

# Linear Algebra Primer

And a video discussion of linear algebra from EE263 is here (lectures 3 and 4):

<https://see.stanford.edu/Course/EE263>

# Outline

- Vectors and matrices
  - Basic Matrix Operations
  - Determinants, norms, trace
  - Special Matrices
- Transformation Matrices
  - Homogeneous coordinates
  - Translation
- Matrix inverse
- Matrix rank
- Eigenvalues and Eigenvectors
- Matrix Calculus

# Outline

- Vectors and matrices
  - Basic Matrix Operations
  - Determinants, norms, trace
  - Special Matrices
- Transformation Matrices
  - Homogeneous coordinates
  - Translation
- Matrix inverse
- Matrix rank
- Eigenvalues and Eigenvectors
- Matrix Calculus



Vectors and matrices are just collections of ordered numbers that represent something: movements in space, scaling factors, pixel brightness, etc. We'll define some common uses and standard operations on them.

# Vector

- A column vector  $\mathbf{v} \in \mathbb{R}^{n \times 1}$  where

$$\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$$

- A row vector  $\mathbf{v}^T \in \mathbb{R}^{1 \times n}$  where

$$\mathbf{v}^T = [v_1 \quad v_2 \quad \dots \quad v_n]$$

$T$  denotes the transpose operation

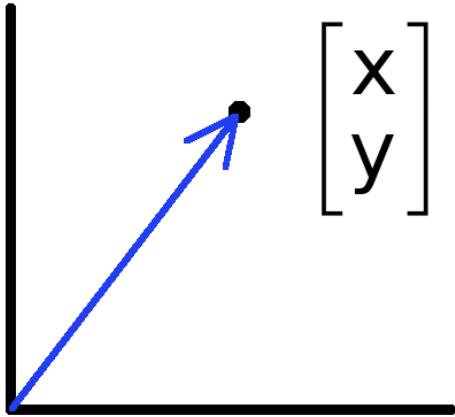
# Vector

- We'll default to column vectors in this class

$$\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$$

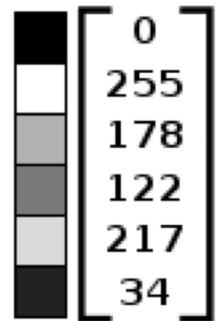
- You'll want to keep track of the orientation of your vectors when programming in python
- You can transpose a vector  $V$  in matlab by writing  $\mathbf{V}'$  (But in class materials, we will **always** use  $V^T$  to indicate transpose, and we will use  $V'$  to mean “ $V$  prime”)

# Vectors have two main uses



- Vectors can represent an offset in 2D or 3D space
- Points are just vectors from the origin

- Data (pixels, gradients at an image keypoint, etc) can also be treated as a vector
- Such vectors don't have a geometric interpretation, but calculations like “distance” can still have value



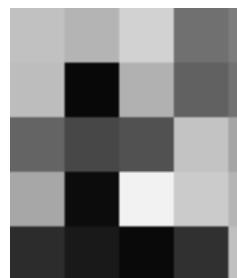
# Matrix

- A matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$  is an array of numbers with size  $m$  by  $n$ , i.e.  $m$  rows and  $n$  columns.

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ \vdots & & & & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \dots & a_{mn} \end{bmatrix}$$

- If  $m = n$ , we say that  $\mathbf{A}$  is square.

# Images



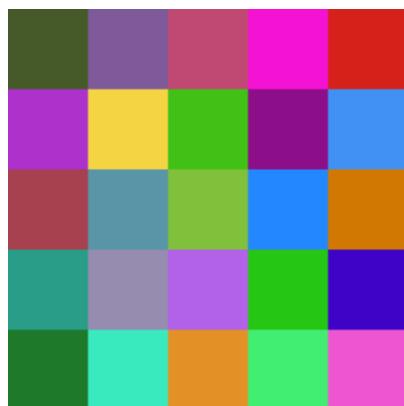
=

$$\begin{bmatrix} 193 & 180 & 210 & 112 & 125 \\ 189 & 8 & 177 & 97 & 114 \\ 100 & 71 & 81 & 195 & 165 \\ 167 & 12 & 242 & 203 & 181 \\ 44 & 25 & 9 & 48 & 192 \end{bmatrix}$$

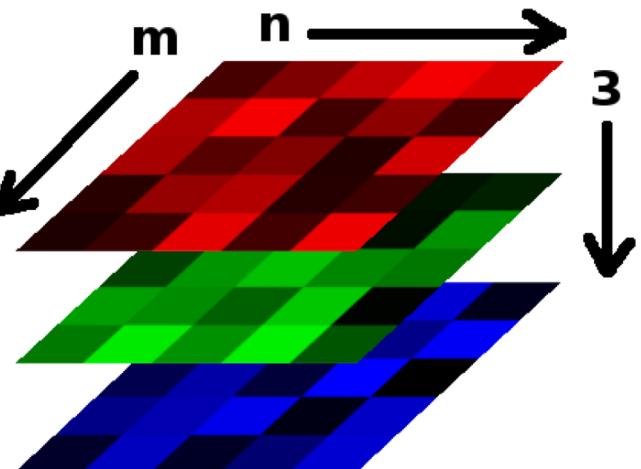
- MATLAB represents an image as a matrix of pixel brightnesses
- Note that the upper left corner is  $(y,x) = (1,1)$

# Color Images

- Grayscale images have one number per pixel, and are stored as an  $m \times n$  matrix.
- Color images have 3 numbers per pixel – red, green, and blue brightnesses (RGB)
- Stored as an  $m \times n \times 3$  matrix



=



# Basic Matrix Operations

- We will discuss:
  - Addition
  - Scaling
  - Dot product
  - Multiplication
  - Transpose
  - Inverse / pseudoinverse
  - Determinant / trace

# Matrix Operations

- Addition

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} + \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} a+1 & b+2 \\ c+3 & d+4 \end{bmatrix}$$

- Can only add a matrix with matching dimensions, or a scalar.

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} + 7 = \begin{bmatrix} a+7 & b+7 \\ c+7 & d+7 \end{bmatrix}$$

- Scaling

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \times 3 = \begin{bmatrix} 3a & 3b \\ 3c & 3d \end{bmatrix}$$

# Vectors

- **Norm**  $\|x\|_2 = \sqrt{\sum_{i=1}^n x_i^2}$
- More formally, a norm is any function  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  that satisfies 4 properties:
  - **Non-negativity:** For all  $x \in \mathbb{R}^n$ ,  $f(x) \geq 0$
  - **Definiteness:**  $f(x) = 0$  if and only if  $x = 0$ .
  - **Homogeneity:** For all  $x \in \mathbb{R}^n$ ,  $t \in \mathbb{R}$ ,  $f(tx) = |t| f(x)$
  - **Triangle inequality:** For all  $x, y \in \mathbb{R}^n$ ,  $f(x + y) \leq f(x) + f(y)$

# Matrix Operations

- **Example Norms**

$$\|x\|_1 = \sum_{i=1}^n |x_i| \quad \|x\|_\infty = \max_i |x_i|.$$

- General  $\ell_p$  norms:

$$\|x\|_p = \left( \sum_{i=1}^n |x_i|^p \right)^{1/p}$$

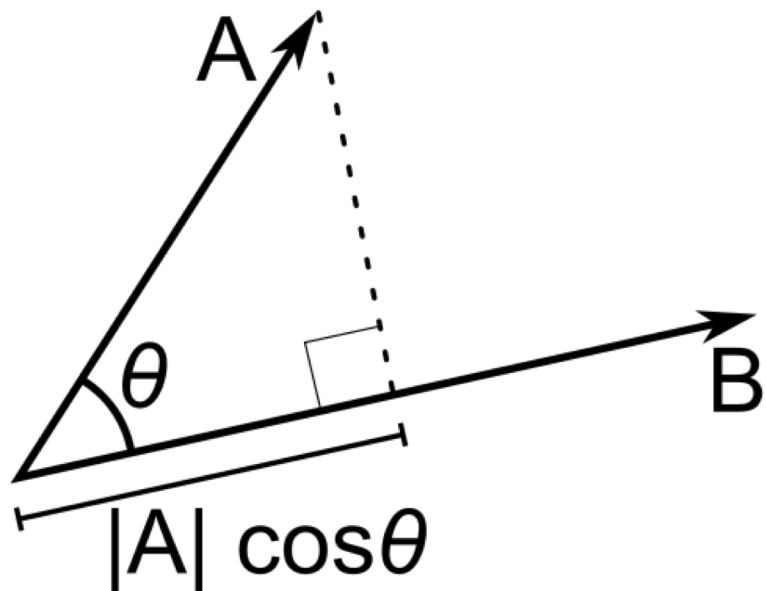
# Matrix Operations

- Inner product (dot product) of vectors
  - Multiply corresponding entries of two vectors and add up the result
  - $\mathbf{x} \cdot \mathbf{y}$  is also  $|\mathbf{x}| |\mathbf{y}| \cos(\text{the angle between } \mathbf{x} \text{ and } \mathbf{y})$

$$\mathbf{x}^T \mathbf{y} = [x_1 \quad \dots \quad x_n] \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} = \sum_{i=1}^n x_i y_i \quad (\text{scalar})$$

# Matrix Operations

- Inner product (dot product) of vectors
  - If  $B$  is a unit vector, then  $A \cdot B$  gives the length of  $A$  which lies in the direction of  $B$



# Matrix Operations

- The product of two matrices

$$C = AB \in \mathbb{R}^{m \times p}$$

$$A \in \mathbb{R}^{m \times n}$$

$$B \in \mathbb{R}^{n \times p}$$

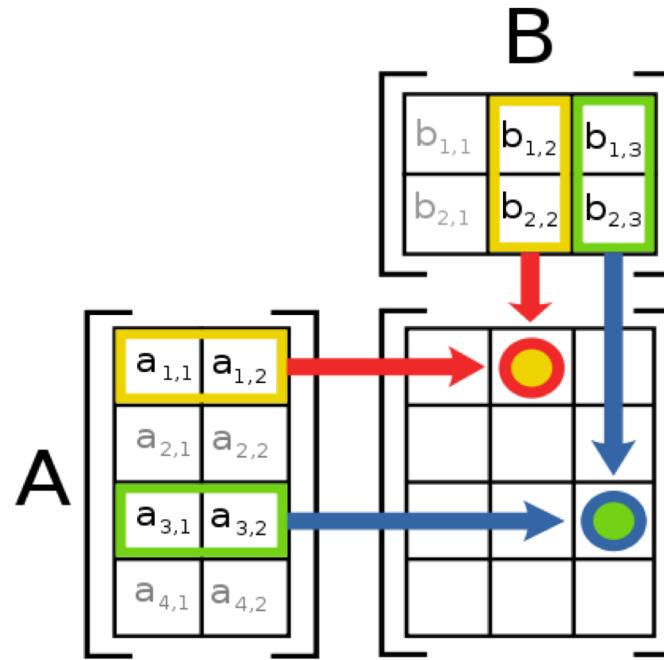
$$C_{ij} = \sum_{k=1}^n A_{ik} B_{kj}$$

$$C = AB = \begin{bmatrix} & a_1^T & \\ & a_2^T & \\ \vdots & & \\ & a_m^T & \end{bmatrix} \begin{bmatrix} | & | & & | \\ b_1 & b_2 & \cdots & b_p \\ | & | & & | \end{bmatrix} = \begin{bmatrix} a_1^T b_1 & a_1^T b_2 & \cdots & a_1^T b_p \\ a_2^T b_1 & a_2^T b_2 & \cdots & a_2^T b_p \\ \vdots & \vdots & \ddots & \vdots \\ a_m^T b_1 & a_m^T b_2 & \cdots & a_m^T b_p \end{bmatrix}.$$

# Matrix Operations

- Multiplication

- The product AB is:



- Each entry in the result is (that row of A) dot product with (that column of B)
- Many uses, which will be covered later

# Matrix Operations

- Multiplication example:

$$\begin{array}{c} A \times B \\ \downarrow \\ \left[ \begin{array}{cc} 0 & 2 \\ 4 & 6 \end{array} \right] \quad \left[ \begin{array}{cc} 1 & 3 \\ 5 & 7 \end{array} \right] \\ \downarrow \\ 0 \cdot 3 + 2 \cdot 7 = 14 \end{array}$$

– Each entry of the matrix product is made by taking the dot product of the corresponding row in the left matrix, with the corresponding column in the right one.

# Matrix Operations

- The product of two matrices

Matrix multiplication is associative:  $(AB)C = A(BC)$ .

Matrix multiplication is distributive:  $A(B + C) = AB + AC$ .

Matrix multiplication is, in general, *not* commutative; that is, it can be the case that  $AB \neq BA$ . (For example, if  $A \in \mathbb{R}^{m \times n}$  and  $B \in \mathbb{R}^{n \times q}$ , the matrix product  $BA$  does not even exist if  $m$  and  $q$  are not equal!)

# Matrix Operations

- Powers
  - By convention, we can refer to the matrix product  $AA$  as  $A^2$ , and  $AAA$  as  $A^3$ , etc.
  - Obviously only square matrices can be multiplied that way

# Matrix Operations

- Transpose – flip matrix, so row 1 becomes column 1

$$\begin{bmatrix} 0 & 1 & \dots \\ \downarrow & \nearrow & \dots \\ \end{bmatrix}$$

$$\begin{bmatrix} 0 & 1 \\ 2 & 3 \\ 4 & 5 \end{bmatrix}^T = \begin{bmatrix} 0 & 2 & 4 \\ 1 & 3 & 5 \end{bmatrix}$$

- A useful identity:

$$(ABC)^T = C^T B^T A^T$$

# Matrix Operations

- Determinant
  - $\det(\mathbf{A})$  returns a scalar
  - Represents area (or volume) of the parallelogram described by the vectors in the rows of the matrix

- For  $\mathbf{A} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ ,  $\det(\mathbf{A}) = ad - bc$

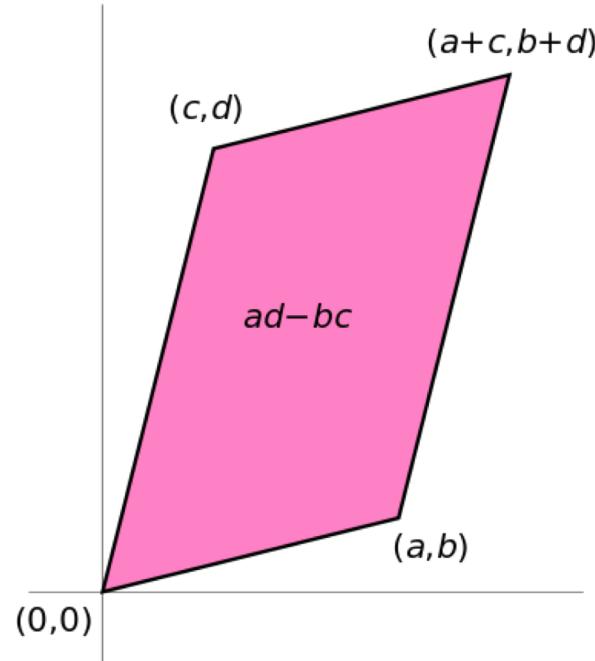
- Properties:  $\det(AB) = \det(A) \det(B)$

$$\det(\mathbf{AB}) = \det(\mathbf{BA})$$

$$\det(\mathbf{A}^{-1}) = \frac{1}{\det(\mathbf{A})}$$

$$\det(\mathbf{A}^T) = \det(\mathbf{A})$$

$$\det(\mathbf{A}) = 0 \Leftrightarrow \mathbf{A} \text{ is singular}$$



# Matrix Operations

- **Trace**

$\text{tr}(\mathbf{A})$  = sum of diagonal elements

$$\text{tr}\left(\begin{bmatrix} 1 & 3 \\ 5 & 7 \end{bmatrix}\right) = 1 + 7 = 8$$

- Invariant to a lot of transformations, so it's used sometimes in proofs. (Rarely in this class though.)
- Properties:

$$\text{tr}(\mathbf{AB}) = \text{tr}(\mathbf{BA})$$

$$\text{tr}(\mathbf{A} + \mathbf{B}) = \text{tr}(\mathbf{A}) + \text{tr}(\mathbf{B})$$

# Matrix Operations

- **Vector Norms**

$$\|x\|_1 = \sum_{i=1}^n |x_i| \quad \|x\|_\infty = \max_i |x_i|.$$

$$\|x\|_2 = \sqrt{\sum_{i=1}^n x_i^2}. \quad \|x\|_p = \left( \sum_{i=1}^n |x_i|^p \right)^{1/p}$$

- Matrix norms: Norms can also be defined for matrices, such as

$$\|A\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n A_{ij}^2} = \sqrt{\text{tr}(A^T A)}.$$

# Special Matrices

- Identity matrix  $\mathbf{I}$ 
    - Square matrix, 1's along diagonal, 0's elsewhere
    - $\mathbf{I} \cdot [\text{another matrix}] = [\text{that matrix}]$
  - Diagonal matrix
    - Square matrix with numbers along diagonal, 0's elsewhere
    - A diagonal  $\cdot$  [another matrix] scales the rows of that matrix
- $$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$
- $$\begin{bmatrix} 3 & 0 & 0 \\ 0 & 7 & 0 \\ 0 & 0 & 2.5 \end{bmatrix}$$

# Special Matrices

- Symmetric matrix

$$\mathbf{A}^T = \mathbf{A}$$

$$\begin{bmatrix} 1 & 2 & 5 \\ 2 & 1 & 7 \\ 5 & 7 & 1 \end{bmatrix}$$

- Skew-symmetric matrix

$$\mathbf{A}^T = -\mathbf{A}$$

$$\begin{bmatrix} 0 & -2 & -5 \\ 2 & 0 & -7 \\ 5 & 7 & 0 \end{bmatrix}$$

# Outline

- Vectors and matrices
  - Basic Matrix Operations
  - Determinants, norms, trace
  - Special Matrices
- Transformation Matrices
  - Homogeneous coordinates
  - Translation
- Matrix inverse
- Matrix rank
- Eigenvalues and Eigenvectors
- Matrix Calculus

Matrix multiplication can be used to transform vectors. A matrix used in this way is called a transformation matrix.

# Transformation

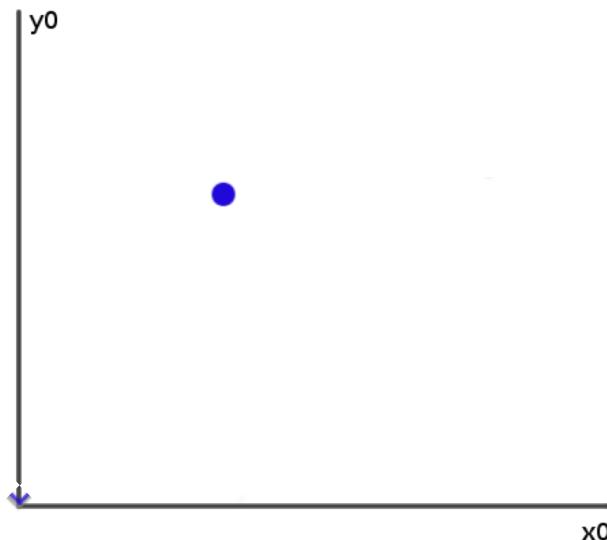
- Matrices can be used to transform vectors in useful ways, through multiplication:  $x' = Ax$
- Simplest is scaling:

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

(Verify to yourself that the matrix multiplication works out this way)

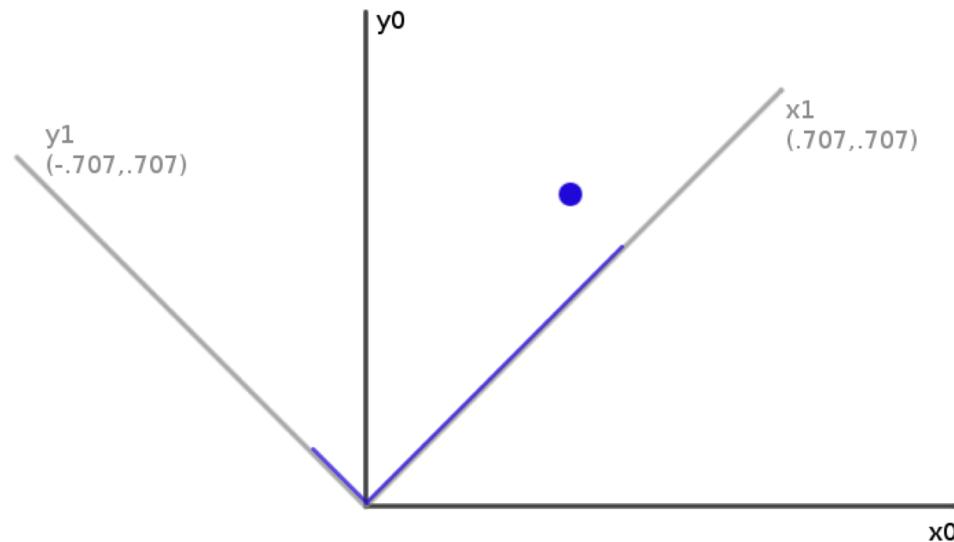
# Rotation

- How can you convert a vector represented in frame “0” to a new, rotated coordinate frame “1”?



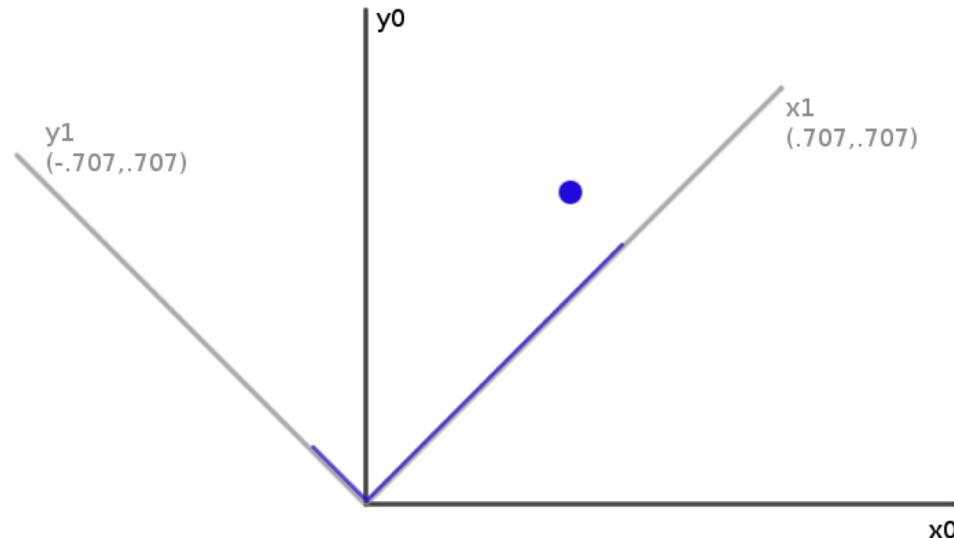
# Rotation

- How can you convert a vector represented in frame “0” to a new, rotated coordinate frame “1”?
- Remember what a vector is:  
[component in direction of the frame’s x axis, component in direction of y axis]



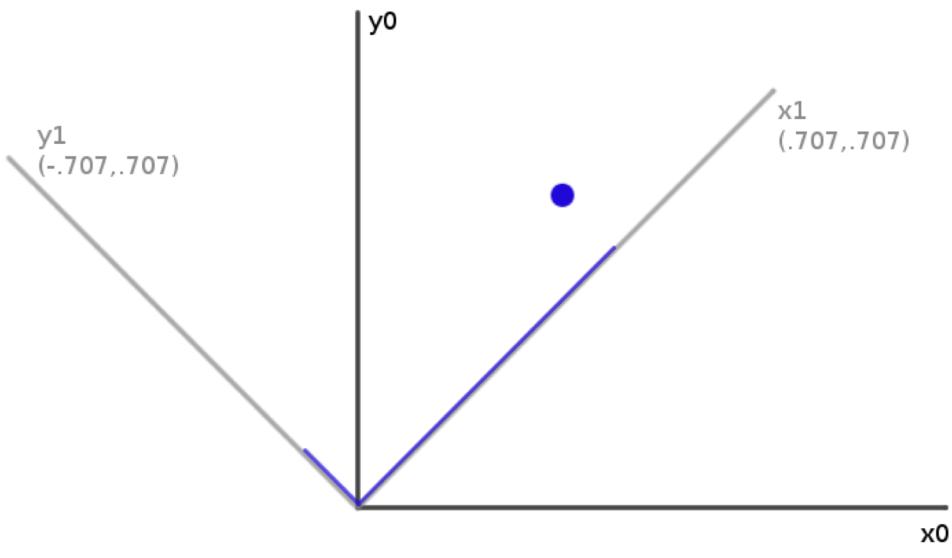
# Rotation

- So to rotate it we must produce this vector:  
[component in direction of **new x axis**, component in direction of **new y axis**]
- We can do this easily with dot products!
- New x coordinate is [original vector] **dot** [the new x axis]
- New y coordinate is [original vector] **dot** [the new y axis]



# Rotation

- Insight: this is what happens in a matrix\*vector multiplication
  - Result x coordinate is:  
[original vector] **dot** [matrix row 1]
  - So matrix multiplication can rotate a vector p:

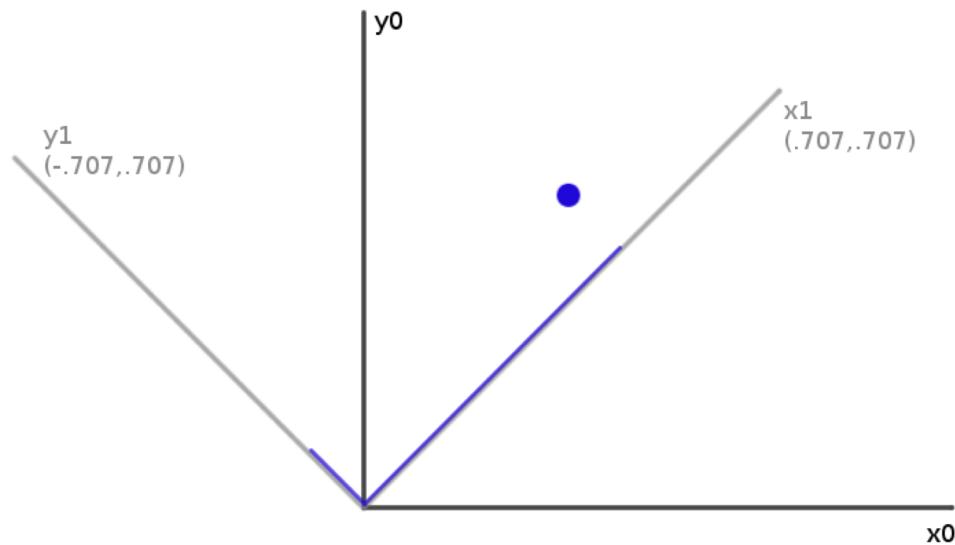


$R \times p =$   
rotated  $p'$

$$\begin{bmatrix} R \\ .707 & .707 \\ -.707 & .707 \end{bmatrix} \begin{bmatrix} p \\ px \\ py \end{bmatrix} = \begin{bmatrix} p' \\ px' \\ py' \end{bmatrix}$$

# Rotation

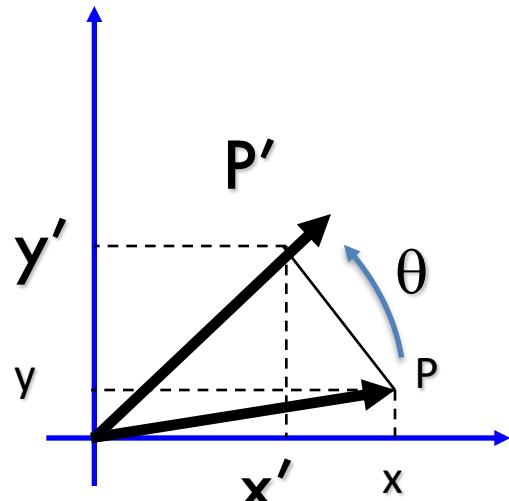
- Suppose we express a point in the new coordinate system which is rotated left
- If we plot the result in the **original** coordinate system, we have rotated the point right



- Thus, rotation matrices can be used to rotate vectors. We'll usually think of them in that sense-- as operators to rotate vectors

# 2D Rotation Matrix Formula

Counter-clockwise rotation by an angle  $\theta$



$$x' = \cos \theta x - \sin \theta y$$

$$y' = \cos \theta y + \sin \theta x$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\mathbf{P}' = \mathbf{R} \mathbf{P}$$

# Transformation Matrices

- Multiple transformation matrices can be used to transform a point:

$$p' = R_2 R_1 S p$$

# Transformation Matrices

- Multiple transformation matrices can be used to transform a point:  
 $p' = R_2 R_1 S p$
- The effect of this is to apply their transformations one after the other, from **right to left**.
- In the example above, the result is  
 $(R_2 (R_1 (S p)))$

# Transformation Matrices

- Multiple transformation matrices can be used to transform a point:  
 $p' = R_2 R_1 S p$
- The effect of this is to apply their transformations one after the other, from **right to left**.
- In the example above, the result is  
 $(R_2 (R_1 (S p)))$
- The result is exactly the same if we multiply the matrices first, to form a single transformation matrix:  
 $p' = (R_2 R_1 S) p$

# Homogeneous system

- In general, a matrix multiplication lets us linearly combine components of a vector

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \times \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} ax + by \\ cx + dy \end{bmatrix}$$

- This is sufficient for scale, rotate, skew transformations.
- But notice, we can't add a constant! ☹

# Homogeneous system

- The (somewhat hacky) solution? Stick a “1” at the end of every vector:

$$\begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} ax + by + c \\ dx + ey + f \\ 1 \end{bmatrix}$$

- Now we can rotate, scale, and skew like before, **AND translate** (note how the multiplication works out, above)
- This is called “homogeneous coordinates”

# Homogeneous system

- In homogeneous coordinates, the multiplication works out so the rightmost column of the matrix is a vector that gets added.

$$\begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} ax + by + c \\ dx + ey + f \\ 1 \end{bmatrix}$$

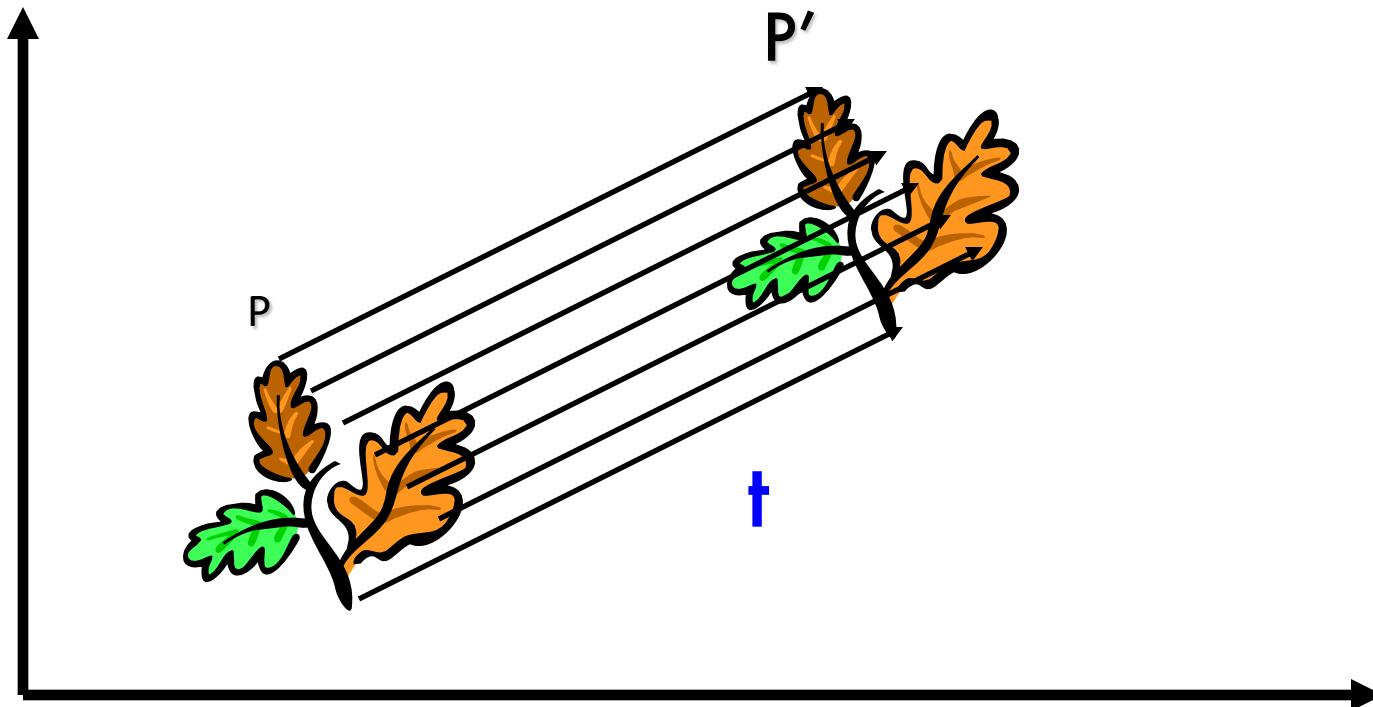
- Generally, a homogeneous transformation matrix will have a bottom row of [0 0 1], so that the result has a “1” at the bottom too.

# Homogeneous system

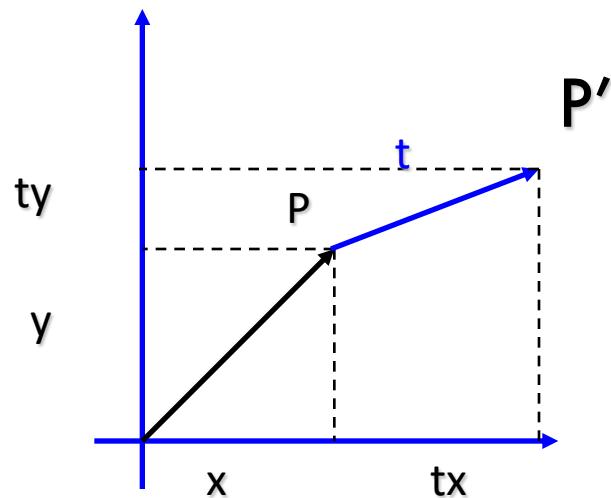
- One more thing we might want: to divide the result by something
  - Matrix multiplication can't actually divide
  - So, **by convention**, in homogeneous coordinates, we'll divide the result by its last coordinate after doing a matrix multiplication

$$\begin{bmatrix} x \\ y \\ 7 \end{bmatrix} \Rightarrow \begin{bmatrix} x/7 \\ y/7 \\ 1 \end{bmatrix}$$

# 2D Translation



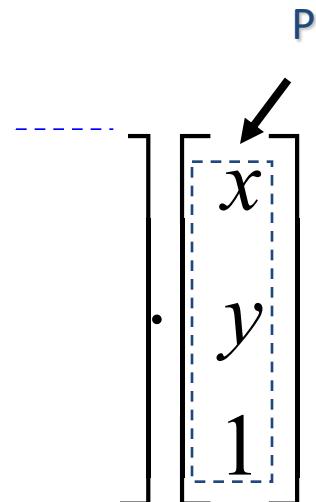
# 2D Translation using Homogeneous Coordinates



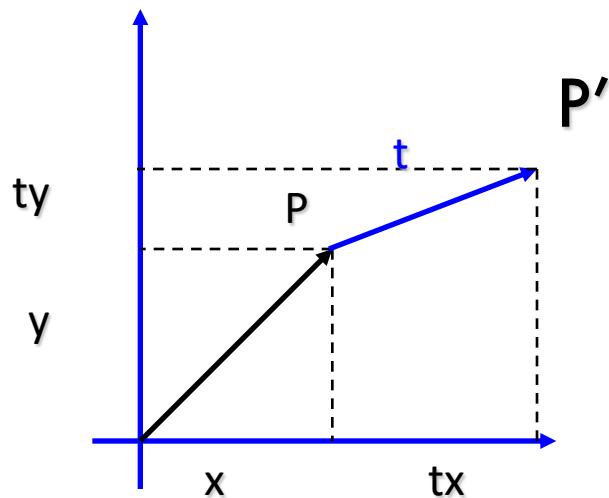
$$P = (x, y) \rightarrow (x, y, 1)$$

$$t = (t_x, t_y) \rightarrow (t_x, t_y, 1)$$

$$P' \rightarrow \begin{bmatrix} x + t_x \\ y + t_y \\ 1 \end{bmatrix} = \quad \cdot \quad \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$



# 2D Translation using Homogeneous Coordinates

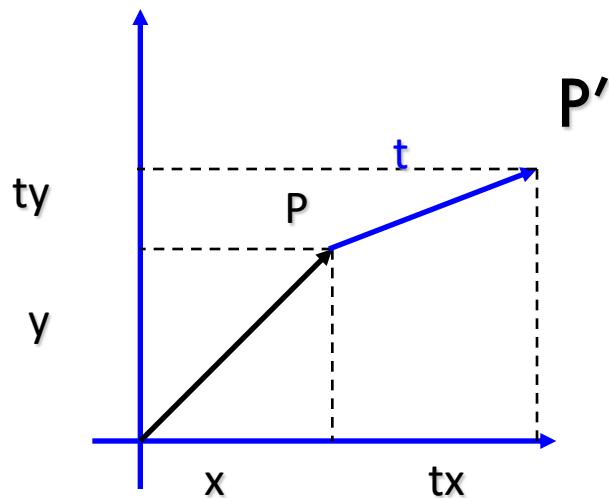


$$\mathbf{P} = (x, y) \rightarrow (x, y, 1)$$

$$\mathbf{t} = (t_x, t_y) \rightarrow (t_x, t_y, 1)$$

$$\mathbf{P}' \rightarrow \begin{bmatrix} x + t_x \\ y + t_y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

# 2D Translation using Homogeneous Coordinates

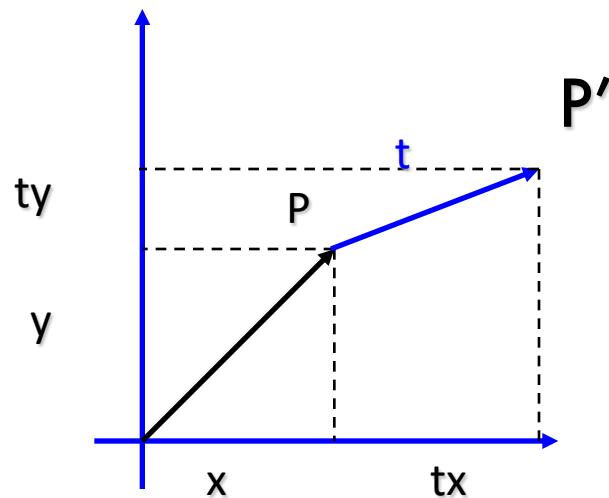


$$P = (x, y) \rightarrow (x, y, 1)$$

$$t = (t_x, t_y) \rightarrow (t_x, t_y, 1)$$

$$P' \rightarrow \begin{bmatrix} x + t_x \\ y + t_y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & | \\ & 1 & \\ & & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

# 2D Translation using Homogeneous Coordinates

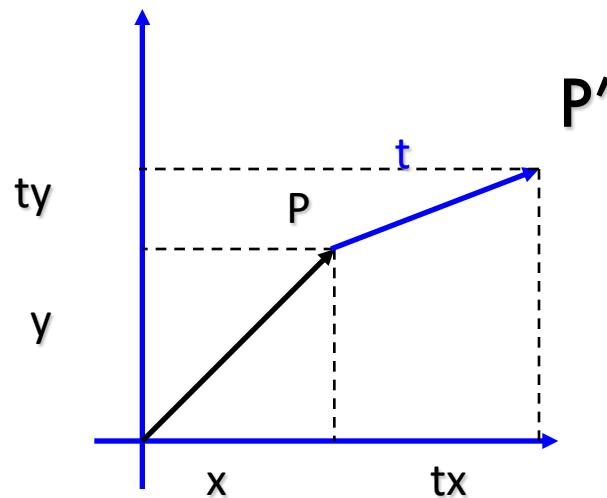


$$P = (x, y) \rightarrow (x, y, 1)$$

$$t = (t_x, t_y) \rightarrow (t_x, t_y, 1)$$

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

# 2D Translation using Homogeneous Coordinates

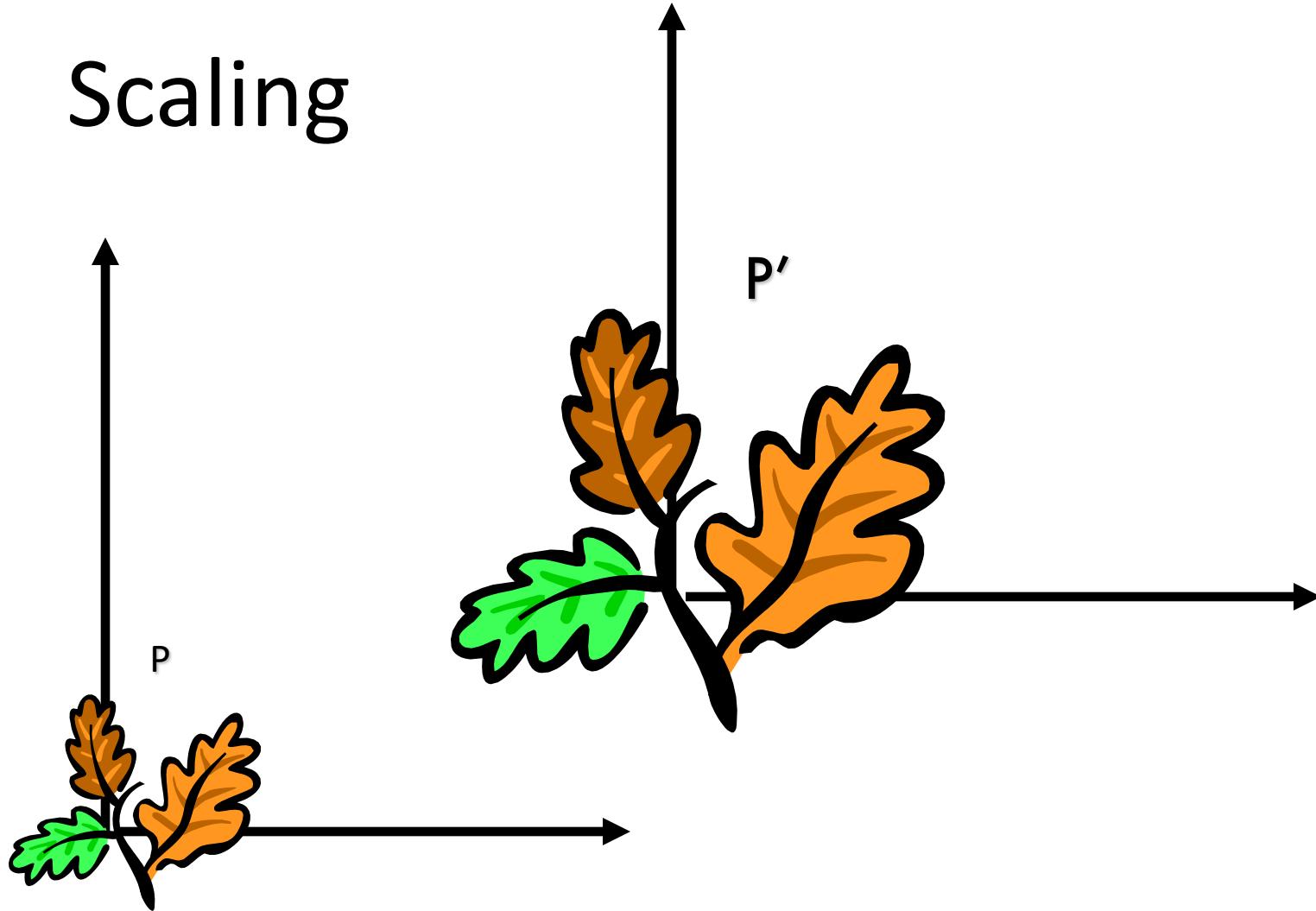


$$P = (x, y) \rightarrow (x, y, 1)$$

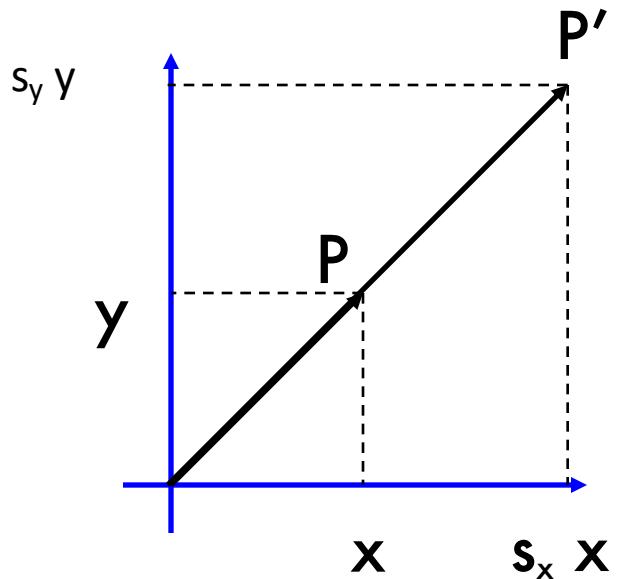
$$t = (t_x, t_y) \rightarrow (t_x, t_y, 1)$$

$$\begin{aligned} P' &\rightarrow \begin{bmatrix} x + t_x \\ y + t_y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} I & t \\ 0 & 1 \end{bmatrix} \cdot P = T \cdot P \end{aligned}$$

# Scaling



# Scaling Equation

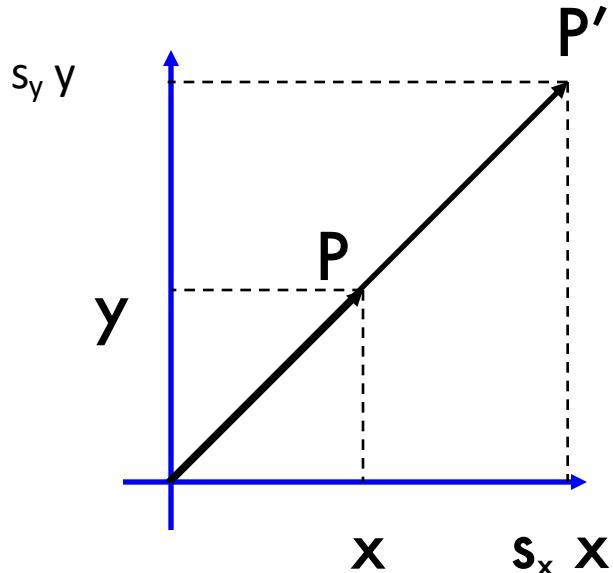


$$\mathbf{P} = (x, y) \rightarrow \mathbf{P}' = (s_x x, s_y y)$$

$$\mathbf{P} = (x, y) \rightarrow (x, y, 1)$$

$$\mathbf{P}' = (s_x x, s_y y) \rightarrow (s_x x, s_y y, 1)$$

# Scaling Equation



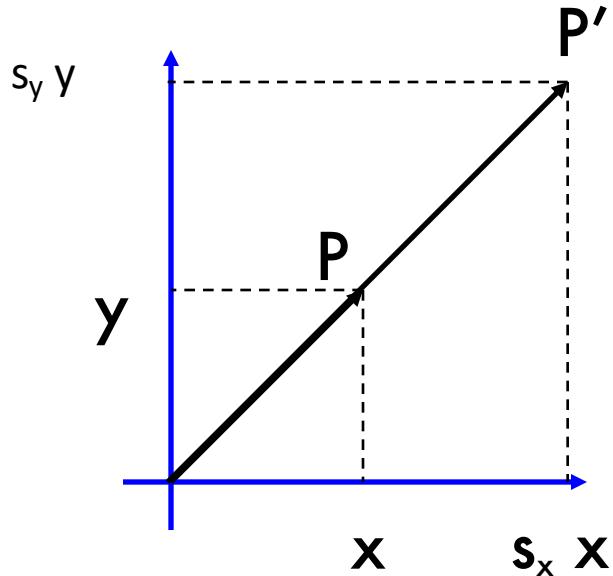
$$\mathbf{P} = (x, y) \rightarrow \mathbf{P}' = (s_x x, s_y y)$$

$$\mathbf{P} = (x, y) \rightarrow (x, y, 1)$$

$$\mathbf{P}' = (s_x x, s_y y) \rightarrow (s_x x, s_y y, 1)$$

$$\mathbf{P}' \rightarrow \begin{bmatrix} s_x x \\ s_y y \\ 1 \end{bmatrix} = \begin{bmatrix} & & \\ & & \\ & & \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

# Scaling Equation



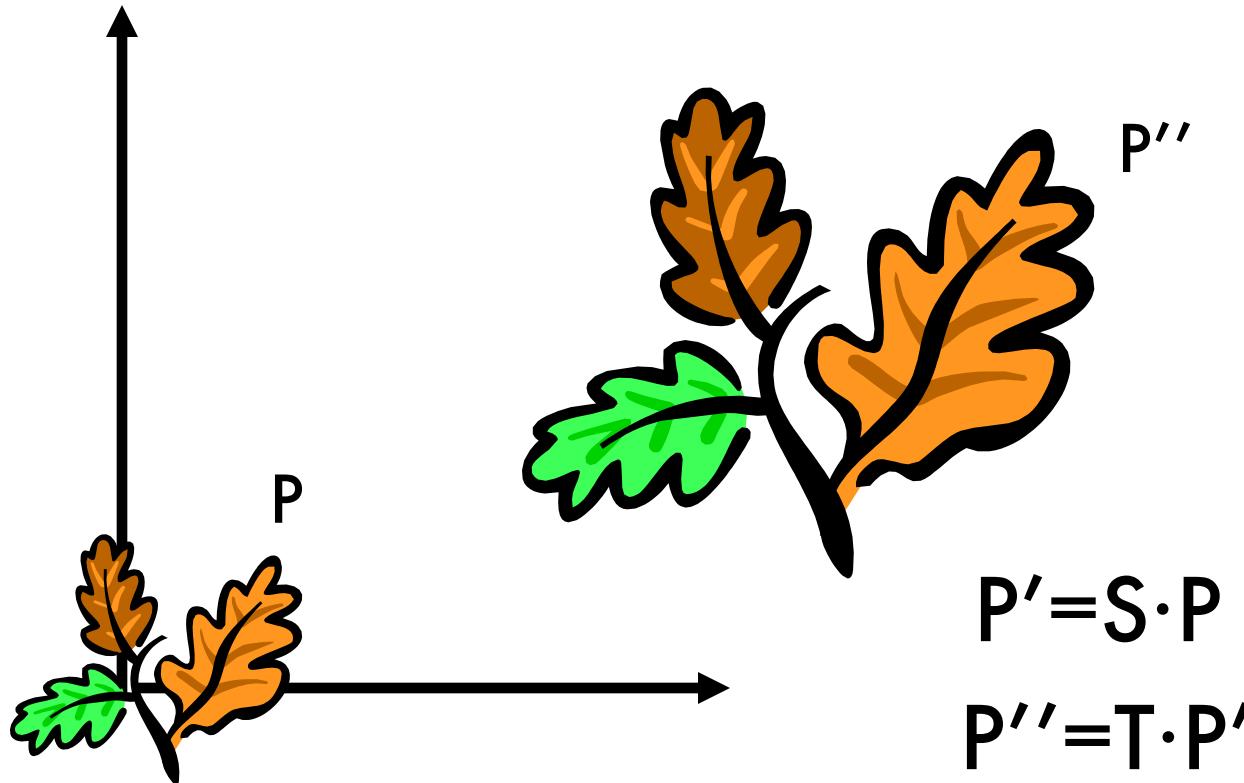
$$\mathbf{P} = (x, y) \rightarrow \mathbf{P}' = (s_x x, s_y y)$$

$$\mathbf{P} = (x, y) \rightarrow (x, y, 1)$$

$$\mathbf{P}' = (s_x x, s_y y) \rightarrow (s_x x, s_y y, 1)$$

$$\mathbf{P}' \rightarrow \begin{bmatrix} s_x x \\ s_y y \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{S}} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{S}' & 0 \\ 0 & 1 \end{bmatrix} \cdot \mathbf{P} = \mathbf{S} \cdot \mathbf{P}$$

# Scaling & Translating



$$P'' = T \cdot P' = T \cdot (S \cdot P) = T \cdot S \cdot P$$

# Scaling & Translating

$$\mathbf{P}'' = \mathbf{T} \cdot \mathbf{S} \cdot \mathbf{P} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

# Scaling & Translating

$$\mathbf{P}'' = \mathbf{T} \cdot \mathbf{S} \cdot \mathbf{P} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} =$$

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

# Translating & Scaling versus Scaling & Translating

$$\mathbf{P}''' = \mathbf{T} \cdot \mathbf{S} \cdot \mathbf{P} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & t_x \\ 0 & s_y & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} s_x x + t_x \\ s_y y + t_y \\ 1 \end{bmatrix}$$

# Translating & Scaling != Scaling & Translating

$$\mathbf{P}''' = \mathbf{T} \cdot \mathbf{S} \cdot \mathbf{P} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & t_x \\ 0 & s_y & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} s_x x + t_x \\ s_y y + t_y \\ 1 \end{bmatrix}$$

$$\mathbf{P}''' = \mathbf{S} \cdot \mathbf{T} \cdot \mathbf{P} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} =$$

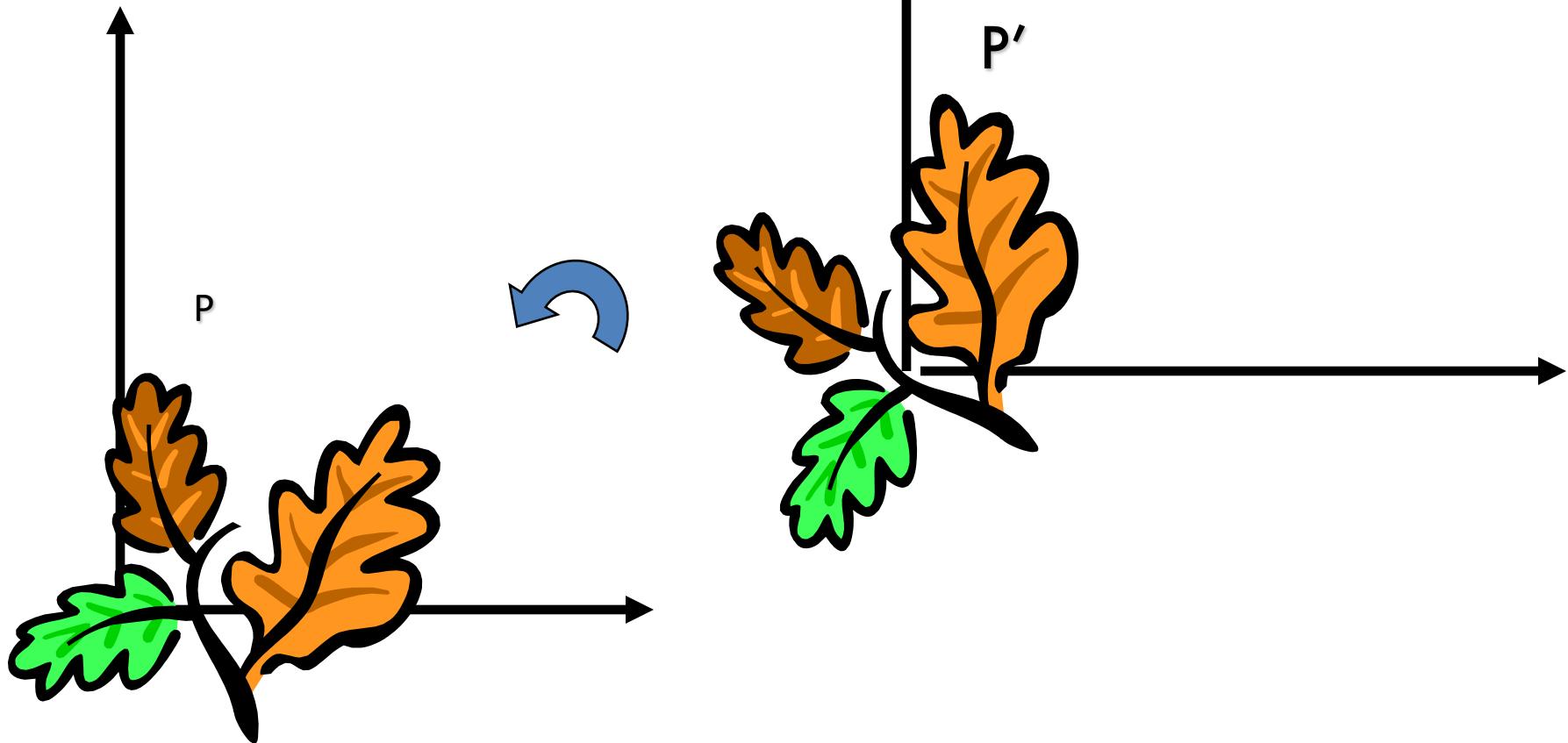
# Translating & Scaling != Scaling & Translating

$$\mathbf{P}''' = \mathbf{T} \cdot \mathbf{S} \cdot \mathbf{P} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & t_x \\ 0 & s_y & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} s_x x + t_x \\ s_y y + t_y \\ 1 \end{bmatrix}$$

$$\mathbf{P}''' = \mathbf{S} \cdot \mathbf{T} \cdot \mathbf{P} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} =$$

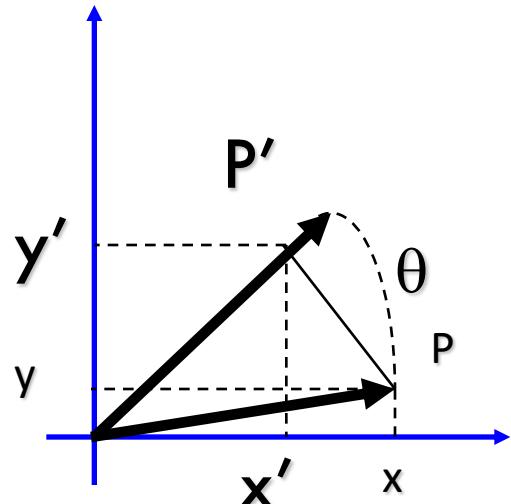
$$= \begin{bmatrix} s_x & 0 & s_x t_x \\ 0 & s_y & s_y t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} s_x x + s_x t_x \\ s_y y + s_y t_y \\ 1 \end{bmatrix}$$

# Rotation



# Rotation Equations

Counter-clockwise rotation by an angle  $\theta$



$$x' = \cos \theta x - \sin \theta y$$

$$y' = \cos \theta y + \sin \theta x$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\mathbf{P}' = \mathbf{R} \mathbf{P}$$

# Rotation Matrix Properties

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

A 2D rotation matrix is 2x2

Note: R belongs to the category of *normal* matrices and satisfies many interesting properties:

$$\mathbf{R} \cdot \mathbf{R}^T = \mathbf{R}^T \cdot \mathbf{R} = \mathbf{I}$$

$$\det(\mathbf{R}) = 1$$

# Rotation Matrix Properties

- Transpose of a rotation matrix produces a rotation in the opposite direction

$$\mathbf{R} \cdot \mathbf{R}^T = \mathbf{R}^T \cdot \mathbf{R} = \mathbf{I}$$

$$\det(\mathbf{R}) = 1$$

- The rows of a rotation matrix are always mutually perpendicular (a.k.a. orthogonal) unit vectors
  - (and so are its columns)

# Scaling + Rotation + Translation

$$\mathbf{P}' = (\mathbf{T} \mathbf{R} \mathbf{S}) \mathbf{P}$$

$$\mathbf{P}' = \mathbf{T} \cdot \mathbf{R} \cdot \mathbf{S} \cdot \mathbf{P} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} =$$

$$= \begin{bmatrix} \cos \theta & -\sin \theta & t_x \\ \sin \theta & \cos \theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} =$$

This is the form of the general-purpose transformation matrix

$$= \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} S & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \boxed{\begin{bmatrix} R & S & t \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}}$$

# Announcements

- HW0 will be released this Friday night

# Outline

- Vectors and matrices
  - Basic Matrix Operations
  - Determinants, norms, trace
  - Special Matrices
- Transformation Matrices
  - Homogeneous coordinates
  - Translation
- **Matrix inverse** ←
- Matrix rank
- Eigenvalues and Eigenvectors
- Matrix Calculate

The inverse of a transformation matrix reverses its effect

# Inverse

- Given a matrix  $\mathbf{A}$ , its inverse  $\mathbf{A}^{-1}$  is a matrix such that  $\mathbf{AA}^{-1} = \mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$
- E.g.  $\begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix}^{-1} = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{3} \end{bmatrix}$
- Inverse does not always exist. If  $\mathbf{A}^{-1}$  exists,  $\mathbf{A}$  is *invertible* or *non-singular*. Otherwise, it's *singular*.
- Useful identities, for matrices that are invertible:

$$(\mathbf{A}^{-1})^{-1} = \mathbf{A}$$

$$(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$$

$$\mathbf{A}^{-T} \triangleq (\mathbf{A}^T)^{-1} = (\mathbf{A}^{-1})^T$$

# Outline

- Vectors and matrices
  - Basic Matrix Operations
  - Determinants, norms, trace
  - Special Matrices
- Transformation Matrices
  - Homogeneous coordinates
  - Translation
- Matrix inverse
- **Matrix rank** 
- Eigenvalues and Eigenvectors
- Matrix Calculate

The rank of a transformation matrix tells you how many dimensions it transforms a vector to.

# Linear independence

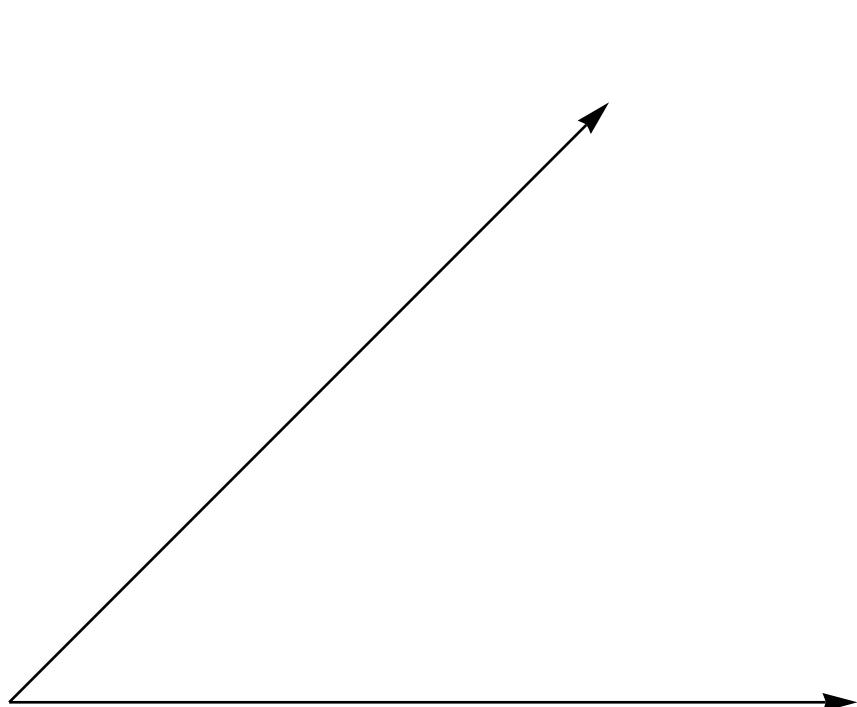
- Suppose we have a set of vectors  $\mathbf{v}_1, \dots, \mathbf{v}_n$
- If we can express  $\mathbf{v}_1$  as a linear combination of the other vectors  $\mathbf{v}_2 \dots \mathbf{v}_n$ , then  $\mathbf{v}_1$  is linearly *dependent* on the other vectors.
  - The direction  $\mathbf{v}_1$  can be expressed as a combination of the directions  $\mathbf{v}_2 \dots \mathbf{v}_n$ . (E.g.  $\mathbf{v}_1 = .7 \mathbf{v}_2 - .7 \mathbf{v}_4$ )

# Linear independence

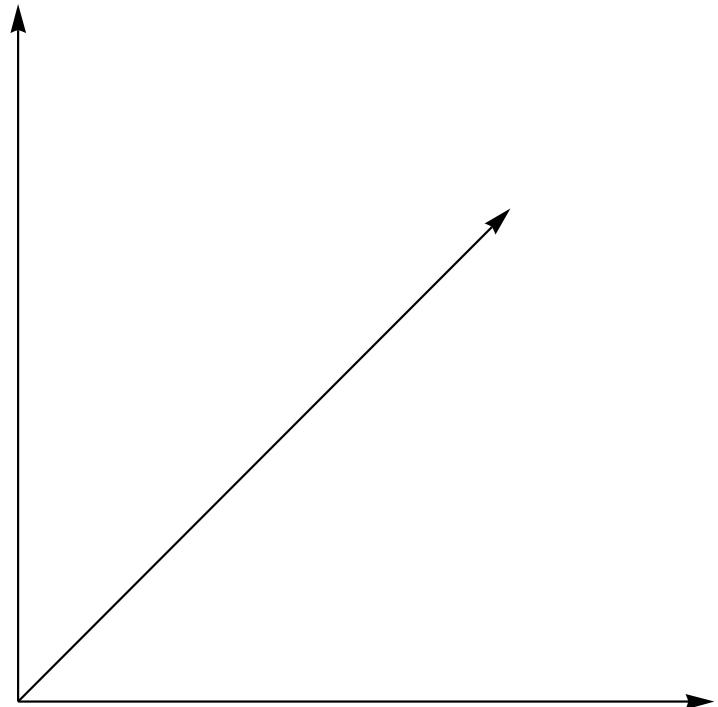
- Suppose we have a set of vectors  $v_1, \dots, v_n$
- If we can express  $v_1$  as a linear combination of the other vectors  $v_2 \dots v_n$ , then  $v_1$  is linearly *dependent* on the other vectors.
  - The direction  $v_1$  can be expressed as a combination of the directions  $v_2 \dots v_n$ . (E.g.  $v_1 = .7 v_2 - .7 v_4$ )
- If no vector is linearly dependent on the rest of the set, the set is linearly *independent*.
  - Common case: a set of vectors  $v_1, \dots, v_n$  is always linearly independent if each vector is perpendicular to every other vector (and non-zero)

# Linear independence

Linearly independent set



Not linearly independent



# Matrix rank

- **Column/row rank**

$\text{col-rank}(\mathbf{A})$  = the maximum number of linearly independent column vectors of  $\mathbf{A}$

$\text{row-rank}(\mathbf{A})$  = the maximum number of linearly independent row vectors of  $\mathbf{A}$

- Column rank always equals row rank

- **Matrix rank**

$$\text{rank}(\mathbf{A}) \triangleq \text{col-rank}(\mathbf{A}) = \text{row-rank}(\mathbf{A})$$

# Matrix rank

- For transformation matrices, the rank tells you the dimensions of the output
- E.g. if rank of  $\mathbf{A}$  is 1, then the transformation

$$\mathbf{p}' = \mathbf{Ap}$$

maps points onto a line.

- Here's a matrix with rank 1:

$$\begin{bmatrix} 1 & 1 \\ 2 & 2 \end{bmatrix} \times \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x + y \\ 2x + 2y \end{bmatrix} \quad \text{← All points get mapped to the line } y=2x$$

# Matrix rank

- If an  $m \times m$  matrix is rank  $m$ , we say it's "full rank"
  - Maps an  $m \times 1$  vector uniquely to another  $m \times 1$  vector
  - An inverse matrix can be found
- If rank  $< m$ , we say it's "singular"
  - At least one dimension is getting collapsed. No way to look at the result and tell what the input was
  - Inverse does not exist
- Inverse also doesn't exist for non-square matrices