中山大學
SUN YAT-SEN UNIVERSITY

Lecture 2. Images and Transformations

Pattern Recognition (in Computer Vision)

Jinhua Ma,
School of Computer Science and Engineering, Sun Yat-Sen University

(Tentative) Schedule

- L1. Introduction to PR
- L2. Images and Transformations
- L3. Color and Filters
- L4. Features and fitting
- L5. Feature descriptors
- L6. Clustering and Segmentation
- L7. Dimensionality Reduction
- L8. Face identification
- L9. Bayesian Decision Theory
- L10. Image Classification
- L11. Regularization and Optimization
- L12. Image Classification with CNNs
- L13. CNN Architectures
- L14. Training Neural Networks
- L15. Object Detection and Image Segmentation
- L16. Recurrent Neural Networks
- L17. Attention and Transformers
- L18. Generative Models
- L19. Self-supervised Learning

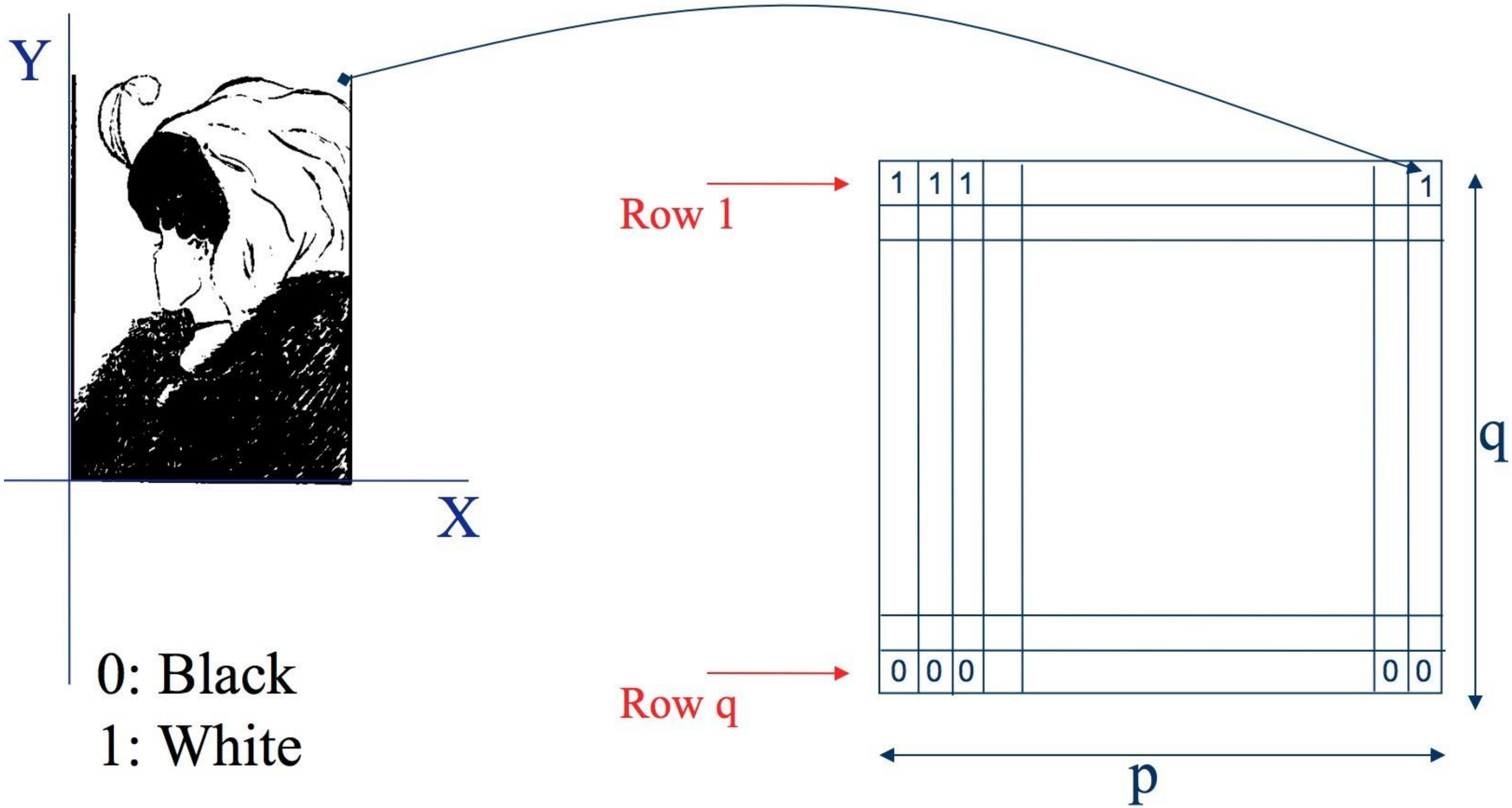
Types of Images

Binary

Grayscale

Color

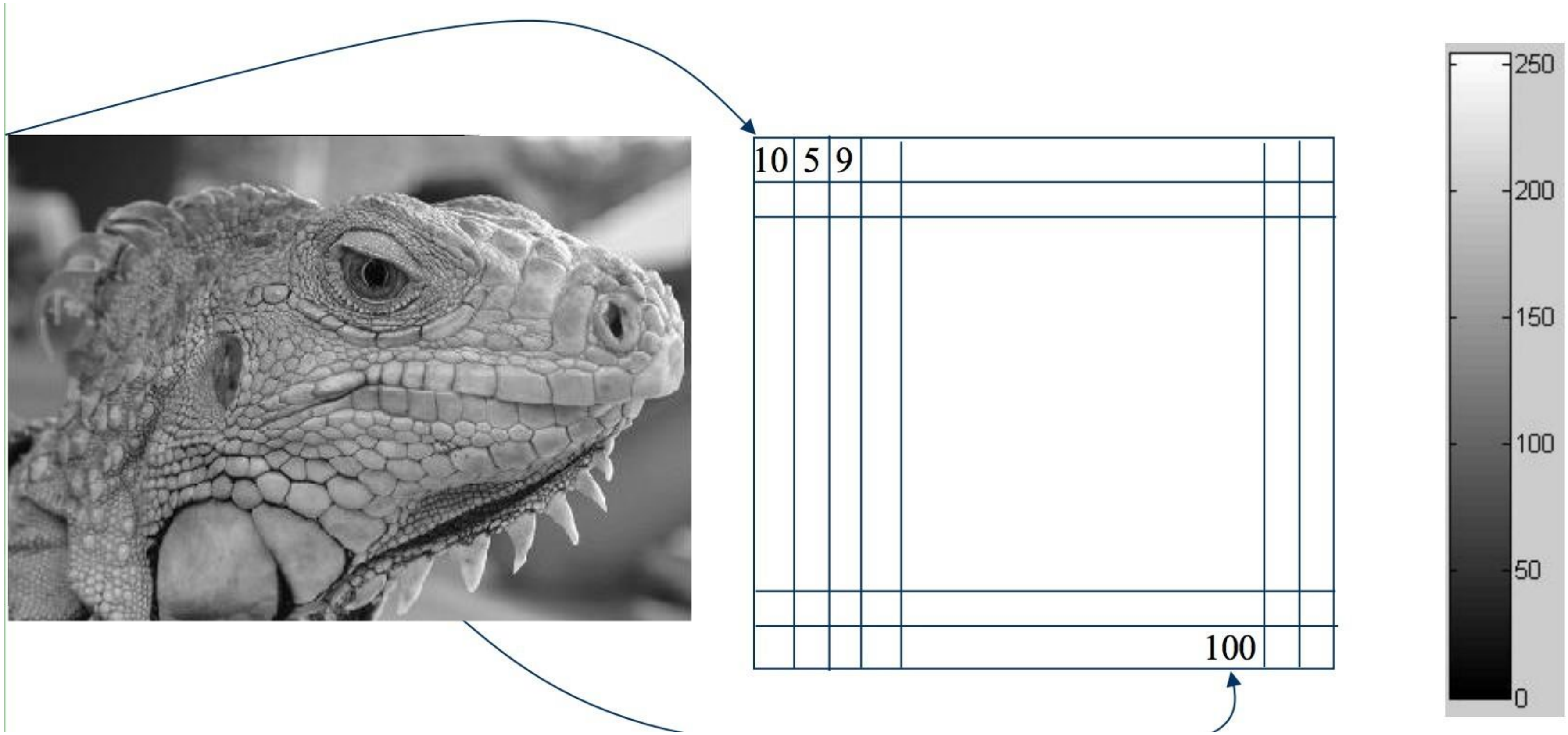
Binary image representation



0: Black
1: White

Slide credit: Ulas Bagci

Grayscale image representation



Slide credit: Ulas Bagci

Color image representation

B channel

G channel

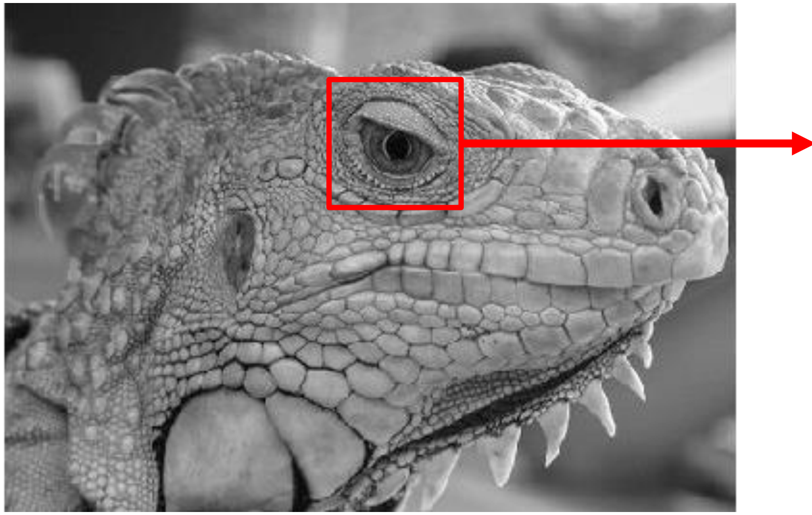
R channel

Slide credit: Ulas Bagci

Color image - one channel

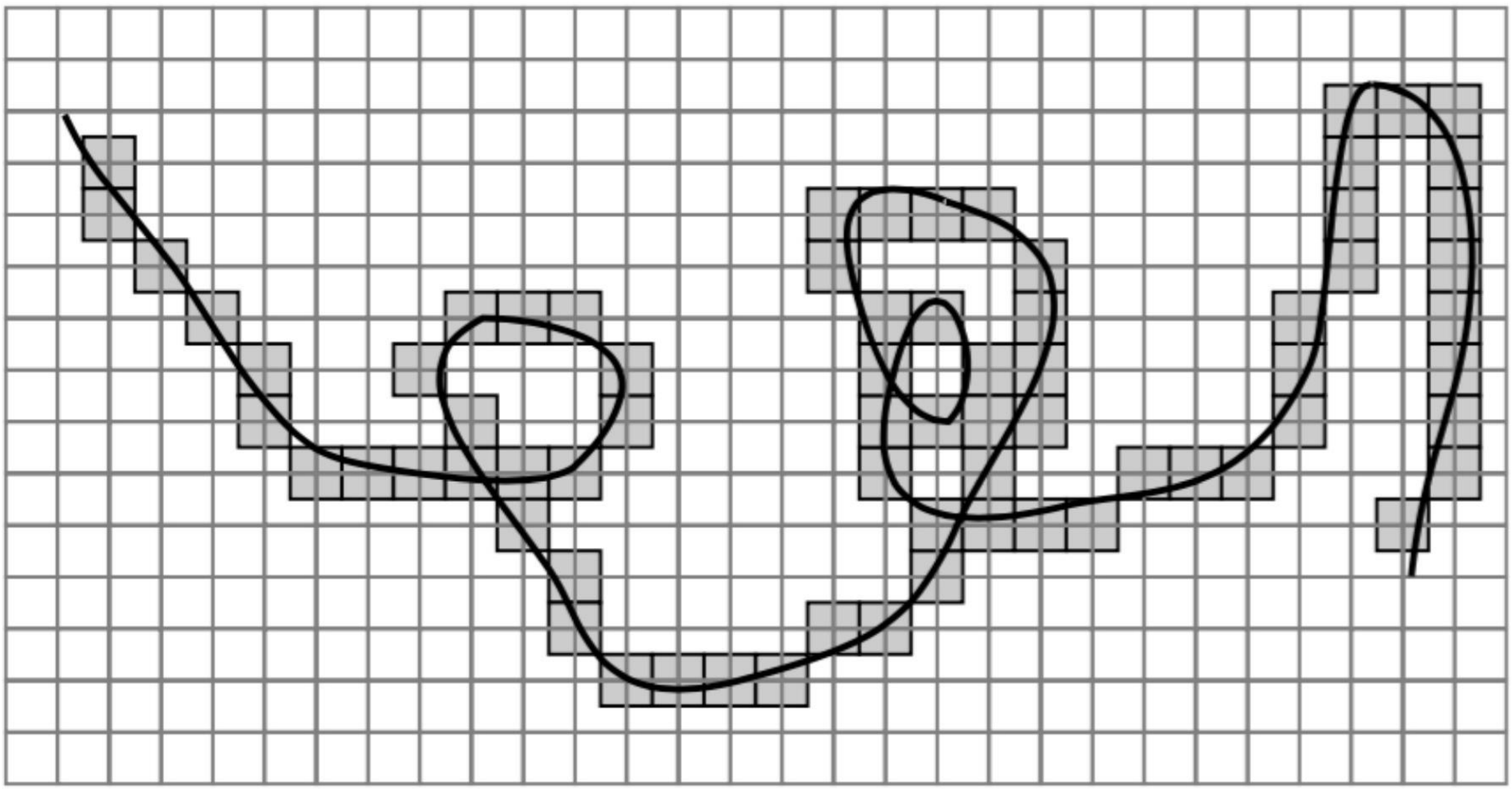
R channel

Digital Images are sampled



- What happens when we zoom into the images we capture?

Errors due to Sampling



Slide credit: Ulas Bagci

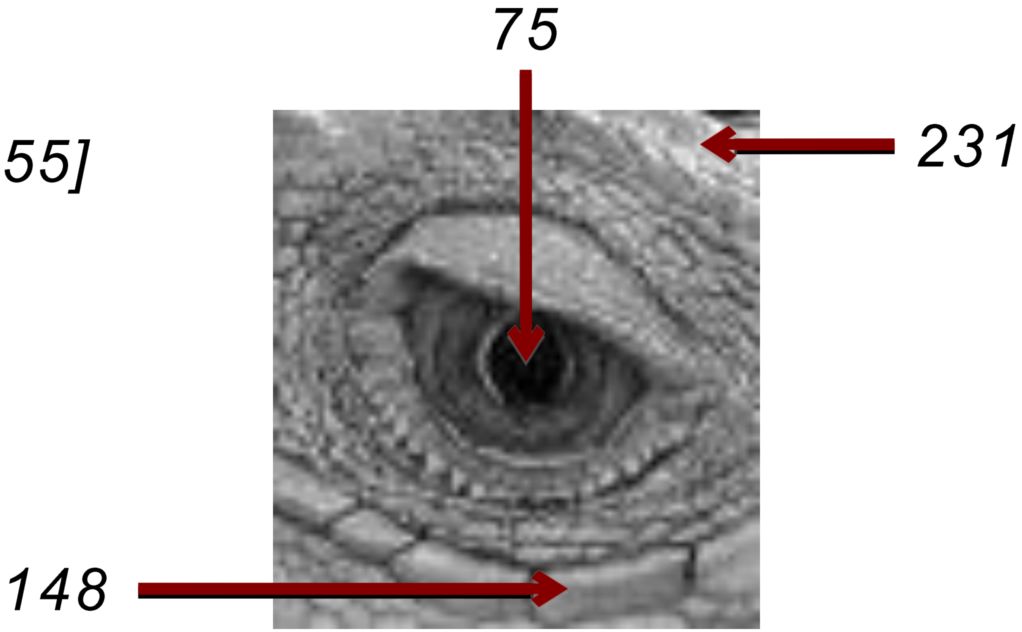
Resolution

is a **sampling** *parameter, defined in dots per inch (DPI) or equivalent measures of spatial pixel density*

Slide credit: Ulas Bagci

Images are Sampled and Quantized

- An image contains discrete number of pixels
  - Pixel value:
    - *“grayscale”*
      (or “intensity”): [0,255]



Images are Sampled and Quantized

- An image contains discrete number of pixels
  - Pixel value:
    - *"grayscale"*
      (or "intensity"): [0,255]
    - *"color"*
  - RGB: [R, G, B]

[90, 0, 53]

[249, 215, 203]

[213, 60, 67]

With this loss of information (from sampling and quantization),

Can we still use images for useful tasks?

Summary

- *Image types (binary, grayscale, color)*
- *Images are sampled and quantized*

2D transformations

- *Transformation Matrices*
- *Homogeneous coordinates*
- *Translation*
- *Scaling*
- *Rotation*

Transformation Matrices

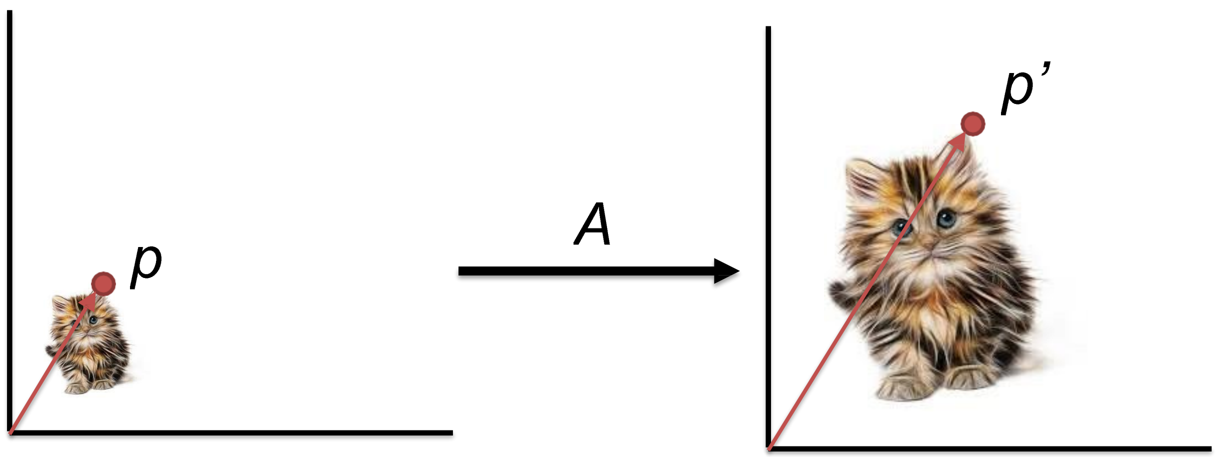
- *Matrices can be used to transform vectors in useful ways, through multiplication: p'= A p*
- *Simplest is scaling:*

$$\underbrace{\begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix}}_{A} \times \underbrace{\begin{bmatrix} x \\ y \end{bmatrix}}_{p} = \underbrace{\begin{bmatrix} s_x x \\ s_y y \end{bmatrix}}_{p'}$$

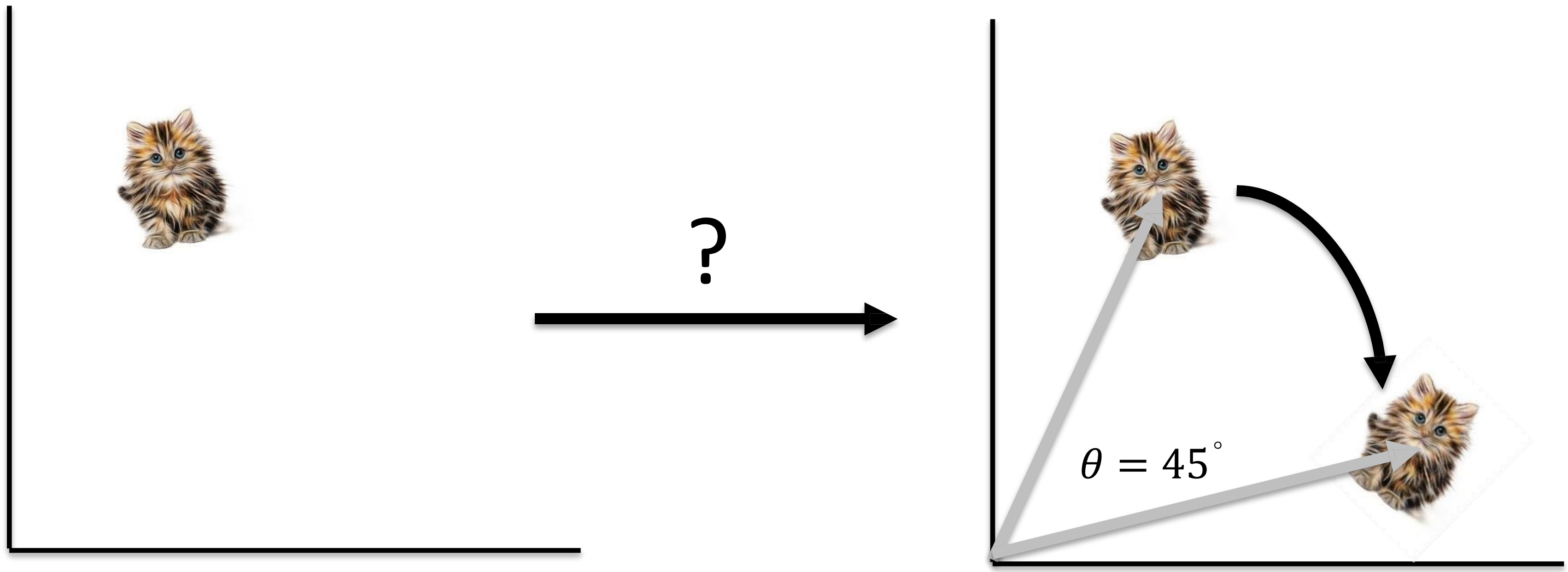
(Verify to yourself that the matrix multiplication works out this way)

Transformation Matrices

$$\begin{array}{ccccc} \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} & \times & \begin{bmatrix} x \\ y \end{bmatrix} & = & \begin{bmatrix} s_x x \\ s_y y \end{bmatrix} \\ A & & p & & p' \end{array}$$

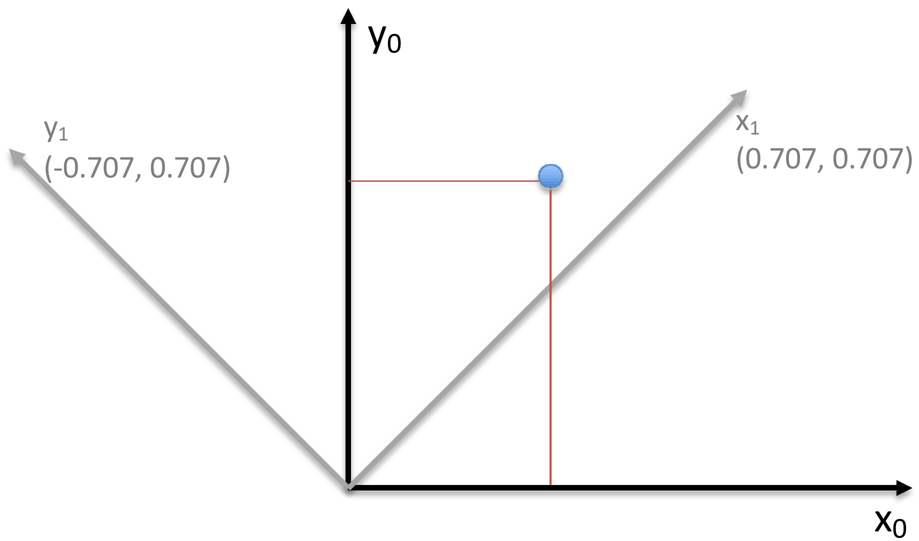


Rotation



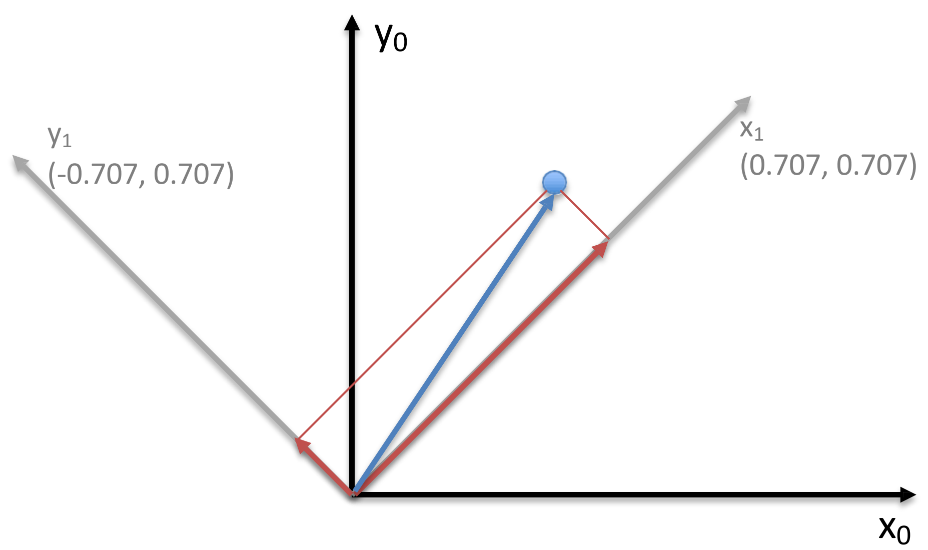
Rotation

- *How can you convert a vector represented in the coordinate 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 [the new x axis]* (x_1) **dot** *[original vector]*
- *New y coordinate is [the new y axis]* (y_1) **dot** *[original vector]*

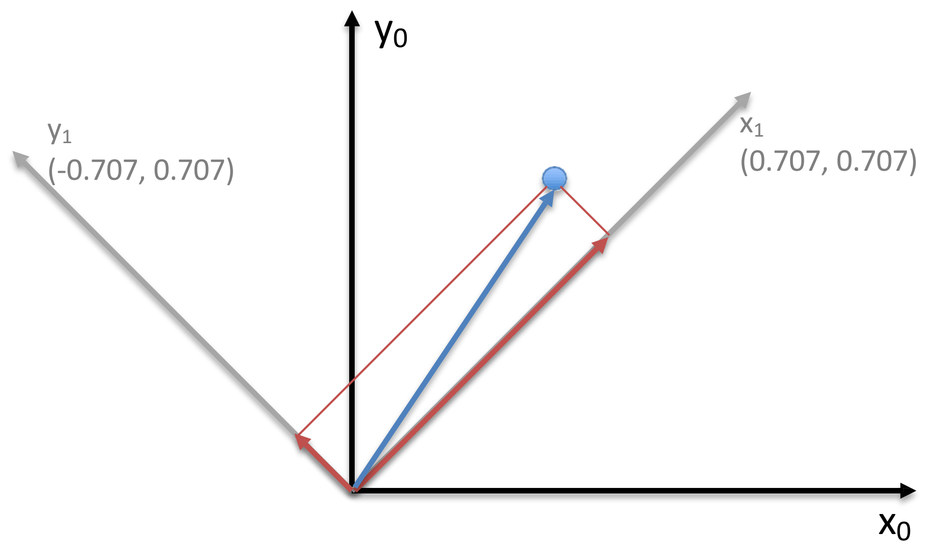


Rotation

- *Insight: something similar happens in a matrix*vector multiplication!*
- *The resulting x coordinate, x', is: [matrix row 1]* **dot** *[original vector]*

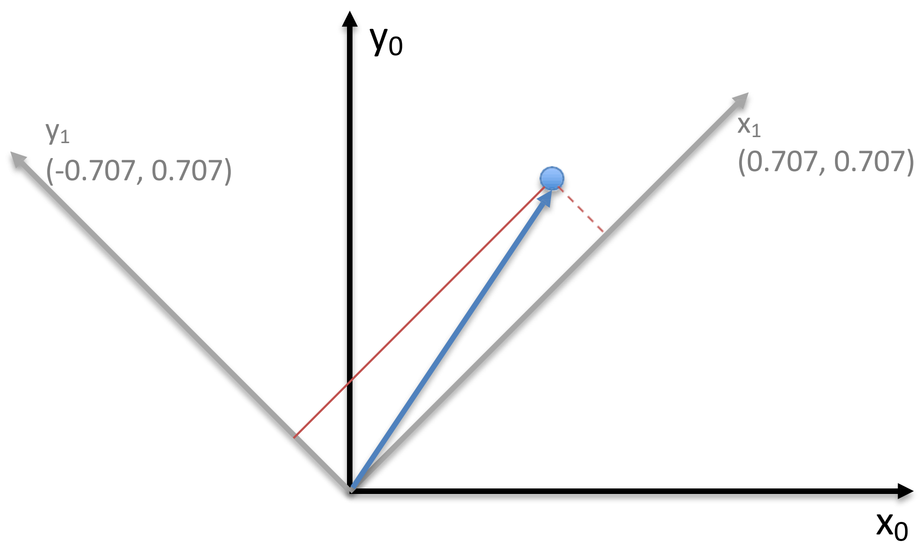
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 0.707 & 0.707 \\ -0.707 & 0.707 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

- *The matrix multiplication Rp = p' produces the coordinates in the new frame.*



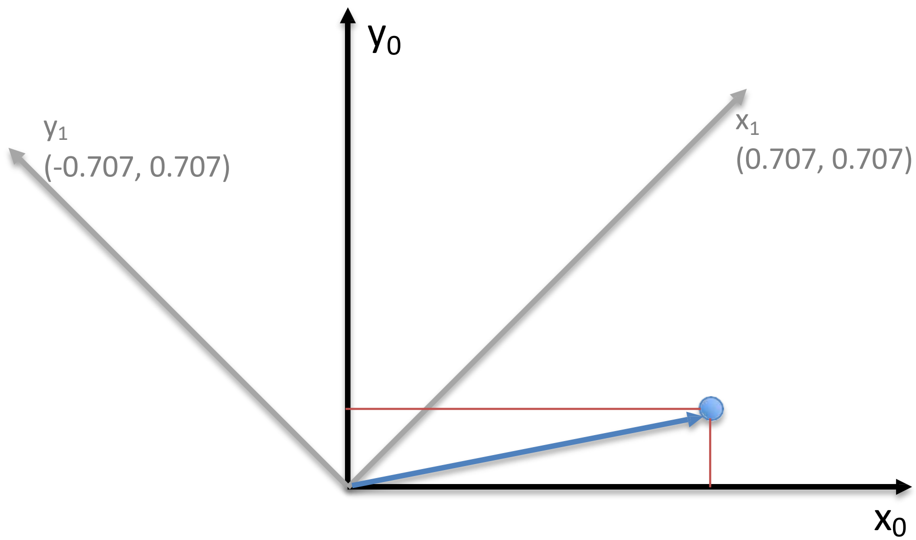
Rotation

- *Now we have our 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*



Rotation

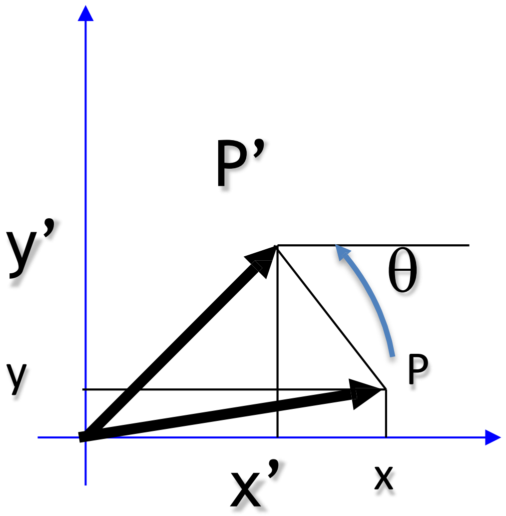
- *Now we have our point in the new coordinate system which is rotated left*
- *If we plot the result in the* **original** *coordinate system, we have rotated the point right*



– *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 θ



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

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

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

$$p' = Rp$$

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 equivalent to*

$$p' = R_2(R_1(Sp))$$

- *The result is exactly the same if we multiply the matrices first, to form a single transformation matrix:*

$$p' = (R_2 R_1 S)p$$

2D transformations

- *Transformation Matrices*
- *Homogeneous coordinates*
- *Translation*
- *Scaling*
- *Rotation*

Homogeneous coordinates

- *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 scaling, rotating, and skewing* （倾斜） *transformations.*
- *But notice, we can't add a constant!* ☹
- *That means, we cannot produce a new (translated) vector* $\begin{bmatrix} \mathrm{x} + k \\ y + k \end{bmatrix}$.

Homogeneous coordinates

- *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 coordinates

- *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 coordinates

- *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 an 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*

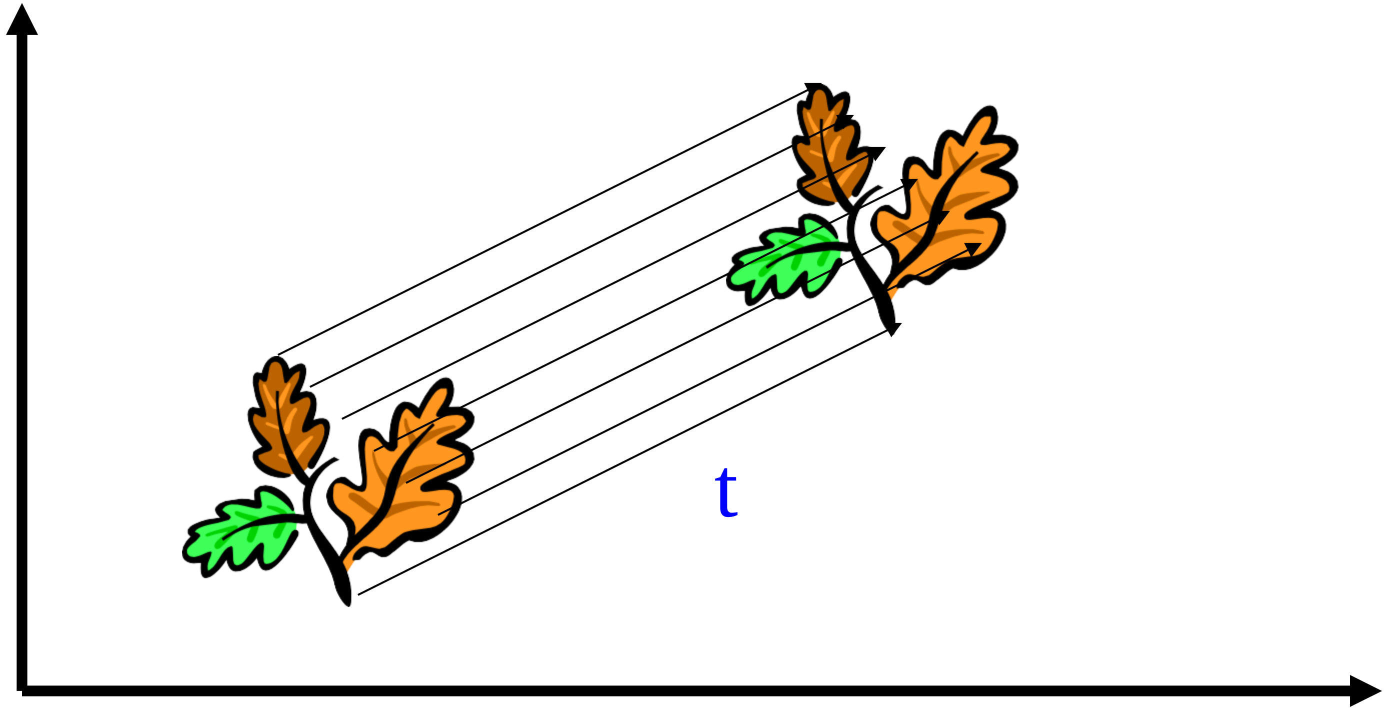
$$\begin{bmatrix} x \\ y \\ 7 \end{bmatrix} \Rightarrow \begin{bmatrix} x/7 \\ y/7 \\ 1 \end{bmatrix}$$

  - *The original Cartesian coordinates are recovered by dividing the first two positions by the third.*
  - *Unlike Cartesian coordinates, a single point can be represented by infinitely many homogeneous coordinates.*

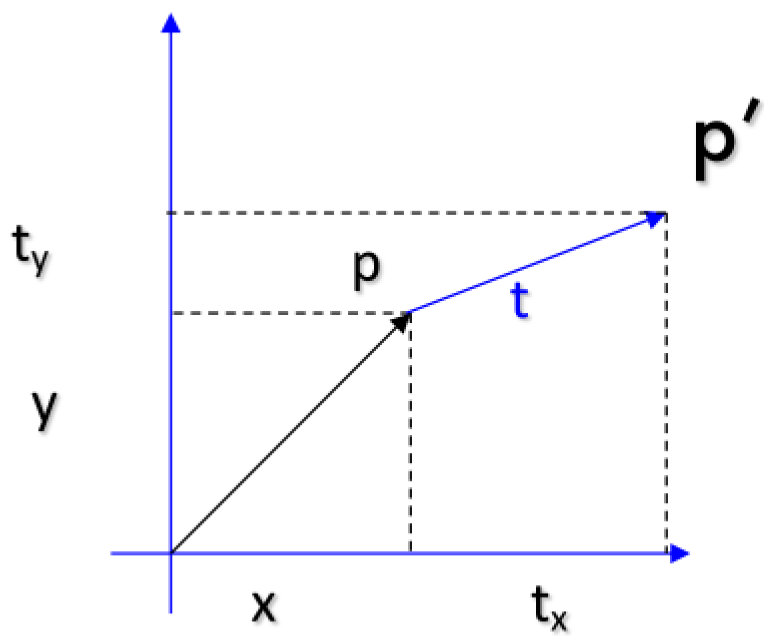
2D transformations

- *Transformation Matrices*
- *Homogeneous coordinates*
- *Translation*
- *Scaling*
- *Rotation*

2D Translation



2D Translation using Homogeneous Coordinates



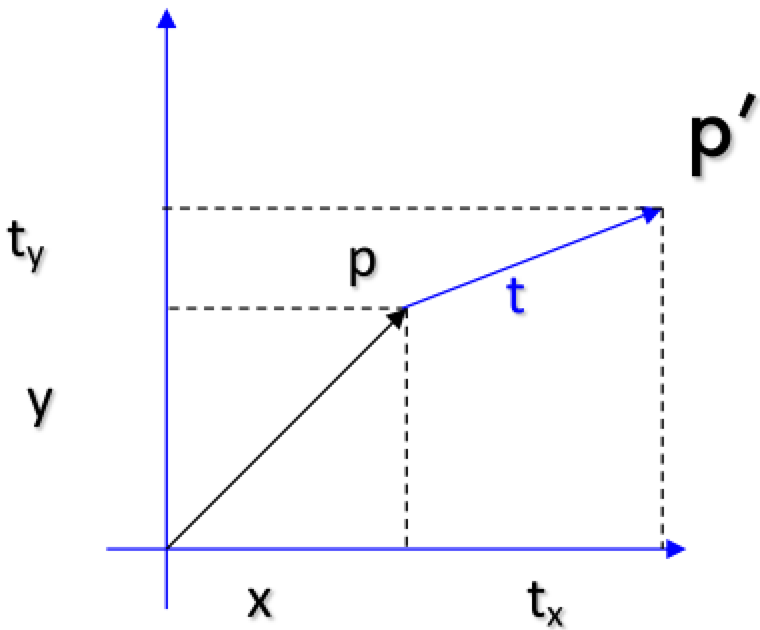
$$p = \begin{bmatrix} x \\ y \end{bmatrix} \rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$t = \begin{bmatrix} t_x \\ t_y \end{bmatrix} \rightarrow \begin{bmatrix} t_x \\ t_y \\ 1 \end{bmatrix}$$

$$p' = Tp$$

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

2D Translation using Homogeneous Coordinates



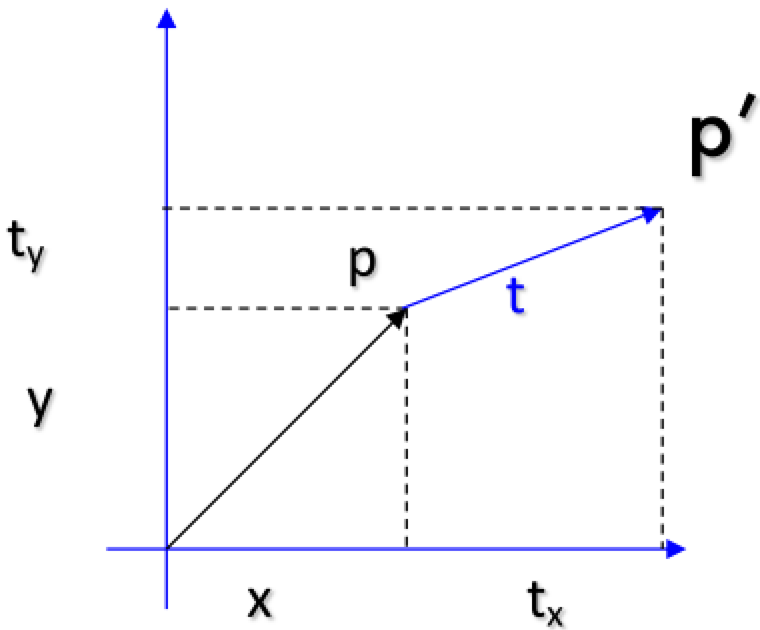
$$p = \begin{bmatrix} x \\ y \end{bmatrix} \rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$t = \begin{bmatrix} t_x \\ t_y \end{bmatrix} \rightarrow \begin{bmatrix} t_x \\ t_y \\ 1 \end{bmatrix}$$

$$p' = Tp$$

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

2D Translation using Homogeneous Coordinates



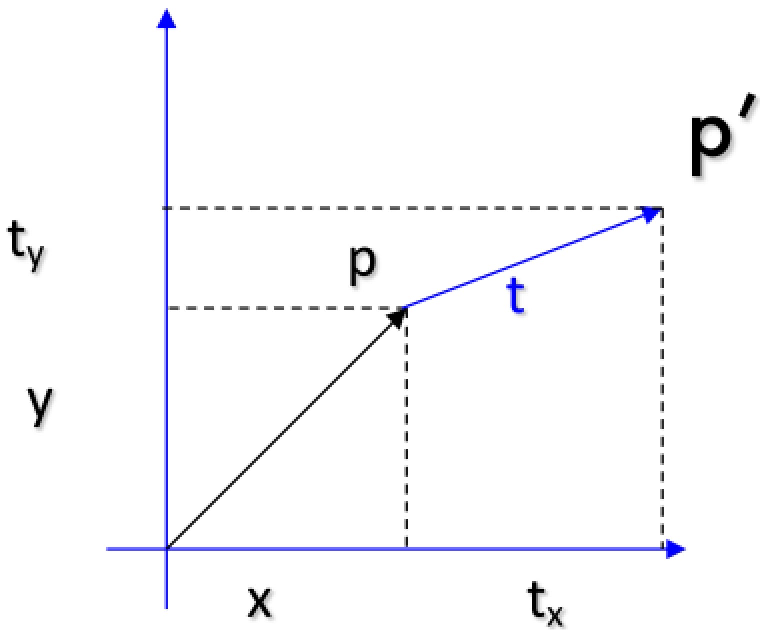
$$p = \begin{bmatrix} x \\ y \end{bmatrix} \rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$t = \begin{bmatrix} t_x \\ t_y \end{bmatrix} \rightarrow \begin{bmatrix} t_x \\ t_y \\ 1 \end{bmatrix}$$

$$p' = Tp$$

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

2D Translation using Homogeneous Coordinates



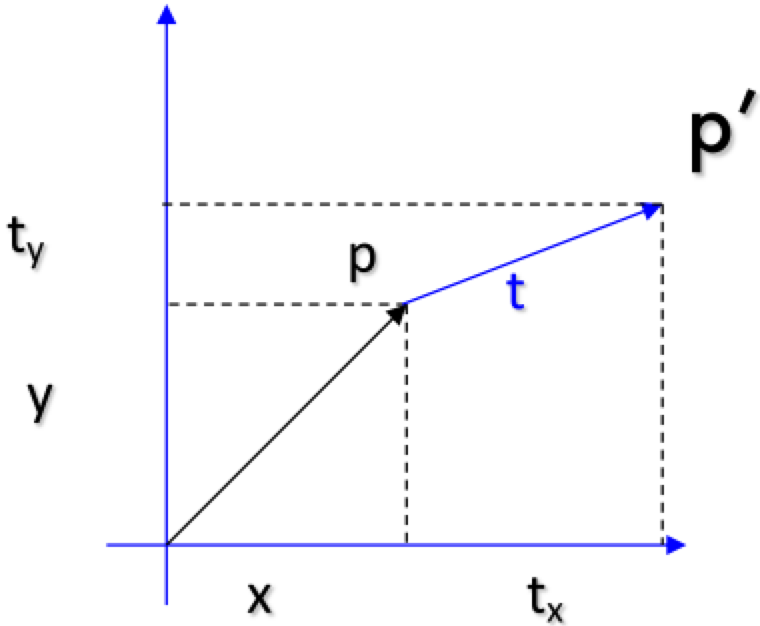
$$p = \begin{bmatrix} x \\ y \end{bmatrix} \rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$t = \begin{bmatrix} t_x \\ t_y \end{bmatrix} \rightarrow \begin{bmatrix} t_x \\ t_y \\ 1 \end{bmatrix}$$

$$p' = Tp$$

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

2D Translation using Homogeneous Coordinates



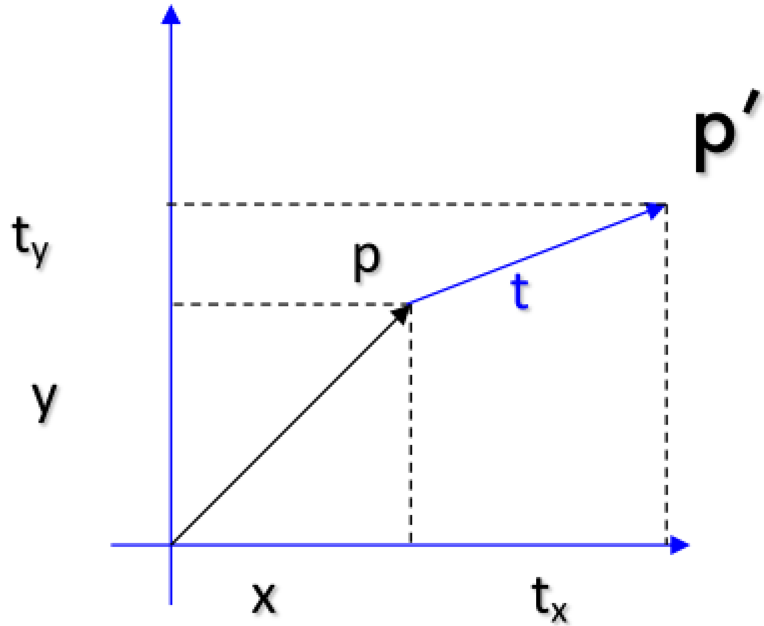
$$p = \begin{bmatrix} x \\ y \end{bmatrix} \rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$t = \begin{bmatrix} t_x \\ t_y \end{bmatrix} \rightarrow \begin{bmatrix} t_x \\ t_y \\ 1 \end{bmatrix}$$

$$p' = Tp$$

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

2D Translation using Homogeneous Coordinates



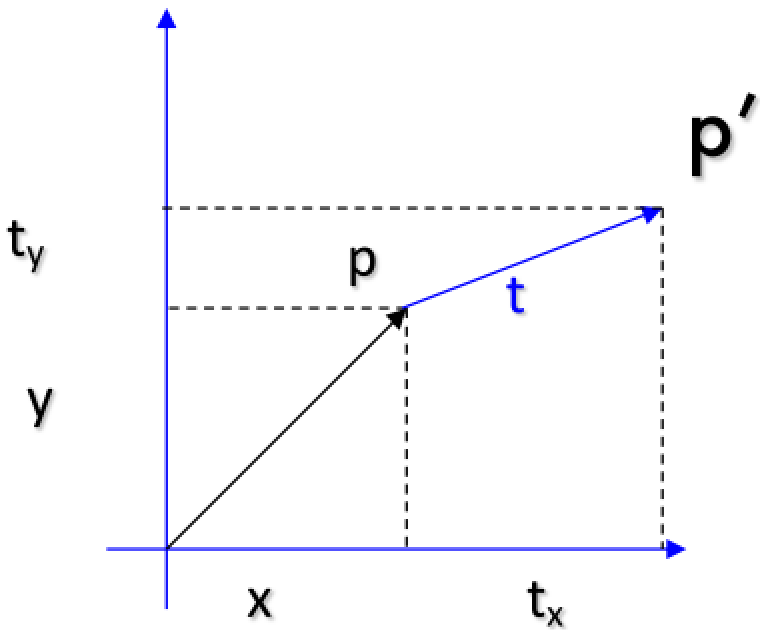
$$p = \begin{bmatrix} x \\ y \end{bmatrix} \rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$t = \begin{bmatrix} t_x \\ t_y \end{bmatrix} \rightarrow \begin{bmatrix} t_x \\ t_y \\ 1 \end{bmatrix}$$

$$p' = Tp$$

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

2D Translation using Homogeneous Coordinates



$$p = \begin{bmatrix} x \\ y \end{bmatrix} \rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$t = \begin{bmatrix} t_x \\ t_y \end{bmatrix} \rightarrow \begin{bmatrix} t_x \\ t_y \\ 1 \end{bmatrix}$$

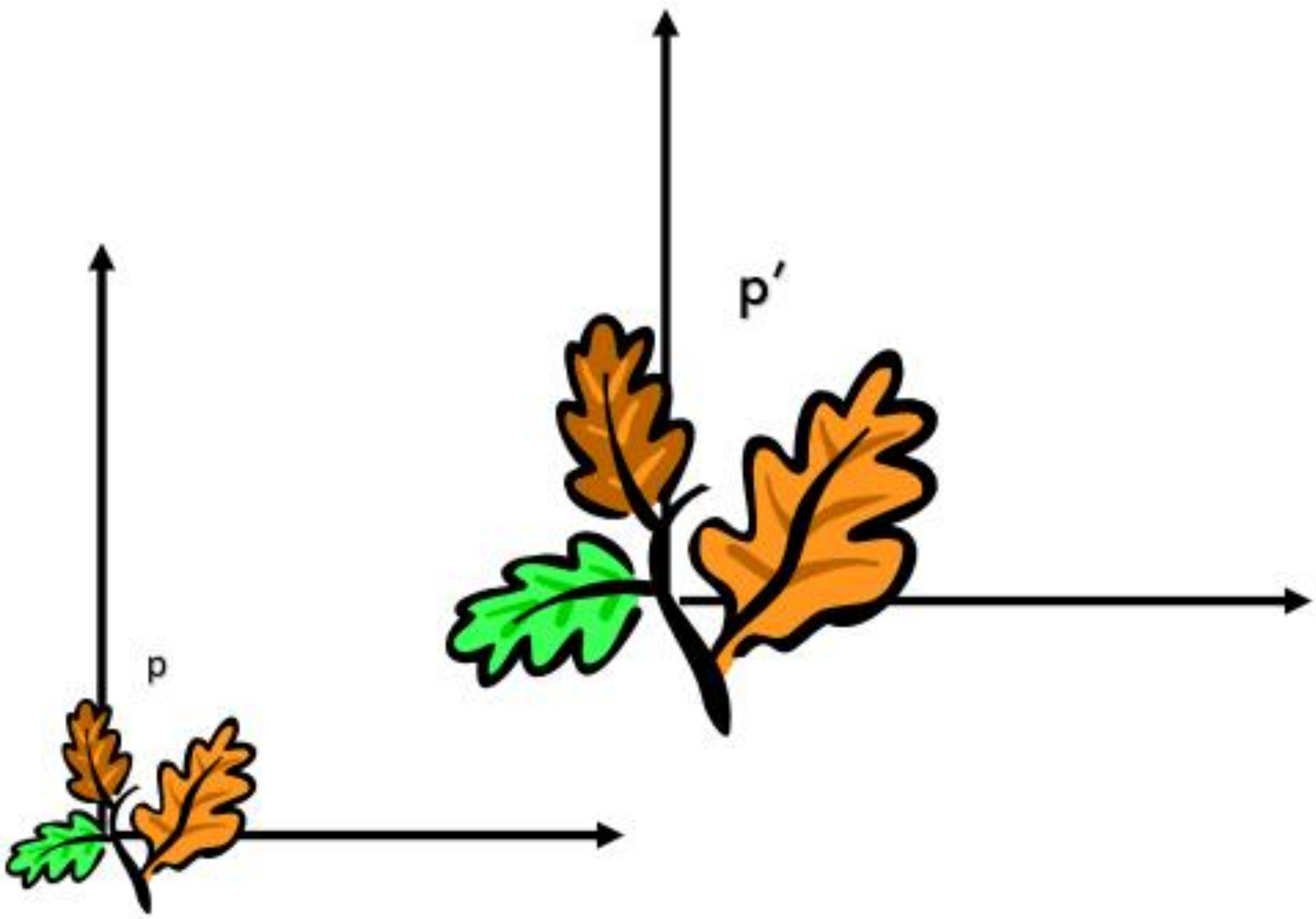
$$p' = Tp$$

$$p' \rightarrow \begin{bmatrix} x + t_x \\ y + t_y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{t} \\ \boldsymbol{0} & 1 \end{bmatrix} p = Tp$$

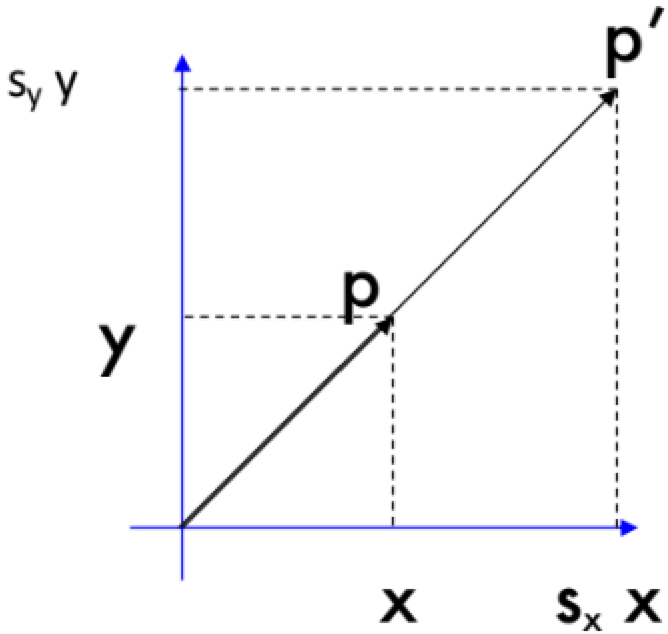
2D transformations

- *Transformation Matrices*
- *Homogeneous coordinates*
- *Translation*
- *Scaling*
- *Rotation*

Scaling



Scaling Equation



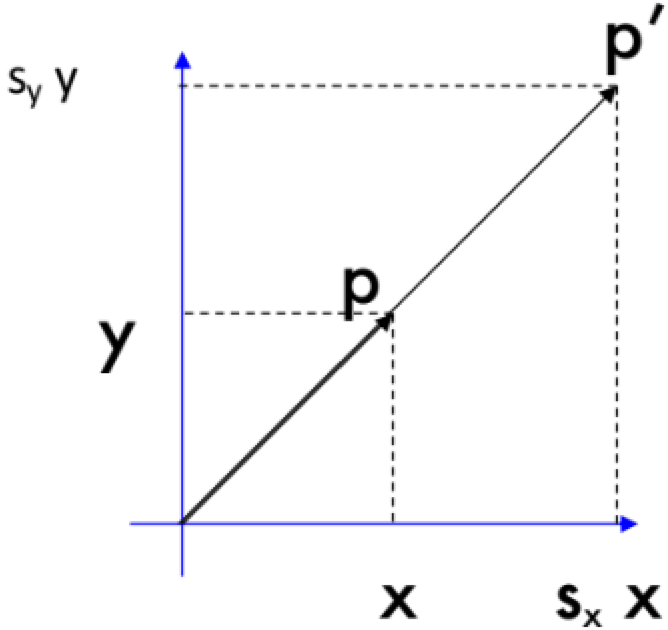
$$p = \begin{bmatrix} x \\ y \end{bmatrix} \rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$p' = \begin{bmatrix} s_x x \\ s_y y \end{bmatrix} \rightarrow \begin{bmatrix} s_x x \\ s_y y \\ 1 \end{bmatrix}$$

$$p' = Sp$$

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

Scaling Equation



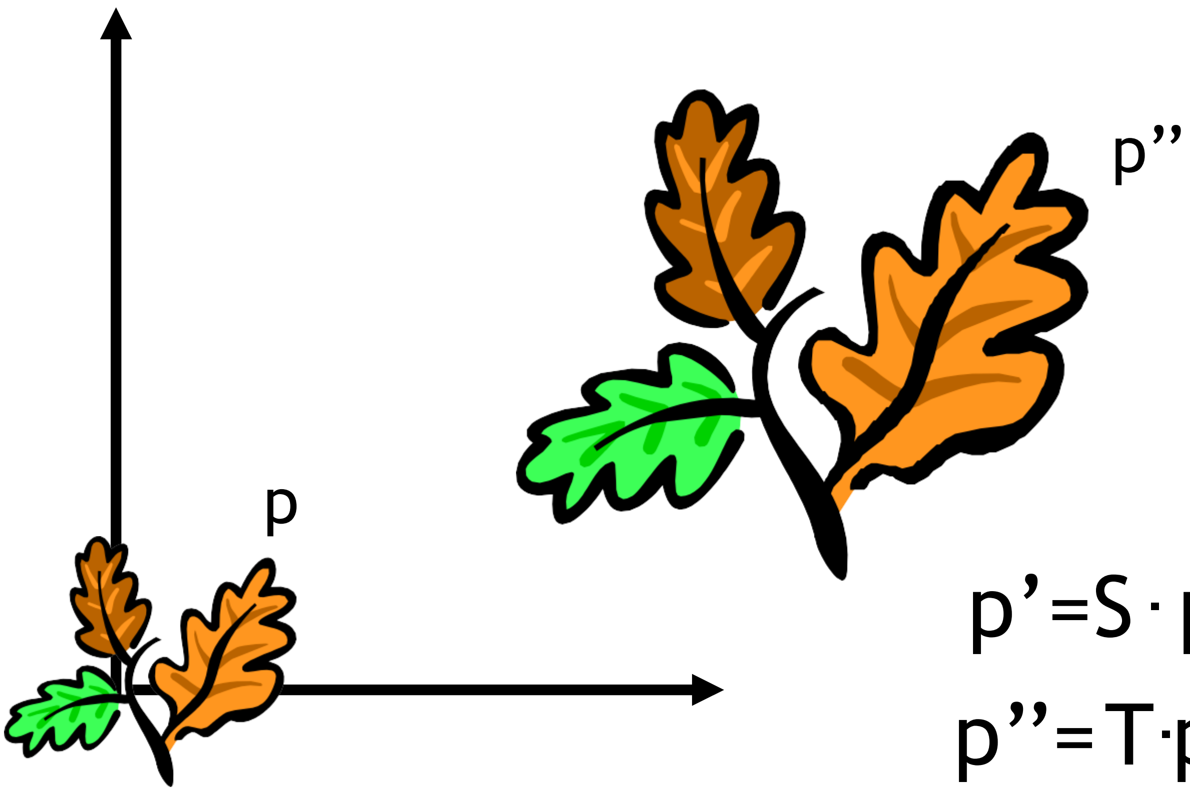
$$p = \begin{bmatrix} x \\ y \end{bmatrix} \rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$p' = \begin{bmatrix} s_x x \\ s_y y \end{bmatrix} \rightarrow \begin{bmatrix} s_x x \\ s_y y \\ 1 \end{bmatrix}$$

$$p' = Sp$$

$$p' \rightarrow \begin{bmatrix} s_x x \\ s_y y \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} S' & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} p = Sp$$

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

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

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

Scaling & Translating != Translating & Scaling

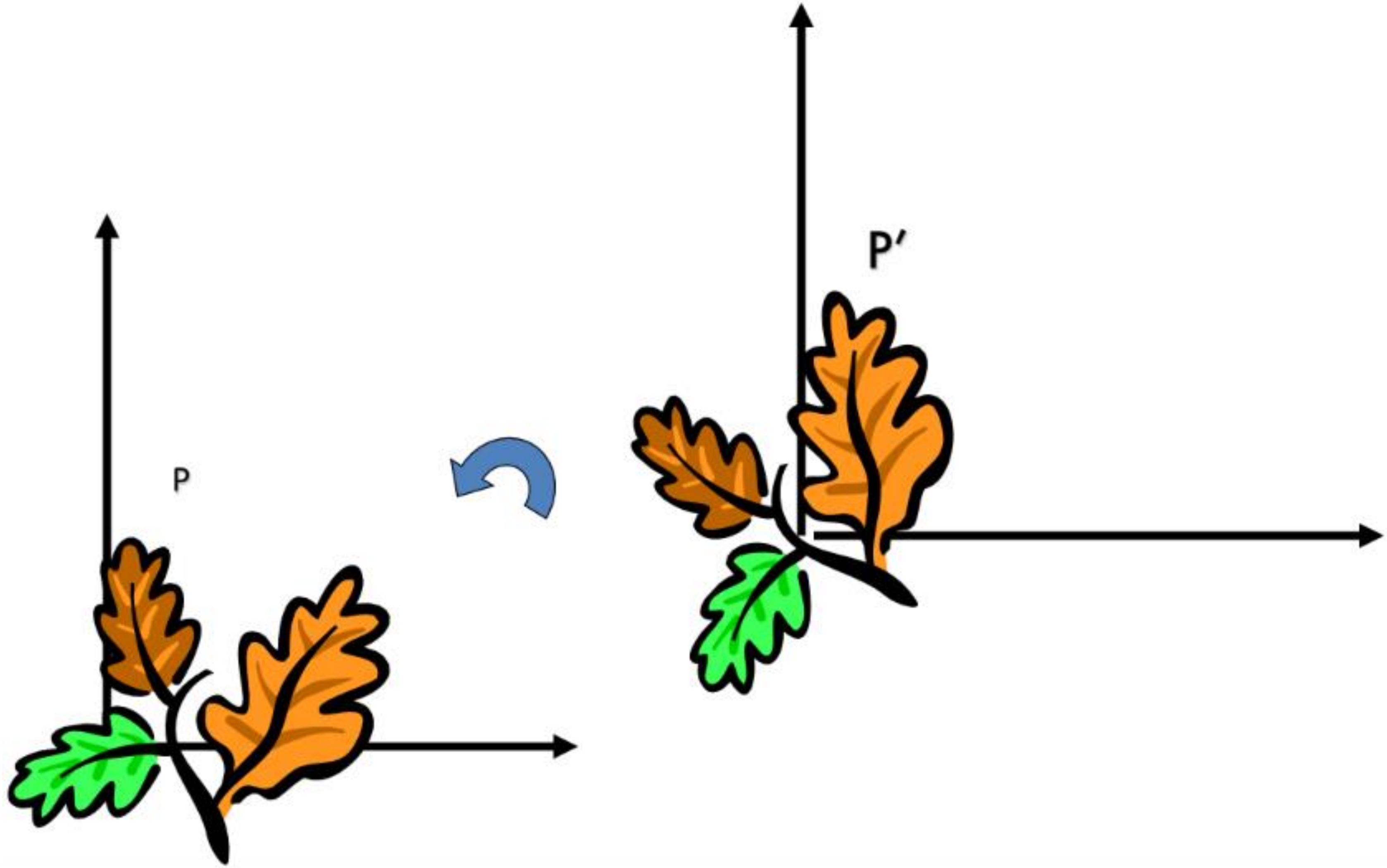
$$p'' = TSp = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & t_x \\ 0 & s_y & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}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

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

2D transformations

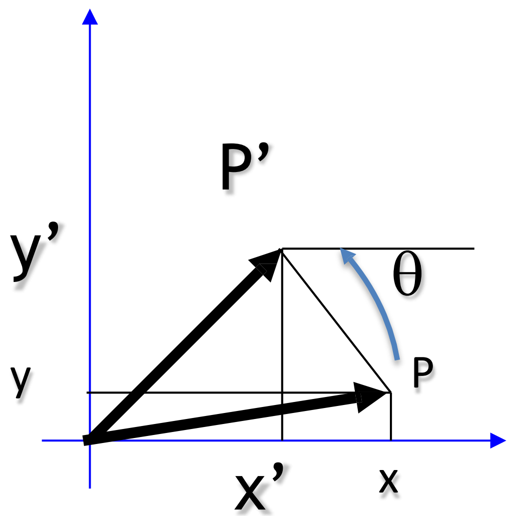
- *Transformation Matrices*
- *Homogeneous coordinates*
- *Translation*
- *Scaling*
- *Rotation*

Rotation



2D Rotation Matrix Formula

Counter-clockwise rotation by an angle θ



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

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

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

$$p' = Rp$$

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:

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

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

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)*

Scaling + Rotation + Translation

p'= (T R S) p

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

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

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

This is the form of the general-purpose transformation matrix

Summary

- *2D transformations*
  - *–Transformation Matrices*
  - *–Homogeneous coordinates*
  - *–Translation*
  - *–Scaling*
  - *–Rotation*

Next time:

Color and Filters

Pattern Recognition (in Computer Vision)

Jinhua Ma,

School of Computer Science and Engineering, Sun Yat-Sen University