Object Detection

Vishal S, IIT Madras

1 Introduction

In this assignment we will try to count the number of objects present in an image. This consists of two parts. The first part comprises of writing a MATLAB script to place a number of circles and rectangles in an image. In the second part a script has been written to read the image created by the previous script and to read the number of objects present.

2 Object Creation

A MATLAB script can be written to create random objects in an image. The process is completely randomized from the number of objects to its size. The MATLAB script is given below.

```
clear all
clf
clc
Nx = 1000;
Ny = 1200;
A = zeros(Nx,Ny);

r = randi([6  10],1); % Number of objects

for i = 1:r
    obj_id = randi([1  2],1);
    if(obj_id==1) % For creating rectangles
        x_center = randi([10,Nx-20],1);
        y_center = randi([10,Ny-20],1);
        length = randi([50,Nx/10],1);
        breadth = randi([50,Nx/10],1);
        for  m = x_center : x_center+length
            for n = y_center : y_center +breadth
                A(m,n) = 1;
            end
        end
    elseif(obj_id==2)    % For creating circles
        rad = randi([10,70],1);
        x_center = randi([5,Nx-4],1);
        y_center = randi([5,Ny-4],1);
        for  m = 3:Nx-2
            for n = 3:Ny-2
                distx = m-x_center ;
                disty =n-y_center;
                dist = sqrt(distx^2 + disty^2);
                if(dist <= rad )
                    A(m,n) = 1;
                end
            end
        end
    end
end

g = im2uint8(A);
imshow(g);
imwrite(g,'squares.jpg');
```

3 Object Detection

The image created by the above code is read. The image is converted to a binary image and then an kernel operation is done on it to detect the edges. Once the edges are detected, the image is converted to binary. Now we have an image which is `1` at the edges and `0` everywhere else. It is made sure that there is no more

than 2 pixels with a value `1` in the Moore neighbourhood [2] of a pixel with a value `1`[1].
Now, the image is scanned until a `1` is found. Once an edge is encountered the `count` is incremented. Then, we move along the edge tagging it as a part of the object. This is carried on until the initial pixel is encountered or we run into a dead end. `Bdash` is the matrix on which the above operations are carried out.
A copy of the matrix is made and stored as `B`. Whenever a pixel corresponding to an edge is detected in `Bdash` , the corresponding pixel in `B` is made `0`. After completing an object the values are copied from `B` into `Bdash`. This removes the object from our image and the above mentioned process is carried on until another object is encountered or there are no more objects in the image.
Finally, the original image is compared with the obtained contours and the objects are allotted colours and then printed out.

3.1 Moving Along the Edge

To move along the edge of an object. First a point in the edge of the object is found. A Moore neighbourhood [2] is defined and then the values in the neighbourhood are checked to see if any one of them is also an edge in a clock-wise manner starting from the pixel adjacent to the previous pixel index. The order in which this is carried out is given below.

1	2	3
`oldi,oldj`	`i,j`	4
7	6	5

The MATLAB script that implements the above algorithm is given below.

```
clear all
clf
clc
%Reading the file
file = imread('object_1.jpg');
g = mat2gray(file);
N = size(g);
Nx = N(1);
Ny = N(2);
kernel = [-1,-1,-1;-1,8,-1;-1,-1,-1];
B =zeros(Nx,Ny);
% Thresholding the image to make it binary
for i= 1:Nx
    for j = 1:Ny
        if(g(i,j)<0.5)
            g(i,j) =0;
        else
            g(i,j) = 1;
        end
    end
end
%% Implementation of Kernel operation for edge detection
for i = 2:Nx-1
    for j = 2:Ny-1
        for m = -1:1
            for n = -1:1
                B(i,j) = B(i,j) + kernel(m+2,n+2)*g(i+m,j+n);
            end
        end
    end
end
for i= 2:Nx-1
    for j = 2:Ny-1
        if((B(i,j) >=0) &&(B(i,j)<0.5))
            B(i,j) =0;
        else
            B(i,j) = 1;
        end
    end
end
Bdash = zeros(Nx,Ny);
contours = zeros(Nx,Ny);
```

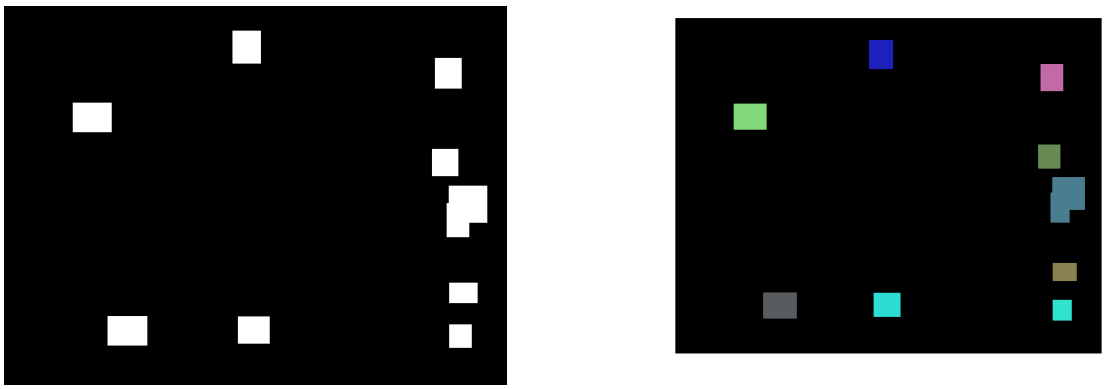
```
for i= 2:Nx-1
    for j = 2:Ny-1
        if((B(i,j)==1) &&(g(i,j)==1))
            Bdash(i,j) = 1;
        end
    end
end

image = im2uint8(B);
imshow(image);
pause(1);
count = 0;
count_before = 0;
%% To reduce the number of edge pixcels in the Moore neighbourhood
order = [-1,-1;-1,0;-1,1;0,1;1,1;1,0;1,-1;0,-1];
k1 = [0,1,0;1,0,0;0,0,0];
k2 = [0,1,0;0,0,1;0,0,0];
k3 = [0,0,0;1,0,0;0,1,0];
k4 = [0,0,0;0,0,1;0,1,0];
for i = 2:Nx-1
    for j = 2:Ny-1
        if(Bdash(i,j) ==1)
            sum1 = 0;
            sum2 = 0;
            sum3 = 0;
            sum4 =0;
            for m = -1:1
                for n = -1:1
                    sum1 = sum1 + k1(m+2,n+2)*Bdash(i+m,j+n);
                    sum2 = sum2 + k2(m+2,n+2)*Bdash(i+m,j+n);
                    sum3 = sum3 + k3(m+2,n+2)*Bdash(i+m,j+n);
                    sum4 = sum4 + k4(m+2,n+2)*Bdash(i+m,j+n);
                end
            end
            if((sum1==2)||(sum2==2)||(sum3==2)||(sum4==2))
                Bdash(i,j) = 0;
            end
        end
    end
end
%% Finding the number of objects
B = Bdash;
for i = 2:Nx-1
    for j = 2:Ny-1
        if(Bdash(i,j) == 1)
            count = count +1;
            B(i,j) =0;
            contours(i,j) = count;
            oldi = i;
            oldj = j-1;
            m = i; n =j;
            starti = i;
            startj = j;
            for l = 1:8
                if((i+order(l,1) == oldi) && (j+order(l,2)==oldj))
                    k = l+1;
                    break
                end
            end
            if(k==9)
                k = 1;
            end
            for a = 1:8
                if(Bdash(m+order(k,1),n+order(k,2)) == 1)
                    B(m,n) =0;
                    contours(m,n) = count;
                    oldi = m;
                    oldj = n;
                    m = m+order(k,1);
                    n = n+order(k,2);
                    for l = 1:8
```

```
                        if((m+order(l,1) == oldi) && (n+order(l,2)==oldj))
                            k = l+1;
                        end
                    end
                    if(k==9)
                        k=1;
                    end
                else
                    k=k+1;

                    if(k==9)
                        k=1;
                    end
                end
            end
            while((m~=starti)||(n~=startj))
                if(Bdash(m+order(k,1),n+order(k,2)) == 1)
                    B(m,n) = 0;
                    contours(m,n) = count;
                    oldi = m;
                    oldj = n;
                    m = m+order(k,1);
                    n = n+order(k,2);
                    for l = 1:8
                        if((m+order(l,1) == oldi) && (n+order(l,2)==oldj))
                            k = l+1;
                            break
                        end
                    end
                    if(k==9)
                        k=1;
                    end
                else
                    k=k+1;

                    if(k==9)
                        k=1;
                    end
                end
            end
        end
        if(count~=count_before)
            count_before = count;
            Bdash = B;
            count
            i=1;
            j=2;
            image = im2uint8(Bdash);
            imshow(image);
            pause(1);
        end

    end
end
count

%% To Tag the pixcels in the interior to the object
for i = 2:Nx-1
    for j = 2:Ny-1
        if((contours(i,j-1)==0)&&(contours(i,j)~=0))
            j=j+1;
            while(g(i,j)==1)
                contours(i,j) = contours(i,j-1);
                j=j+1;
            end
        end
    end
end

%% Assigning different colours to different objects and then printing

```

```
image_contour = zeros(Nx,Ny,3);
red_index = randi([30,230],count);
blue_index = randi([30,230],count);
green_index = randi([30,230],count);
for i = 1:Nx
    for j = 1:Ny
        if(contours(i,j)~=0)
            image_contour(i,j,1) = red_index (contours(i,j));
            image_contour(i,j,2) = blue_index (contours(i,j));
            image_contour(i,j,3) = green_index (contours(i,j));
        end
    end
end

image = uint8(image_contour);
imshow(image);
```

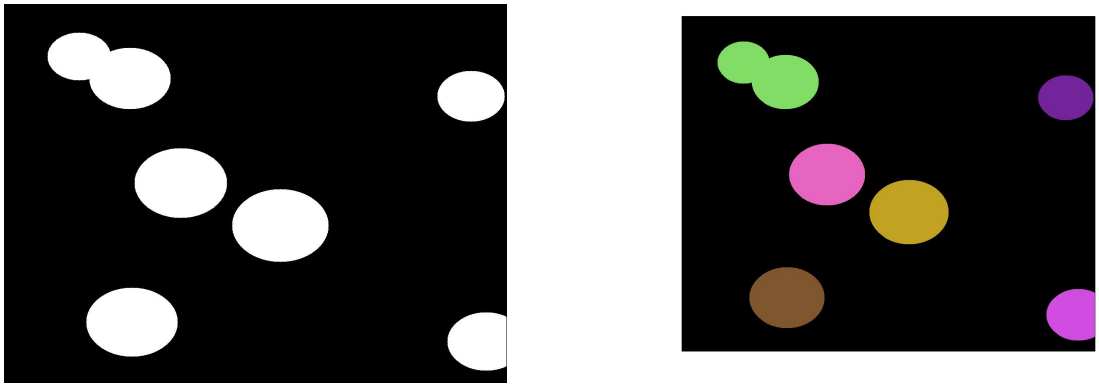
4 Results

The above code is run with different images generated. The results are given below.



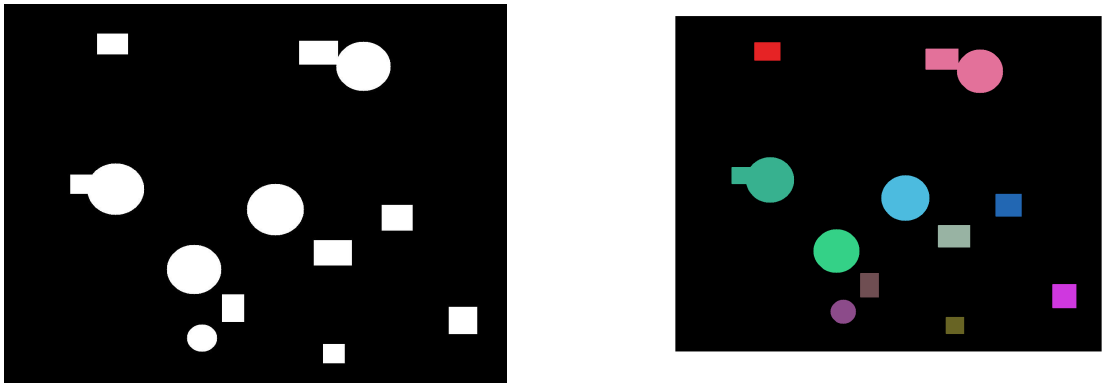
(a) Input (b) Output

Figure 1: The number of objects detected was equal to 9



(a) Input (b) Output

Figure 2: The number of objects detected was equal to 6



(a) Input (b) Output

Figure 3: The number of objects detected was equal to 11

References

[1] Rafael C. Gonzalez, Richard E. Woods, and Steven L. Eddins. *Digital Image Processing Using MATLAB*. Mc Graw Hill Education, 2010.

[2] Wikipedia. *Moore neighborhood — Wikipedia, The Free Encyclopedia*. [Online; accessed 2-October-2017]. 2017. URL: https://en.wikipedia.org/w/index.php?title=Moore_neighborhood&oldid=759043720.