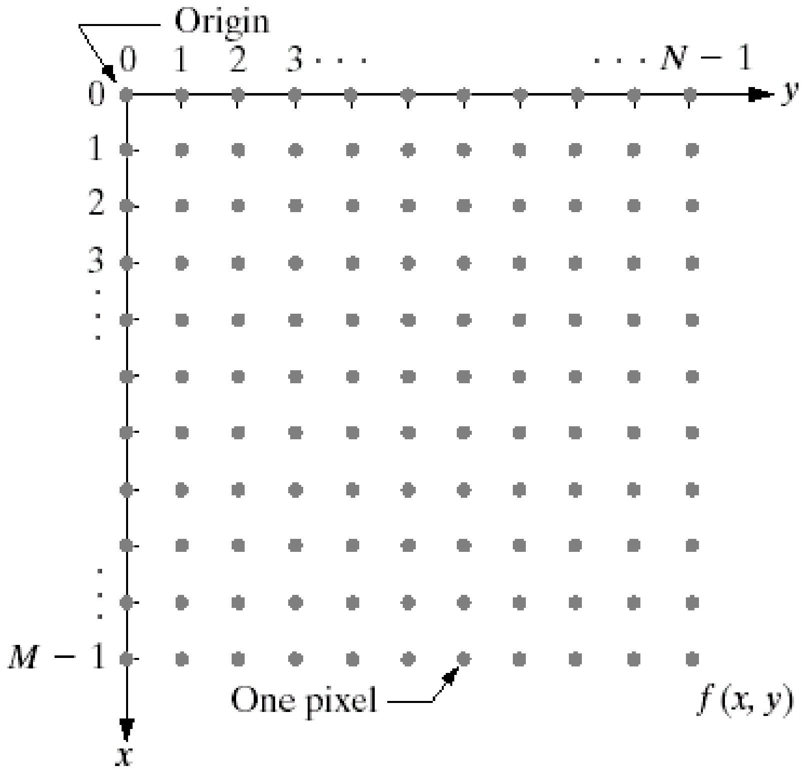
Basics of Digital Imaging

- Basic image representation.
- Getting digital images: Sampling and quantization
- Geometrical transformation and interpolation
- Color perception
- Color image representation: Color spaces

Digital Image Representation

The most standard representation of a digital image, as a grid of **pixels** (picture elements):



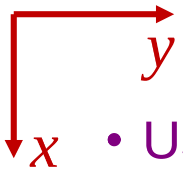
$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0,N-1) \\ f(1,0) & f(1,1) & \cdots & f(1,N-1) \\ \vdots & \vdots & & \vdots \\ f(M-1,0) & f(M-1,1) & \cdots & f(M-1,N-1) \end{bmatrix}$$

$$\boldsymbol{A} = \begin{bmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,N-1} \\ a_{1,0} & a_{0,0} & \cdots & a_{1,N-1} \\ \vdots & \vdots & & \vdots \\ a_{M-1,0} & a_{M-1,1} & \cdots & a_{M-1,N-1} \end{bmatrix}$$

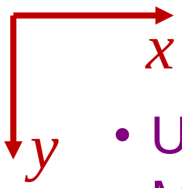
Digital Image Representation

A few things to note about the image coordinate systems:

- Where the origin is (top-left corner is the convention).
- Axis designations (be careful when doing implementation):



- Used in textbook
- Keep the right-hand convention
- May cause confusion



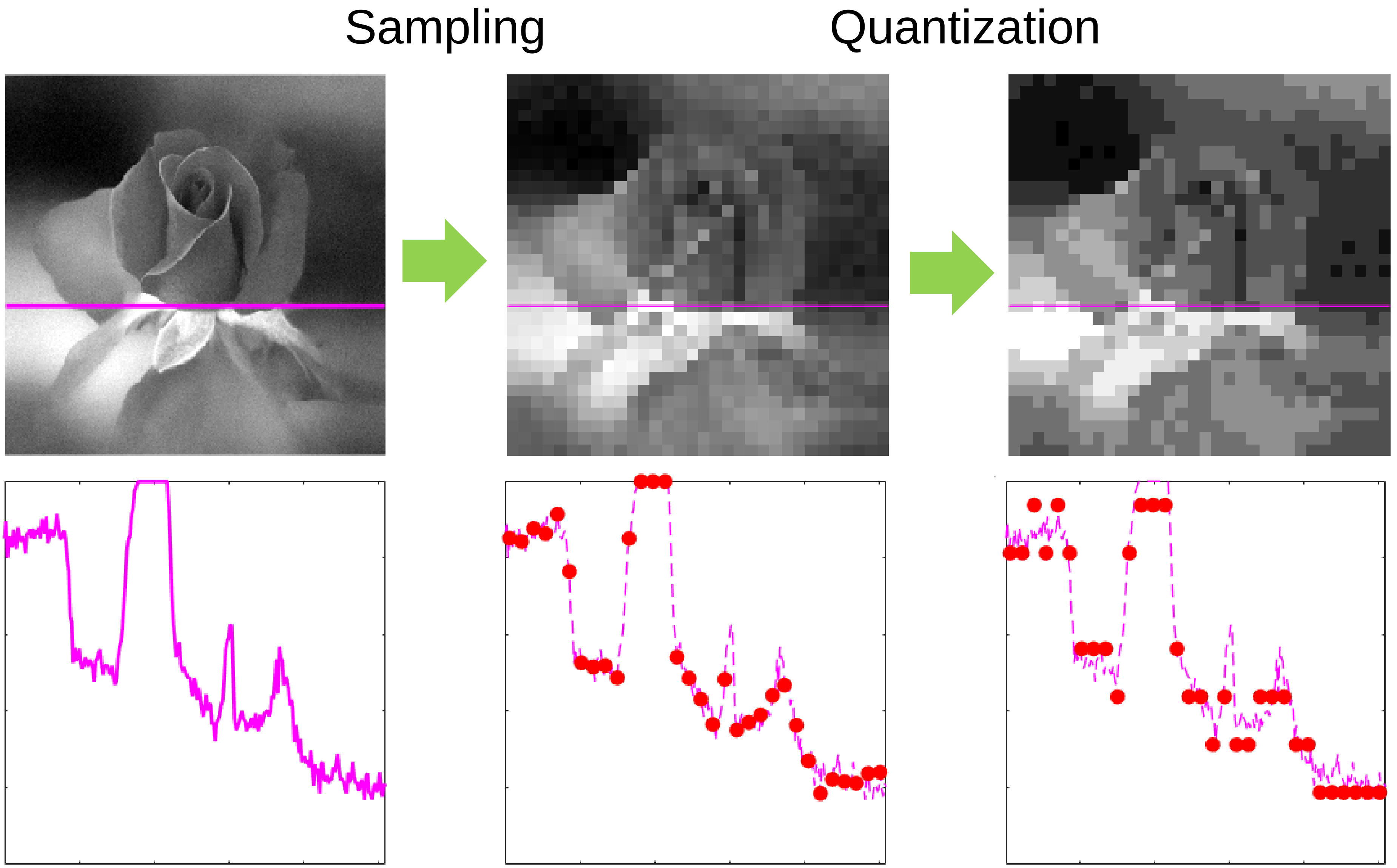
- Used in practice (e.g., Matlab, OpenCV)
- Not right-hand convention

- Representation of multi-valued pixels: The **channel** dimension; $f(x, y, c)$
  - Color images
  - Hyperspectral images
- Third spatial dimension; **voxels** (volume elements); $f(x, y, z)$
- Temporal dimension; videos; $f(x, y, t)$

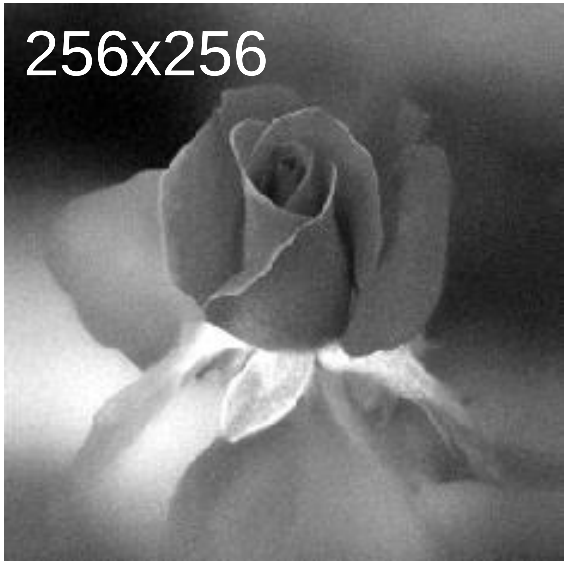
Sampling and Quantization

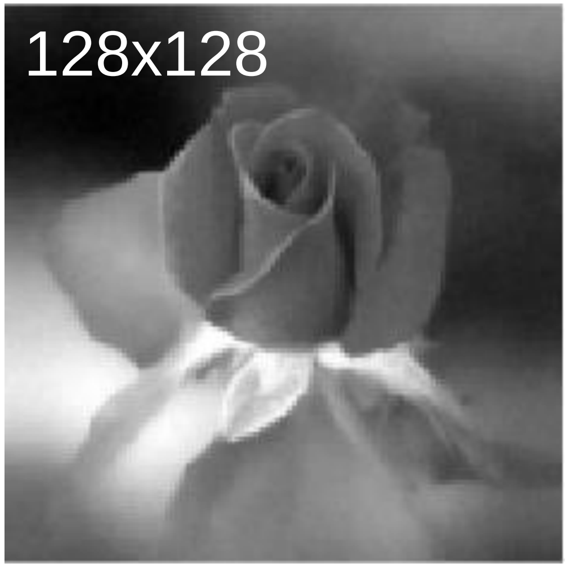
- The process of going from the ***continuous*** real world to the ***discrete*** digital world.
- **Sampling**: Obtaining values at discrete points in the spatial / temporal domains.
- **Quantization**: The representation of the sampled values with a set of discrete values (**quantization levels**).
  - Typically there are 2^k levels (k = bits per pixel).
  - The quantization levels are commonly evenly spaced, but this is not required.

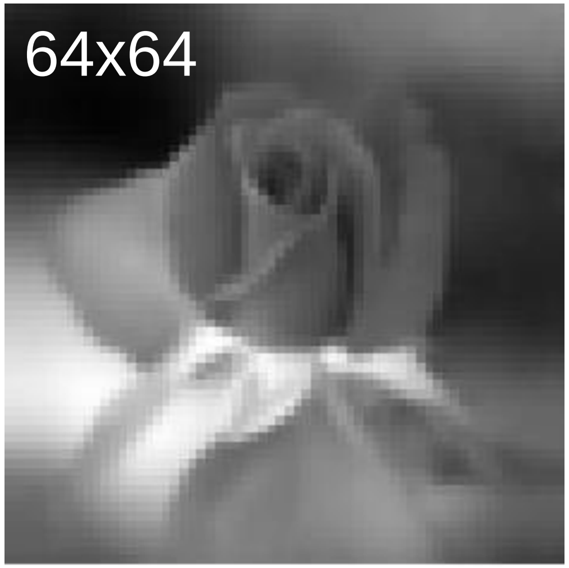
Sampling and Quantization

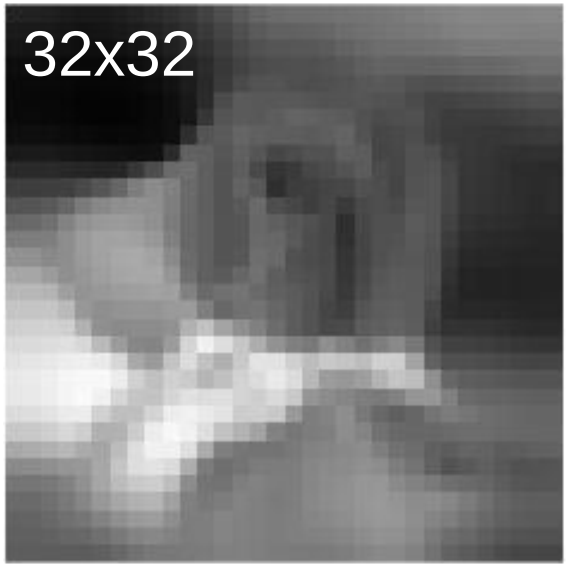


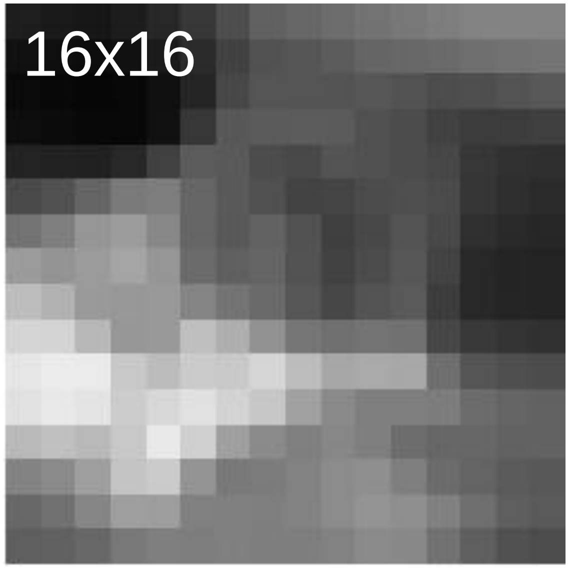
Sampling and Spatial Resolution



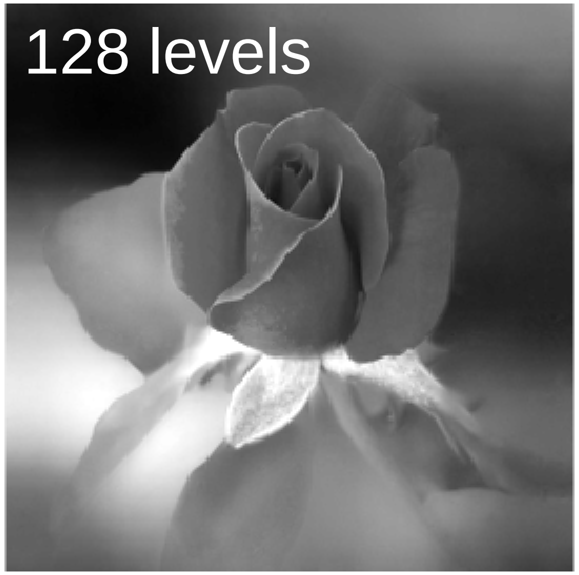


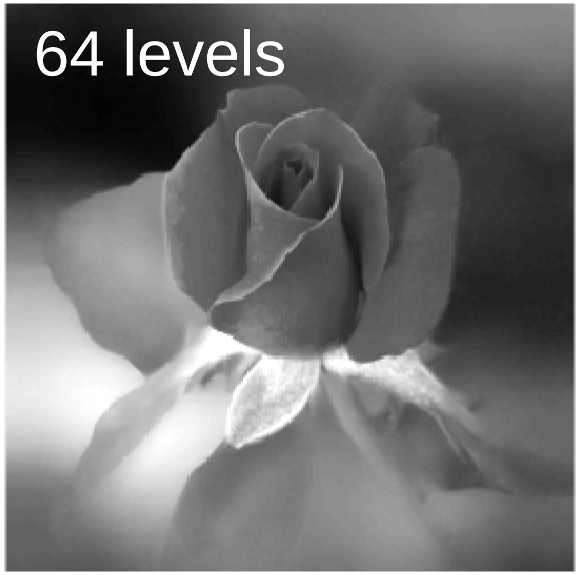


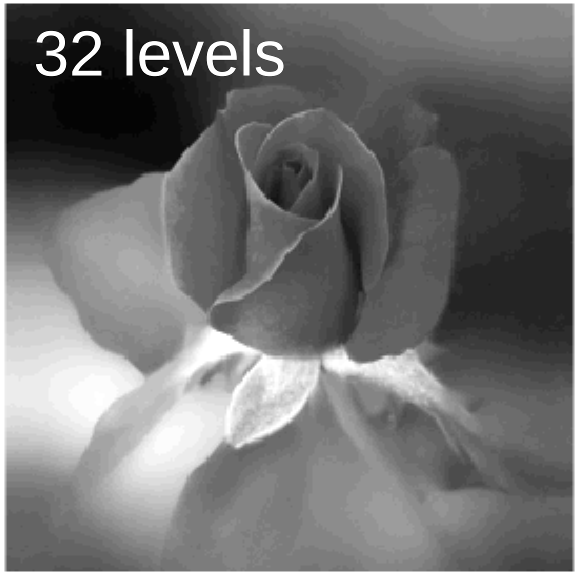


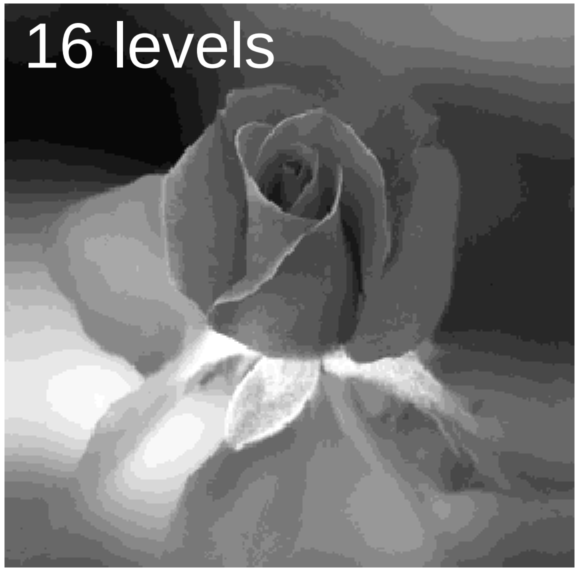


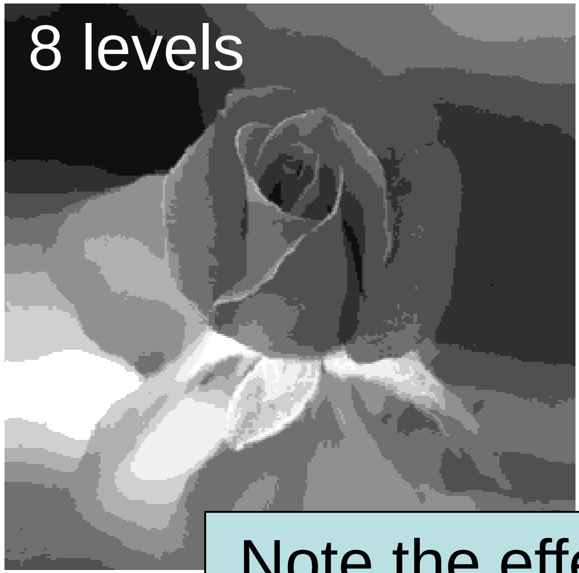
Quantization and Intensity Resolution

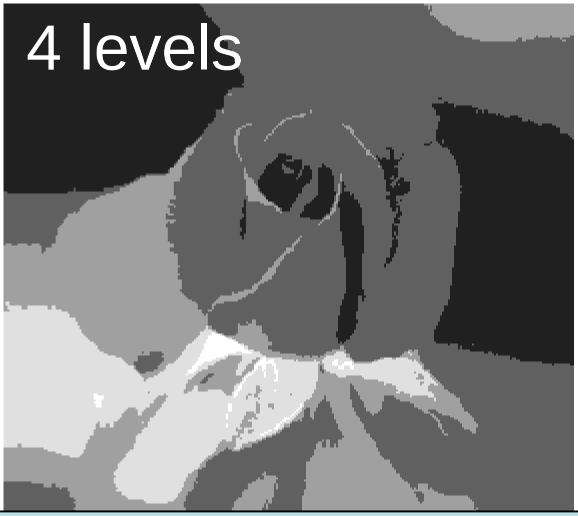






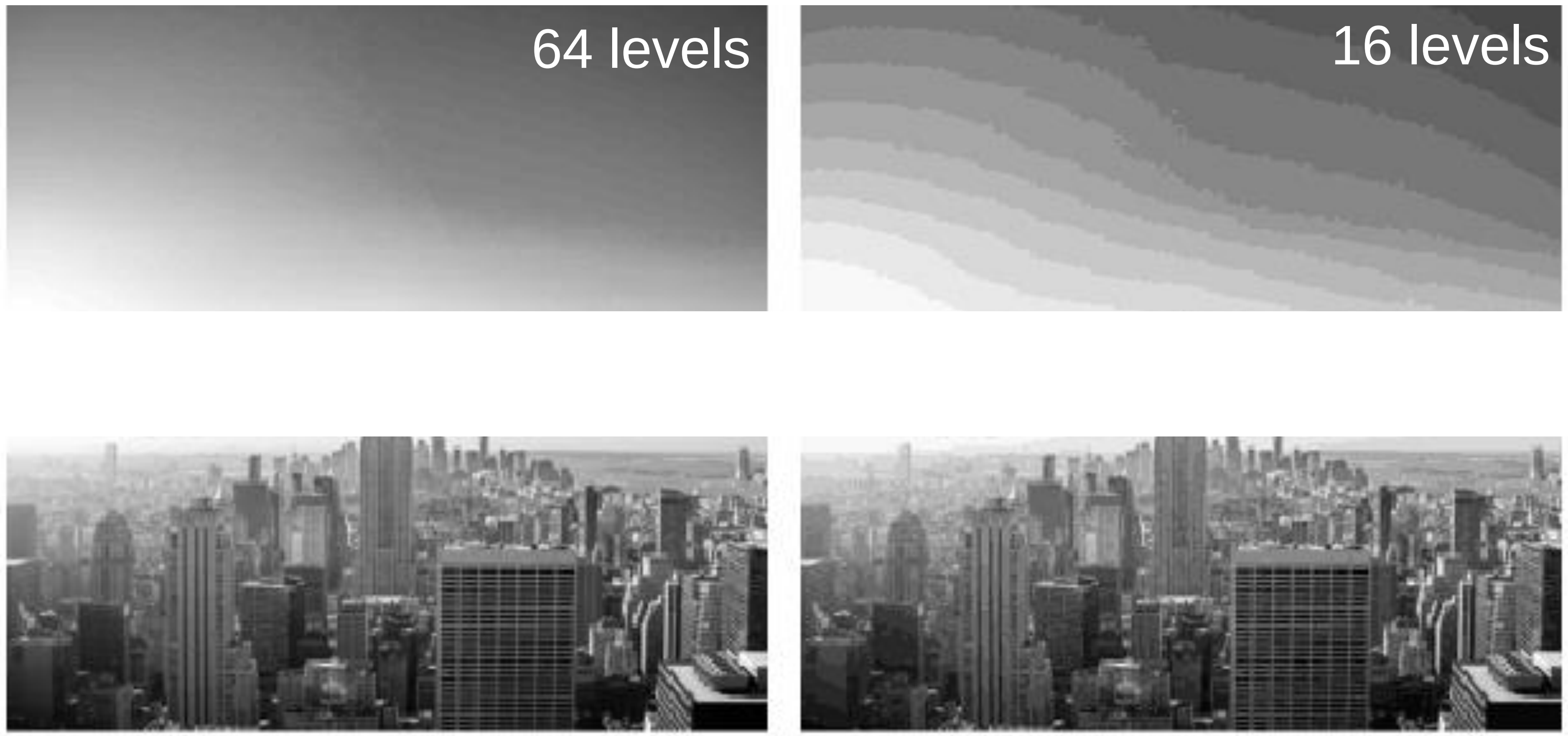






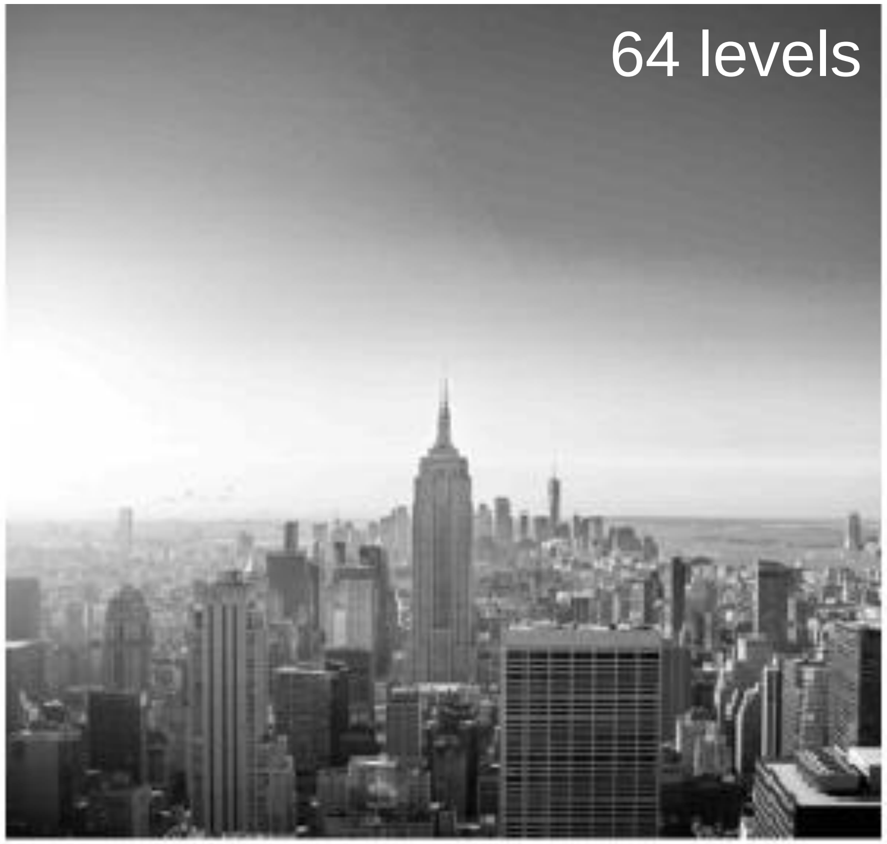
Note the effect of **false contours**

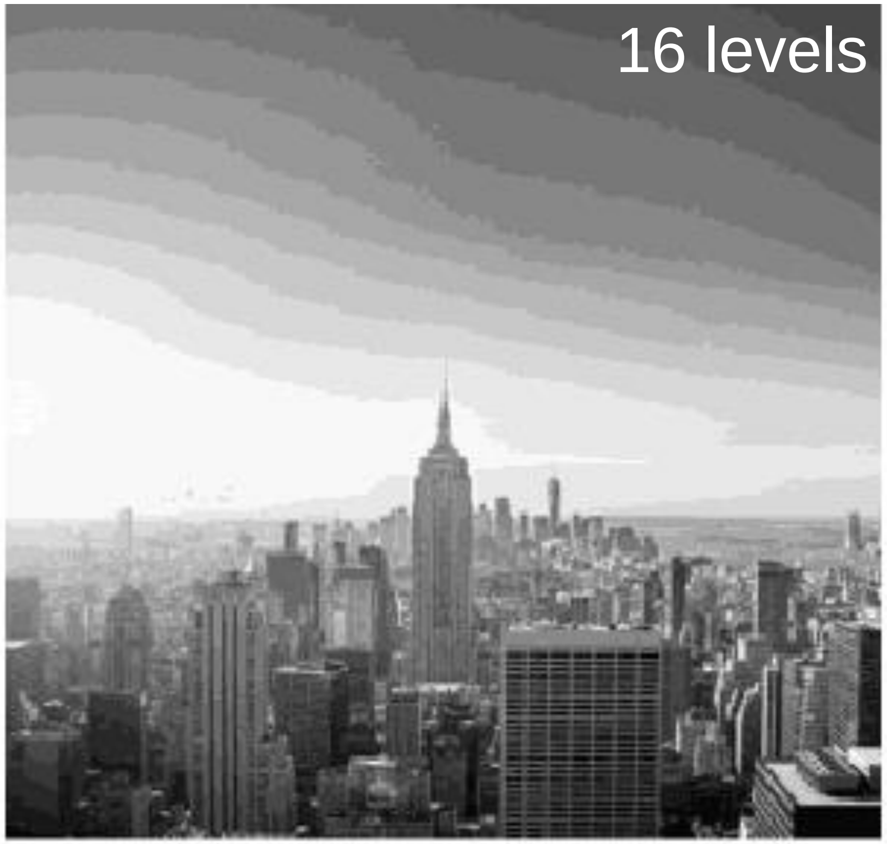
Quantization and Visual Quality



Compare these two images and assess the visual qualities.
How do the two regions differ?

Quantization and Visual Quality



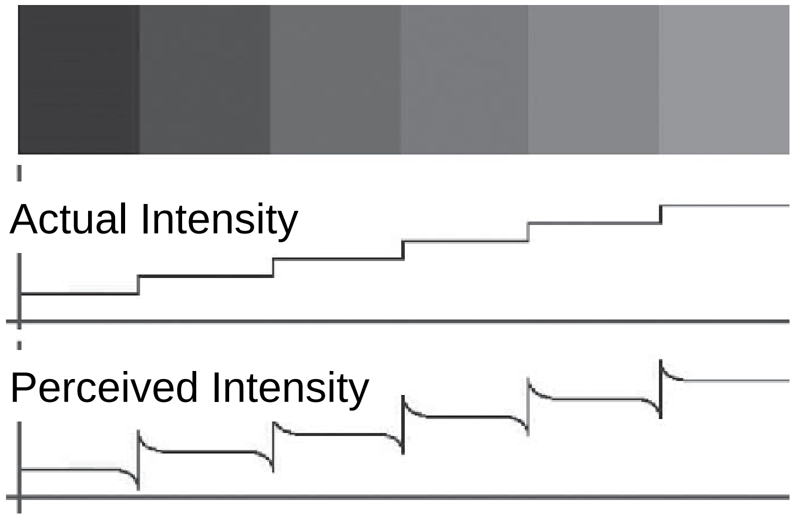


The reduction of “perceived quality” due to quantization depends on the level of image complexity.

Mach Band and Visual Quality

For a little more understanding:

- Perceived edges are important for understanding what we see; they allow us to tell different regions/objects.
- As a result, our visual system automatically enhances edges in what we see. This effect is called **Mach Band**.

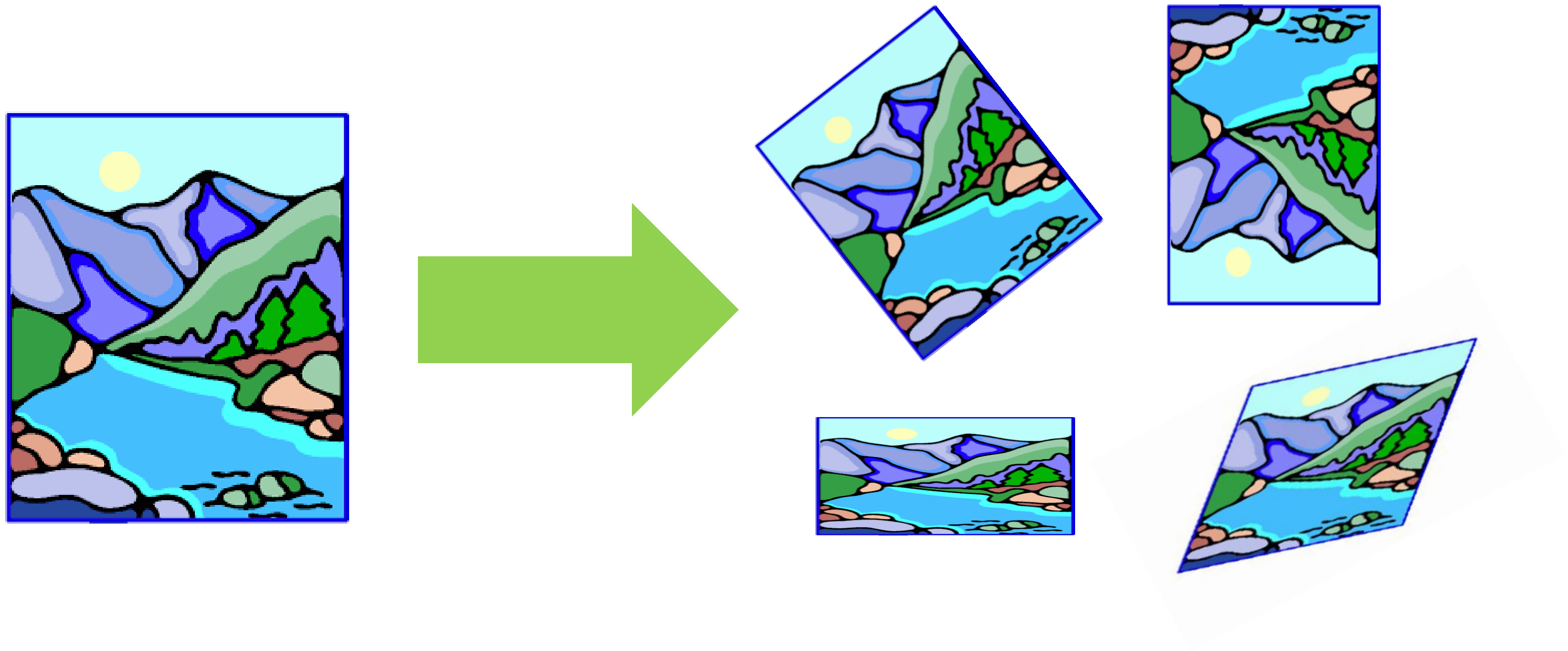


- Therefore, edges that are not real (such as false contours) are undesirable. (Not a problem in complex scenes: Why?)

Source of image: https://www.semanticscholar.org/paper/Perceptual-Adaptation-of-Image-Based-on-Bands-Khosravy-Gupta/c747f171003f29c972e5d7d7ed469983f7a76382

Geometric Transformations

- Generally speaking, geometric transformations (warping) change the spatial arrangement of pixels.
  - Can be linear or nonlinear.
  - Some examples of linear geometric transformations:



Linear Geometric Transformations

- **Affine** transformations (straight lines and parallelism are preserved):

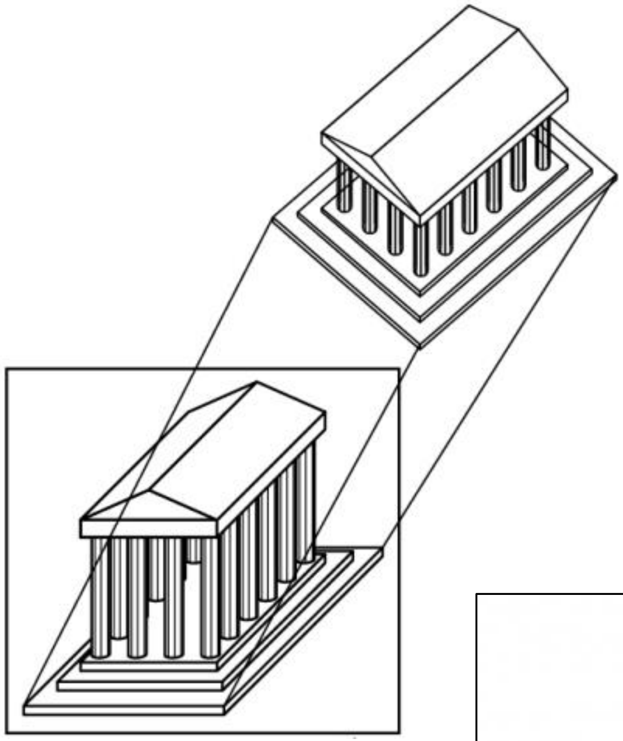
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} d_1 \\ d_2 \end{bmatrix}$$

- A more easy-to-use form:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \boldsymbol{A} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- **Scaling (zooming)**, **translation**, and **rotation** are all special cases of affine transformations. See the textbook for their specific expressions of the matrix $\boldsymbol{A}$.

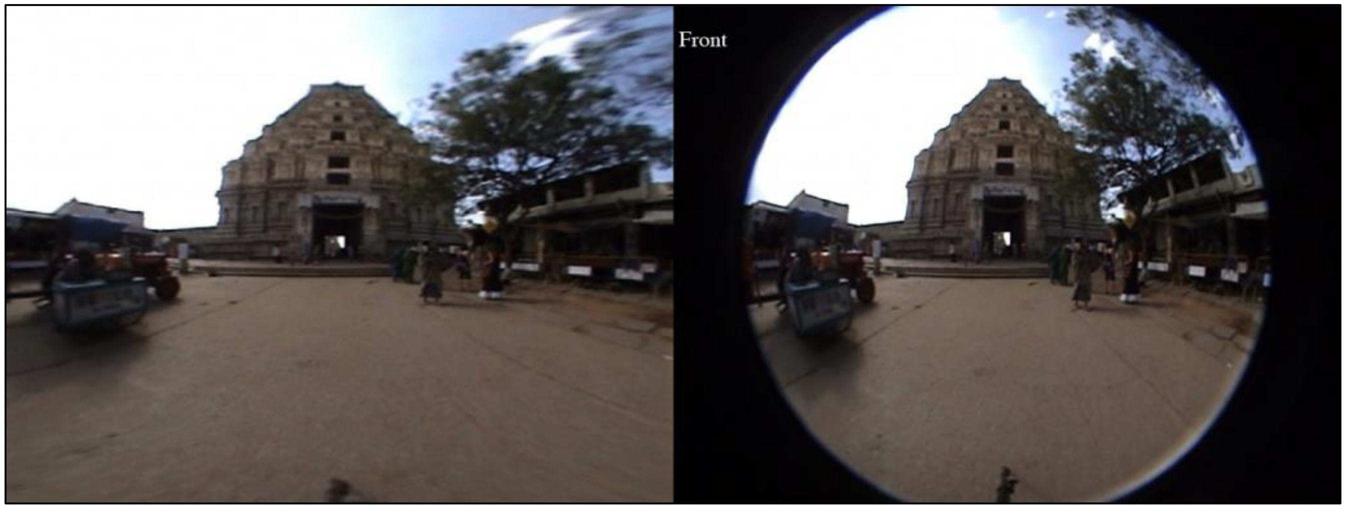
Nonlinear Geometric Transformations



Just examples …

Perspective projection

Fisheye image dewarping



Source of images:
http://info.ee.surrey.ac.uk/Teaching/Courses/eem.cgi/lectures_pdf/lecture3.pdf
http://www.kscottz.com/category/simplecv-2/

Geometric Transformations

- However, to construct the "transformed" image from a source image, the standard practice is to consider the transformation in the opposite direction.
  - This is called ***inverse mapping*** in the textbook.
  - We need to determine the coordinates of all destination pixels in the source image.
  - This is like the resampling of the source image:

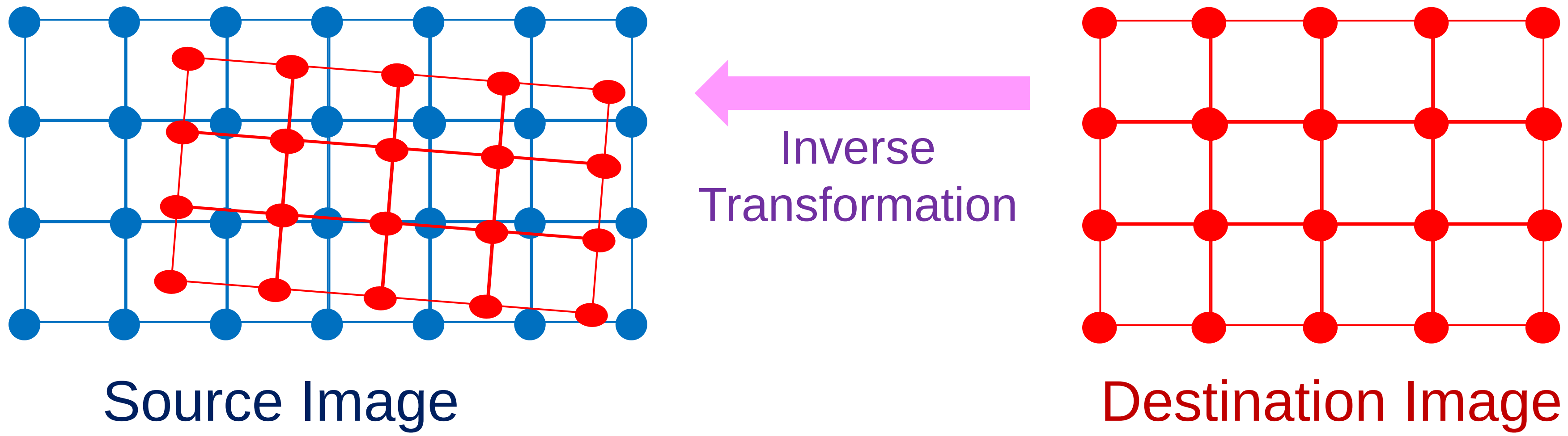
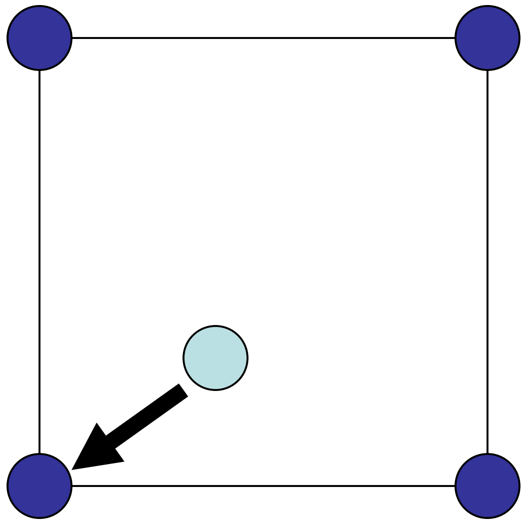


Image Interpolation

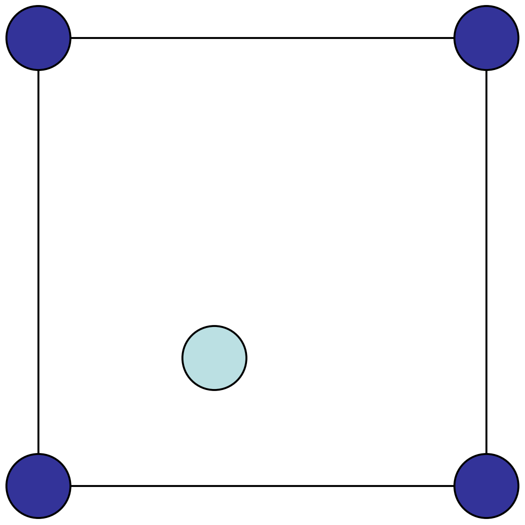
For image resampling as in geometrical transformations, Interpolation is necessary for obtaining the values of pixels on the new grid from the values of the original grid.

Common methods are:

Nearest-neighbor



Bilinear



$$v(x', y') =$$

$$ax' + by' + cx'y' + d$$

Bicubic

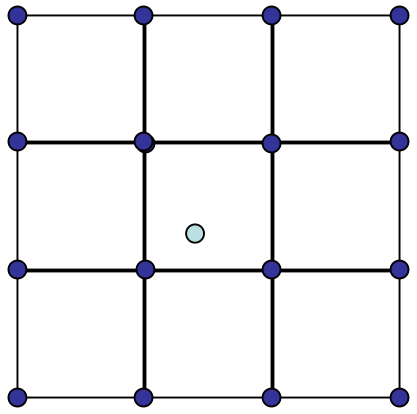
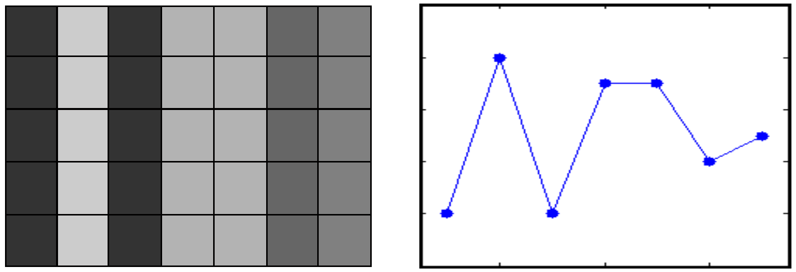


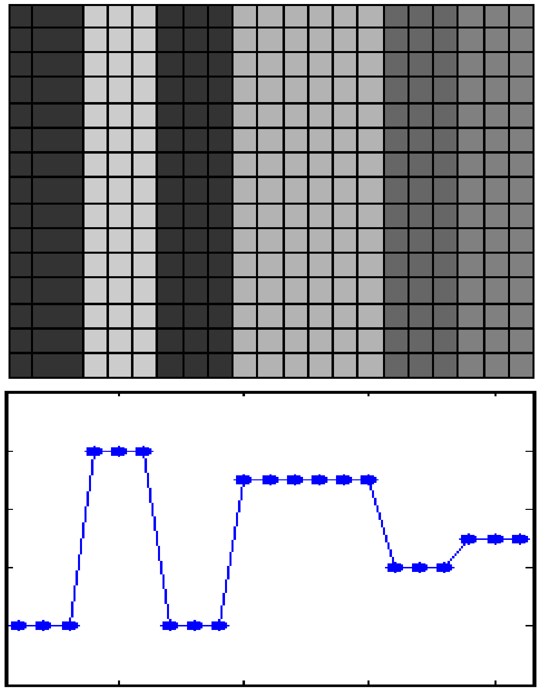
Image Interpolation Methods

Example: Zoom-in in 1-D, for better illustration:

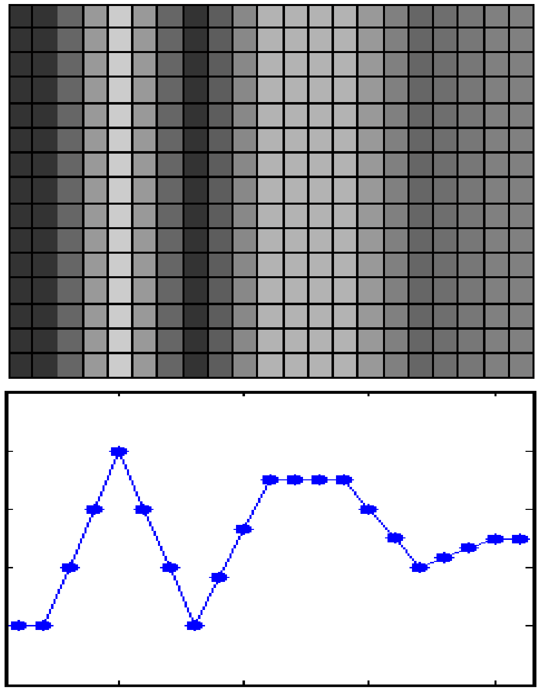
Original



Nearest-Neighbor



Bilinear



Bicubic

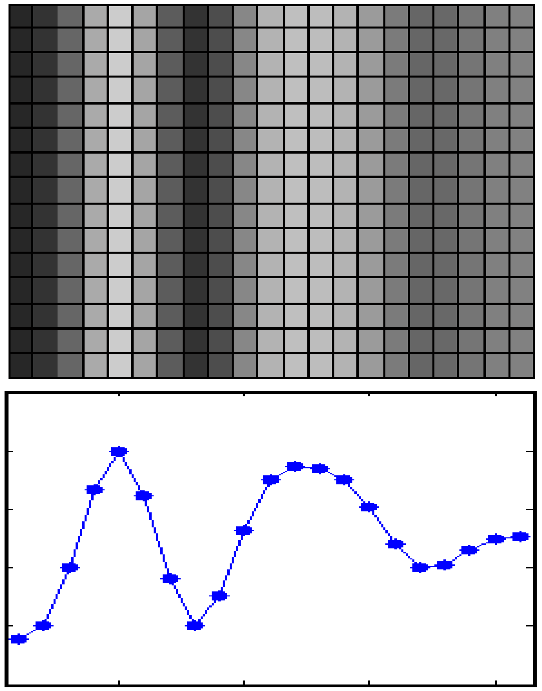
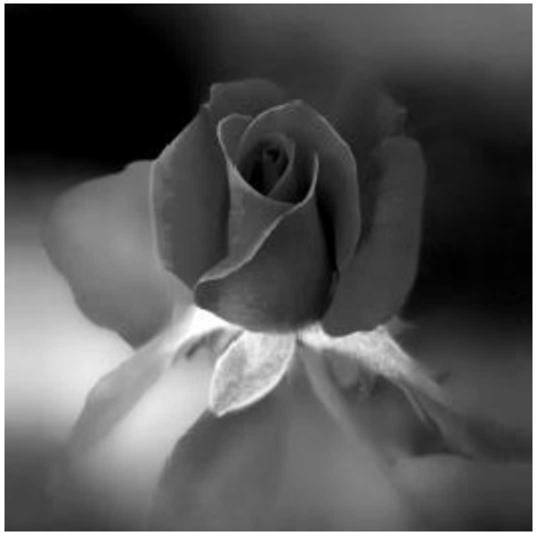


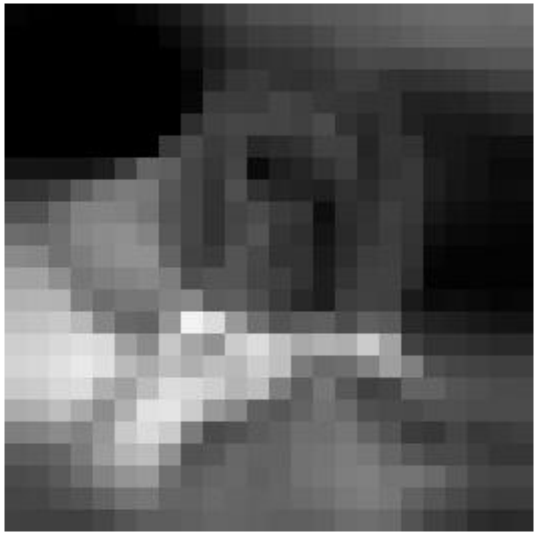
Image Interpolation Methods



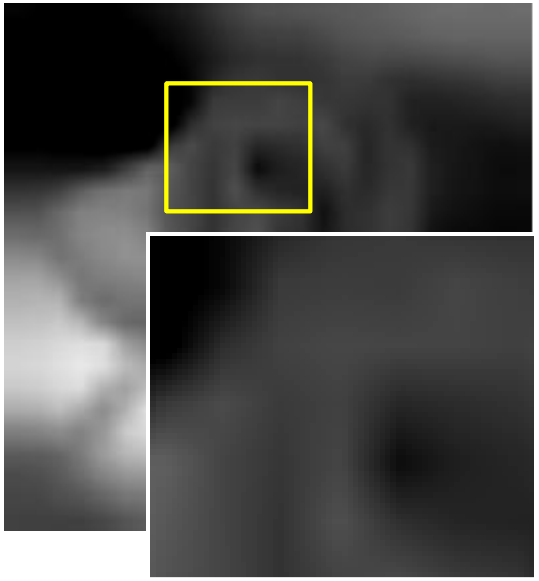
256x256 original
(for comparison)

24x24 original

Nearest-Neighbor



Bilinear



Bicubic

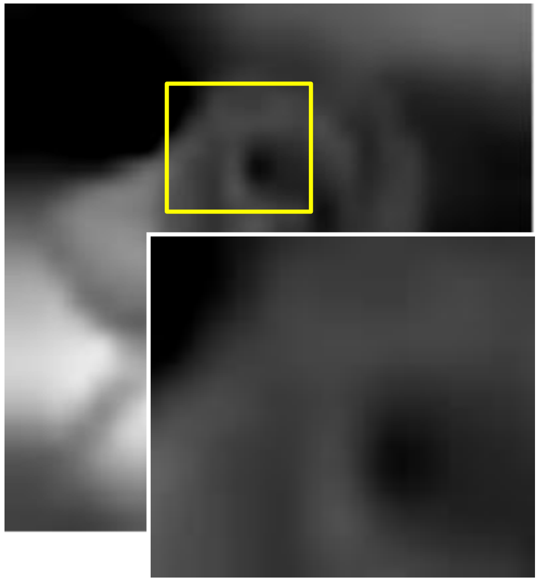


Image Interpolation Methods

- Number of source pixels considered for each destination pixel: Nearest: 1; Bilinear: 4; Bicubic: 16.
- Speed: Nearest > Bilinear > Bicubic
- Smoothness: Nearest < Bilinear < Bicubic
- Used in the <u>digital-zoom</u> function of single-lens digital cameras. (larger image ≠ more image detail)
- **Super-resolution**: Techniques used when enlarging images that

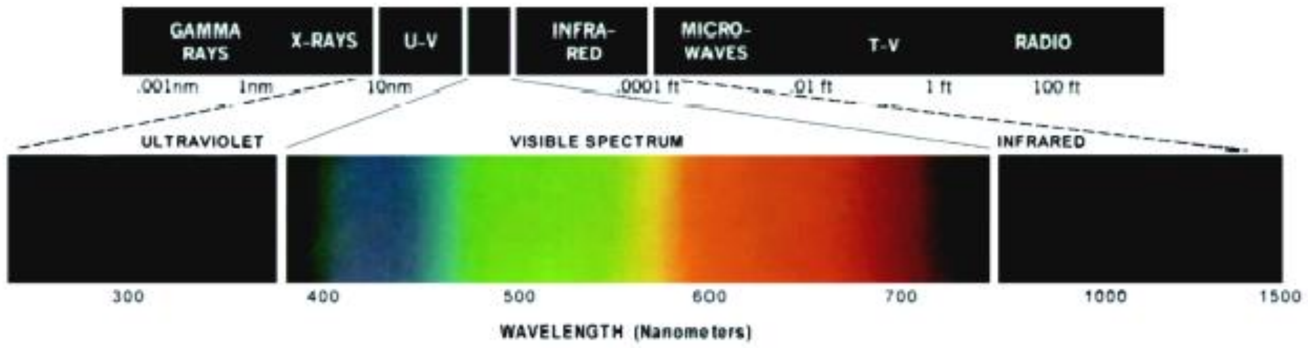
  (1) use more information, and

  (2) produce better visual quality

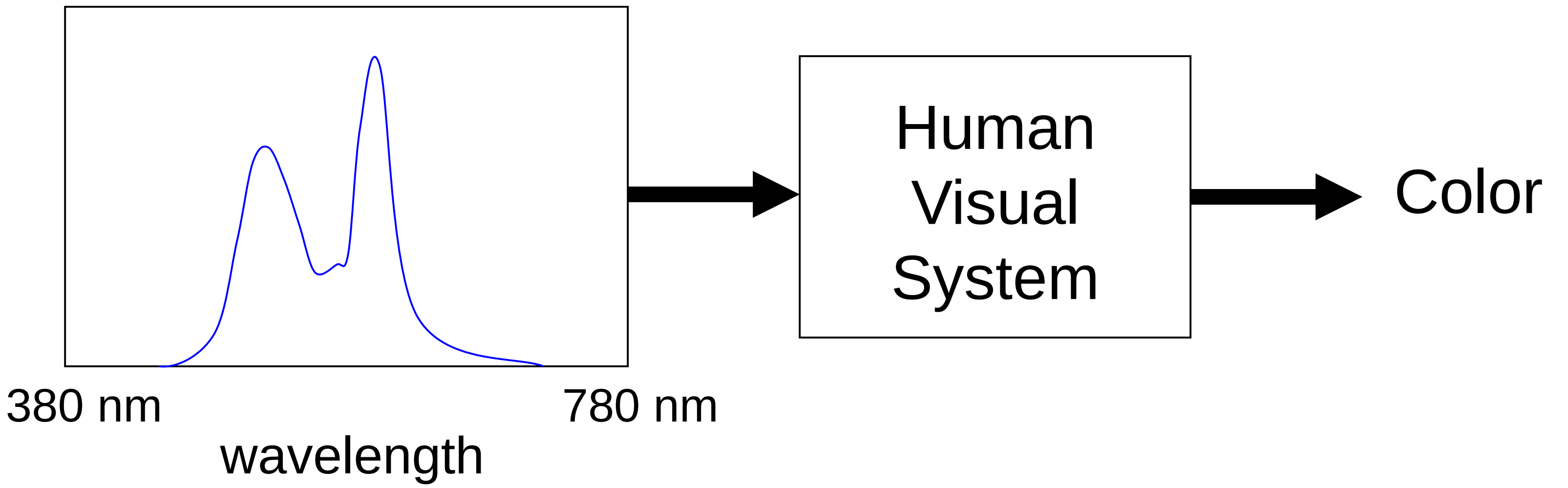
  compared to basic interpolation methods.

Color Perception

Spectrum of EM waves:



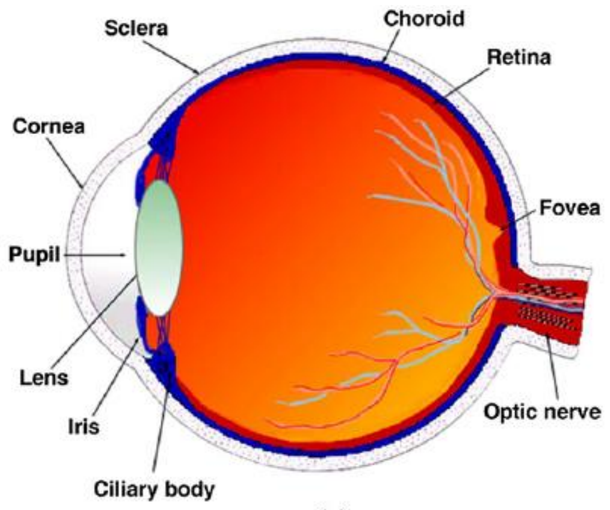
Color is not the underlying physical property (the spectrum of light is), but the human perception of it.



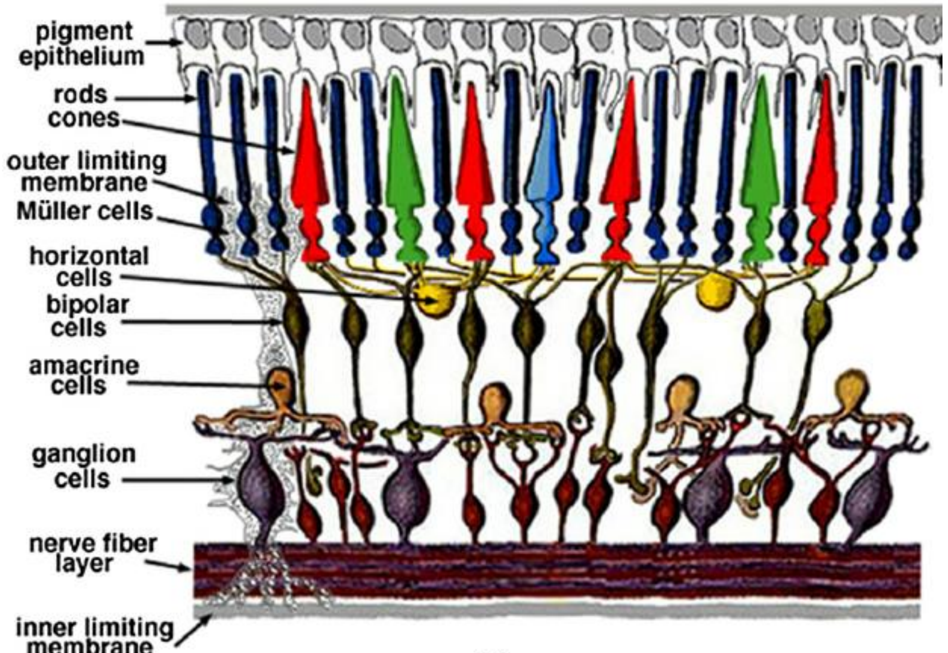
Color Perception

There are two types of cells in human retina that sense light:

- **Cones**: Bright light vision; 3 types; Color perception
- **Rods**: Low light vision



(a)

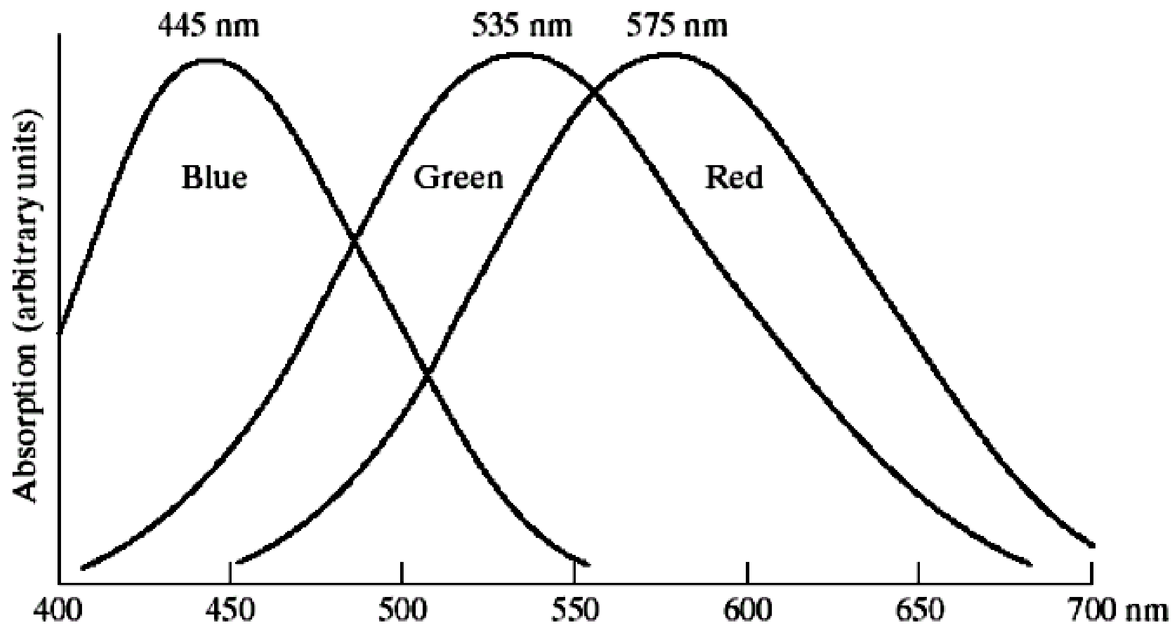


(b)

https://www.researchgate.net/publication/263289543_Retinal_Imaging_as_a_Source_of_Biomarkers_for_Diagnosis_Characterisation_and_Prognosis_of_Chronic_Illness_or_Long-Term_Conditions/figures?lo=1

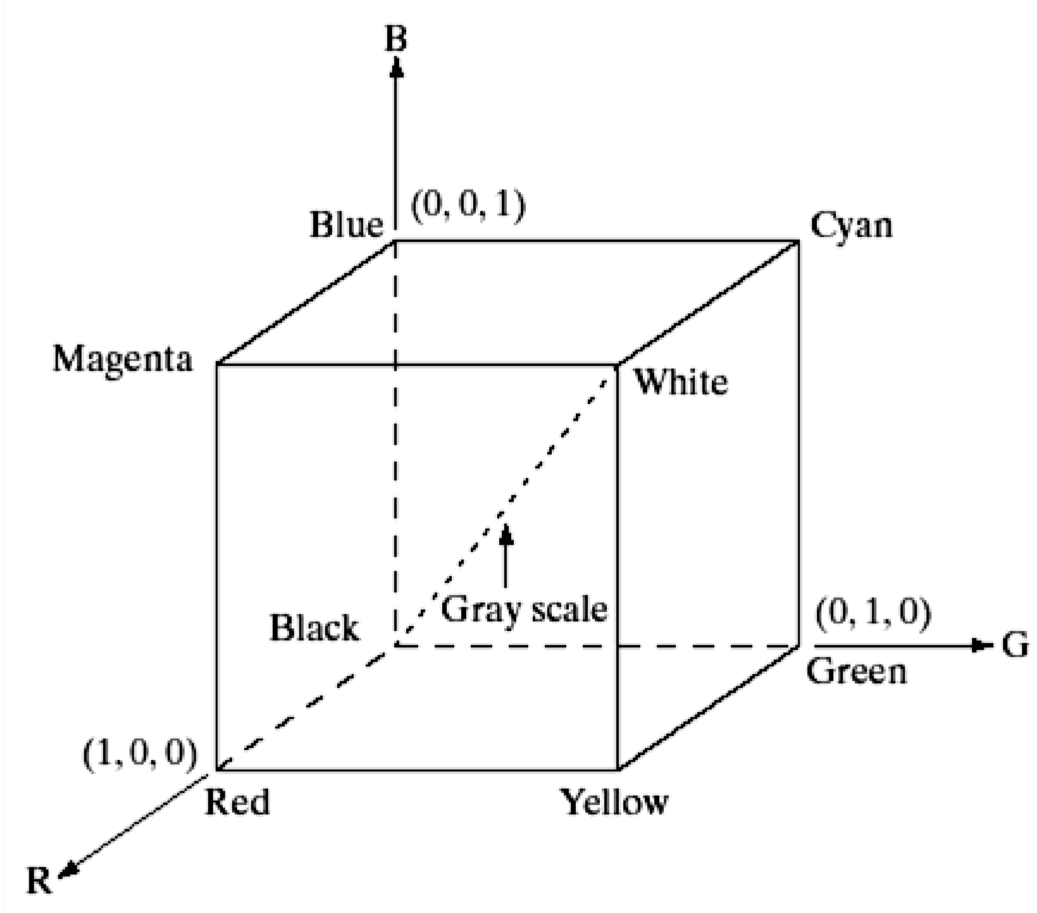
Color Perception

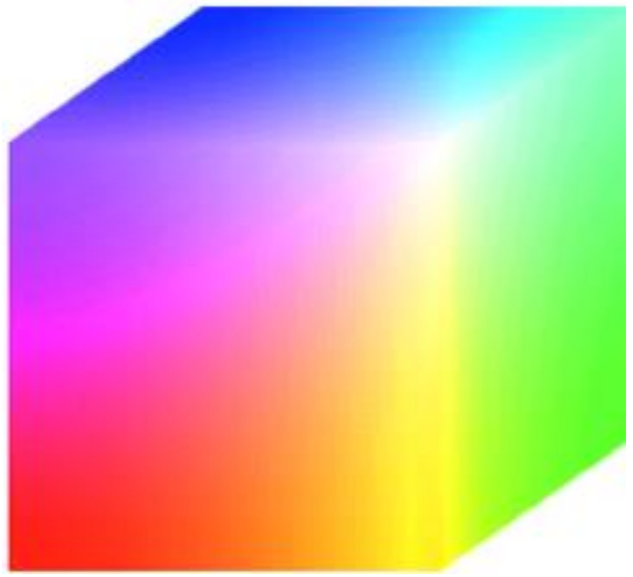
- The reason we use three components (e.g., RGB) to represent colors is because there are three types of cone cells in human retinas.
- The human visual system converts the outputs of the 3 types of cones (3 color components) into one **luminance** and two **chrominance** values.



RGB Color Model

- Corresponding to the cones in human retina
- Representing the "content of light"
- Common for displays





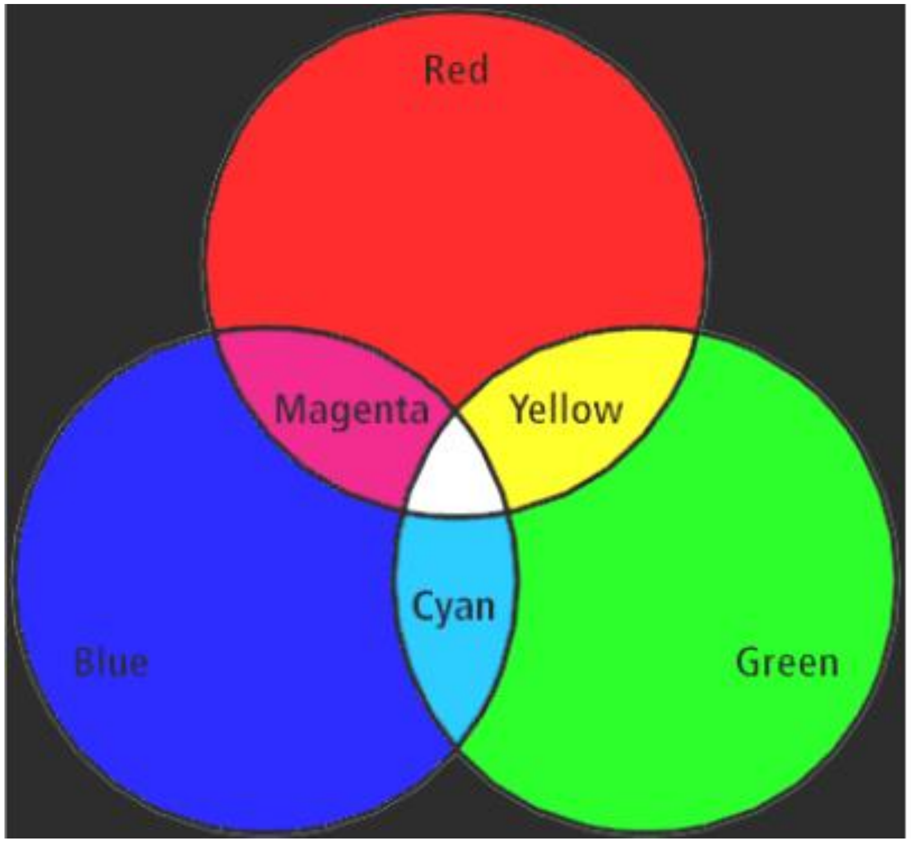
CMY / CMYK Color Models

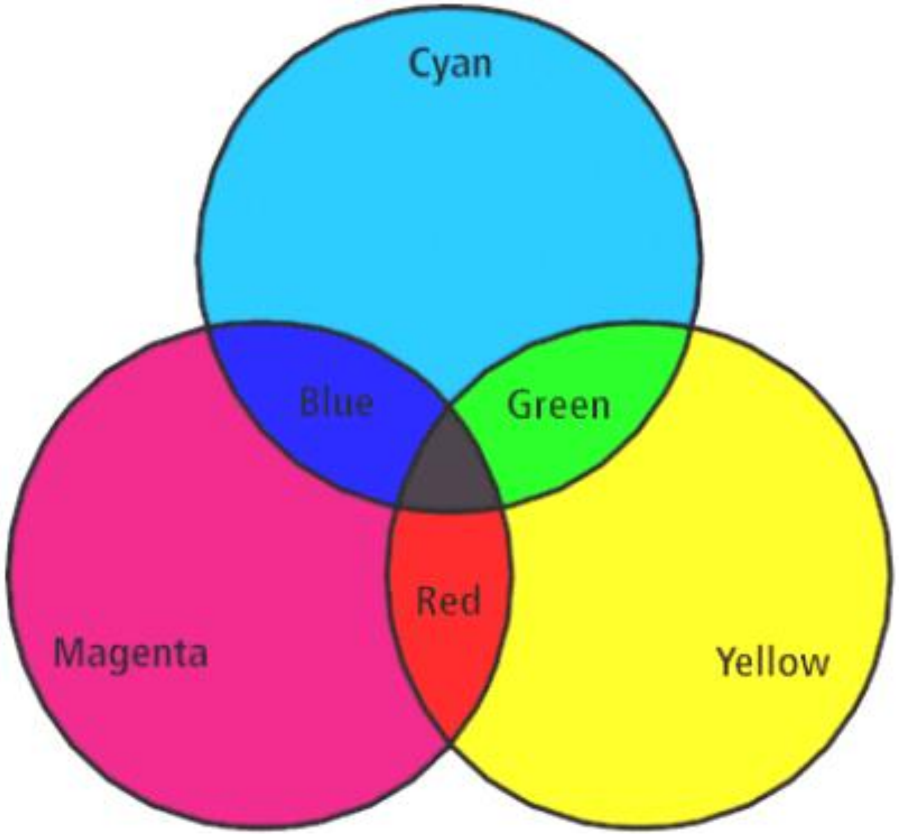
$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

- Opposite (complement) of RGB
- Used for altering the content of light by changing how a surface reflects light
- Common for color printing devices
- CMYK = CMY + black

$$C \leftarrow (C - K)/(1 - K)$$
$$M \leftarrow (M - K)/(1 - K)$$
$$Y \leftarrow (Y - K)/(1 - K)$$
$$C, M, Y \leftarrow 0 \quad \text{if} \;\; K = 1$$

RGB and CMY Colors





Luminance+Chrominance Models

- Color models that have one luminance component and two chrominance components
- This matches better with how human understand and process colors.
- Human visual system does not respond to luminance and chrominance, or even different colors, equally.
- Transform to luminance and chrominance helps to provide the best human viewing quality with limited resources (data storage and transmission bandwidth).
    - In lossy image/video compression, chrominance is usually subsampled relative to luminance. For example, MPEG-1/2 uses 4:2:0 chrominance subsampling (only 1/4 chrominance samples).

Luminance+Chrominance Models

Types of Luminance/Chrominance Specifications:

- HSI (hue, saturation, intensity), HSV, HSL
- YUV and YIQ (analog TV standards)
- YC_bC_r (digital TV, JPEG compression)

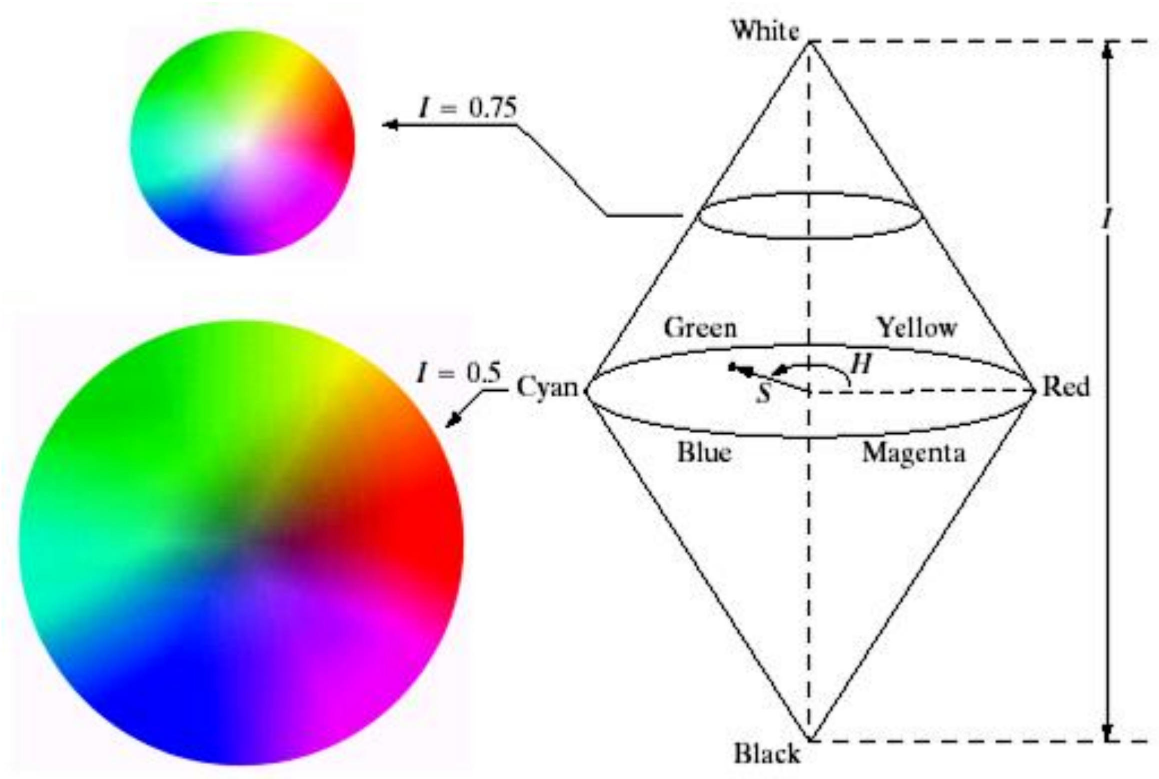
Transformation between Y** and RGB is linear. For example:

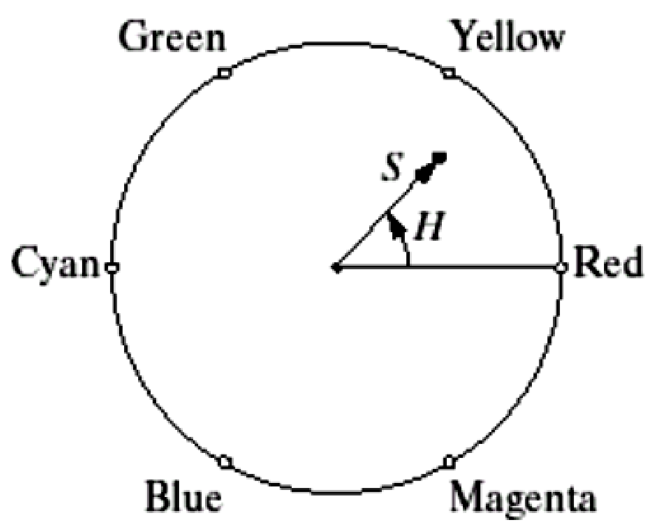
$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.275 & -0.321 \\ 0.212 & -0.523 & 0.311 \end{bmatrix} \begin{bmatrix} \tilde{R} \\ \tilde{G} \\ \tilde{B} \end{bmatrix}$$

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 0.257 & 0.504 & 0.098 \\ -0.148 & -0.291 & 0.439 \\ 0.439 & -0.368 & -0.071 \end{bmatrix} \begin{bmatrix} 255\tilde{R} \\ 255\tilde{G} \\ 255\tilde{B} \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix}$$

HSI Color Model

- **Hue**: The pure-color component
- **Saturation**: How pure the color is
- Intensity: A measure of brightness





HSI Color Model

RGB to HSI conversion:

$$H = \begin{cases} \theta & \text{if } B \le G \\ 360^\circ - \theta & \text{if } B > G \end{cases}$$

$$\theta = \cos^{-1}\left\{ \frac{[(R-G)+(R-B)]/2}{\sqrt{(R-G)^2+(R-B)(G-B)}} \right\}$$

$$I = (R+G+B)/3$$

$$S = 1 - \frac{3}{R+G+B}\min(R,G,B) = 1 - \frac{\min(R,G,B)}{I}$$

HSI Color Model

To understand the formula for saturation:

- Separate the color into a “pure part” (at least one component being zero) and a “gray part” (all components being equal).
- Example: $[0.3, 0.4, 0.8] = [0, 0.1, 0.5] + [0.3, 0.3, 0.3]$
- Here intensity $= 0.5$, and the intensity of the “gray part” is 0.3. So the ratio of gray part is $0.3 / 0.5 = 0.6$.

  ➔ Saturation $= 1 – 0.3 / 0.5 = 0.4$

HSV and HSL Color Models

Similar to HSI in terms of meanings, but the actual numbers are different. These are more commonly used in actual systems.

$$MIN = \min(R, G, B) \qquad MAX = \max(R, G, B)$$

$$H = \begin{cases} \text{undefined}, & \text{if } MAX = MIN \\ 60^\circ \times \dfrac{G-B}{MAX - MIN}, & \text{if } MAX = R \text{ and } G \geq B \\ 60^\circ \times \dfrac{G-B}{MAX - MIN} + 360^\circ, & \text{if } MAX = R \text{ and } G < B \\ 60^\circ \times \dfrac{B-R}{MAX - MIN} + 120^\circ, & \text{if } MAX = G \\ 60^\circ \times \dfrac{R-G}{MAX - MIN} + 240^\circ, & \text{if } MAX = B \end{cases}$$

HSV and HSL Color Models

HSV (V=value):

$$V = MAX \qquad S = 1 - \frac{MIN}{MAX}$$

HSL (L=Luminance):

$$L = (MAX + MIN)/2$$

$$S = \begin{cases} 0 & \text{if } L = 0 \\ (MAX - MIN)/(2L) & \text{if } 0 < L \leq 0.5 \\ (MAX - MIN)/(2 - 2L) & \text{if } L > 0.5 \end{cases}$$

HSV/HSL to RGB

HSV: $MAX = V, \quad MIN = (1-S)V$

HSL: $L = 0 \quad \Rightarrow \quad MAX = MIN = 0$

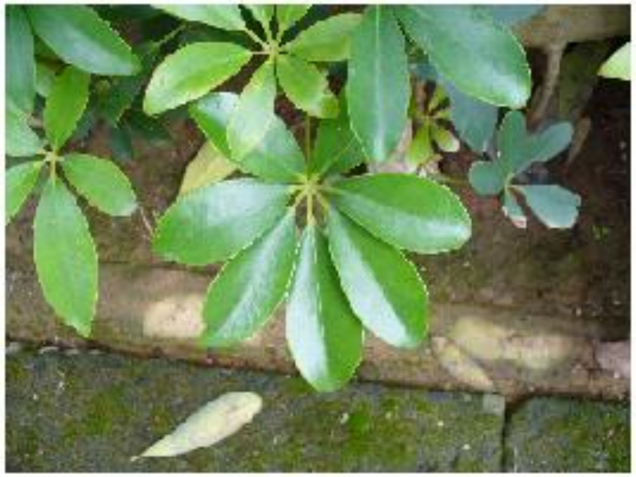
$0 < L \leq 0.5 \quad \Rightarrow \quad MAX = L(1+S), \quad MIN = L(1-S)$

$L > 0.5 \quad \Rightarrow \quad MAX = L + (1-L)S, \quad MIN = L - (1-L)S$

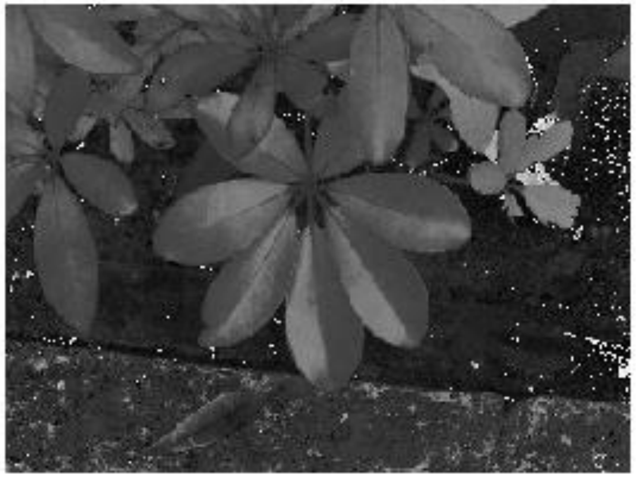
- In these two color spaces, we are ensured that any combination of V (or L) and S values in $[0,1]$ results in valid RGB values (in $[0,1]$). This is not the case for the HSI color space.
  - For example, MS PowerPoint provides HSL, and MATLAB provides HSV.
- Need to use Hue as well as the MAX and MIN values to compute RGB values.

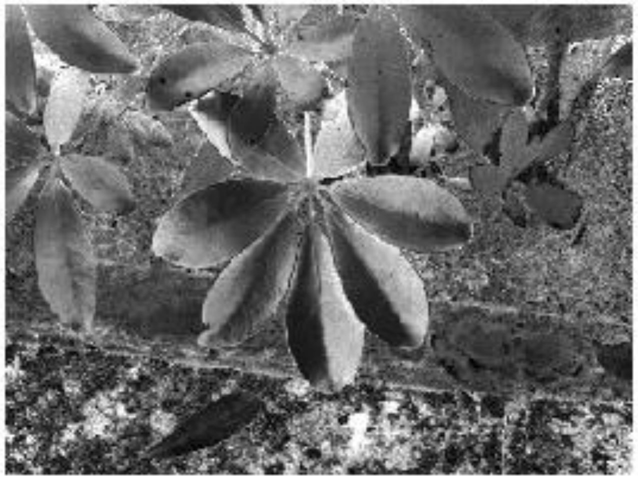
HSV Components

Original

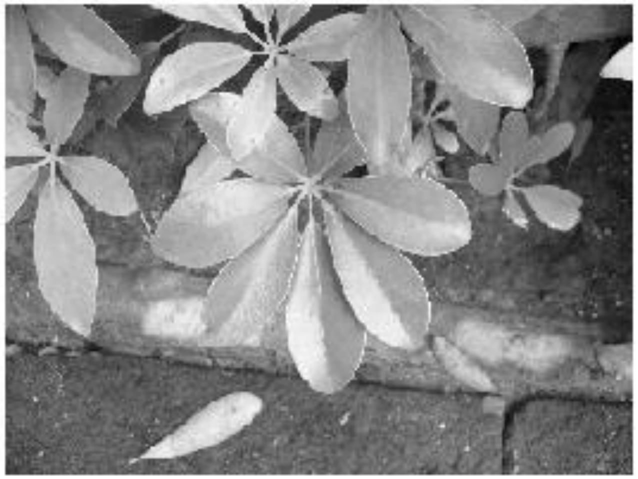


Hue



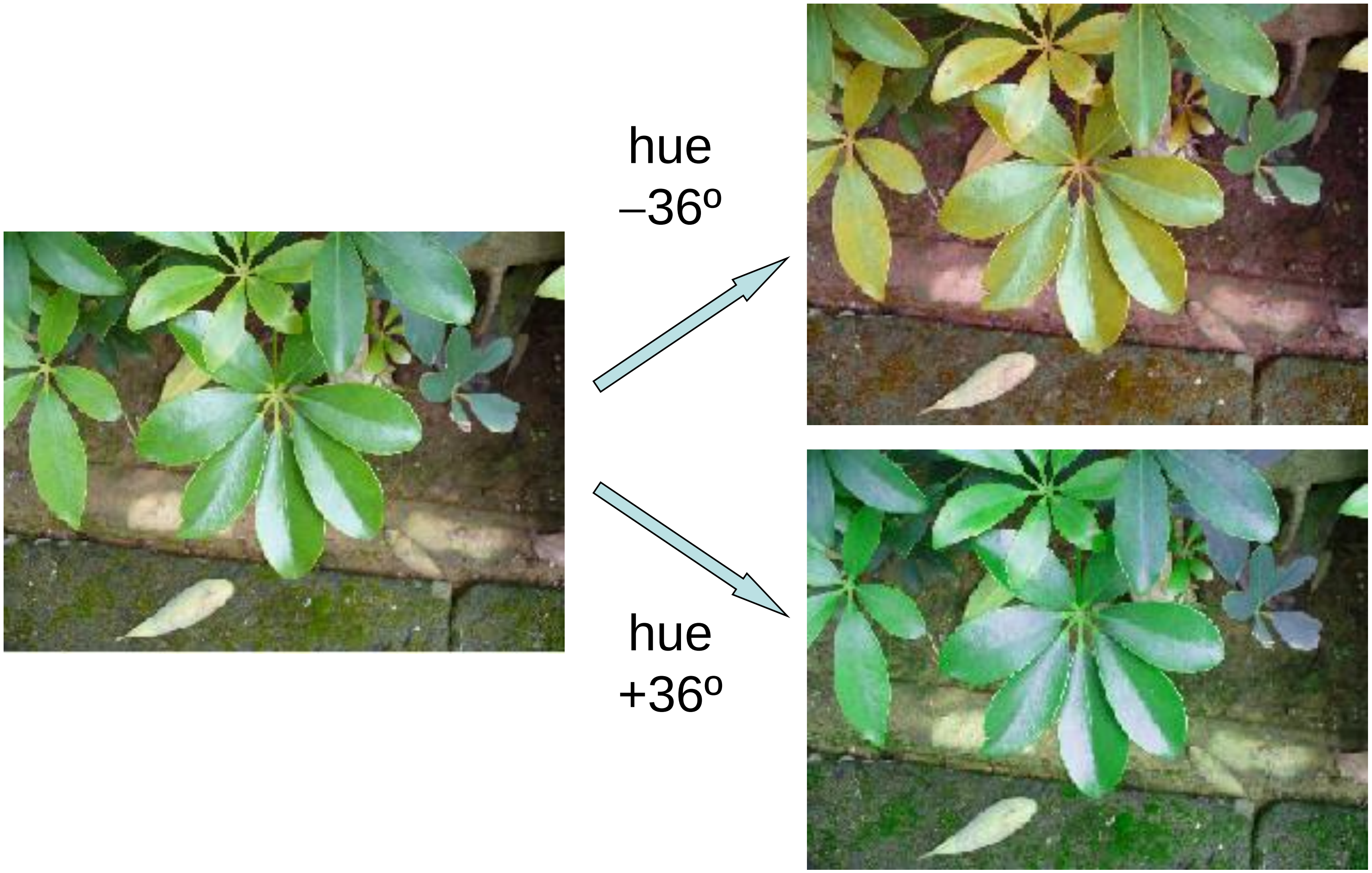


Saturation

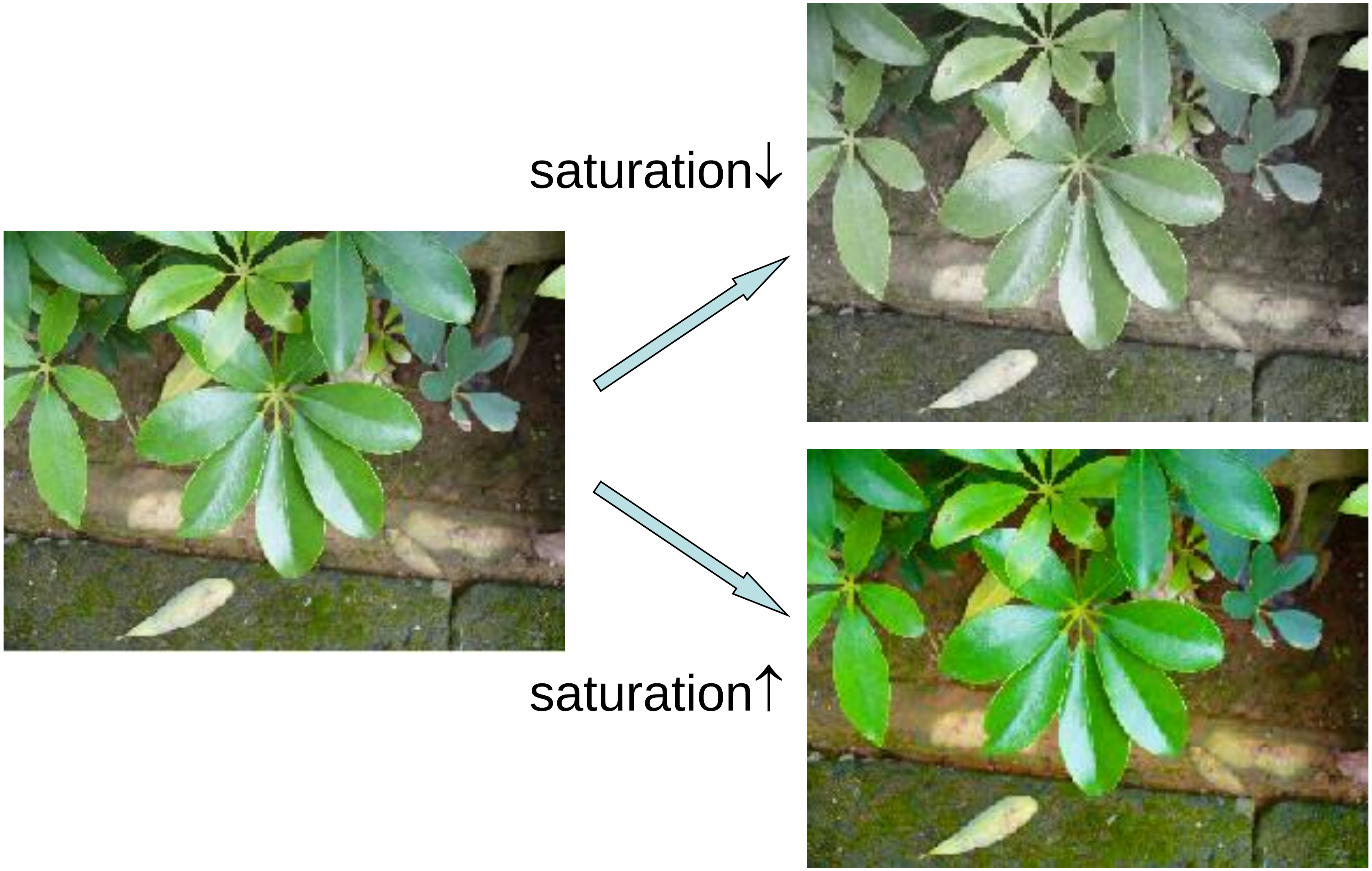


Value

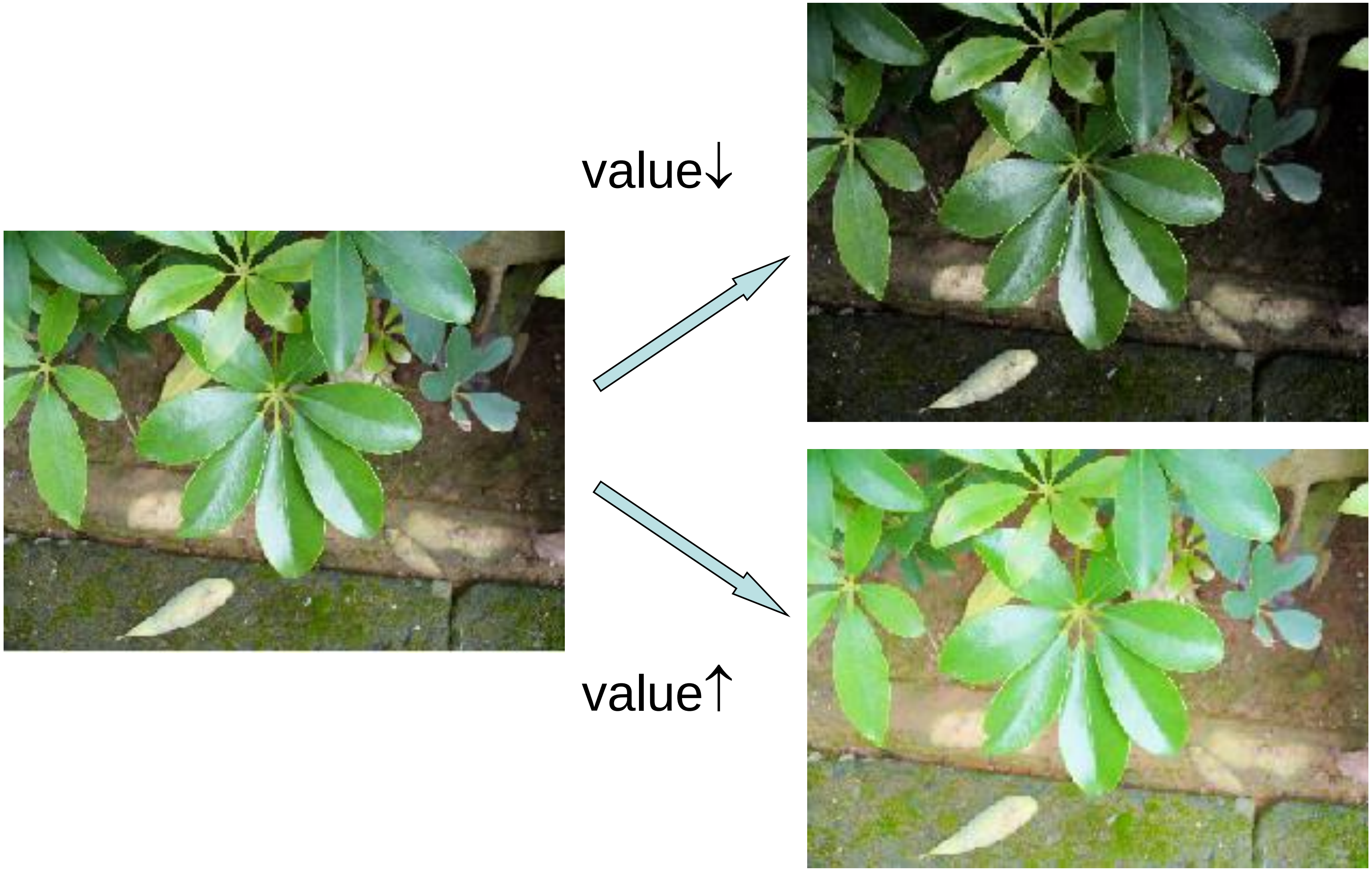
Manipulating HSV Components



Manipulating HSV Components

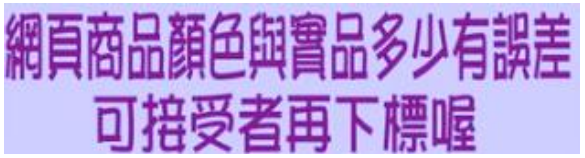


Manipulating HSV Components



CIE XYZ and CIE LAB Color Models

- Common color models such as RGB are device-dependent. For example, the same RGB values can correspond to different colors on different computer screens.



網頁商品顏色與實品多少有誤差
可接受者再下標喔

- For the most accurate color reproduction, we need to be able to convert colors across devices.

- A device-independent color model can serve as the medium. Each device only needs to know how to transform between its own (displayed, printed) colors and the device-independent model. (Such transformations are the **color profiles** of the devices.)

CIE = International Commission on Illumination

CIE XYZ Color Models

- This defines a color space where all visible colors are in the range of $X, Y, Z \in [0,1]$.
- These are called the **tristimulus values** (degrees of stimulus to the three types of cones in the retina).
- The color-only part can be represented using x (red) and y (green) only:

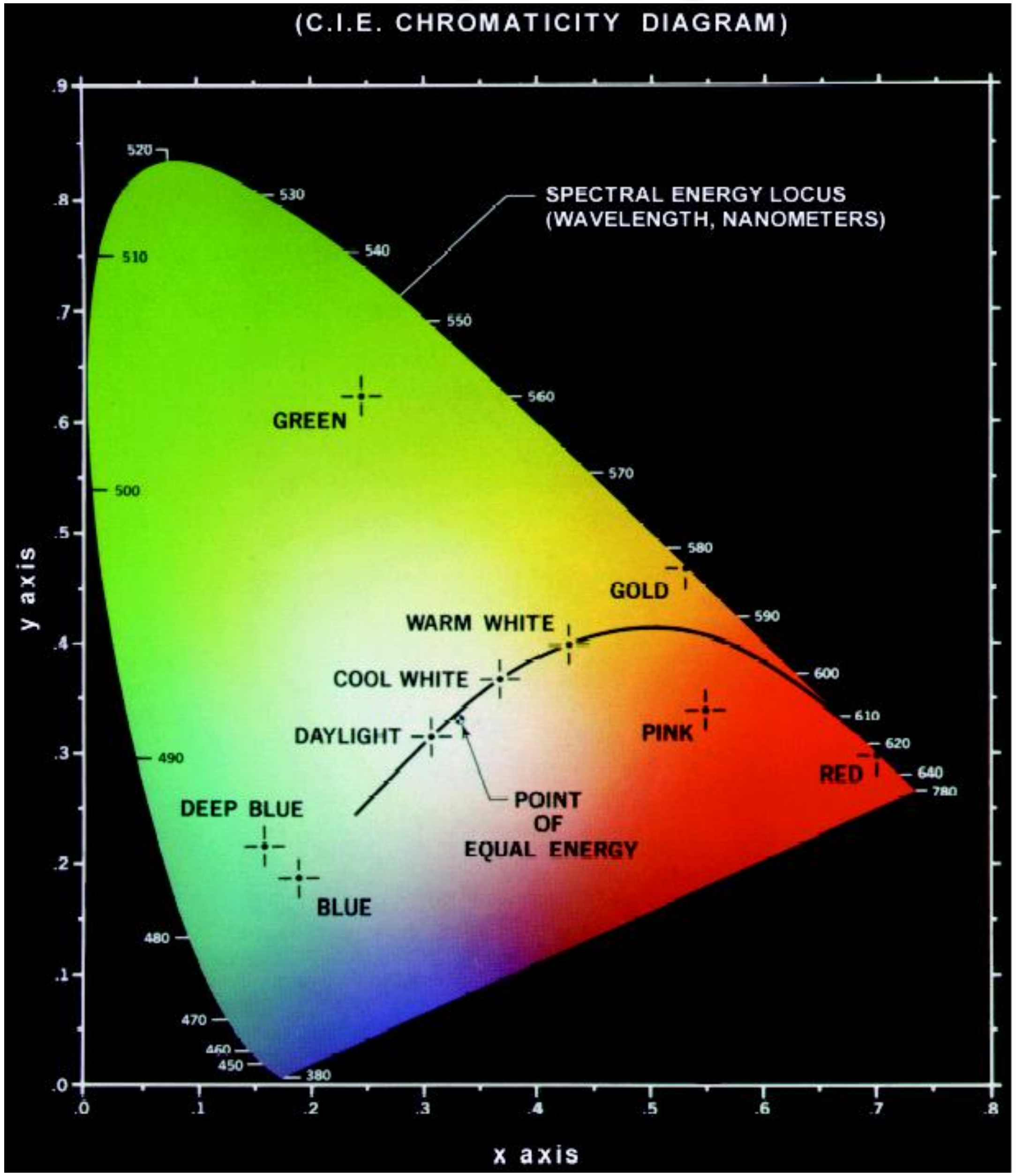
$$x = X/(X+Y+Z)$$
$$y = Y/(X+Y+Z)$$
$$z = Z/(X+Y+Z)$$
$$x + y + z = 1$$

- Some xyz values, such as (1,0,0), do not actually exist. The simple way to understand this: The responses of the three types of cones are not independent of each other.

CIE XYZ and Chromaticity Diagram

This contains all the colors visible to human.

A straight line segment indicates possible results of additive color mixing.



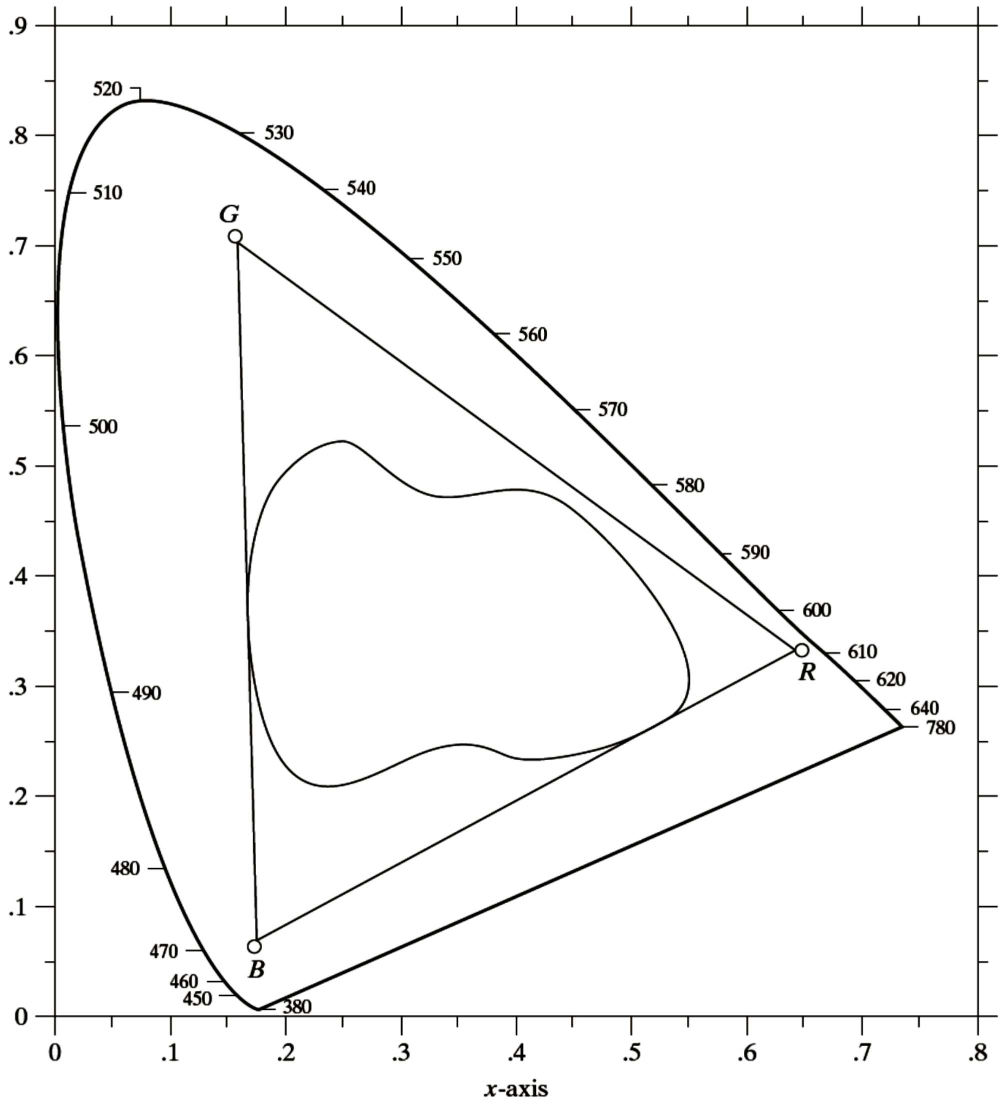
CIE Chromaticity Diagram

Color gamut:
Range of reproducible colors by a device.

Displays have triangular color gamuts.

Printing devices can have more irregular gamuts.

The color model sRGB represents a standard gamut for common CRTs.



CIE LAB Color Model

- CIE LAB is the most commonly used device-independent color model.
- Three components: **L***, **a***, **b***.
- Each visible color can be converted to L*a*b* values through the comparison with an "objective" white color (the perfect diffuse reflection of a particular white light source at a particular viewing angle).
- L*a*b* color model is *perceptually uniform*. This means that the perceived difference between two colors is proportional to their Euclidean distances in this color model.