Spectral Imaging

Objectives

1. Render image color given spectral information
2. Find hyperspectral image databases
3. Demonstrate how color changes under varying illumination.

Key Take-aways

- Color information is affected by light source, object spectral property, and sensor sensitivity.
- Spectral information is a more stable descriptor of an object.
- There is a variety of hyperspectral databases available.
- Converting spectral information into color is necessary for many reasons.

Trinity of color

The color detected by a camera is dependent on three factors: (1) the light source with spectral power distribution $P(\lambda)$, (2) the reflectance of the object $R(\lambda)$, and (3) the spectral sensitivity of the sensor sensitivity $S_n(\lambda)$ where n = 1…N are the N color channels of the camera. These three are known as the **color trinity** depicted in Figure 1.

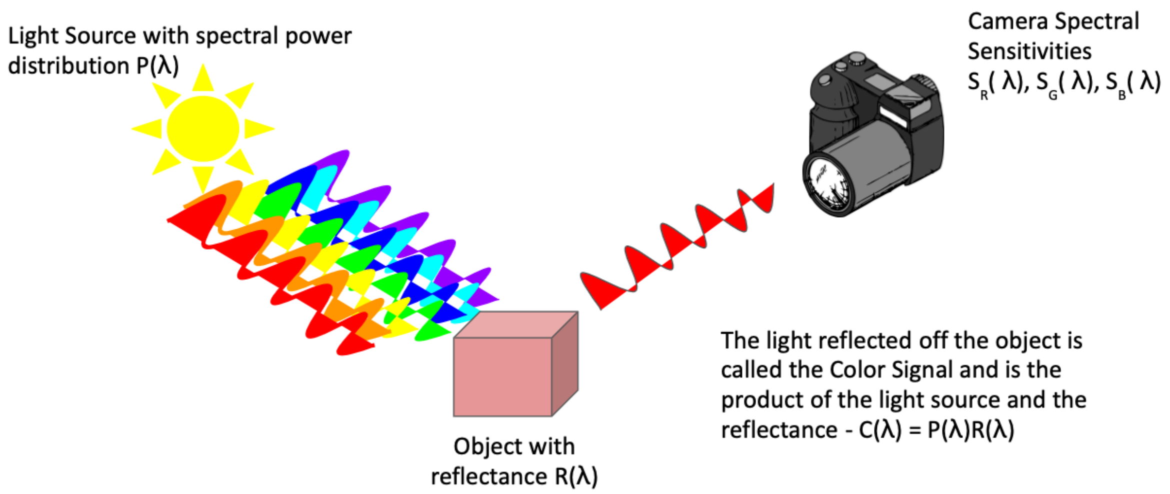


Figure 1. Color trinity

Light that reaches the sensor is called the **Color Signal** $C(\lambda)$ and is the product of $P(\lambda)$ and $R(\lambda)$. Figure 1 shows the case for a color camera with 3 color channels, Red, Green and Blue, (R,G,B). A color camera's spectral sensitivity is a product of its sensor's quantum efficiency and the transmittance of its red, green and blue filters.

Because of the color trinity, color information can be ambiguous. Take for example an image of an orange-looking paper - unless there are other image cues, it may be a white paper illuminated by orange light, or an orange paper illuminated by white light. On the other hand, different cameras will have different spectral sensitivities. Therefore, the same object illuminated by the same light but captured by different cameras will appear to have different color. This is the reason why color is considered an "unstable" object characteristic.

Spectral characteristics of the object such as its reflectance, transmittance, absorbance, fluorescence, or emittance, are not dependent on the light source or the sensor and therefore are preferred over color as a feature for describing the object.

However, color rendition of spectral characteristics still is necessary for a number of reasons. A hyperspectral image, because of its massive dimensionality, is difficult to visualize. By rendering it in color, it becomes easier to comprehend for when one needs to communicate findings or results. Another use is compression. Hyperspectral images may also be depicted in false color. Hyperspectral images can be reduced to multispectral images by summing the spectrum over defined bandwidths. The ratio of these bands can then be combined mathematically and rendered in false color. This leads to the third use of color rendering which is analysis. A color-rendered hyperspectral image may make it easier to find clusters, changes, or similarities.

Sensor Model

To gain insight on how cameras compress spectral information this activity will teach us how to render hyperspectral images using the sensor model.

Consider a color signal $C(\lambda)$ coming from a point on an object. When it reaches the camera sensor it will be converted into an analog signal which is modeled as

Equation 1:
$$V_n = \int_\lambda C(\lambda)S_n(\lambda)d\lambda = \int_\lambda P(\lambda)R(\lambda)S_n(\lambda)d\lambda$$

where Vn is the analog signal for the nth color channel.

The camera output should be normalized such that white objects look white in a step known as white-balancing. In digital color white or gray objects will have the same value of DN in the red, green and blue channels. The analog signal in Equation 1 is normalized by the analog signal of a white object. Recall that a white object will have a reflectance equal to 1.0 across all wavelengths. Thus the white balanced analog signal is now

Equation 2: $$\overline{V_n} = \frac{\int_\lambda P(\lambda)R(\lambda)S_n(\lambda)d\lambda}{\int_\lambda P(\lambda)S_n(\lambda)}$$

Notice that if the camera points to a white object the value of Equation 2 will be 1 for all channels. This signal will then be discretized by and ADC to a digital number DN_n. For an 8-bit camera, the DN range is 0-255. Cameras for satellites or microscopes will typically have DN's with 10 to 12 bits range. Smartphones that can save in raw format will typically save in 10 bits.

Spectral Databases

Here is a sample list of light source, object reflectance, and camera sensitivity databases we can use for this activity. Feel free to use other databases you may find.

Light source

- CIE Illuminats A and D65 : http://www.rit-mcsl.org/UsefulData/D65_and_A.xls
- CIE Fluorescent lamps : http://www.rit-mcsl.org/UsefulData/Fluorescents.xls
- https://lspdd.org/app/en/home

Reflectance

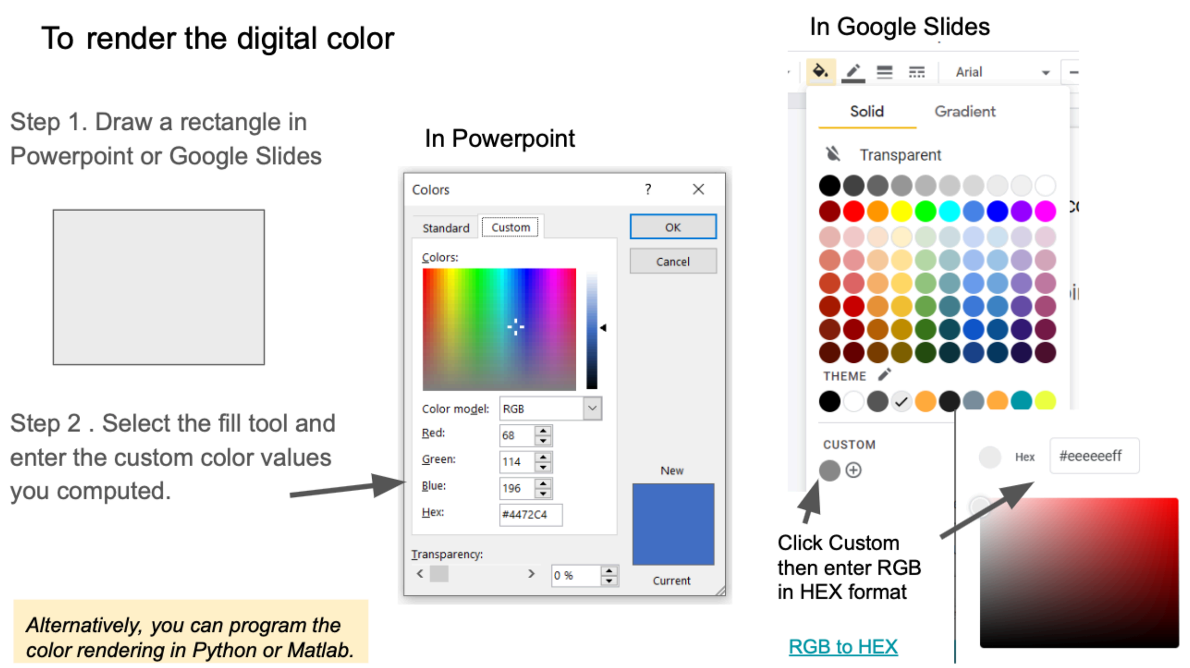
- Macbeth color checker : http://www.rit-mcsl.org/UsefulData/MacbethColorChecker.xls
- Landscapes, faces, fruits, and charts : https://exhibits.stanford.edu/data/browse/iset-hyperspectral-image-database
- Hyperspectral Imager for the Coastal Ocean (HICO) : https://oceancolor.gsfc.nasa.gov/data/hico/

Camera Sensitivity

- https://nae-lab.org/~rei/research/cs/zhao/database.html

Activity Procedure

1. Practice rendering the digital color of an object by using the spectral reflectance of a Macbeth color checker chart. Select a light source and camera from the database and compute Equation 2 for each color patch. Firstly, “align” the dataset by interpolating them such that they have the same wavelength range (for example, 400 nm to 700nm) and same $d\lambda$ (for example, 5nm or 10nm). This means, interpolate the data such that, for example, their reflectance values are defined from 400 nm to 700 nm in steps of 10nm.
2. Once aligned, Equation 2 becomes a simple matrix element-to-element multiplication and a summation. Since the result of Equation 2 will be a number from 0 to 1, discretize to 8 bit by multiplying by 255 and rounding the numbers to the nearest integer.
3. Render the patches using Google Slides, Powerpoint, or by creating an image in Matlab (or Python).



4. Render the same patches under different light sources and comment if the difference in color is perceptible.
5. Once you are able to render a patch, try to render a scene. Use any sample from the hyperspectral databases available.

Due in 1 week (Mar 9, 2022)