EE417 Computer Vision and Image Processing Fall-2019
Hakan Buğra Erentuğ 23637
04/12/19 Post-Lab Report #7

In this assignment, we need to apply stereo vision by looking the correspondence point in both left and right image and calculating depth analysis. Human binocular vision makes us enable to perceive depth using Stereo disparity which refers to the difference in image location of an object seen by the left and right eyes. We apply stereo vision by finding the corresponding window. To see how similar the windows are we do SSD error calculation. Next, we shift the image 1 pixel to the right with accord with offset values and measure varieties between 2 window. And the d1-d2 distance between these corresponding windows window gives us the depth. Below you can find different figures with different window size and offset value.

(Fig. 1 W=3, w=10)

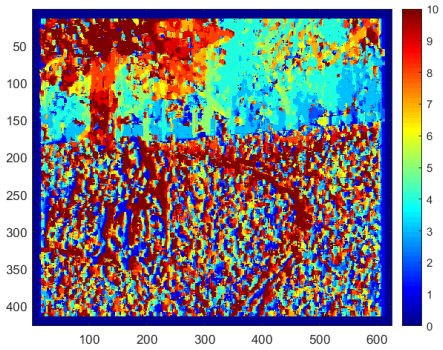
W=3 w=30

W=3 w=60

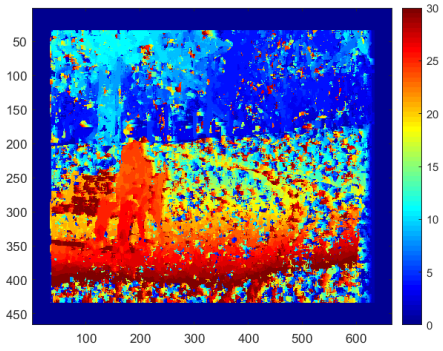
W=3 w=100

And below you can find the disparity maps which formed from these to left and right image, for various offset values.

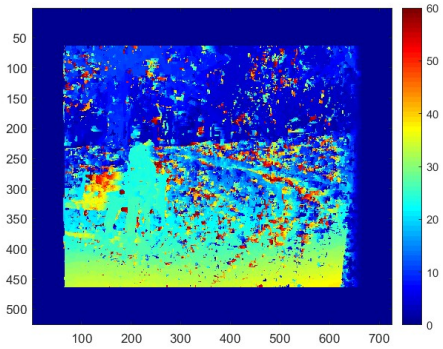
EE417 Computer Vision and Image Processing Fall-2019
Hakan Buğra Erentuğ 23637
04/12/19 Post-Lab Report #7



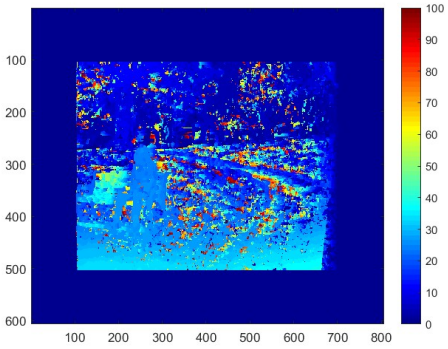
W=3 w=10



W=3 w=30



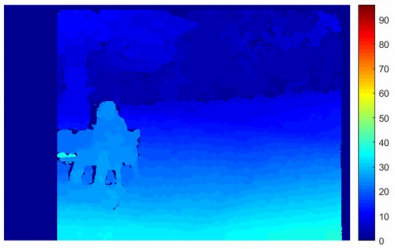
W=3 w=60



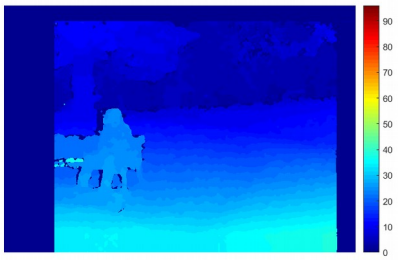
W=3, w=150

As you can see the optimal disparity map is resulted from 30 offset value. If the offset value is too small, many corresponding points can be mismatch. If offset is too high, we started to redundant and inseparable data and low depth values. Below you can also see the other figures which is related with offset value.

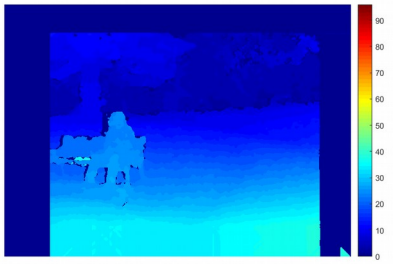
EE417 Computer Vision and Image Processing Fall-2019
Hakan Buğra Erentuğ 23637
04/12/19 Post-Lab Report #7



W=3 , w=10

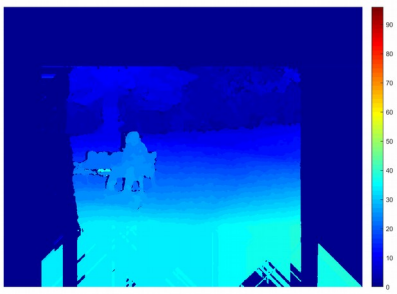


W=3 , w=30



W=3 , w=60

W=3 , w=100



W=3 , w=150

EE417 Computer Vision and Image Processing Fall-2019
Hakan Buğra Erentuğ 23637
04/12/19 Post-Lab Report #7

Below we investigated the quality of stereo vision and window size W.

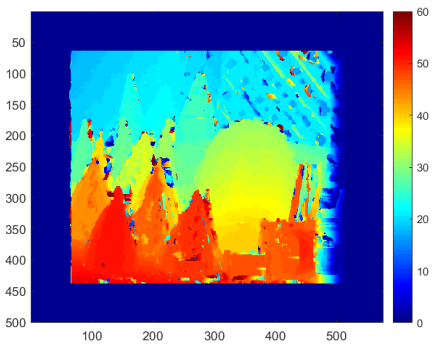
W=3, w=60

W=5, w=60

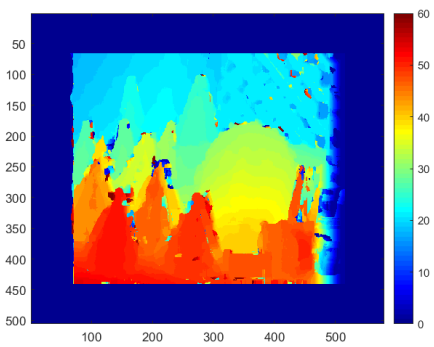
W=13, w=60

W=21 w=60

The all followings images have fixed 60 offset value.

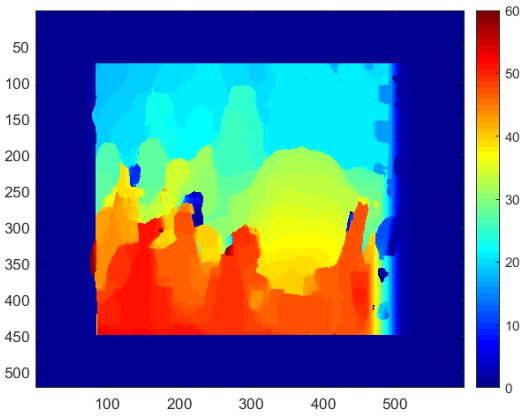


W=3

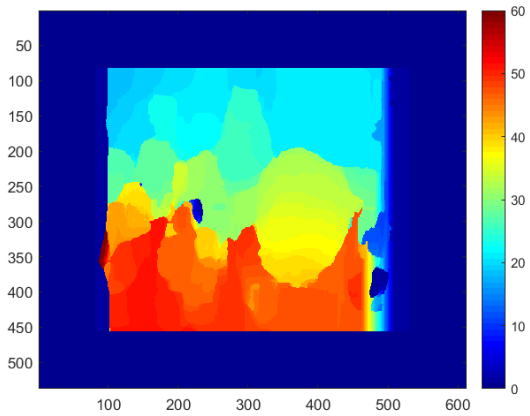


W=5

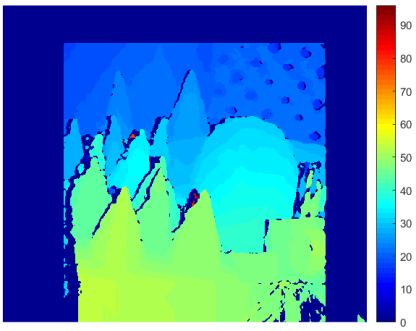
EE417 Computer Vision and Image Processing Fall-2019
Hakan Buğra Erentuğ 23637
04/12/19 Post-Lab Report #7



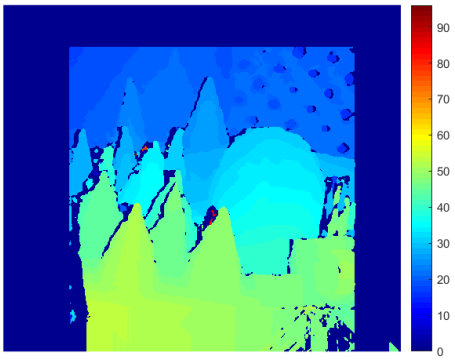
W=13



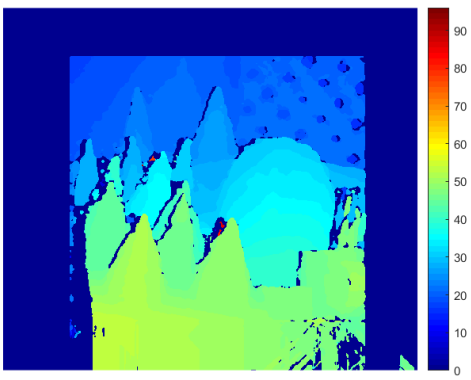
W=21



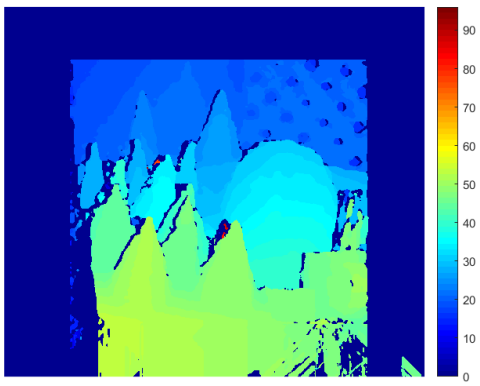
W=3



W=5



W=13



W=21

EE417 Computer Vision and Image Processing Fall-2019
Hakan Buğra Erentuğ 23637
04/12/19 Post-Lab Report #7

As you can see the W window size value can change the details in the sterio image. The large W value can make the details detail and low W value can create many unnecessary details. Unlike the offset value, it seems like there is no significant effect on the depth.

Conclusion:

We analyzed the various images and by looking them we investigate the relation of stereo vision and its parameters. In short, we can say both offset and Window size can be optimal depends on the situation. However, we can say the too large or too low W and w values can create unwanted results. The w offset may play crucial role on depth calculations and disparity maps. The low offset value can struggle to finding correspondence while large offset value can create low depth image due to high correspondences. However, it seems like window size is no directly related to depth of the image. It is related to quality of the details in the disparity maps and both offset and window size have to be hold on the their optimal values.

In below you can see the appendix where my code is placed. Thanks for reading.
Appendix.

```
% Hakan Buğra Erentuğ
close all; clc; clear all;
ImRight = imread('S01R.png');
ImLeft = imread('S01R.png');

[row, col, ch] = size(ImLeft);
W=3;
w = 200;

ImLeft = padarray(ImLeft,[W+w W+w],'both');
ImRight = padarray(ImRight,[W+w W+w],'both');

if (ch==3)
        ImLeft = rgb2gray(ImLeft);
        ImRight = rgb2gray(ImRight);
end

imshow(stereoAnaglyph(ImLeft,ImRight));

ImLeft = double(ImLeft);
ImRight = double(ImRight);

dispar = zeros(size(ImLeft));

for x=W+w+1:1:row+W+w
    for y=W+w+1:1:col+W+w
        dist = zeros(w,3);
        for i = w:-1:1
                SSD = sum(sum((ImLeft(x-W:x+W, y+i-W:y+i+W)-ImRight(x-W:x+W, y-W:y+W)).^2));
                comp = [0, i, SSD];
                dist(i,:) = comp(:);
        end
            ind = find(dist(:,3) == min(dist(:,3)));
            dispar(x,y) = dist(ind(1), 2);
    end
end
figure; imagesc(dispar); colormap jet; colorbar
ImLeft = uint8(ImLeft);
ImRight = uint8(ImRight);

disparityRange = [0, 96];disparityMap = disparity(ImLeft, ImRight,'BlockSize',
7,'DisparityRange',disparityRange);
hold on;

figure; imshow(disparityMap,disparityRange); colormap jet; colorbar
```

EE417 Computer Vision and Image Processing Fall-2019
Hakan Buğra Erentuğ 23637
04/12/19 Post-Lab Report #7