Digital Image Processing

Le Thanh Ha, Dr.

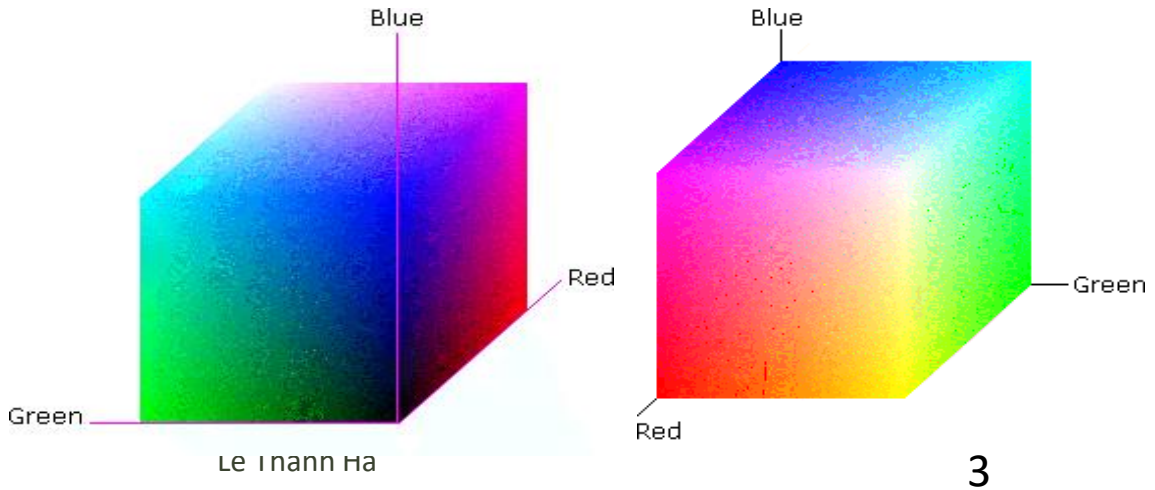
Laboratory of
Human Machine Interface

Color model

- **Color model is a mathematical tools describing color by numbers.**
- **A color model is a specification of a coordinate system where each color is represented by a single point.**
- **Each color model is used for different purpose.**
- **For example:**
  - RGB: used for display and acquaring hardware devices
  - HSV: human cognitive.
  - Y’CbCr: image compression.
  - CMYK: printers.

RGB color model

- **Based on 3D-Cartersien coordinate system in which each axis corresponds to a color component (Red, Green, Blue).**
  - The origin (0,0,0) presents Black color
  - Point (1,1,1) presents White color
  - Point in each axis present a single color component.

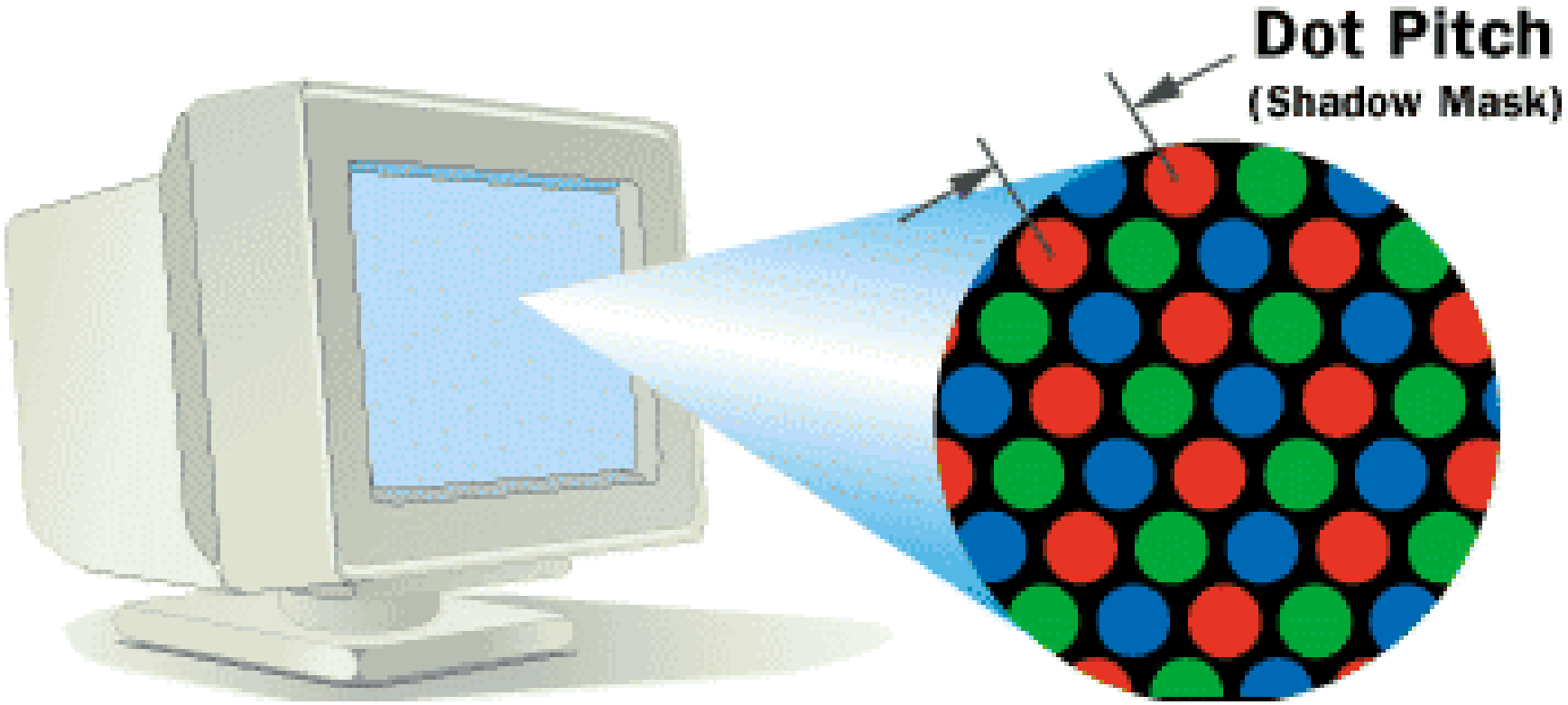


RGB color model

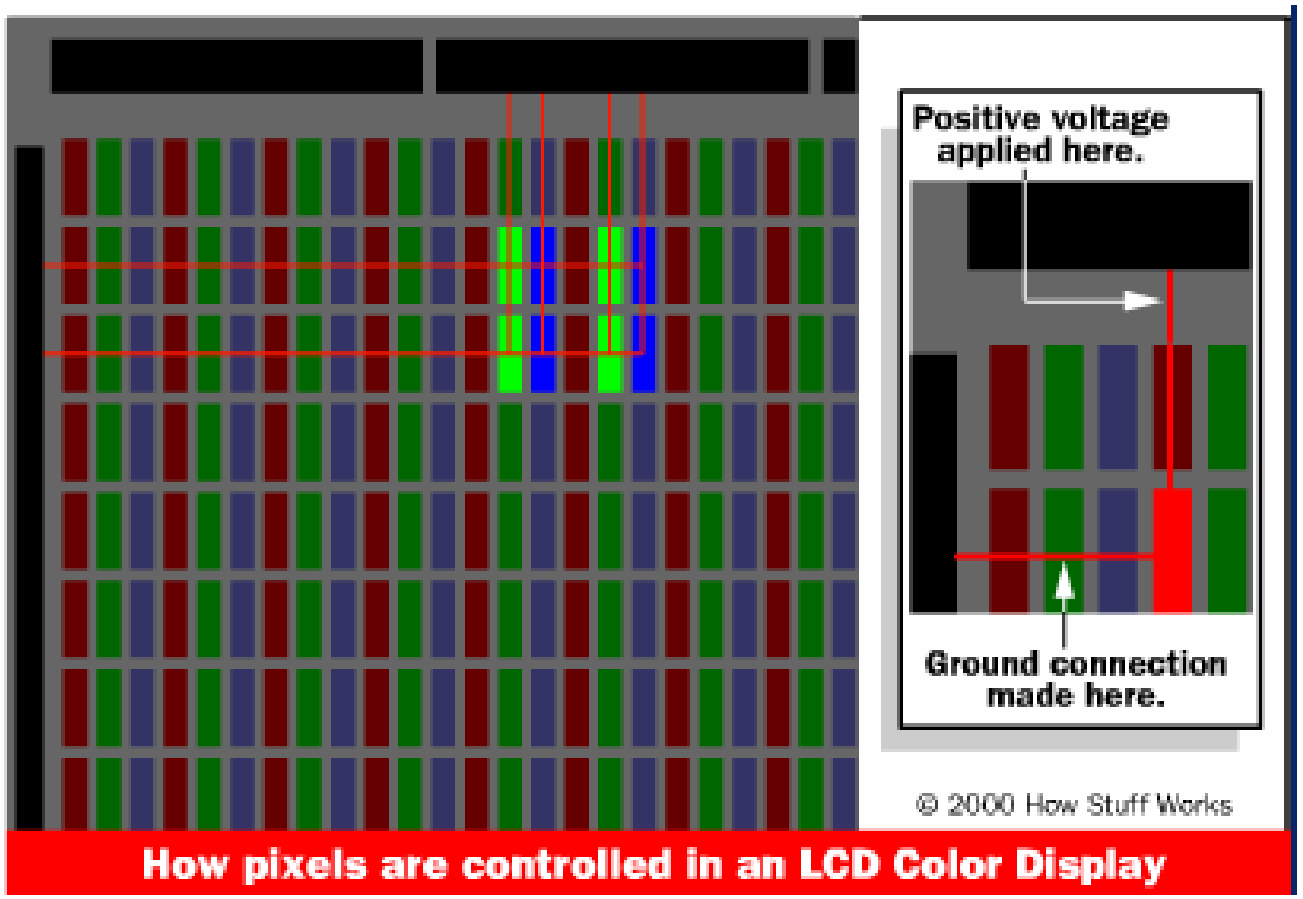
- **R, G, B are valued from 0 to 1**
  - Red+Blue -> Magenta (1, 0, 1)
  - The diagonal from (0, 0, 0) to (1, 1, 1) presents gray color.

- **Comments:**
  - This color model cannot present all color in visible spectrum, but is enough for compter applications.
  - TV, Computer display, and camera are using this color model.
  - Most widely used.

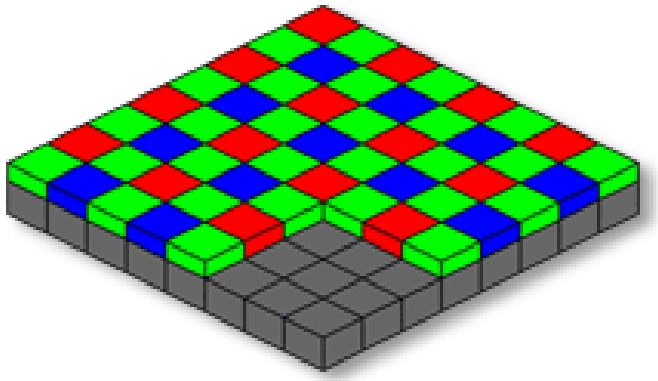
RGB color model - CRT



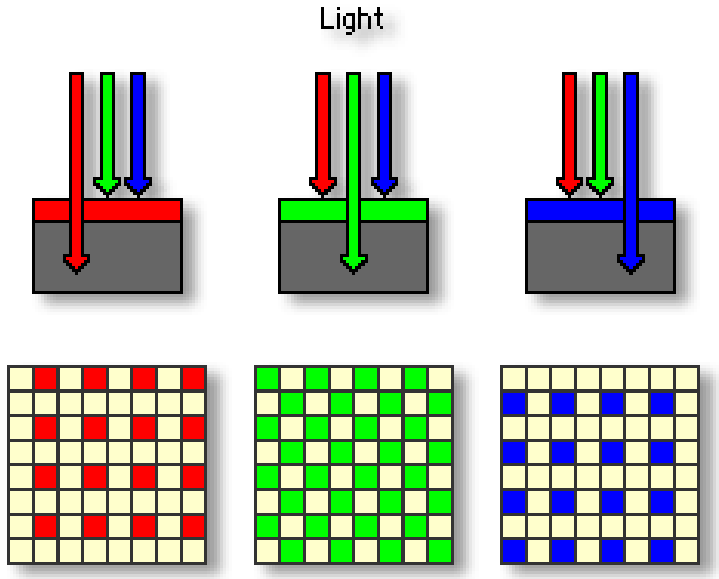
RGB color model - LCD



RGB color model – Digital Camera



Color Filter Array Sensor



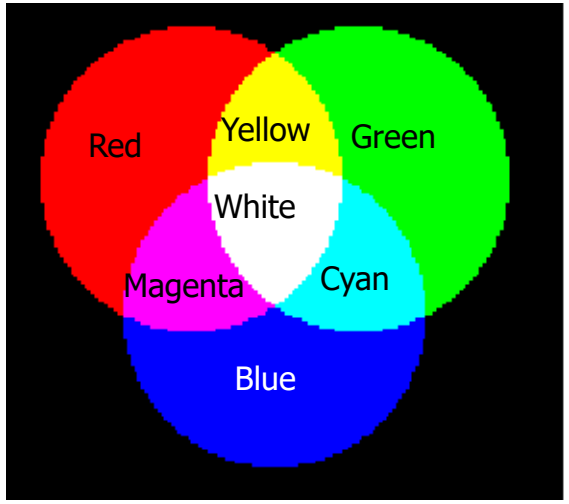
CMYK color model

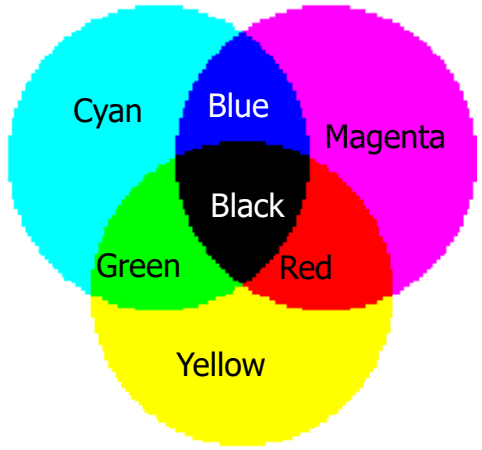
- **For a computer display: color is a combination of color component radiated from phosphor.**
- **For paper: color is reflected from the materials printed on the paper.**
  - Black ink absorb all lights coming to it.

- **→CMYK is subtractive of RGB**

CMYK color model

- **CMY uses primary colors Cyan, Magenta, and Yellow (as for color printer). They are subtractive of RGB.**
- **K = Key (black color):**
  - Mixing primary is not always perfect.
  - Use black color instead of mixing all expensive primary color to produce cheaper one.





RGB to CMY conversion

RGB -> CMY

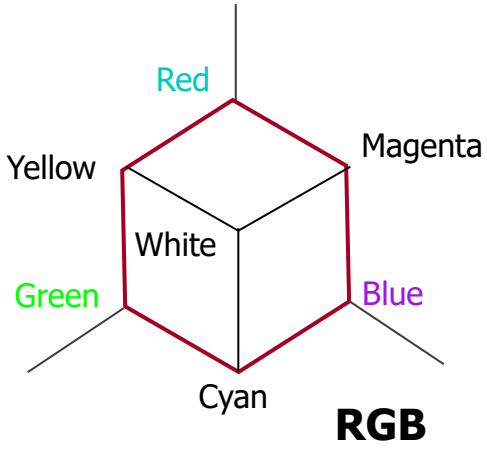
```
void RGB2CMY(float R,float G,float B,float &C,float &M,float &Y)
{
  C = 1 - R;
  M = 1 - G;
  Y = 1 - B;
}
```

RGB -> CMYK

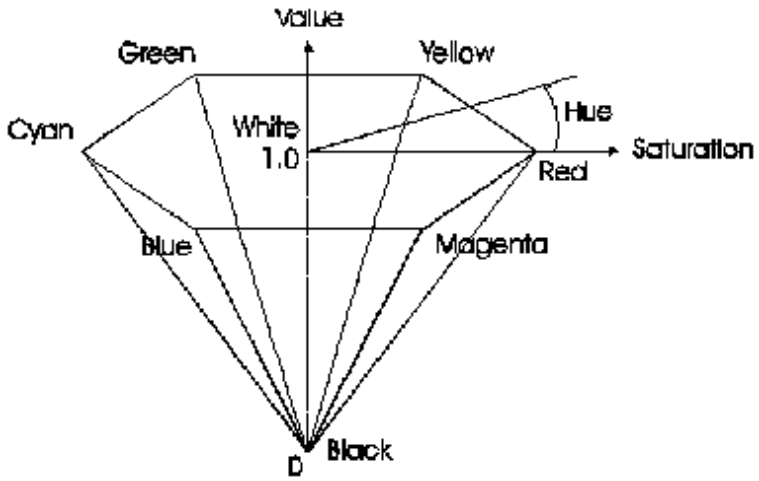
```
void RGB2CMYK(float R,float G,float B,float &C,float &M,float &Y,float &K)
{
  RGB2CMY(R, G, B, C, M, Y);
  K = min3(C, M, Y);
  C = C - K;
  M = M - K;
  Y = Y - K;
}
```

HSV color model

- **Instead of using R, G, B primary color components, this color model use Hue, Saturation and Value (HSV)**
- **RGB and HSV comparison:**
  - RGB color space is a cube
  - HSV color space is a cone

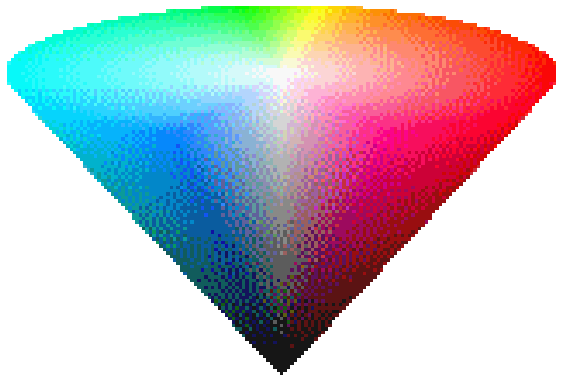


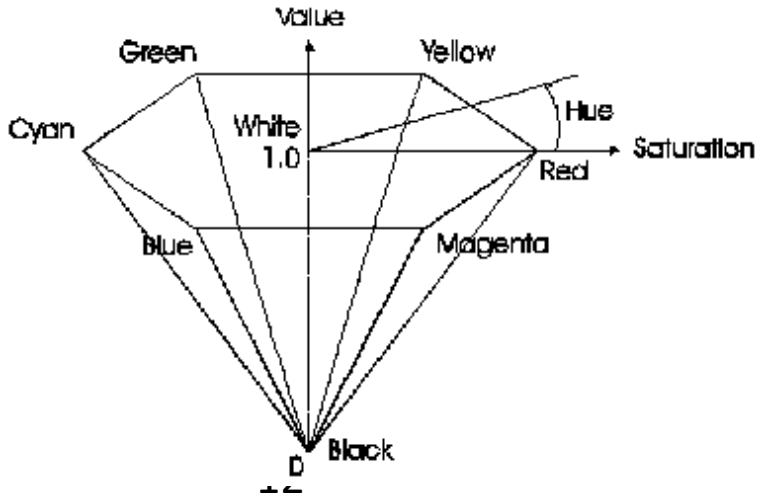
RGB



HSV color model

- **Hue: wavelength of color**
  - Is presented by an angle from 0^0 to 360^0
- **Value: value of brightness**
  - Ranging in [0, 1]
  - V=0 -> black color.
- **Saturation: purity of color**
  - Ranging in [0, 1]





Y’CbCr color model

- **Y’CbCr is used for image compression.**
  - Y’ – luminance
  - Cb - Blue-Y
  - Cr - Red-Y
- **RGB to Y’CbCr conversion**

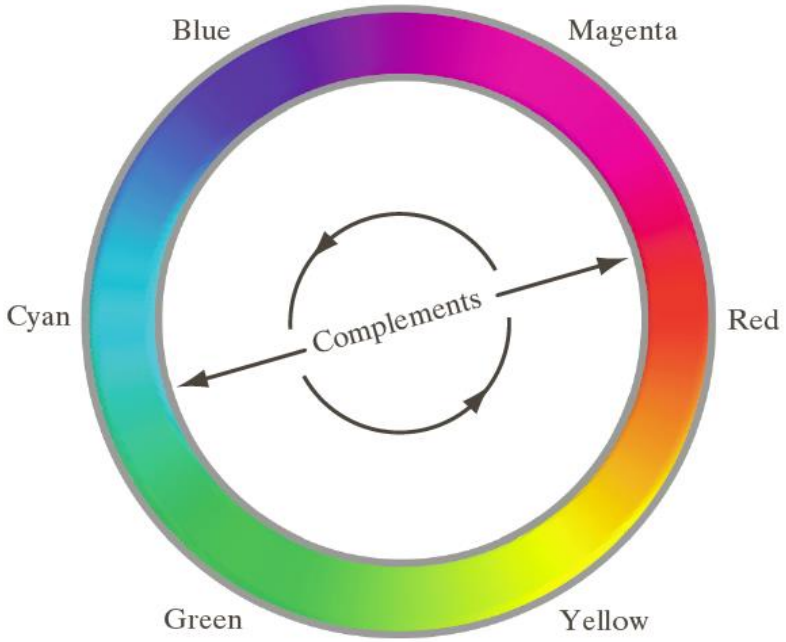
$$Y' = 0.299R + 0.587G + 0.114B$$
$$Cb = -0.147R - 0.289G + 0.436B$$
$$Cr = 0.615R - 0.515G - 0.1B$$

Color Image Processing

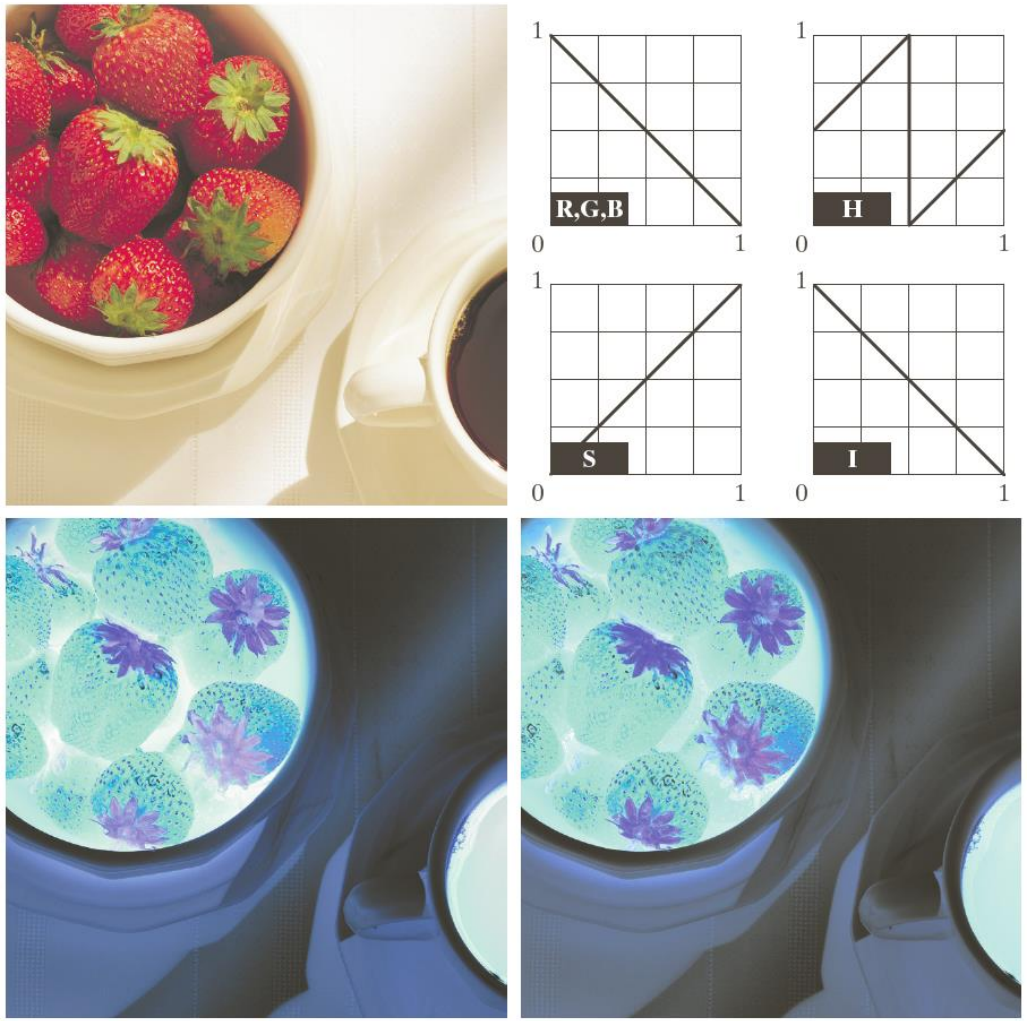
- **Color compliment**
- **Color slicing**
- Color correction
- Color image segmentation

Color compliment

- Like negative operator in gray-scale images
- Enhancing detail embedded in dark regions.



Color compliment



a b
c d

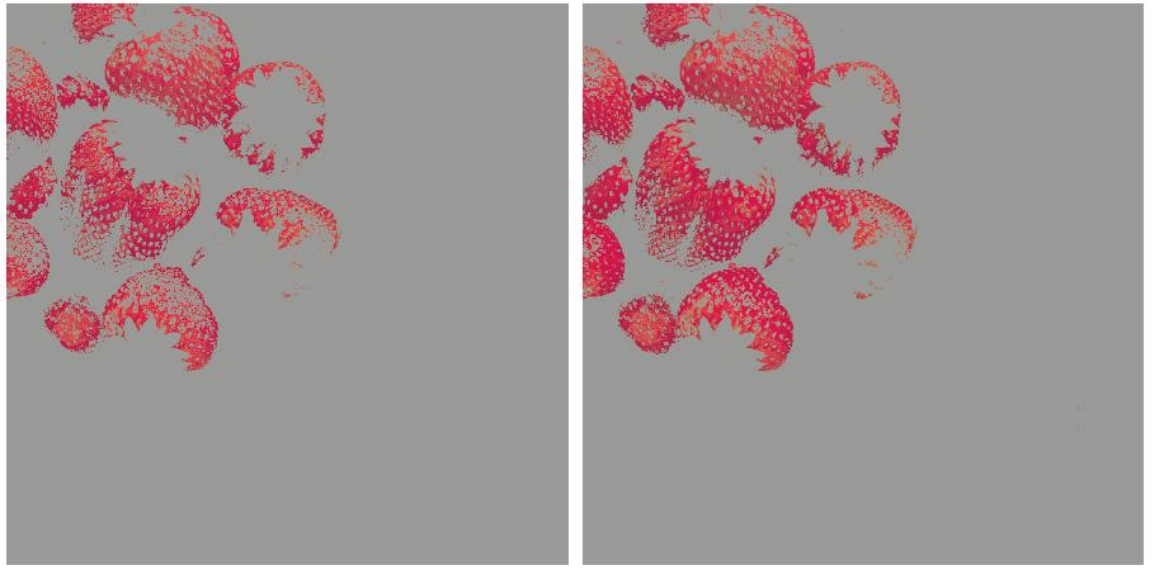
FIGURE 6.33 Color complement transformations. (a) Original image. (b) Complement transformation functions. (c) Complement of (a) based on the RGB mapping functions. (d) An approximation of the RGB complement using HSI transformations.

Color slicing

- Highlighting a specific range of color in an image (object extraction)
- Extending the intensity slicing in gray-scale case

$$s_i = \begin{cases} 0.5 & \text{if } \left[\left| r_{ij} - a_j \right| > \frac{W}{2} \right]_{\text{any } 1 \le j \le 3} \\ r_i & \text{otherwise} \end{cases}$$

Color slicing



a b

FIGURE 6.34 Color-slicing transformations that detect (a) reds within an RGB cube of width $W = 0.2549$ centered at (0.6863, 0.1608, 0.1922), and (b) reds within an RGB sphere of radius 0.1765 centered at the same point. Pixels outside the cube and sphere were replaced by color (0.5, 0.5, 0.5).

Human skin region extraction

- Have wide variety of real-life applications related with the human activity
- A very simple and fast method may look like this:

```
int Y  =  16 + ( 65.4810*red + 128.5530*green +  24.9660*blue)/256;
int Cb = 128 + (-37.7745*red -  74.1592*green + 111.9337*blue)/256;
int Cr = 128 + (111.9581*red -  93.7509*green -  18.2072*blue)/256;

if((Y>8) && (85<Cb) && (Cb<135) && (135<Cr) && (Cr<180)) //Skin pixel
    count++;
```