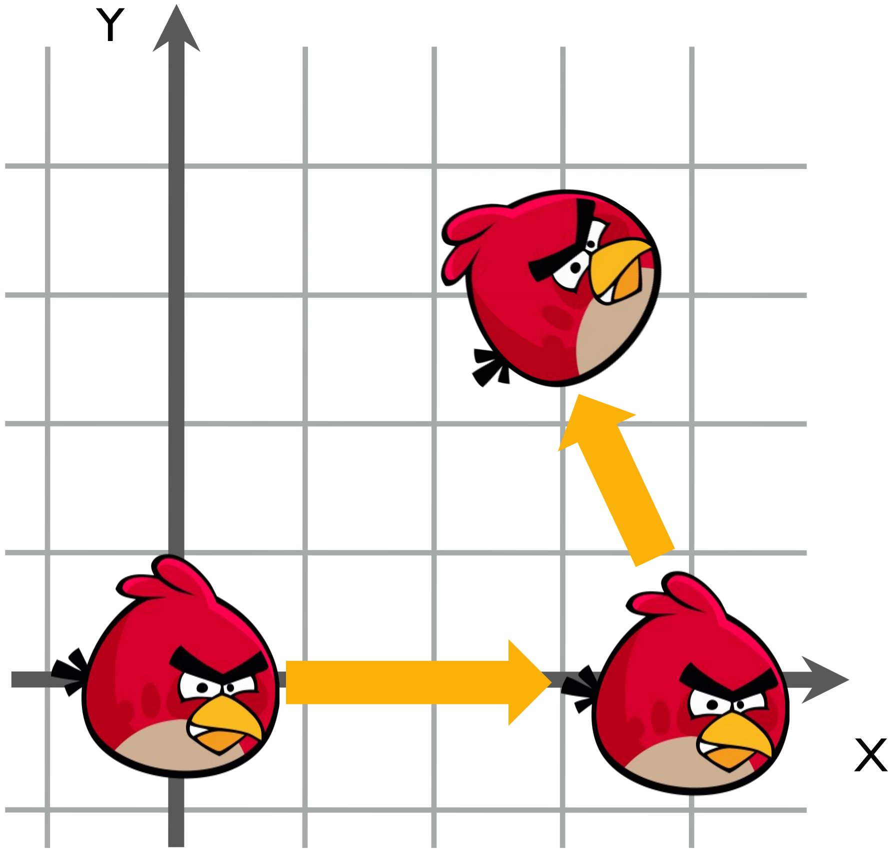
A Composite Case



- Rotate around **the origin** by 45 degrees
- Translate in Y by 3 units

$$\begin{bmatrix} \sqrt{2}/2 & -\sqrt{2}/2 & 0 \\ \sqrt{2}/2 & \sqrt{2}/2 & 3 \\ 0 & 0 & 1 \end{bmatrix}$$

Another Composite Case



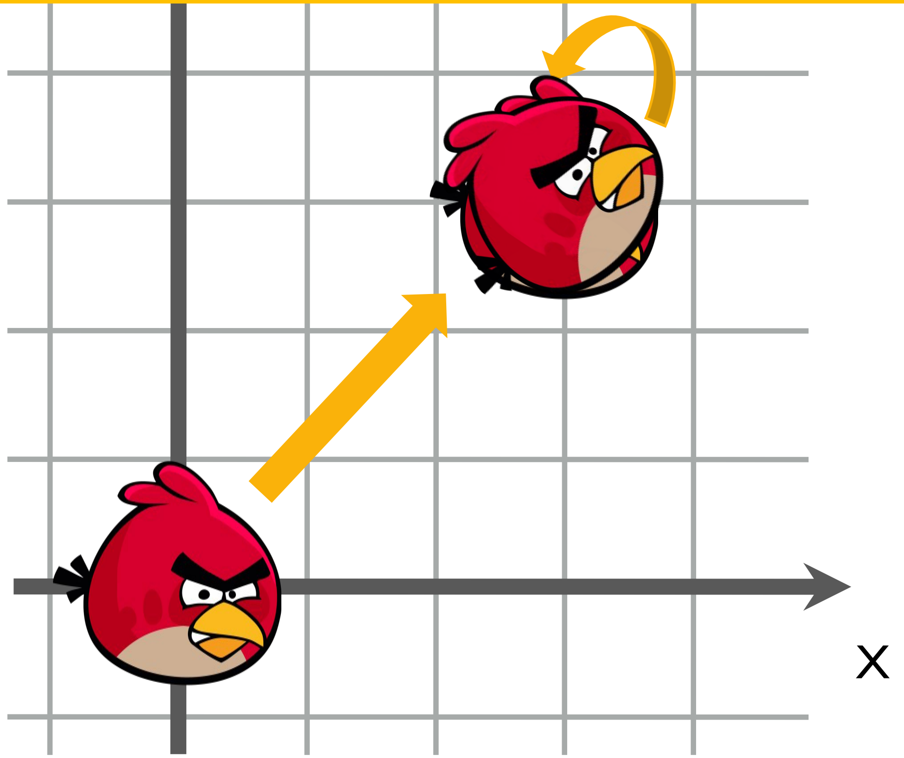
- Translate in X by 3 units
- Rotate around **the origin** by 45 degrees

$$\begin{bmatrix} \sqrt{2}/2 & -\sqrt{2}/2 & 0 \\ \sqrt{2}/2 & \sqrt{2}/2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 3 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

A More Challenging Composite Case

- A very common case in practice: **translate, and then rotate**
- The matrix multiplication order is:

$$M_{translation}M_{rotation}$$



- Translate in both X and Y by 3 units each
- Rotate around its current center by 45 degrees

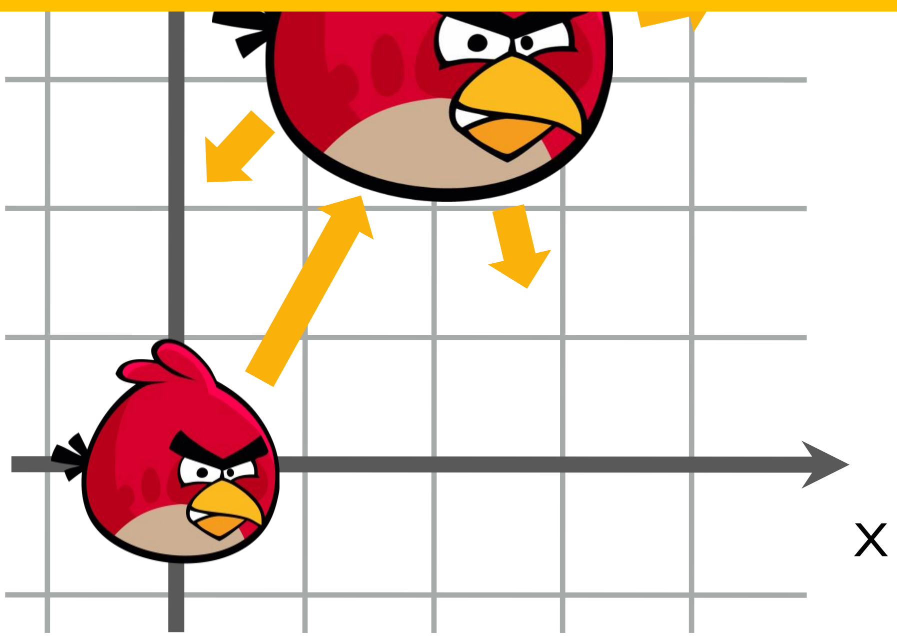
$$\begin{bmatrix}1 & 0 & 3\\0 & 1 & 3\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}\sqrt{2}/2 & -\sqrt{2}/2 & 0\\\sqrt{2}/2 & \sqrt{2}/2 & 0\\0 & 0 & 1\end{bmatrix}$$

Why? $\begin{bmatrix}1 & 0 & 2\\0 & 1 & 3\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}2 & 0 & 0\\0 & 2 & 0\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0 & -3\\0 & 1 & -3\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0 & 3\\0 & 1 & 3\\0 & 0 & 1\end{bmatrix}$, the rightmost two matrices cancel each other

Another Common Practice Case

- A very common case in practice: **translate, and then scale**
- The matrix multiplication order is:

$$M_{translation} M_{scaling}$$

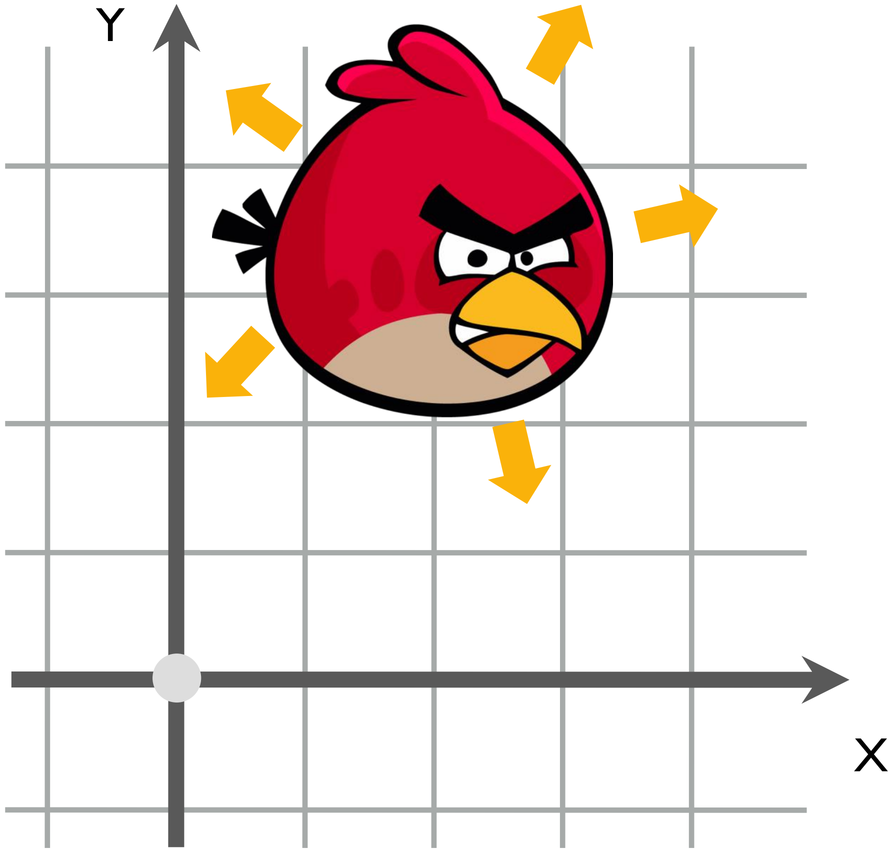


- Translate in X by 2 units and Y by 3 units
- Scale around its current center by 2 times

$$\begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Why? $\begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -2 \\ 0 & 1 & -3 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{bmatrix}$,
the rightmost two matrices cancel each other

What will happen if we don’t start from the origin?



- Scale around [2,3] by 2 times

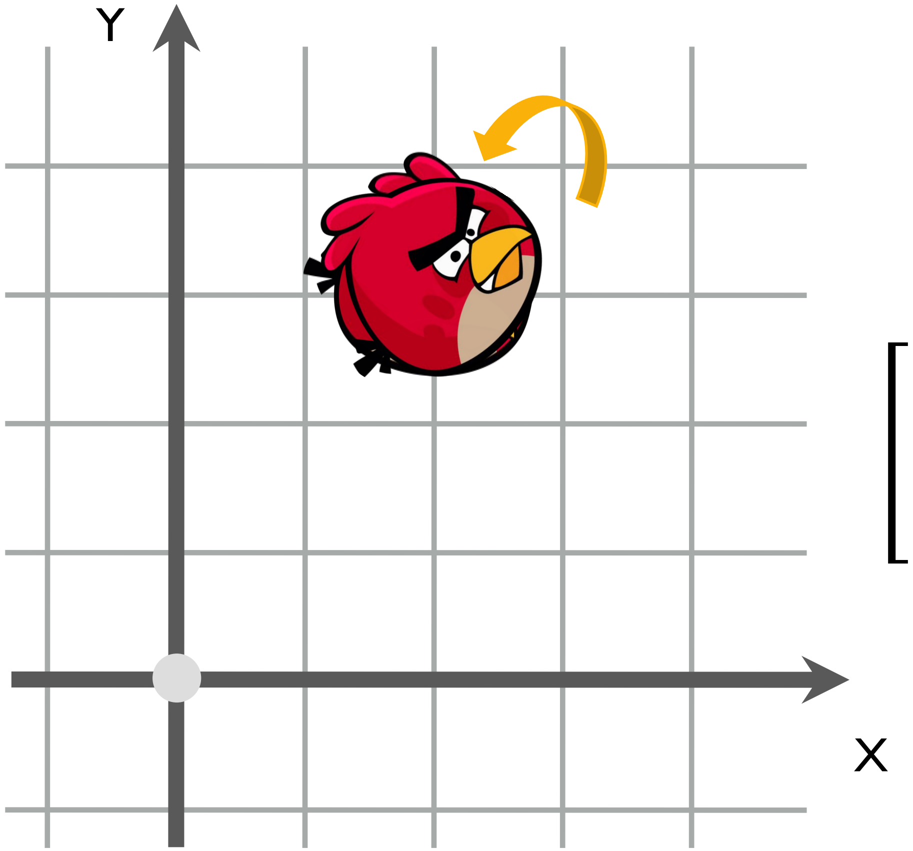
$$\begin{bmatrix}1 & 0 & 2\\0 & 1 & 3\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}2 & 0 & 0\\0 & 2 & 0\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0 & -2\\0 & 1 & -3\\0 & 0 & 1\end{bmatrix}$$

Translate to current location

Scaling

Translate back to origin

Another example not starting from the origin



- Rotate around [2,3] by 45 degrees

$$\begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \sqrt{2}/2 & -\sqrt{2}/2 & 0 \\ \sqrt{2}/2 & \sqrt{2}/2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -2 \\ 0 & 1 & -3 \\ 0 & 0 & 1 \end{bmatrix}$$

Translate to current location

Rotation

Translate back to origin

3D Transformations

3D Transformations

- Adopt homogeneous formulation in 3D
  - points have 4 coordinates
  - affine transformations are 4x4 matrices
- Most concepts generalize very easily
  - though rotation gets much more complex
- Example:
  - Extend a 2D transform to 3D by assuming no transforms happen in Z axis

$$\begin{bmatrix} M_{11} & M_{12} & t_x \\ M_{21} & M_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} M_{11} & M_{12} & 0 & t_x \\ M_{21} & M_{22} & 0 & t_y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3D Transformations

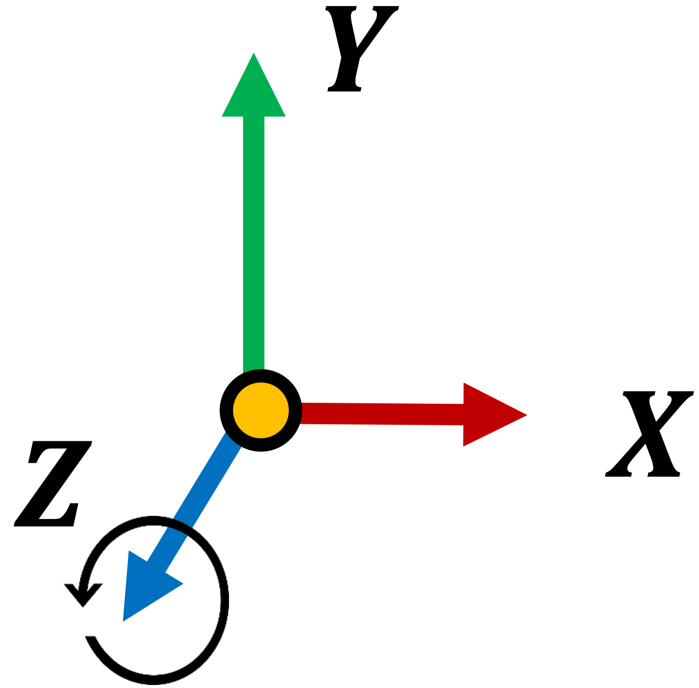
$$T_{\mathbf{t}}\mathbf{p} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} = \begin{bmatrix} p_x + t_x \\ p_y + t_y \\ p_z + t_z \\ 1 \end{bmatrix}$$

3D Scaling

$$S_{\mathbf{s}}\mathbf{p} = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} = \begin{bmatrix} s_x p_x \\ s_y p_y \\ s_z p_z \\ 1 \end{bmatrix}$$

3D Rotation around Z Axis

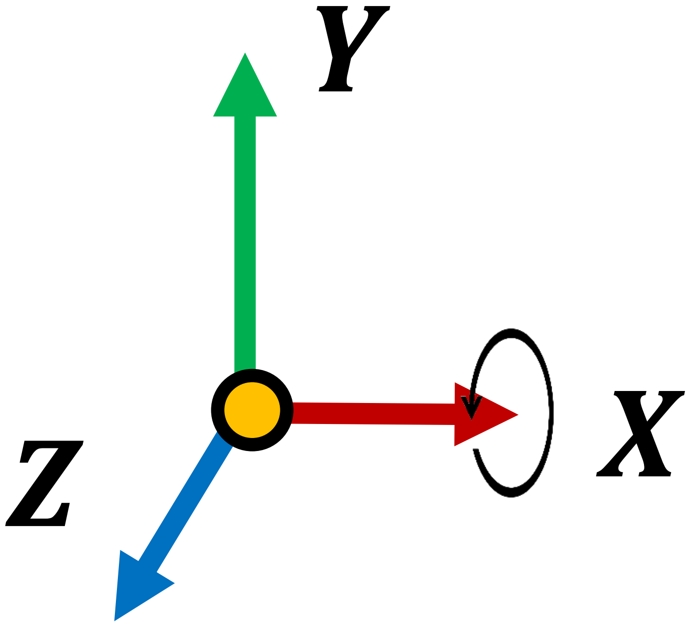
$$R_\theta^z \mathbf{p} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} = \begin{bmatrix} p_x\cos\theta - p_y\sin\theta \\ p_x\sin\theta + p_y\cos\theta \\ p_z \\ 1 \end{bmatrix}$$



- The row and column corresponding to Z axis are filled with 0 (off-diagonal) and 1 (diagonal)
- The submatrix by eliminating the Z's row and column is a standard rotation matrix in 2D with homogeneous coordinates

3D Rotation around X Axis

$$R_\theta^x \mathbf{p} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} = \begin{bmatrix} p_x \\ p_y\cos\theta - p_z\sin\theta \\ p_y\sin\theta + p_z\cos\theta \\ 1 \end{bmatrix}$$



- The row and column corresponding to X axis are filled with 0 (off-diagonal) and 1 (diagonal)
- The submatrix by eliminating the X's row and column is a standard rotation matrix in 2D with homogeneous coordinates

Can you extend this to rotation around Y?

3D Rotation around Y Axis

Rotation X

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

Rotation Y

$$\cancel{\begin{bmatrix} c & 0 & -s & 0 \\ 0 & 1 & 0 & 0 \\ s & 0 & c & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}$$

Rotation Z

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

This is the correct matrix!

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

Two perspectives to understand:

(1) Transform the axis: calculate the rotated X and Z axes

(2) Think of the permutation ZXY

A General Case

The matrix of a proper rotation R by angle ϑ around the axis $\mathbf{u} = (u_x, u_y, u_z)$, a unit vector with $u^2_x + u^2_y + u^2_z = 1$, is given by:

$$R = \begin{bmatrix} \cos\theta + u_x^2\left(1-\cos\theta\right) & u_x u_y\left(1-\cos\theta\right) - u_z\sin\theta & u_x u_z\left(1-\cos\theta\right) + u_y\sin\theta \\ u_y u_x\left(1-\cos\theta\right) + u_z\sin\theta & \cos\theta + u_y^2\left(1-\cos\theta\right) & u_y u_z\left(1-\cos\theta\right) - u_x\sin\theta \\ u_z u_x\left(1-\cos\theta\right) - u_y\sin\theta & u_z u_y\left(1-\cos\theta\right) + u_x\sin\theta & \cos\theta + u_z^2\left(1-\cos\theta\right) \end{bmatrix}$$

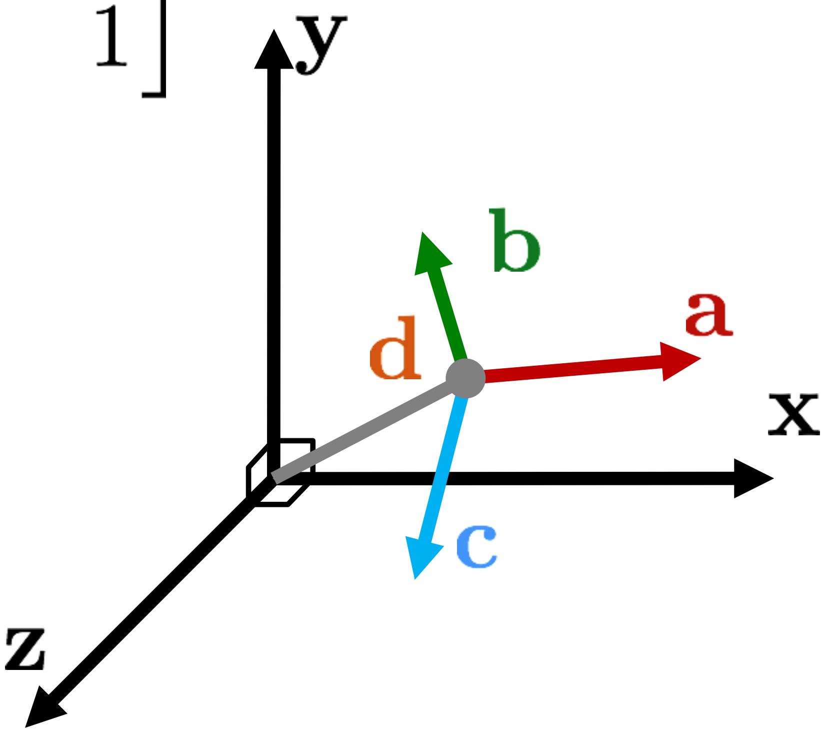
Beyond the scope of this class: requiring some advanced math on Euler angle to derive the formula

https://en.wikipedia.org/wiki/Rotation_matrix

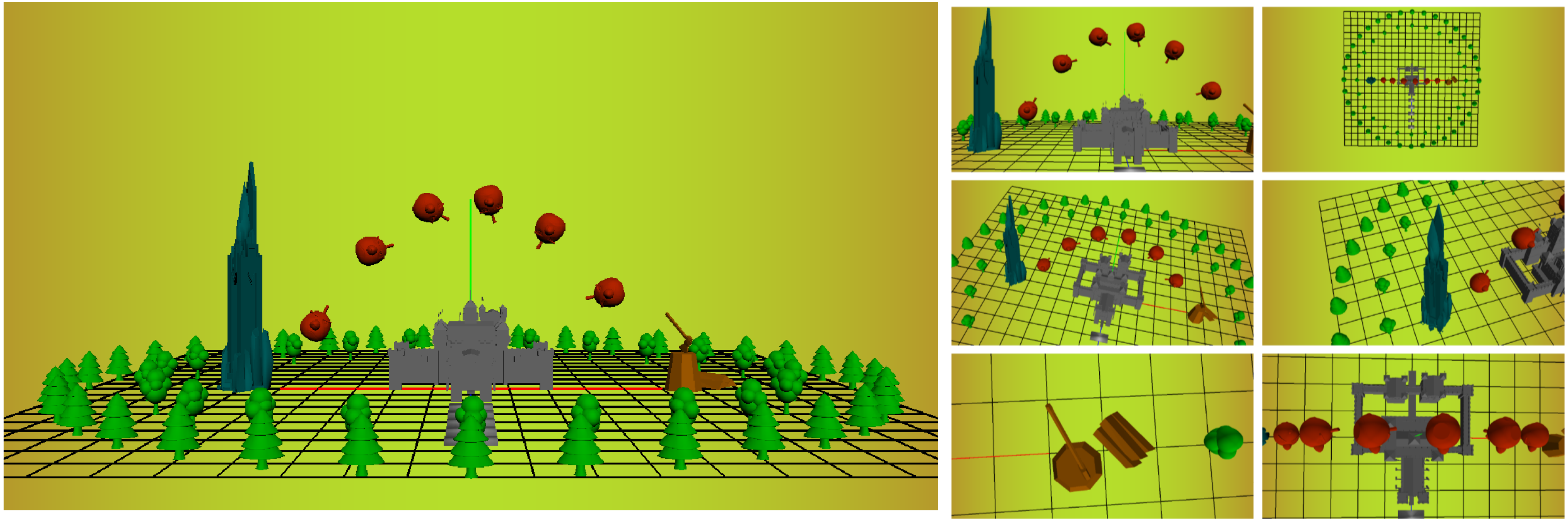
Summary: 3D Linear Transformation

- The column vectors provide a geometric interpretation of any affine matrix

$$X = \begin{bmatrix} \boxed{\begin{matrix} a_x \\ a_y \\ a_z \end{matrix}} & \boxed{\begin{matrix} b_x \\ b_y \\ b_z \end{matrix}} & \begin{matrix} c_x \\ c_y \\ c_z \end{matrix} & \boxed{\begin{matrix} d_x \\ d_y \\ d_z \end{matrix}} \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{a} & \mathbf{b} & \mathbf{c} & \mathbf{d} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



where a, b, c are the coordinate axes, and d is the position/origin of the coordinate system



Live Demo: A3: Angry Bird Palace