Computer Vision

HW 10

nannapaneni.4

Question 1

In this question we were asked to compute Disparity map for the two images provided left.png and right.png using Basic stereo matching algorithm. To perform template matching we were asked to use Normalized Cross Correlation(NCC).

Here are the 2 images provided to us

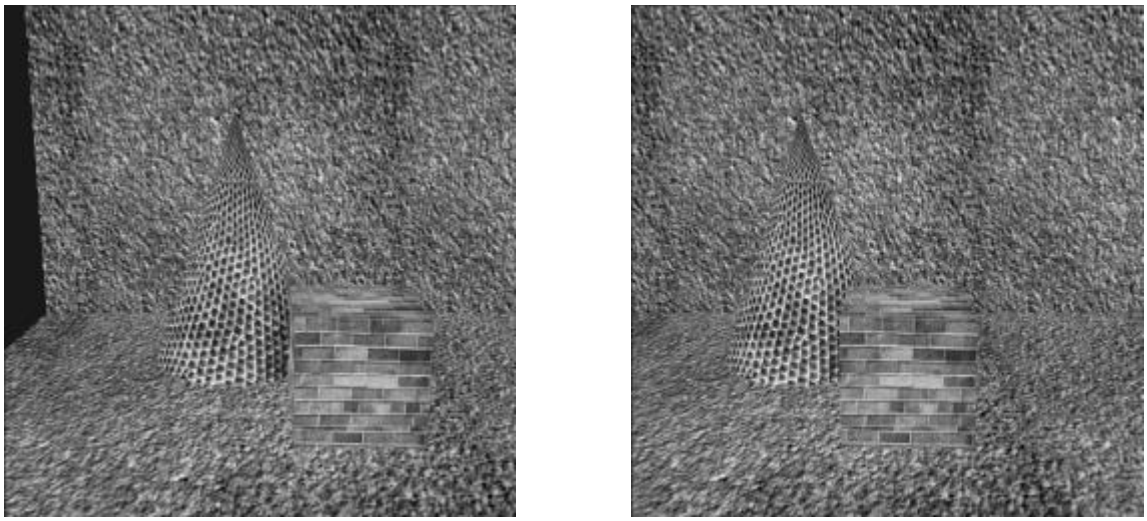


Fig 1. Left.png (left) and Right.png (right)

Code:

```
myleftIm = double(imread('left.png'));
myrightIm = double(imread('right.png'));
% imagesc(myrightIm);
% colormap gray;
windowsize = 11;
dis =
BasicStereoMatchingAlgo(myleftIm,myrightIm,windowsize);

%% testing

imagesc(dis, [0 50]);
axis equal;
colormap gray;

%% functions

function disparity =
BasicStereoMatchingAlgo(mylIm,myrIm,windowsize)
```

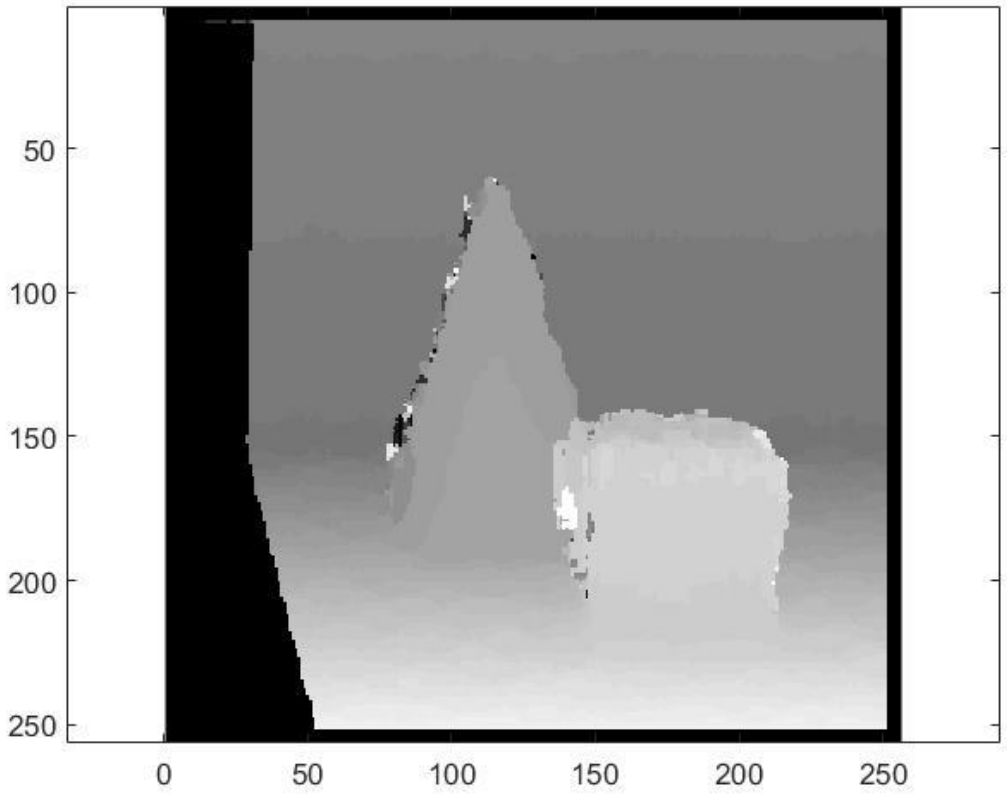
```
    disparity = zeros(size(mylIm));
    for i = ceil(windowsize/2) : size(mylIm,1)-
ceil(windowsize/2)+1
        for j = ceil(windowsize/2) : size(mylIm,1)-
ceil(windowsize/2)+1
            temptem = mylIm(i-5:i+5,j-5:j+5);
            temptemmean = mean(temptem(:,:),'all');
            temptemstd = std(temptem(:,:),0,'all');
%             if j > 56
%                 ncct = zeros(51,2);
%             else
%                 ncct = zeros(j-5,2);
%             end
            n = 1;
            for k = j:-1:j-50
                if k == 5
                    break;
                else
                    temp2 = myrIm(i-5:i+5,k-5:k+5);
                    ncct(n,:) = [
(NCC(temptem,temp2,temptemmean,temptemstd,windowsize))/12
0  k ];
                    n=n+1;
                end
            end
            [f,~] = find(ncct == max(ncct(:,1)));
            fin = ncct(f,2);
            disparity(i,j) = j - fin;
        end
    end

end

function nn =
NCC(temptem,temp2,temptemmean,temptemstd,windowsize)
    temp2mean = mean(temp2(:,:),'all');
    temp2std = std(temp2(:,:),0,'all');
    nn = 0;
    for k = 1:windowsize
        for l = 1:windowsize
            nn = nn + ((temp2(k,l)-
temp2mean)*(temptem(k,l)-
temptemmean)/(temp2std*temptemstd));
        end
    end
end
```

Results:

Here is the Disparity Map that I got



Question 2

In this question we were asked to implement KNN classification and were given training data and test data.

Code:

```
train = importdata('train.txt');
test = importdata('test.txt');
K = 1;
testtest = KNN(K,test,train);
result = zeros(size(test));
result(:,1:2) = test(:,1:2);
result(:,3) = testtest(:);
c = 0;
```

```
c1 = 1;
c2 = 1;
er = 1;
for i = 1:size(test,1)
    if testtest(i) == test(i,3)
        c = c+1;
    else
        resulter(er,:) = result(i,:);
        er = er+1;
    end
    if result(i,3) == 1
        result1(c1,:) = result(i,:);
        c1 = c1+1;
    elseif result(i,3) == 2
        result2(c2,:) = result(i,:);
        c2 = c2+1;
    end
end
accuracy = c/3000;

%% test

plot(result1(:,1),result1(:,2),'r.',result2(:,1),result2(
:,2),'b.');
hold on
plot(resulter(:,1),resulter(:,2),'ko');
hold off

%% functions
function finaltest = KNN(K,test,train)
    finaltest = zeros(size(test,1),1);
    for i = 1:size(test,1)
        finaltest(i) =
nearestneighbor(K,test(i,1:2),train);
    end
end

function resul = nearestneighbor(K,test,train)
    distvect = zeros(size(train,1),2);
    for j = 1:size(train,1)
        distvect(j,:) = [ sqrt(((test(1) - train(j,1))^2)
+ ((test(2) - train(j,2))^2))   train(j,3) ];
    end
    mini = sort(distvect(:,1));
    count1 = 0;
    count2 = 0;
    for i = 1:K
```

```
            xc = find(distvect == mini(i));
            if distvect(xc,2) == 1
                count1 = count1+1;
            elseif distvect(xc,2) == 2
                count2 = count2+1;
            end
        end
        if count1 > count2
            resul = 1;
        else
            resul = 2;
        end
end
```

Results:

We were asked to plot test data points, color coded by class label for different values of K. Here are the plots

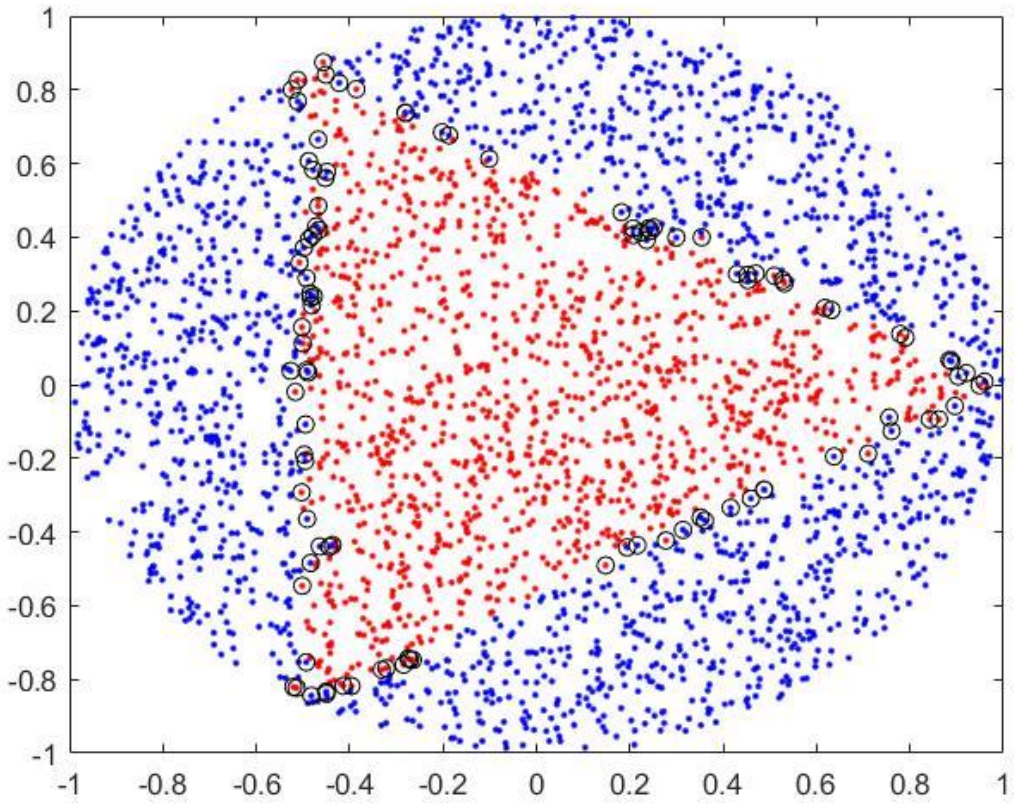


Fig 2. K = 1 (accuracy = 96.77%)

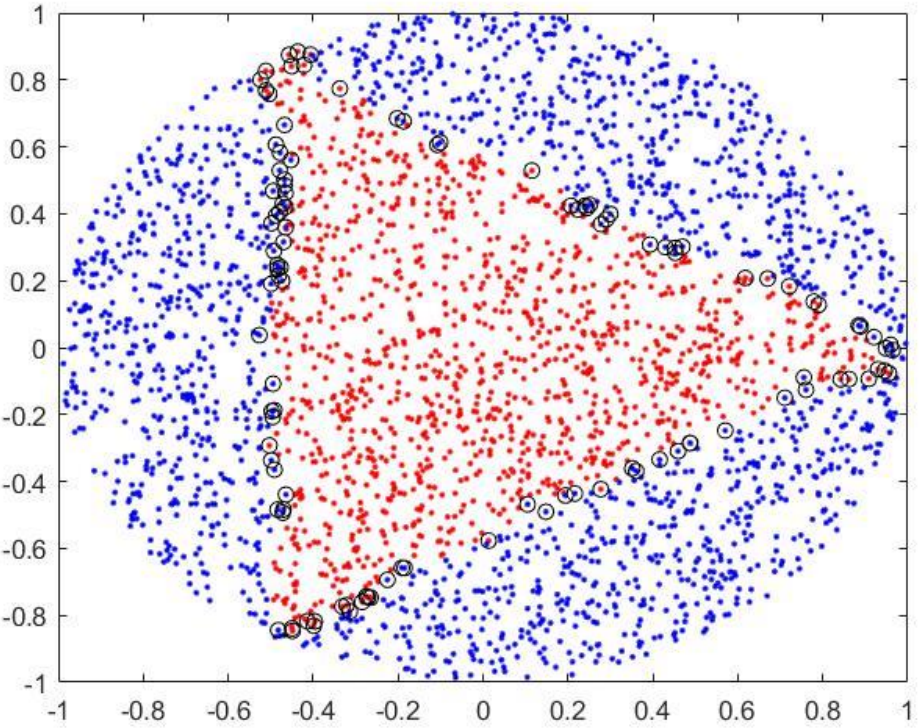


Fig 3. K = 5 (accuracy = 96.33%)

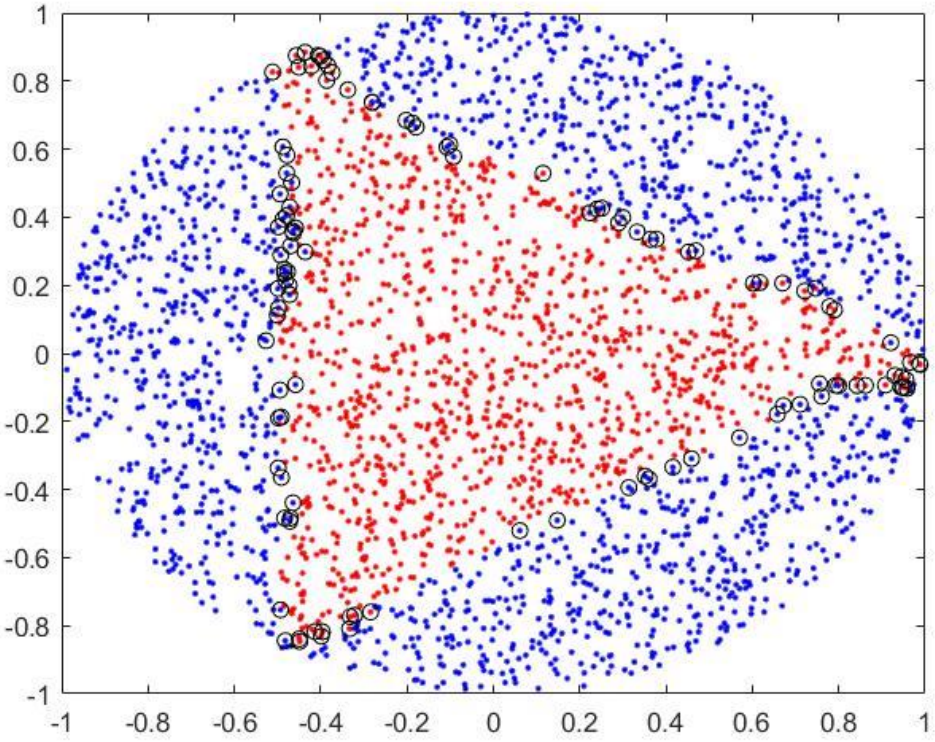


Fig 4. K = 11 (accuracy = 96.27%)

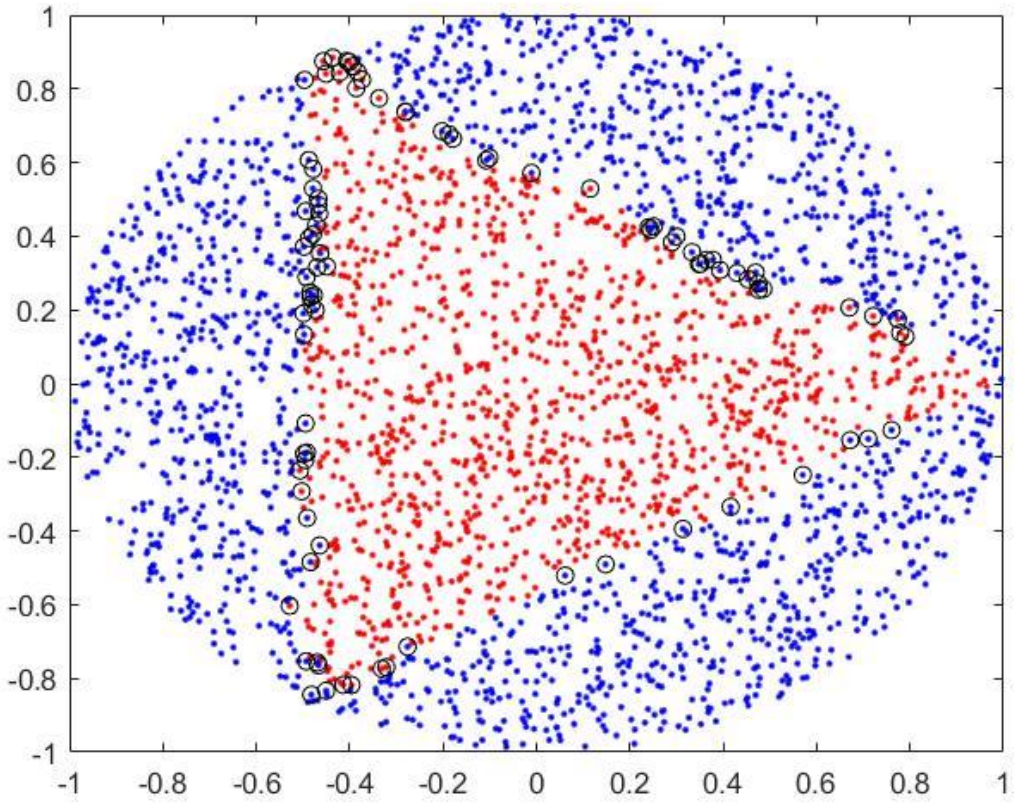


Fig 5. K = 15 (accuracy = 96.9%)

Discussion:

The accuracy we got for different window sizes is as follows

K = 1 Accuracy = 96.77%

K = 5 Accuracy = 96.33%

K = 11 Accuracy = 96.27%

K = 15 Accuracy = 96.90%

Theoretically as we increase the window size(K) the accuracy should increase but, in our case, it slightly decreased then increased. This maybe because the first nearest neighbour is classifying properly in our problem.

In the above plots, the dots surrounded by a circle are the misclassified points according to the actual classes of test points.