**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;

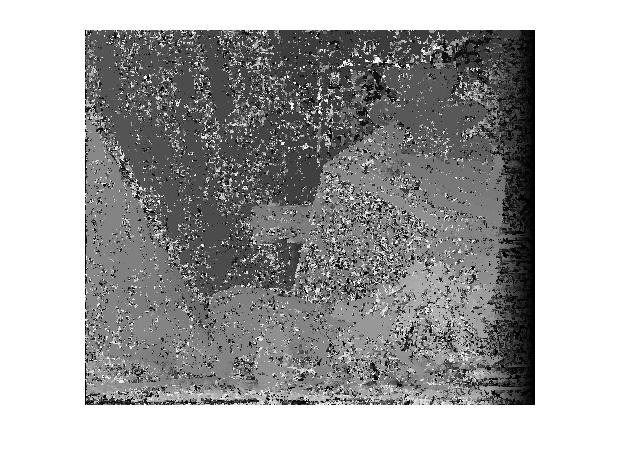
end

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;

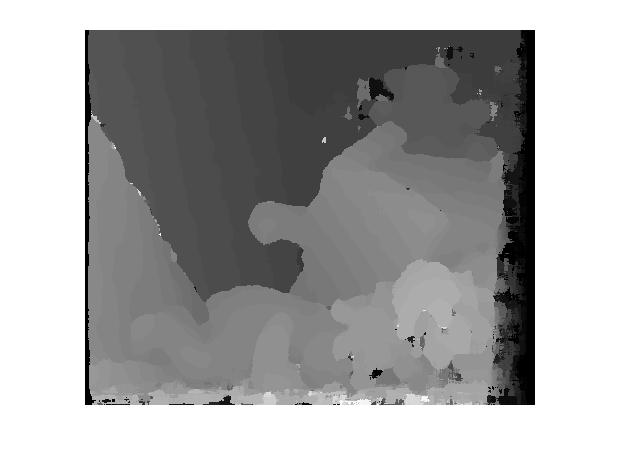
end

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

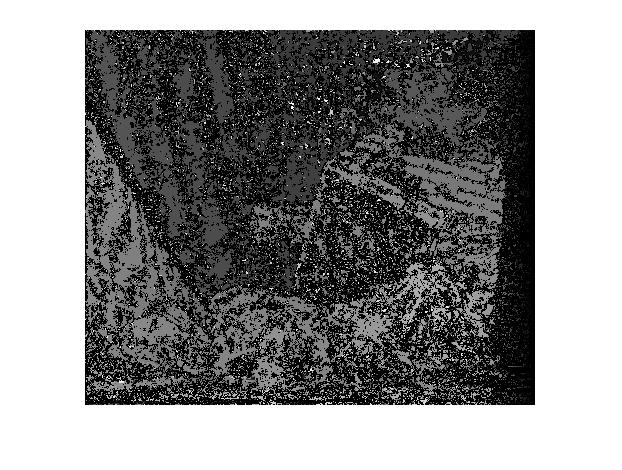
error\_percent = (bad\_pixel\_count/ (size(GroundTruth,1)\*size(GroundTruth,2)))\*100;

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.