QUANSER
INNOVATE · EDUCATE

Concept Review

Image Colour Spaces

Why learn about Colour Spaces?

When detecting objects of interest, being able to pick out colors from an image or video is often the first step. Color selection can be used in a variety of applications such as line following, object avoidance, segmentation of images, etc. Color spaces are mathematical models for representing the notion of color in the digital world. Different models have been developed over the years as different infrastructures demanded different solutions. This document will start by explaining how cameras store information about color, and then will go over two popular color spaces as well as explain how greyscale and bitonal image work.

Images as Matrices

Images captured by cameras can be represented in the form of matrices. Each position in the matrix corresponds to a pixel in the image plane. Rows and columns represent all the pixels in two dimensions as shown in Figure 1. Each pixel can then be indexed using a (row, column) tuple.

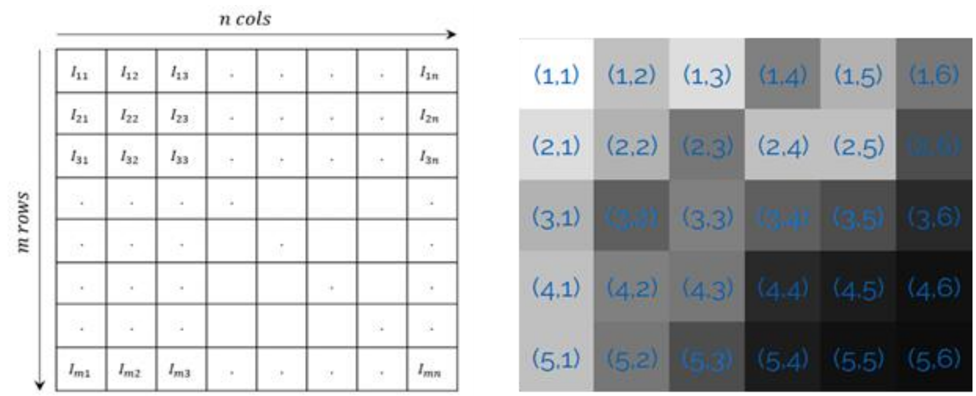

Figure 1. Spread of pixels on the image plane (left) and tuple indexing (right)

At a location in the matrix specified by the (row, col) index tuple, some valuable information such as color or intensity of light can be specified along a third dimension, also referred to as a **channel**. For example, consider a grayscale image similar to that in Figure 1 with a resolution of 5 x 6. Any given pixel would have an integer intensity $I_{i,j}$ where intensity ranges from 0 (specifying black) up to 255 (specifying white). This can be seen in Figure 1 with tuple (5,6) (black) and tuple (1,1) (white). The intensity values represent an 8-bit unsigned integer value range. Values within the range specify varying intensities of gray. This corresponds to a single channel, or overall, an m rows x n cols x 1 channel **[m,n,1]** image. Representing more colors requires more channels, as we will see next.

Data Type and Resolution

Depending on the implementation, a color in one of the color spaces either Red Green Blue (RGB) or Hue Saturation Value (HSV) can be done differently. Common implementations for RGB use values from 0 to 255 or 0 to 1. In this document RGB intensities all fall in the range [0, 255]. When looking at Hue Saturation and Value (HSV), Hue intensity values typically fall in the range [0, 360] or [0,1], and saturation and value intensity values fall in the range of [0, 255] or [0,1]. In this document, hue will be in the range of 0 to 360 and saturation and value will be reported as percentages (implementation independent).

Red Green Blue (RGB)

The **RGB** color space, which is the one used for most images, uses the red, green and blue color primaries to describe the spectrum of colors in an image. In this case, we continue to use the first 2 dimensions to represent the pixel position, but use a third dimension with 3 channels to allow the specification of the red, green and blue intensities. This would correspond to a 3 channel, or over all, an **[m,n,3]** image, as shown in Figure 2.

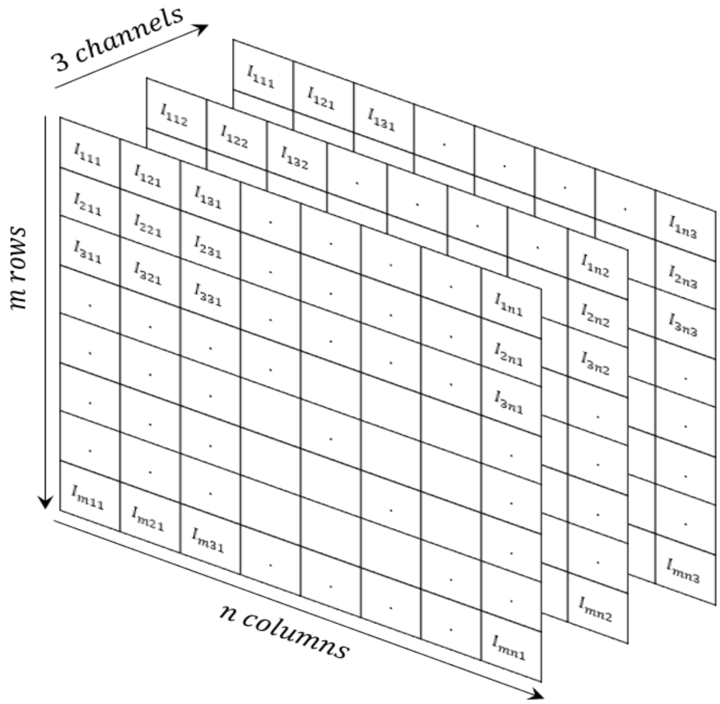

Figure 2. RGB images as 3D matrices

The red, green and blue intensities also vary between 0 and 255 each, with intermediary values specifying varying intensities of that particular primary color. For example, a bright pure red would correspond to an intensity of 255 in the red channel/plane, and 0 in the green and blue channels, and vice versa for a bright green or blue. Mixing intensities in different channels creates additive colors. For example, white corresponds to a 255 in all the channels and black corresponds to all zeros. Red and green intensities at 255 and blue at 0 would correspond to a yellow. This can be observed in Figure 3.

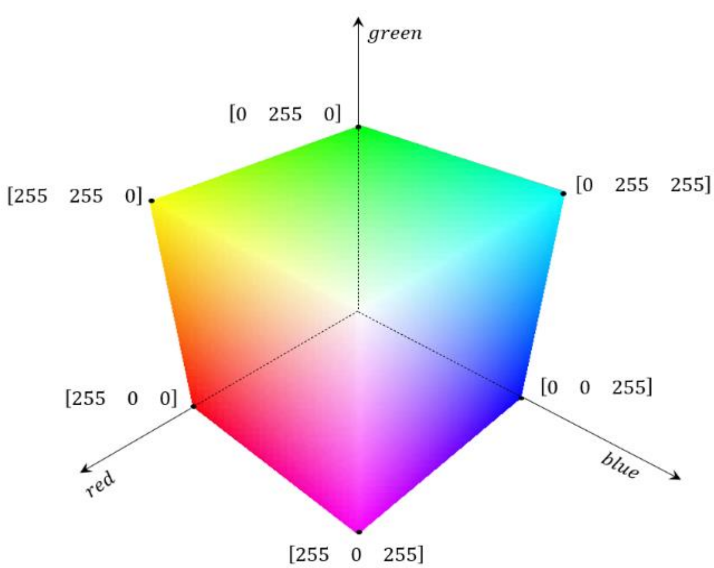

Figure 3. RGB primary colors

Figure 4 shows a picture of various objects as well as the 3 individual red, blue and green planes. Note that the individual planes appear grayscale as they only have a single channel. Lighter pixels in each plane mean more of that color on that specific pixel. For example, the blue object on the bottom left becomes almost black on the red plane since there is no red there, but the yellow objects are lighter since yellow contains a red and green element in RGB space.

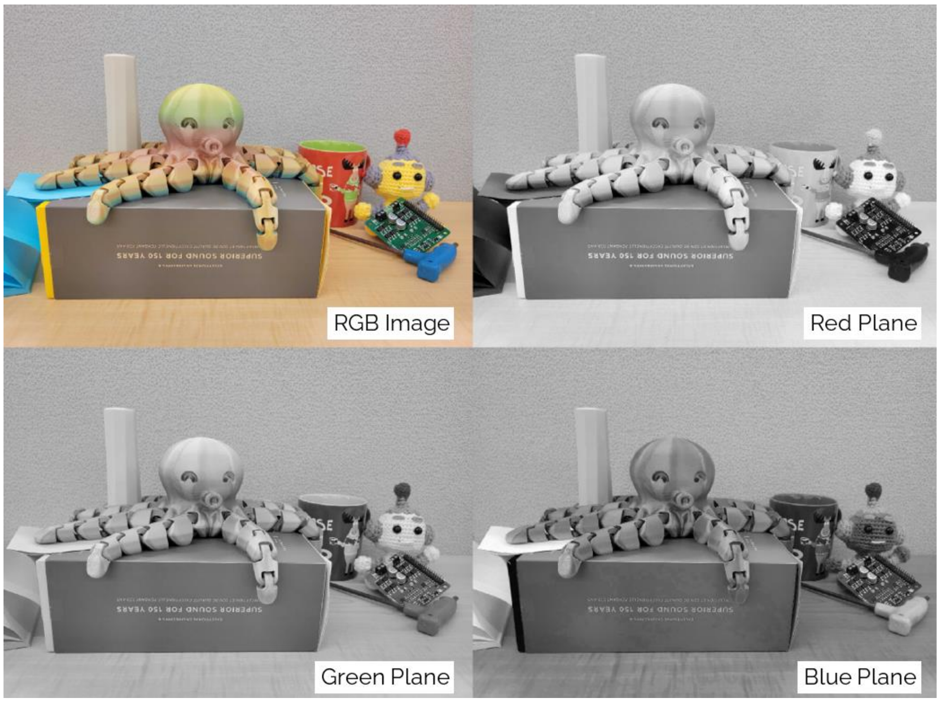

Figure 4. RGB Image and individual color planes

The RGB color space, although intuitive, has some downsides, since it combines luma (or light intensity) with chroma (or color intensity) which does not describe observed colors under different viewing conditions well. A red stop sign in daylight may correspond to a RGB intensity of [255, 0, 0] but at night, may correspond to [50, 0, 0], even though the color is still red. In a real-world example in Figure 5, note the same do not enter sign under different lighting conditions and the varying RGB intensities.

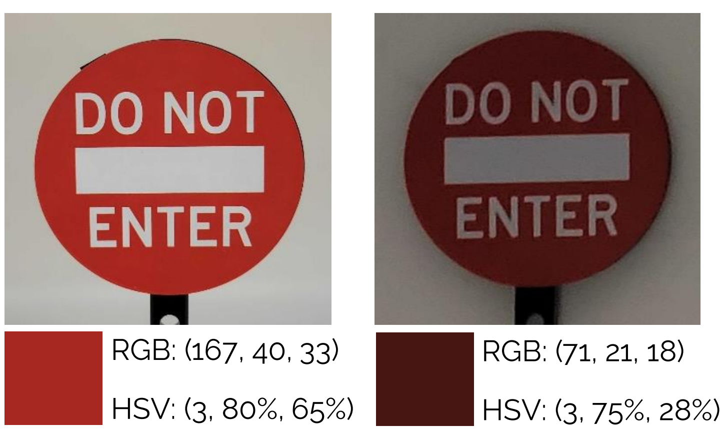

Figure 5. Road sign under different lighting conditions

Hue Saturation Value (HSV)

The **HSV** space distinguishes the luma (light intensity information) from the chroma (color information). Taking a cross-section of the color cylinder shown in Figure 6 (right) portrays hue representation shown in Figure 6 (left). **Hue** is a color wheel which ranges from red at 0˚, to green at 120˚, blue at 240˚ and back to red at 360˚. Hue parameter of an object remains stationary regardless of the light intensities. Figure 5 shows that a road sign under different lighting conditions has the same hue value. What varies, is the saturation and value.

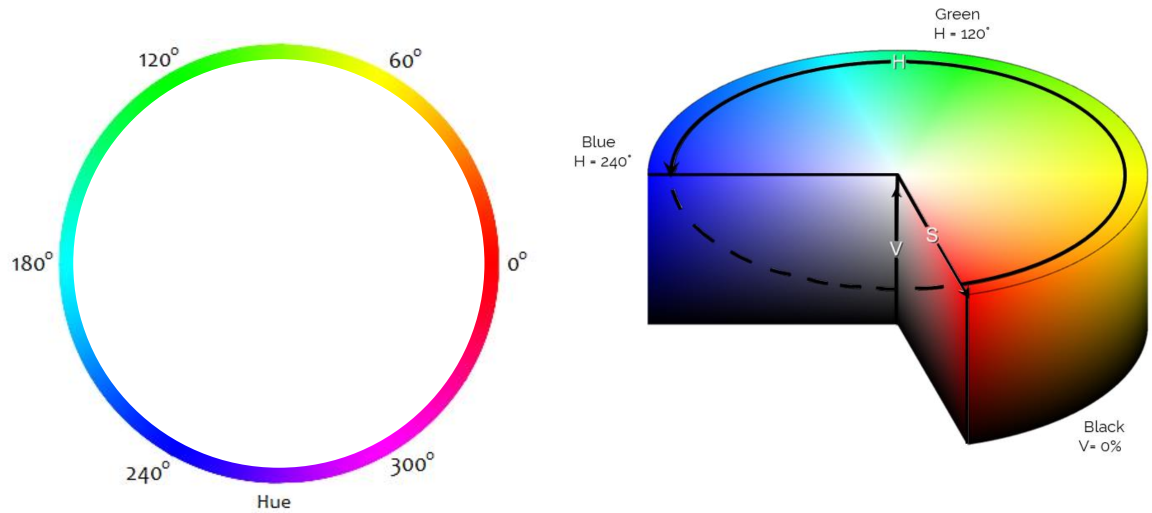

Figure 6: Hue color wheel (left) and HSV (right)

Saturation represents the amount of gray in a pixel. For example, consider an RGB color [100, 100, 100]. This represents a medium to dark gray, with equal red, green and blue intensities. This would correspond to any hue value and a saturation of 0 (center of the cylinder in Figure 6). As long as the three RGB intensities match, the output of the color will be a shade of grey; as each color intensity deviates from each other, the color saturation becomes larger, up to 100%. Lastly, the **Value** represents the overall lightness of the pixel. If the saturation is zero, the value will describe how dark the grey is, the lower the value, the darker the color. So the value plane by itself looks the same as a grayscale image, the darker the object, the darker the pixel.

Note: RGB values can be converted to HSV values using the rgb2hsv() function in MATLAB and the cvtColor() method in opencv for Python.

Grayscale

Various formulas can be used to convert RGB images to a grayscale format. A weighted sum of each layer (or color channel) is a common approach,

$$I_{grey} = \frac{I_r + I_g + I_b}{3}$$

However, since our eyes have most sensitivity to green and least to blue, we can use the following equation (which changes the weights of each color channel), to better represent the intensity perception of typical human eyes to primary colors.

$$I_{gray} = 0.2126 I_r + 0.7152 I_g + 0.0722 I_b$$

Since value in HSV represents the lightness of a pixel, we can obtain a greyscale image by just using the value plane itself.

Bitonal

Bitonal or binary images have binary intensities being high (1 or 255) and low (0). These are the result of typical thresholding algorithms. They can be used masks to highlight different regions of images by thresholding for a specific range of pixel values.

Quanser Inc.
119 Spy Court
Markham, Ontario
L3R 5H6
Canada

info@quanser.com
Phone: 19059403575
Fax: 19059403576
Printed in Markham, Ontario.

For more information on the solutions Quanser Inc. offers, please visit the web site at:
http://www.quanser.com