# CS 232 Homework 2

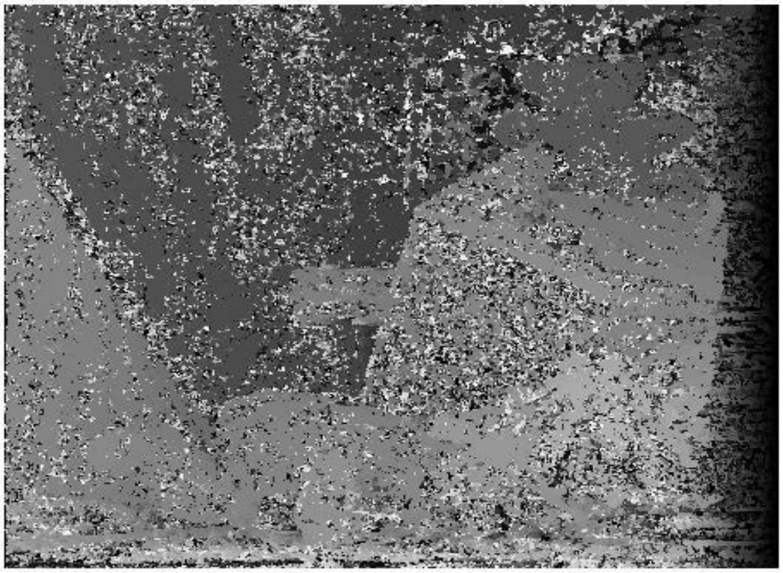
## Part 1a) Disparity for a 3x3 Window

**CODE:**

```
clc
close all;
clear all;

GroundTruth = imread('C:\Users\Jolton\Desktop\Files\Homeworks\Semester3\3D
Computer Vision\Homework2\disp2.pgm');
figure

imshow(GroundTruth);

I_L = imread('C:\Users\Jolton\Desktop\Files\Homeworks\Semester3\3D Computer
Vision\Homework2\teddyL.pgm');
figure
imshow(I_L);

I_R = imread('C:\Users\Jolton\Desktop\Files\Homeworks\Semester3\3D Computer
Vision\Homework2\teddyR.pgm');
figure
imshow(I_R);

I_L_ranked = zeros(size(I_L,1),size(I_L,2));
I_R_ranked = zeros(size(I_R,1),size(I_R,2));

I_L = padarray(I_L,[2 2],0,'both');

I_R = padarray(I_R,[2 2],0,'both');

%% Rank of Left Image
rnk1 = 0;
for i = 3:size(I_L,1)-2
    for j = 3:size(I_L,2)-2
        for k = i-2:i+2
            for l = j-2:j+2
                if I_L(k,l) < I_L(i,j)
                    rnk1 = rnk1+1;
                end
            end
        end
        I_L_ranked(i-2,j-2) = rnk1;
        rnk1 = 0;
    end
end
I_L_ranked = padarray(I_L_ranked,[1 1],0,'both');

%% Rank of Right Image
rnk2 = 0;
for i = 3:size(I_R,1)-2
    for j = 3:size(I_R,2)-2
        for k = i-2:i+2
```

```
            for l = j-2:j+2
                if I_R(k,l) < I_R(i,j)
                    rnk2 = rnk2+1;
                end
            end
        end
        I_R_ranked(i-2,j-2) = rnk2;
        rnk2 = 0;
    end
end
I_R_ranked = padarray(I_R_ranked,[1 1],0,'both');

%% Disparity Map Generation
minimum = 255; brk = 0; c = 1; d = 1; sum = 0; o =1; p = 1; q = 1; SAD_array
= zeros(1,60); disparity_map = zeros(1,2); SAD_min = zeros(1,2);
for i = 1:size(I_R_ranked,1)-2
% for i = 3
    c = i-1;
       for j = 1:size(I_R_ranked,2)-2
%       for j = 2:4
        d = j-1;
        x = j+62;
        if x > size(I_R_ranked,2)-2
            x = size(I_R_ranked,2)-2;
            SAD_array = zeros(1,63-((j+62)-(size(I_R_ranked,2)-2)));
        end
        for k = i:size(I_R_ranked,1)-1
            for l = j:x
                for m = k:k+2
                    c = c+1;
                    for n = l:l+2
                        d = d+1;
                         difference = int16(I_R_ranked(c,d)) -
int16(I_L_ranked(m,n));
                         sum = sum + abs(difference);
                    end
                     d = j-1;
                end
                    if sum > 255
                        sum = 255;
                    end
                    SAD_array(o) = sum;
                    o = o+1;
                    sum = 0;
                    c = i-1;
            end

                SAD_min(q,p) = min(SAD_array);
                [y,SAD_min_index] = find(SAD_array == min(SAD_array));
                disparity_map(q,p) = (j+(min(SAD_min_index)-1)) - p;




                p = p+1;
                SAD_array = zeros(1,63);
                o = 1;
                if j == size(I_R_ranked,2)-2
                    brk = brk+1;
                    break;
                else
                    break;
                end
        end
            if brk == 1
                p = 1;
                q = q+1;
                brk = 0;
                break;
            end
      end
end
figure;
imshow(uint8(disparity_map*4));

%% Error Rate
bad_pixel_count = 0;
GroundTruth = round(GroundTruth/4);
for i = 1:size(GroundTruth,1)
    for j = 1:size(GroundTruth,2)
        if abs(disparity_map(i,j) - GroundTruth(i,j)) > 1
            bad_pixel_count = bad_pixel_count + 1;
        end
    end
end
error_percent = (bad_pixel_count/
(size(GroundTruth,1)*size(GroundTruth,2)))*100;
```

**OUTPUT: The output is multiplied by a scale of 4 to show a disparity map which can be comparable to Ground Truth**



**Error Rate: The error rate in a 3x3 window comes to be 27.4127 %**

**Part 1b) Disparity for a 15x15 window**

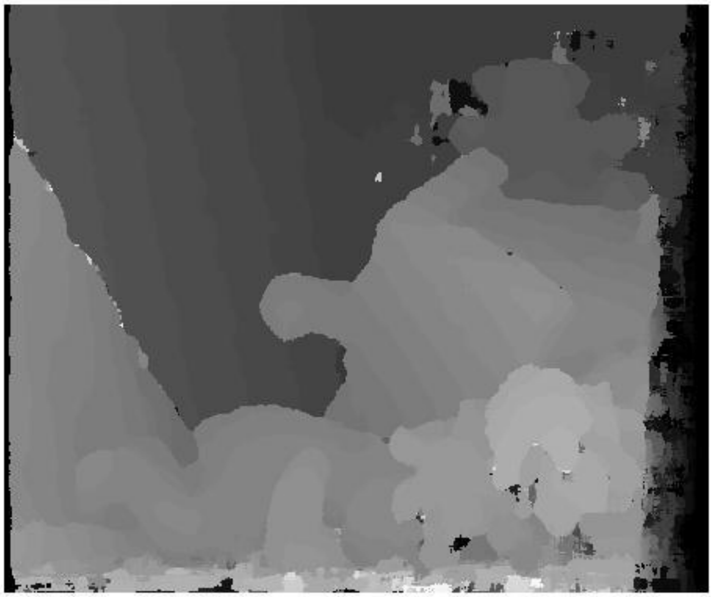
**Code:**

```
clc
close all;
clear all;

GroundTruth = imread('C:\Users\Jolton\Desktop\Files\Homeworks\Semester3\3D
Computer Vision\Homework2\disp2.pgm');
figure
% GroundTruth = int8(GroundTruth);
imshow(GroundTruth);

I_L = imread('C:\Users\Jolton\Desktop\Files\Homeworks\Semester3\3D Computer
Vision\Homework2\teddyL.pgm');
figure
% I_L = int8(I_L);
imshow(I_L);

I_R = imread('C:\Users\Jolton\Desktop\Files\Homeworks\Semester3\3D Computer
Vision\Homework2\teddyR.pgm');
figure
% I_R = int8(I_R);
imshow(I_R);
```

```
I_L_ranked = zeros(size(I_L,1),size(I_L,2));
I_R_ranked = zeros(size(I_R,1),size(I_R,2));

I_L = padarray(I_L,[2 2],0,'both');

I_R = padarray(I_R,[2 2],0,'both');

%% Rank of Left Image
rnk1 = 0;
for i = 3:size(I_L,1)-2
    for j = 3:size(I_L,2)-2
        for k = i-2:i+2
            for l = j-2:j+2
                if I_L(k,l) < I_L(i,j)
                    rnk1 = rnk1+1;
                end
            end
        end
        I_L_ranked(i-2,j-2) = rnk1;
        rnk1 = 0;
    end
end

I_L_ranked = padarray(I_L_ranked,[7 7],0,'both');

%% Rank of Right Image
rnk2 = 0;
for i = 3:size(I_R,1)-2
    for j = 3:size(I_R,2)-2
        for k = i-2:i+2
            for l = j-2:j+2
                if I_R(k,l) < I_R(i,j)
                    rnk2 = rnk2+1;
                end
            end
        end
        I_R_ranked(i-2,j-2) = rnk2;
        rnk2 = 0;
    end
end
I_R_ranked = padarray(I_R_ranked,[7 7],0,'both');


%% Disparity-map Generation
minimum = 255; brk = 0; c = 1; d = 1; sum = 0; o =1; p = 1; q = 1; SAD_array
= zeros(1,63); disparity_map = zeros(1,2); SAD_min = zeros(1,2);
for i = 1:size(I_R_ranked,1)-14
    c = i-1;
       for j = 1:size(I_R_ranked,2)-14
        d = j-1;
        x = j+62;
        if x > size(I_R_ranked,2)-14
           x = size(I_R_ranked,2)-14;
           SAD_array = zeros(1,63-((j+62)-(size(I_R_ranked,2)-7)));
        end
        for k = i:size(I_L_ranked,1)-14
            for l = j:x
                for m = k:k+14
                    c = c+1;
                    for n = l:l+14
                        d = d+1;
                         difference = int16(I_R_ranked(c,d)) -
int16(I_L_ranked(m,n));
                         sum = sum + abs(difference);
                    end
                        d = j-1;
                end

                    SAD_array(o) = sum;
                    o = o+1;
                    sum = 0;
%                     c1 = c
                    c = i-1;
            end

                SAD_min(q,p) = min(SAD_array);
                [y,SAD_min_index] = find(SAD_array == min(SAD_array));
                disparity_map(q,p) = (j+(min(SAD_min_index)-1)) - p;

                p = p+1;
                SAD_array = zeros(1,63);
                o = 1;
                if j == size(I_R_ranked,2)-14
                    brk = brk+1;
                    break;
                else
                    break;
                end
        end
            if brk == 1
                p = 1;
                q = q+1;
                brk = 0;
                break;
            end
      end
end

figure;
imshow(uint8(disparity_map*4));

%% Error Rate
bad_pixel_count = 0;
GroundTruth = round(GroundTruth/4);
for i = 1:size(GroundTruth,1)
    for j = 1:size(GroundTruth,2)
        if abs(disparity_map(i,j) - GroundTruth(i,j)) > 1
            bad_pixel_count = bad_pixel_count + 1;
        end
    end
end
```

```
error_percent = (bad_pixel_count/
(size(GroundTruth,1)*size(GroundTruth,2)))*100
```

**Output: The output is multiplied by a scale of 4 to show a disparity map which can be comparable to Ground Truth**



**Error Rate: The error rate for a 15x15 window comes to be 11.9084 %**

**Part 2) Confidence Match**

**Code:**

```
clc
close all;
clear all;

clc
close all;
clear all;
```

```
GroundTruth = imread('C:\Users\Jolton\Desktop\Files\Homeworks\Semester3\3D
Computer Vision\Homework2\disp2.pgm');
figure

imshow(GroundTruth);

I_L = imread('C:\Users\Jolton\Desktop\Files\Homeworks\Semester3\3D Computer
Vision\Homework2\teddyL.pgm');
figure
imshow(I_L);

I_R = imread('C:\Users\Jolton\Desktop\Files\Homeworks\Semester3\3D Computer
Vision\Homework2\teddyR.pgm');
figure
imshow(I_R);

I_L_ranked = zeros(size(I_L,1),size(I_L,2));
I_R_ranked = zeros(size(I_R,1),size(I_R,2));

I_L = padarray(I_L,[2 2],0,'both');

I_R = padarray(I_R,[2 2],0,'both');

%% Rank Transform of the Left Image
rnk1 = 0;
for i = 3:size(I_L,1)-2
    for j = 3:size(I_L,2)-2
        for k = i-2:i+2
            for l = j-2:j+2
                if I_L(k,l) < I_L(i,j)
                    rnk1 = rnk1+1;
                end
            end
        end
        I_L_ranked(i-2,j-2) = rnk1;
        rnk1 = 0;
    end
end
I_L_ranked = padarray(I_L_ranked,[1 1],0,'both');

%% Rank Transform of the Right Image
rnk2 = 0;
for i = 3:size(I_R,1)-2
    for j = 3:size(I_R,2)-2
        for k = i-2:i+2
            for l = j-2:j+2
                if I_R(k,l) < I_R(i,j)
                    rnk2 = rnk2+1;
                end
            end
        end
        I_R_ranked(i-2,j-2) = rnk2;
        rnk2 = 0;
    end
end
```

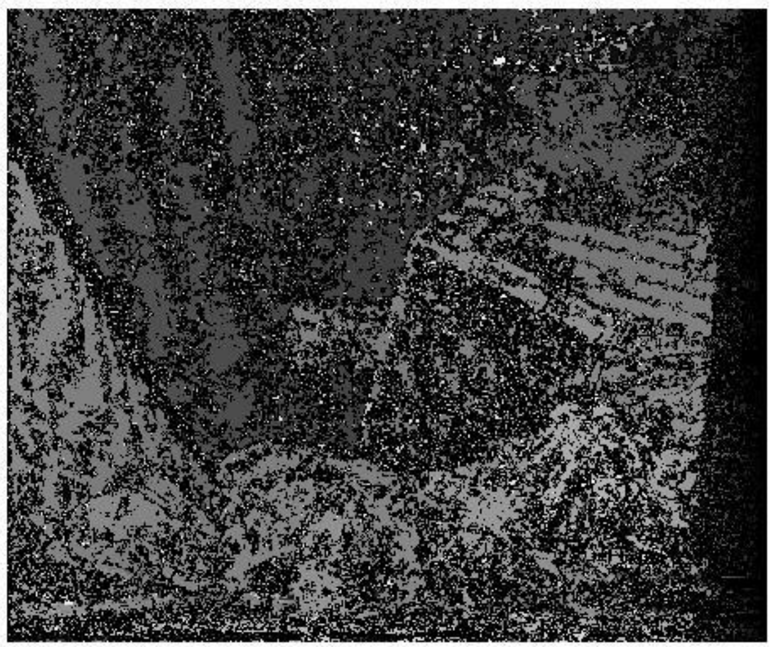
```
I_R_ranked = padarray(I_R_ranked,[1 1],0,'both');

%% Disparity Map Generation
minimum = 255; brk = 0; c = 1; d = 1; sum = 0; o =1; p = 1; q = 1; SAD_array
= zeros(1,60); disparity_map = zeros(1,2); SAD_min = zeros(1,2); Confidence =
zeros(1,2);
for i = 1:size(I_R_ranked,1)-2
% for i = 3
    c = i-1;
       for j = 1:size(I_R_ranked,2)-2
%       for j = 2:4
        d = j-1;
        x = j+62;
        if x > size(I_R_ranked,2)-2
            x = size(I_R_ranked,2)-2;
            SAD_array = zeros(1,63-((j+62)-(size(I_R_ranked,2)-2)));
        end
        for k = i:size(I_R_ranked,1)-1
            for l = j:x
                for m = k:k+2
                    c = c+1;
                    for n = l:l+2
                        d = d+1;
                         difference = int16(I_R_ranked(c,d)) -
int16(I_L_ranked(m,n));
                         sum = sum + abs(difference);
                    end
                     d = j-1;
                end
                    if sum > 255
                        sum = 255;
                    end
                    SAD_array(o) = sum;
                    o = o+1;
                    sum = 0;
                    c = i-1;
            end
% Confidence Matrix Generation
                SAD_min(q,p) = min(SAD_array);
                c1 = SAD_min(q,p);
                SAD_array_sort = sort(unique(SAD_array));
                if length(SAD_array_sort) == 1
                    c2 = SAD_array_sort(1);
                else
                    c2 = SAD_array_sort(2);
                end

                Confidence(q,p) = c1/c2;
%
                [x,SAD_min_index] = find(SAD_array == min(SAD_array));
                disparity_map(q,p) = (j+(min(SAD_min_index)-1)) - p;
%
                p = p+1;
                SAD_array = zeros(1,63);
                o = 1;
                if j == size(I_R_ranked,2)-2
```

```
                    brk = brk+1;
                    break;
                else
                    break;
                end
        end
            if brk == 1
                p = 1;
                q = q+1;
                brk = 0;
                break;
            end
      end
end


Confidence_vec = Confidence(:);
Confidence_median = median(Confidence_vec);
disparity_map2 = disparity_map;
no_of_pixels = 0; ind = 1; r = zeros(2,1); s = zeros(2,1);

for i = 1:size(disparity_map,1)
    for j = 1:size(disparity_map,2)
        if Confidence(i,j) < Confidence_median
            r(ind) = i; s(ind) = j;
            disparity_map2(i,j) = 0;
            ind = ind+1;
        else
            no_of_pixels = no_of_pixels + 1;
        end
    end
end

fprintf('Number of pixels included is %d', no_of_pixels);
figure;
imshow(uint8(disparity_map*4));
figure;
imshow(uint8(disparity_map2*4));

%% Error Rate
bad_pixel_count = 0;
GroundTruth = round(GroundTruth/4);
for i = 1:length(r)
    if abs(GroundTruth(r(i),s(i)) - disparity_map2(r(i),s(i))) > 1
        bad_pixel_count = bad_pixel_count + 1;
    end
end
error_percent = (bad_pixel_count/
(size(GroundTruth,1)*size(GroundTruth,2)))*100;
fprintf('\nThe error percentage is %f', error_percent);
```

**Output1: For c2/c1 confidence match,**

**The number of pixels included is 85048**

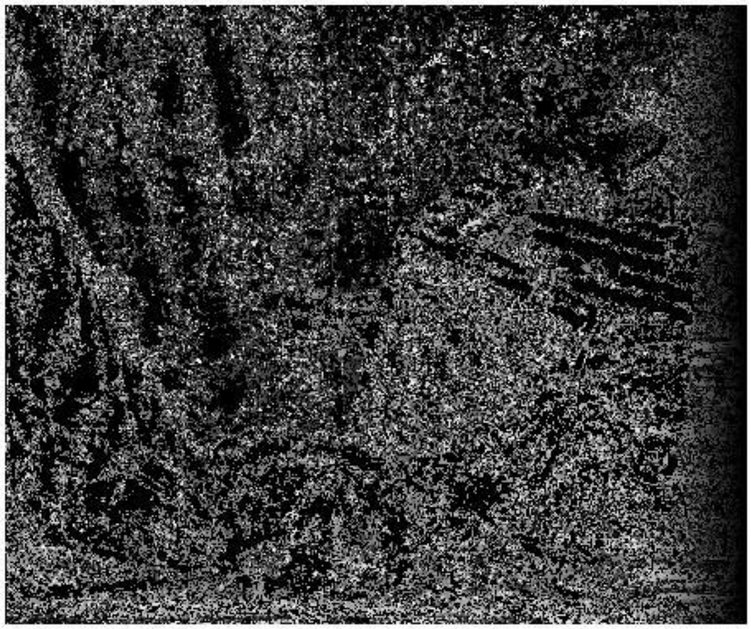
**The error percentage is 48.467556**



**Output 2: For confidence match c1/c2**

**The number of pixels included is 84500**

**The error percentage is 49.054222**



**NOTE:**

In the first part of the homework, we need to find the disparity map of the left and the right images. First, we find the rank transform of the left and the right images and store them as ranked left and ranked right image respectively. Using these ranked images, we keep the right image as the reference and slide a window of 3x3 size on the left ranked image till 63 values and find the SAD for each pixel of the right reference image. We are interested in the minimum of the SAD values for each of the pixel of the reference image. Wherever the minimum value of the SAD is found for a pixel on the reference image, we deduct the x location of the minimum value from the x location of the reference image. This is the disparity for the corresponding pixel of the reference image. We find the disparity map by putting these values in a matrix. Similarly, we do the same for a 15x15 window and find out the error rates for both the windows.

In the second part, we do the following steps

1) Find the confidence for each pixel by formula c2/c1 (Here c2 is the second minimum value of SAD)

2) Find the median of the confidence values

3) For the confidence values greater than the median, store the location of those values

4) Go to the 3x3 disparity map and keep the values at the location of the values you found in step 3 and make rest values in the disparity map 0

5) Find the error rate only for the pixels you kept, comparing them with the ground truth.