Lecture 8
Introduction to
Color Image Processing

Lin ZHANG, PhD
School of Software Engineering
Tongji University
Spring 2013
Outline

- Color fundamentals
- Color models
- Pseudocolor image processing
- Basics of full-color image processing
Color Fundamentals

In 1666, Isaac Newton discovered that when a beam of sunlight passes through a glass prism, the emerging beam is split into a spectrum of colors.
Color Fundamentals

The colors that humans and most animals perceive in an object are determined by the nature of the light reflected from the object.

For example, green objects reflect light with wavelengths primarily in the range of 500 – 570 nm while absorbing most of the energy at other wavelengths.
Color Fundamentals

Chromatic light spans the electromagnetic spectrum from approximately 400 to 700 nm.

As we mentioned before, human color vision is achieved through 6 to 7 million cones in each eye.
Outline

• Color fundamentals

• Color models
  • RGB
  • CMY (CMYK)
  • HSI
  • NTSC
  • YCbCr
  • CIE L*a*b*

• Pseudocolor image processing

• Basics of full-color image processing
Color Model

- The purpose of a color model is to facilitate the specification of colors in some standard
- In essence, a color model is a specification of a coordinate system and a subspace within that system where each color is represented by a single point
- Most color models are oriented either toward specific hardware or toward applications
RGB Model

- **Red**, **green**, and **blue**, three primary colors
  - Cone cells in human eye are responsible for color vision

![Diagram showing absorption spectrum with peaks at 445 nm, 535 nm, and 575 nm, corresponding to blue, green, and red colors respectively. The x-axis represents wavelength from 400 nm to 700 nm, and the y-axis represents absorption in arbitrary units.](image-url)
RGB Model

- **Red**, **green**, and **blue**, three primary colors
  - Approximately 66% of these cones are sensitive to red light, 33% to green light and 6% to blue light
  - Absorption curves for the different cones have been determined experimentally
  - For standardization, the CIE (in 1931) designated red (700nm), green (546.1nm) and blue (435.8nm) light as three primary colors; they are mixed to generate other spectral colors

Note: the CIE standard is not quite consistent with the response curves of the cone cells; this is because the curves are not available till 1965
RGB Model

• Some notes on primary colors
  
  • Having three specific primary wavelengths for the purpose of standardization does not mean these three RGB components acting alone can generate all spectrum colors
  
  • The word “primary” has been widely misinterpreted to mean that the three standard primaries, when mixed in various intensity proportions, can produce all visible colors
  
  • When the three primaries are fixed, we can approximate all the spectrum colors by mixing the primaries (usually through subjective experiments)
RGB Model

- RGB model
  - This model is based on a Cartesian coordinate system
  - The color subspace of interest is the cube, in which RGB primary values are at three corners; black is at the origin while white is at the corner farthest from the origin
RGB Model

- RGB model
  - An RGB color image is an $M \times N \times 3$ array of color pixels, where each color pixel is a triplet, corresponding to the red, green, and blue components.
  - An RGB image may be viewed as a “stack” of three grayscale images.
RGB Model

- RGB model

The three color components of a color pixel, arranged as a column vector:

\[
\begin{bmatrix}
Z_R \\
Z_G \\
Z_B
\end{bmatrix}
\]

Red component image
Green component image
Blue component image
RGB Model

- RGB model

In Matlab, routine "cat" can be used to stack three components to form an RGB image.
Outline

• Color fundamentals
• Color models
  • RGB
  • CMY (CMYK)
  • HSI
  • NTSC
  • YCbCr
  • CIE L*a*b*
• Pseudocolor image processing
• Basics of full-color image processing
CMY (CMYK) Model

- The primary colors (R, G, B) can be added to produce the *secondary* colors
  - Red plus blue can generate magenta
  - Green plus blue can generate cyan
  - Red plus green can generate yellow
CMY (CMYK) Model

- CMY color space
  - It takes cyan, magenta, and yellow as “primary colors”
  - CMY is widely used for painting and printing

CMY model can be easily converted from RGB model by

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

where all color values have been normalized to the range [0, 1]
CMY (CMYK) Model

- CMY color space
  - It takes cyan, magenta, and yellow as “primary colors”
  - CMY is widely used for painting and printing

Implementation Tips

1) Conversion between RGB and CMY images can be easily implemented via “imcomplement”
CMY (CMYK) Model

- CMYK color space
  - Theoretically speaking, equal amounts of pigments primaries, cyan, magenta, and yellow should produce “black”
  - In practice, such a method can generate a quite muddy black
  - Thus, to produce “true black”, a fourth color, “black” is added and gives rise to the CMYK color model. (K means “black in $K$”)
Outline

• Color fundamentals
• Color models
  • RGB
  • CMY (CMYK)
  • HSI
  • NTSC
  • YCbCr
  • CIE L*a*b*
• Pseudocolor image processing
• Basics of full-color image processing
Human eye distinguish one color from the other based on \textit{hue, saturation, and brightness}

- Hue, is a color that is evoked by a single wavelength of light in the visible spectrum, or by a relatively narrow band of wavelengths; hue represents dominant color as perceived by the observer.

- Saturation (purity) refers to the relative amount of white light mixed with a hue; it is inversely proportional to the amount of white light added; to desaturate a color of given intensity in a subtractive system (such as watercolor), one can add white, black, gray.

- Brightness embodies the achromatic notion of intensity.
HSI Model

- Hue and saturation together are called **chromaticity**
- For any particular color, the amounts of red, green, and blue needed to form it are called **tristimulus** values, and they are denoted as $X$, $Y$, and $Z$

A color is specified by its **trichromatic coefficients**, defined as

$$
x = \frac{X}{X + Y + Z}
$$

$$
y = \frac{Y}{X + Y + Z}
$$

$$
z = \frac{Z}{X + Y + Z}
$$

Of course,

$$
x + y + z = 1
$$
HSI Model

- CIE chromaticity diagram
  - It is a function of $x$ (red) and $y$ (green)
  - $z$ can be derived by $z = 1 - x - y$
HSI Model

• Some notes on CIE chromaticity diagram
  • Any color located on the boundary of the chromaticity chart is fully saturated; any point not on the boundary but within the diagram represents some mixture of spectrum colors
  • The point of equal energy represents the standard white light; its saturation is zero
  • As a point leaves the boundary and approaches the point of equal energy, more white light is added to the color and it becomes less saturated
  • A straight line segment joining any two points in the diagram defines all the different color variations that can be obtained by combining these two colors additively
HSI Model

- The HSI model uses three measures to describe colors, hue, saturation, and intensity
- Their relationship can be represented in a cylindrical coordinate system
  - angle around the central vertical axis corresponds to "hue",
  - the distance from the axis corresponds to "saturation",
  - the distance along the axis corresponds to "value"
HSI Model

- Illustration for HSI model using a figure
HSI Model

- Conversion from RGB to HSI

\[
H = \begin{cases} 
\theta & \text{if } B \leq G \\
360 - \theta & \text{if } B > G 
\end{cases}
\]

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

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

\[
I = \frac{1}{3} (R + G + B)
\]
HSI Model

• Conversion from HSI to RGB

• RG sector ($0 \leq H < 120^\circ$)

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

• GB sector ($120^\circ \leq H < 240^\circ$)

\[H = H - 120^\circ\]

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

- Conversion from HSI to RGB

  - BR sector \((240^\circ \leq H \leq 360^\circ)\)

  \[
  H = H - 240^\circ 
  \]

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

- Color fundamentals
- Color models
  - RGB
  - CMY (CMYK)
  - HSI
  - NTSC
  - YCbCr
  - CIE L*a*b*
- Pseudocolor image processing
- Basics of full-color image processing
NTSC Model

- NTSC (National Television Standards Committee) color system is used in analog television
- In this space, gray-scale information is separated from color data; so the same signal can be used for both color and monochrome TV sets
- Image data consists of three components, luminance (Y), hue (I), and saturation (Q)
NTSC Model

- Conversion from RGB to YIQ

\[
\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.211 & -0.523 & 0.312 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}
\]

Especially,

\[ Y = 0.299R + 0.587G + 0.114B \]

This is what Matlab does when you call rgb2gray
NTSC Model

- Conversion from YIQ to RGB

\[
\begin{bmatrix}
R \\
G \\
B
\end{bmatrix}
= \begin{bmatrix}
1.000 & 0.956 & 0.621 \\
1.000 & -0.272 & -0.647 \\
1.000 & -1.106 & 1.703
\end{bmatrix}
\begin{bmatrix}
Y \\
I \\
Q
\end{bmatrix}
\]

**Implementation Tips**

1) Conversion from RGB to YIQ is implemented in “rgb2ntsc”
2) Conversion from YIQ to RGB is implemented in “ntsc2rgb”
NTSC Model

- An example, an RGB image and its Y, I, Q components
Outline

• Color fundamentals
• Color models
  • RGB
  • CMY (CMYK)
  • HSI
  • NTSC
  • YCbCr
  • CIE L*a*b*
• Pseudocolor image processing
• Basics of full-color image processing
YCbCr Model

- YCbCr color space is extensively used in digital video
- In this model, luminance is represented by a single component, Y
- Color information is stored as two color-difference components, Cb, and Cr
- Cb is the difference between the blue component and a reference value; Cr is the difference between the red component and a reference value
YCbCr Model

- Conversion from RGB to YCbCr

\[
\begin{bmatrix}
Y \\
Cb \\
Cr
\end{bmatrix} = \begin{bmatrix}
16 \\
128 \\
128
\end{bmatrix} + \begin{bmatrix}
65.481 & 128.553 & 24.966\\
-37.797 & -74.203 & 112.000\\
112.000 & -93.786 & -18.214
\end{bmatrix} \begin{bmatrix}
R \\
G \\
B
\end{bmatrix}
\]

Implementation Tips

1) Conversion from RGB to YCbCr is implemented in “rgb2ycbcr”
2) Conversion from YCbCr to RGB is implemented in “ycbcr2rgb”
YCbCr Model

- An RGB image and its Y, Cb, Cr components
Outline

• Color fundamentals
• Color models
  • RGB
  • CMY (CMYK)
  • HSI
  • NTSC
  • YCbCr
  • CIE L*a*b*
• Pseudocolor image processing
• Basics of full-color image processing
CIE L*a*b* Model

• CIE L*a*b* Model (proposed in CIE, 1976)
  • The L*a*b* system is based on the three dimensional coordinate system based on the opponent theory using black-white L*, red-green a*, and yellow-blue b* components.
  • It is perceptually uniform, which means that numerical distances can be related to human perceptual differences
  • It is device independent
  • L*a*b* values do not define absolute colors unless the white point is also specified
CIE L*a*b* Model

The \( L^*a^*b^* \) coordinates are computed from the \( X, Y \) and \( Z \) tristimulus values as follows:

\[
L^* = 116 f \left( \frac{Y}{Y_n} \right) - 16
\]

\[
a^* = 500 \left[ f \left( \frac{X}{X_n} \right) - f \left( \frac{Y}{Y_n} \right) \right], \text{ where } f(t) = \begin{cases} 
  t^{1/3}, & \text{if } t > \left( \frac{6}{29} \right)^3 \\
  \frac{1}{3} \left( \frac{29}{6} \right)^2 t + \frac{4}{29}, & \text{otherwise}
\end{cases}
\]

\[
b^* = 200 \left[ f \left( \frac{Y}{Y_n} \right) - f \left( \frac{Z}{Z_n} \right) \right]
\]

Here \( X_n, Y_n \) and \( Z_n \) are the CIE XYZ tristimulus values of the reference white point (the subscript \( n \) suggests "normalized")
CIE L*a*b* Model

Implementation Tips

1) Conversions between RGB and CIEL*a*b* can be implemented by “makecform” and “applycform”

An example,

```matlab
imRGB = imread('gate1.bmp');
cform = makecform('srgb2lab','AdaptedWhitePoint', whitepoint('D65'));
imLab = applycform(imRGB, cform);
```
CIE L*a*b* Model

- CIE L*a*b* Model

RGB image

I* component

a* component

b* component
Outline

• Color fundamentals
• Color models
• Pseudocolor image processing
• Basics of full-color image processing
Pseudocolour Image Processing

- Pseudocolour image processing consists of assigning colors to grey values based on a specific criterion.
- The principle use of pseudocolour image processing is for human visualization.
  - Humans can discern between thousands of color shades and intensities, compared to only about two dozen or so shades of grey.
Pseudocolour Image Processing – Intensity Slicing

- Intensity slicing and color coding is one of the simplest kinds of pseudocolour image processing
  - First we consider an image as a 3D function mapping spatial coordinates to intensities (that we can consider heights)
  - Now consider placing planes at certain levels parallel to the coordinate plane
  - If a value is one side of such a plane it is rendered in one color, and a different color if on the other side
Pseudocolour Image Processing – Intensity Slicing

- Intensity slicing and color coding is one of the simplest kinds of pseudocolour image processing
Pseudocolor Image Processing – Intensity Slicing

- Intensity slicing and color coding is one of the simplest kinds of pseudocolor image processing

In general intensity slicing can be summarized as:

- Let $[0, L-1]$ represent the grey scale
- Let $l_0$ represent black [$f(x, y) = 0$] and let $l_{L-1}$ represent white [$f(x, y) = L-1$]
- Suppose $P$ planes perpendicular to the intensity axis are defined at levels $l_1, l_2, ..., l_p$
- Assuming that $0 < P < L-1$, then the $P$ planes partition the grey scale into $P + 1$ intervals $V_1, V_2, ..., V_{P+1}$
Grey level color assignments can then be made according to the relation:

$$f(x,y) = c_k \quad \text{if } f(x,y) \in V_k$$

where $c_k$ is the color associated with the $k^{th}$ intensity level $V_k$ defined by the partitioning planes at $l = k - 1$ and $l = k$.
Pseudocolour Image Processing – Intensity Slicing

An example
Pseudocolour Image Processing – Intensity Slicing

Another example
Pseudocolour Image Processing – Intensity Slicing

One more example
Outline

- Color fundamentals
- Color models
- Pseudocolor image processing
- Basics of full-color image processing
Basics of Full-Color Image Processing

- Full-color image processing approaches fall into two major categories
  - In the first category, we process each component image individually and then form a composite processed color image from the individually processed components
  - In the second category, we work with color pixels directly
Basics of Full-Color Image Processing

- Color transformations

Color transformations can be of the form

\[ s_i = T_i (r_i), \quad i = 1, 2, \ldots, n \]

where \( r_i \) and \( s_i \) are the color components of the input and output images, \( n \) is the dimension of the color space. \( T_i \) are referred to as **full-color transformation** or **mapping** functions.
Basics of Full-Color Image Processing

- Color transformations

Examples for color mapping functions. (a) and (c) are linear interpolation; (b) and (d) are cubic spline interpolation.
Basics of Full-Color Image Processing

- Color transformations

Implementation Tips

1) Linear interpolation by using control points is implemented in "interp1q"
2) Cubic spline interpolation by using control points is implemented in "spline"
Basics of Full-Color Image Processing

- Color transformations

Gonzalez provides an interactive color editor, see our course website
It implements the color transformation functions
It also provides an excellent demo for how to develop application with GUIs in Matlab

Source code is available on our course website
Basics of Full-Color Image Processing

- Color transformations

GUI for ice
Thanks for your attention