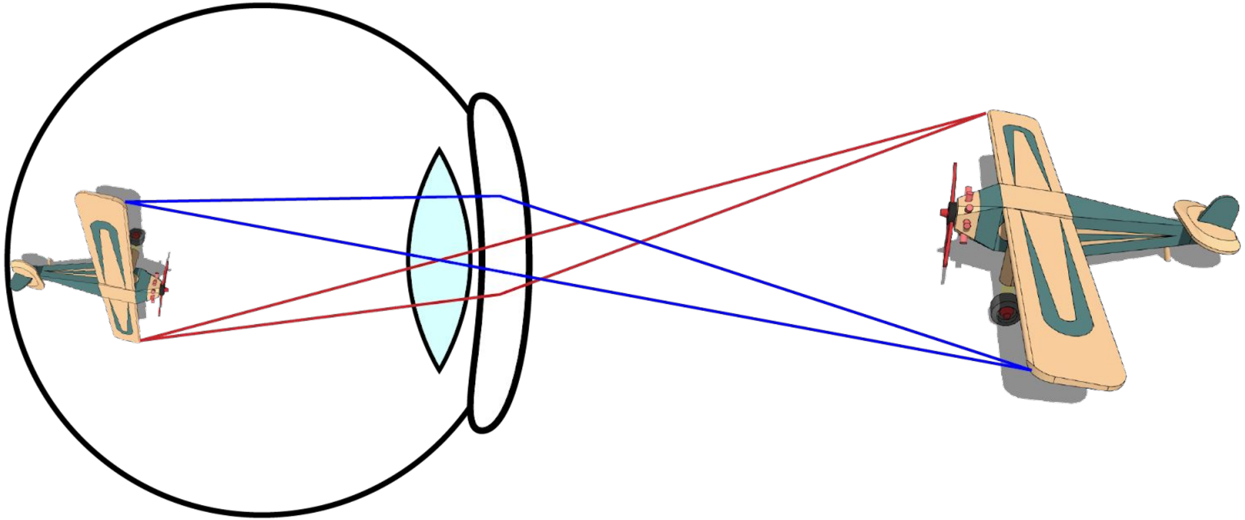
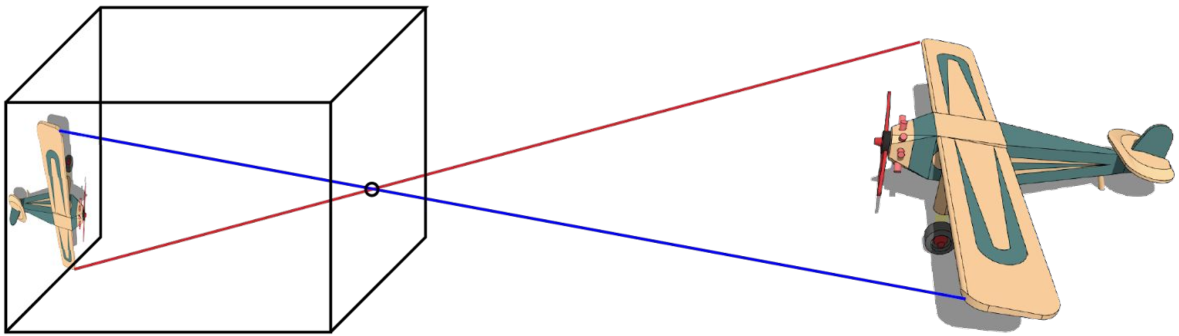
Image Basics

Eyes: projection onto retina



Model: pinhole camera



Model: pinhole camera

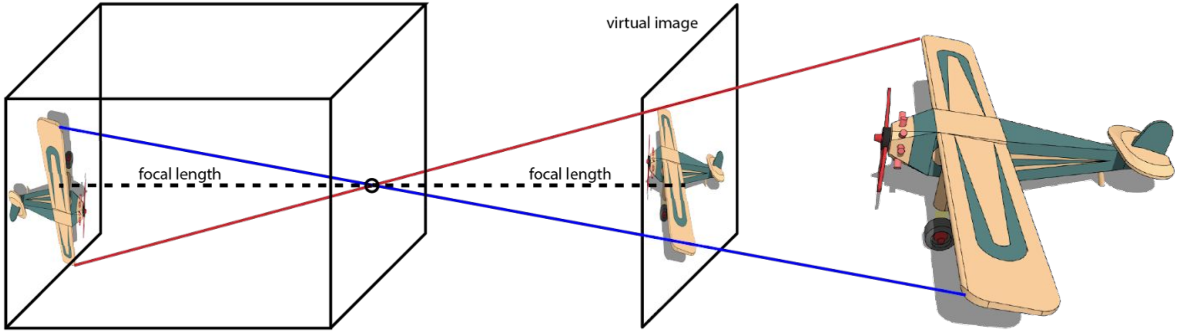


Image: 3d -> 2d projection of the world

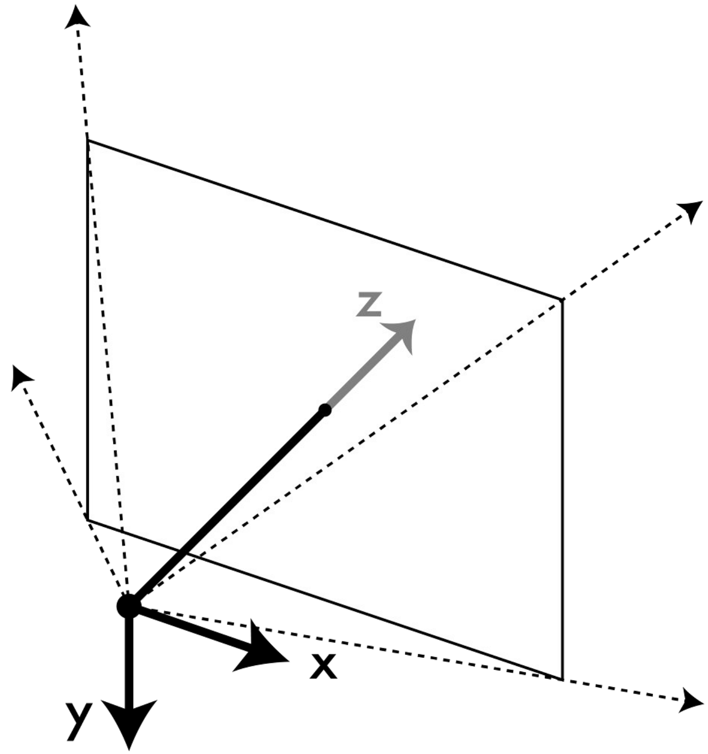


Image: 3d -> 2d projection of the world

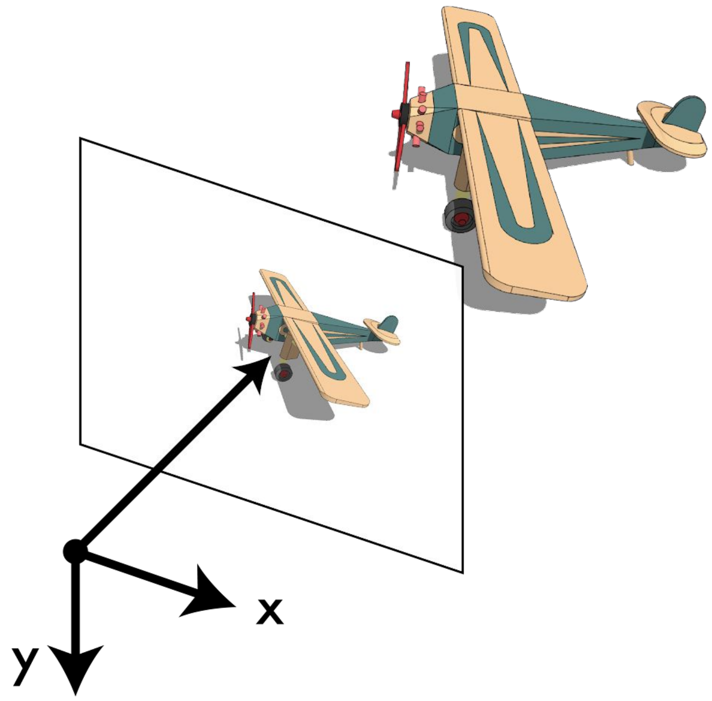


Image: 3d -> 2d projection of the world

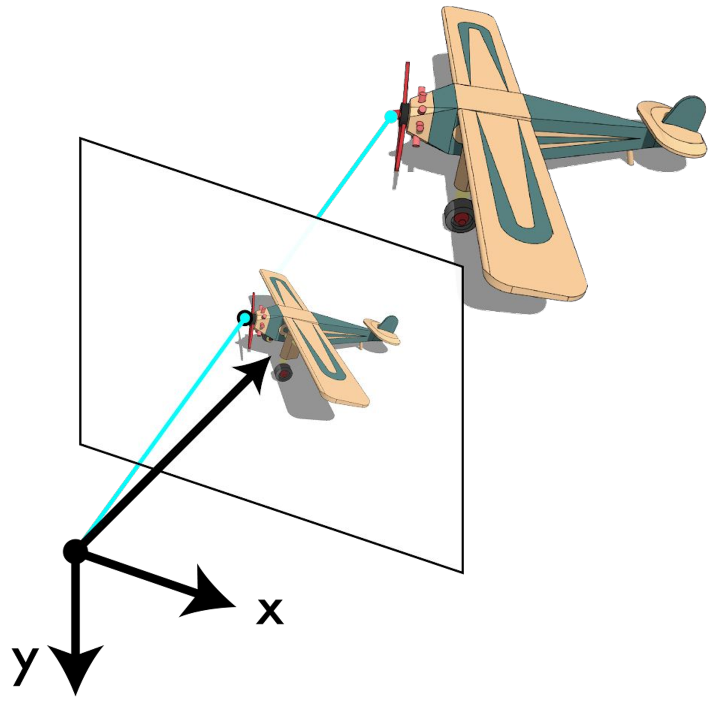


Image: 3d -> 2d projection of the world

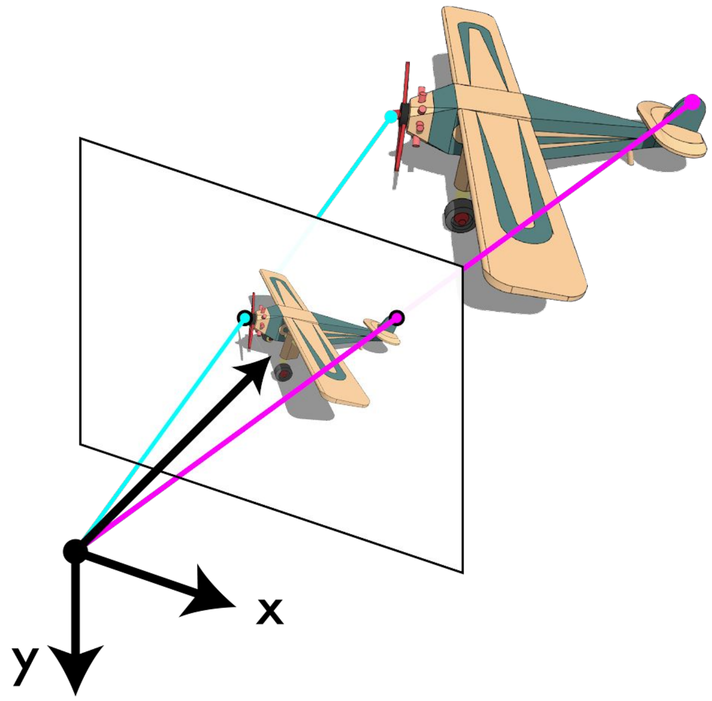


Image: 3d -> 2d projection of the world

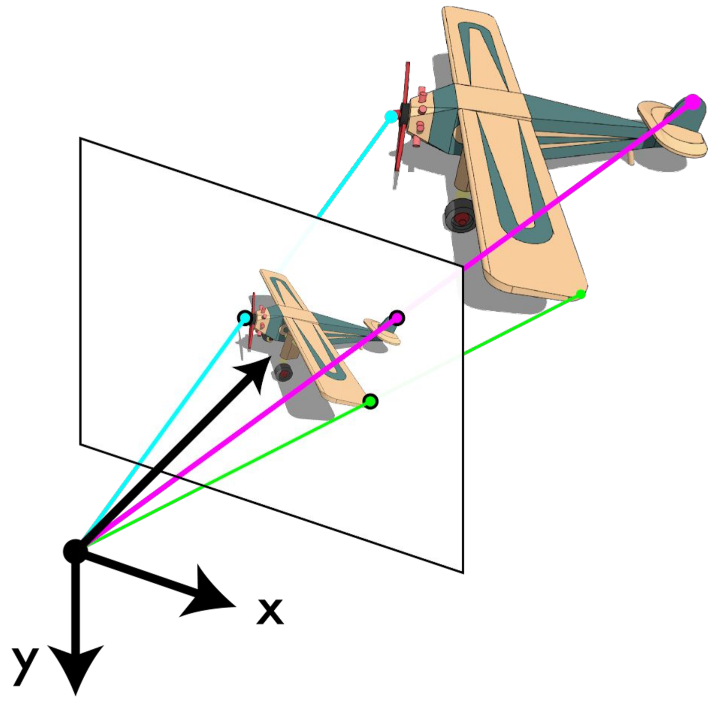
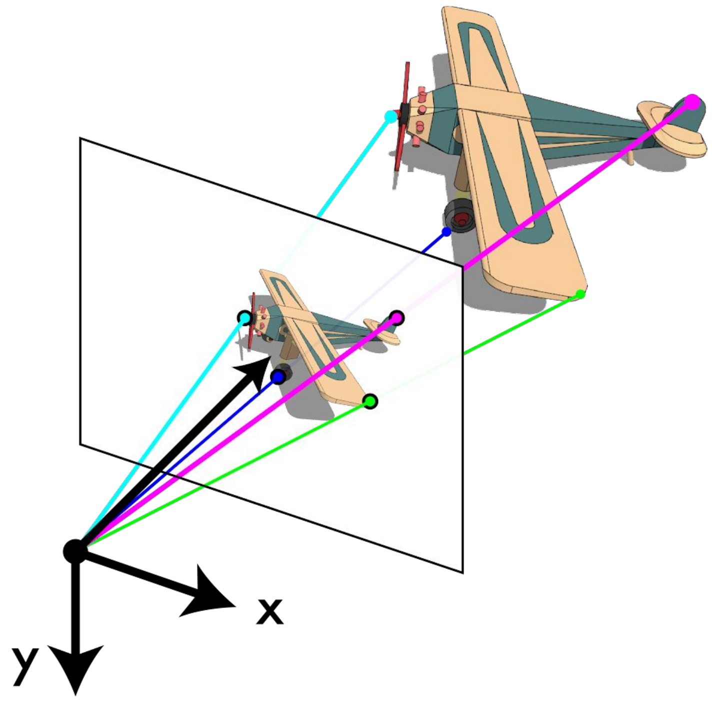
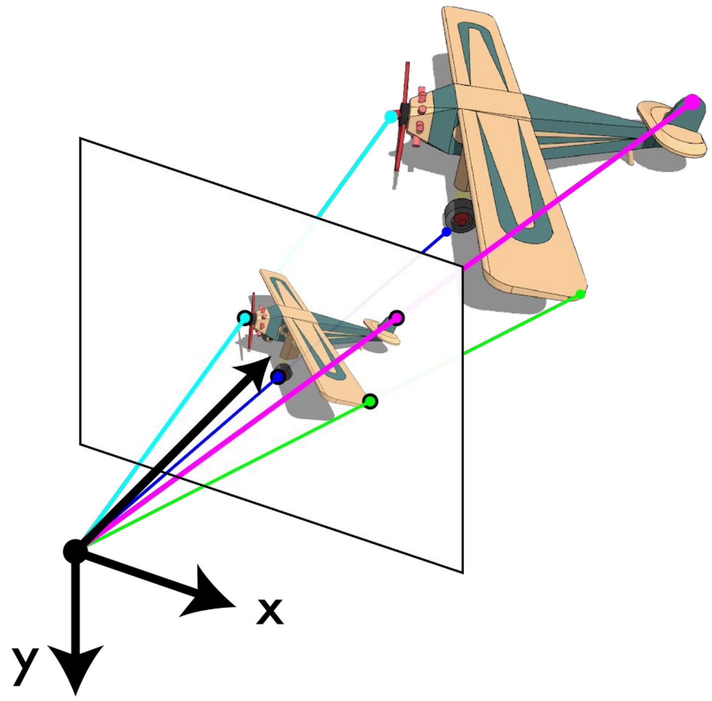


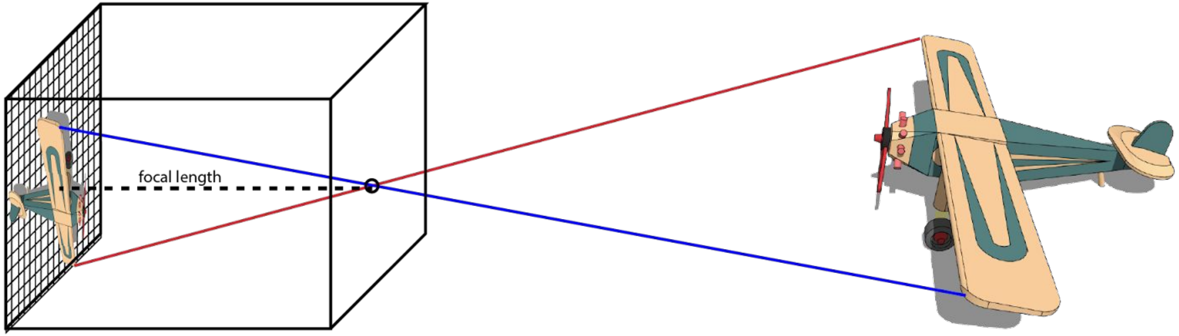
Image: 3d -> 2d projection of the world



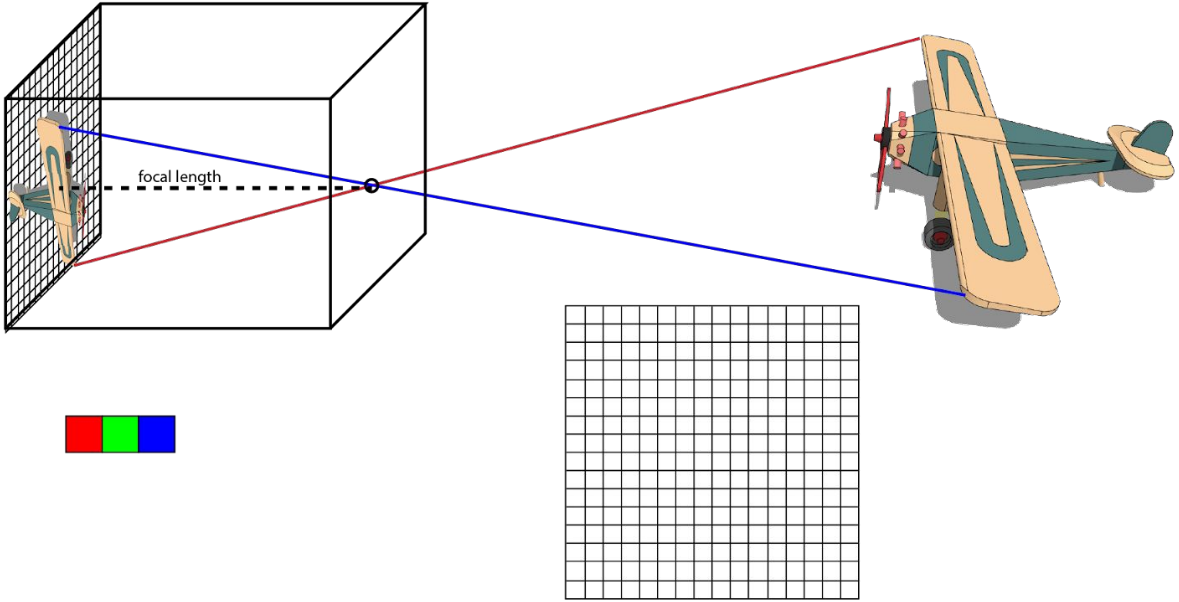
At each point we record incident light



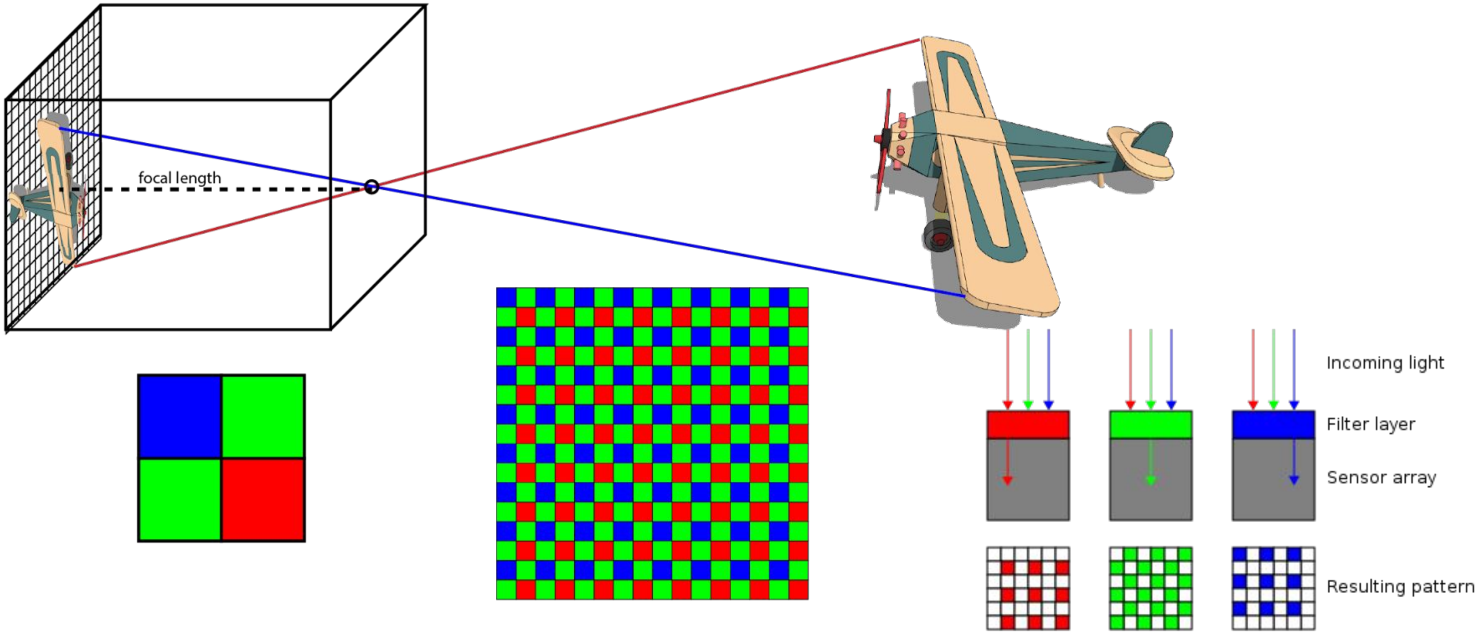
At each point we record incident light



How do we record color?



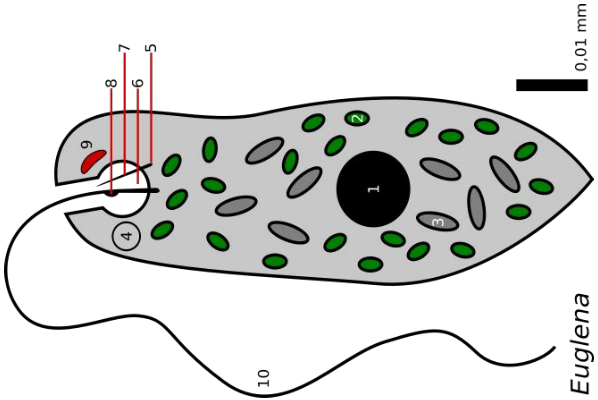
Bayer pattern for CMOS sensors



Evolution’s Big Bang

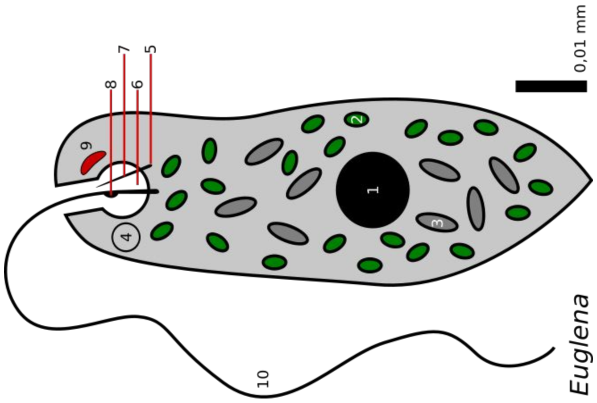
543 million years, B.C

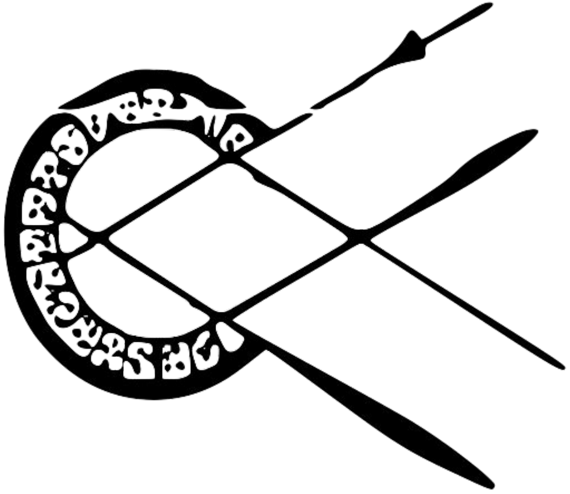
- Euglena
- Photosynthesis
- No nerves, brain, or processing

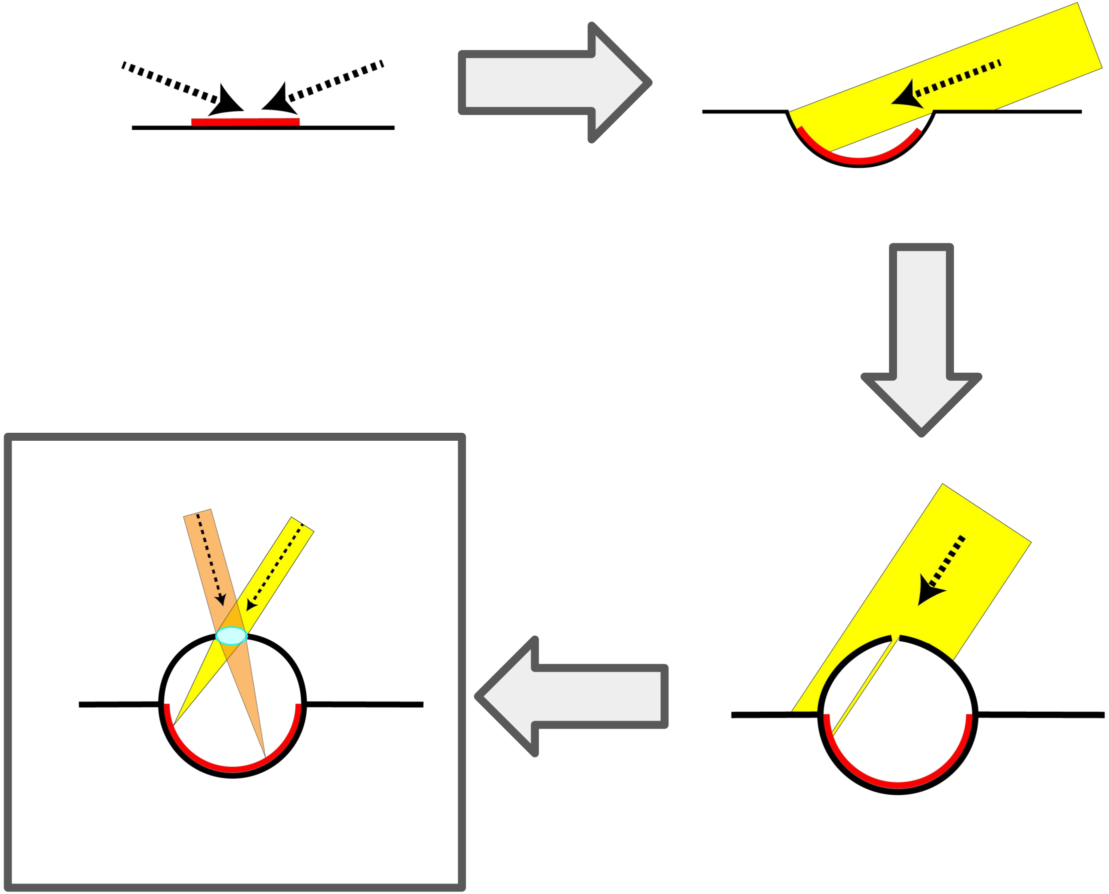


- Euglena
- Photosynthesis
- No nerves, brain, or processing

- Pit eyes
- Photosensitive cells in pits
- Block some light

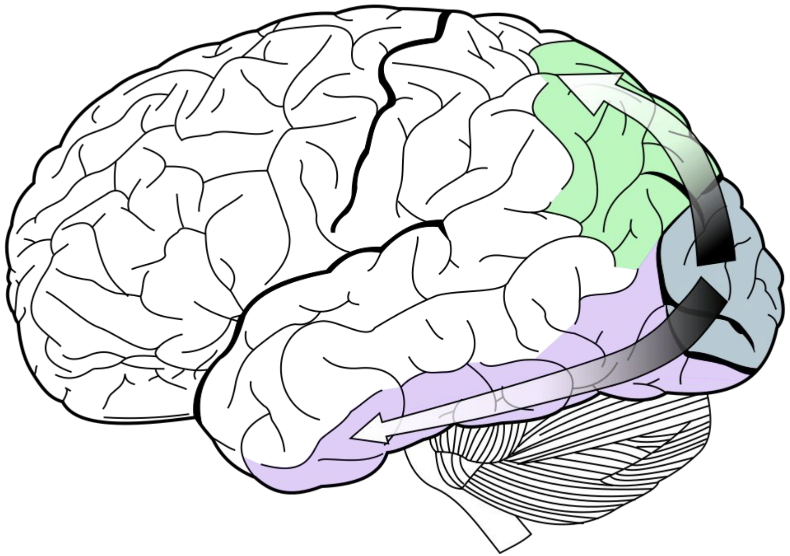




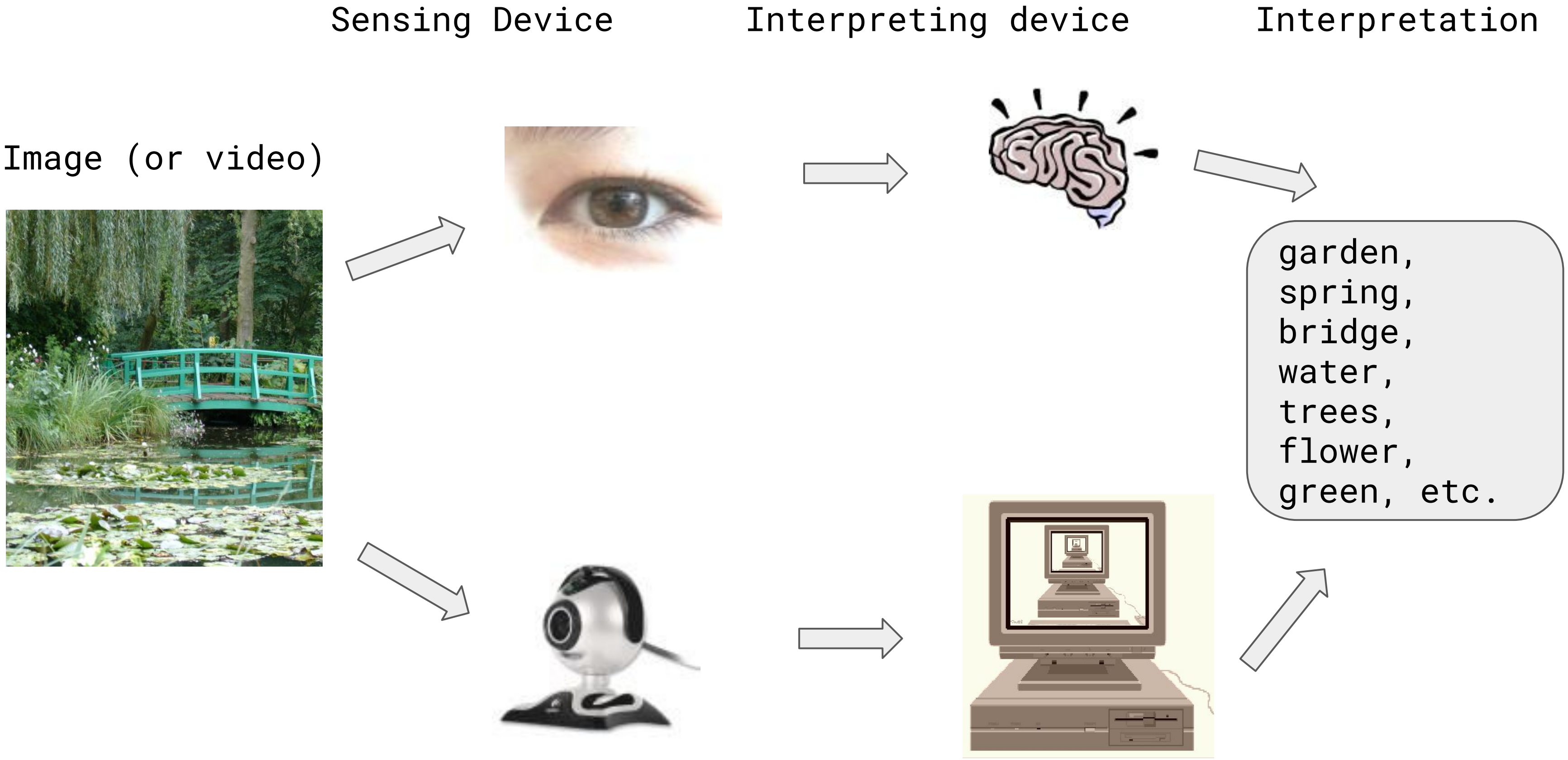


Ventral/dorsal split

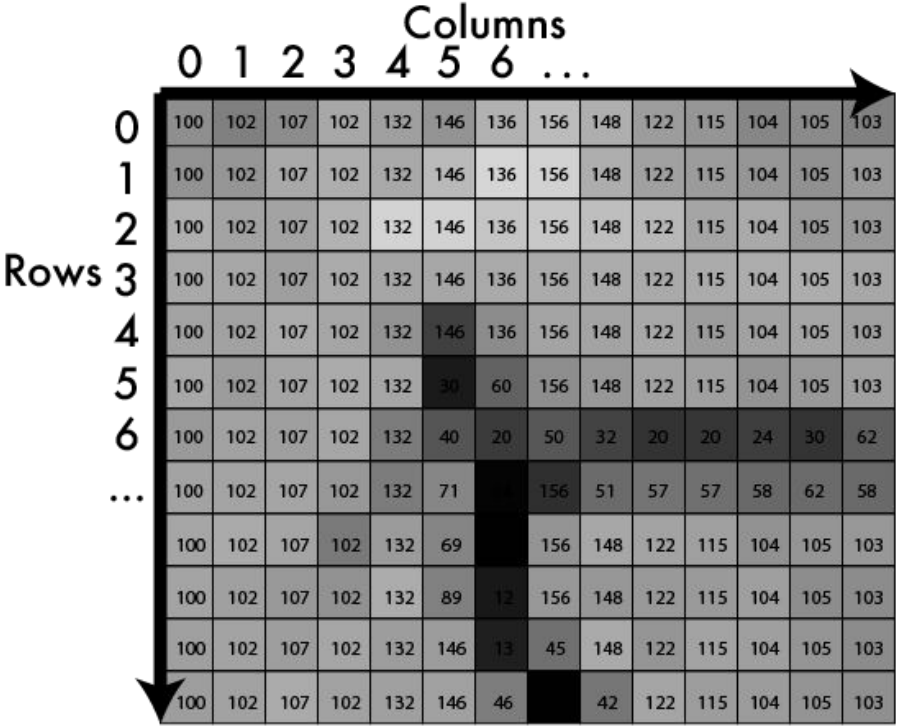
- Ventral: fine grained visual recognition
- Dorsal: vision related to motion and planning
- The brain is integral to human vision, even when we are conscious of it!



- Recognition and action are split!
- Damage to dorsal system:
  - Can recognize objects
  - Poor visual control for tasks like grasping
- Damage to ventral system
  - Cannot recognize objects
  - Can still manipulate them, grasping, etc.
- Much of the information in the dorsal system is not consciously accessible


Sensing Device
Interpreting device
Interpretation
Image (or video)
garden,
spring,
bridge,
water,
trees,
flower,
green, etc.

Digital Images are Matrix



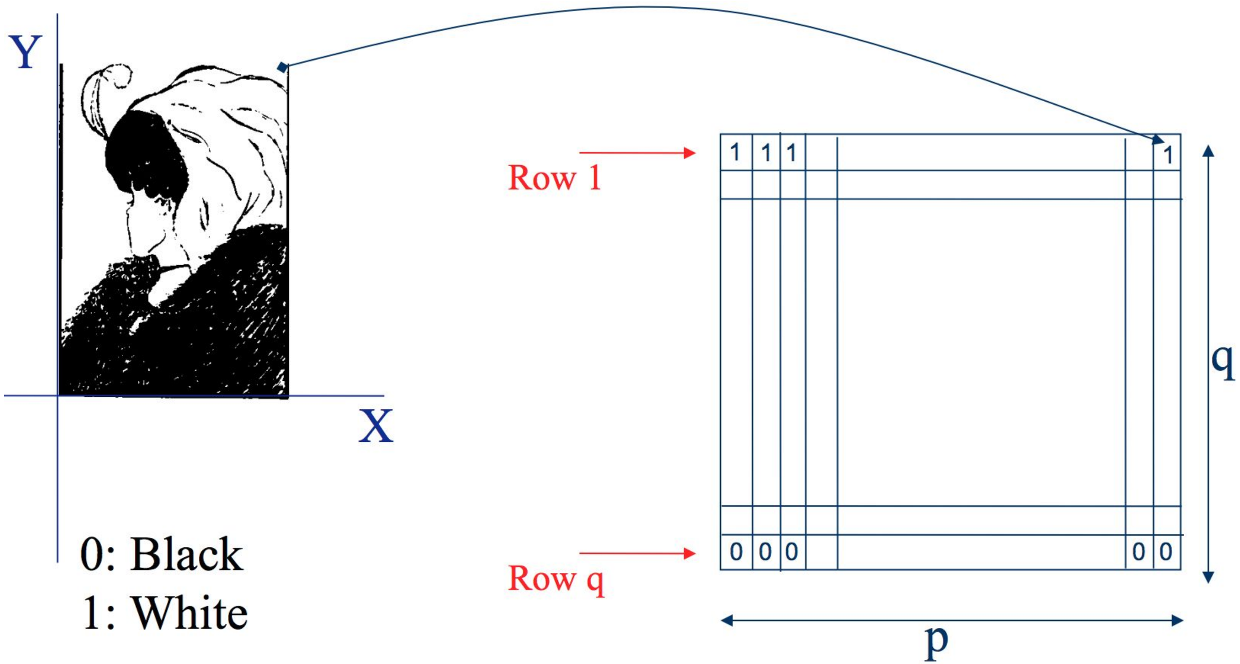
Types of Images

Binary

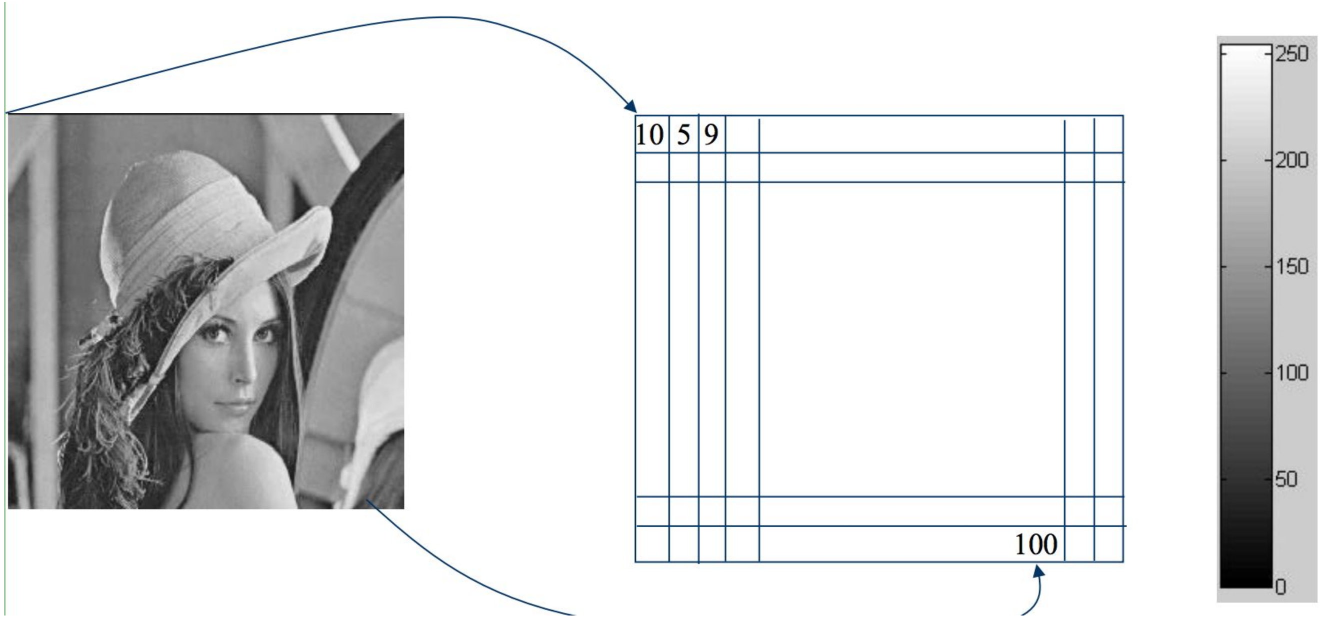
Gray Scale

Color

Binary Image

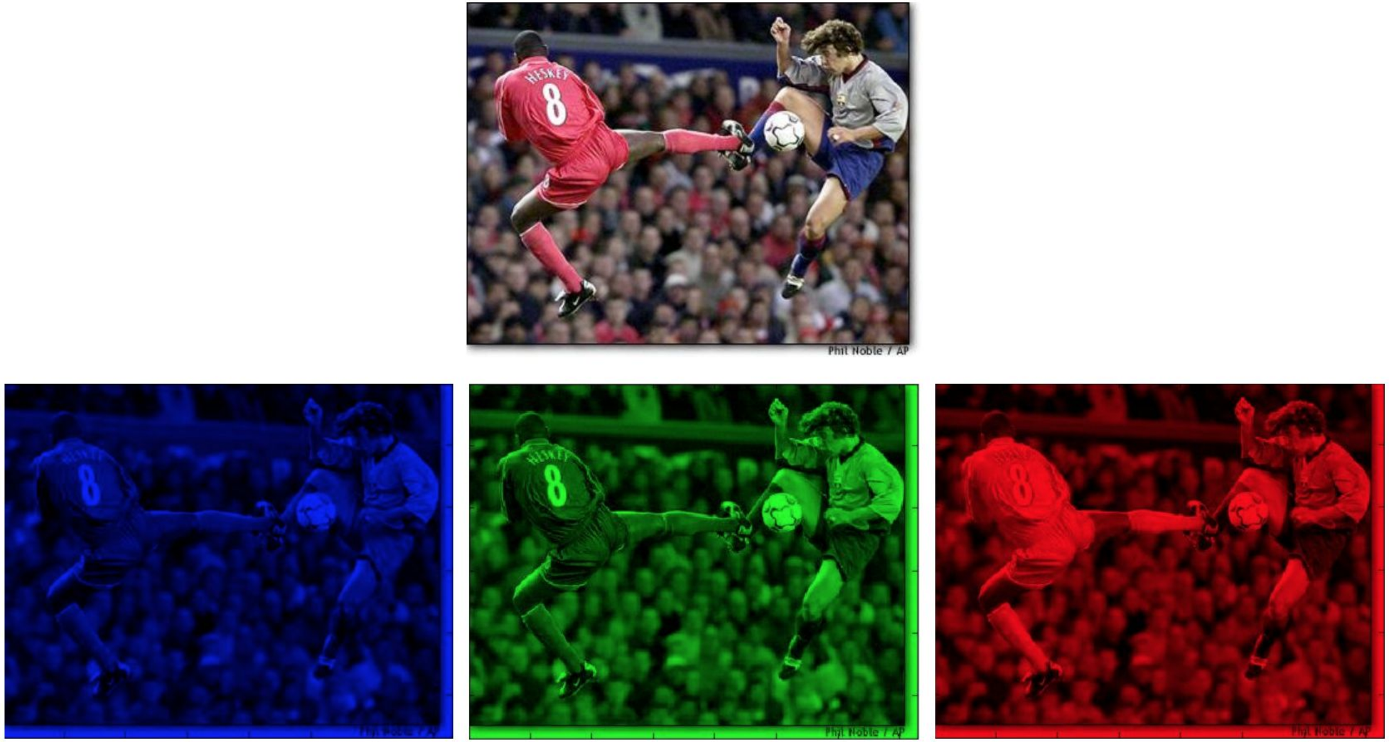


Gray Scale Image

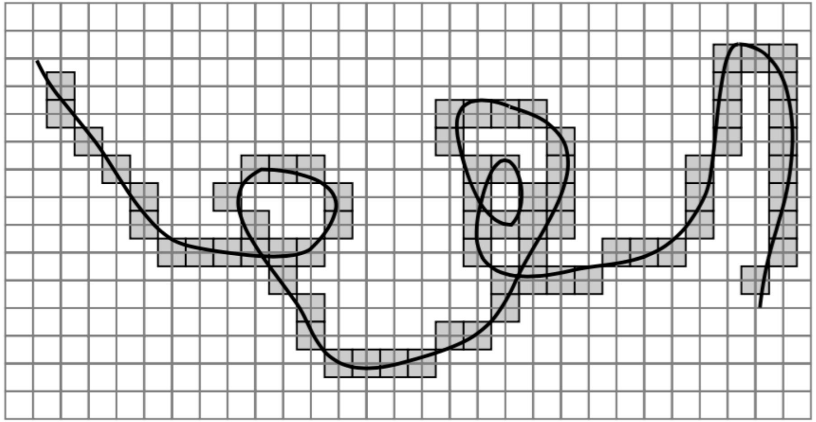


Color Image - On Channel

3 Channels - RGB



Images are sampled

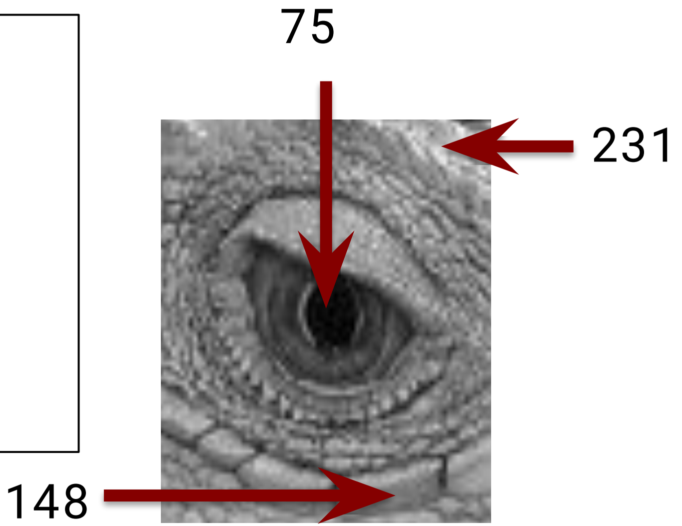


What happens when we zoom into the images we capture?

Resolution is a **sampling** parameter, defined in dots per inch (DPI) or equivalent measures of spatial pixel density, and its standard value for recent screen technologies is 72 dpi.

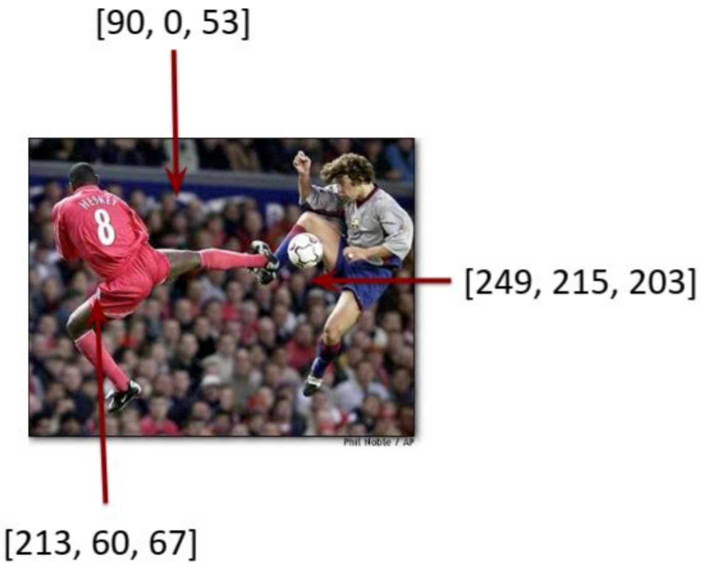
Images are Sampled and Quantized

- An image contains discrete number of pixels
  - A simple example
  - Pixel value:
    - "grayscale" (or "intensity"): [0,255]



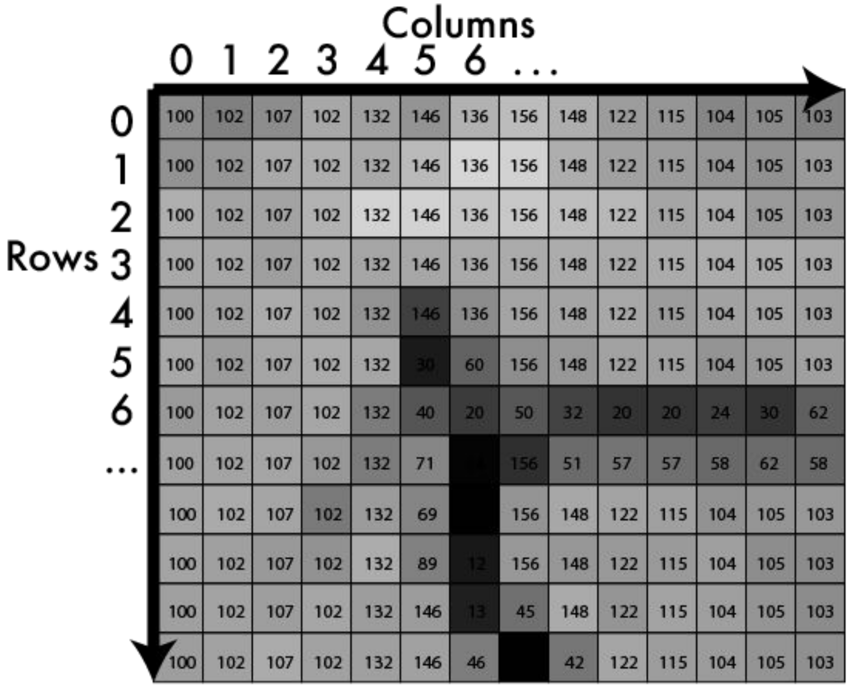
Cont'd : Images are Sampled and Quantized

- An image contains discrete number of pixels
  - A simple example
  - Pixel value:
    - "grayscale"
      (or "intensity"): [0,255]
    - "color"
      - RGB: [R, G, B]
      - Lab: [L, a, b]
      - HSV: [H, S, V]



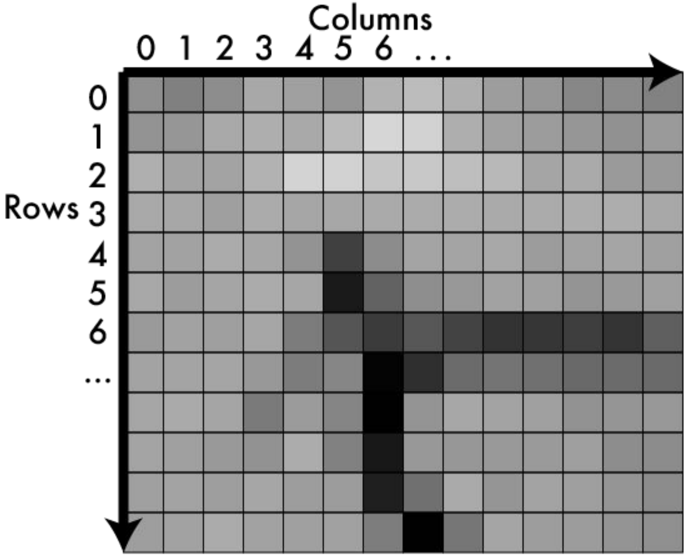
Values in matrix = how much light

- Higher = more light
- Lower = less light
- Bounded
  - No light = 0
  - Sensor/device limit = max
  - Typical ranges:
    - [0-255], fit into byte
    - [0-1], floating point
- Called pixels

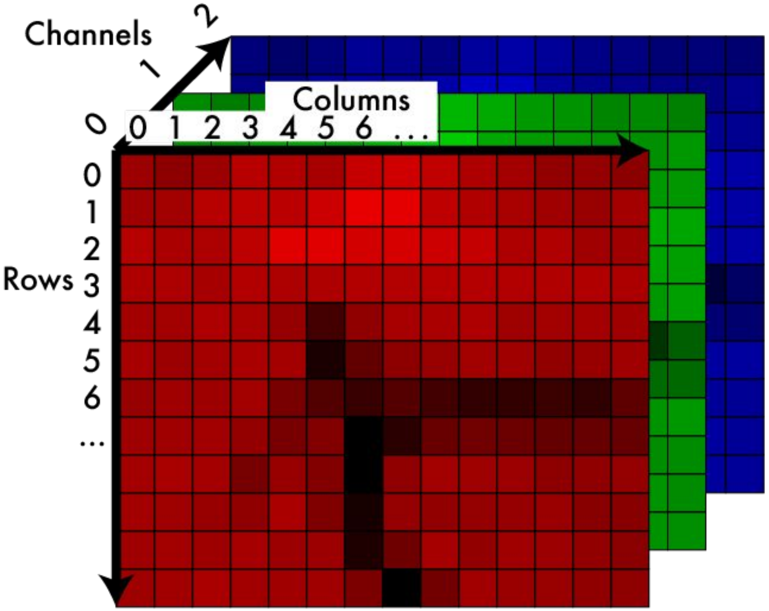


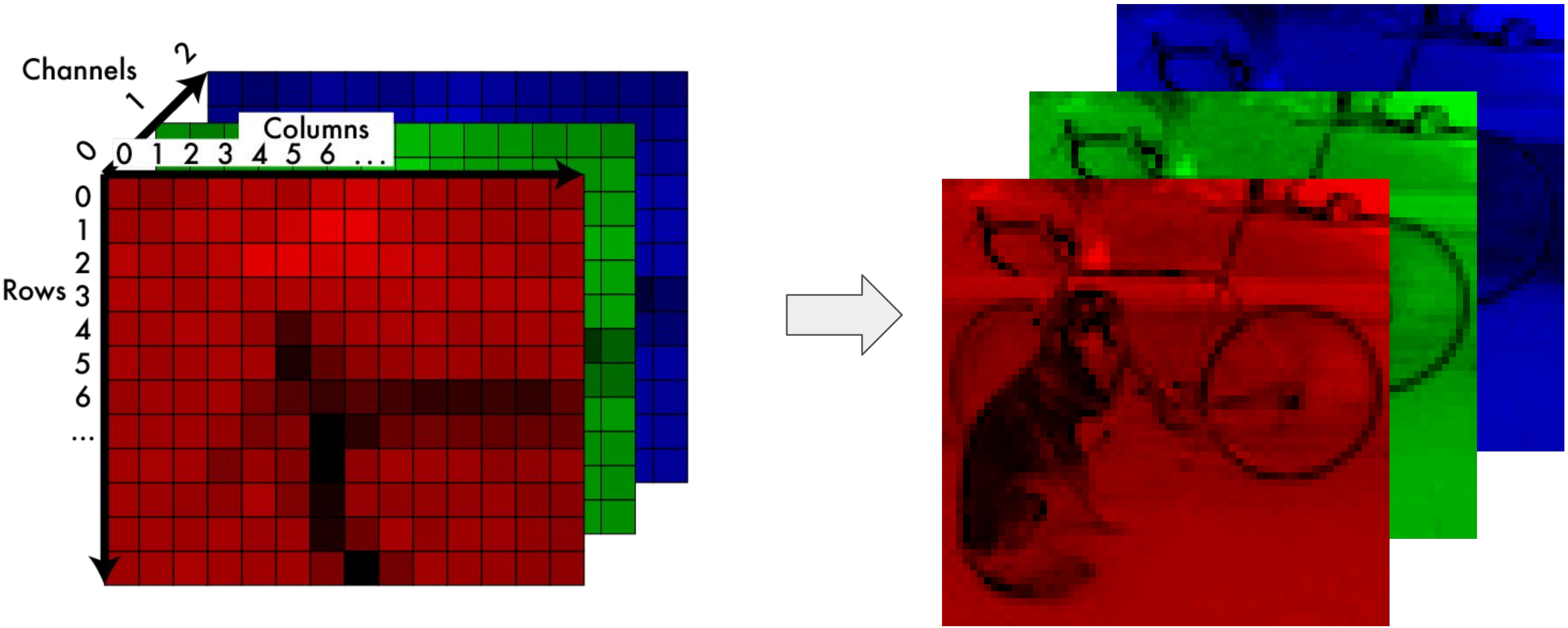
Addressing pixels

- Ways to index:
  - (x,y)
    - Like cartesian coordinates
    - (3,6) is column 3 row 6
  - (r,c)
    - Like matrix notation
    - (3,6) is row 3 column 6
- I use (x,y)
  - So does your homework!
  - Arbitrary
  - Only thing that matters is consistency

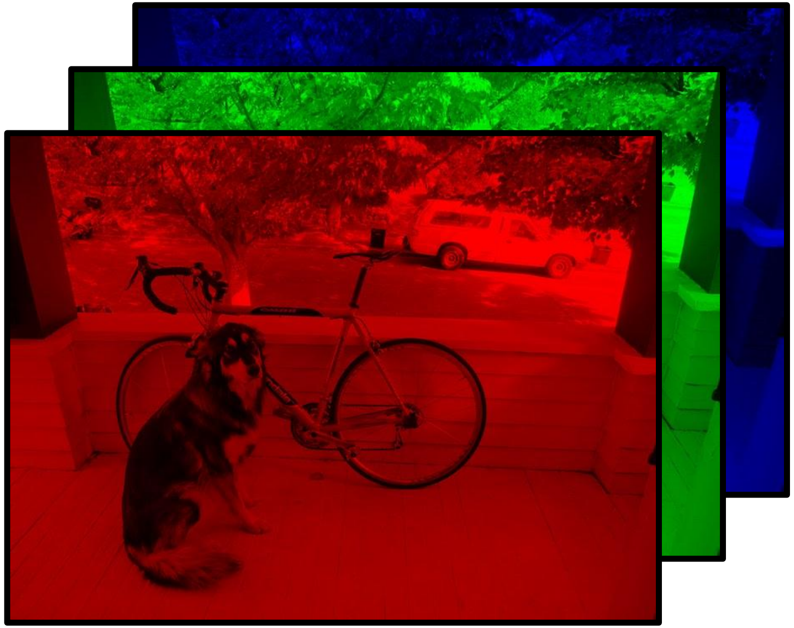


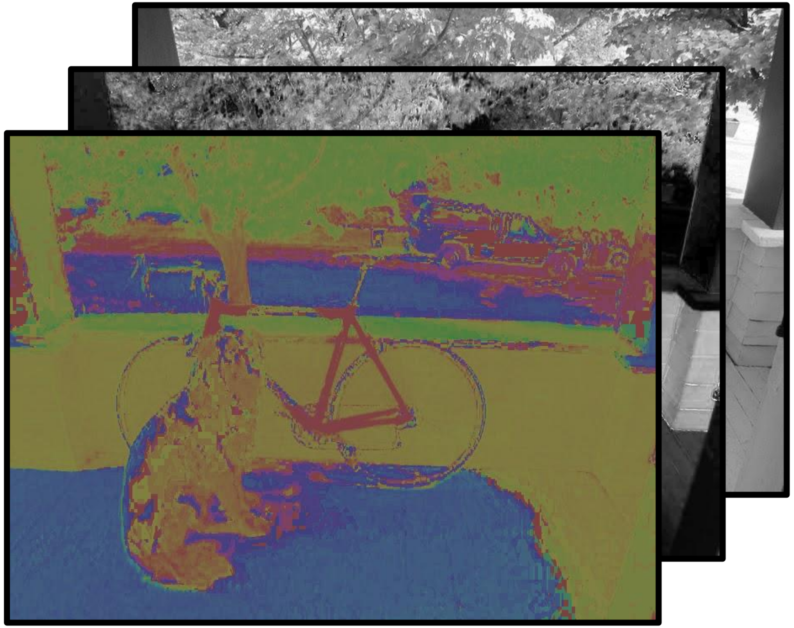
Color image: 3d tensor in colorspace



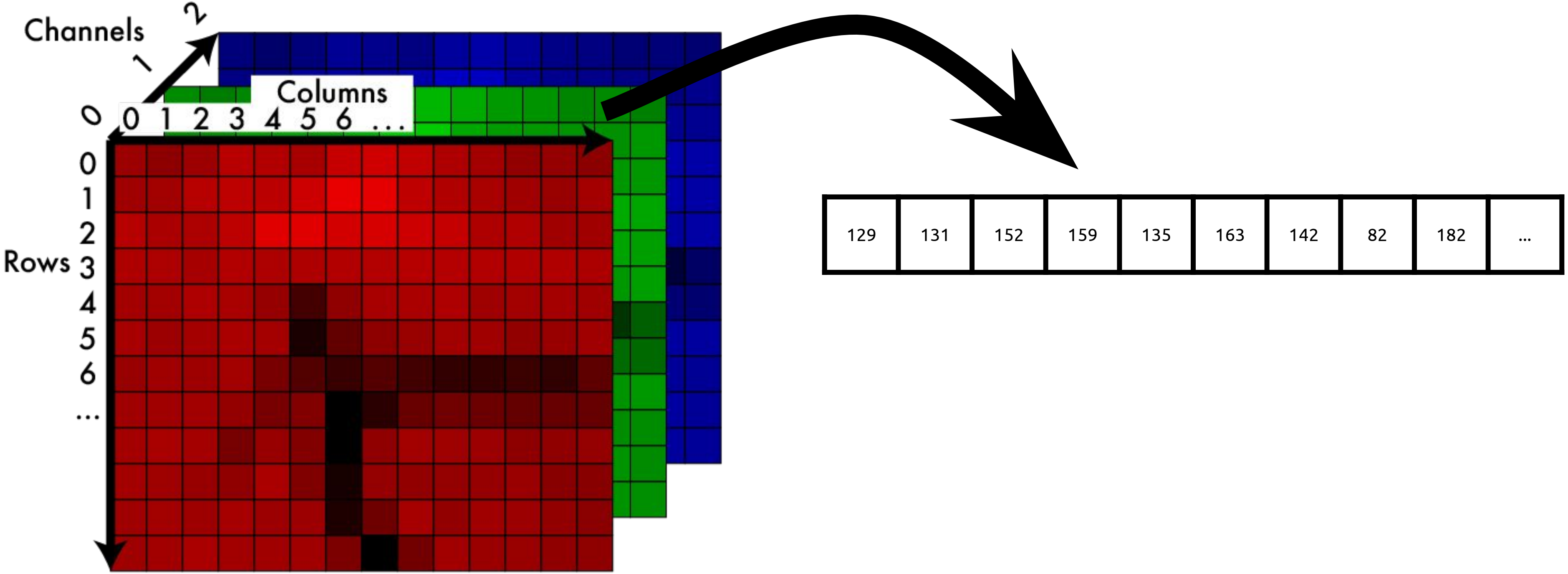

Channels
0
1
2
Columns
0 1 2 3 4 5 6 ...
Rows
0
1
2
3
4
5
6
...

Still 3d tensor, different info - HSV

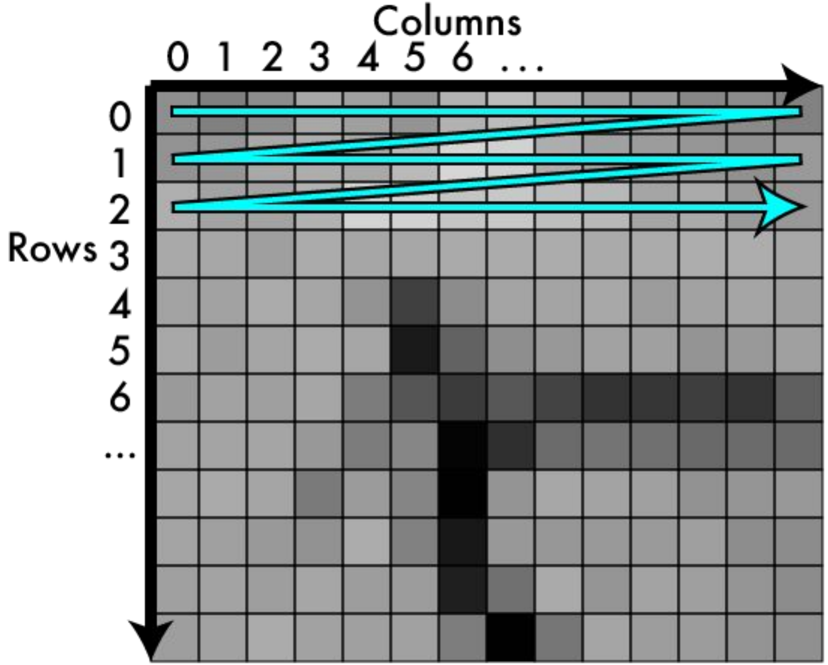


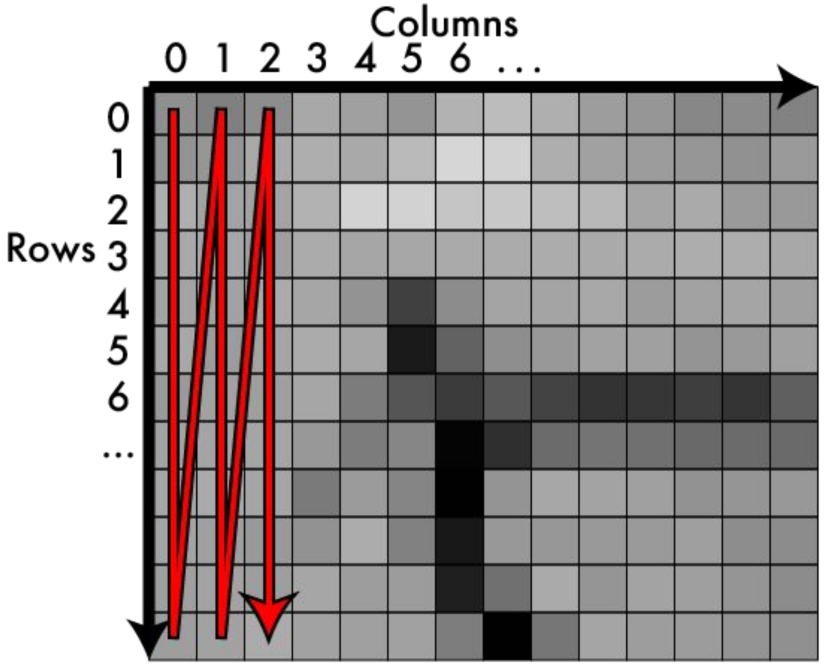


How do we store them?



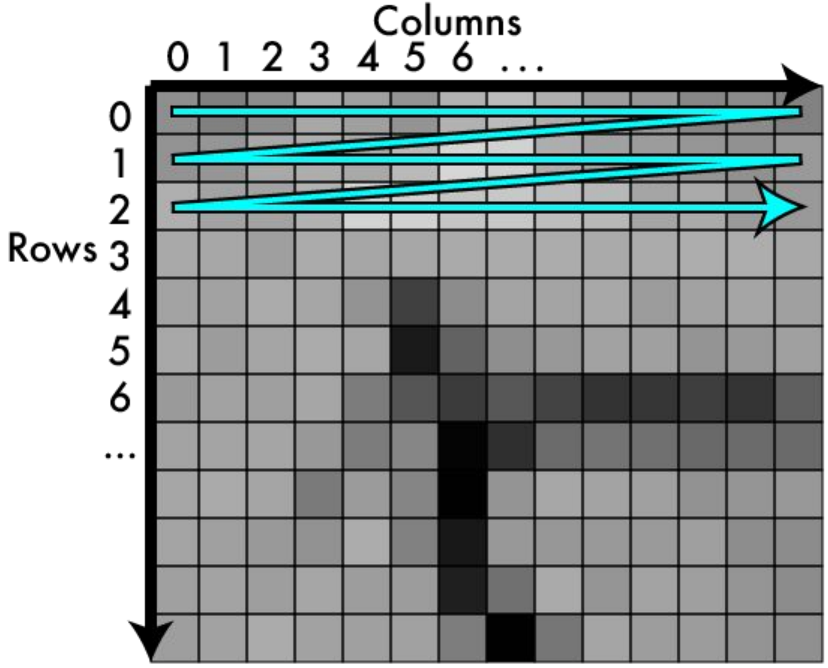
Storage: row major vs column major



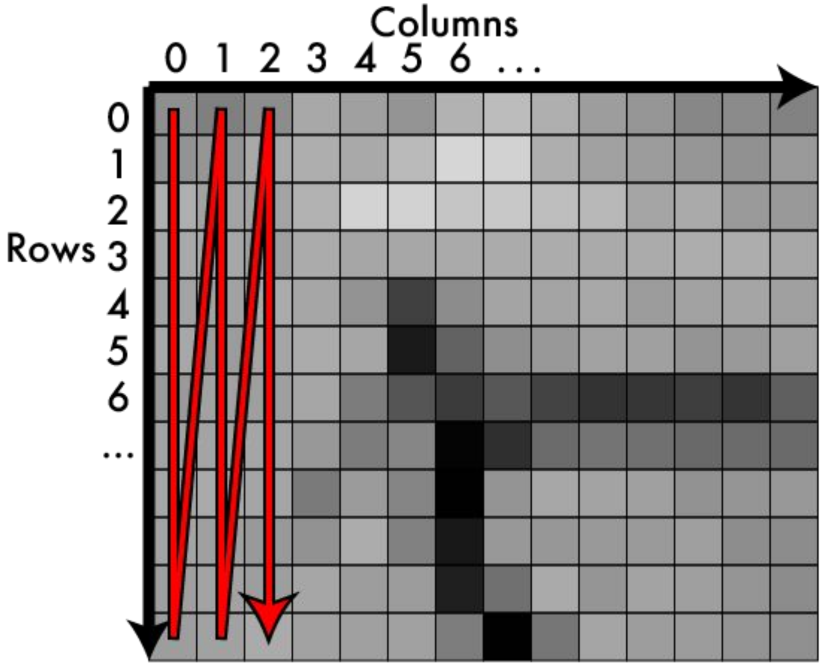


Storage: row major vs column major

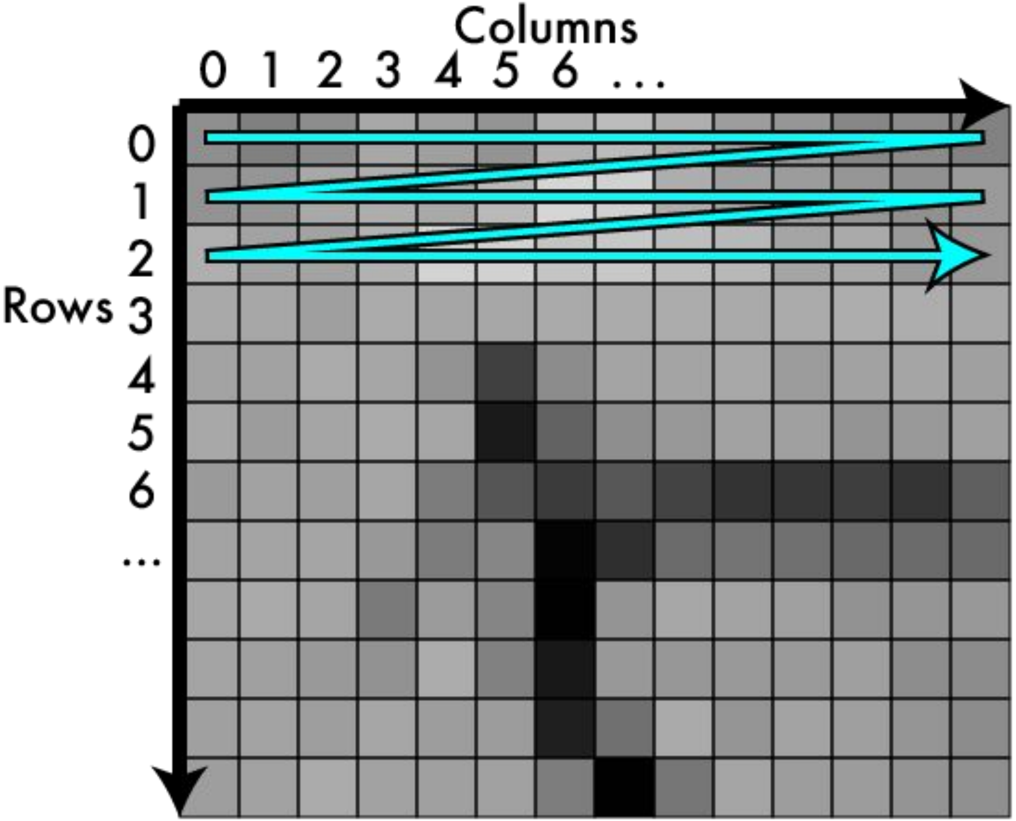
HW



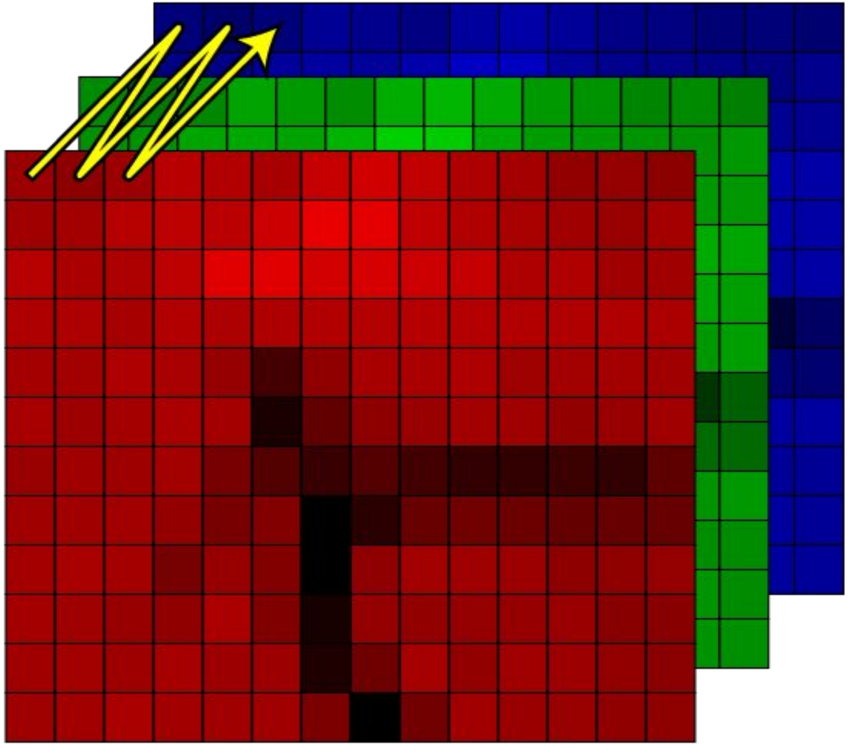
WH

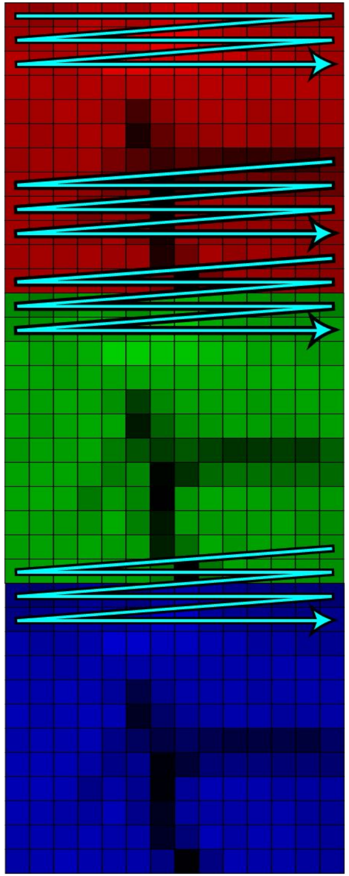


Typically use row-major or HW

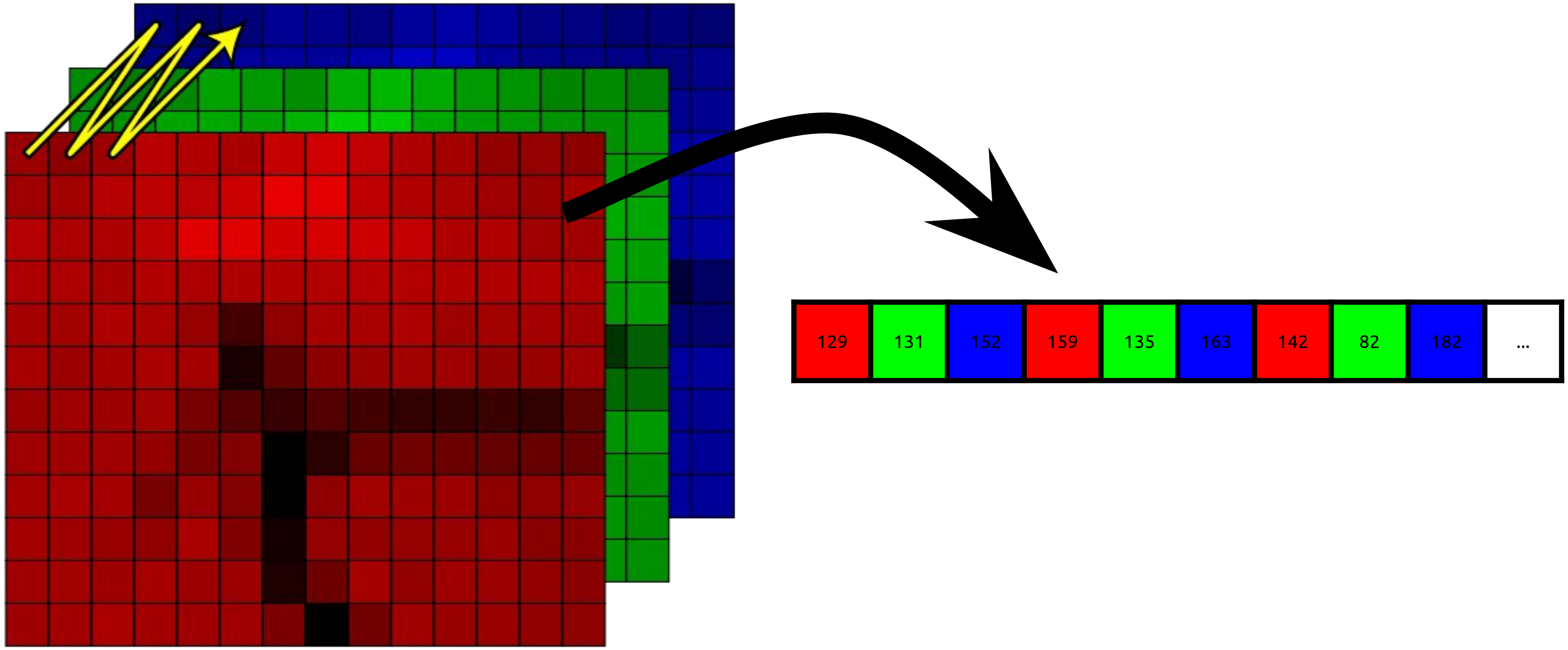


In 3d we have more choices!

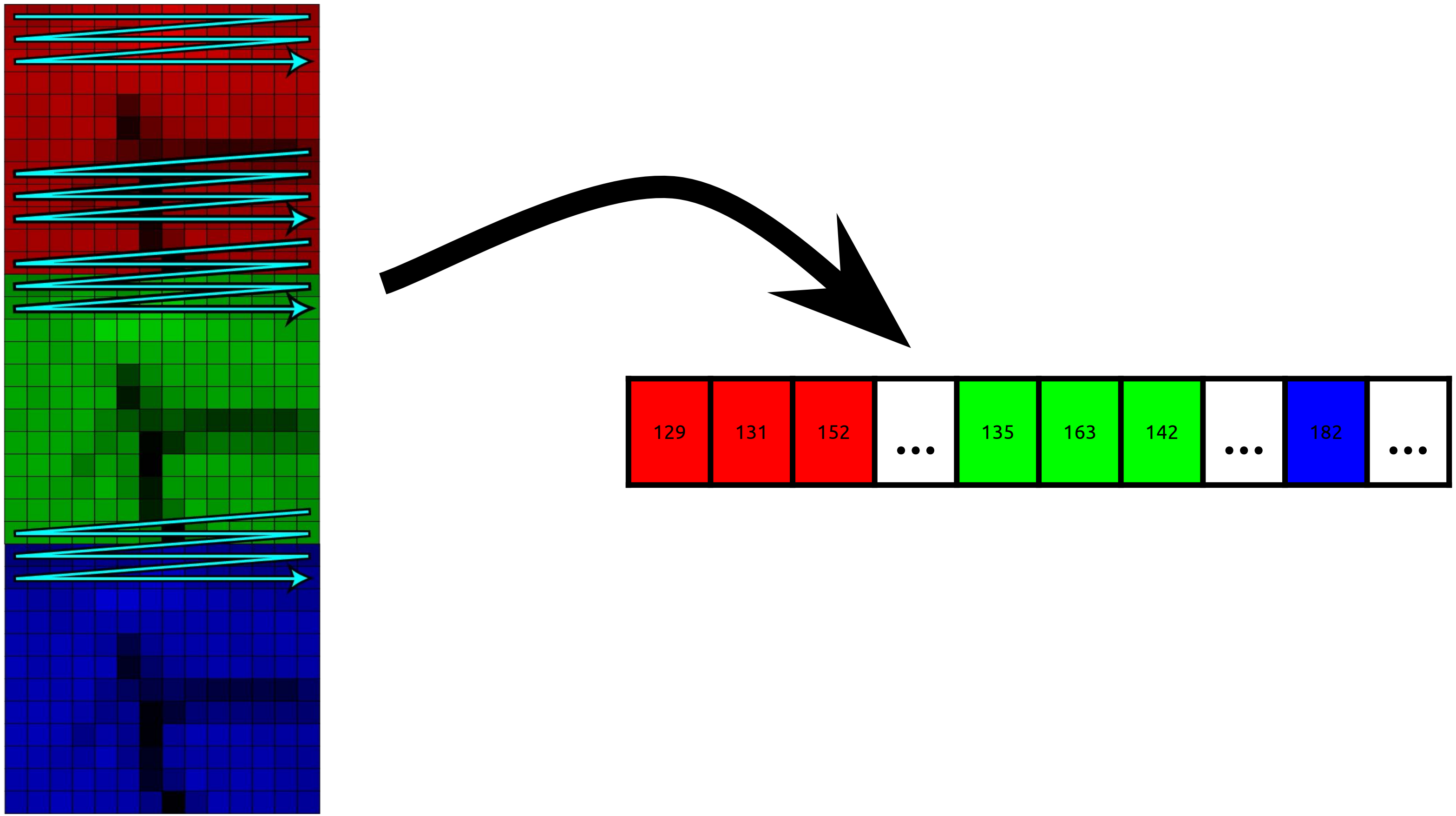




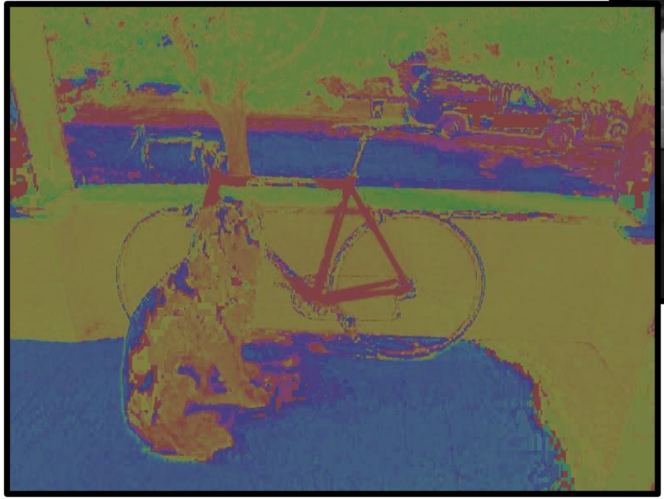
HWC: channels interleaved



CHW: channels separated



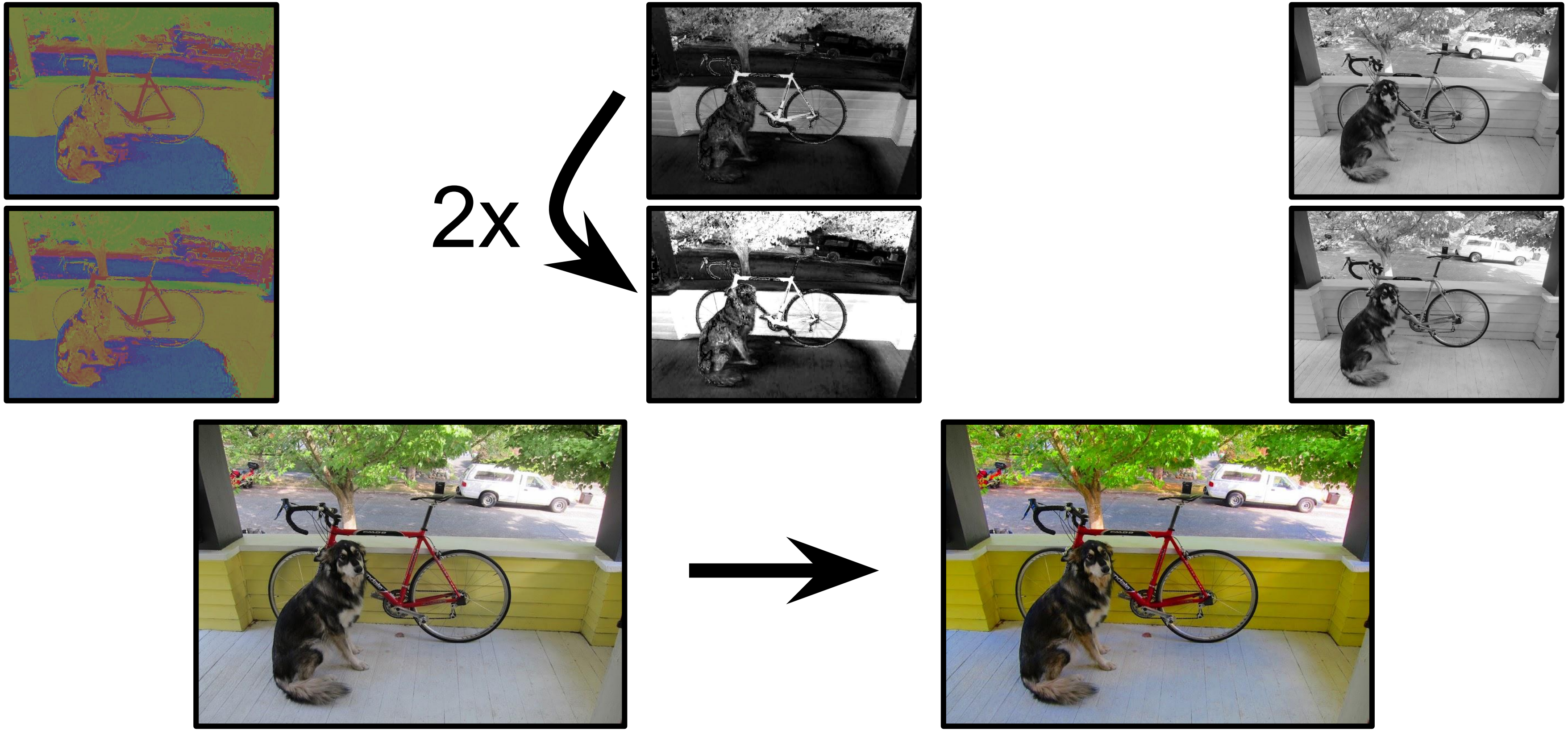
Hue



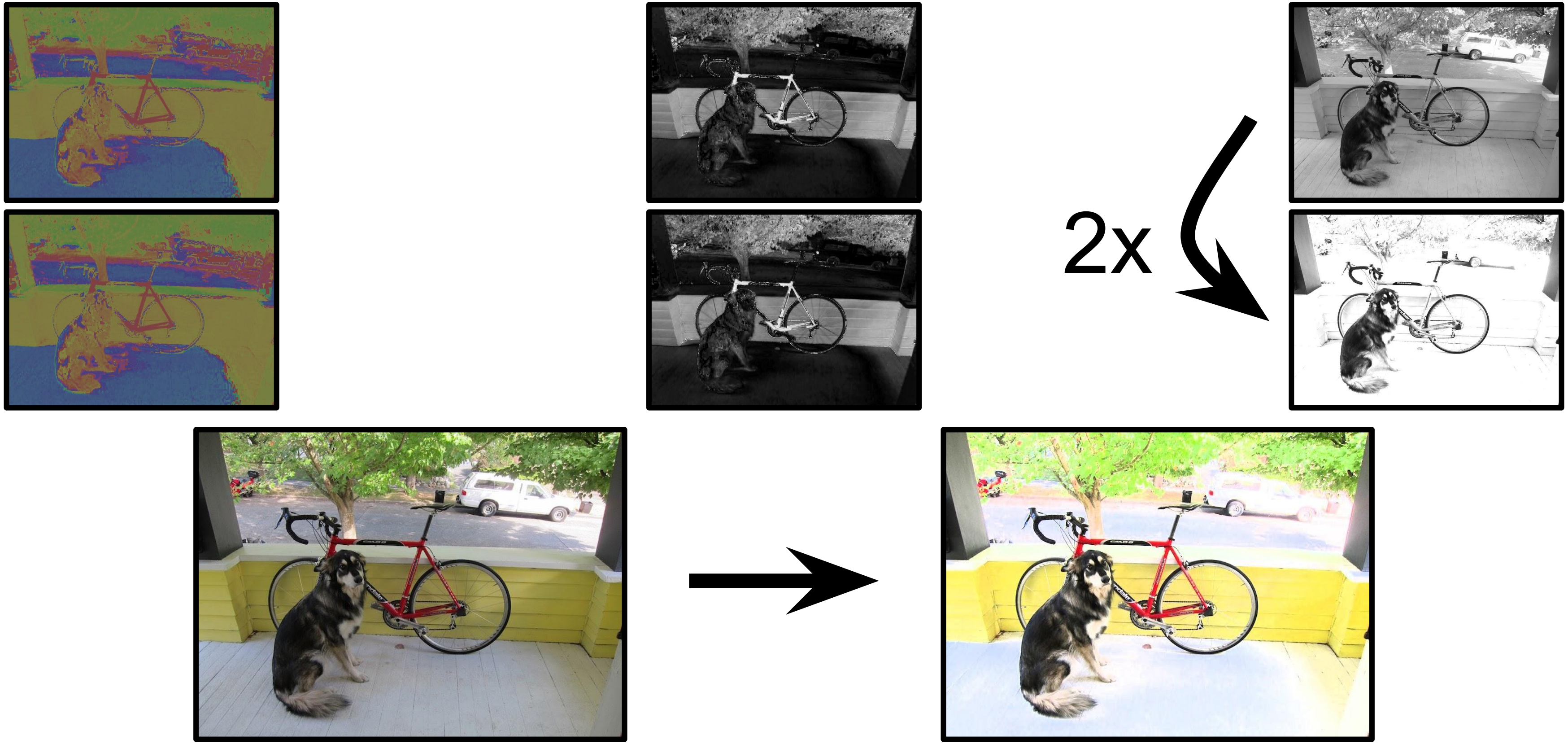
Saturation

Value

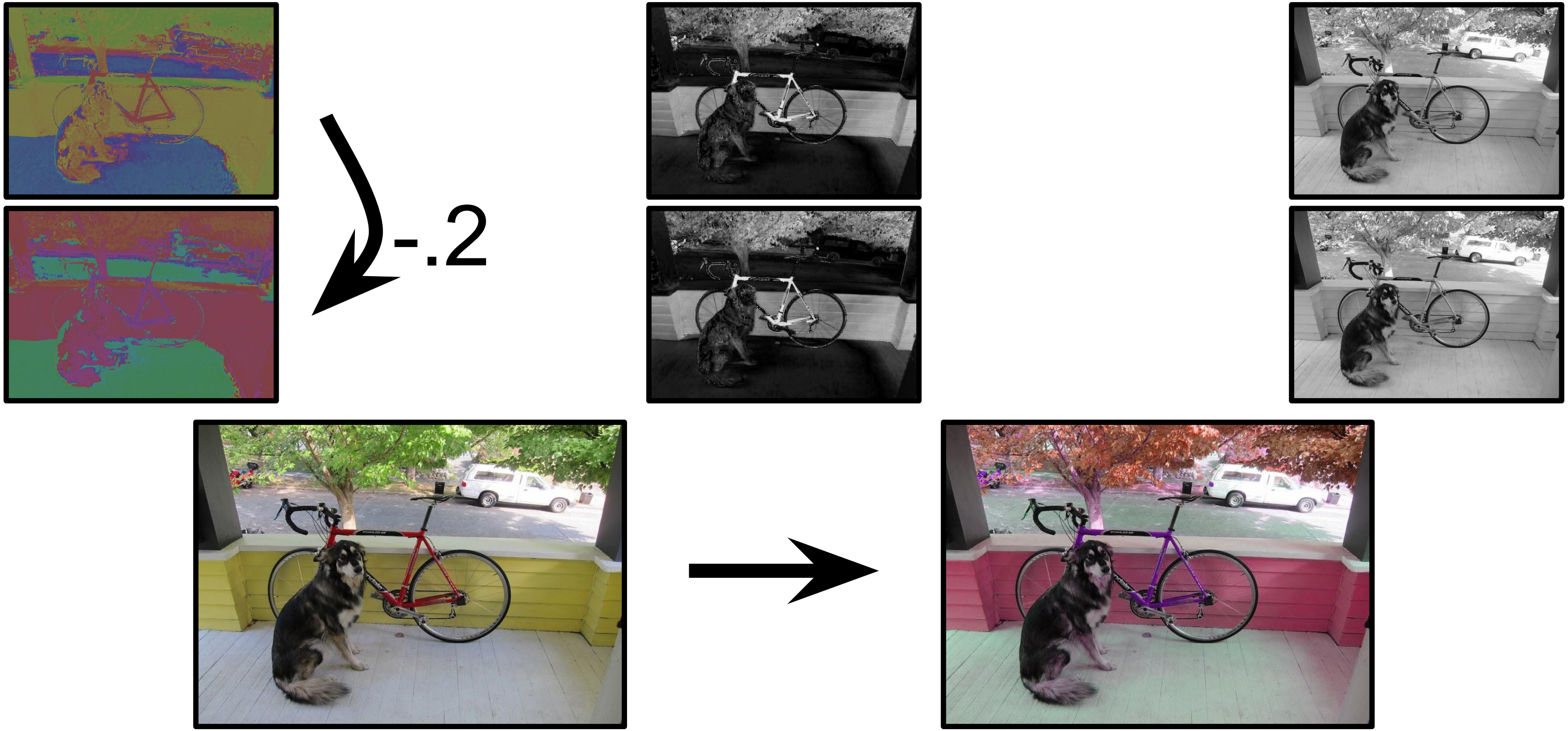
More saturation = intense colors



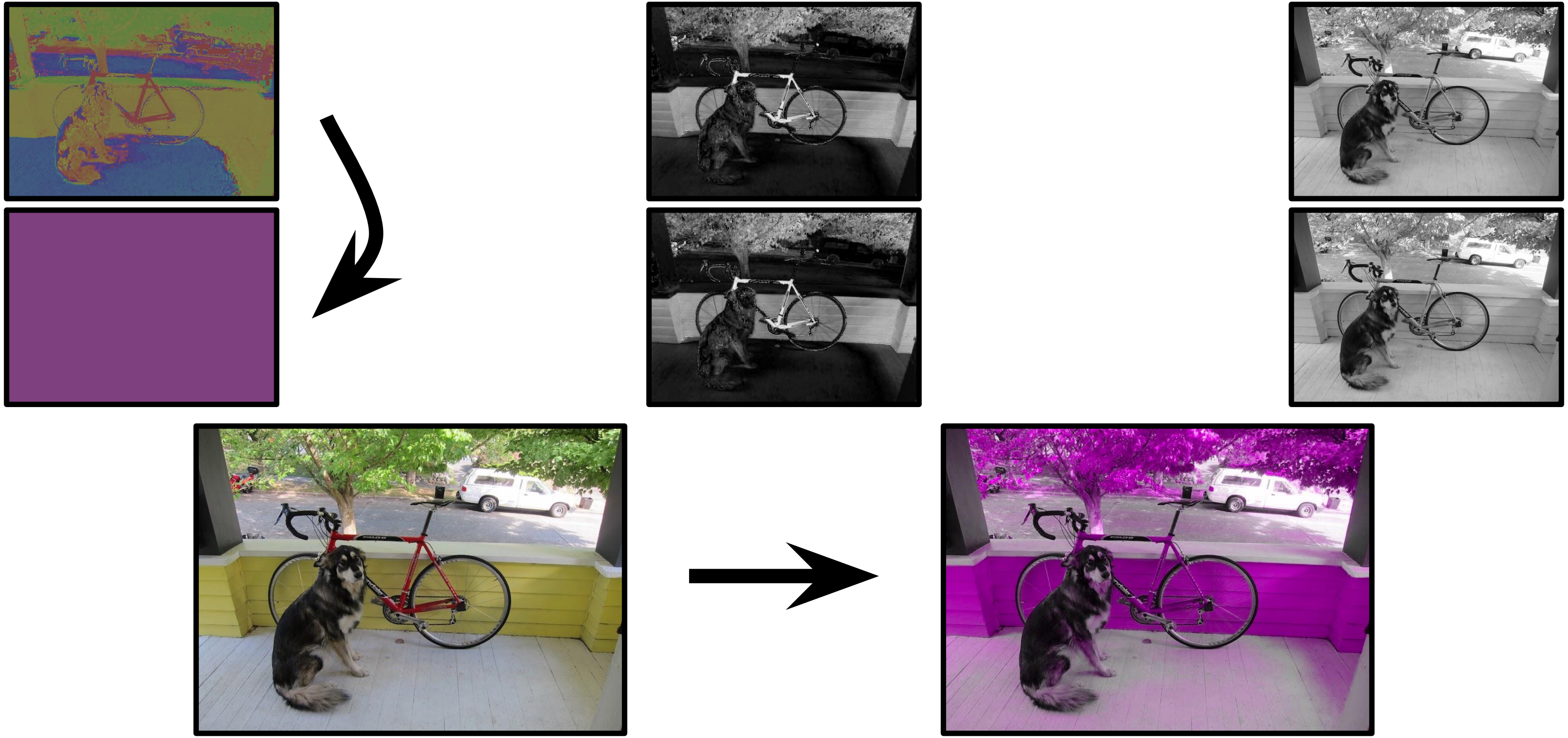
More value = lighter image



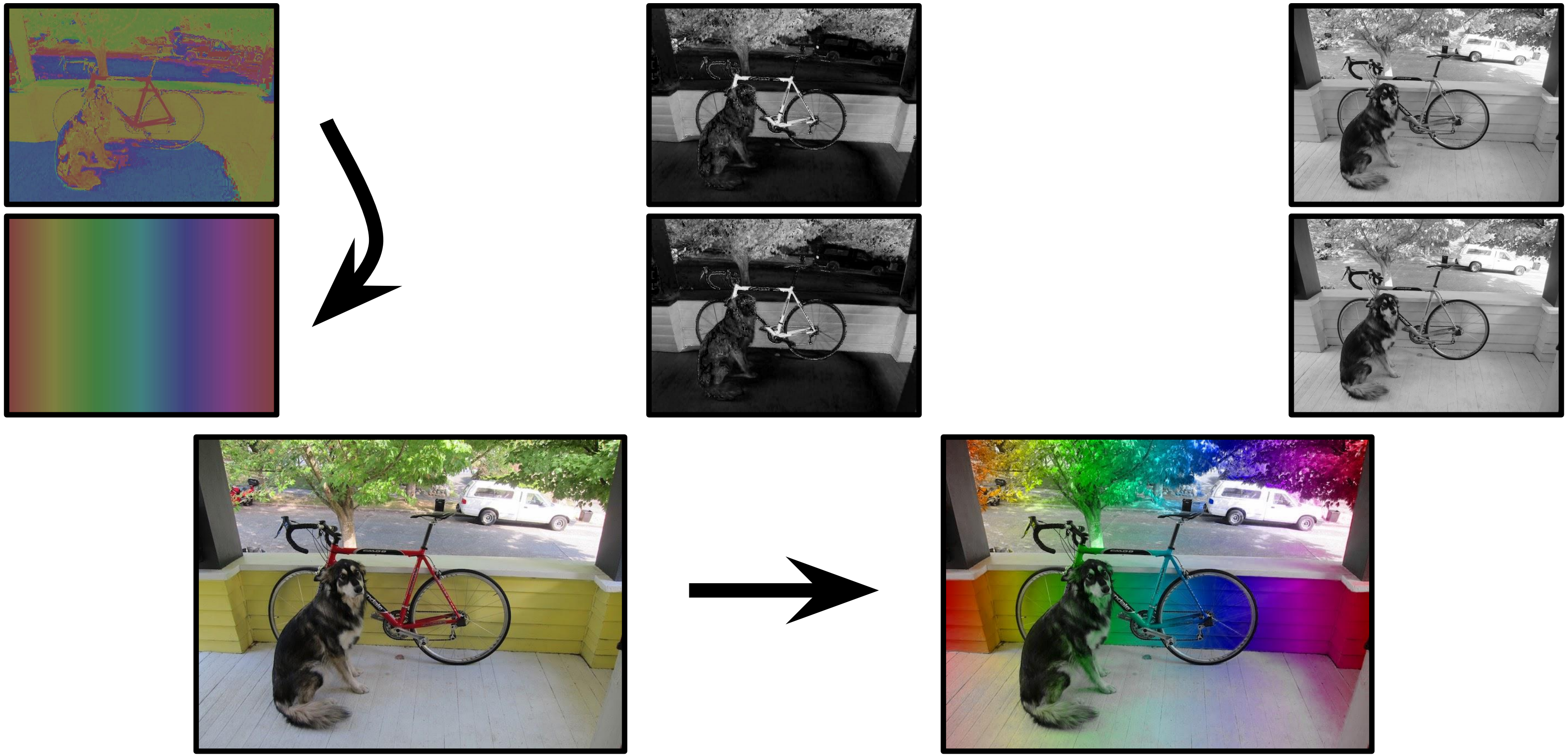
Shift hue = shift colors



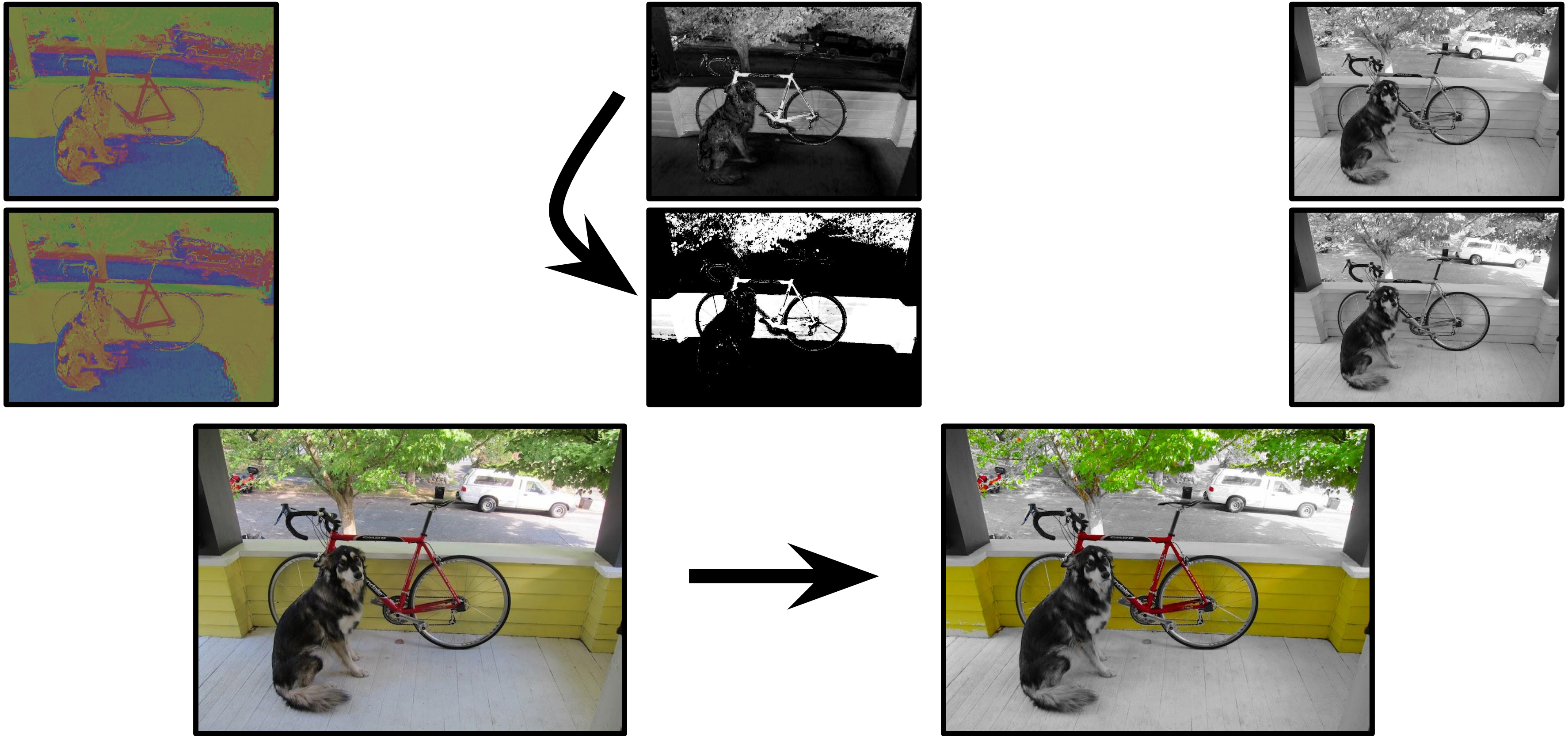
Set hue to your favorite color!



Or pattern...



Increase and threshold saturation



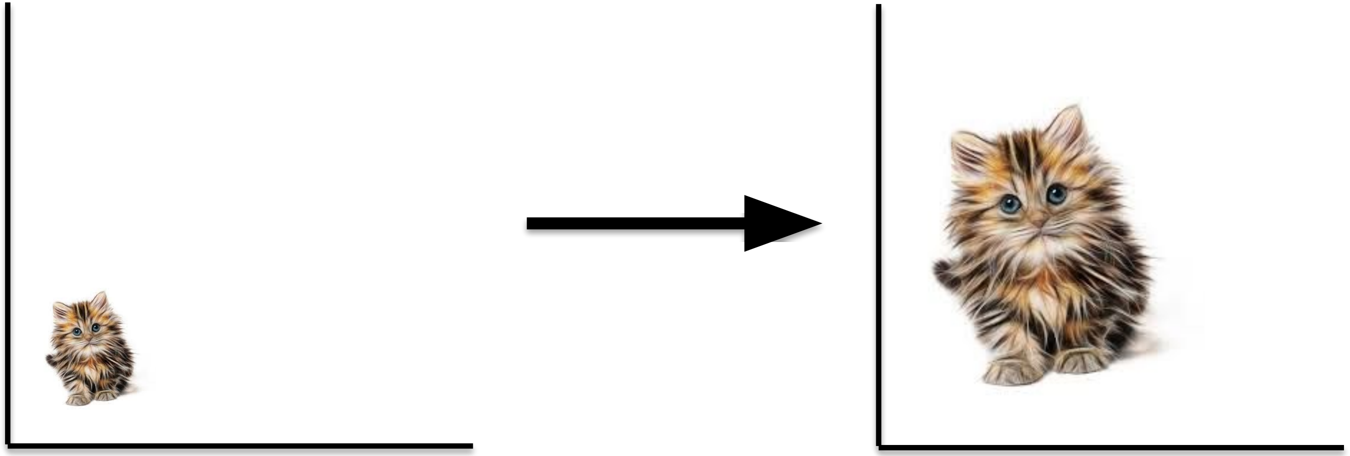
Transformation

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

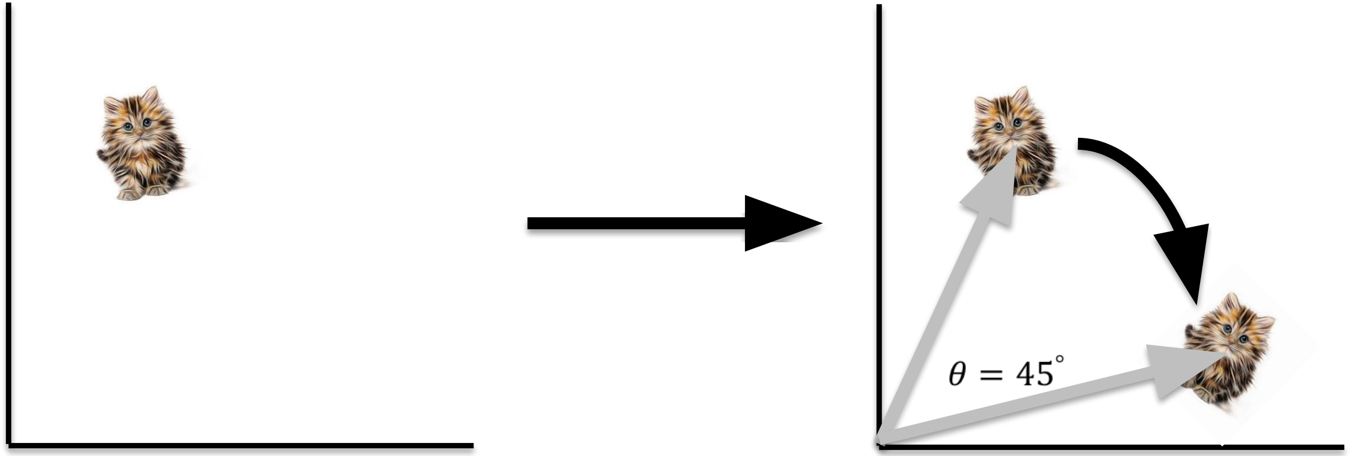
$$\begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \times \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} s_x x \\ s_y y \end{bmatrix}$$

Transformation

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

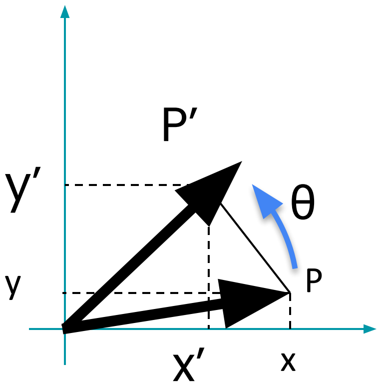


Rotation



2D Rotation Matrix Formula

Counter-clockwise rotation by an angle θ



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

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

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

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

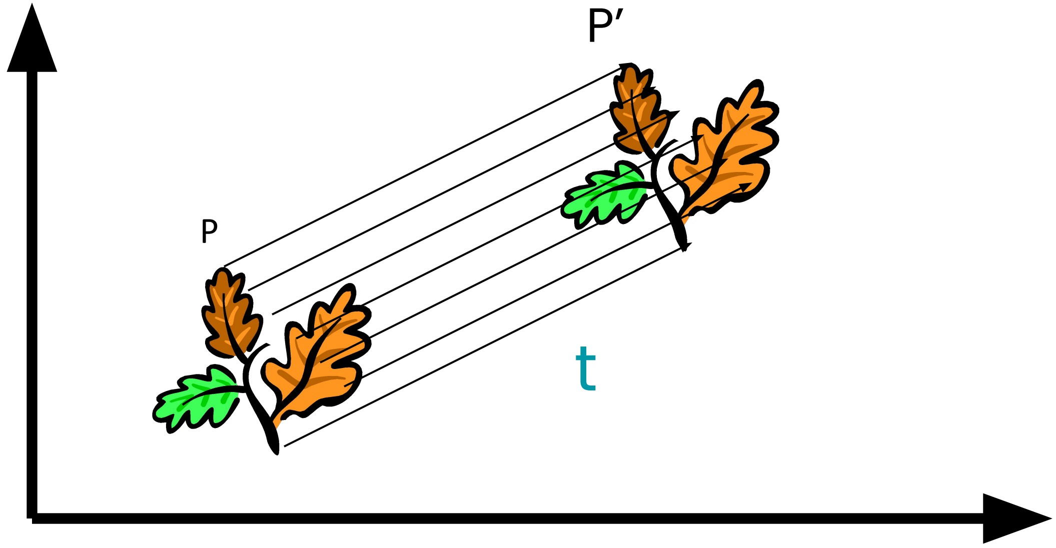
Homogeneous system

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

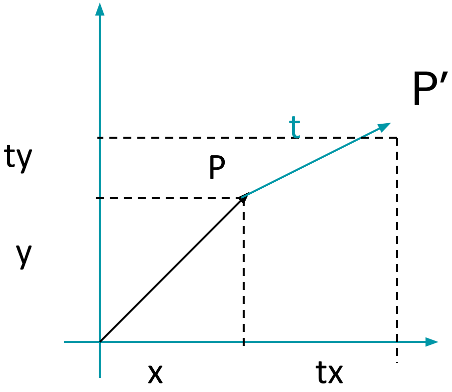
$$\begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} ax + by + c \\ dx + ey + f \\ 1 \end{bmatrix}$$

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

2D Translation



2D Translation using Homogeneous Coordinates



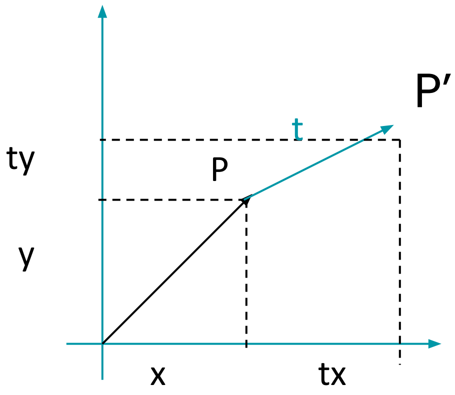
$$\mathbf{P} = (x, y) \rightarrow (x, y, 1)$$

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

P

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

2D Translation using Homogeneous Coordinates

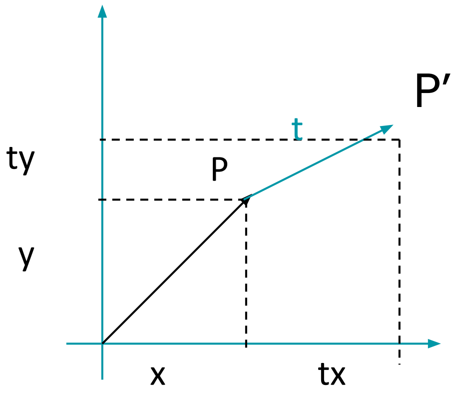


$$\mathbf{P} = (x, y) \rightarrow (x, y, 1)$$

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

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

2D Translation using Homogeneous Coordinates

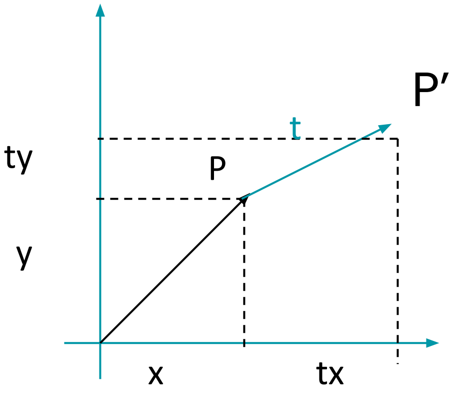


$$\mathbf{P} = (x, y) \rightarrow (x, y, 1)$$

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

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

2D Translation using Homogeneous Coordinates

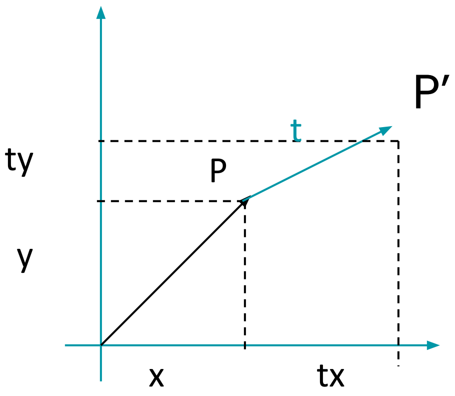


$$\mathbf{P} = (x, y) \rightarrow (x, y, 1)$$

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

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

2D Translation using Homogeneous Coordinates



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

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

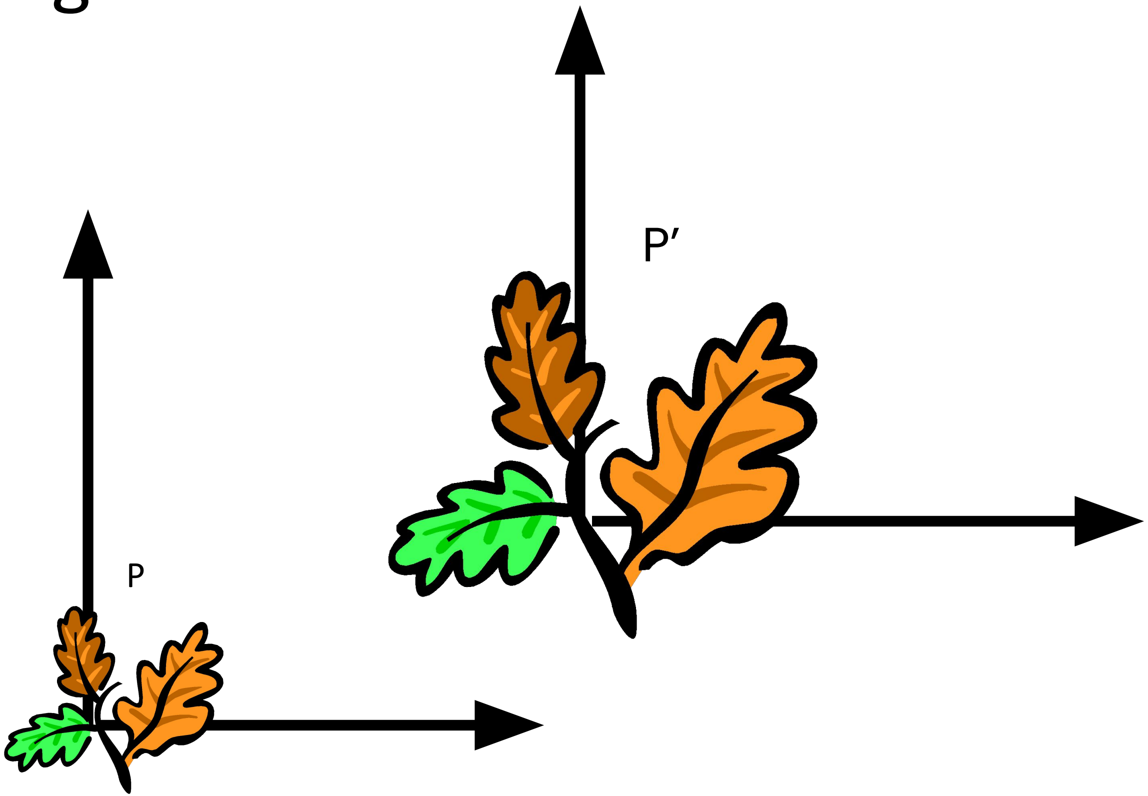
t

P

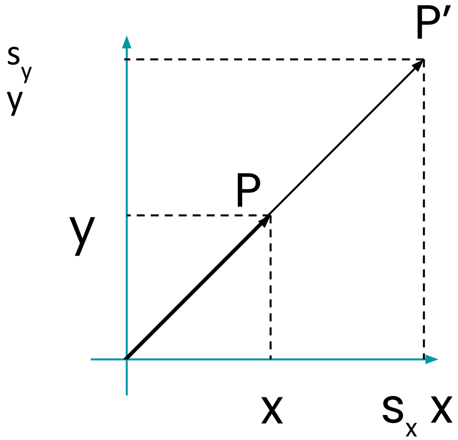
$$\mathbf{P}' \rightarrow \begin{bmatrix} x + t_x \\ y + t_y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

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

Scaling



Scaling Equation

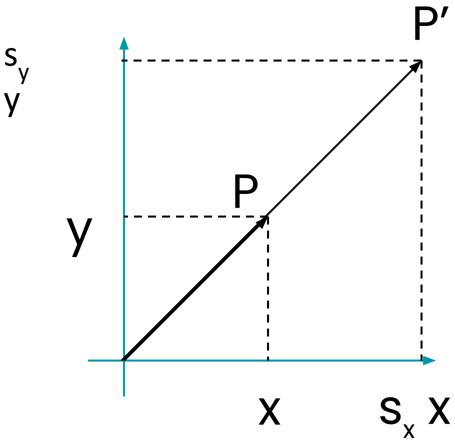


$$\mathbf{P} = (\mathrm{x}, \mathrm{y}) \rightarrow \mathbf{P'} = (\mathrm{s}_\mathrm{x}\mathrm{x}, \mathrm{s}_\mathrm{y}\mathrm{y})$$

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

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

Scaling Equation



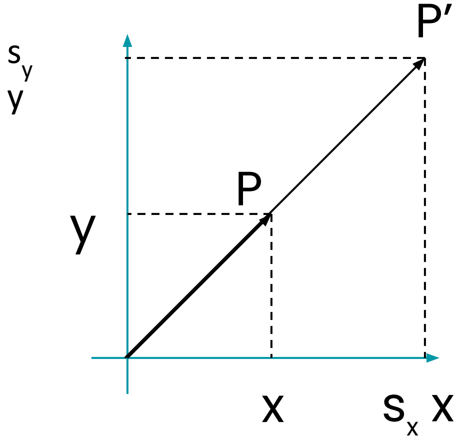
$$\mathbf{P} = (x, y) \rightarrow \mathbf{P}' = (s_x x, s_y y)$$

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

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

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

Scaling Equation



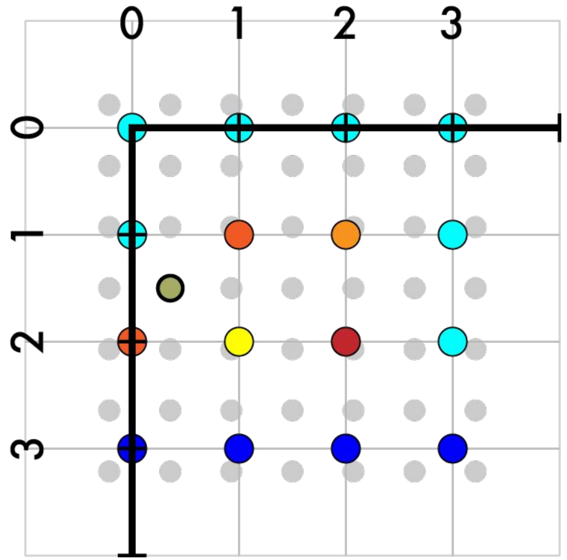
$$\mathbf{P} = (\mathrm{x}, \mathrm{y}) \rightarrow \mathbf{P}' = (\mathrm{s_x x}, \mathrm{s_y y})$$

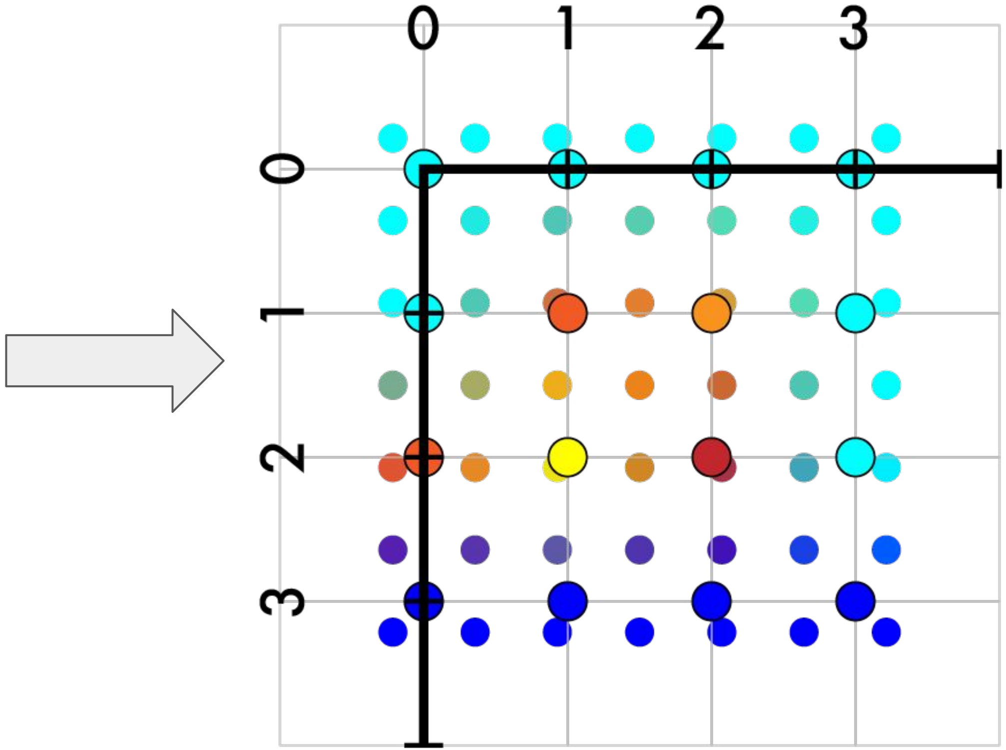
$$\mathbf{P} = (x, y) \rightarrow (x, y, 1)$$

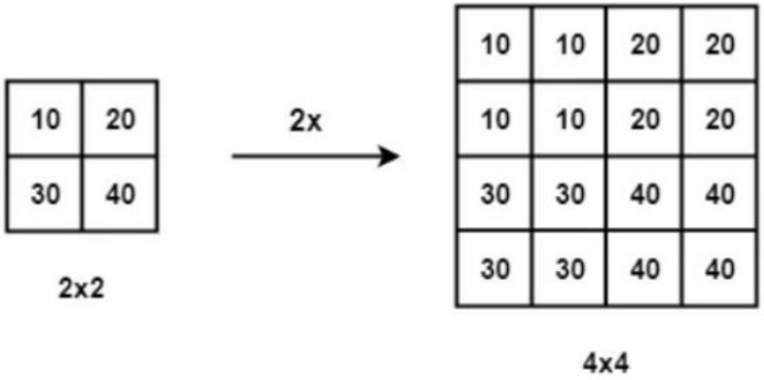
$$\mathbf{P}' = (s_x x, s_y y) \rightarrow (s_x x, s_y y, 1)$$

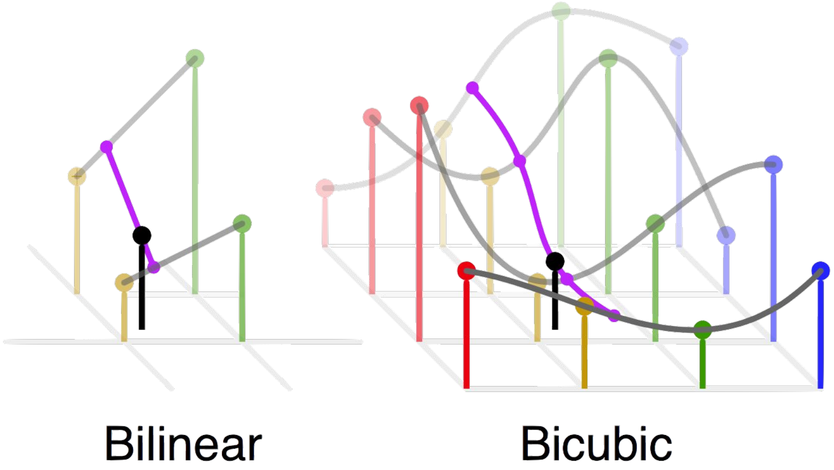
$$\mathbf{P}' \rightarrow \begin{bmatrix} s_x x \\ s_y y \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{S}} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{S}' & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{bmatrix}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

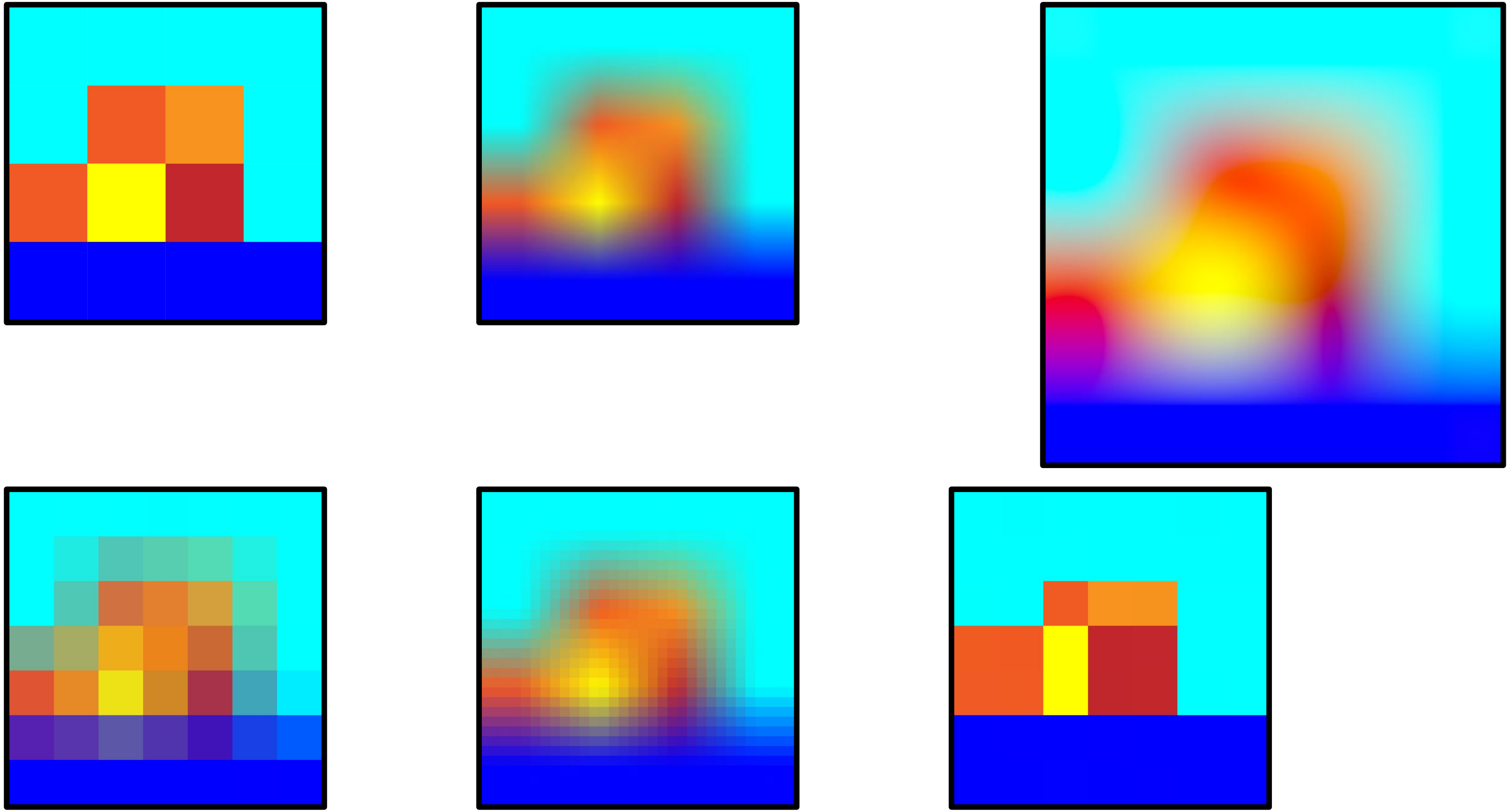
4x4 -> 7x7



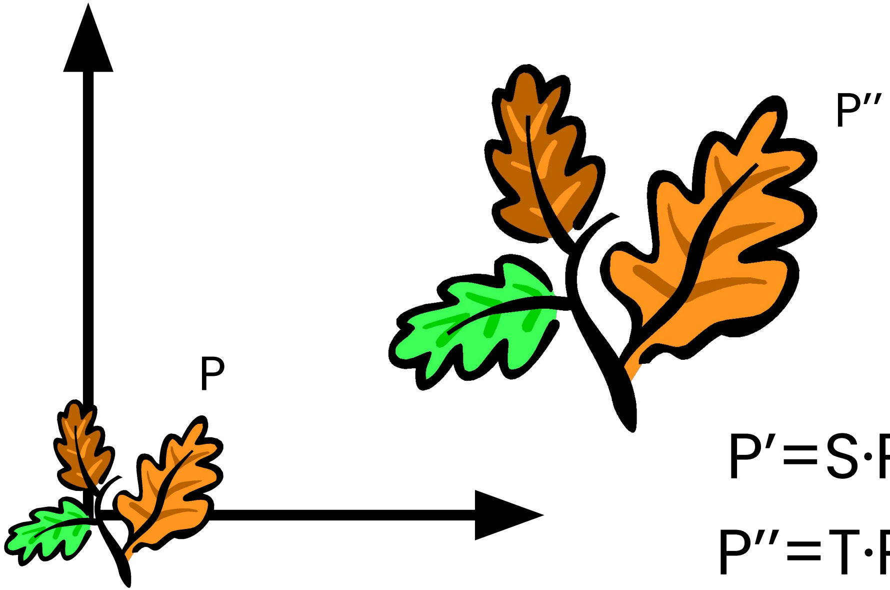



10
20
30
40
2x2
2x
10
10
20
20
10
10
20
20
30
30
40
40
30
30
40
40
4x4


Bilinear
Bicubic



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

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

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 Vs Scaling & Translating

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

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

Q & A

External Readings and Lecture Notes :

- Rate of Change

https://thuraaung-1601.medium.com/rate-of-change-89213d5ce034

- Introduction to Differentiation (Optional)

https://thuraaung-1601.medium.com/introduction-to-differentiation-a4739042e523

- Digital Image Basics : Resampling Images

https://thuraaung-ai.medium.com/resampling-images-55025fb34ee7

- Digital Image Basics : Geometric Transformations

https://thuraaung-ai.medium.com/digital-image-basics-2af4ee3448e5

Thank You !