Multimedia Technology Lab Assignment
Assignment - 3

Name:Shishu
Reg No:2020CA089

1. SMQT

```
function A = alg_a(RGBimage, n)
%This algorithm transforms an RGB image to an HSV image, and
transforms the V component.
%
[row, column, d] = size(RGBimage);%We get the size of the image
if (d==3) %if the image has 3 dimensions, it's an rgb image
HSVimage = rgb2hsv(RGBimage); %Transform from RGB to HSV V =
HSVimage(:,:,3); %Get the component V (brightness) else %if the
image has 1 dimension, it is a gray-scale image
V =double(RGBimage)/255;%In this case, we don't need to transform end
V=V(:); %The matrix of brightness is now a vertical vector
[Vsorted, ix] = sort(V); %Sorting the vector
s = (row*column)/n; %size of the intervals
i=0; %initializing i h=[]; %initialaizing h
% now, there is a loop to process every interval while (i < n) i=i+1;
z = Vsorted(((floor(s*(i-1))+1)):floor(s*i)); %we define the
interval Vstart = (s*(i-1))/(row*column); %We define the start and
the end of the interval
Vstop = (s*i)/(row*column); %linear
transform for each segment r=z-
z(1);
f = (1/n)/(r(size(r,1)));
g = r*f; if(isnan(g(1))) g
= r + Vstop; else g = g +
Vstart; end
h=vertcat(h,g); %Bulding the transformed vector end
```

2. Histogram Equalization Matlab

```
% Function to
call Histogram Equalization Matlab.
% Input:
% - OriginalImage is original image %
Output:
% - HistogramEqualization is enhancement image
%% function HistogramEqualization =
alg_hm(OriginalImage)
[row, column, d] = size(OriginalImage);
i=0; while(i<d) i=i+1;
HistogramEqualization(:,:,i) = histeq(OriginalImage(:,:,i));
end end
```

3. Auxiliary functions

```
%% ImageEnhancement.m
```

```
% Function applies SMQT, Histog Equalization(own function and Matlab
% function), and HSV transform.
% To use: % nameImagen =
imread('c:\matlab\nameImagen.tif'); % Input:
% - OriginalImage is the original image %
Output:
% Print Original and histogram picture and Images enhancement and
% histogram enhacement
%% function
ImageEnhancement(OriginalImage)
OriginalImage = uint8(OriginalImage);
%Call SMQT function
img_m=alg_m(OriginalImage,1,8);%variables: Image, l, L. l must be
one. L is the number of levels of the SMQT %call Histogram
equalization (own function) img_h=alg_h(OriginalImage); %Call
Histogram equalization (matlab) img_hm=alg_hm(OriginalImage);
%Call HSV, V transform algorithm, n=1
img_a1=alg_a(OriginalImage,1); %Call HSV, V transform algorithm,
n=10 img_a10=alg_a(OriginalImage,10);
%% PRINT ON SCREEN
% Print on screen a general comparative of all images
generalComp(OriginalImage,img_m,img_h,img_hm,img_a1,img_a10); %
Print on screen an individual comparative between original image
and the % others
individualComp(OriginalImage,img_m,img_h,img_hm,img_a1,img_a10) end
```

4. Split RGB components

```
%% splitRGB.m
% Function to split all components of RGB image and then return count
and
% bin of histogram of each component %
Input:
% - image is de original image %
Output:
% - yRed, yGreen, yBlue are histogram counts of components
% - xr, xg, xb are bin locations of components
%% function
[yRed,xr,yGreen,xg,yBlue,xb]=splitRGB(image)
%Split into RGB Channels
Red = image(:,:,1);
Green = image(:,:,2);
Blue = image(:,:,3);
%Get histValues for each channel
[yRed, xr] = imhist(Red);
[yGreen, xg] = imhist(Green);
[yBlue, xb] = imhist(Blue); end
```

5. Print on screen Individual comparative between original image and transform image

```
%% individualComp.m
% Function to print on screen an individual comparative between
original
% image and the others.
% Input:
```

```
% - OriginalImage,img_m,img_h,img_hm,img_a1,img_a10 are image %
Output:
% Print Original and histogram picture and Images enhancement and
% histogram enhacement function
individualComp(OriginalImage,img_m,img_h,img_hm,img_a1,img_a
10)
%Original vs SQMT figure;
subplot(2,2,1);imshow(OriginalImage);title('Original Pic.');
subplot(2,2,3);imshow(img_m);title('SMQT (L=8)'); if
(size(OriginalImage,3)<=1)% if image is in greyscale
subplot(2,2,2);imhist(OriginalImage);title('Original histogram');
subplot(2,2,4);imhist(img_m);title('SMQT histogram(L=8)'); else %
if image is in color
[yRed,xr,yGreen,xg,yBlue,xb]=splitRGB(OriginalImage);subplot(2,2,2);
plot(xr, yRed, 'r', xg, yGreen, 'g', xb, yBlue, 'b');title('Original
histogram');
[yRed,xr,yGreen,xg,yBlue,xb]=splitRGB(img_m); subplot(2,2,4);
plot(xr, yRed, 'Red', xg, yGreen, 'Green', xb, yBlue, 'Blue');
title('SMQT histogram(L=8)'); end
% Original vs Histogram Equalization figure;
subplot(2,2,1);imshow(OriginalImage);title('Original Pic.');
subplot(2,2,3);imshow(img_h);title('Histogram Equalization'); if
(size(OriginalImage,3)<=1)% if image is in greyscale
subplot(2,2,2);imhist(OriginalImage);title('Original histogram');
subplot(2,2,4);imhist(img_h);title('Histogram Equalization');
else % if image is in color
[yRed,xr,yGreen,xg,yBlue,xb]=splitRGB(OriginalImage);subplot(2,2,2);
plot(xr, yRed, 'r', xg, yGreen, 'g', xb, yBlue, 'b');title('Original
histogram');
[yRed,xr,yGreen,xg,yBlue,xb]=splitRGB(img_h); subplot(2,2,4);
plot(xr, yRed, 'Red', xg, yGreen, 'Green', xb, yBlue, 'Blue');
title('Histogram Equalization'); end
% Original vs Histogram Equalization (Matlab) figure;
subplot(2,2,1);imshow(OriginalImage);title('Original Pic.');
subplot(2,2,3);imshow(img_hm);title('Histogram Equalization
(Matlab)');
if (size(OriginalImage,3)<=1)% if image is in greyscale
subplot(2,2,2);imhist(OriginalImage);title('Original histogram');
subplot(2,2,4);imhist(img_hm);title('Histogram Equalization
(Matlab)');
else % if image is in color
[yRed,xr,yGreen,xg,yBlue,xb]=splitRGB(OriginalImage);subplot(2,2,2);
plot(xr, yRed, 'r', xg, yGreen, 'g', xb, yBlue, 'b');title('Original
histogram');
[yRed,xr,yGreen,xg,yBlue,xb]=splitRGB(img_hm); subplot(2,2,4);
plot(xr, yRed, 'Red', xg, yGreen, 'Green', xb, yBlue,'Blue');
title('Histogram Equalization (Matlab)'); end
% Original vs HSV (n=1) figure;
subplot(2,2,1);imshow(OriginalImage);title('Original Pic.');
subplot(2,2,3);imshow(img_a1);title('HSV, V transform algorithm
(n=1)');
if (size(OriginalImage,3)<=1)% if image is in greyscale
subplot(2,2,2);imhist(OriginalImage);title('Original histogram');
subplot(2,2,4);imhist(img_a1);title('HSV histogram, V transform
algorithm (n=1)');else % if image is in color
[yRed,xr,yGreen,xg,yBlue,xb]=splitRGB(OriginalImage);subplot(2,2,2);
plot(xr, yRed, 'r', xg, yGreen, 'g', xb, yBlue, 'b');title('Original
histogram');
[yRed,xr,yGreen,xg,yBlue,xb]=splitRGB(img_a1); subplot(2,2,4);
plot(xr, yRed, 'Red', xg, yGreen, 'Green', xb, yBlue, 'Blue');
title('HSV histogram, V transform algorithm (n=1)');end
% Original vs HSV (n=10) figure;
subplot(2,2,1);imshow(OriginalImage);title('Original Pic.');
subplot(2,2,3);imshow(img_a10);title('HSV, V transform algorithm
(n=10)');
if (size(OriginalImage,3)<=1)% if image is in greyscale
subplot(2,2,2);imhist(OriginalImage);title('Original histogram');
subplot(2,2,4);imhist(img_a10);title('HSV histogram, V transform
algorithm (n=10)');else % if image is in color
[yRed,xr,yGreen,xg,yBlue,xb]=splitRGB(OriginalImage);subplot(2,2,2);
plot(xr, yRed, 'r', xg, yGreen, 'g', xb, yBlue, 'b');title('Original
histogram');
[yRed,xr,yGreen,xg,yBlue,xb]=splitRGB(img_a10); subplot(2,2,4);
plot(xr, yRed, 'Red', xg, yGreen, 'Green', xb, yBlue, 'Blue');
title('HSV histogram, V transform algorithm (n=10)');end end
```