Color Processing

Objectives

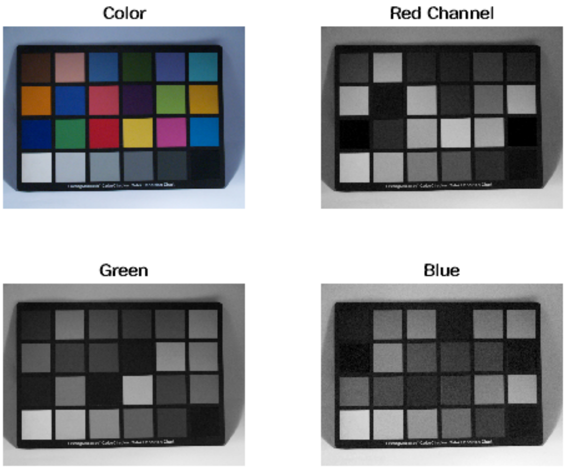
1. Use digital color histograms to segment regions of interest in an image.
2. Demonstrate how color changes under varying illumination.

Digital Color Image

Color imaging devices, such as digital cameras and displays, exploit the *trichromaticity* of human vision. Any color that we see can be matched by the superposition of three primary lights combined in a certain proportion. This is why true color digital images are saved in three primary color channels, Red, Green and Blue. Taken individually, each color channel looks like a grayscale image with the brightness of a pixel indicating the contribution of that channel to the color of the image.

The Matlab script below opens an image of a Macbeth Colorchecker Chart and displays its RGB channels. Notice how dominantly red patches have brighter pixels in the red channel, bluish patches are bright in the blue channel, etc. White and shades of gray have red, green and blue channels in equal proportion so the bottom row of patches appear equally bright in all channels.

```
I = imread("Tan Macbeth Chart.JPG");
R = I(:,:,1);
G = I(:,:,2);
B = I(:,:,3);
figure (1);
subplot(2,2,1); imshow(I);title("Color");
subplot(2,2,2); imshow(R);title("Red Channel");
subplot(2,2,3); imshow(G);title("Green");
subplot(2,2,4); imshow(B);title("Blue");
```



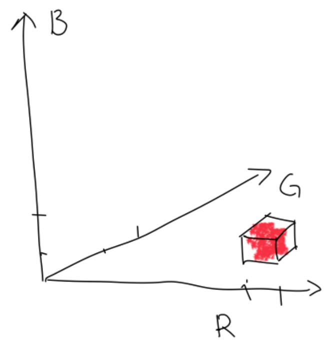
Color Segmentation 3 Ways

In segmentation we want to pick out a region of interest (ROI) in a scene. Through image processing, segmentation can be done automatically for when we have a large number of images that we need to go through. For example, you need to examine images of red blood cells to estimate the severity of malarial infection in a patient. This is done by counting how many cells have the malaria parasite over the total number of cells observed in 200 image frames. Too tedious to do by hand but that's what parasitologists do.

If the ROI has a distinct color we can use its color distribution for automatic segmentation. Here we explore three ways of color segmentation in increasing order of sophistication. The best one is the one that works for your application so don't discount the simplest one just yet. The first step is to determine the color distribution of the object of interest. All three techniques will use a color histogram in different ways.

Thresholding

Here we simply examine the ROI histogram of in color channel and note the range of digital number the histograms are within. We then use logical AND operators to determine if the pixel is within the range of the ROI color histogram. Just like in the figure below, it's as if we model the color of the ROI by a cube.   Let's open your image.




```
%Read  Image
    [file, path] = uigetfile('*.jpg');
    filename =[path,file];
    I = imread(filename);
    figure(1);imshow(I);
```

Next let's crop out the colored region you want to segment.

```
%Crop region of interest and note the range
  J = imcrop(I);
  figure(2);
  subplot(1,3,1); imhist(J(:,:,1));
  subplot(1,3,2); imhist(J(:,:,2));
  subplot(1,3,3); imhist(J(:,:,3));
```

Take note of the edges of the histogram

```
% For red patch in Macbeth Chart: 150<R166, 17<G<34, 24<B<62
       R = I(:,:,1); G = I(:,:,2); B = I(:,:,3);
       BW = (R >150) & (R<166);
       figure(3);
       imshow (BW); % is it enough?
       BW = (R >150) & (R<166) & (G>17) & (G<34) & (B>24) & (B<62);
       imshow(BW2)Parametric Segmentation
```

Parametric Segmentation

Here we model the ROI color histogram by a distribution function. For example, we can use the multivariate Gaussian distribution function given by

$$p(\mathbf{x};\mu,\mathbf{\Sigma}) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)\right)$$

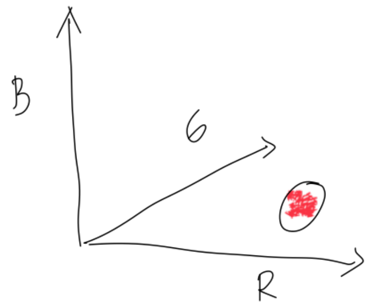
where

$\mathbf{x}$: [R,G,B] value of the pixel under test,

μ : mean [R,G,B] of the region of interest, and

$\mathbf{\Sigma}$: covariance matrix of the [R,G,B] of the region of interest.

Once we've cropped the region of interest we compute the mean and covariance matrix of the R, G, B of it. Per pixel in the original scene, we compute the probability that it is similar to the region of interest. We then display the probability pixel-per-pixel. The higher the probability, the brighter the pixel will look. In this case we are modeling the color distribution as an ellipsoid.



We may also do a joint 1-D Gaussian distribution per channel and multiply the probabilities per channel point-per-point. In this way you can easily increase the standard deviation up to 5 times to include 99% of all the colors in the ROI histogram.

Non-Parametric Segmentation

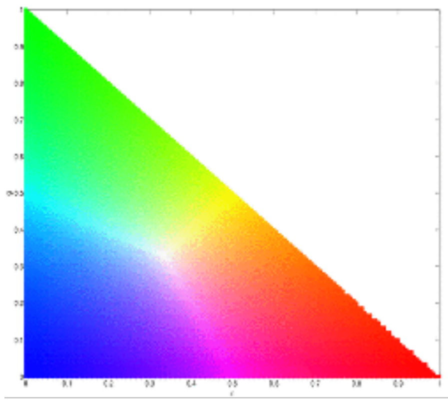
The earlier two techniques work well if the color of the ROI object is uniform. But what if the ROI is multi-colored? Sure we can do multiple thresholds or Gaussian mixture models, but an easier way will be to not make a model of the histogram at all. Instead we use it as is. The technique is known as *Histogram Backprojection*.

Because a 3D color histogram is actually 4D (4th D is the number of pixels), it is better to represent color space not by the RGB but by one that can separate brightness and chromaticity (pure color) information. One such color space is the ***normalized chromaticity coordinates*** or NCC.

Per pixel, let I = R+G+B. Then the normalized chromaticity coordinates are

$$r = R/I; g = G/I; b = B/I$$

We note that r+g+b = 1 which implies r,g and b can only have values between 1 and 0 and b is dependent on r and g since b = 1-r-g. Therefore, it is enough to represent chromaticity by just two coordinates, r and g. Thus, from R G B, the color space has been transformed to r g I where chromatic information is in r and g while brightness information is in I. We have thus reduced color information from 3 dimensions to 2. Below is the r-g chromaticity space (x-axis is r, y-axis is g).



In histogram backprojection a pixel is given a value equal to its histogram value in chromaticity space. This has the advantage of faster processing because no computations are needed, just a look-up of histogram values.
Obtaining a 2D histogram can be implemented rapidly by converting the r,g values into integers and binning the image values in a matrix. The matrix can then be concatenated into a look-up table with the indices being a function of the bin indices for r and g.

Procedure

1. Select images for color segmentation. Make sure the ROI in each is distinctly colored.
2. Use all three color segmentation techniques and report on how well they perform. Explore their strengths and limitations.

Due in 1 week (Mar 2, 2022)