Color

Examples and additional material
Contrast Enhancement for Color

• Recall (from lecture 8) that we can do “gamma correction” to enhance the contrast of an image.
For grayscale images, this works great. How about color images?
clear all
close all

RGB = imread('colorful-flower.jpg');
figure, imshow(RGB), title('Input');

% Split into R,G,B bands.
R = RGB(:,:,1);
G = RGB(:,:,2);
B = RGB(:,:,3);

% Show histograms.
figure,
subplot(1,3,1), imhist(R), title('R');
subplot(1,3,2), imhist(G), title('G');
subplot(1,3,3), imhist(B), title('B');
Apply gamma correction to R,G,B

• Note distortion of the colors

% Apply Gamma correction.
gamma = 0.2;
RGB1(:,:,1) = uint8(255*power(double(R)./255, gamma));
RGB1(:,:,2) = uint8(255*power(double(G)./255, gamma));
RGB1(:,:,3) = uint8(255*power(double(B)./255, gamma));

% Show the output.
figure, imshow(RGB1), title('Gamma corrected on R,G,B');
Convert to HSV, apply to value

- Convert to hue, saturation, value (intensity)
- Apply gamma correction to intensity only

% Split into H,S,V bands. Output is double, 0..1.
HSV = rgb2hsv(RGB);
H = HSV(:, :, 1); S = HSV(:, :, 2); V = HSV(:, :, 3);

% Apply Gamma correction on intensity (value) only.
HSV2(:, :, 1) = H;
HSV2(:, :, 2) = S;
HSV2(:, :, 3) = power(double(V), gamma);

% Go back to RGB.
RGB2 = im2uint8(hsv2rgb(HSV2));

% Show the output.
figure, imshow(RGB2), title('Gamma corrected on intensity only');
Indexed Storage of Color Images

- If an image uses 8-bits for each of R,G,B, there are $2^{24}$ possible colors
- Most images don’t use the entire color space of possible values – we can get by with fewer
Indexed Storage of Color Images

- For each image, we find a set of colors that are a good approximation of the entire set of pixels in the image; and put those into a colormap.
- Then for each pixel, we just store the indices into the colormap.

Image of indices (0..63)

Color image, using 64 colors
Indexed storage of Color Images

• Use a colormap
  – Image \( f(x,y) \) stores indices into a lookup table (colormap)
  – Colormap specifies RGB for each index

>> cmap(1:20,:)
ans =
0.0980    0.0941    0.1020
0.1451    0.1020    0.1098
0.1608    0.1412    0.1804
0.2196    0.1216    0.1020
0.2431    0.1569    0.1373
0.2196    0.1843    0.2118
0.2471    0.2353    0.2824
0.3137    0.1490    0.1020
0.3294    0.2039    0.1569
0.4118    0.1725    0.0902
0.4235    0.2314    0.1373
0.3176    0.2471    0.2431
0.4039    0.2784    0.2118
0.4078    0.3137    0.2627
0.3255    0.3059    0.3490
0.5176    0.2039    0.0157
0.5059    0.2275    0.0902
0.6039    0.2392    0.0471
0.6392    0.3059    0.0353
0.5098    0.2706    0.1529

Display system will display these values of RGB

Index = 17
clear all
close all

% Read an indexed color image and its colormap.
[I,cmap] = imread('forest.tif');

% Displaying just the image of indices isn't meaningful.
imshow(I, []), title('Indices into colormap');

% To display the image properly, you must provide the colormap.
figure, imshow(I, cmap), title('Indexed image with colormap');

% You can convert from an indexed color image to an RGB image.
RGB = ind2rgb(I,cmap);

% You can convert an RGB to an indexed color image.
numberofcolors = 8;    % Specify the number of colors to be used
[I2,cmap2] = rgb2ind(RGB,numberofcolors);

% This image looks ok even using very few colors!
figure, imshow(I2,cmap2);
title(sprintf('Using only %d colors', numberofcolors));
imtool(I2,[]);        % Verify that the indices are only 1..numberofcolors
disp(cmap2);        % Show the colormap