Linear Algebra Primer

Slides by Juan Carlos Niebles*, Ranjay Krishna* and Maxime Voisin
*Stanford Vision and Learning Lab

Another, very in-depth linear algebra review from CS229 is available here:
http://cs229.stanford.edu/section/cs229-linalg.pdf
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: location in space, speed, pixel brightness, etc. We'll define some common uses and standard operations on them.

$$\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} \qquad \mathbf{v}^T = \begin{bmatrix} v_1 & v_2 & \dots & v_n \end{bmatrix}$$

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

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

$$\mathbf{v}^T = \begin{bmatrix} v_1 & v_2 & \dots & v_n \end{bmatrix}$$

T denotes the transpose operation

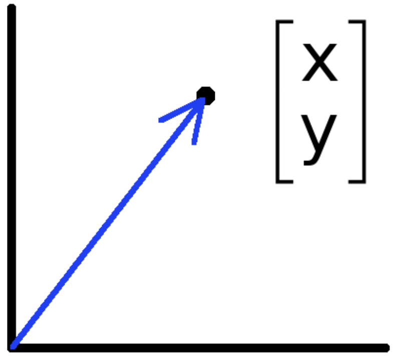
Vector

- We'll default to column vectors in this class

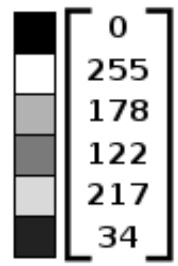
$$\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$$

- You'll want to keep track of the orientation of your vectors when programming in python

Some vectors have a geometric interpretation, others don’t…



- Some vectors have a geometric interpretation:
  - Points are just vectors from the origin.
  - We can make calculations like “distance” between 2 vectors



- Other vectors don’t have a geometric interpretation:
  - Vectors can represent any kind of data (pixels, gradients at an image keypoint, etc)
  - Such vectors don’t have a geometric interpretation
  - We can still make calculations like “distance” between 2 vectors

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

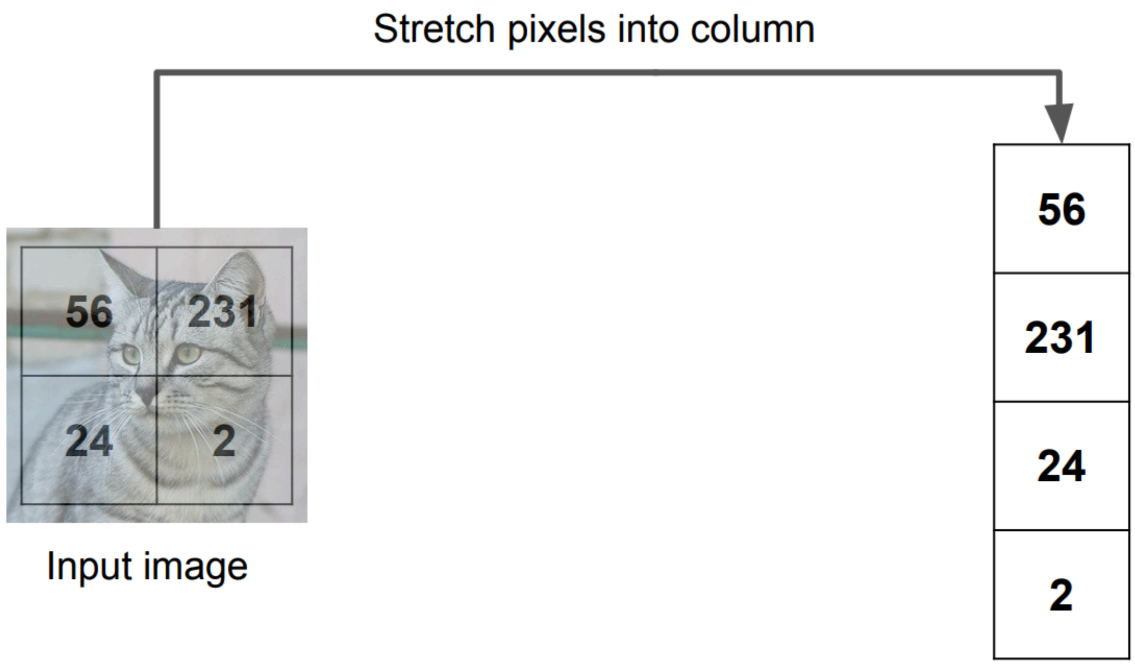
$$= \begin{bmatrix} 193 & 180 & 210 & 112 & 125 \\ 189 & 8 & 177 & 97 & 114 \\ 100 & 71 & 81 & 195 & 165 \\ 167 & 12 & 242 & 203 & 181 \\ 44 & 25 & 9 & 48 & 192 \end{bmatrix}$$

- Python represents an image as a matrix of pixel brightnesses
- Note that the upper left corner is [x, y] = (0,0)

row column

Python indices start at 0

Images can be represented as a **matrix** of pixels.
Images can also be represented as a **vector** of pixels

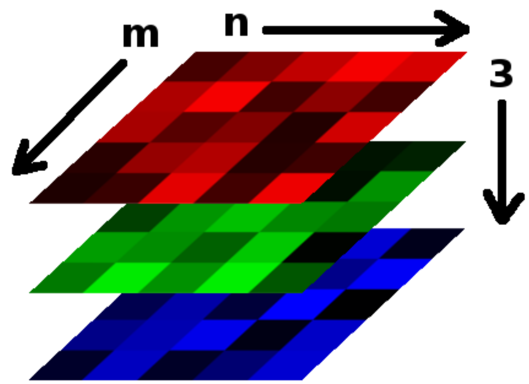


Color Images

- **Grayscale images** have **1** number per pixel, and are stored as an **m×n matrix**.

- **Color images** have **3** numbers per pixel – red, green, and blue brightnesses (RGB) - and are stored as an **m×n×3 matrix**

=



Basic Matrix Operations

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

Matrix Operations

- Addition

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

  – We can only add a matrix with matching dimensions, or a scalar. Good to know for Python assignments ☺

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

- Scaling

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

Vector Norms

- **Examples of vector norms**

$$\|x\|_1 = \sum_{i=1}^{n} |x_i| \qquad \|x\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2}. \qquad \|x\|_\infty = \max_i |x_i|.$$

Vector Norms

- More formally, a norm is any function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ that satisfies the following 4 properties:

- **Non-negativity:** For all $x \in \mathbb{R}^n,\ f(x) \geq 0$
- **Definiteness**: f(x) = 0 if and only if x = [0,0…0].
- **Homogeneity**: For all $x \in \mathbb{R}^n,\ t \in \mathbb{R},\ f(tx) = |t| f(x)$
- **Triangle inequality**: For all $x, y \in \mathbb{R}^n,\ f(x + y) \leq f(x) + f(y)$

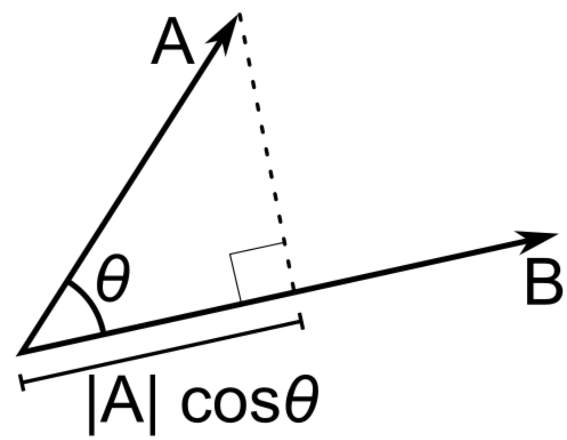
Vector Operation: inner product

- **Inner product** (dot product) **of two vectors**
  - Multiply corresponding entries of two vectors and add up the result
  - x·y is also |x||y|cos(*the angle between x and y*)

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

Vector Operation: inner product

- **Inner product** (dot product) **of two vectors**
  - If B is a **unit** vector:
    - Then $A \cdot B = |A||B|\cos(\Theta) = |A| \times 1 \times \cos(\Theta) = |A|\cos(\Theta)$
    - **A·B gives the length of A which lies in the direction of B**



Matrix Operations

- The **product of two matrices**

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

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

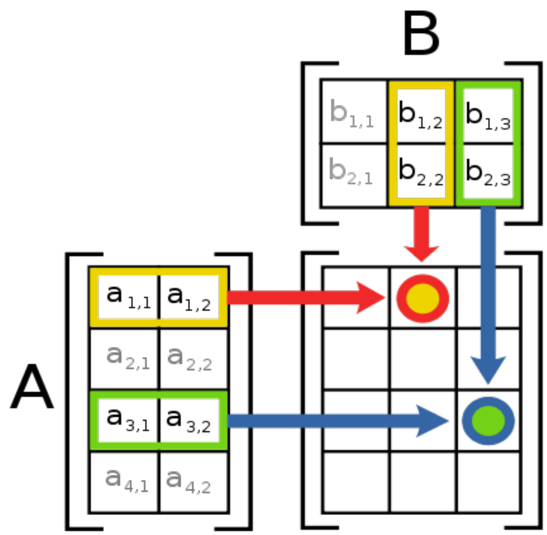
$$C = AB \in \mathbb{R}^{m \times p}$$

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

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

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:

A x B

$$\begin{bmatrix} 1 & 3 \\ 5 & 7 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 2 \\ 4 & 6 \end{bmatrix} \quad \begin{bmatrix} 10 & 14 \\ 34 & 54 \end{bmatrix}$$

$$0 \cdot 3 + 2 \cdot 7 = 14$$

– 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: $A(B + C) = AB + AC$.

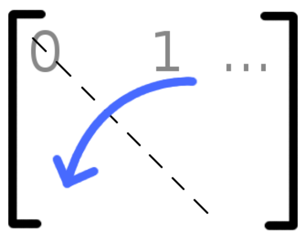
Matrix multiplication is, in general, *not* commutative; that is, it can be the case that $AB \neq BA$. (For example, if $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{n\times q}$, the matrix product BA does not even exist if m and q are not equal!)

Matrix Operations

- Powers
  - By convention, we can refer to the matrix product AA as A^2, and AAA as A^3, etc.
  - **Important: only square matrices can be multiplied that way!** (make sure you understand why)

Matrix Operations

- Transpose a matrix: flip the matrix, so row 1 becomes column 1



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

- A useful identity:

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

Matrix Operations

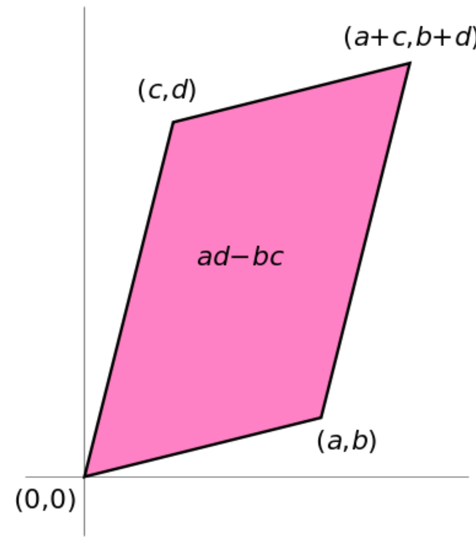
- **Determinant**
  - $\det(\mathbf{A})$ returns a scalar
  - Represents area (or volume) of the parallelogram described by the vectors in the rows of the matrix
  - For $\mathbf{A} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$, $\det(\mathbf{A}) = ad - bc$
  - Properties:

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

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

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

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



Matrix Operations

- **Trace**

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

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

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

  – Properties:

$$\mathrm{tr}(\mathbf{AB}) = \mathrm{tr}(\mathbf{BA})$$
$$\mathrm{tr}(\mathbf{A} + \mathbf{B}) = \mathrm{tr}(\mathbf{A}) + \mathrm{tr}(\mathbf{B})$$

Matrix Operations

- **Vector Norms (we've talked about them earlier)**

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

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

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

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

Special Matrices

- **Identity matrix $\mathbf{I}$**

  Square matrix, 1's along diagonal, 0's elsewhere

  $\mathbf{I} \cdot$ [a matrix A] = [that matrix A]
  [a matrix A] $\cdot$ $\mathbf{I}$ = [that matrix A]

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

- **Diagonal matrix**

  Square matrix with numbers along diagonal, 0's elsewhere

  [diagonal matrix A] $\cdot$ [another matrix B] scales the rows of matrix B

$$\begin{bmatrix} 3 & 0 & 0 \\ 0 & 7 & 0 \\ 0 & 0 & 2.5 \end{bmatrix}$$

Special Matrices

- Symmetric matrix

$$\mathbf{A}^T = \mathbf{A} \qquad \begin{bmatrix} 1 & 2 & 5 \\ 2 & 1 & 7 \\ 5 & 7 & 1 \end{bmatrix}$$

- Skew-symmetric matrix

$$\mathbf{A}^T = -\mathbf{A} \qquad \begin{bmatrix} 0 & -2 & -5 \\ 2 & 0 & -7 \\ 5 & 7 & 0 \end{bmatrix}$$

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: scaling

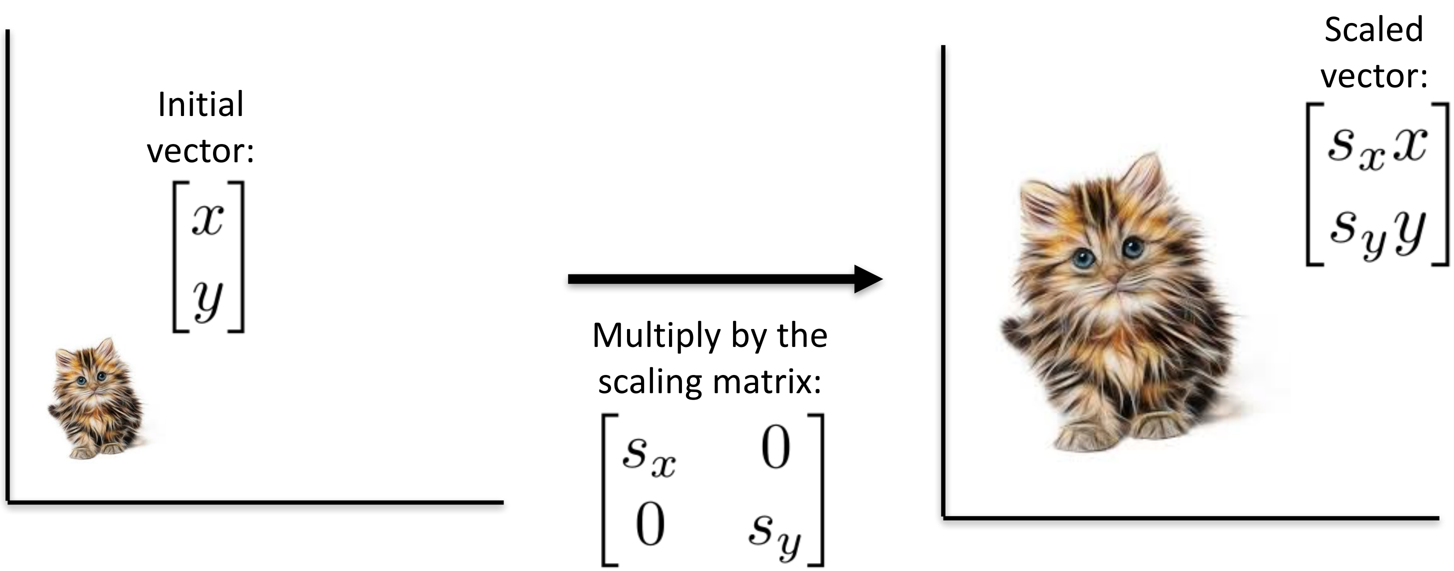
- **Matrices can be used to transform vectors** in useful ways, through multiplication: Ax = x'
- Simplest transformation is **scaling**:

$$\underbrace{\begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix}}_{\text{Scaling matrix}} \times \underbrace{\begin{bmatrix} x \\ y \end{bmatrix}}_{\text{Initial vector}} = \underbrace{\begin{bmatrix} s_x x \\ s_y y \end{bmatrix}}_{\text{Scaled vector}}$$

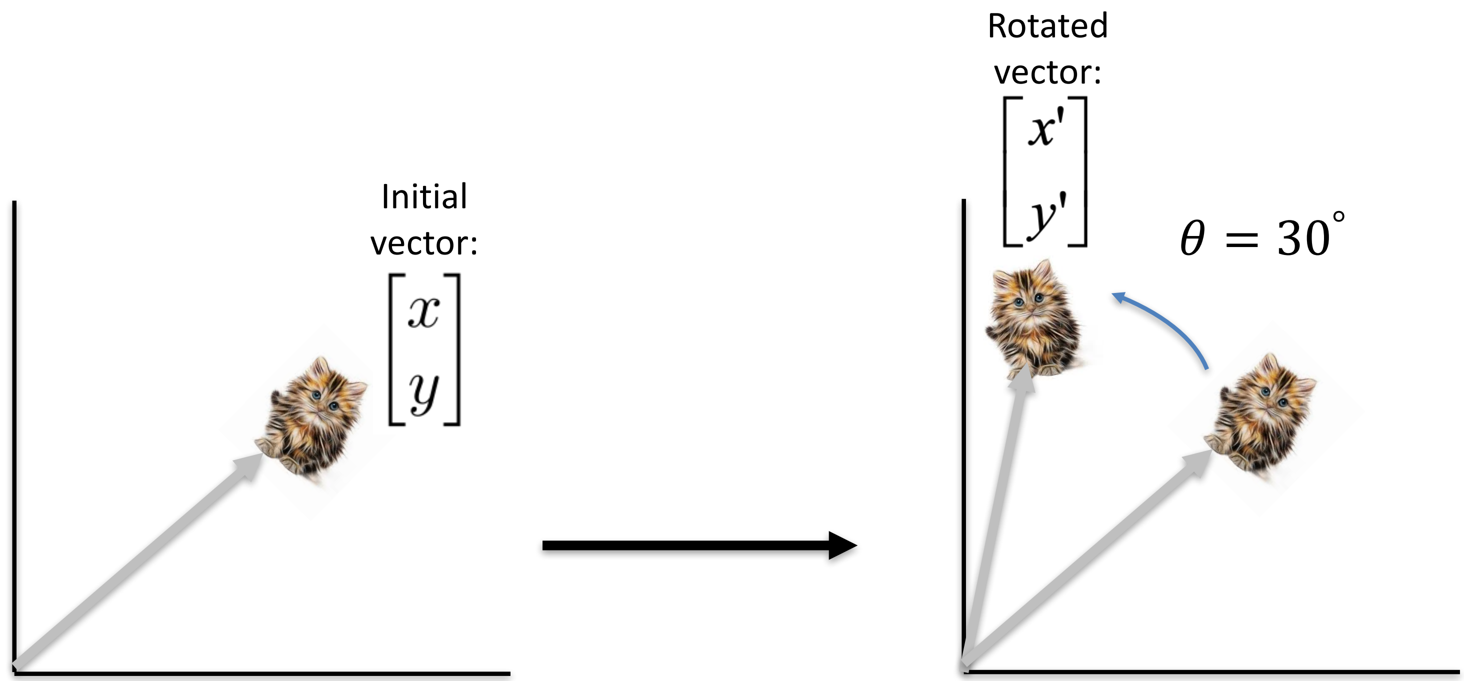
(Verify to yourself that the matrix multiplication works out this way)

Transformation: scaling

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

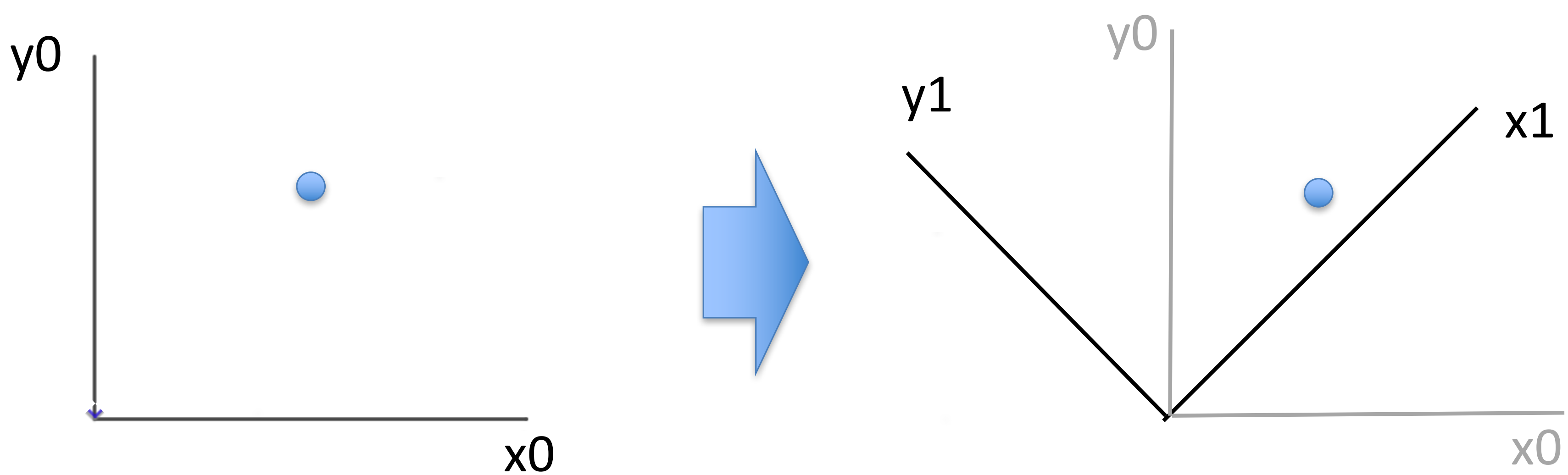


Transformation: rotation



Before learning how to rotate a vector, let's understand how to do something slightly different

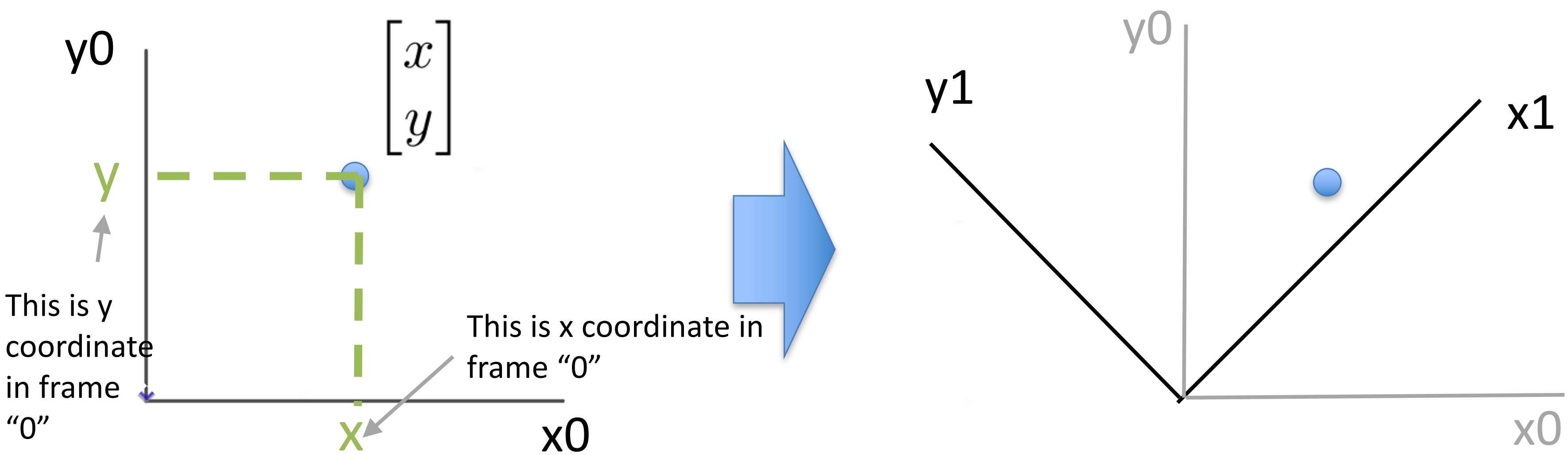
- How can you convert a vector represented in frame "0" to a new, rotated coordinate frame "1"?



It is the **same** data point, but it is represented in a new, rotated frame.

Before learning how to rotate a vector, let's understand how to do something slightly different

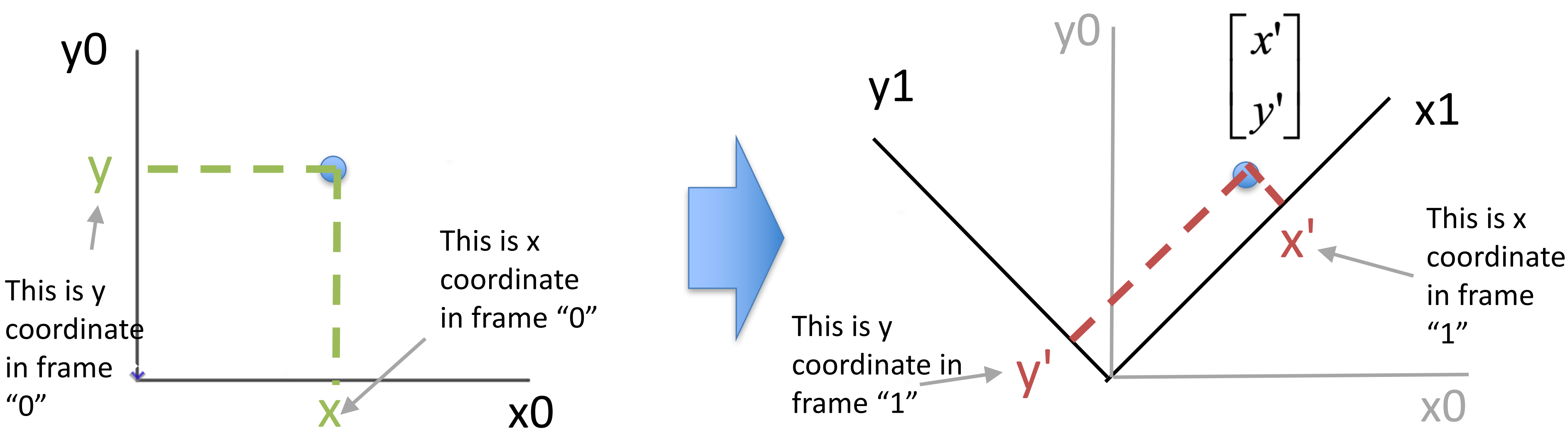
- How can you convert a vector represented in frame "0" to a new, rotated coordinate frame "1"?



It is the **same** data point, but it is represented in a new, rotated frame.

Before learning how to rotate a vector, let's understand how to do something slightly different

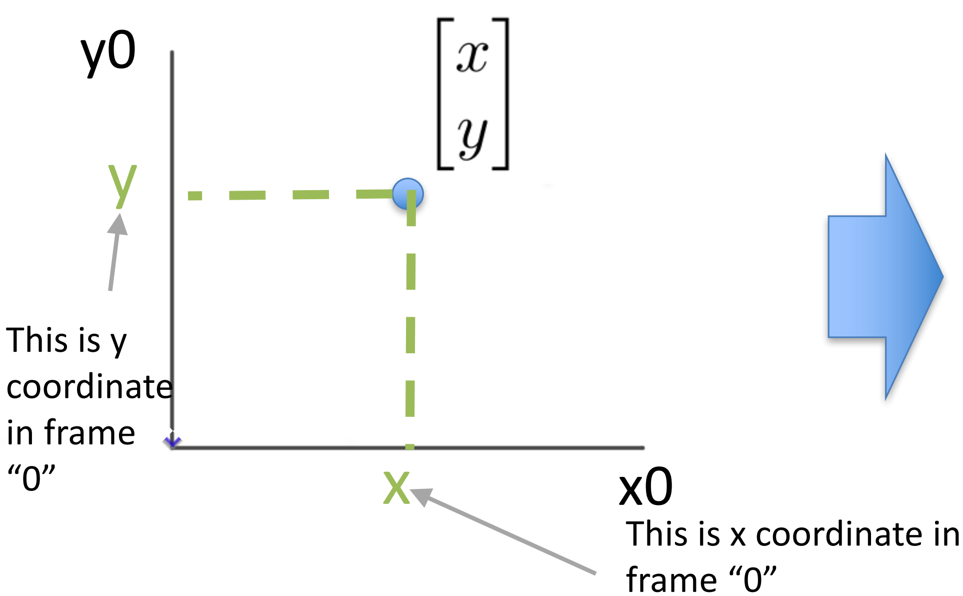
- How can you convert a vector represented in frame "0" to a new, rotated coordinate frame "1"?

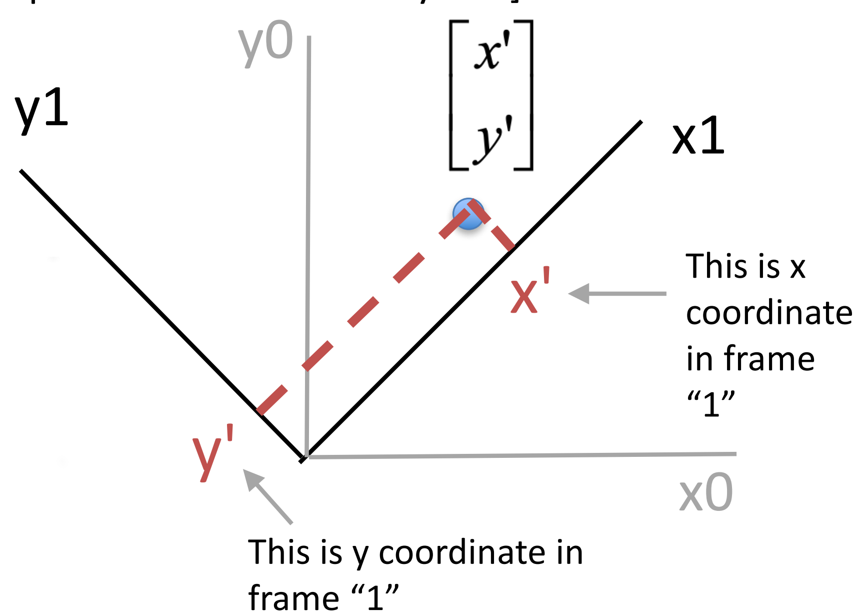


It is the **same** data point, but it is represented in a new, rotated frame.

Before learning how to rotate a vector, let's understand how to do something slightly different

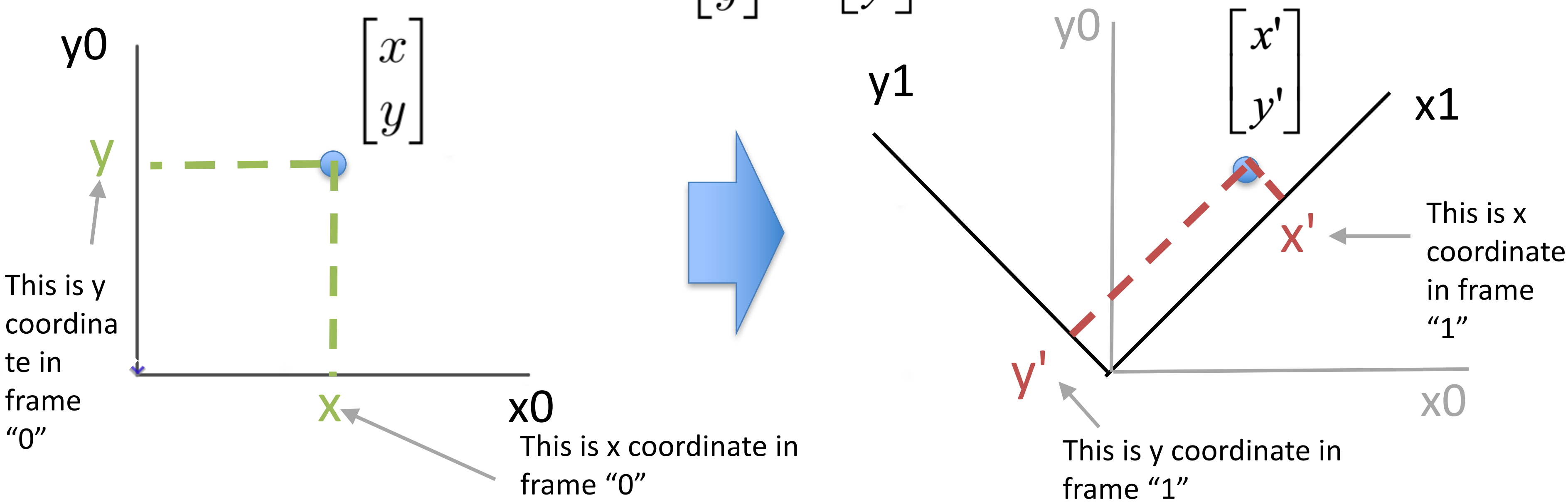
- How can you convert a vector represented in frame "0" to a new, rotated coordinate frame "1"?
- Remember what a vector is:
  [component in direction of the frame's x axis, component in direction of y axis]





Before learning how to rotate a vector, let's understand how to do something slightly different

- How can you convert a vector represented in frame "0" to a new, rotated coordinate frame "1"?
- Goal: How do we go from $\begin{bmatrix} x \\ y \end{bmatrix}$ to $\begin{bmatrix} x' \\ y' \end{bmatrix}$?



Before learning how to rotate a vector, let's understand how to do something slightly different

- How can you convert a vector represented in frame "0" to a new, rotated coordinate frame "1"?
- Goal: How do we go from $\begin{bmatrix} x \\ y \end{bmatrix}$ to $\begin{bmatrix} x' \\ y' \end{bmatrix}$?

- Answer: we use dot products!!
  - x' (the new x coordinate) is the length of the original vector which lies in the direction of the new x axis
  - y' (the new y coordinate) is the length of the original vector which lies in the direction of the new y axis
- So:
  - x' is [original vector] **dot** [the new x axis]
  - y' is [original vector] **dot** [the new y axis]

Before learning how to rotate a vector, let's understand how to do something slightly different

- How can you convert a vector represented in frame "0" to a new, rotated coordinate frame "1"?
- Goal: How do we go from $\begin{bmatrix} x \\ y \end{bmatrix}$ to $\begin{bmatrix} x' \\ y' \end{bmatrix}$?
  - x' (the new x coordinate) is [original vector] **dot** [the new x axis]
  - y' (the new y coordinate) is [original vector] **dot** [the new y axis]

- So:
  - x' = [x, y] **dot** [the new x axis]
  - y' = [x, y] **dot** [the new y axis]

Before learning how to rotate a vector, let's understand how to do something slightly different

- How can you convert a vector represented in frame "0" to a new, rotated coordinate frame "1"?
- Goal: How do we go from $\begin{bmatrix} x \\ y \end{bmatrix}$ to $\begin{bmatrix} x' \\ y' \end{bmatrix}$?
  - x' = [x, y] **dot** [the new x axis]
  - y' = [x, y] **dot** [the new y axis]
- So:
  - $\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \text{[the new x axis] } \textbf{dot} \text{ [x, y]} \\ \text{[the new y axis] } \textbf{dot} \text{ [x, y]} \end{bmatrix}$

Before learning how to rotate a vector, let's understand how to do something slightly different

- How can you convert a vector represented in frame "0" to a new, rotated coordinate frame "1"?
- Goal: How do we go from $\begin{bmatrix} x \\ y \end{bmatrix}$ to $\begin{bmatrix} x' \\ y' \end{bmatrix}$?
    - $\begin{bmatrix} x' \\ y' \end{bmatrix}$ $\begin{bmatrix} \text{[the new x axis] } \textbf{dot} \text{ [x, y]} \\ \text{[the new y axis] } \textbf{dot} \text{ [x, y]} \end{bmatrix}$
- So:
    - $\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \text{[the new x axis]} \\ \text{[the new y axis]} \end{bmatrix} \text{X} \begin{bmatrix} x \\ y \end{bmatrix}$
    - $\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \text{2x2 Rotation Matrix} \end{bmatrix} \text{X} \begin{bmatrix} x \\ y \end{bmatrix}$

→ it is a matrix-vector multiplication!!

Before learning how to rotate a vector, let's understand how to do something slightly different

- How can you convert a vector represented in frame "0" to a new, rotated coordinate frame "1"?
- Goal: How do we go from $\begin{bmatrix} x \\ y \end{bmatrix}$ to $\begin{bmatrix} x' \\ y' \end{bmatrix}$?

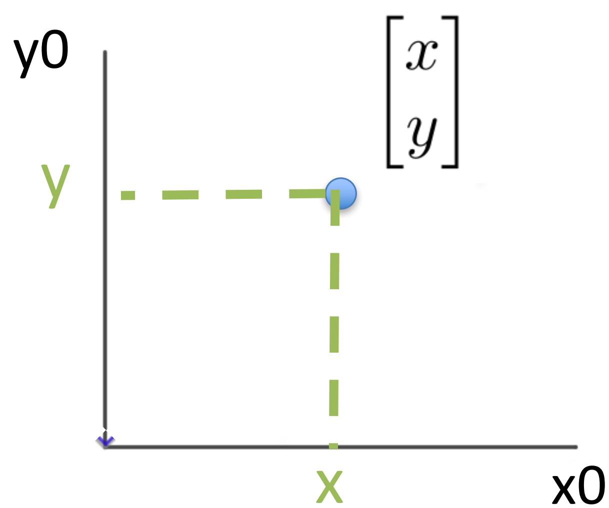
→ answer: it is a matrix-vector multiplication!!

- $\begin{bmatrix} x' \\ y' \end{bmatrix}$ = [2x2 Rotation Matrix] x $\begin{bmatrix} x \\ y \end{bmatrix}$
- $\begin{bmatrix} x' \\ y' \end{bmatrix}$ = R x $\begin{bmatrix} x \\ y \end{bmatrix}$

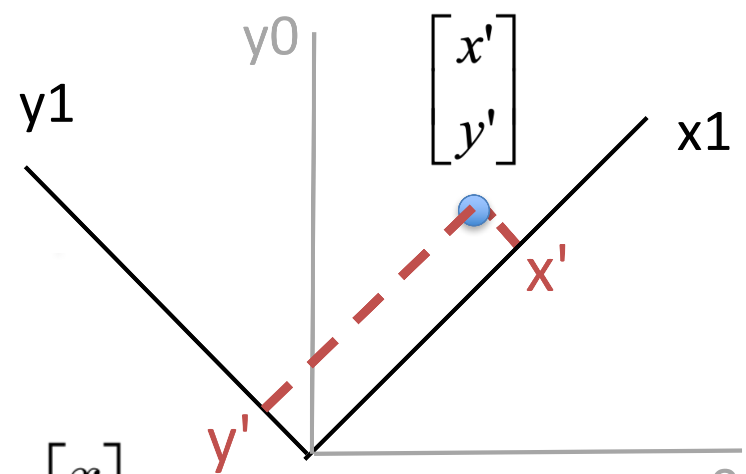
$$\mathbf{P'} = \mathbf{R}\ \mathbf{P}$$

Before learning how to rotate a vector, let's understand how to do something slightly different

- How can you convert a vector represented in frame "0" to a new, rotated coordinate frame "1"?
- Answer: using a matrix-vector multiplication!!

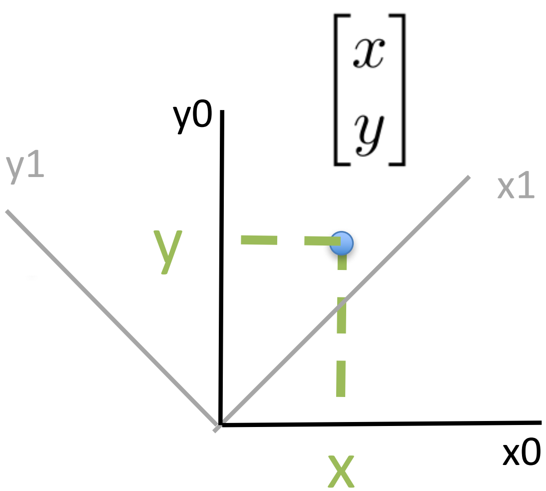


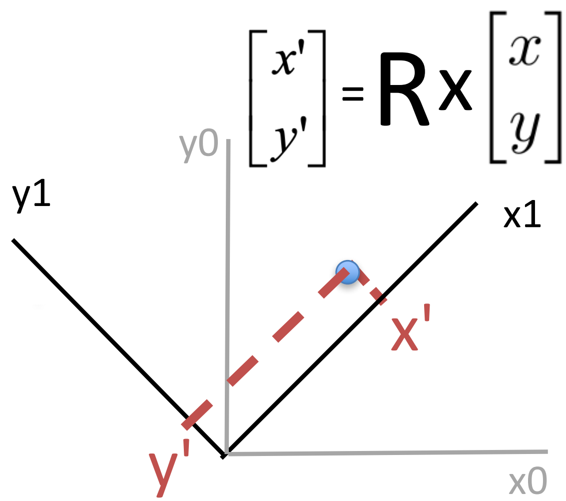
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = R \times \begin{bmatrix} x \\ y \end{bmatrix}$$



Now, here's how we do a rotation!!

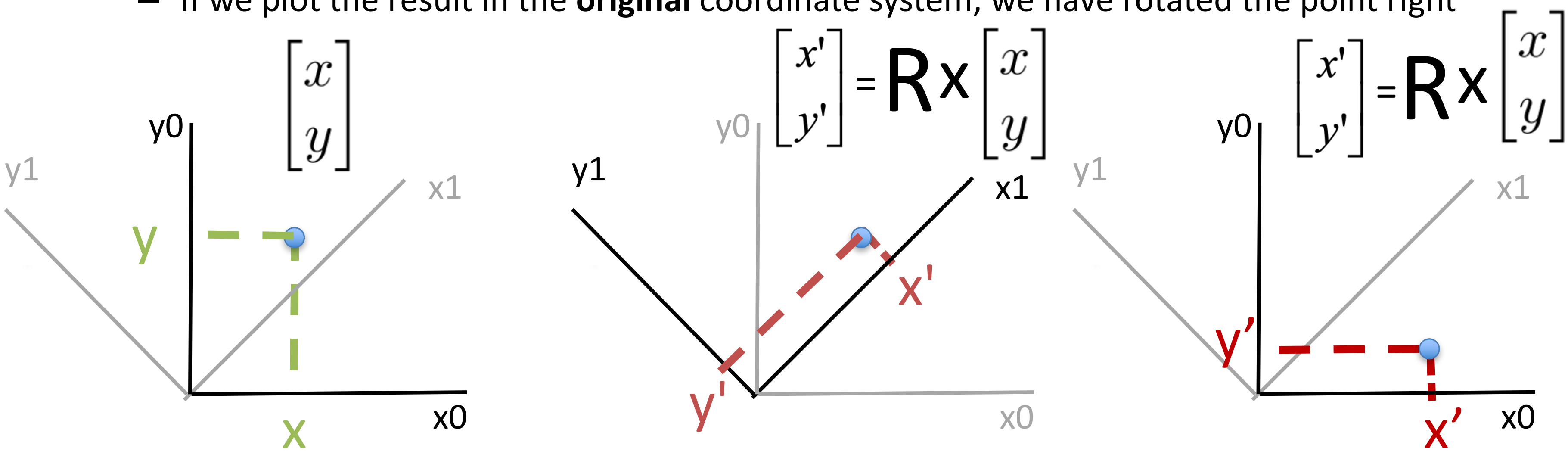
- Until now, we've seen how to convert a data point represented in frame "0" to a new, rotated coordinate frame "1". We rotated the frame to the left.





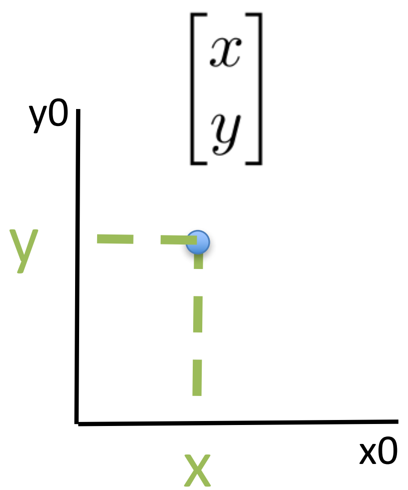
Now, here's how we do a rotation!!

- Until now, we've seen how to convert a data point represented in frame "0" to a new, rotated coordinate frame "1". We rotated the frame to the left.
- Insight: rotating the frame to the left == rotating a data point to the right
  - Suppose we express a point in the new coordinate system "1"
  - If we plot the result in the **original** coordinate system, we have rotated the point right



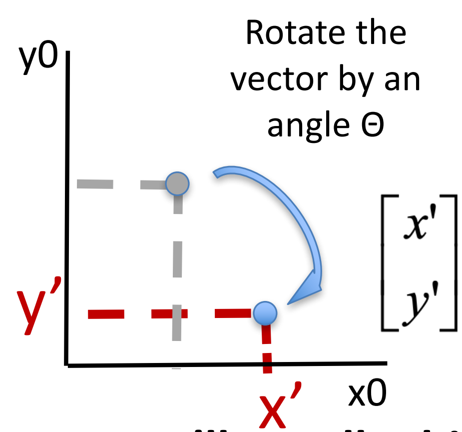
Now, here's how we do a rotation!!

- Until now, we've seen how to convert a data point represented in frame "0" to a new, rotated coordinate frame "1".
- Insight: rotating the frame to the left == rotating a data point to the right



Rotation by an angle Θ

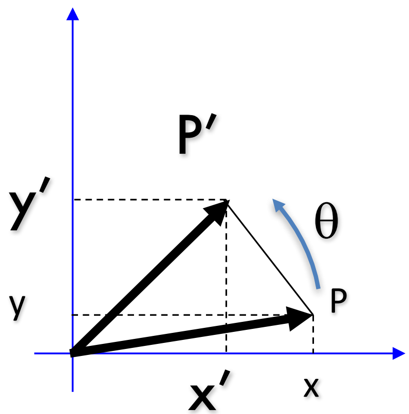
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = R \times \begin{bmatrix} x \\ y \end{bmatrix}$$



- **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: what is R?

Counter-clockwise rotation by an angle θ



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

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

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

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

Transformation Matrices

- Multiple transformation matrices can be used to transform a point:
  $p' = R_2 R_1 S p$
- 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.

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

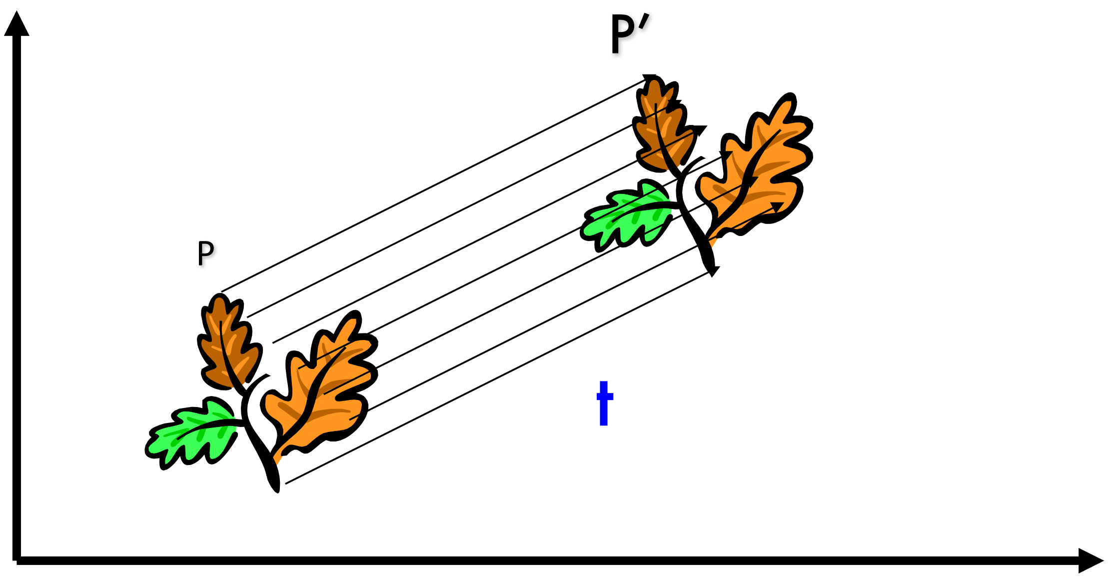
- Generally, a homogeneous transformation matrix will have a bottom row of [0 0 1], so that the result has a "1" at the bottom too.

Homogeneous system

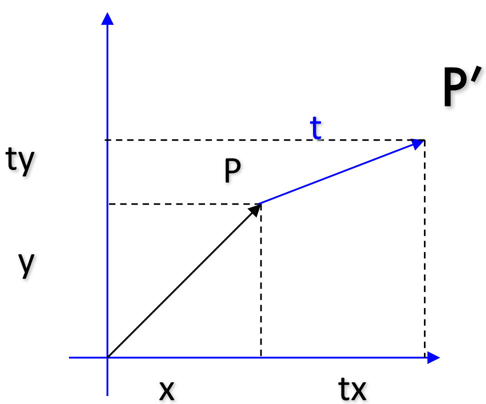
- One more thing we might want: to divide the result by something
  - For example, we may want to divide by a coordinate, to make things scale down as they get farther away in a camera image
  - Matrix multiplication can't actually divide
  - So, **by convention**, in homogeneous coordinates, we'll divide 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 using Homogeneous Coordinates



2D Translation using Homogeneous Coordinates

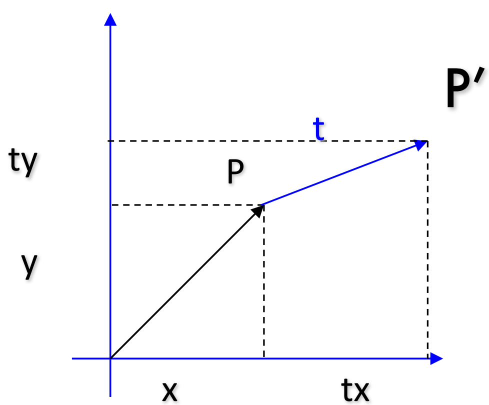


$$\mathbf{P} = (x, y) \rightarrow (x, y, 1)$$

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

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

2D Translation using Homogeneous Coordinates

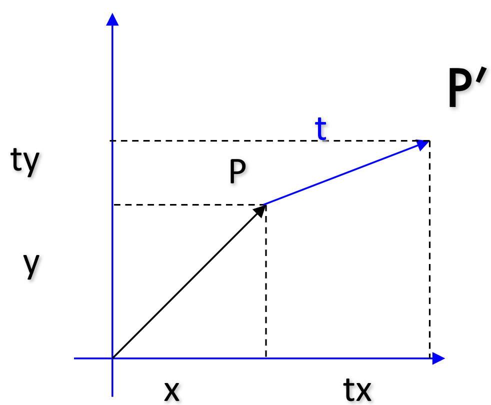


$$\mathbf{P} = (x, y) \rightarrow (x, y, 1)$$

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

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

2D Translation using Homogeneous Coordinates

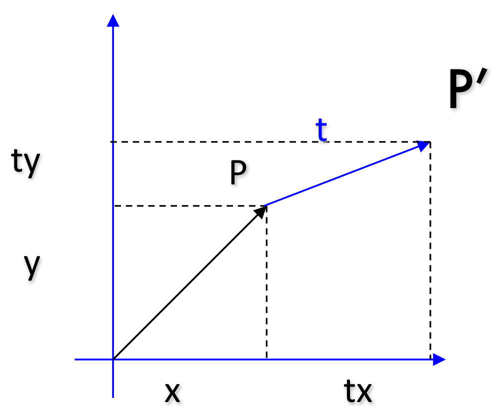


$$\mathbf{P} = (x, y) \rightarrow (x, y, 1)$$

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

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

2D Translation using Homogeneous Coordinates

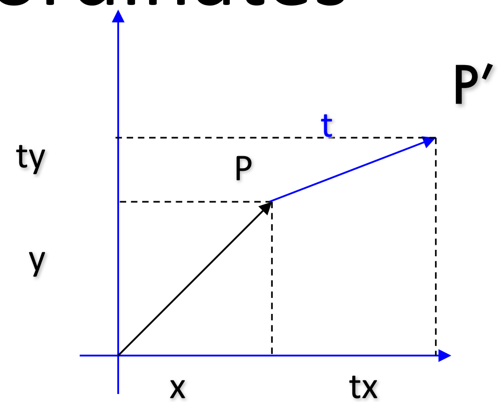


$$\mathbf{P} = (x, y) \rightarrow (x, y, 1)$$

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

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

2D Translation using Homogeneous Coordinates



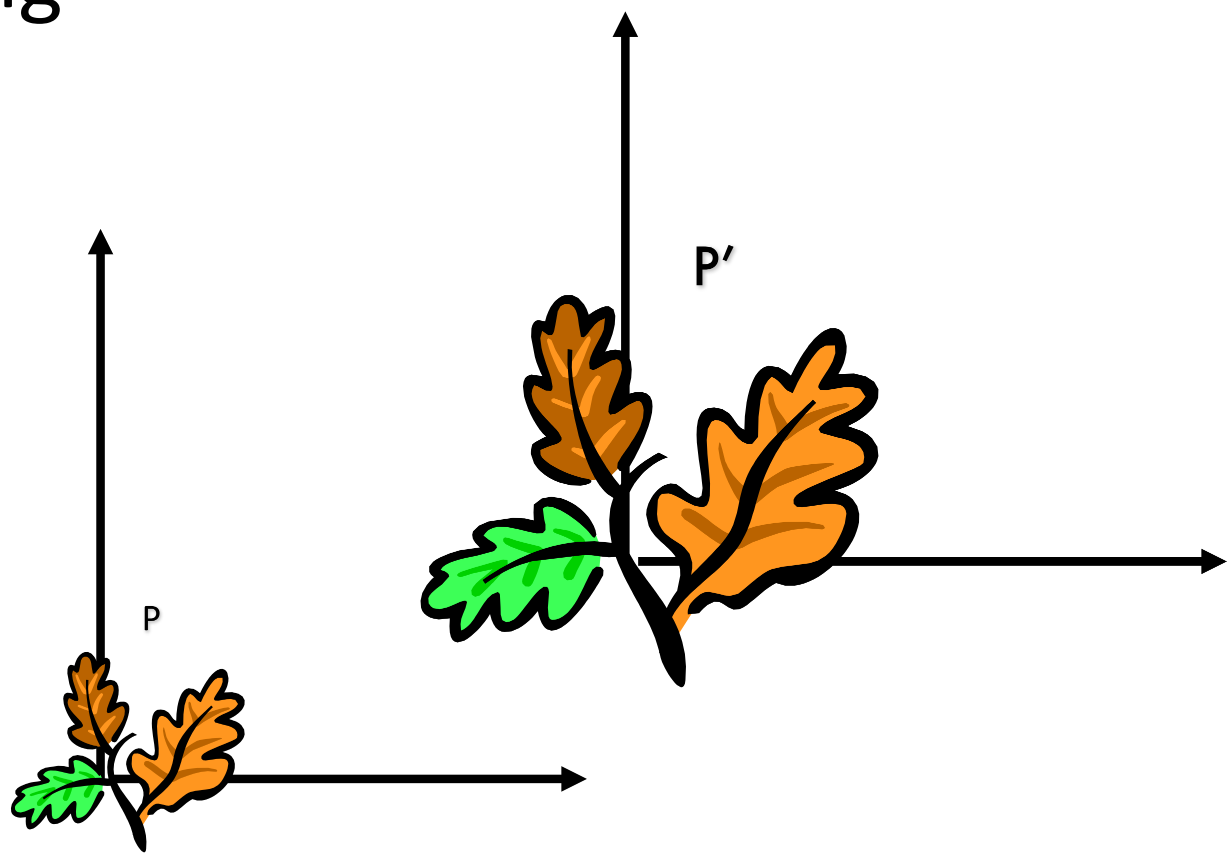
$$\mathbf{P} = (x, y) \rightarrow (x, y, 1)$$

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

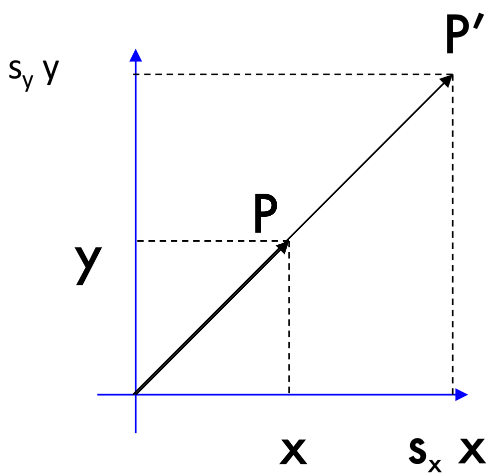
$$\mathbf{P}' \rightarrow \begin{bmatrix} x + t_x \\ y + t_y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & \overbrace{t_x}^{\text{t}} \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \overbrace{x}^{\text{P}} \\ y \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{I} & \mathbf{t} \\ 0 & 1 \end{bmatrix} \cdot \mathbf{P} = \mathbf{T} \cdot \mathbf{P}$$

Scaling



Scaling Equation

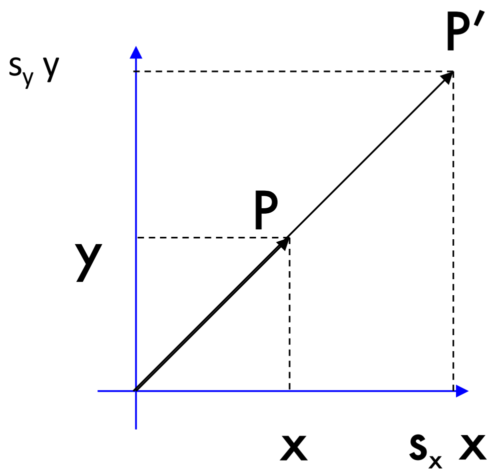


$$\mathbf{P} = (\mathrm{x}, \mathrm{y}) \rightarrow \mathbf{P'} = (\mathrm{s_x x}, \mathrm{s_y y})$$

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

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

Scaling Equation



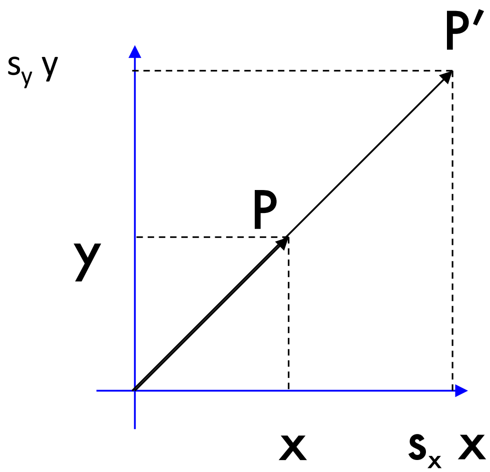
$$\mathbf{P} = (\mathrm{x}, \mathrm{y}) \rightarrow \mathbf{P'} = (\mathrm{s_x x}, \mathrm{s_y y})$$

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

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

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

Scaling Equation



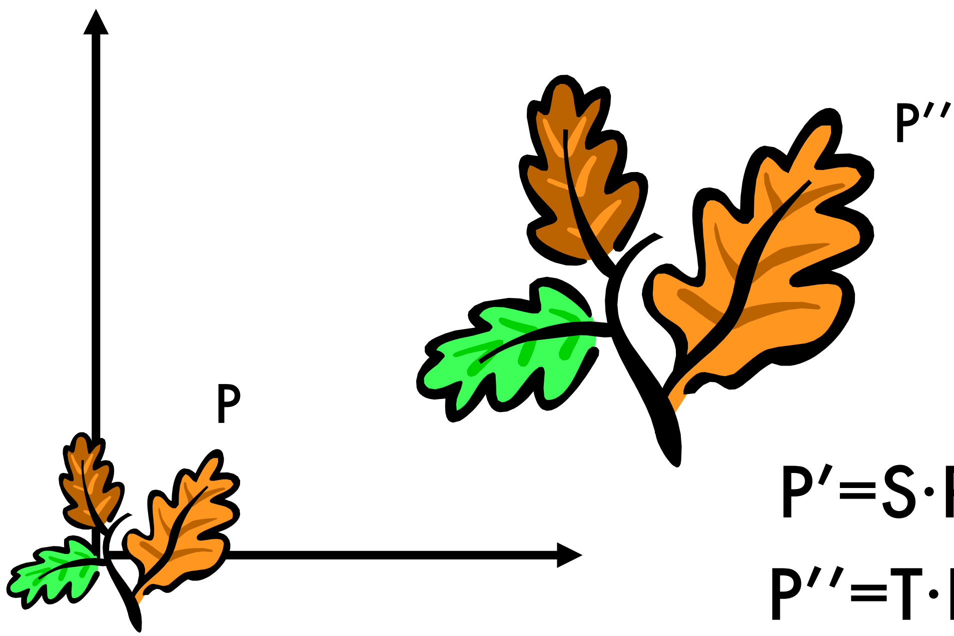
$$\mathbf{P} = (\mathrm{x}, \mathrm{y}) \rightarrow \mathbf{P}' = (\mathrm{s_x x}, \mathrm{s_y y})$$

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

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

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

Scaling & Translating



$P' = S \cdot P$

$P'' = T \cdot P'$

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

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 != 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

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

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

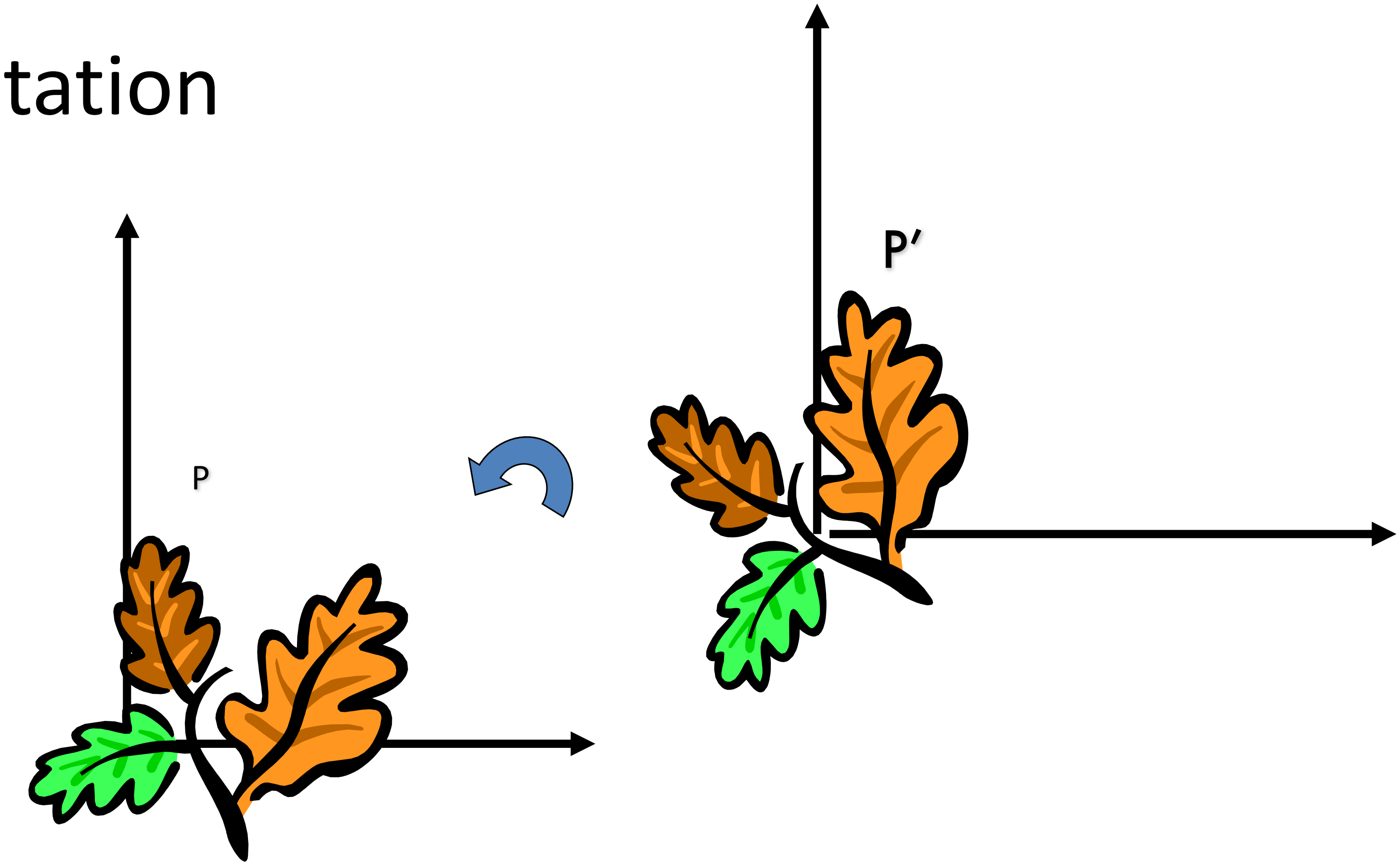
Translating & Scaling != Scaling & Translating

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

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

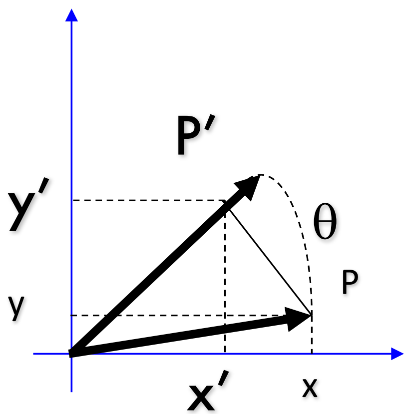
$$= \begin{bmatrix} s_x & 0 & s_x t_x \\ 0 & s_y & s_y t_y \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} s_x x + s_x t_x \\ s_y y + s_y t_y \\ 1 \end{bmatrix}$$

Rotation



Rotation Equations

Counter-clockwise rotation by an angle θ



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

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

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

Rotation Matrix Properties

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

$$\mathrm{x}' = \cos\theta\ \mathrm{x} - \sin\theta\ \mathrm{y}$$
$$\mathrm{y}' = \cos\theta\ \mathrm{y} + \sin\theta\ \mathrm{x}$$

A 2D rotation matrix is 2x2

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

$$\mathbf{R} \cdot \mathbf{R}^{\mathrm{T}} = \mathbf{R}^{\mathrm{T}} \cdot \mathbf{R} = \mathbf{I}$$
$$\det(\mathbf{R}) = 1$$

Rotation Matrix Properties

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

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

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

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

Rotation Equation

$$\mathbf{P} = (\mathrm{x}, \mathrm{y}) \rightarrow \mathbf{P}' = (\cos\theta.\mathrm{x} - \sin\theta.y, \qquad \cos\theta.y + \sin\theta.x)$$

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

$$\mathbf{P}' = (\cos\theta.\mathrm{x} - \sin\theta.\mathrm{y}, \cos\theta.\mathrm{y} + \sin\theta.\mathrm{x})$$
$$\rightarrow (\cos\theta.\mathrm{x} - \sin\theta.\mathrm{y},\ \cos\theta.\mathrm{y} + \sin\theta.\mathrm{x}, 1)$$

Rotation Equation

$$\mathbf{P} = (\mathrm{x}, \mathrm{y}) \rightarrow \mathbf{P}' = (\cos(\theta)\mathrm{x} - \sin(\theta)y, \cos(\theta)\, y + \sin(\theta)\, x)$$

$$\mathbf{P} = (\mathrm{x}, \mathrm{y}) \rightarrow (\mathrm{x}, \mathrm{y}, 1)$$
$$\mathbf{P}' = (\cos(\theta)\mathrm{x} - \sin(\theta)\mathrm{y}, \cos(\theta)\,\mathrm{y} + \sin(\theta)\,\mathrm{x})$$
$$\rightarrow (\cos(\theta)\mathrm{x} - \sin(\theta)\mathrm{y}, \cos(\theta)\,\mathrm{y} + \sin(\theta)\,\mathrm{x}, 1)$$

$$\mathbf{P}' \rightarrow \begin{bmatrix} \cos(\theta)\,\mathrm{x} - \sin(\theta)\,\mathrm{y} \\ \cos(\theta)\,\mathrm{y} + \sin(\theta)\,\mathrm{x} \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} \mathrm{x} \\ \mathrm{y} \\ 1 \end{bmatrix}$$
$$= \begin{bmatrix} \mathbf{R}' & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \cdot \mathbf{P} = \mathbf{R} \cdot \mathbf{P}$$

Scaling + Rotation + Translation

P'= (T R S) P

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

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

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

This is the form of the general-purpose transformation matrix