**Experiment – 8**

***MCQ:*** D) none of these

**Region Growing Algorithm**

**FUNCTION**

function J=regiongrowing(I,x,y,reg\_maxdist)

if(exist('reg\_maxdist','var')==0), reg\_maxdist=0.2; end

if(exist('y','var')==0), figure, imshow(I,[]); [y,x]=getpts; y=round(y(1)); x=round(x(1)); end

J = zeros(size(I)); % Output

Isizes = size(I); % Dimensions of input image

reg\_mean = I(x,y); % The mean of the segmented region

reg\_size = 1; % Number of pixels in region

% Free memory to store neighbours of the (segmented) region

neg\_free = 10000; neg\_pos=0;

neg\_list = zeros(neg\_free,3);

pixdist=0; % Distance of the region newest pixel to the regio mean

% Neighbor locations (footprint)

neigb=[-1 0; 1 0; 0 -1;0 1];

% Start regiogrowing until distance between regio and posible new pixels become

% higher than a certain treshold

while(pixdist<reg\_maxdist&&reg\_size<numel(I))

% Add new neighbors pixels

for j=1:4,

% Calculate the neighbour coordinate

xn = x +neigb(j,1); yn = y +neigb(j,2);

% Check if neighbour is inside or outside the image

ins=(xn>=1)&&(yn>=1)&&(xn<=Isizes(1))&&(yn<=Isizes(2));

% Add neighbor if inside and not already part of the segmented area

if(ins&&(J(xn,yn)==0))

neg\_pos = neg\_pos+1;

neg\_list(neg\_pos,:) = [xn yn I(xn,yn)]; J(xn,yn)=1;

end

end

% Add a new block of free memory

if(neg\_pos+10>neg\_free), neg\_free=neg\_free+10000; neg\_list((neg\_pos+1):neg\_free,:)=0; end

% Add pixel with intensity nearest to the mean of the region, to the region

dist = abs(neg\_list(1:neg\_pos,3)-reg\_mean);

[pixdist, index] = min(dist);

% Calculate the new mean of the region

reg\_mean= (reg\_mean\*reg\_size + neg\_list(index,3))/(reg\_size+1);

% Save the x and y coordinates of the pixel (for the neighbour add proccess)

x = neg\_list(index,1); y = neg\_list(index,2);

% Remove the pixel from the neighbour (check) list

neg\_list(index,:)=neg\_list(neg\_pos,:); neg\_pos=neg\_pos-1;

end

% Return the segmented area as logical matrix

J=J>1;

**MAIN**

clc;

clear variables;

close all;

I = imread('Exp8 Image1.png');

G = rgb2gray(I);

imshow(G);

Y = im2double(G);

x=120; y=200;

J = regiongrowing(Y,x,y,0.2);

figure, imshow(Y+J);

**FIGURE 1**



**FIGURE 2**



**K MEANS CLUSTERING**

clc;

clear variables;

close all;

he = imread('Exp8 Image1.png');

%Convert image from rgb to La\*b\*

cform = makecform('srgb2lab');

lab\_he = applycform(he,cform);

imshow (lab\_he)

ab = double(lab\_he(:,:,2:3));

nrows = size(ab,1);

ncols = size(ab,2);

ab = reshape(ab,nrows\*ncols,2);

nColors = 3;

% repeat the clustering 3 times to avoid local minima

[cluster\_idx, cluster\_center] = kmeans(ab,nColors,'distance','sqEuclidean', ...

'Replicates',3);

pixel\_labels = reshape(cluster\_idx,nrows,ncols);

segmented\_images = cell(1,3);

rgb\_label = repmat(pixel\_labels,[1 1 3]);

for k = 1:nColors

color = he;

color(rgb\_label ~= k) = 0;

segmented\_images{k} = color;

end

%figure, imshow(segmented\_images{1}), title('objects in cluster 1');

figure, imshow(segmented\_images{2}), title('objects in cluster 2');

%figure, imshow(segmented\_images{3}), title('objects in cluster 3');

% The object mentioned in the exercise is in cluster 2 (The red circle)





