

HW2

Problem 1: Adaptive histogram equalization

The resulting image with different window size is attached below

- The original image



- Image after global histogram equalization and AHE

hist



AHE w = 33



AHE w = 65



AHE w = 129



- How does the original image qualitatively compare to the image after AHE and HE respectively.

HE and AHE both enhance the contrast of the original image. We can observe more details in the

enhanced images. For all the image after enhancement, they look brighter compared to the original one. However, all the images after histogram equalization seems unreal compared to the original one

- **Which strategy (AHE or HE) works best for beach.png and why? Is this true for any image in general?**

AHE works better on the beach.png. I think the reason probably would be that the image contains regions that are significantly lighter or darker than most of the image. Thus, the normal histogram equalization does not work very well.

This conclusion will not always hold in general. Since AHE only operates on a small region, when the pixels in the neighborhood are very similar, the mapping function will map the narrow range of pixel values to the entire intensity range in the resulting image. In this case, AHE tends to enhance subtle area but are very sensitive to the noise. Thus, it will not work very well for the noisy image.

Source Code

- **AHE.m(the adaptive histogram equalization function)**

```
1 % This function is designed to perform the adaptive histogram equalization
2 % Before the operation, it will first pad the image based on the window
3 % size
4 % for each pixel, it will perform histogram equalization around the certain
5 % regions. The region is defined to be a square , it's size always equals
6 % to win_size
7 function [output] = AHE(image, win_size)
8     %pad the image based on the window size
9     pad_size = floor(win_size/2);
10    paddedImage = padarray(image, [pad_size,pad_size], 'symmetric');
11    [height,width] = size(paddedImage);
12    output= uint8(zeros(size(image,1), size(image, 2)));
13    %perform Adaptive histogram equalization
14    for x = 1 + pad_size : height - pad_size
15        for y = 1 + pad_size : width - pad_size
16            rank = 0;
17            %iterate through the window around center pixel
18            for i = x - pad_size : x + pad_size
19                for j = y - pad_size : y + pad_size
20                    if paddedImage(x,y) > paddedImage(i,j)
21                        rank = rank + 1;
22                    end
23                end
24            end
25            intensity = 255 * (rank/(win_size * win_size));
26            output(x- pad_size, y - pad_size) = intensity;
27        end
28    end
29 end
30 ```
31
32 * **P1.m(script for problem 1)**
33
```

- **P1.m(script for problem 1)**

```
1
2 image = imread('beach.png');
3 imshow(image);
4 win_size = 129;
5 pad_size = floor(win_size/2);
6 disp(pad_size);
7 paddedImage = padarray(image,[pad_size,pad_size],'symmetric');
8 [height,width] = size(paddedImage);
9 output= uint8(zeros(size(image,1), size(image, 2)));
10
11 padImage33 = AHE(image,33);
12 padImage65 = AHE(image, 65);
13 padImage129 = AHE(image, 129);
14 histEq = histeq(image);
15 figure, subplot(2,2,1)
16 imshow(histEq);
17 title('hist')
18 subplot(2,2,2);
19 imshow(padImage33);
20 title('AHE w = 33')
21 subplot(2,2,3);
22 imshow(padImage65);
23 title('AHE w = 65')
24 subplot(2,2,4);
25 imshow(padImage129);
26 title('AHE w = 129')
```