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

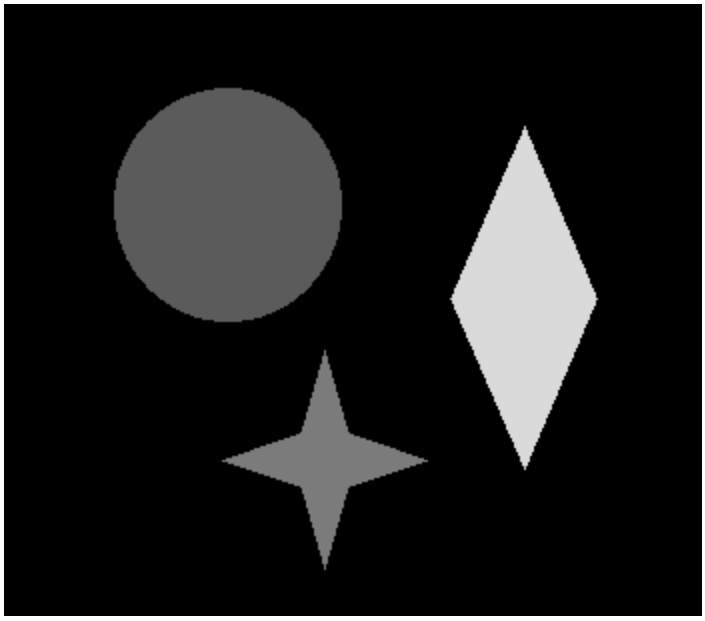
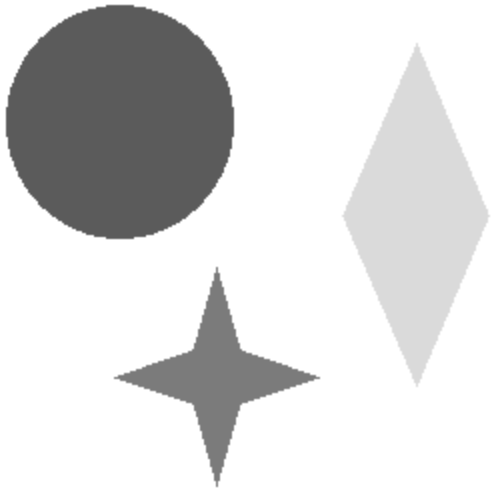
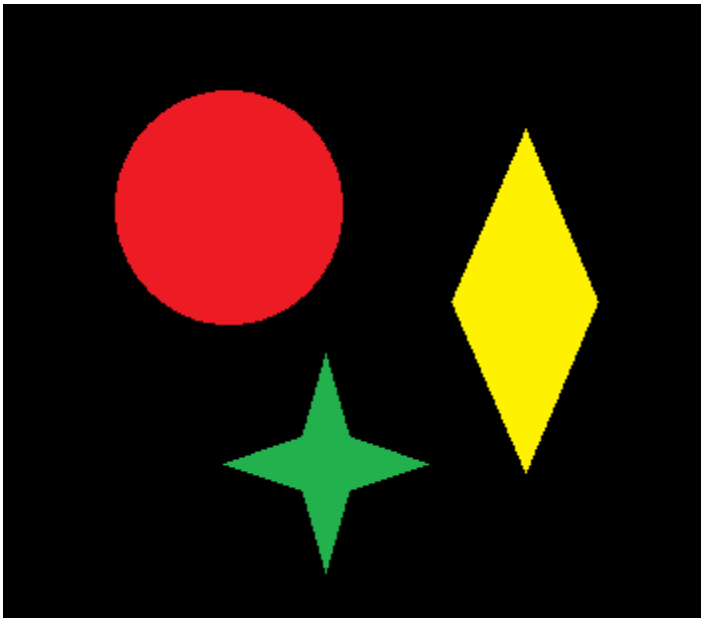


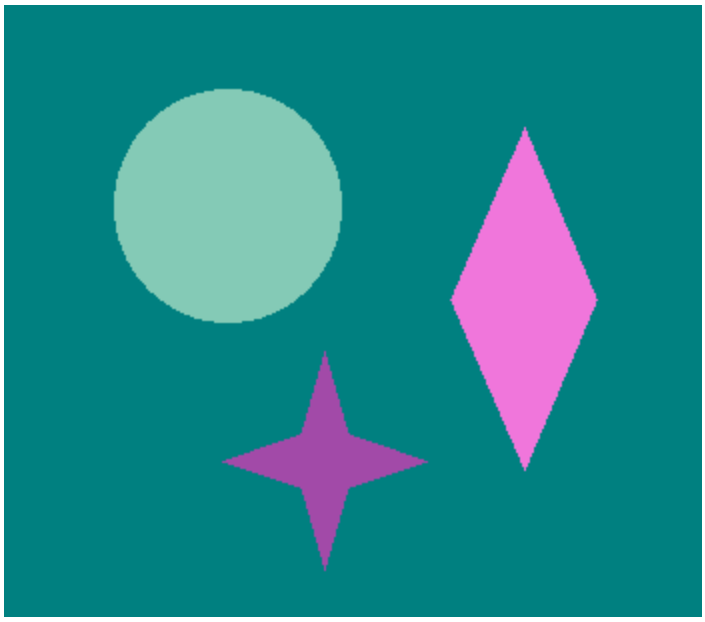
FIGURE 2



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)
```





objects in cluster 2