|  |  |
| --- | --- |
| **Homework #4**  *Digital Image Processing(EEE5320), 2019-2* | Due Date: 2019. 12.02 |

2018311303

YoungWoong KWON (권영웅)

1. Conversion between RGB and HSI (30pt)

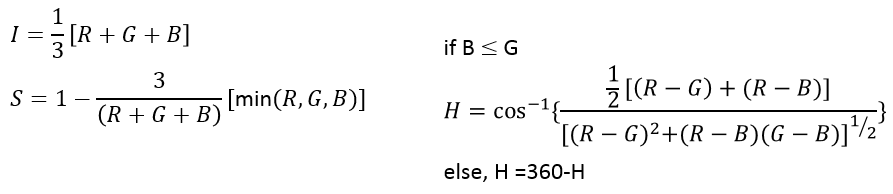


**Figure 1. pepper.png**

* 1. Read the attached “pepper.png” and please answer following questions.
  2. Implement a Matlab function for HSI conversion model

**(You should not use ‘rgb2hsi’ built-in function)**

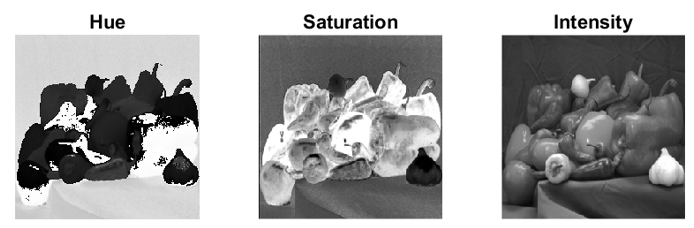
The conversion model from RGB to HSI is as follow:



Where R,B, and G are the red channel, blue channel, and green channel, respectively.

|  |
| --- |
| function output = toHSI(im\_rgb)    im\_hsi = zeros(size(im\_rgb));  [size\_w,size\_h] = size(im\_rgb(:,:,3));  sum\_rgb = sum(im\_rgb,3);    %I  im\_hsi(:,:,3) = (1/3) .\* sum\_rgb;  %S  im\_hsi(:,:,2) = ones(size(im\_rgb(:,:,2))) - 3 ./ (sum\_rgb+0.000001) .\* min(im\_rgb,[],3);  %H  for i = 1:size\_w  for j = 1:size\_h  p\_r = im\_rgb(i,j,1);  p\_g = im\_rgb(i,j,2);  p\_b = im\_rgb(i,j,3);  nume = 1/2\*((p\_r-p\_g) + (p\_r-p\_b));  denom = ((p\_r-p\_g)^2 + ((p\_r-p\_b)\*(p\_g-p\_b)))^(1/2);  h = acosd( nume / (denom+0.000001) );  if p\_g >= p\_b  im\_hsi(i,j,1) = h;  else  im\_hsi(i,j,1) = 360 - h;  end  end  end    output = im\_hsi;  end |
| **[toHSI.m]** |

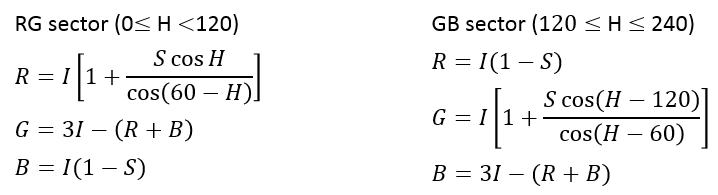
* 1. Show conversion results of RGB image with H(Hue), S(Saturation), I(intensity), respectively.



* 1. Implement a Matlab function for RGB conversion model

**(You should not use ‘hsi2rgb’ built-in function)**

The conversion model from HSI to RGB is as follow:



Where H,S, and I are the hue, saturation, and intensity, respectively.

|  |
| --- |
| function output = toRGB(im\_hsi)  im\_rgb = zeros(size(im\_hsi));  [size\_w,size\_h] = size(im\_hsi(:,:,2));    % calc RGB value  for i = 1:size\_w  for j = 1:size\_h  p\_h = im\_hsi(i,j,1);  p\_s = im\_hsi(i,j,2);  p\_i = im\_hsi(i,j,3);    if ((p\_h >= 0) && (p\_h < 120))  p\_r = p\_i\*(1+p\_s\*cosd(p\_h)/cosd(60-p\_h));  p\_b = p\_i\*(1-p\_s);  p\_g = 3\*p\_i-(p\_r + p\_b);  elseif ((p\_h >= 120) && (p\_h < 240))  p\_r = p\_i\*(1-p\_s);  p\_g = p\_i\*(1+p\_s\*cosd(p\_h-120)/cosd(p\_h-180));  p\_b = 3\*p\_i-(p\_r + p\_g);  elseif ((p\_h >= 240) && (p\_h <= 360))  p\_g = p\_i\*(1-p\_s);  p\_b = p\_i\*(1 + p\_s\*cosd(p\_h-240)/cosd(p\_h-300));  p\_r = 3\*p\_i-(p\_g + p\_b);  end    im\_rgb(i,j,1) = p\_r;  im\_rgb(i,j,2) = p\_g;  im\_rgb(i,j,3) = p\_b;  end  end  % output  output = im\_rgb;  end |
| **[toRGB.m]** |

* 1. Show conversion results of HSI image with original RGB image. 
  2. Explain your implementation and discuss your results

[ Implementation ]

1. Read image ‘pepper.png’

2. Convert the RGB image to HSI using the conversion formula

3. Check conversion result by plotting HSI image

4. Convert the HSI image to RGB using the conversion formula

5. Check that the image are the same before and after conversion

6. Check the results and plotted image

[ Discussion ]

- The original image and the HSI->RGB converted image are the same, indicating that no information loss occurs during the conversion process.

- To avoid the problem of ‘dividing by zero’ in the HSI conversion process, a small value was added to the denominator.

|  |
| --- |
| % image read  im = im2double(imread('pepper.png'));    % 1. RGB to HSI conversion  im\_hsi = toHSI(im);  % plot  figure(1);  subplot(1,3,1);  imshow(im\_hsi(:,:,1)./360);  title('Hue');  subplot(1,3,2);  imshow(im\_hsi(:,:,2));  title('Saturation');  subplot(1,3,3);  imshow(im\_hsi(:,:,3));  title('Intensity');    % 2. HSI to RGB conversion  im\_rgb2 = toRGB(im\_hsi);  figure(2);  subplot(1,2,1);  imshow(im);  title('Original Image');  subplot(1,2,2);  imshow(im\_rgb2);  title('Converted from HSI image'); |
| **[hw4\_1.m]** |

1. Pseudocolor Image Processing (30pt)



**Figure 2. cameraman.png**

* 1. Read the attached “cameraman.png” and please answer following questions.
  2. Implement pseudo coloring process with following criterion.

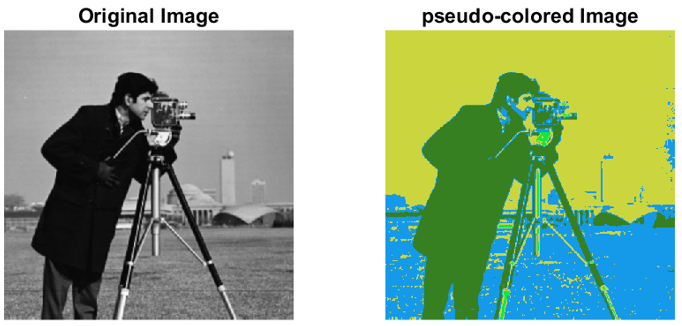
**(You should not use ‘imGray2Pseudocolor’ built-in function)**

|  |  |
| --- | --- |
| Gray value | RGB values |
| Input >=0 && Input < 72 | R = 54  G = 128  B = 33 |
| Input >=72 && Input < 144 | R = 21  G = 154  B = 233 |
| Input >=144 && Input < 216 | R = 203  G = 213  B = 62 |
| Input >=216 && Input < 255 | R = 16  G = 233  B = 59 |

Where input, R, B, and G are the intensity of input image, red channel, blue channel, and green channel, respectively.

|  |
| --- |
| % image read  im = im2double(imread('cameraman.png'));  % change scale to maximum 255  im = rgb2gray(im) .\* 255;    % Pseudo coloring  [size\_w,size\_h] = size(im);  pc\_im = zeros([size\_w size\_h 3]);  pc\_rgb = [54 21 203 16;128 154 213 233;33 233 62 59];    for i = 1:size\_w  for j = 1:size\_h  % get intensity from the pixel of gray image  intensity = fix(im(i,j)/72);  switch intensity  % set color according to intensity value  case 0  pc\_im(i,j,:) = pc\_rgb(:,1);  case 1  pc\_im(i,j,:) = pc\_rgb(:,2);  case 2  pc\_im(i,j,:) = pc\_rgb(:,3);  case 3  pc\_im(i,j,:) = pc\_rgb(:,4);  end  end  end    % Plot  figure(1);  subplot(1,2,1);  imshow(im./255);  title('Original Image');  subplot(1,2,2);  imshow(pc\_im./255);  title('pseudo-colored Image'); |
| **[hw4\_2.m]** |

* 1. Show your pseudo coloring result.



* 1. Explain your implementation and discuss your results

[ Implementation ]

1. Read image ‘cameraman.png’

2. Change the read image to black and white and readjust the maximum brightness   
 to 255.

3. Divide the intensity value of the input grayscale image by 72, and enter the  
 corresponding RGB value for each section in the output color image.

4. Check conversion result by plotting pseudo-colored image

[ Discussion ]

- By using the psudo-coloring method, the original grayscale image was converted into a color image.

- To improve the quality of color images, it is necessary to define well the values of color images corresponding to the intensity values of grayscale images.

1. Color Image Sharpening (40pt)

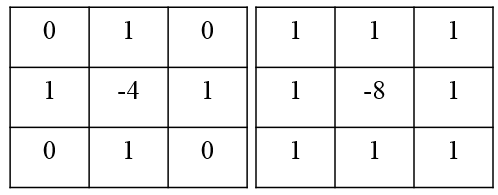


**Figure 3. bird.png**

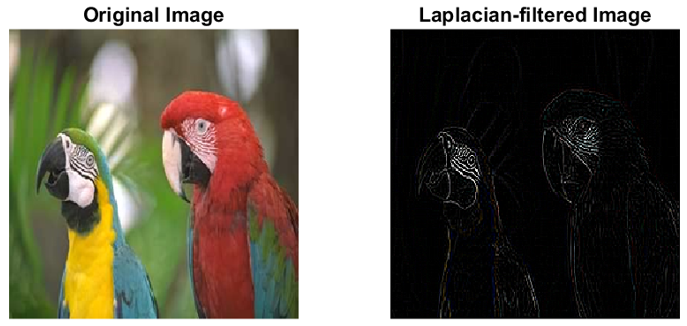
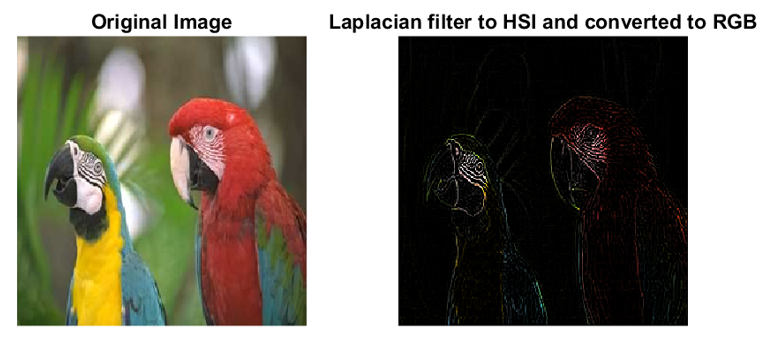
* 1. Read the attached “bird.png” and please answer following questions.
  2. Implement a Matlab function for Laplacian filtering.

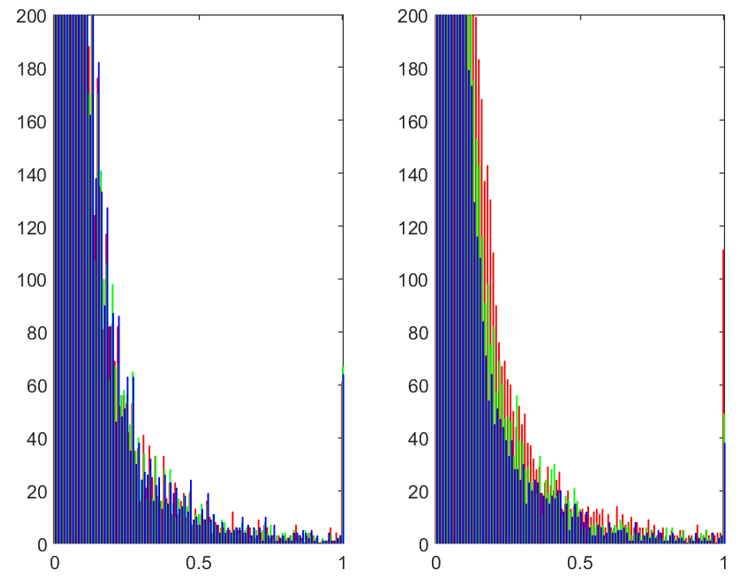
**(You should not use ‘imfilter’ built-in function)**

The Laplacian filter mask is as follow:



|  |
| --- |
| function output = LaplacianFilter(I,M)  size\_I = size(I);  filtered = zeros(size\_I);  % for color image  if length(size\_I) == 3  ch = size\_I(3);  % apply Laplacian filter  for i=1:ch  filtered(:,:,i) = conv2(I(:,:,i),M,'same');  end  else % for grayscale image  filtered = conv2(I,M,'same');  end    output = filtered;  end |
| **[LaplacianFilter.m]** |

1. Show filtering results of the input image (=Laplacian filtered image). 
   1. Implement HSI conversion using question 1-ii
   2. Apply laplacian filter mask to the HSI intensity component
   3. Show conversion from HSI to RGB using question 1-iv.   
      
   4. Show difference between the two results (results of processing RGB and results of processing the HSI intensity component)



* 1. Explain your implementation and discuss your results

[ Implementation ]

1. Read image ‘bird.png’

2. Defines the value of the Laplacian filter mask.

3. Apply the Laplacian filter to the original image and check the results.

4. Converts the original image to an HSI image.

5. Apply the Laplacian filter to the converted HSI image and check the results.

6. Using the plotting and histogram, check the effect of applying the Laplacian filter of the original image and HSI image.

[ Discussion ]

- When you apply a Laplacian filter to an RGB image, only the values corresponding to the edge part of the image are filtered out.

- Applying a Laplacian filter to an RGB image eliminates color information from the edge of the image

- But it can be seen that color information of the edge will be preserved if Laplacian filter apply to the intensity value of the HSI image and then converted back to RGB.

- By checking the histogram of the output image, it can be seen that color information of the edge portion is preserved when the Laplacian filter is applied to the strength value of the HSI.

|  |
| --- |
| % i. image read  im = im2double(imread('bird.png'));  % ii. Laplacian filtering  % make Laplacian filter mask and apply Laplacian filter  mask = [0 1 0;1 -4 1;0 1 0];  laped\_im = LaplacianFilter(im,mask);    % iii. Show Laplacian filtered image  figure(1);  subplot(1,2,1);  imshow(im);  title('Original Image');  subplot(1,2,2);  imshow(laped\_im);  title('Laplacian-filtered Image');    % iii. HSI conversion  hsi\_im = toHSI(im);    % iv. apply Laplacian filter to HSI intensity  laped\_hsi = hsi\_im;  laped\_hsi(:,:,3) = LaplacianFilter(laped\_hsi(:,:,3),mask);    % v. HSI to RGB  laped\_rgb = toRGB(laped\_hsi);  % show conversion  figure(2);  subplot(1,2,1);  imshow(im);  title('Original Image');  subplot(1,2,2);  imshow(laped\_rgb);  title('Laplacian filter to HSI and converted to RGB');    % vi. show difference  % plotting  figure(3);  subplot(1,2,1);  imshow(laped\_im);  title('Laplacian filter to Original Image');  subplot(1,2,2);  imshow(laped\_rgb);  title('Laplacian filter to HSI and converted to RGB');  % histogram  figure(4)  d = 0:0.01:1;  subplot(1,2,1);  hist(reshape(laped\_im,[],3),d);  colormap([1 0 0;0 1 0;0 0 1]);  ylim([0 200]);  subplot(1,2,2);  hist(reshape(laped\_rgb,[],3),d);  colormap([1 0 0;0 1 0;0 0 1]);  ylim([0 200]); |
| **[hw4\_3.m]** |