ShanTou University

School of Mathematics and Computer Science

CG Assignment 8

Author:

ZiXian-Zhu

Supervisor:

Prof Liao

An Assignment submitted for the STU:

202420251001081 计算机图形学

2024 年 12 月 11 日

Format Instructions

This article uses the following format to emphasize different types of information:

- blue bold: important concepts or definitions
- red bold: very important information
- green italic: additional explanation or comment
- underline: keywords or terms

1 Parameters and Applications of Color Models

1.1 RGB Color Model

RGB is an additive color model that produces various colors by combining three primary colors (Red, Green, Blue) at different intensities. In computer systems, each color component is typically represented using 8 bits (0-255).

Parameter	Description
Red (R)	Range 0-255, represents red light intensity. 0 means no red light, 255 means maximum red intensity
Green (G)	Range 0-255, represents green light intensity. 0 means no green light, 255 means maximum green intensity
Blue (B)	Range 0-255, represents blue light intensity. 0 means no blue light, 255 means maximum blue intensity

Table 1: Detailed RGB Color Model Parameters

1.1.1 RGB Storage in Computer Systems

In computer systems, RGB colors are typically stored in one of the following formats:

- 24-bit True Color: 8 bits per color component, totaling 24 bits
- 32-bit Color: 8 bits each for RGB, plus 8 bits for Alpha channel (transparency)

```python
import numpy as np
from PIL import Image

# Create RGB image
img = np.zeros((100, 100, 3), dtype=np.uint8)
# Set red color
img[:, :, 0] = 255  # Set R channel to maximum
# Create PIL image object
pil_img = Image.fromarray(img)
```

Listing 1: RGB Color Processing in Python

1.2 CMY Color Model

CMY is a subtractive color model primarily used in printing and pigment mixing. Each parameter represents the degree of absorption of specific wavelengths of light.

Parameter	Description
Cyan (C)	Range 0-100%, absorbs red light. 0% means no absorption, 100% means complete absorption
Magenta (M)	Range 0-100%, absorbs green light. 0% means no absorption, 100% means complete absorption
Yellow (Y)	Range 0-100%, absorbs blue light. 0% means no absorption, 100% means complete absorption

Table 2: Detailed CMY Color Model Parameters

```python
import cv2
import numpy as np

def rgb_to_cmy(rgb_image):
    # Normalize RGB values
    rgb_normalized = rgb_image.astype(float) / 255.0
    # Convert to CMY
    cmy = 1 - rgb_normalized
    return cmy
```

Listing 2: CMY Conversion Example in OpenCV

1.3 HSV Color Model

HSV model describes colors in a way that aligns with human perception, making it more intuitive for human understanding.

Parameter	Description
Hue (H)	Range 0°-360°, represents color type. 0° is red, 120° is green, 240° is blue
Saturation (S)	Range 0-100%, represents color intensity. 0% is grayscale, 100% is pure color
Value (V)	Range 0-100%, represents brightness. 0% is black, 100% is maximum brightness

Table 3: Detailed HSV Color Model Parameters

```python
import cv2
import numpy as np

# Read image
img = cv2.imread('image.jpg')
# Convert to HSV space
hsv = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)

# Define HSV range for red color
lower_red = np.array([0, 100, 100])
upper_red = np.array([10, 255, 255])

# Create mask
mask = cv2.inRange(hsv, lower_red, upper_red)
```

Listing 3: HSV Color Detection in OpenCV

2 Relationship Between CMY and RGB

2.1 Mathematical Relationship

The RGB and CMY color spaces have an exact mathematical relationship, expressed as:

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

2.2 Physical Principles

- RGB Principle: Emits different wavelengths of light (additive)
- CMY Principle: Absorbs different wavelengths of light (subtractive)
- Complementary Relationships:
  - Cyan (C) absorbs Red (R)
  - Magenta (M) absorbs Green (G)
  - Yellow (Y) absorbs Blue (B)

2.3 Implementation Examples

```python
def rgb_to_cmy(r, g, b):
    """
    Convert RGB values to CMY values
    Input: r,g,b (0-255)
    Output: c,m,y (0-1)
    """
    # Normalize RGB values
    r = r / 255.0
    g = g / 255.0
    b = b / 255.0

    # Convert to CMY
    c = 1 - r
    m = 1 - g
    y = 1 - b

    return c, m, y

def cmy_to_rgb(c, m, y):
    """
    Convert CMY values to RGB values
    Input: c,m,y (0-1)
    Output: r,g,b (0-255)
    """
```

```
    # Convert to RGB
    r = (1 - c) * 255
    g = (1 - m) * 255
    b = (1 - y) * 255

    return int(r), int(g), int(b)
```

Listing 4: Color Space Conversion Implementation

2.4 Real-world Engineering Applications

- Adobe Photoshop: Simultaneous use of RGB and CMY
  - Monitor preview uses RGB
  - Print output uses CMY(K)
- OpenCV: Applications in Computer Vision

```
import cv2

# Read RGB image
rgb_img = cv2.imread('image.jpg')
# Convert to CMYK (OpenCV uses BGR format)
cmyk_img = cv2.cvtColor(rgb_img, cv2.COLOR_BGR2CMYK)

```

Listing 5: OpenCV Color Space Conversion

- Printer Drivers: Real-time Color Conversion
  - Receives RGB data
  - Converts to CMY(K) using ICC profiles
  - Performs color correction and optimization