CS 558: Homework Assignment 3 -- Image Segmentation
Program Language: MATLAB
Name: Lan Chang
Date: 04/05/17

1. k-means Segmentation

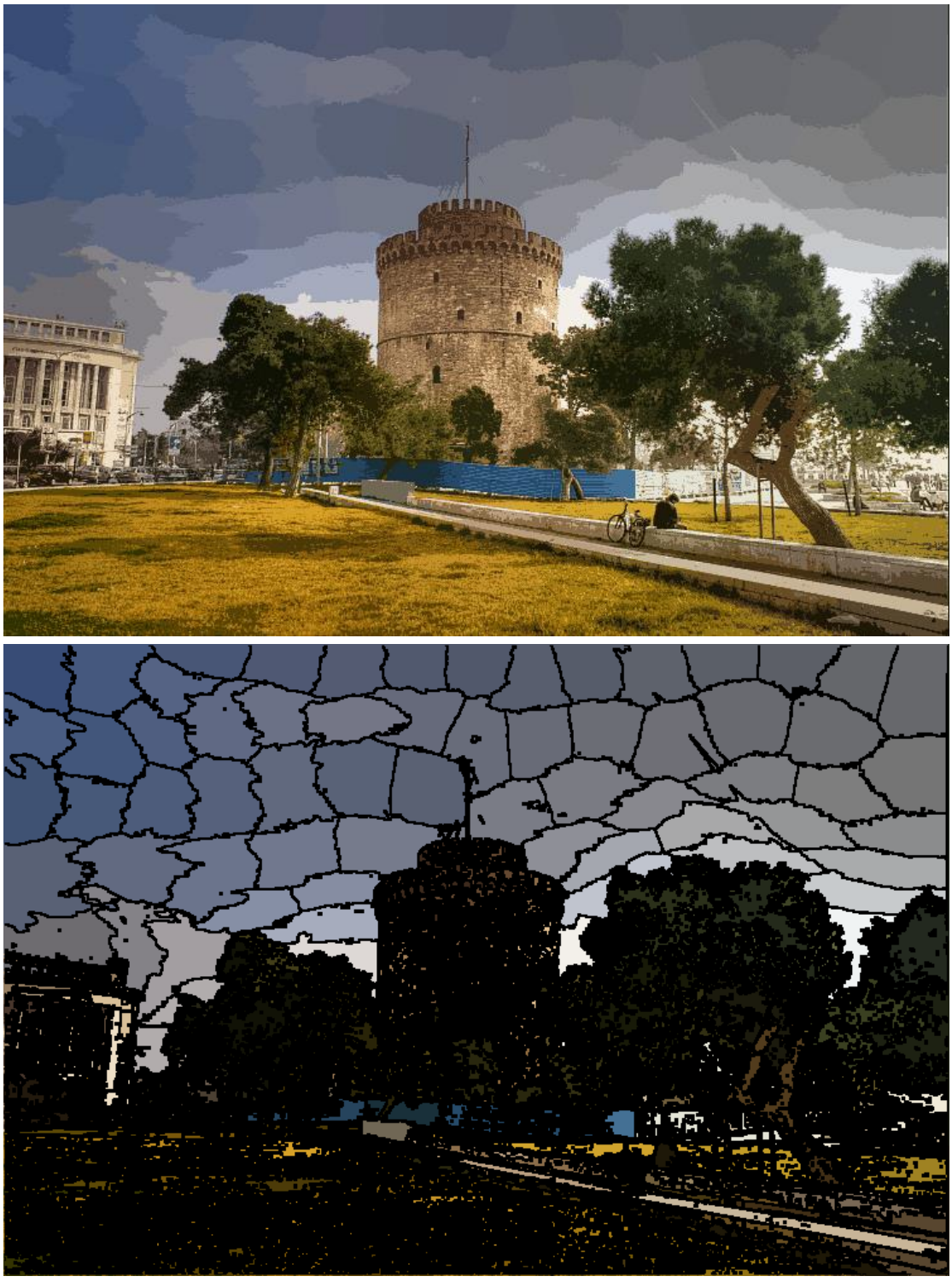
First, I randomly initialize ten cluster centers, and for each center, get the RGB. Next, I calculate the distance using red, green and blue between each pixel and each cluster's center, and find which center these pixels are closest to. Then, I find the centroid of these points in each cluster and use it as the new cluster's center. If there is no different between the new one and the old one, I will stop the loop. Finally, I will color each cluster using the new center's RGB.

Result:

2. SLIC

First, I filter the image's RGB channels using sobel filter to get the gradient magnitude. Next, in each 50×50 block's center, initialize it as centroid and then move the centroid to the smallest gradient magnitude in 3×3 window. Then, I apply k-means to the image using red, green, blue, x-coordinate/2 and y-coordinate/2 and get the result. Finally, I color each pixel black if either the eight pixels around it has a different color with it.

Result:



3. MATLAB Code

(1) homework3_1.m

```
clear all;

% Problem 1: k-means Segmentation
% input the image
image1=imread('white-tower.png');
h1=size(image1,1);
w1=size(image1,2);
image1=double(image1);

% parameter
k=10;

% randomly initialize the cluster centers
center=rand(k,2);
center(:,1)=round(center(:,1).*h1)+1;
center(:,2)=round(center(:,2).*w1)+1;

% k-means segmentation
image2=kmeans(image1,center,k);

% show the result
image2=uint8(image2);
figure,imshow(image2);
imwrite(image2,'kmeans.bmp','bmp');
```

(2) homework3_2.m

```
clear all;

% Problem 2: SLIC
% input the image
image3=imread('wt_slic.png');
h2=size(image3,1);
w2=size(image3,2);
image3=double(image3);

% parameter
s=50;

% slic
image4=slic(image3,s);

% show the result
image4=uint8(image4(:,:,1:3));
figure,imshow(image4);
imwrite(image4,'slic1.bmp','bmp');

% color the pixel that touch two different clusters black
image5=image4;
for i=2:h2-1
    for j=2:w2-1
        flag=0;
        for t=1:3
            if image4(i,j,t)~=image4(i-1,j-1,t)
                flag=1;
            end
            if image4(i,j,t)~=image4(i-1,j,t)
                flag=1;
            end
            if image4(i,j,t)~=image4(i-1,j+1,t)
                flag=1;
            end
            if image4(i,j,t)~=image4(i,j-1,t)
                flag=1;
            end
            if image4(i,j,t)~=image4(i,j+1,t)
                flag=1;
            end
            if image4(i,j,t)~=image4(i+1,j-1,t)
                flag=1;
```

```
                end
                if image4(i,j,t)~=image4(i+1,j,t)
                    flag=1;
                end
                if image4(i,j,t)~=image4(i+1,j+1,t)
                    flag=1;
                end
            end
            if flag==1
                image5(i,j,:)=[0 0 0];
            end
        end
end

% show the result
figure,imshow(image5);
imwrite(image5,'slic2.bmp','bmp');
```

(3) kmeans.m

```
function image2=kmeans(image1,center,k)
% get the height and width of the image
h=size(image1,1);
w=size(image1,2);
m=size(image1,3);

% get the center's RGB
c1=zeros(k,m);
c2=zeros(k,m);
for i=1:k
    c1(i,1:m)=image1(center(i,1),center(i,2),:);
end

% initialize the distance
dis=inf(h,w,k);

flag=0;
while ~flag
    % calculate the distance between each point and each cluster's center
    for i=1:k
        for d=1:m
            temp(:,:,d)=repmat(c1(i,d),h,w);
        end

        dis(:,:,i)=sqrt(sum(power(image1-temp,2),3));
    end

    % find out which cluster's center these points are closest to
    [~,d] = min(dis,[],3);

    % find the centroid of these points in each cluster
    for i=1:k
        idx=d==i;
        rgb=reshape(image1(repmat(idx,1,1,m)),sum(idx(:)),m);
        c2(i,:)=round(mean(rgb));
    end

    % if no change, stop
    if abs(norm(c1-c2))<0.1
        flag=1;
    end

    c1=c2;
```

```
end

% draw a new image
image2=zeros(h,w,m);
for i=1:h
    for j=1:w
        image2(i,j,:)=c1(d(i,j),:);
    end
end

end
```

(4) slic.m

```
function image2=slic(image1,s)
% get the height and width of the image
h=size(image1,1);
w=size(image1,2);

% gradient magnitude
sobel_x=[-1,0,1;-2,0,2;-1,0,1];
sobel_y=[1,2,1;0,0,0;-1,-2,-1];
gradient(:,:,1)=filtering(image1(:,:,1),sobel_x);
gradient(:,:,2)=filtering(image1(:,:,2),sobel_x);
gradient(:,:,3)=filtering(image1(:,:,3),sobel_x);
gradient(:,:,4)=filtering(image1(:,:,1),sobel_y);
gradient(:,:,5)=filtering(image1(:,:,2),sobel_y);
gradient(:,:,6)=filtering(image1(:,:,3),sobel_y);
gradient=sqrt(sum(power(gradient,2),3));

% initialize the centroids and move it to the position with the smallest gradient magnitude
count=1;
centroids=zeros(h,w);
for i=round((s+1)/2):s:h
    for j=round((s+1)/2):s:w
        window=gradient(i-1:i+1,j-1:j+1);
        [~,small]=min(window(:));
        [i2,j2]=ind2sub(size(window),small);
        centroids(i+i2-2,j+j2-2)=1;
        center(count,:)=[i+i2-2,j+j2-2];
        count=count+1;
    end
end

% divide x and y by 2
[x,y]=meshgrid(1:h,1:w);
x=x';
y=y';
image3(:,:,1:3)=image1;
image3(:,:,4)= x./2;
image3(:,:,5)= y./2;

% k-means
image2=kmeans(image3,center,count-1);

end
```

(5) filtering.m

```
function image_fil=filtering(image_ori,filter)
% width and height of original image
width_i=size(image_ori,2);
height_i=size(image_ori,1);

% width and height of filter
width_f=size(filter,2);
height_f=size(filter,1);
half_wf=(width_f-1)/2;
half_hf=(height_f-1)/2;

% extend
image_temp1=zeros(height_i+half_hf*2,width_i+half_wf*2);
for i=1:height_i
    for j=1:width_i
        image_temp1(i+half_hf,j+half_wf)=image_ori(i,j);
    end
end

% replicate boundary
for i=1:height_i+half_hf*2
    for j=1:width_i+half_wf*2
        if j<=half_wf&&i>=half_hf+1&&i<=height_i+half_hf
            image_temp1(i,j)=image_temp1(i,half_wf+1);
        elseif j>=width_i+half_wf+1&&i>=half_hf+1&&i<=height_i+half_hf
            image_temp1(i,j)=image_temp1(i,width_i+half_wf);
        elseif i<=half_hf&&j>=half_wf+1&&j<=width_i+half_wf
            image_temp1(i,j)=image_temp1(half_hf+1,j);
        elseif i>=height_i+half_hf+1&&j>=half_wf+1&&j<=width_i+half_wf
            image_temp1(i,j)=image_temp1(height_i+half_hf,j);
        elseif j<=half_wf&&i<=half_hf
            image_temp1(i,j)=image_temp1(half_hf+1,half_wf+1);
        elseif j<=half_wf&&i>=height_i+half_hf+1
            image_temp1(i,j)=image_temp1(height_i+half_hf,half_wf+1);
        elseif i<=half_hf&&j>=width_i+half_wf+1
            image_temp1(i,j)=image_temp1(half_hf+1,width_i+half_wf);
        elseif j>=width_i+half_wf+1&&i>=height_i+half_hf+1
            image_temp1(i,j)=image_temp1(height_i+half_hf,width_i+half_wf);
        end
    end
end

% filtering
```

```
image_temp2=image_temp1;
for i=1+half_hf:height_i+half_hf
      for j=1+half_wf:width_i+half_wf
            image_temp2(i,j)=sum(sum(filter.*image_temp1(i-half_hf:i+half_hf,j-
half_wf:j+half_wf)));
      end
end

% cut
image_fil=zeros(height_i,width_i);
for i=1:height_i
      for j=1:width_i
            image_fil(i,j)=image_temp2(i+half_hf,j+half_wf);
      end
end

end
```