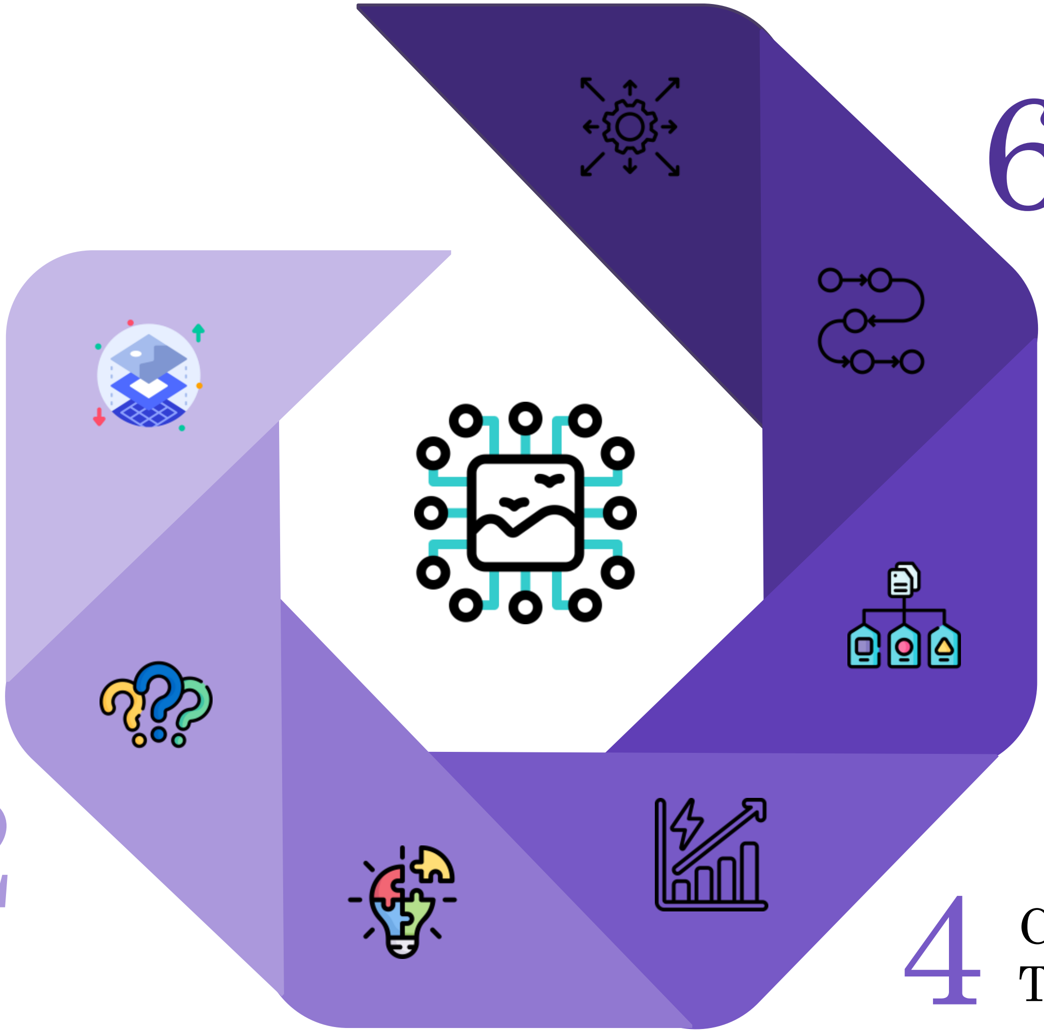
Digital Image Processing

Lecture 07

Color Image Processing

Dr. Muhammad Sajjad
R.A: Asad Ullah

Overview



1 Introduction

2 Color Fundamentals

3 Color Models

4 Color Transformations

5 Histogram Processing

6 Color Image Smoothing

7 Color Image Sharpening

Introduction

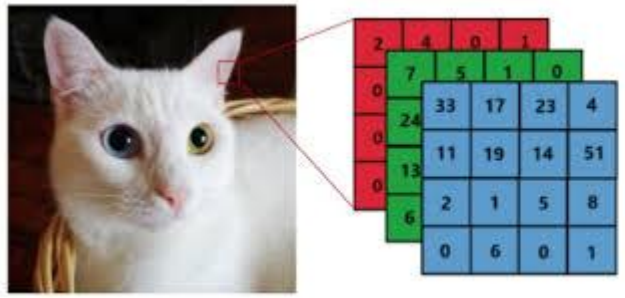
The process of analyzing and modifying color information in digital images.

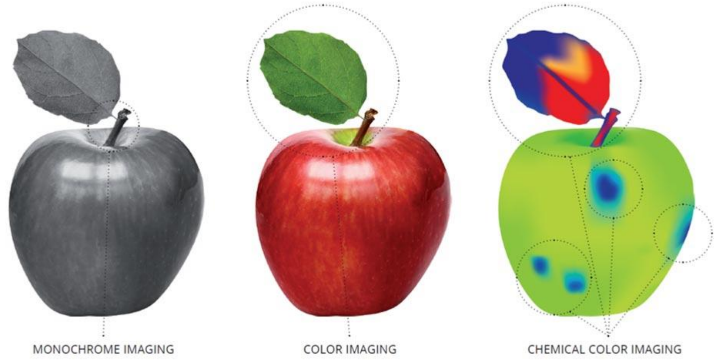
Motivation for Color Image Processing

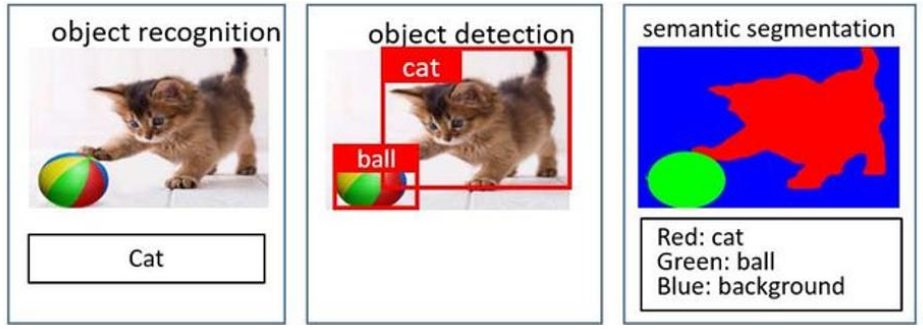
- Color is a powerful descriptor that simplifies object identification and extraction.
- Humans can distinguish thousands of color, but only about two dozen shades of gray.

Two Areas of Color Image Processing

1. **Full-Color Processing:**
   - Uses images captured by full-color devices (e.g., cameras, scanners).
   - Operates on all color components like RGB.
2. **Pseudo color Processing:**
   - Assigns colors to grayscale images for better visualization.
   - Common in medical imaging and satellite imagery.







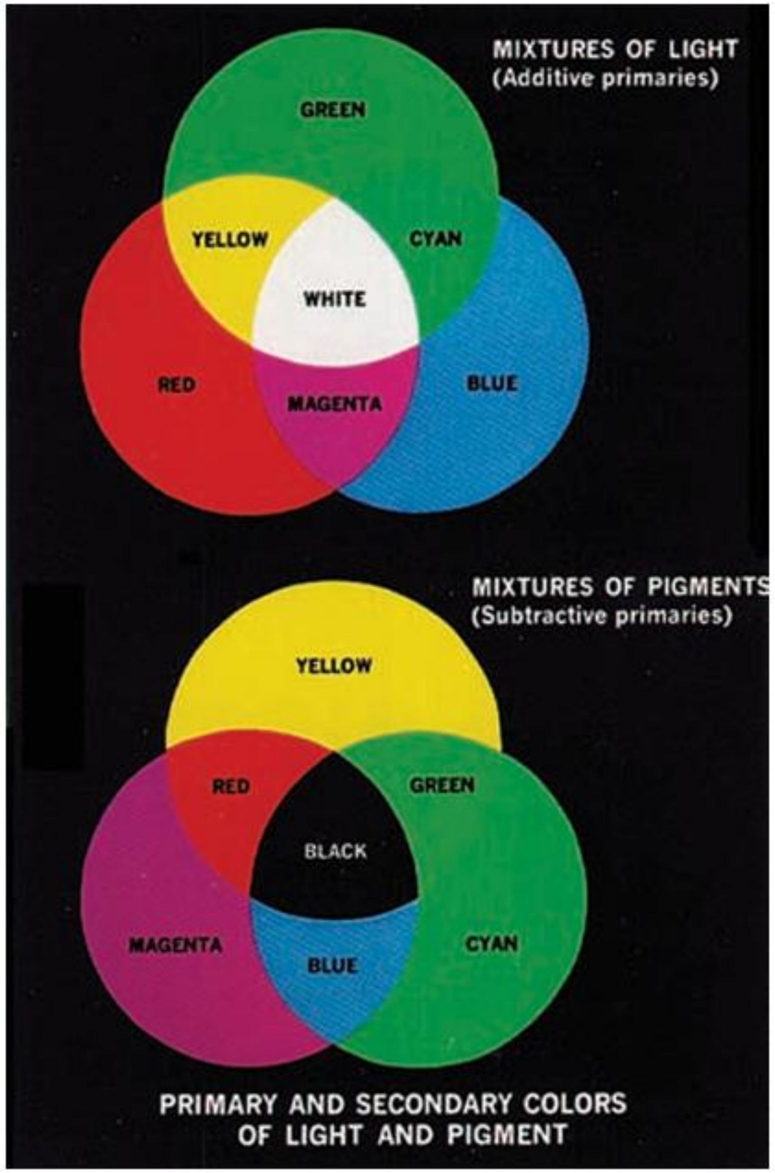
Color Fundamentals

Primary Colors of Light (RGB):

- **Red (R), Green (G), and Blue (B)**
- Mixing RGB produces **secondary colors**:
  - **Magenta** (Red + Blue)
  - **Cyan** (Green + Blue)
  - **Yellow** (Red + Green)
- Combining all three primaries or a secondary with its opposite primary creates white light.

Primary Colors of Pigments (CMY):

- **Cyan (C), Magenta (M), and Yellow (Y)**
- Mixing CMY produces **secondary colors**:
  - **Red** (Magenta + Yellow)
  - **Green** (Cyan + Yellow)
  - **Blue** (Cyan + Magenta)
- Combining all three pigment primaries or a secondary with its opposite primary produces black.



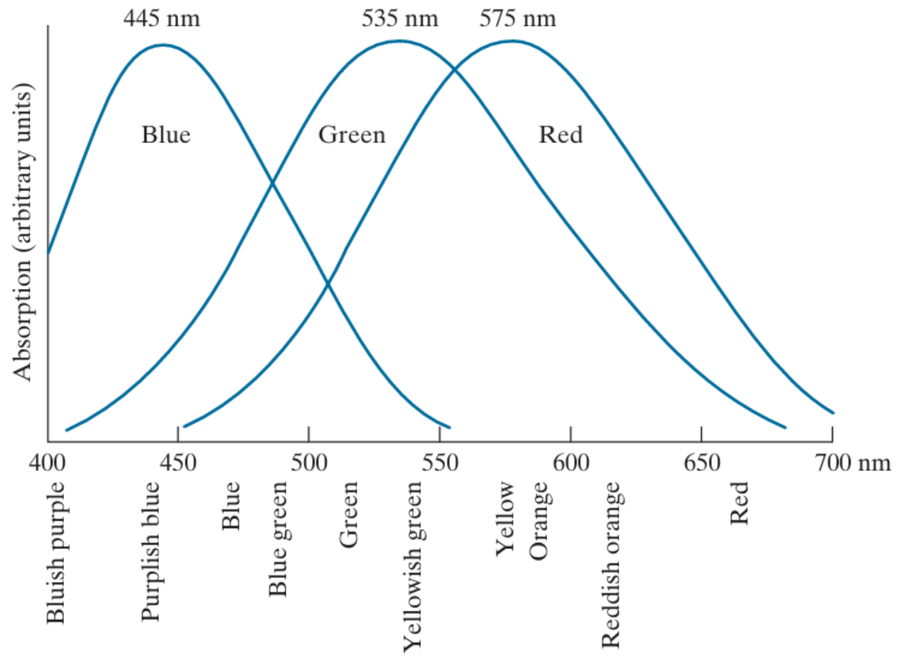
Color Fundamentals

Additive Color Model: Used in displays, combines light (e.g., RGB in screens).

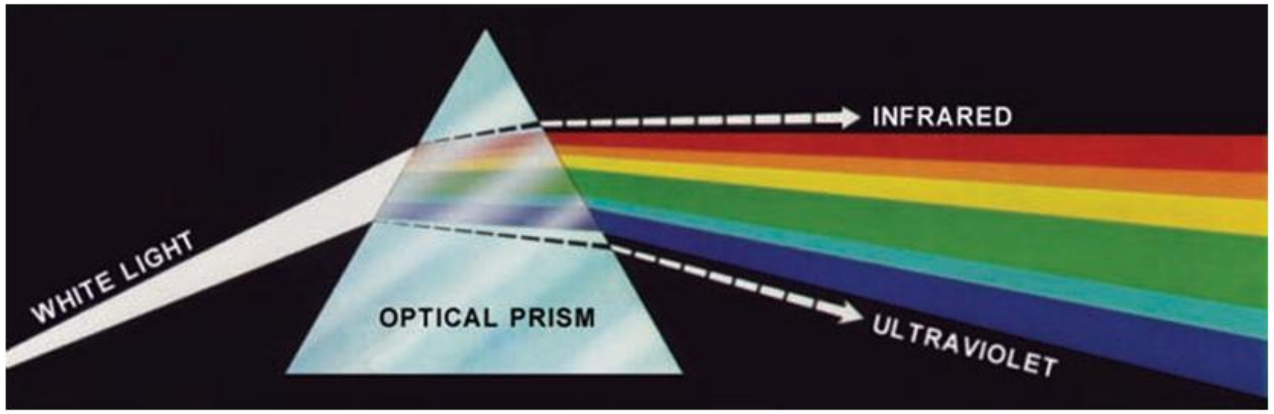
Subtractive Color Model: Used in printing, combines pigments (e.g., CMY in printers).

Human vision is trichromatic, perceiving colors through **cone cells** sensitive to red, green, and blue wavelengths.

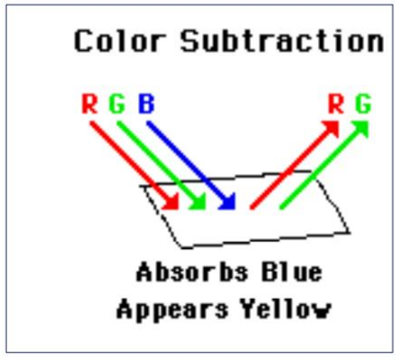
White light splits into colors through a prism, showcasing the visible spectrum.

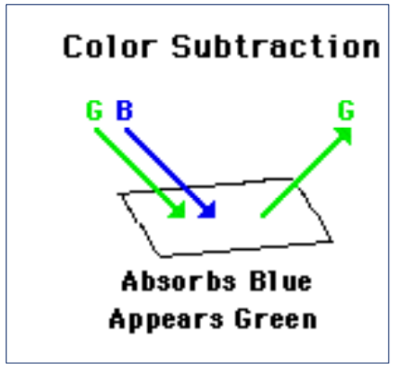


Absorption of light by the red, green, and blue cones in the human eye as a function of wavelength.



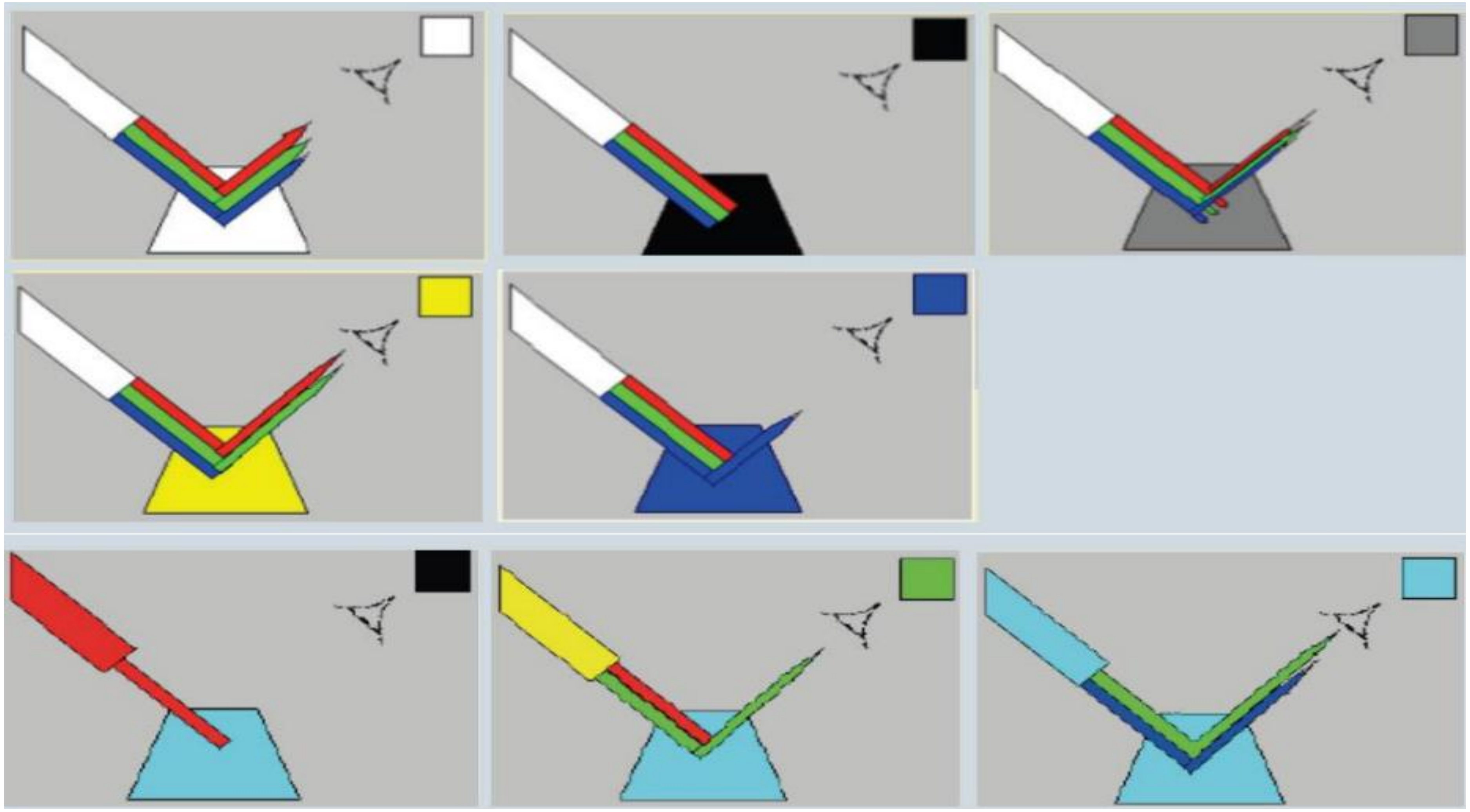
Color spectrum seen by passing white light through a prism.





Color Fundamentals

Light Reflection and Dispersion Across Colored Surfaces



Color Models

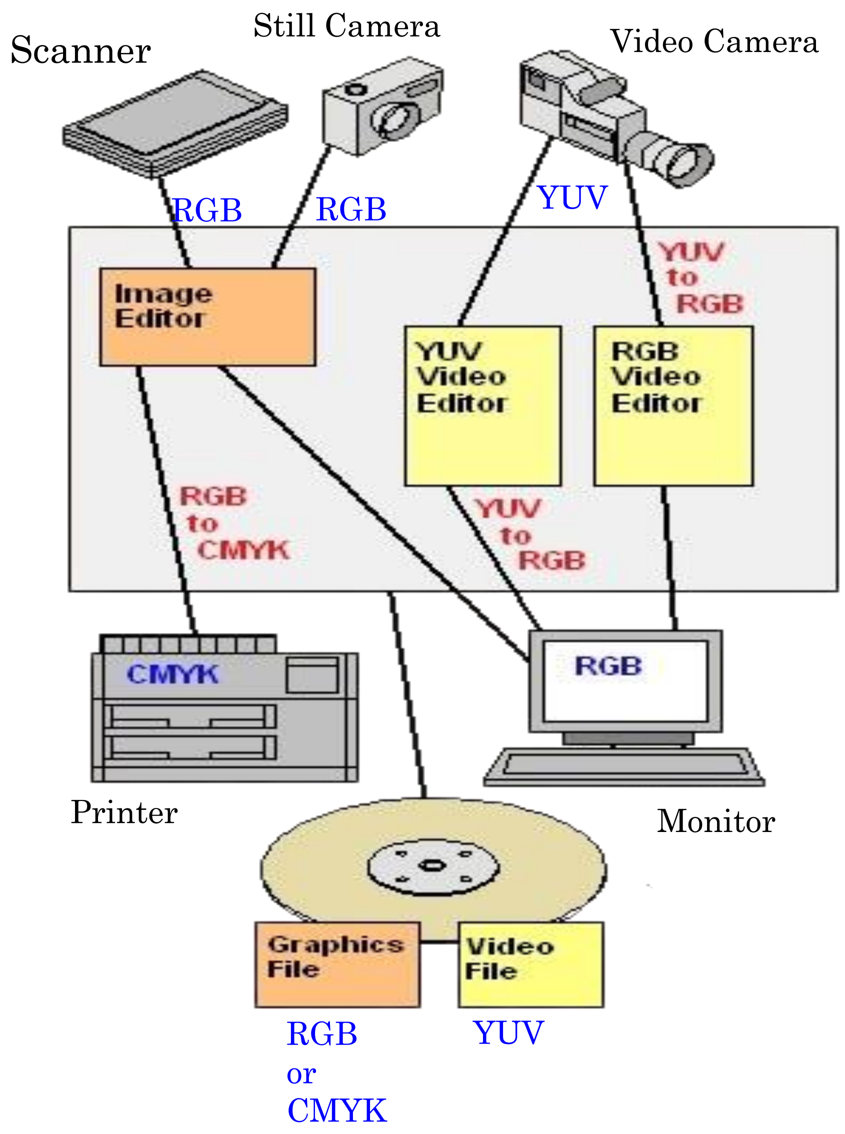
Purpose of Color Models:

- A color model (or color space/system) standardizes the specification of color.
- It defines:
  1. A coordinates system.
  2. A subspace where each color corresponds to a unique point.

Types of Color Models:

1. **Hardware-Oriented Models:**
   - RGB: Used in color monitors and Video Cameras.
   - CMY/CMYK: Used in color printing.
2. **Application Oriented Models**
   - HIS: Matches human perception of color.

Color Space Conversion



Color Models

RGB Color Model

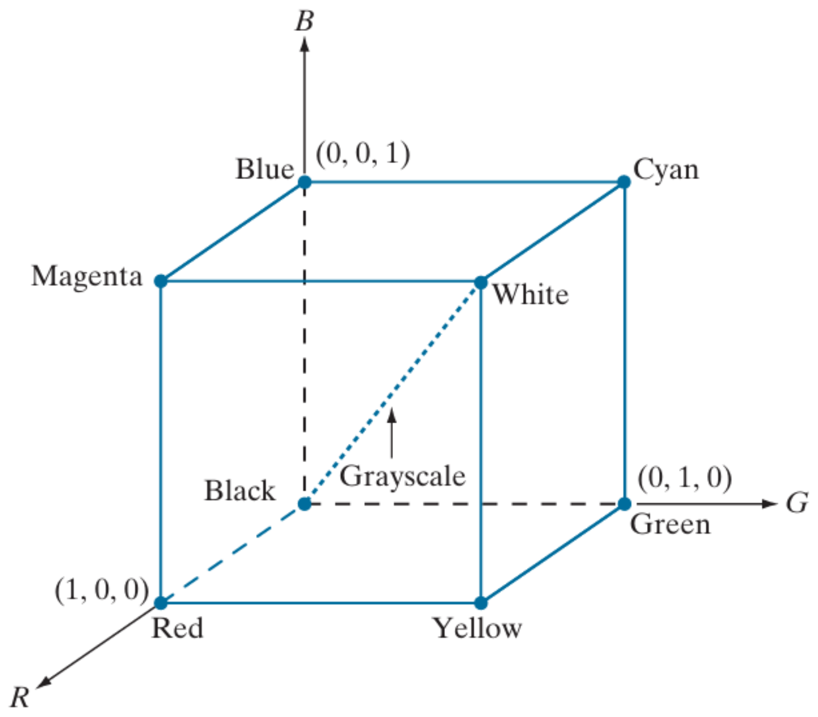
- Represents color as a combination of three primary colors: Red (R), Green (G), and Blue (B).
- Based on a Cartesian coordinates system with a cube representing the color space.

Primary Colors: Red, Green, and Blue at three corners of the cube.

Secondary Colors: Cyan, Magenta, and Yellow at three other corners.

Pixels Depth:

- Each pixel is represented by three 8-bit values (R, G, B).
- Pixel Depth: 24 bits (8bits x 3 channels).
- Total Colors: $2^{24} = 16{,}777{,}216$

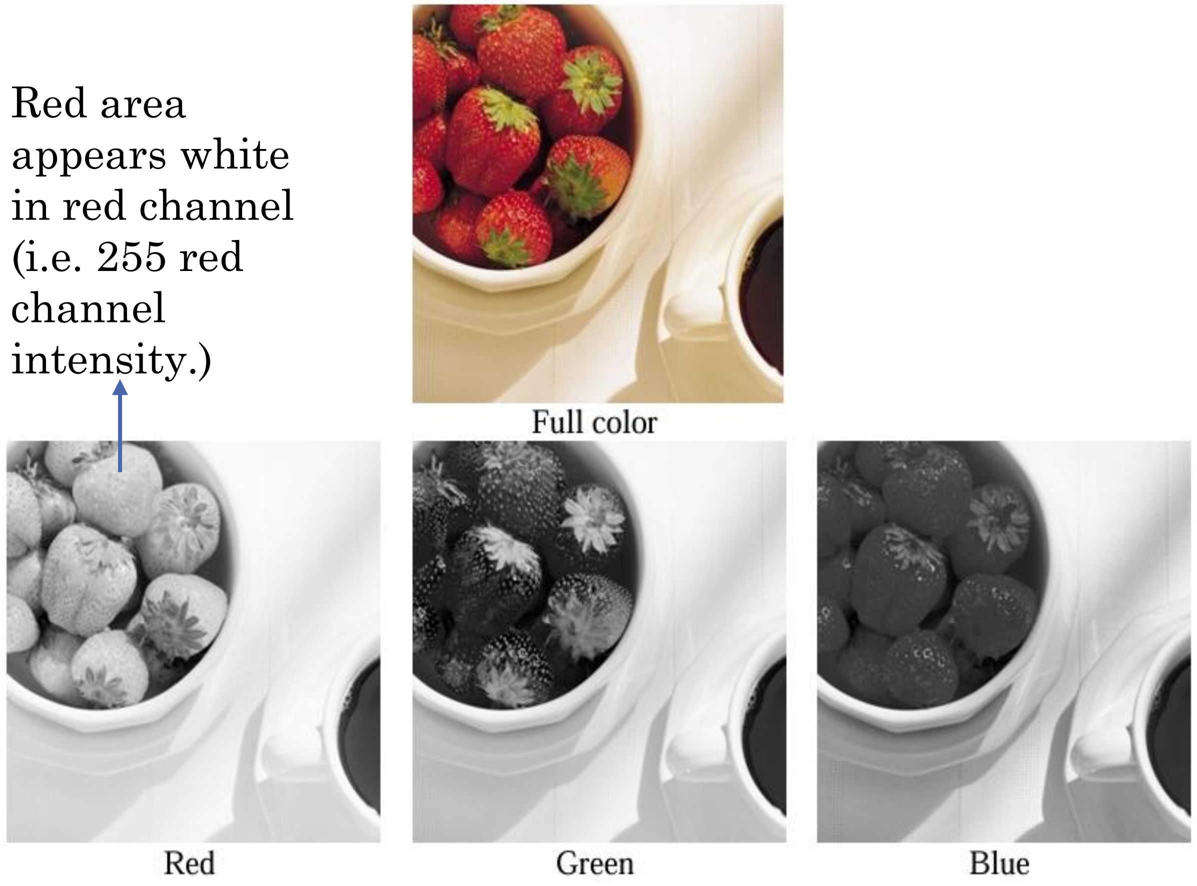


A 24-bit RGB Color Cube

Color Models

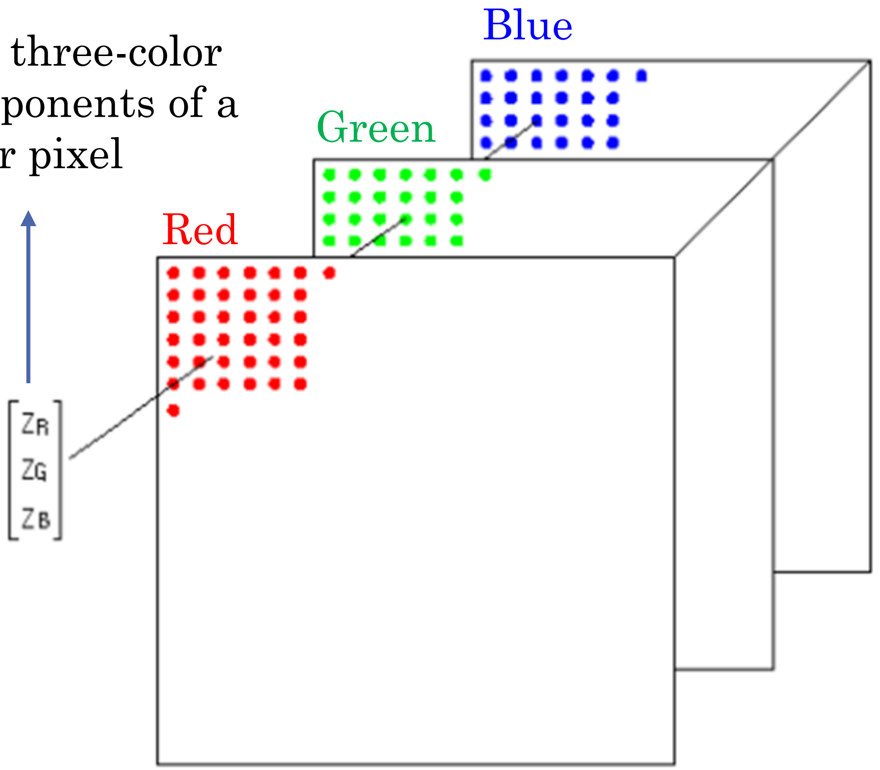
RGB Color Model

Red area appears white in red channel (i.e. 255 red channel intensity.)



RGB Color Channels of a Full-Color Image

The three-color components of a color pixel



Scheme of RGB color image

Color Models

CMY and CMYK color Models

Secondary Colors of Light

- Cyan, magenta, and yellow are secondary colors and primary pigment colors.
- Printing devices use CMY data or convert RGB to CMY.

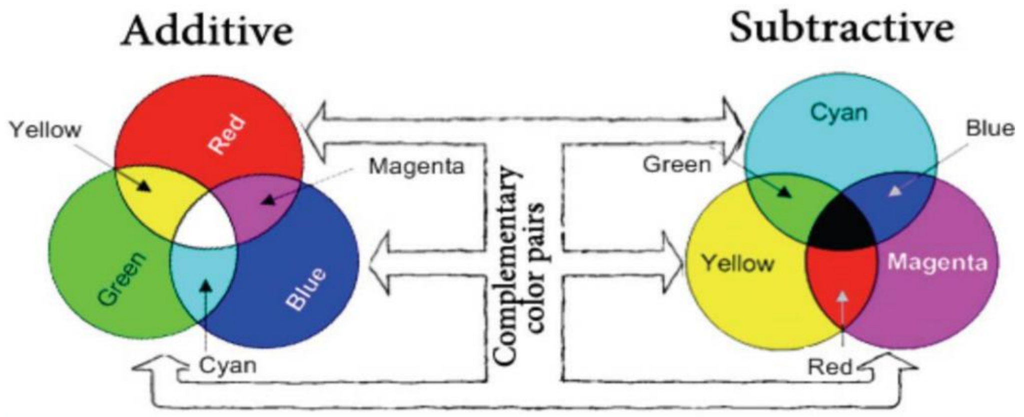
Limitations of CMY

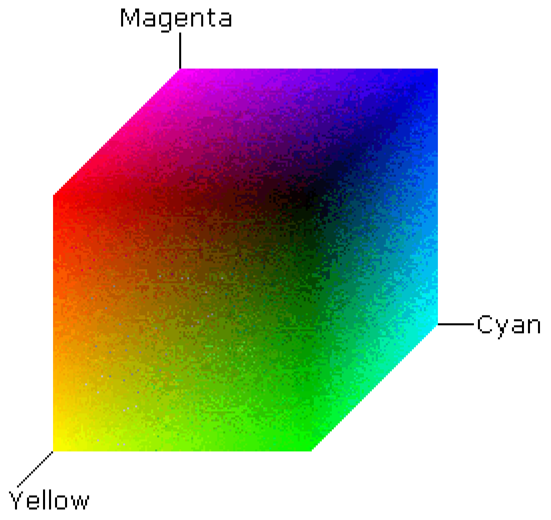
- Equal CMY amounts should create black but result in muddy brown due to pigment impurities.

CMYK Color Model

Why Black (K)?

- Added to produce true black in printing.
- Widely used in four-color printing: Cyan, Magenta, Yellow, Black.





Color Models

CMY and CMYK color Models

RGB to CMY Conversion

Assume all color values are normalized to [0,1]

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

CMY to CMYK Conversion

The conversion from CMY to CMYK begins by letting

K = min(C,M,Y)

If K=1, then we have pure black, with no color contributions, from which it follows that,

C = 0, M = 0, Y = 0.

Otherwise,

$$Y = (Y - K) / (1 - K) M = (M - K) / (1 - K) C = (C - K) / (1 - K)$$

RGB to CMYK conversion

$$\acute{R} = \frac{R}{255}$$

$$\acute{G} = \frac{G}{255}$$

$$\acute{B} = \frac{B}{255}$$

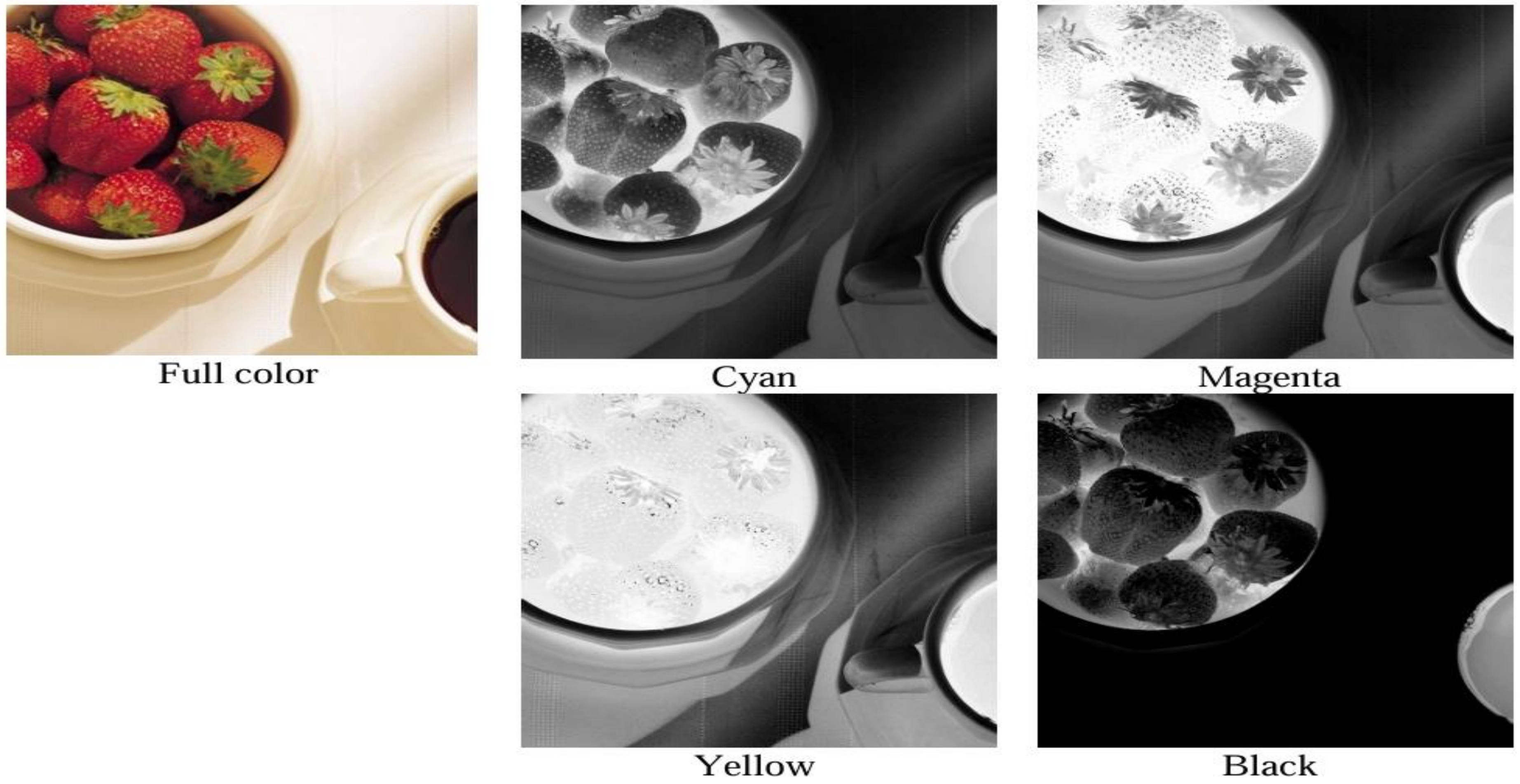
$$K = 1 - \max(\acute{R}, \acute{G}, \acute{B})$$

$$C = \frac{(1 - \acute{R} - K)}{1 - K}$$

Color Models

CMY and CMYK color Models

A full-color image and its CMYK component images



Color Models

The HSI Color Model

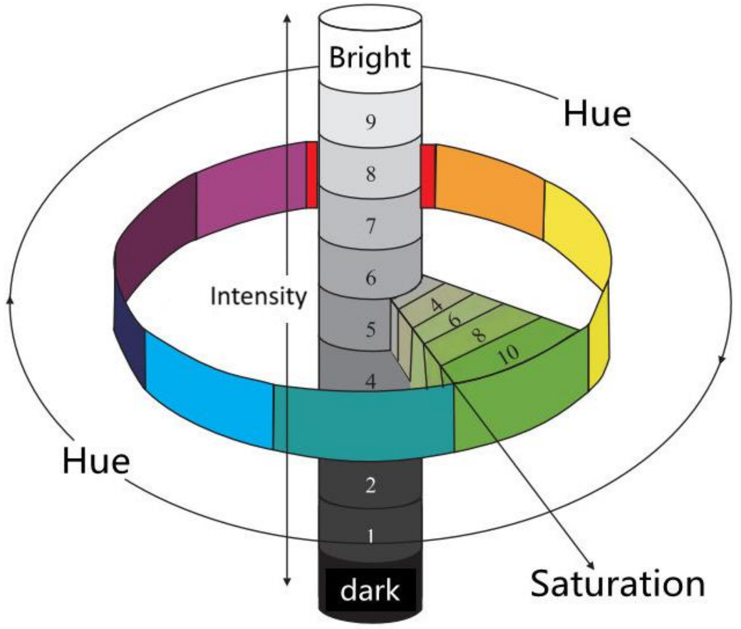
- Decouples intensity from color information for natural and intuitive color representation.

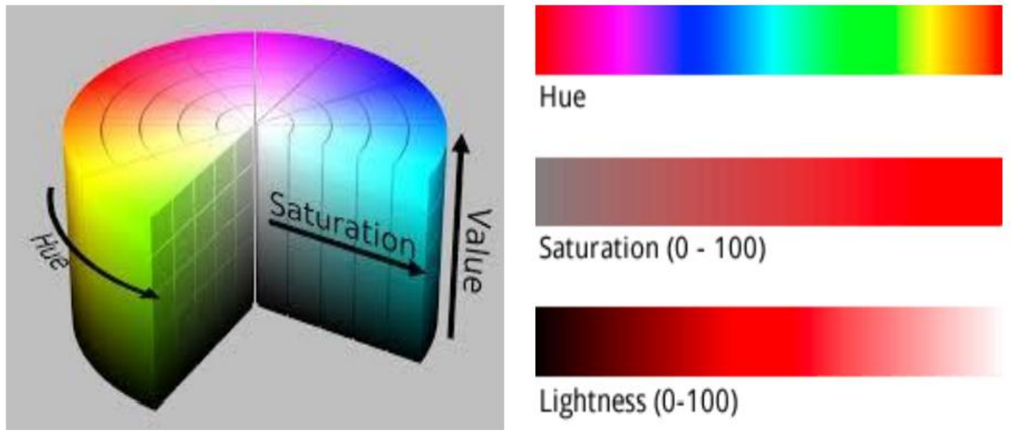
Components:

- **Hue (H)**: Describes the pure color (e.g., red, yellow, blue).
- **Saturation (S)**: Measures color purity (dilution with white light).
- **Intensity (I)**: Represents brightness (achromatic notion).

Why Use HSI?

- Matches human color perception (hue, saturation, brightness).
- Simplifies processing by separating intensity from color information.





Color Models

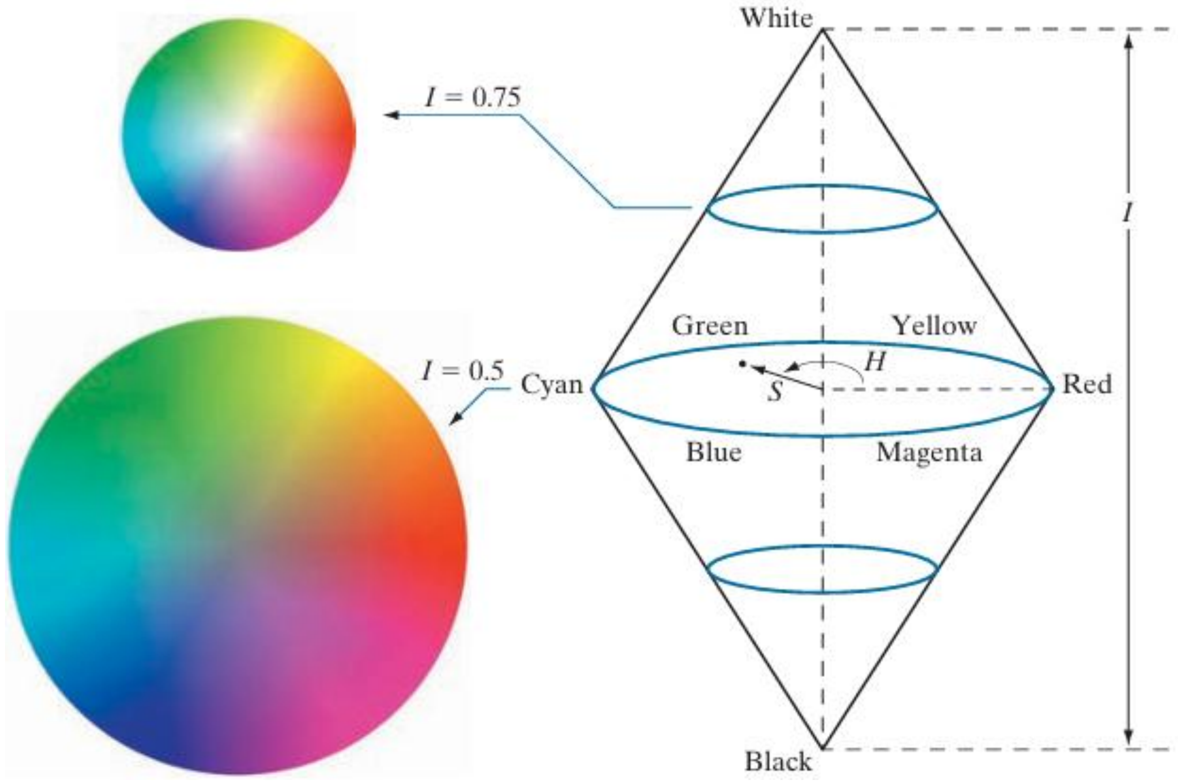
The HSI Color Model

Representation in HSI Space:

- **Hue**: Measured in degrees (0° = red, 120° = green, 240° = blue).
- **Saturation**: Length of vector from the origin to the color point.
- **Intensity**: Grayscale axis value.

Hexagonal Shape:

  - Primary colors separated by 120°.
  - Secondary colors 60° apart.
  - Other representations include triangles or circles.



Color Models

The HSI Color Model

Converting Colors from RGB to HSI

Given an image in RGB color format, the H component of each RGB pixel is obtained using the equation.

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

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

The saturation component is given by

$$S = 1 - \frac{3}{(R+G+B)}\left[\min(R,G,B)\right]$$

Finally, the intensity component is obtained from the equation

$$I = \frac{1}{3}(R+G+B)$$

Color Models

The HSI Color Model

Converting Colors from HSI to RGB

Given values of HSI in the interval [1, 0] we now want to find the corresponding RGB values in the same range.

The applicable equations depend on the values of H. There are three sectors of interest, corresponding to the 120° intervals in the separation of primaries

RG sector ($0^0 \leq H < 120^0$): When H is in this sector, the RGB components are given by the equations,

$$B = I(1-S) \qquad R = I\left[1 + \frac{S\cos H}{\cos(60^\circ - H)}\right] \qquad G = 3I - (R+B)$$

GB sector ($120^0 \leq H < 240^0$): If the given value of H is in this sector, we first sub tract 120° from it.

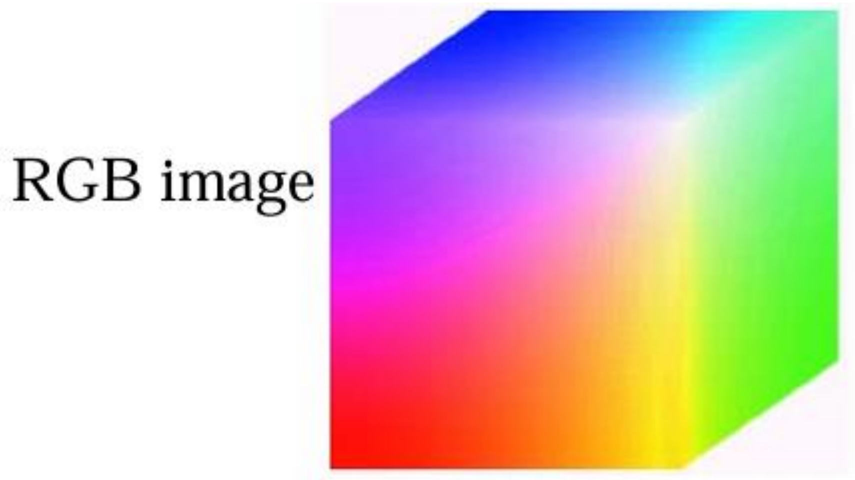
$$H = H - 120^\circ \qquad R = I(1-S) \quad G = I\left[1 + \frac{S\cos H}{\cos(60^\circ - H)}\right] \quad B = 3I - (R+G)$$

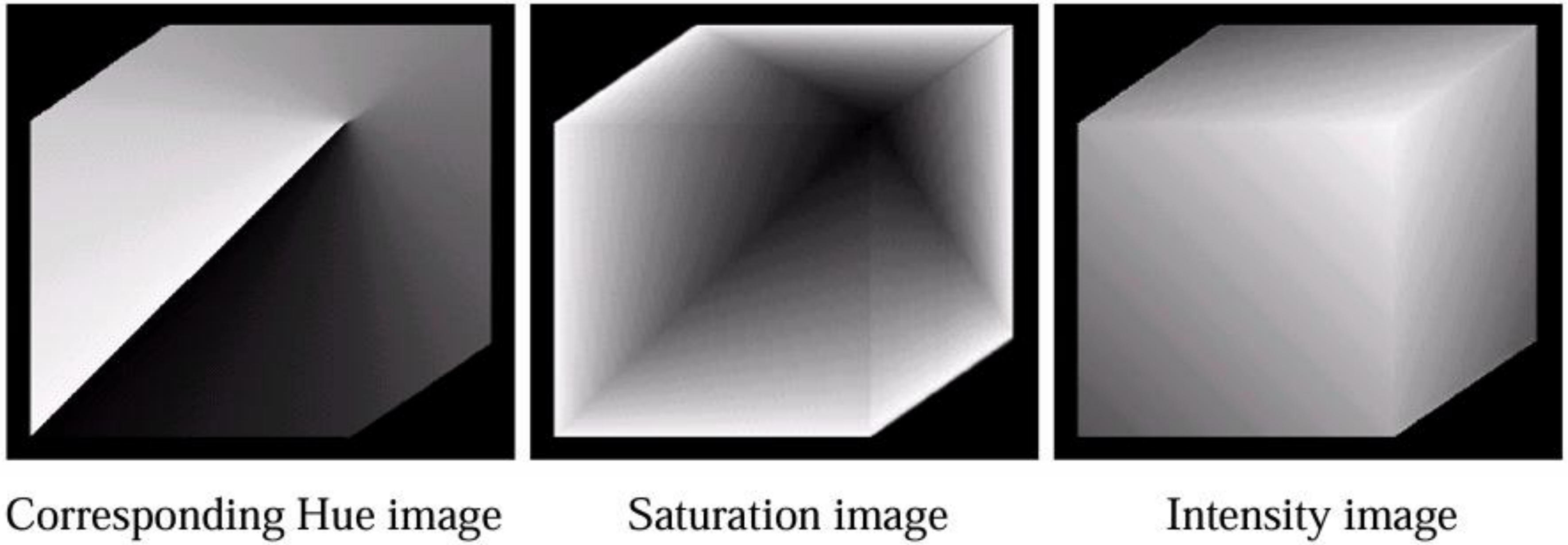
BR sector ($240^0 \leq H \leq 360^0$): Finally, if H is in this range, we subtract 240° from it:

$$H = H - 240^\circ \qquad G = I(1-S) \qquad B = I\left[1 + \frac{S\cos H}{\cos(60^\circ - H)}\right] \quad R = 3I - (G+B)$$

Color Models

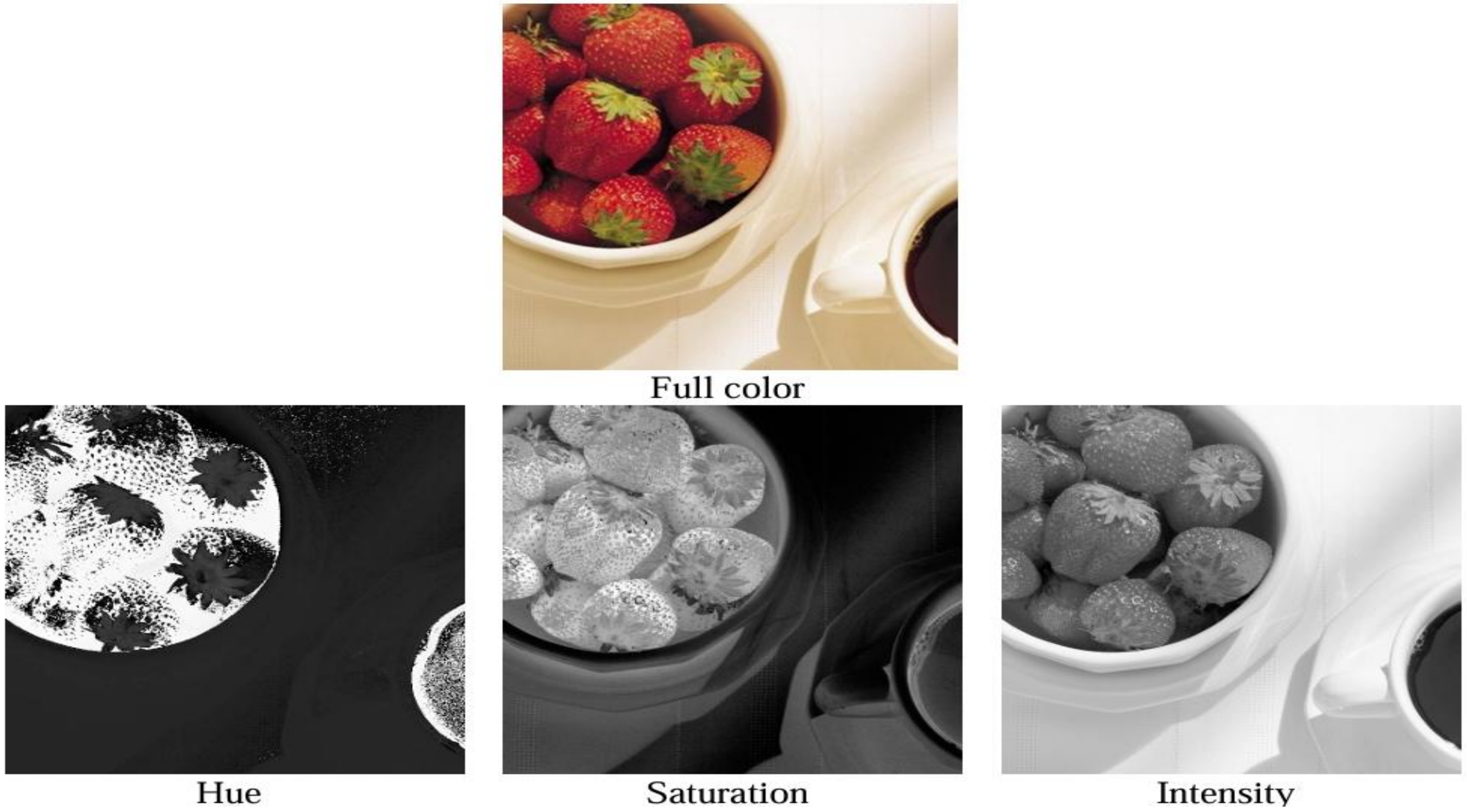
Conversion between RGB and HIS models





Color Models

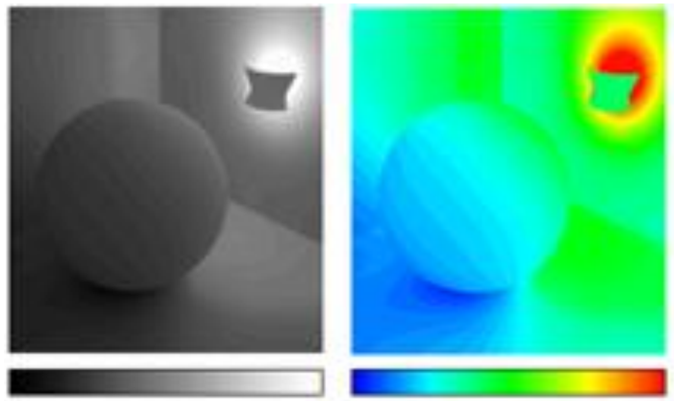
Conversion between RGB and HIS models

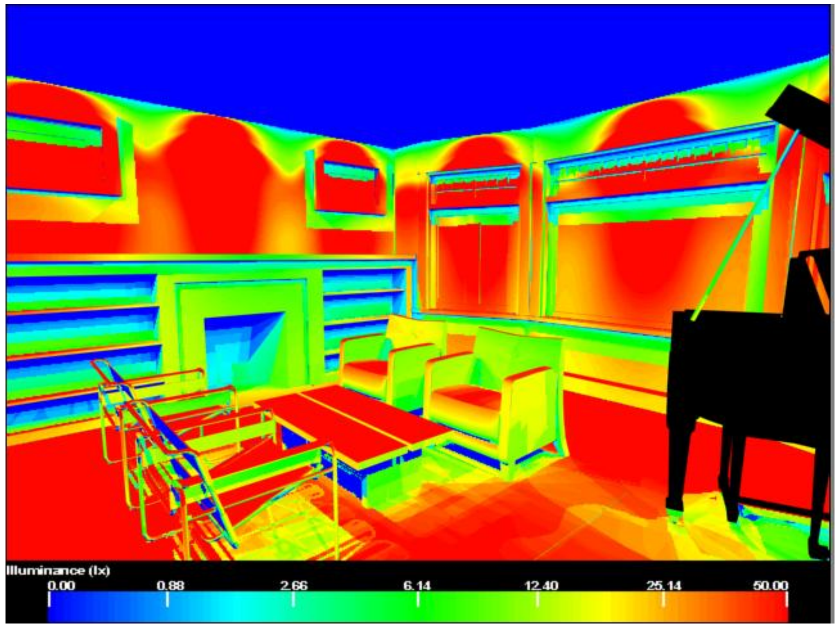


A full-color image and its HSI component images

Pseudo-Coloring: Making the Invisible Visible

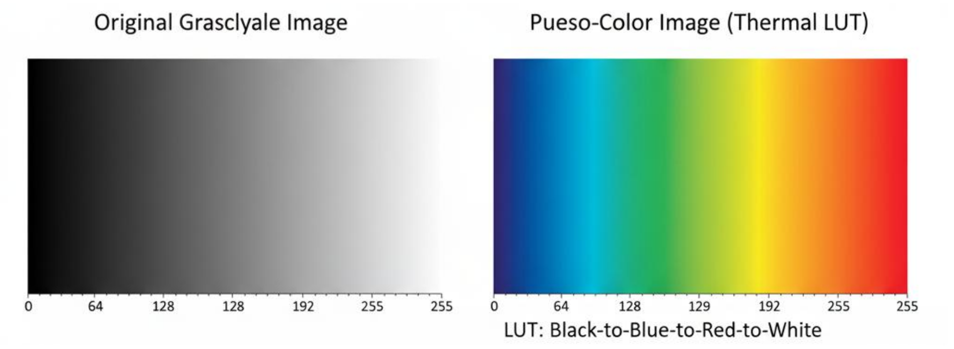
- **The Limit of Gray:** The human eye can reliably distinguish only about **2 dozen gray levels**. Subtle detail is easily lost in standard grayscale images.
- **The Power of Color:** The human eye can distinguish **thousands of color values**—it's far more sensitive to variation in hue than shade.
- How **it Works (LUT):** Each of the 256 gray levels in the image is assigned a unique color via a **Lookup Table (LUT)**.

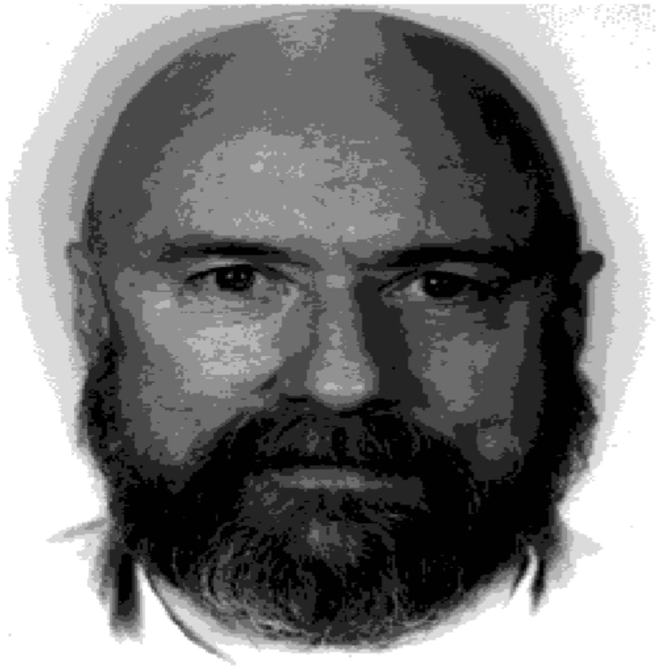


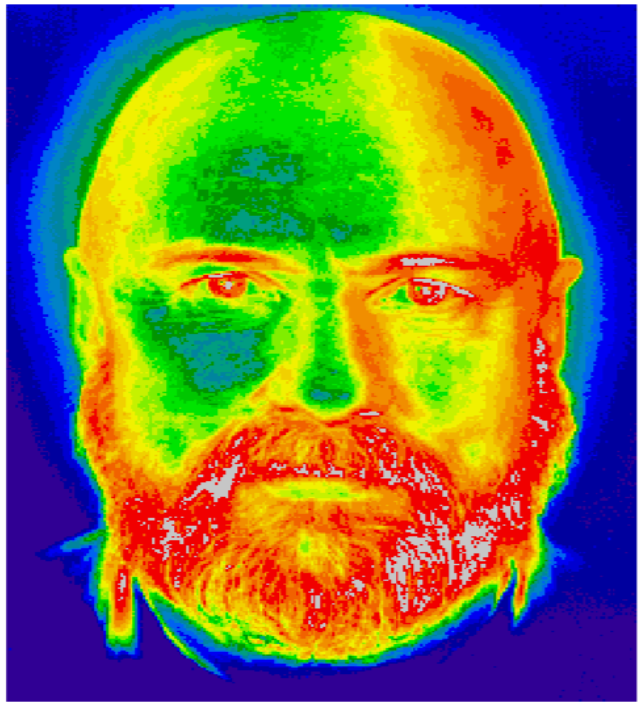


Pseudo-Coloring Using LUT

- CLUT(Color lookup table): A mapping of a pixel value to a color value shown on a display device.
- For example, in a grayscale image with levels 0, 1, 2, 3, and 4,pseudo-coloring is a color lookup table that maps 0 to black, 1to red, 2 to green, 3 to blue, and 4 to white.



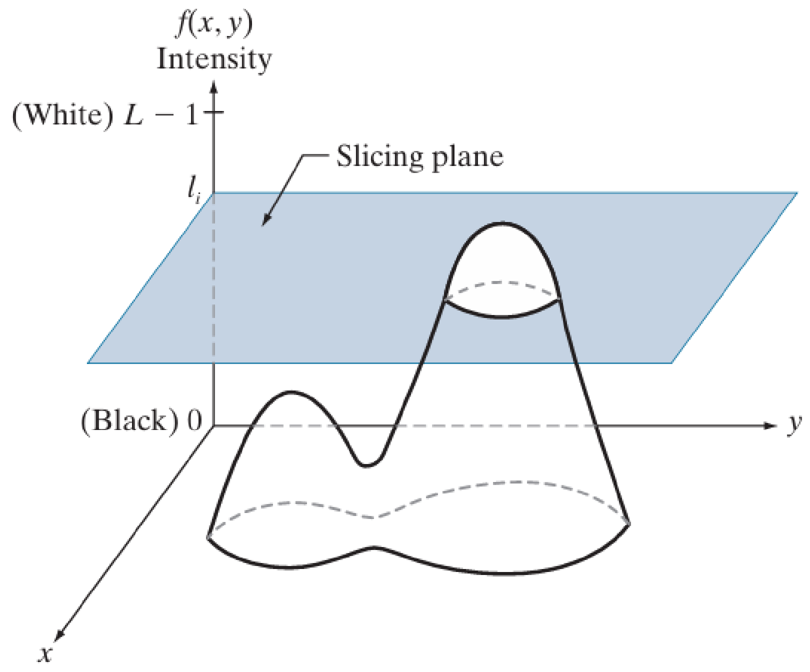




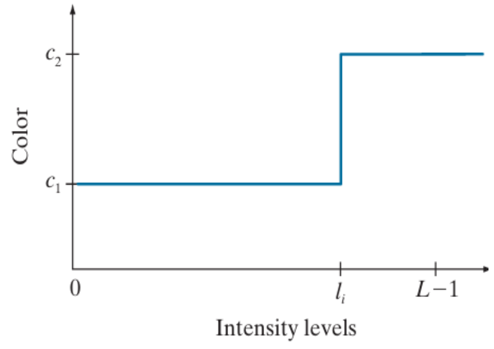
Intensity Slicing: Pseudo-Coloring by Intervals

- Intensity Slicing (also known as Density Slicing or Color Coding) is one of the simplest techniques for converting a grayscale image into color.
- **The Concept**
  - **Goal:** To highlight specific ranges of grayscale intensity by assigning them a distinct color.
  - **The Problem:** The original grayscale range [0, L-1] (Black to White) is visually difficult to interpret.
- **How It Works (The Partition)**
  - **Define Slices:** You define **P** planes (or thresholds) that are perpendicular to the intensity axis at levels $l_1, \ l_2, \dots, l_P$.
  - **Create Intervals:** These **P** planes divide the entire grayscale range into **P+1** distinct intensity intervals ($V_1, V_2, \dots, V_{p+1}$).
  - **Assign Color:** Every pixel whose intensity value $f(x, y)$ falls within a specific interval V_k is assigned a single, unique, constant color C_k.

Analogy: It's like taking a cake (the grayscale) and slicing it into layers, then painting each layer a solid, different color.

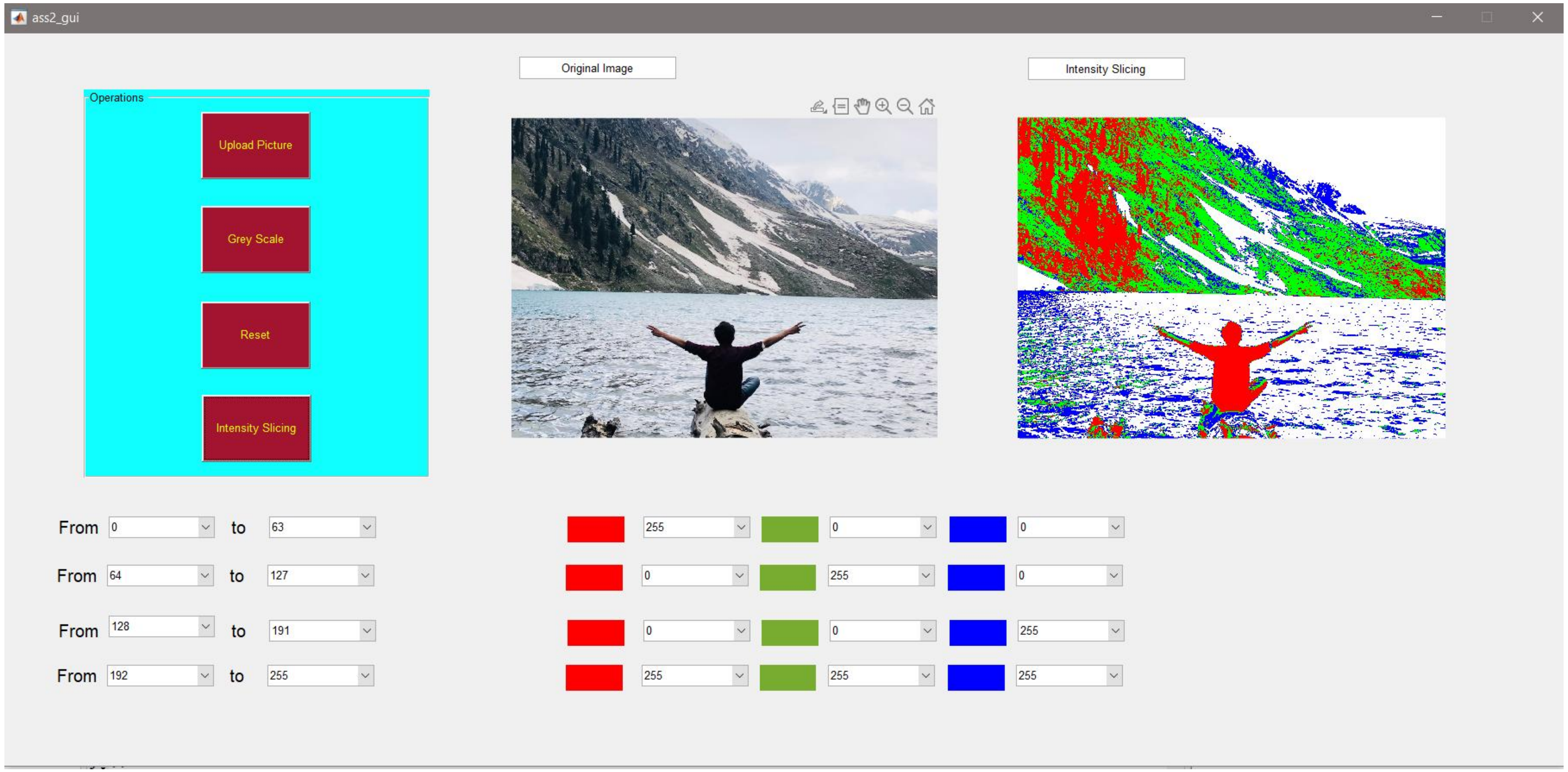


Graphical interpretation of the intensity- slicing technique.



An alternative representation of the intensity-slicing technique

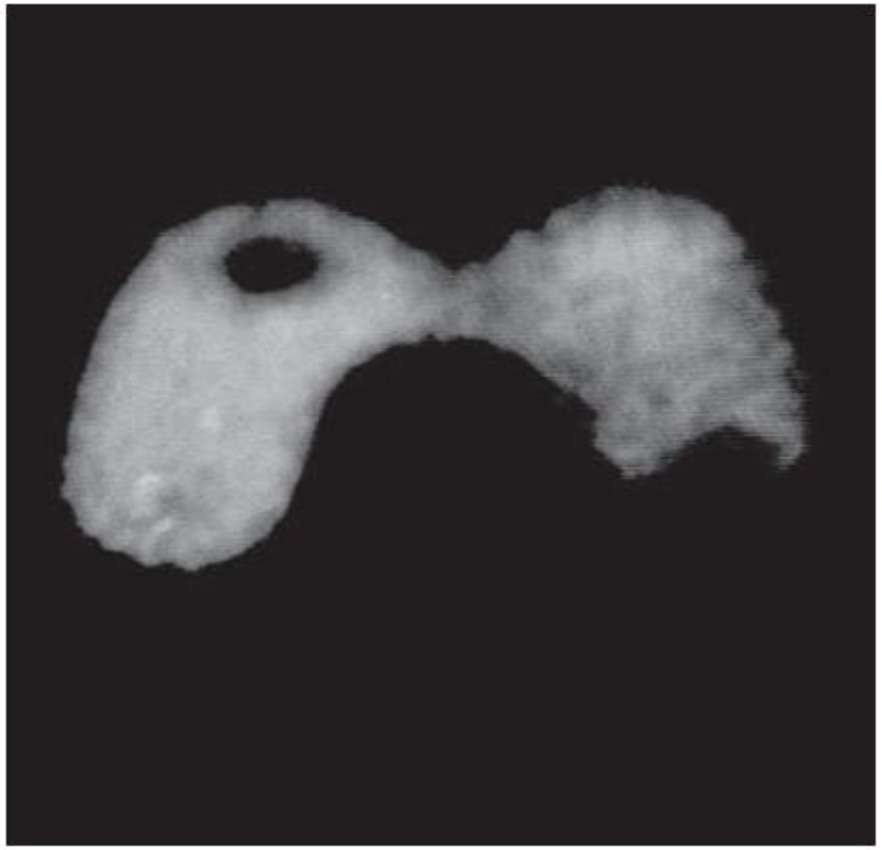
Intensity Slicing: Example

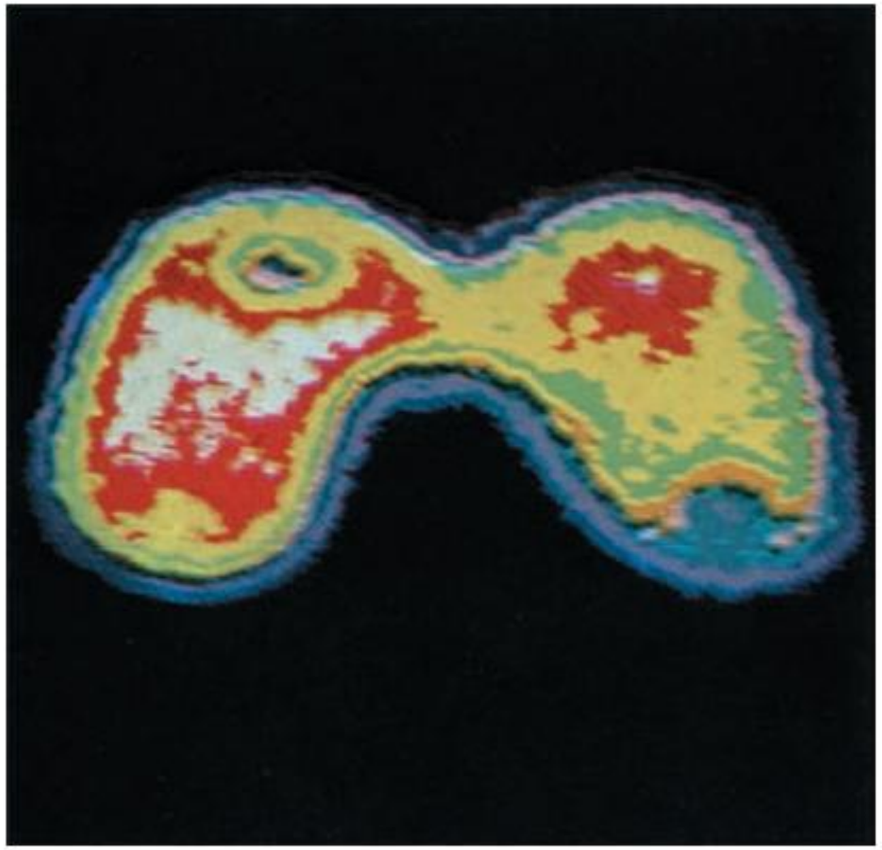


From	0	to	63	🟥	255	🟩	0	🟦	0
From	64	to	127	🟥	0	🟩	255	🟦	0
From	128	to	191	🟥	0	🟩	0	🟦	255
From	192	to	255	🟥	255	🟩	255	🟦	255

Pseudo-Color Image Processing

Intensity Slicing:

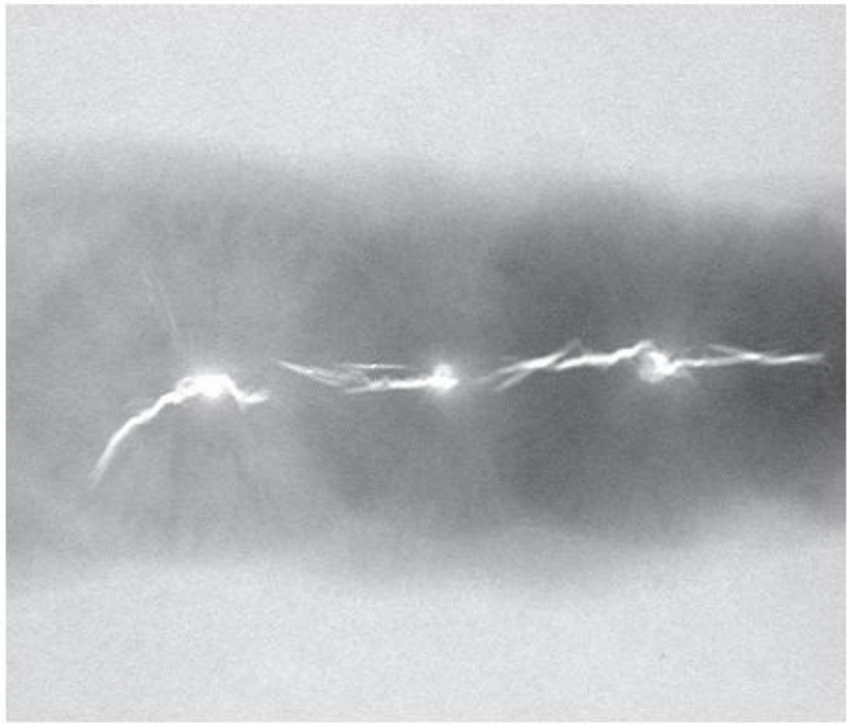

Grayscale image of the Picker Thyroid Phantom.

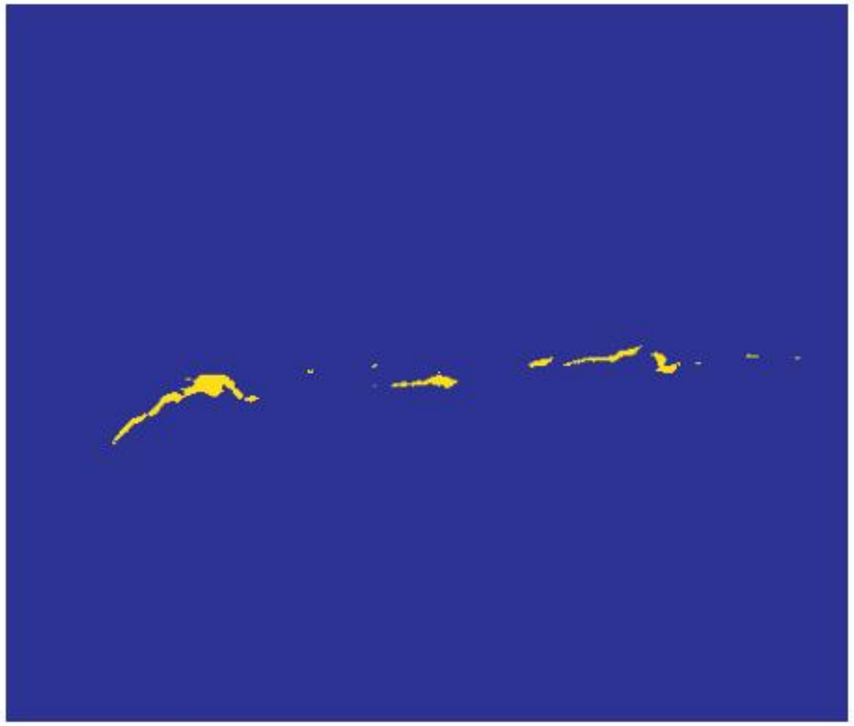

Result of intensity slicing using eight colors.

Pseudo-Color Image Processing

Example: Color Coding in Weld X-rays

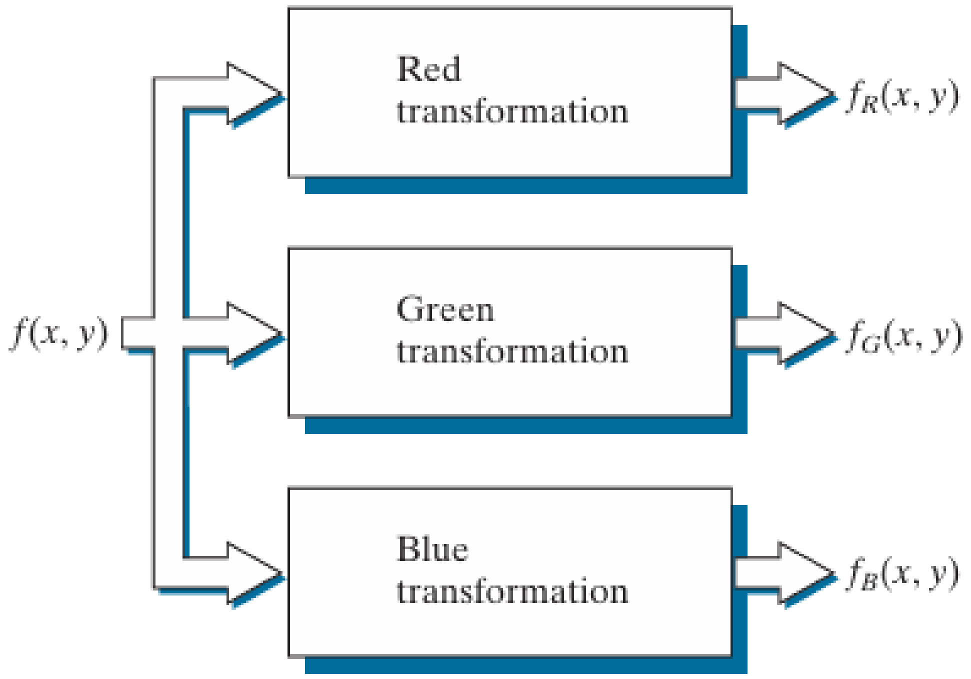
Problem: Cracks saturate sensors, producing an intensity value of 255 in an 8-bit image, indicating defects.


X-ray of a weld shows cracks and porosities as bright streaks.

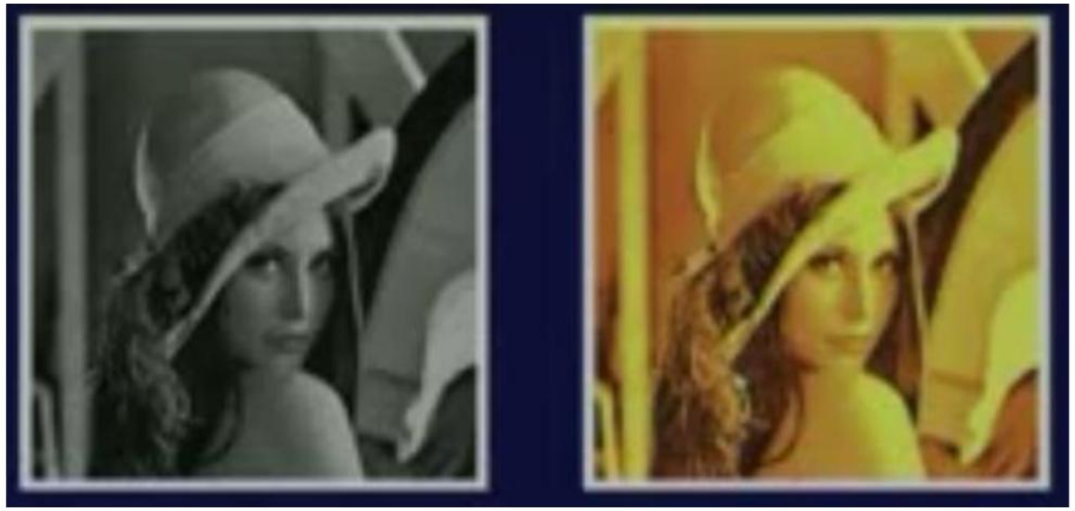

Color coding highlights defects at level 255 for better visibility.

Gray to Color Conversion

- Gray-to-Color Conversion addresses the limitation of the human eye, which can only distinguish ≈ 24 shades of gray, by exploiting its ability to see thousands of colors.
- The working mechanism of Gray to Color conversion showed in the figure below.

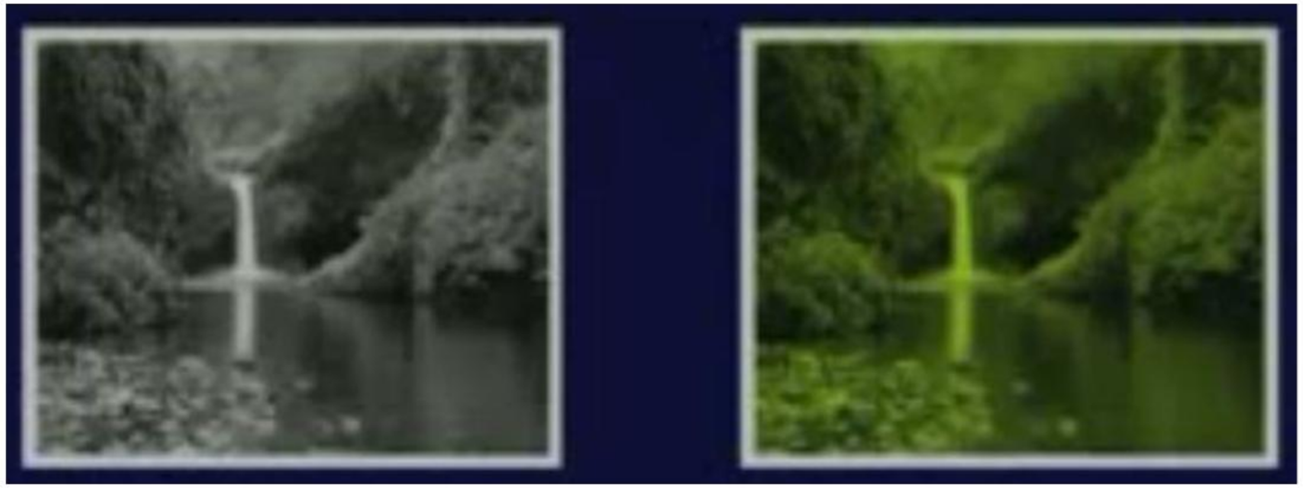


Gray to Color Conversion



$$f_R(x, y) = f(x, y)$$
$$f_G(x, y) = 0.33 f(x, y)$$
$$f_B(x, y) = 0.11 f(x, y)$$

Gray to Color Conversion



$$f_R(x,y) = 0.33f(x,y)$$
$$f_G(x,y) = f(x,y)$$
$$f_B(x,y) = 0.11f(x,y)$$

Basics of Full-Color Image Processing

- **Full color image processing fall into 2 categories.**
  - ❑In 1st category we process each component image individually and then form a composite processed color image from the individually processed component.
  - ❑In 2nd category we work with color pixels directly. Because full color images have at least three components, color pixels are really vectors.
  - ❑Let c represent an arbitrary vector in RGB color space:

$$C = \begin{bmatrix} C_r \\ C_g \\ C_b \end{bmatrix} = \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Color components are the function of co-ordinates (x,y) so we can write it as:

$$C(x,y) = \begin{bmatrix} C_r\,(x,y) \\ C_g(x,y) \\ C_b(x,y) \end{bmatrix} = \begin{bmatrix} R\,(x,y) \\ G\,(x,y) \\ B\,(x,y) \end{bmatrix}$$

For an image of size MxN there are MN such vectors, c(x,y), for x=0,1,2,...,M-1; y=0,1,2,...,N-1

Color Transformations

- Color transformation can be represented by the expression

$$g(x, y) = T[f(x, y)]$$

$f(x, y)$	input image
$g(x, y)$	processed image
T[*]	Transformation Function

- The pixel values here are triplets or quartets (i.e group of 3or 4 values)

$$s_i = T_i(r_1, r_2, \dots, r_n) \qquad i = 1,2,3, \dots, n$$

  - ❑r_i and S_i are variables denoting the color components of f(x,y) and g(x,y) at any point (x,y).
  - ❑n is the no of color components
  - ❑$\{T_1, T_2, \dots, T_n\}$ is a set of transformation or color mapping functions.

- Note that **n** transformations combine to produce a single transformation T.

Color Transformations

- The color space chosen determine the value of n.
- If RGB color space is selected then n=3 & r_1, r_2, r_3 denotes the red, blue and green components of the image.
- If CMYK color space is selected then n=4 & r_1, r_2, r_3, r_4 denotes the cyan, hue, magenta and black components of the image.
- Suppose we want to modify the intensity of the given image using **g(x,y)=k*f(x,y)** where 0<k<1.

Original Image

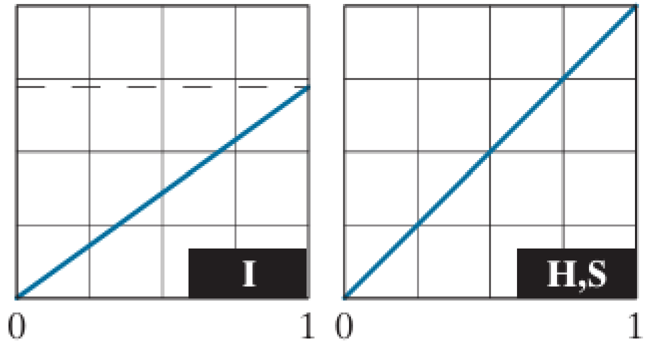
$$s_i = k(r_i) \quad i = 1,2,3$$

?

Processed Image

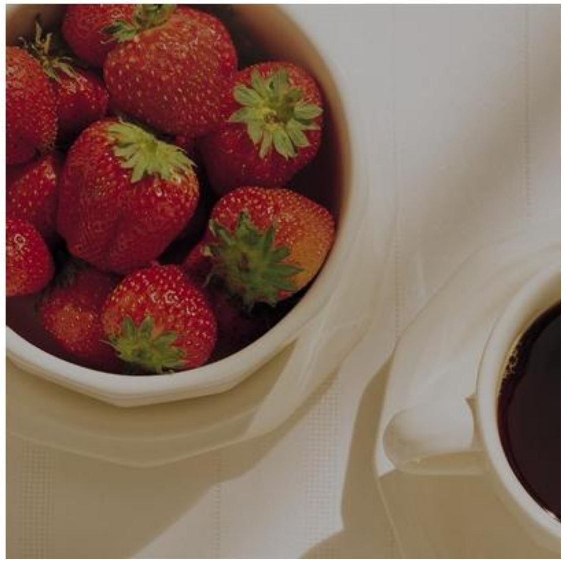
Color Transformations

- In HSI color space this can be done with the simple transformation.
  - ❑$s_3 = k * r_3$
  - ❑where $s_1 = r_1$ and $s_2 = r_2$
  - ❑Only intensity component r_3 is modified.



Original Image

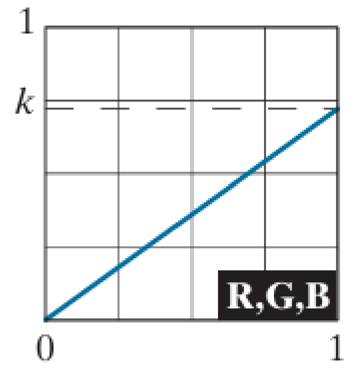
$s_3 = 0.7(r_3) \quad i = 1, k = 0.7$



Processed Image

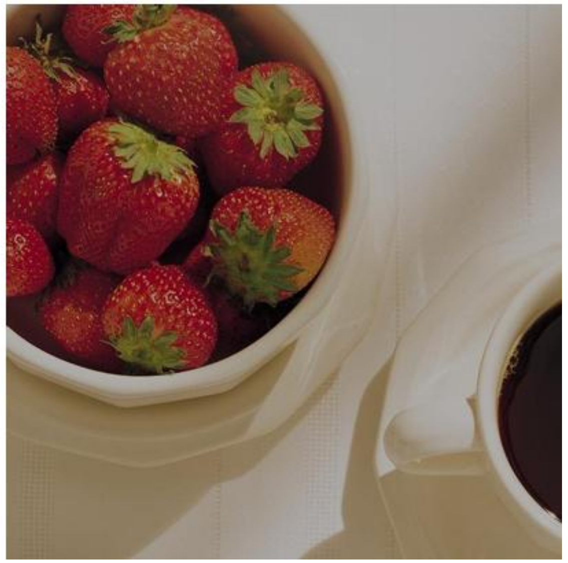
Color Transformations

- In RGB color space 3 components must be transformed:
- $s_i = k * r_i$



Original Image

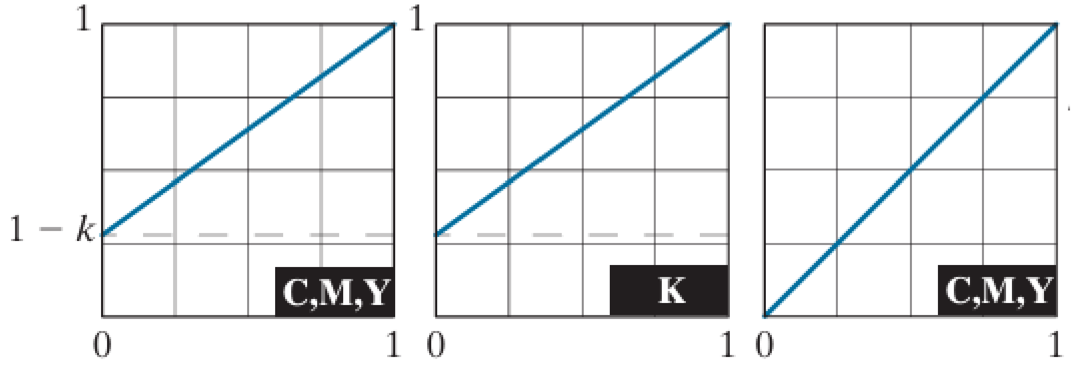
$s_i = 0.7(r_i) \quad i = 1,2,3$



Processed Image

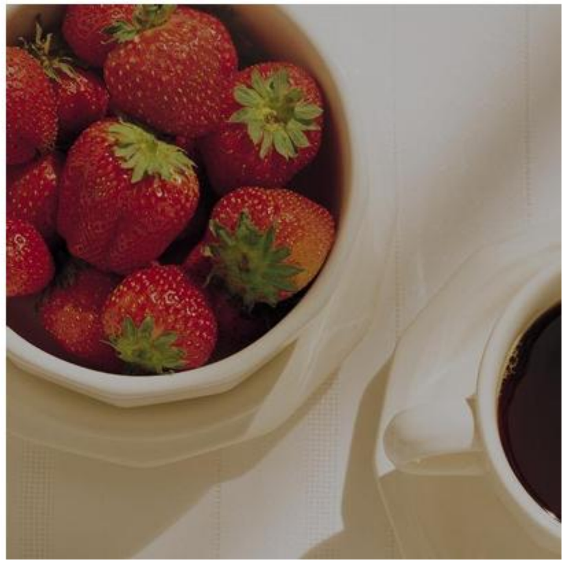
Color Transformations

- In CMY color space 3 components must be transformed:
- $s_i = k * r_i + (1 - k) \qquad i = 1,2,3$



Original Image

$$s_i = 0.7(r_i) + (1 - K)$$
$$i = 1,2,3$$

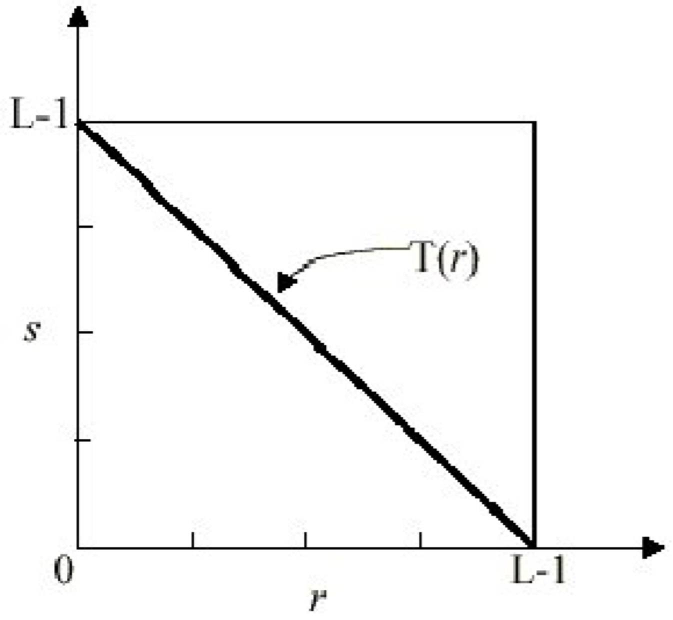

Processed Image

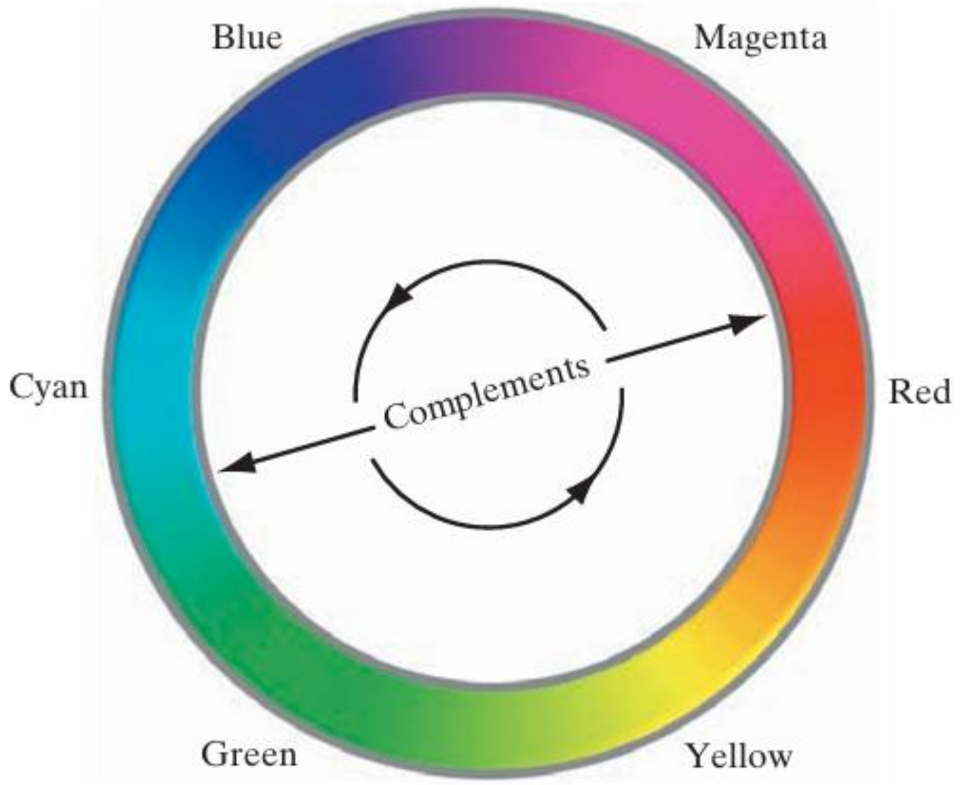
Color Complements

- The hues opposite to one another on the Color Circle arecalled Complements.
- Color Complement transformation is equivalent to image negative in Grayscale images

- $S = T(r) = L - 1 - r$ For Gray scale image
- $S_i = T(r_i) = L - 1 - r_i$ For Color image

Where $i = 1,2,3$





Color Complements

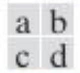
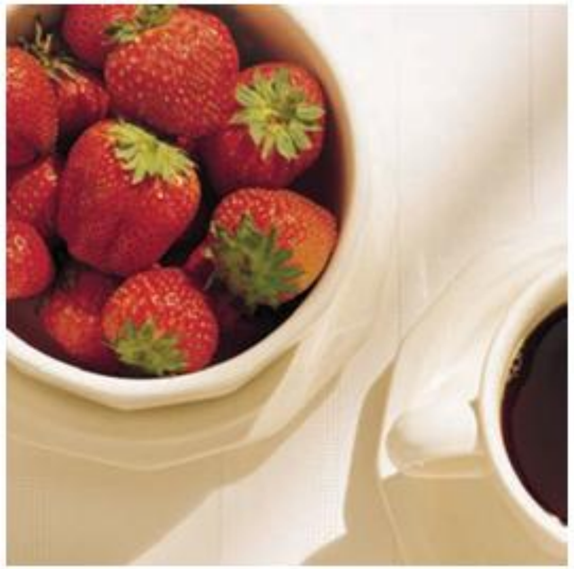
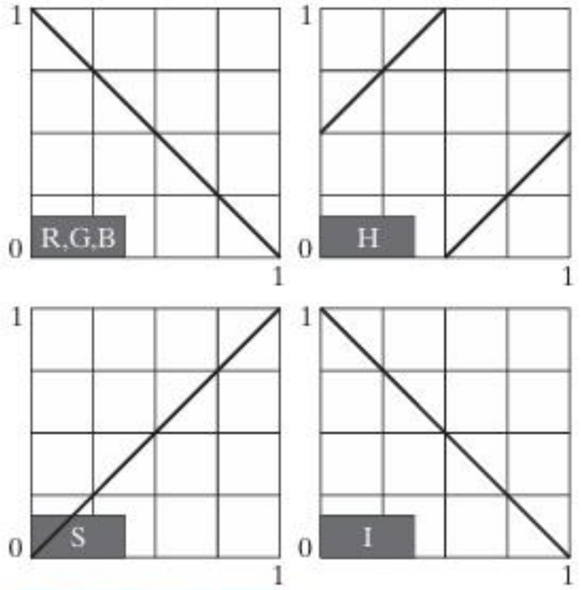
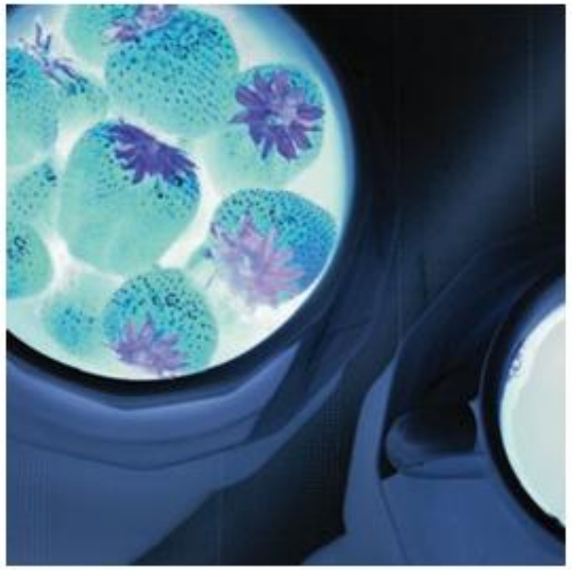
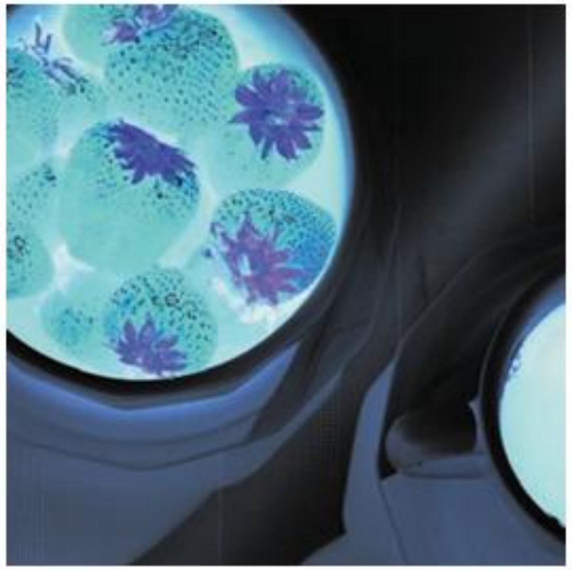


FIGURE 6.31 Color complement transformations. (a) Original image. (b) Complement transformation functions. (c) Complement of (a) based on the RGB mapping functions. (d) An approximation of the RGB complement using HSI transformations.









Color Slicing

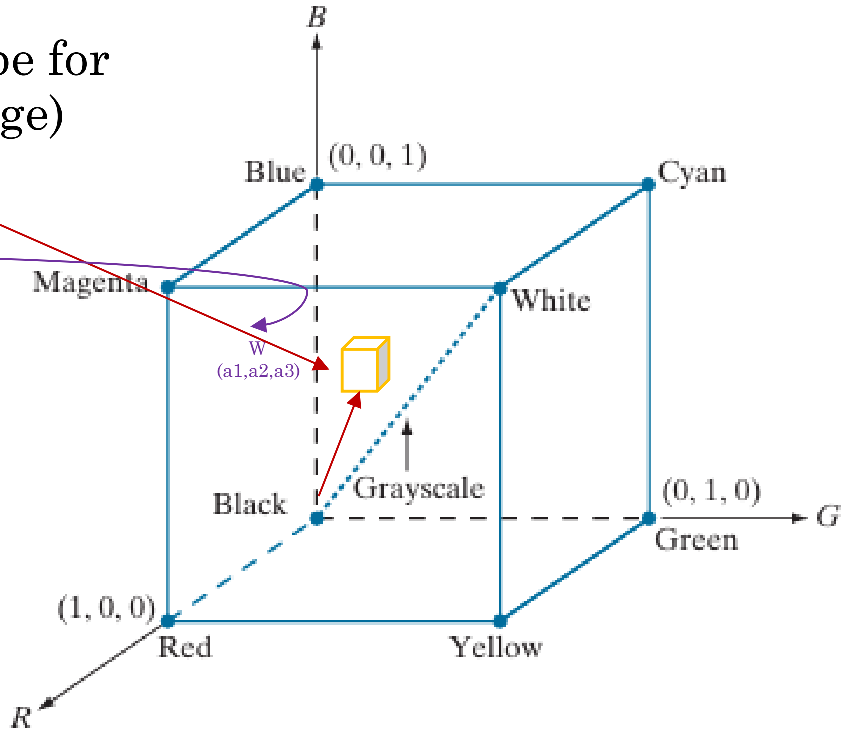
- Highlighting a specific range of colors in an image is useful for separating objects from their surroundings.
- Display the colors of interest so that they are distinguished from background.
- One way to slice a color image is to map the color outside some range of interest to a non prominent neutral color.
- If the colors of interest are enclosed by a cube (or hypercube for n>3) of width W and centered at a prototypical (e.g., average) color with components $(a_1, a_2, \dots, a_n)$, the necessary set of transformations are given by,

$$s_i = \begin{cases} 0.5 & \text{if } \left[\left| r_j - a_j \right| > \frac{W}{2} \right]_{\text{any } 1 \le j \le n} \\ r_i & \text{otherwise} \end{cases} \qquad i = 1, 2, \dots, n$$

- If a sphere is used to specify the colors of interest,

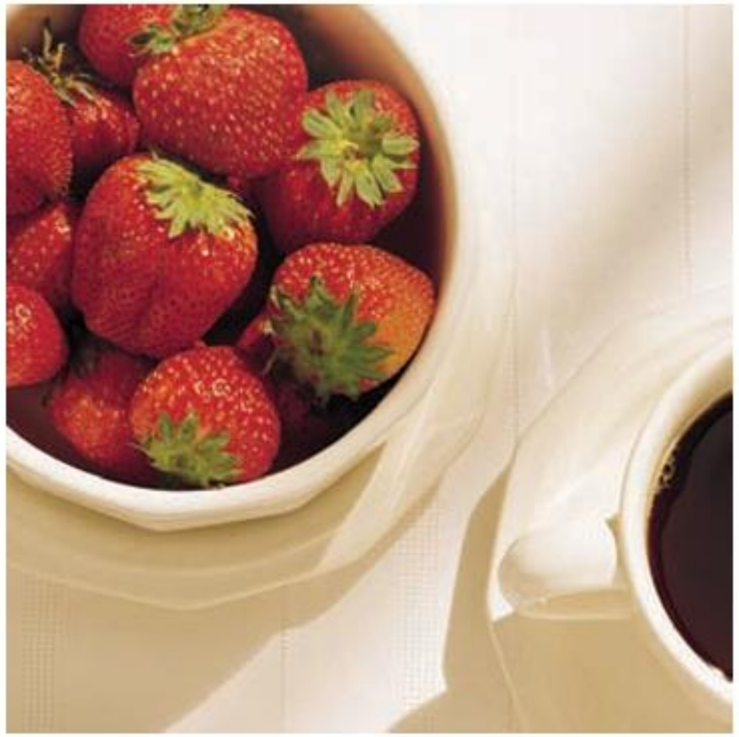
R_0 is the radius of the sphere being used.

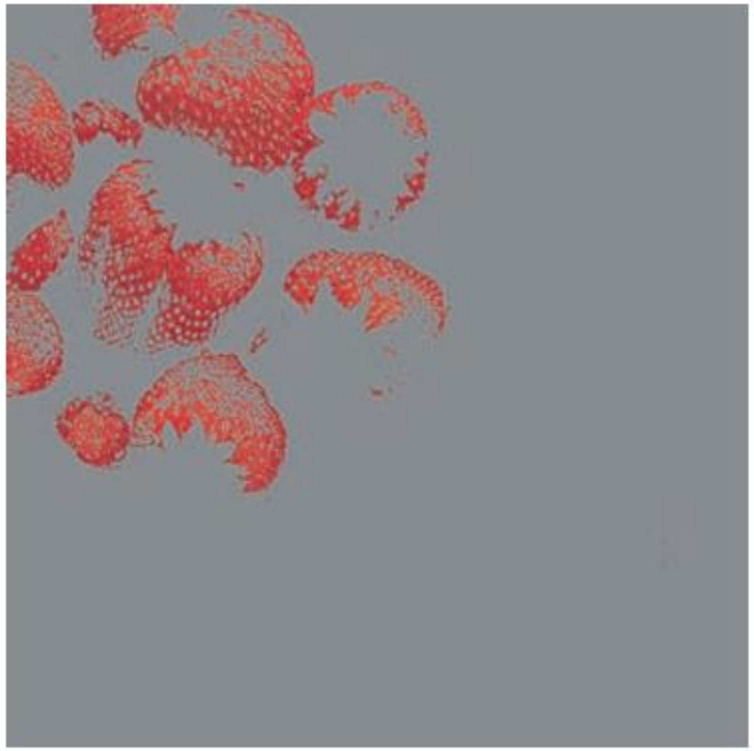
$$s_i = \begin{cases} 0.5 & \text{if } \sum_{j=1}^{n} \left(r_j - a_j \right)^2 > R_0^2 \\ r_i & \text{otherwise} \end{cases} \qquad i = 1, 2, \dots, n$$

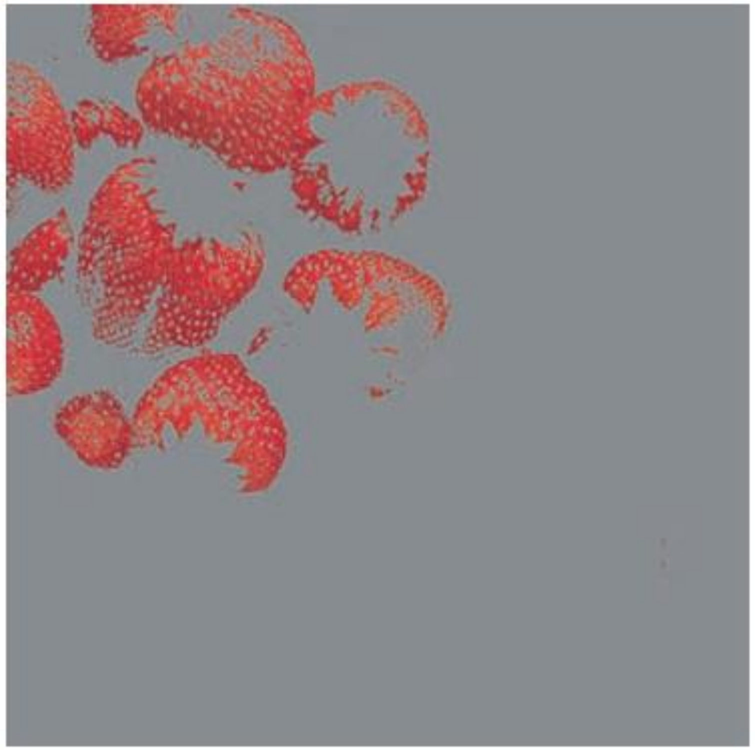


Color Slicing

Pixels outside the cube and sphere were replaced by color (0.5, 0.5, 0.5)


Original image

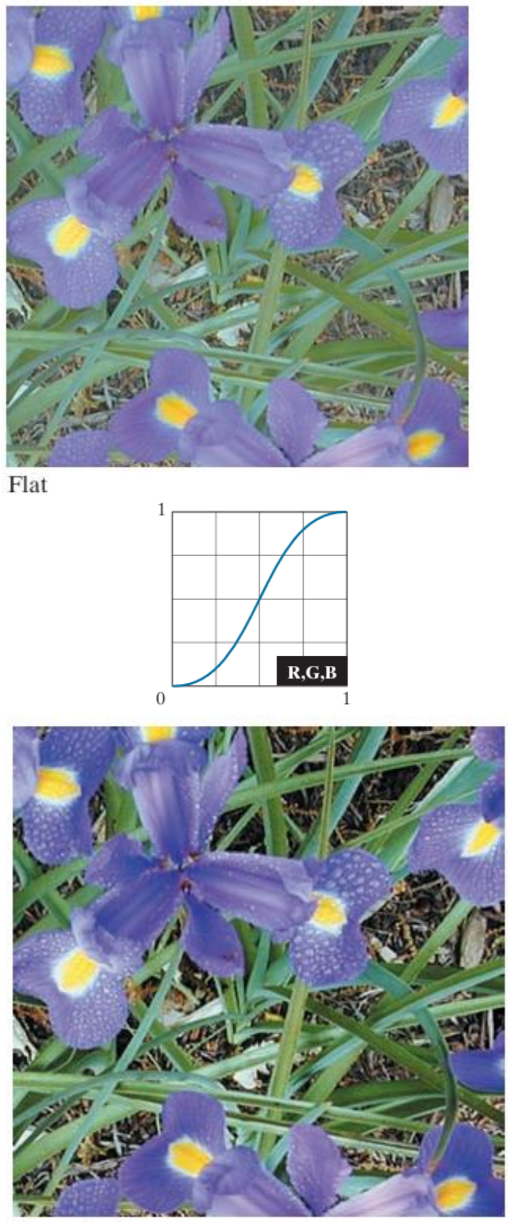

reds within an RGB cube of width W =02549 centered at (0.6863, 0.1608, 0.1922)


reds within an RGB sphere of radius 0.1765 centered at the same point.

Tone and Color Corrections

Tone Correction

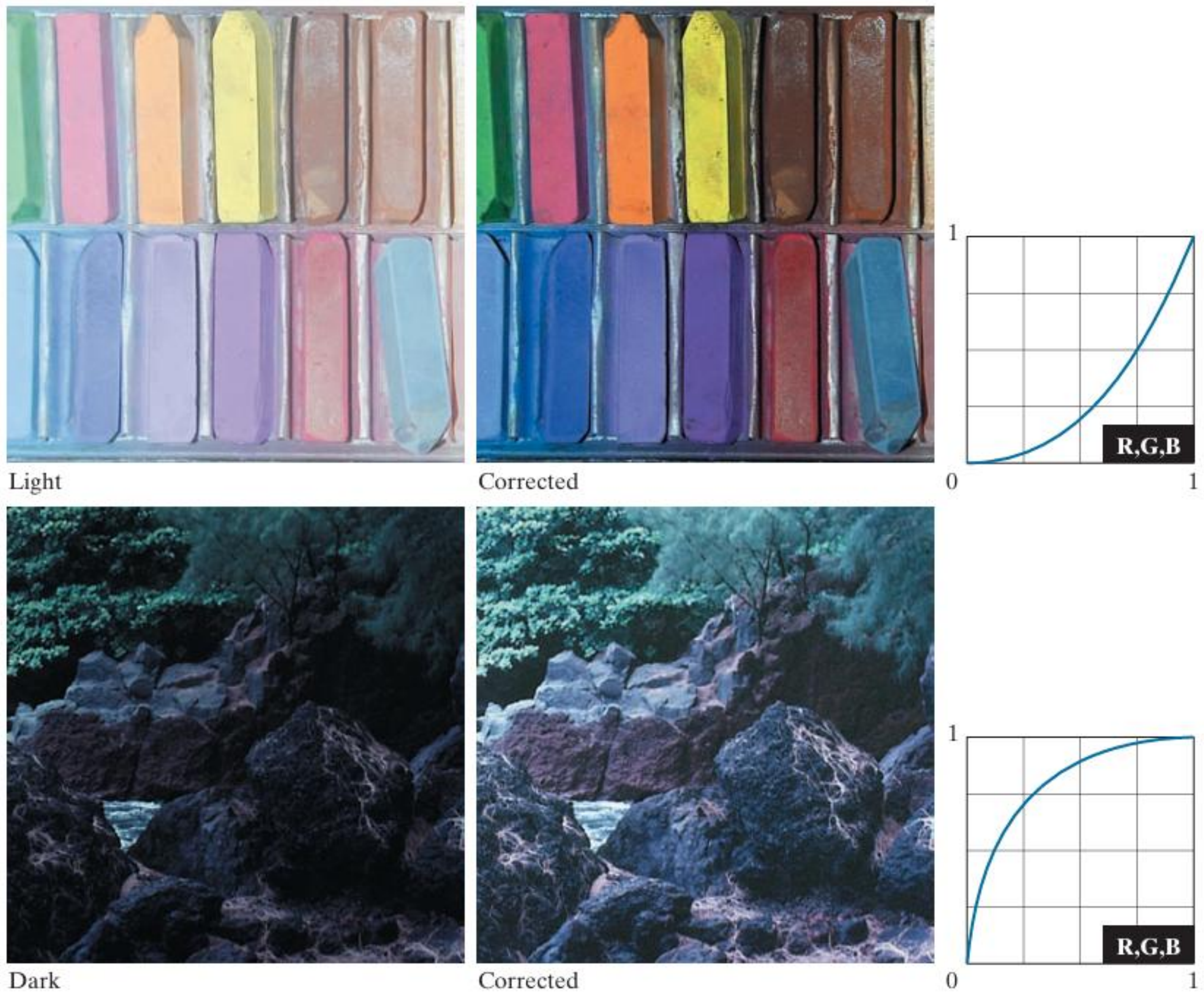
- Addressing an image's tonal distribution is the **first critical step** before fixing specific color irregularities (like oversaturation or undersaturation).
- **Tonal Range (Key Type):** Refers to the general distribution of color intensities across the image.
- **Goal:** To evenly distribute intensities between the highlights (light areas) and the shadows (dark areas), similar to grayscale image equalization.
- **Correction Goal**
- To achieve a balanced image, we use **color transformations** to shift the current intensity distribution (High- or Low-Key) closer to the desirable **Middle-Key** distribution. This corrects the fundamental tonal imbalance.



Corrected

Tone and Color Corrections

- **Tone Correction**

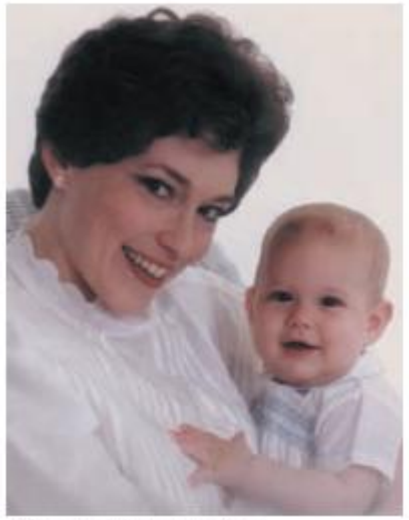


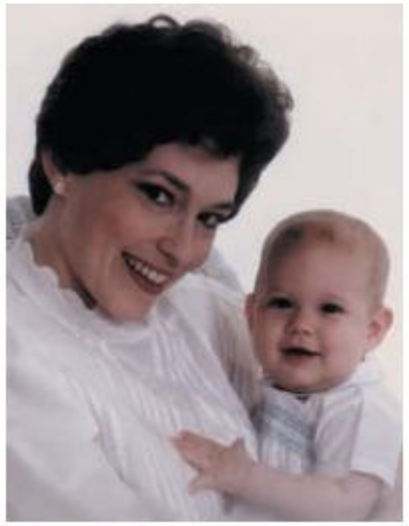
Tonal corrections for flat, light (high key), and dark (low key) color images. Adjusting the red, green, and blue components equally does not always alter the image hues significantly.

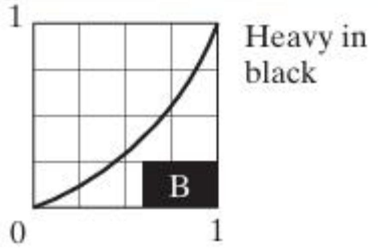
Tone and Color Corrections

Color Correction

- When a color imbalance is noted, there are a variety of ways to correct it.
- When adjusting the color components of an image it is important to realize that every action affects the overall color balance of the image (Perception of one color is affected by its surrounding colors)
- Based on the color wheel, the proportion of any color can be increased by decreasing the amount of the opposite color in the image.
- Similarly, it can increase by raising the proportion of two immediately adjacent colors or decreasing the percentage of the two colors adjacent to the complement.
- Suppose for example there is an abundance of magenta in an RGB image, it can be decreased by,
  - ❑Reducing both red and blue or
  - ❑Adding Green


Original/Corrected

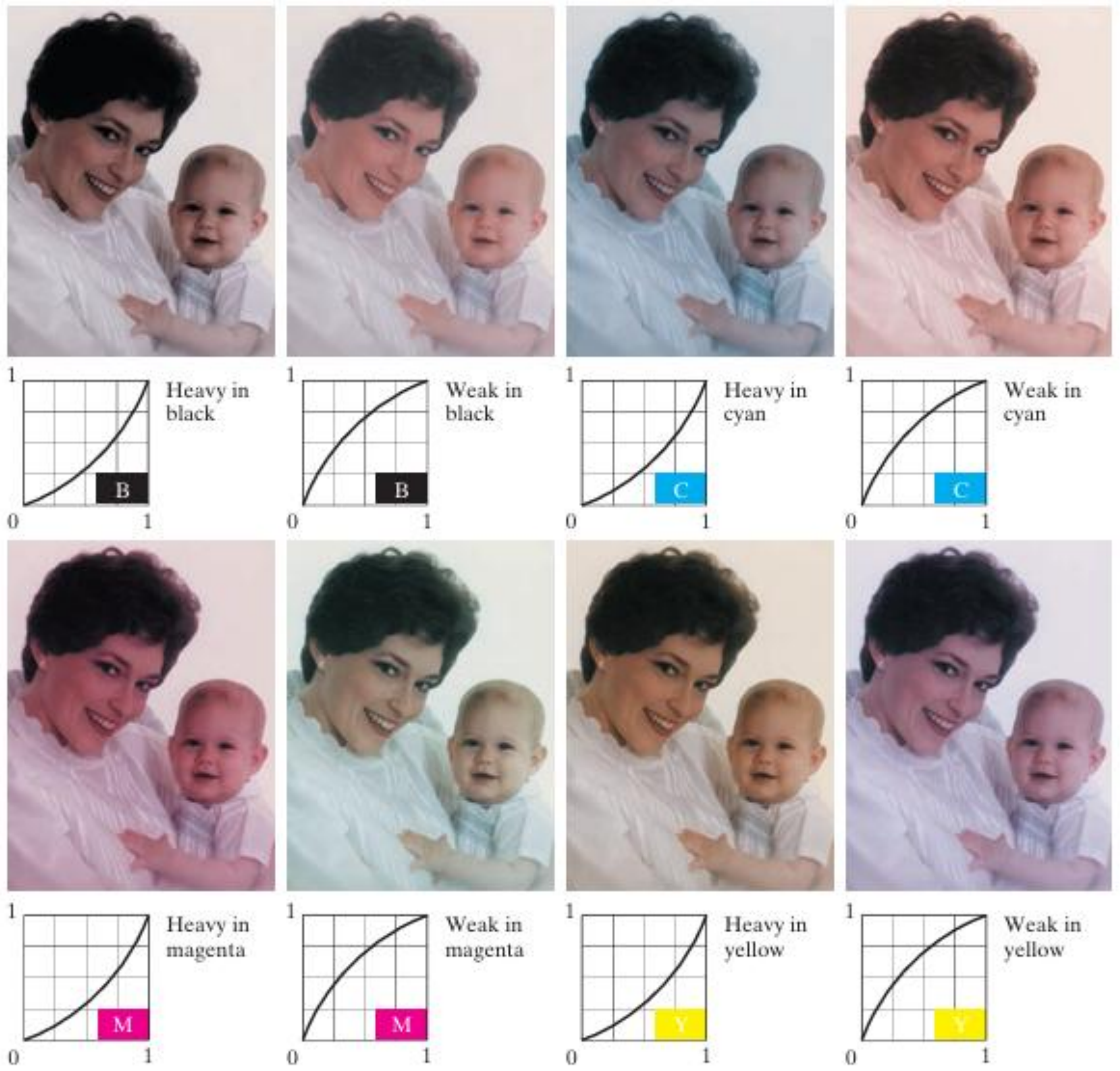




Tone and Color Corrections

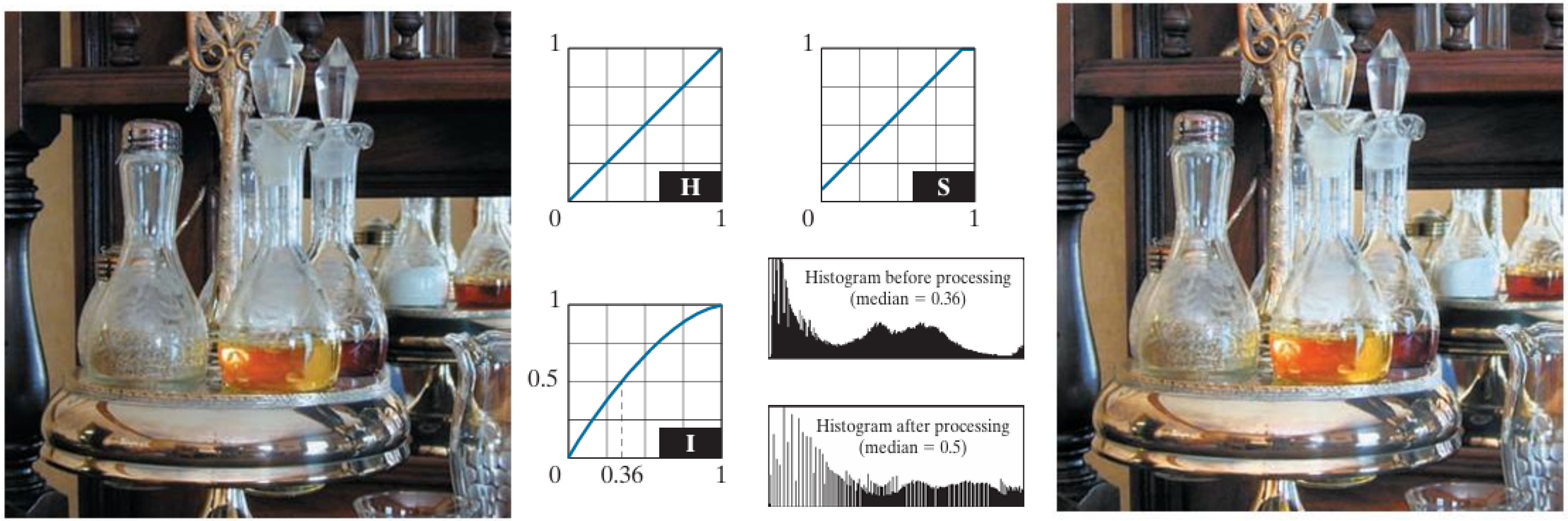
Color Correction

Original/Corrected



Histogram Processing

- Color images are composed of multiple components; however, it is not suitable to process each plane independently in case of histogram equalization. This results in erroneous color.
- A more logical approach is to spread the color intensities uniformly, leaving the colors themselves(hue, saturation) unchanged.
- HSI approach is ideally suited to this type of approach.



Color Image Smoothing

- Color images can be smoothed in the same way as grayscale images; the difference is that instead of scalar gray level values we must deal with component vectors of the following form:

$$C(x,y) = \begin{bmatrix} C_r(x,y) \\ C_g(x,y) \\ C_b(x,y) \end{bmatrix} = \begin{bmatrix} R(x,y) \\ G(x,y) \\ B(x,y) \end{bmatrix}$$

The average of the RGB component vector in this neighborhood is:

$$\bar{C}(x,y) = \frac{1}{K} \sum_{(x,y) \epsilon S_{xy}} C(x,y)$$

Also,

$$\bar{C}(x,y) = \begin{bmatrix} \frac{1}{K} \sum_{(x,y) \epsilon S_{xy}} R(x,y) \\ \frac{1}{K} \sum_{(x,y) \epsilon S_{xy}} G(x,y) \\ \frac{1}{K} \sum_{(x,y) \epsilon S_{xy}} B(x,y) \end{bmatrix}$$

We recognize the components of this vector as the scalar images that would be obtained by independently smoothing each plane of the starting RGB image using conventional gray scale neighborhood processing.

Thus, we conclude that smoothing by neighborhood averaging can be carried out on a per color plane basis.

Color Image Smoothing

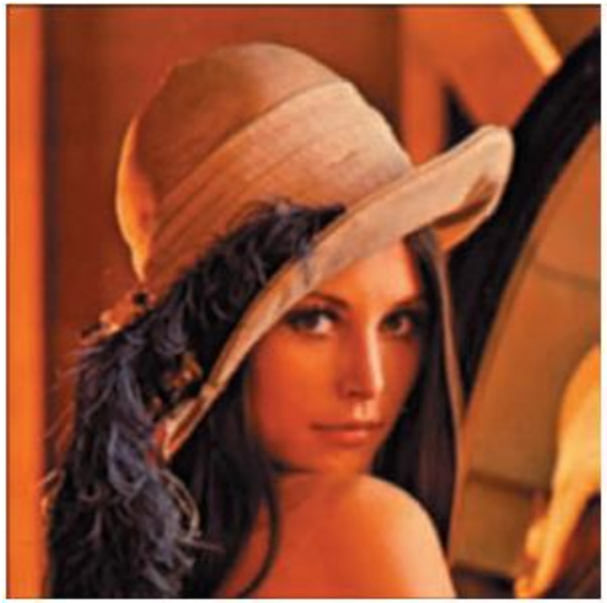
Original Image

R

G

B

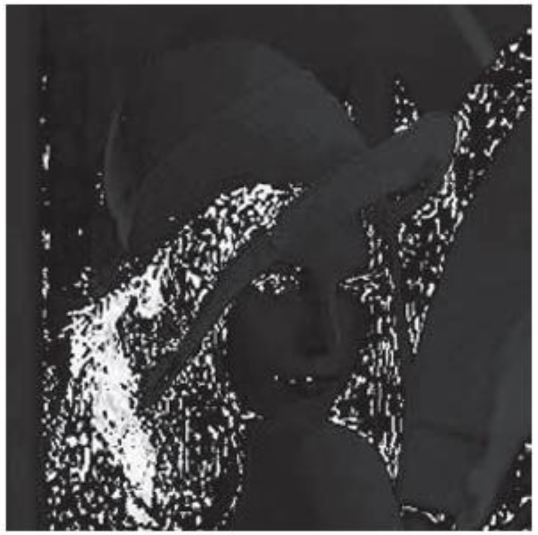
We smoothed each component image of the RGB image in left Fig. independently using a 5 × 5 averaging kernel. We then combined the individually smoothed images to form the smoothed, full-color RGB result in right Fig.



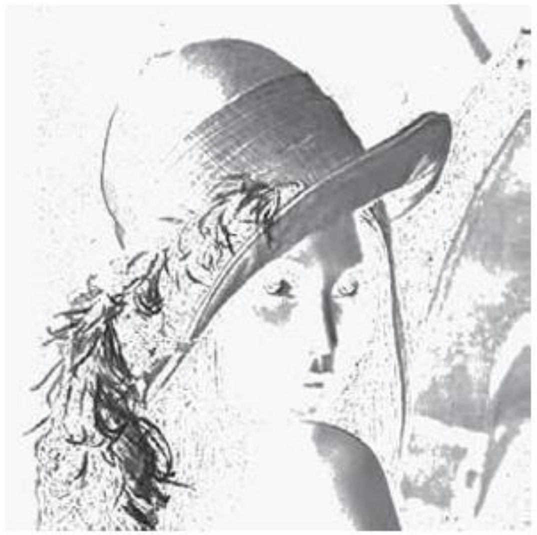
Average (smooth) Image

Color Image Smoothing

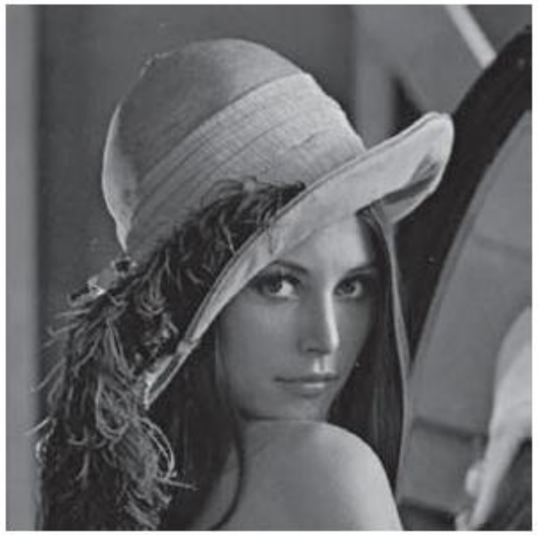
Image Smoothing via HSI



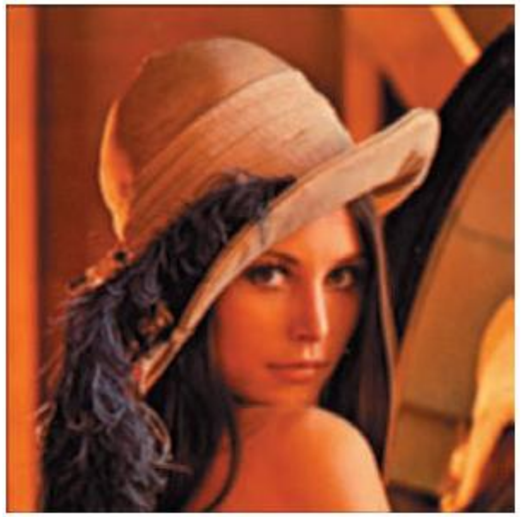
H



S



I

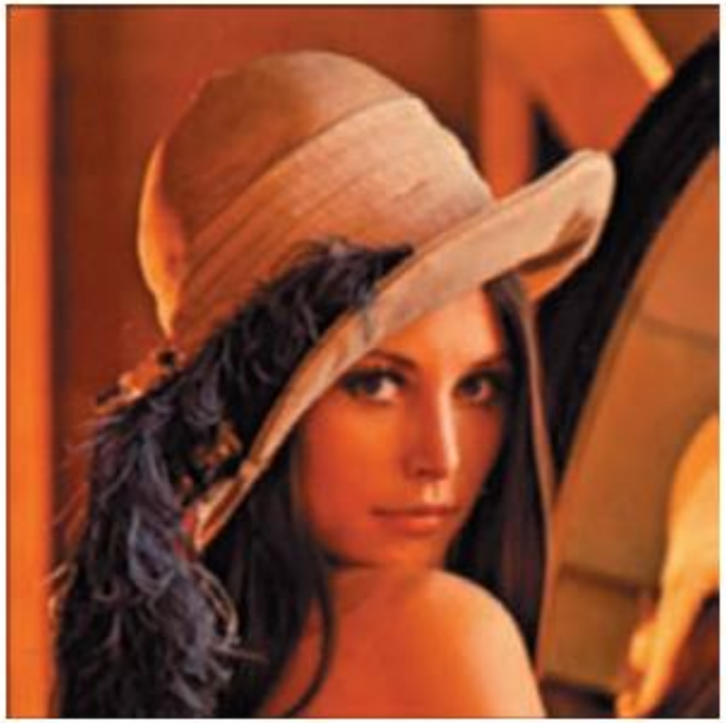


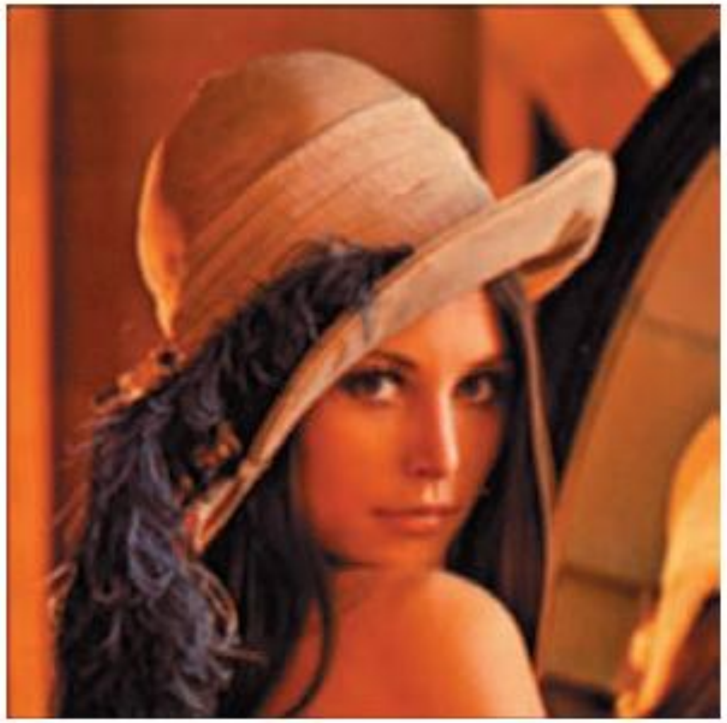
Average (smooth) Image

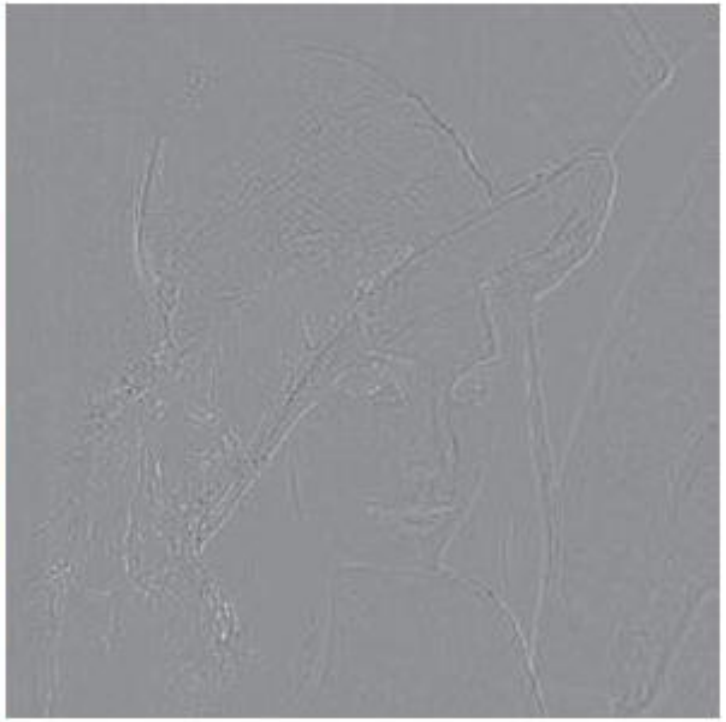
We next smooth only the intensity component, **I** (leaving the hue and saturation components unmodified) and convert the processed result to an RGB image for display.

Color Image Smoothing

Difference between both Smoothened Images


Result of processing each component i.e. RGB


Result of processing only one component i.e. I


Difference between the two results

Color Image Sharpening

We consider image sharpening using the Laplacian kernel discussed in Lecture 4.

From vector analysis, we know that the Laplacian of a vector is defined as a vector whose components are equal to the Laplacian of the individual scalar components of the input vector.

In the RGB color system, the Laplacian of vector c is,

$$\nabla^2 [c(x,y)] = \begin{bmatrix} \nabla^2 R(x,y) \\ \nabla^2 G(x,y) \\ \nabla^2 B(x,y) \end{bmatrix}$$

which, as in the previous section, tells us that we can compute the Laplacian of a full-color image by computing the Laplacian of each component image separately.

-1	-1	-1
-1	9	-1
-1	-1	-1

Kernel produce a sharper image by preserving the original intensity values

Color Image Sharpening

Image Sharpening via processing each component and only I component respectively. Also the difference of their results.

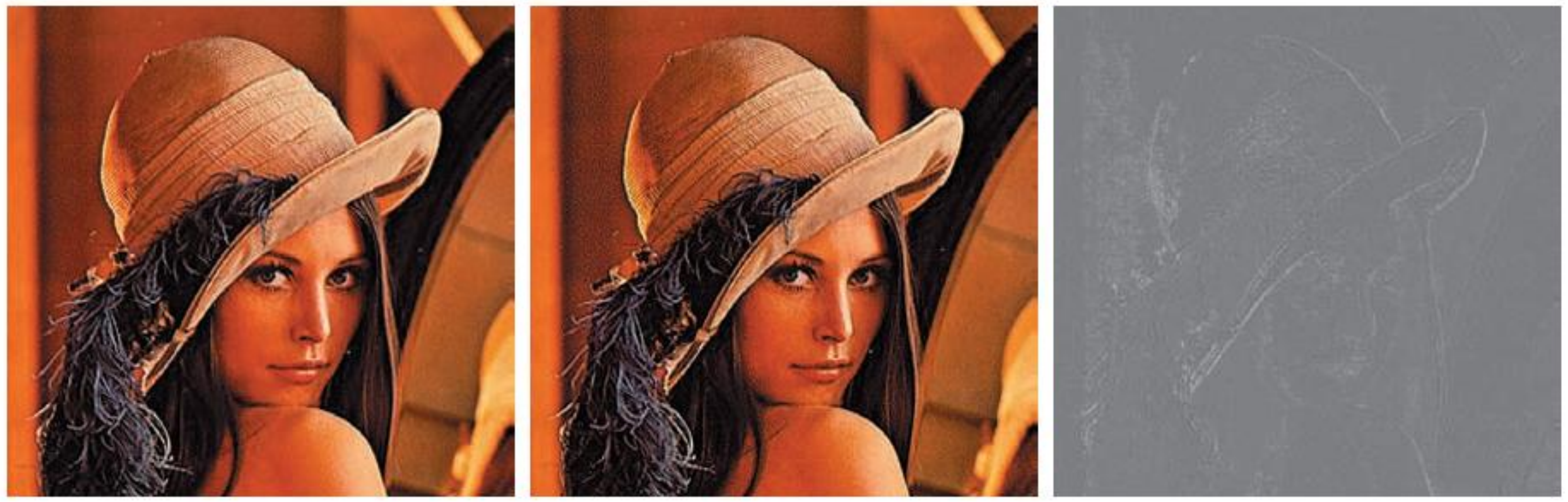


FIGURE 6.39 Image sharpening using the Laplacian. (a) Result of processing each RGB channel. (b) Result of processing the HSI intensity component and converting to RGB. (c) Difference between the two results.

Noise in Color Images

Noise in color images can be removed through various noise models which we use in Image Restoration in case the noise content of a color image has the same characteristics in each color channel.

But it is possible for color channels to be affected differently by noise so in this case noise are removed from the image by independently processing each plane.

Remove noise by applying smoothing filters (e.g gaussian, average, median) to each plane individually and then combine the result.

Here we will only focus on how noise can be spread when converting from one color model to another.

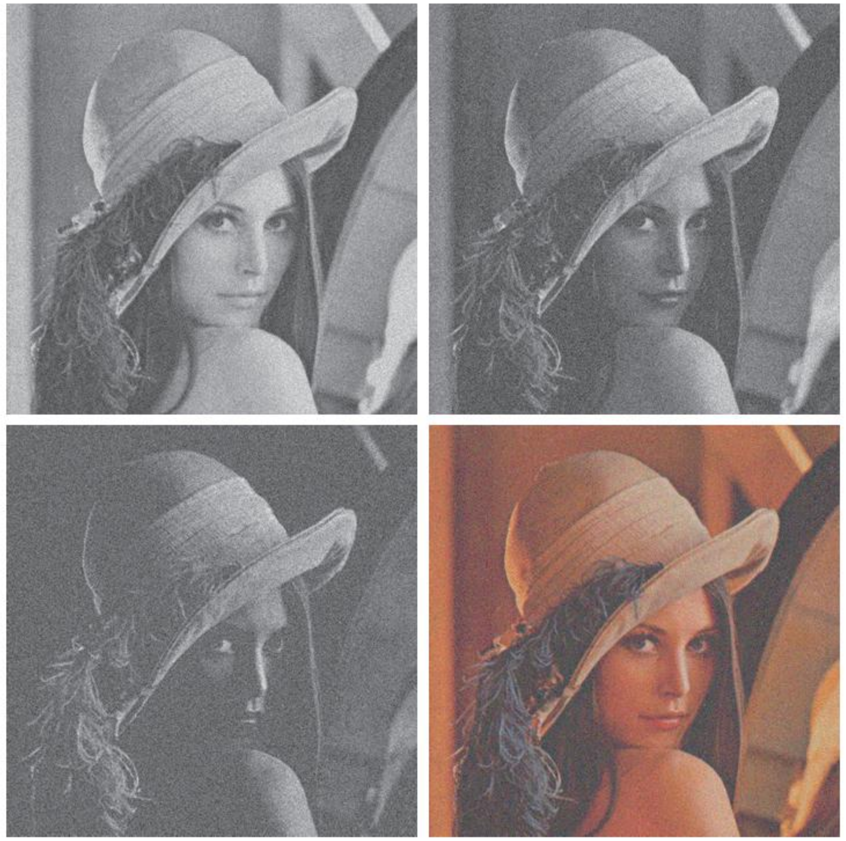
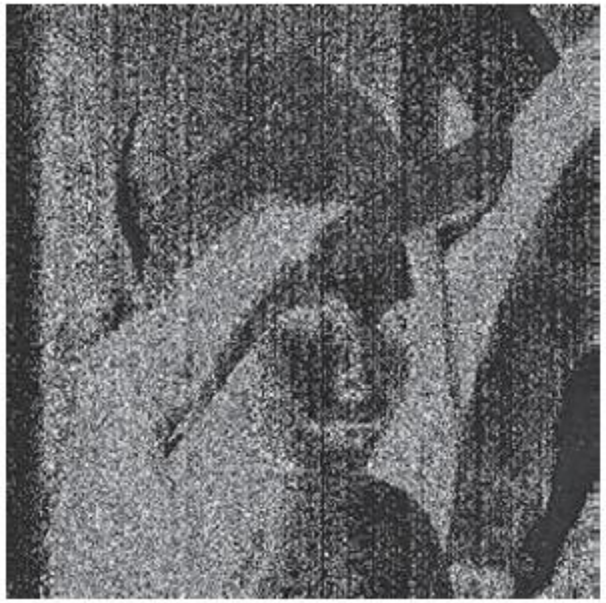

a b
c d

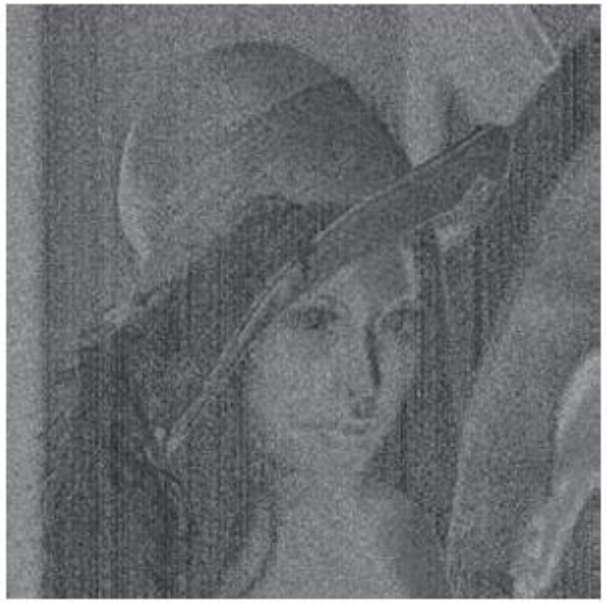
FIGURE 6.46 (a)–(c) Red, green, and blue 8-bit component images corrupted by additive Gaussian noise of mean 0 and standard deviation of 28 intensity levels. (d) Resulting RGB image. [Compare (d) with Fig. 6.44(a).]

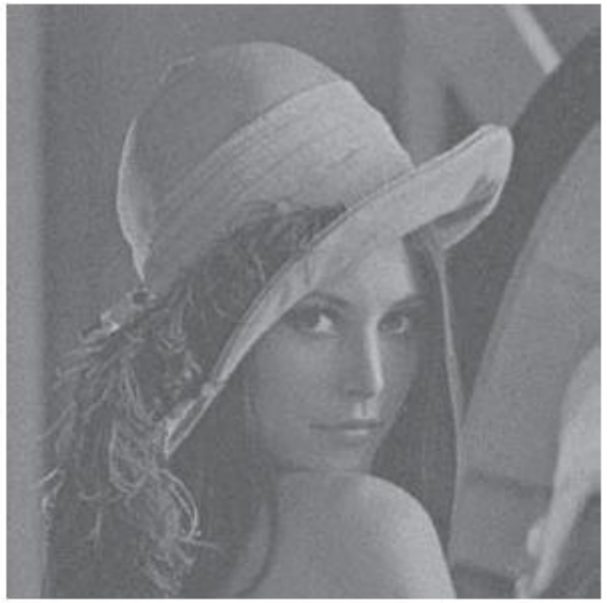
Noise in Color Images

Note how significantly degraded the hue and saturation components of the noisy image are. This was caused by the nonlinearity of the cos and min operations in Eqs. For finding both **H** and **S**, respectively.

On the other hand, the intensity component in Fig.(c) is slightly smoother than any of the three noisy RGB component images. This is because the intensity image is the average of the RGB images, as indicated in Eq. of **I**.


H


S


I

We converted the RGB image along with noise to HIS to see the effect of noise in conversion.

Color Image Compression

Compression is the process of reducing or eliminating redundant and/or irrelevant information

A compressed image is not directly displayable it must be decompressed before input to a color monitor.

In case if in a compressed image 1 bit of data represents 230 bits of data in the original image, then compressed image could be transmitted over internet in 1 minute as compared to original image which will take4 hours to transmit.

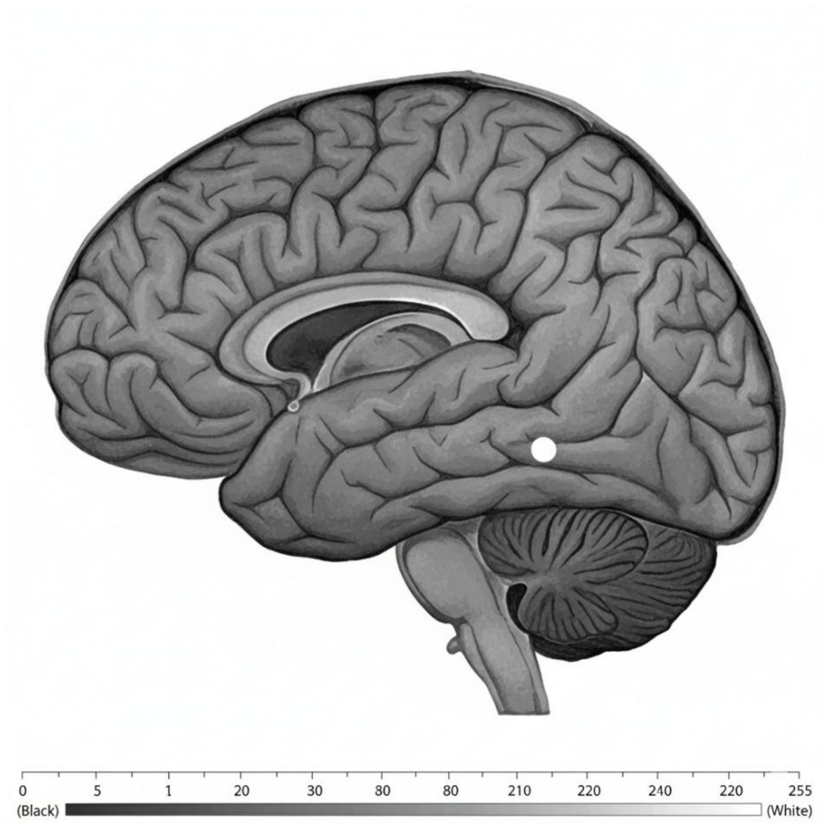
Original Image

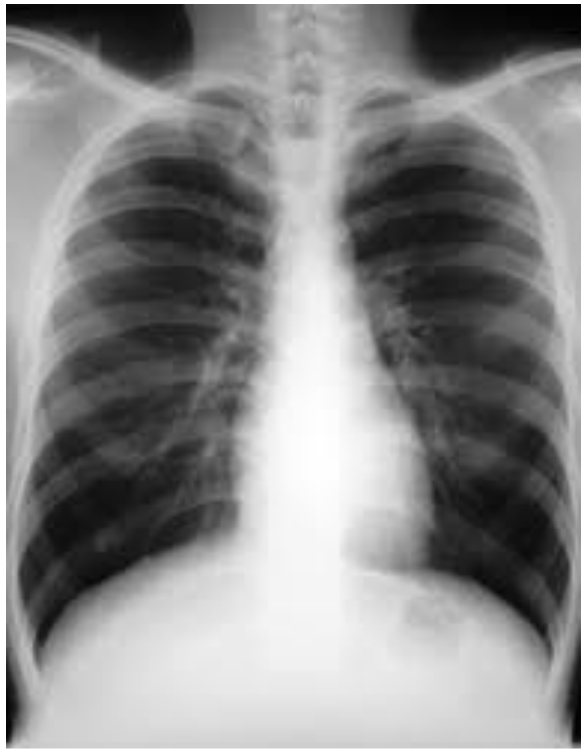
Result after Compressing and decompressing

The JPEG 2000 compression algorithm used to generate the right fig. image.

Task

Add Colors in the given images.





Thank You

For Your Attention