Color Image Processing

Color Image Processing

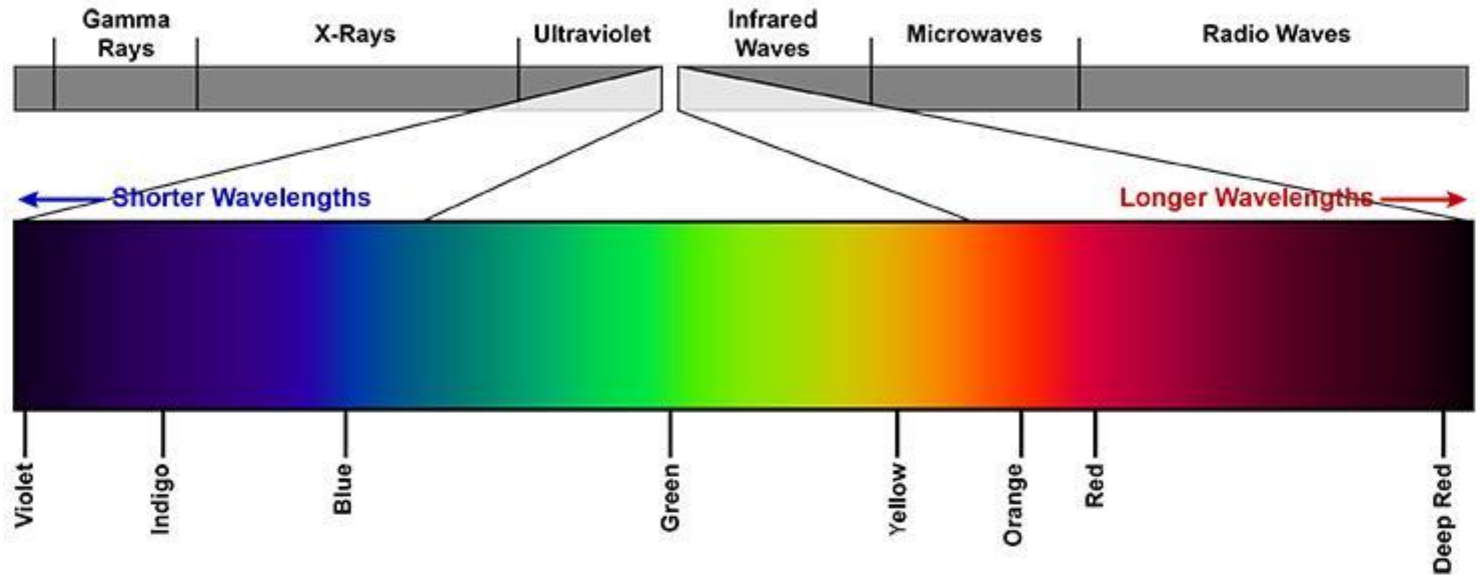
- Motivation
  - In automated image processing Color is a powerful descriptor that simplifies object identification and extraction from the scene
  - Humans can distinguish thousands of color shades and intensities. (Important in manual image analysis)
- Color image processing is divided into two major areas
  - Full color processing
  - Pseudocolor processing

Color Fundamentals

- Color spectrum seen by passing white light through a prism

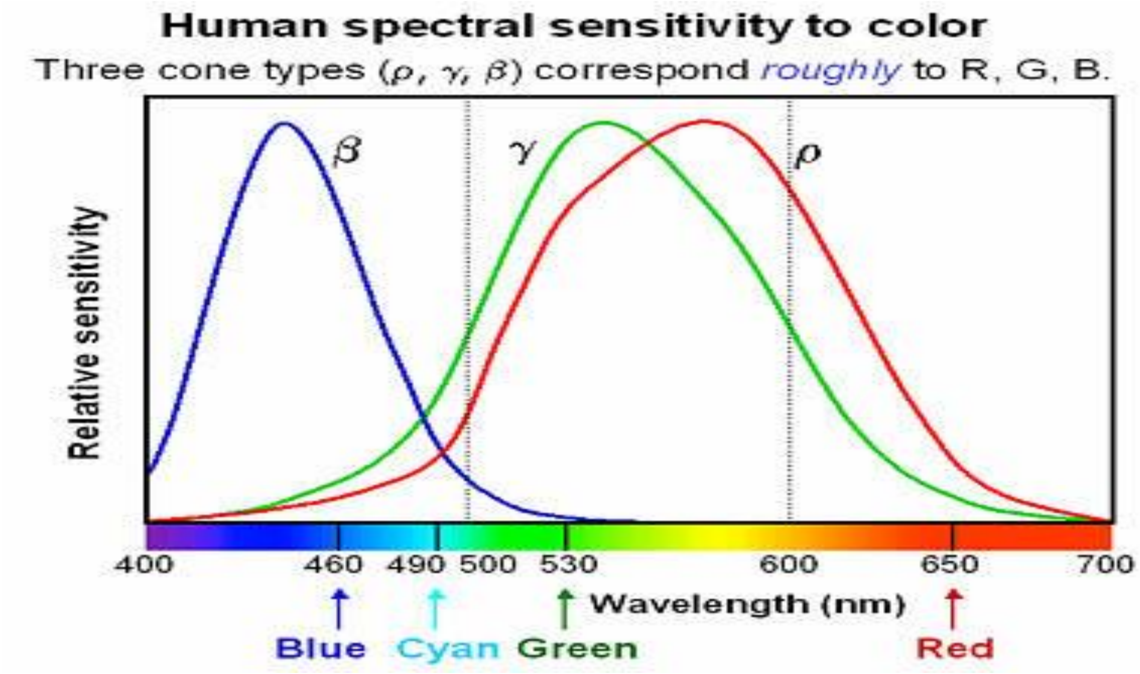
Visible range of the electromagnetic spectrum

- No color in the spectrum ends abruptly, but rather each color blends smoothly into the next color



Absorption of light as a function of wavelength

- Chromatic light spans the electromagnetic spectrum approximately 400 to 700 nm.
- Quality of light described by
  - Radiance
  - Luminance
  - Brightness



Cont..

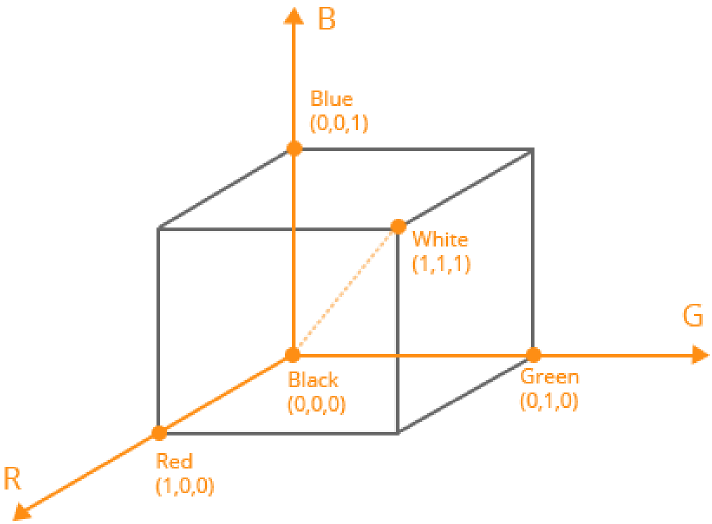
- Radiance
  - is the total amount of energy that flows from the light source
- Luminance
  - is the amount of energy an observer perceives from a light source
- Brightness
  - is the achromatic notion of intensity

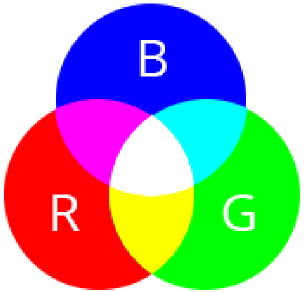
Color space model

- Purpose - is to facilitate the specification of colors in generally accepted way.
- Color model is a specification of coordinate system and a subspace within that system where each color is a represented by a single point.
- The color models are
    - RGB Color Model (Red, Green, Blue)
    - CMY and CMYK Color Model (Cyan, Magenta, Yellow, K - Black)
    - HSI Color Model (Hue, Saturation, Intensity)

RGB Color Model

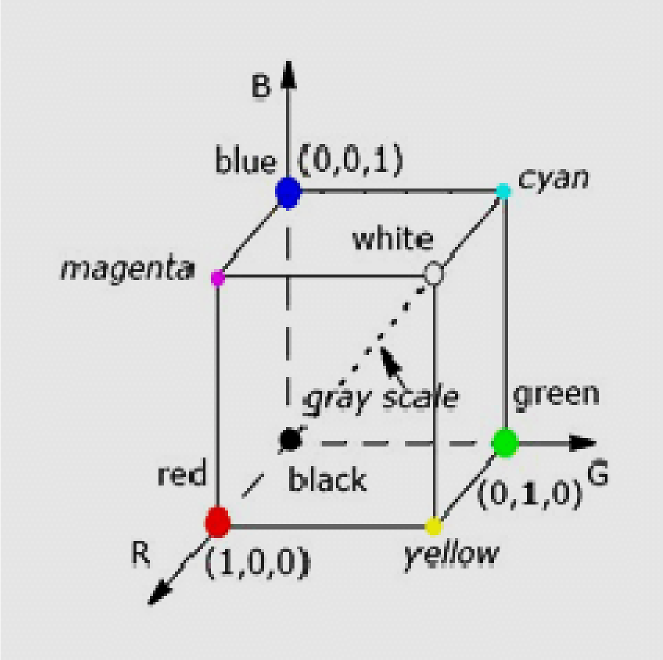
- Any arbitrary color must be inside the color cube (0,0,0) to (1,1,1)
- Most commonly used hardware oriented model
- Application - Color monitors, Color video cameras, TV





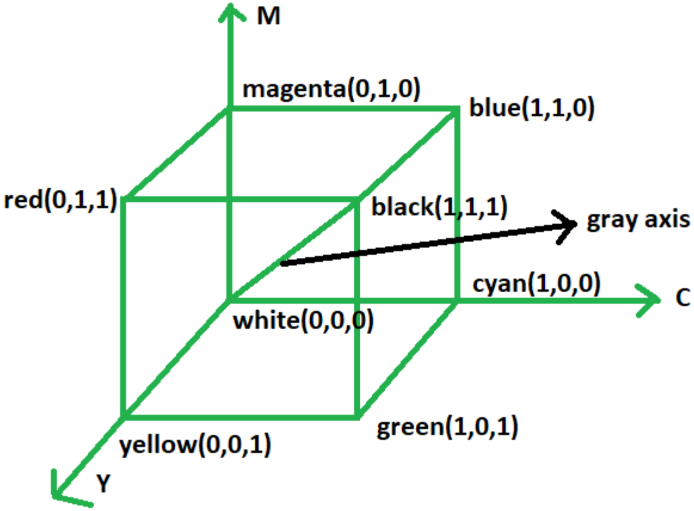
Cont..

- Color Image I_c is defined as
  $I_c : f_c(x,y) = (f_r(x,y), f_g(x,y), f_b(x,y))$



CMY and CMYK model

- These models are especially for generating hard copy output
- Application - color printers, color copiers
- Cyan, Magenta and Yellow are the primary colors of pigments. (A pigment is a material that changes the color of reflected or transmitted light as the result of wavelength-selective absorption.)



Cont..

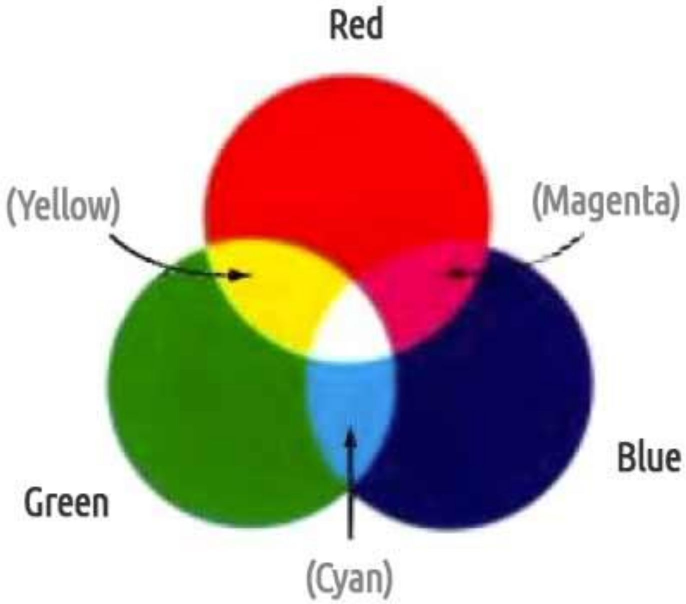
- Color printers and copiers deposit colored pigments on the paper
- So it requires CMY data input or internally convert RGB to CMY model
- Conversion is performed by the following operation

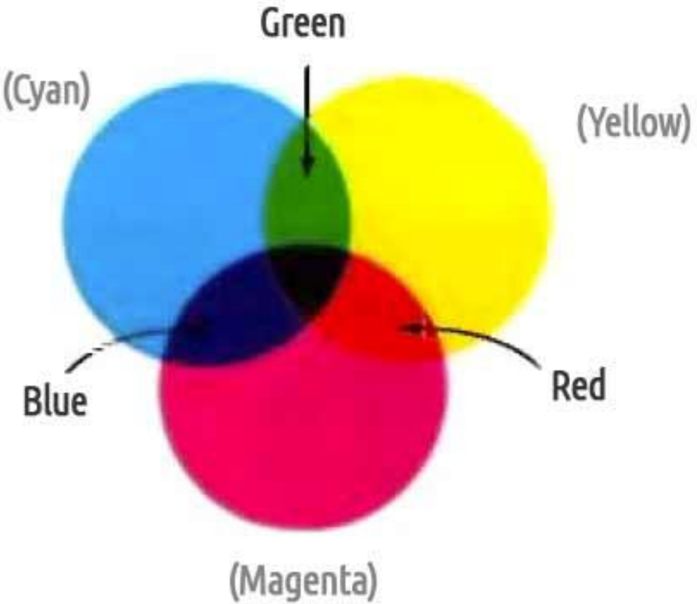
$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

CMYK Color Model

- Motivation
  - Combining CMY colors produces a muddy-looking black for printing purpose
- In order to produce pure black in printing a fourth color black is added (Black is the predominant color in printing)

RGB to CMYK Conversion





Cont..

Python code:

```
from PIL import Image
image = Image.open('img22.bmp')
cmyk_image = image.convert('CMYK')
cmyk_image.save("your_file.jpeg")
```

Cont..

Input

Output

Pixel value of RGB image at the location (0,1)
[157 165 214   0]

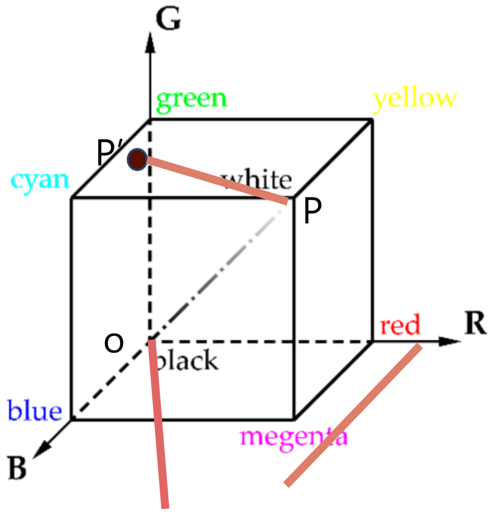
Pixel value of CMYK image at the location (0,1)
[98 90 41]

HSI model

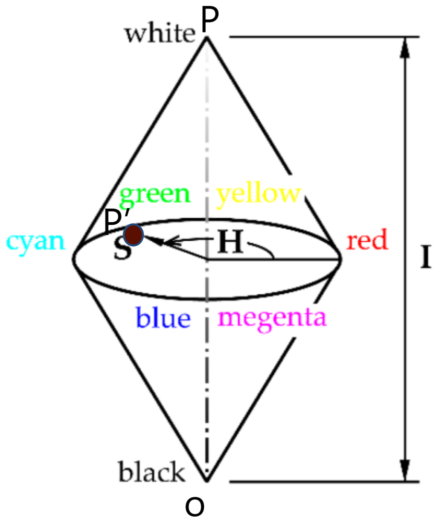
- The color model which closely with the way humans describe and interpret color
- Advantage
  - It decouples the color and grayscale information in an image
- Application - grayscale processing, human interpretation

Cont..

- The angle between the plane OPP′ and red axis is defined as Hue
- Saturation for the point P′ is the distance between the grayscale axis and the point P′



(a)



(b)

RGB to HSI Conversion

Given an RGB image the following equation is used to convert to HSI Image

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

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

if $B \leq G$

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

else, $H = 360 - H$

HSI to RGB Conversion

RG sector: $0 \le H < 120$

$$R = I\left[1 + \frac{S\cos H}{\cos(60^0 - H)}\right]$$

$$B = I(1 - S)$$

$$G = 1 - (R + B)$$

BR sector: $240 \le H \le 360$

$$H = H - 240$$

$$B = I\left[1 + \frac{S\cos H}{\cos(60^0 - H)}\right]$$

$$G = I(1 - S)$$

$$R = 1 - (G + B)$$

GB sector: $120 \le H < 240$

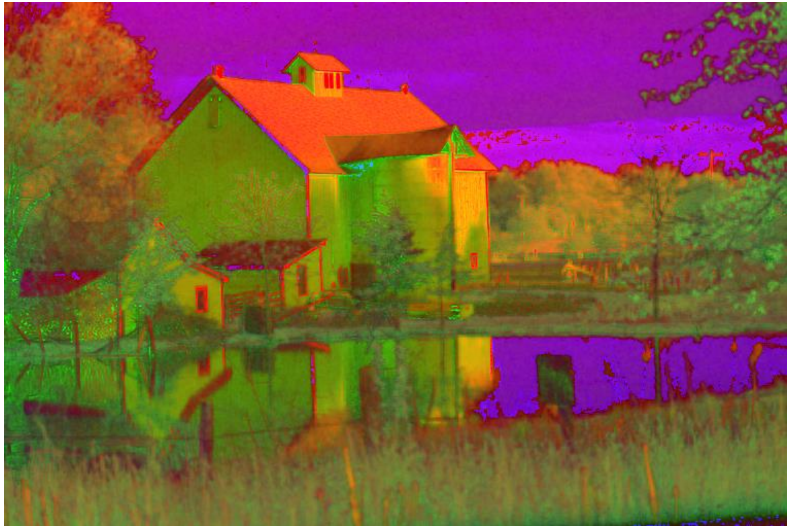
$$H = H - 120$$

$$R = I(1 - S)$$

$$G = I\left[1 + \frac{S\cos H}{\cos(60^0 - H)}\right]$$

$$B = 1 - (R + G)$$

RGB to HSI Conversion



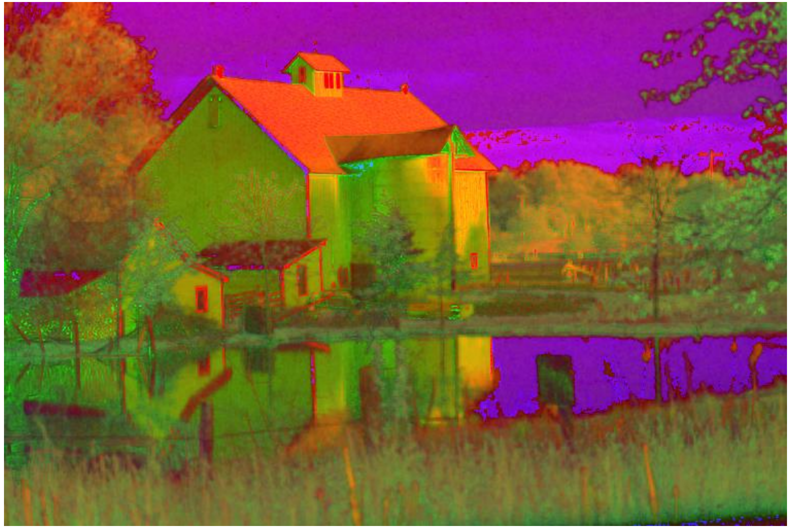
Input image (RGB)

Output image (HSI)

Cont..

```
from skimage.color import rgb2hsv
import cv2
rgb_img = Image.open('img20.bmp')
hsv_img = rgb2hsv(rgb_img)
cv2.imshow("df",hsv_img)
cv2.waitKey()
cv2.destroyAllWindows()
```

HSI to RGB Conversion



Input image (HSI)
image (RGB)

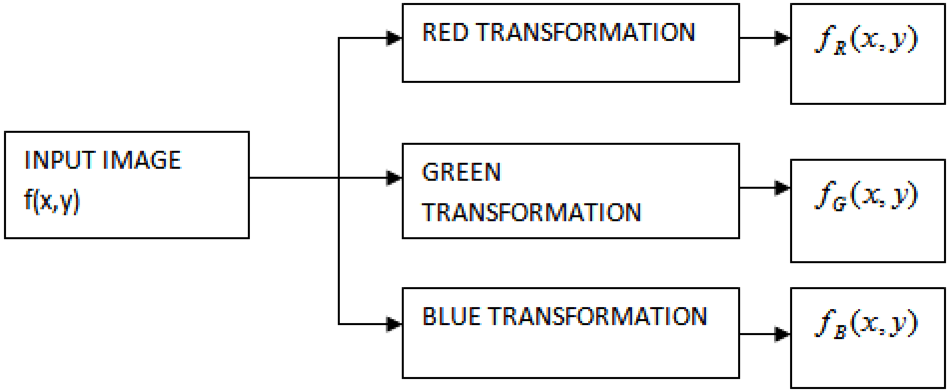
Output

HSI to RGB Conversion

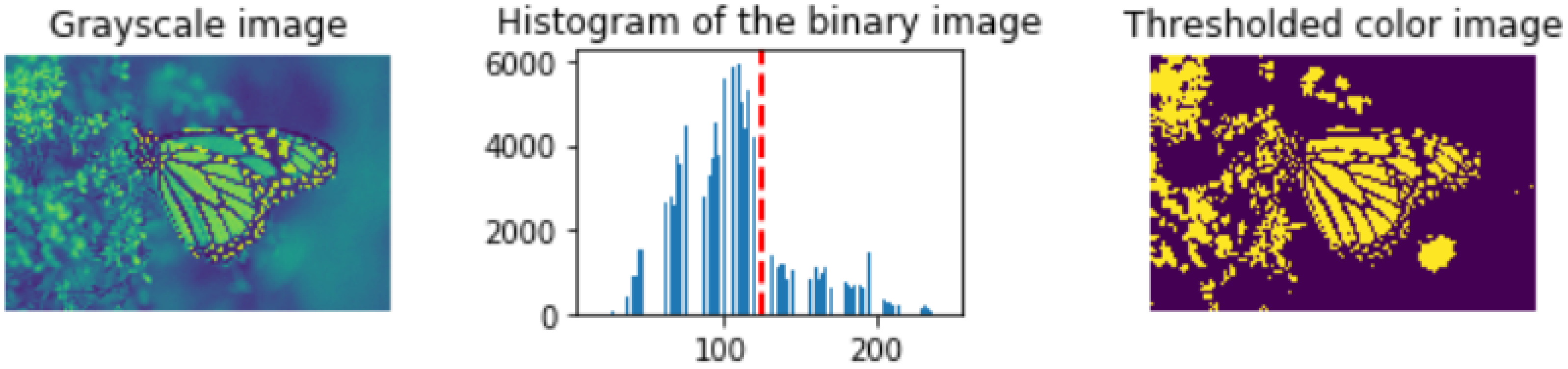
```
from skimage.color import hsv2rgb
image = Image.open('one.jpg')
rgb_image = hsv2rgb(image)
import matplotlib
matplotlib.image.imsave('back.jpg',rgb_image)
```

Pseudocolor Image Processing

- Pseudocolor image processing consists of assigning colors to grayscale based on specified criterion
- Input is Grayscale Image and Output is Color Image



Cont..



Input image

Output image

Cont..

```
import matplotlib.pyplot as plt
import cv2
binary_img = cv2.imread('img22.bmp',0)
new_img = binary_img > 125
fig, (ax2, ax0, ax1) = plt.subplots(ncols=3, figsize=(8, 2))
ax2.imshow(binary_img)
ax2.set_title("Grayscale image")
ax2.axis('off')
```

Cont..

```
ax0.hist(binary_img.ravel(), 512)
ax0.set_title("Histogram of the binary image")
ax0.axvline(x=125, color='r', linestyle='dashed', linewidth=2)
ax1.imshow(new_img)
ax1.set_title("Thresholded color image")
ax1.axis('off')
fig.tight_layout()
```