# **Question 3**

Part-a 1

Part-b 1

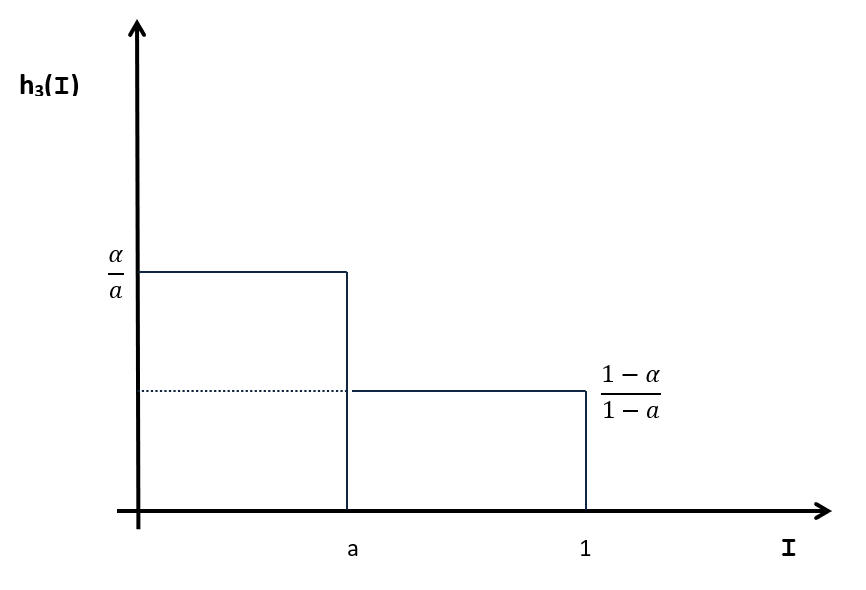
Part-c 2

Part-d 2

## Part-a

We know that in the continuous case, Histogram Equalization (HE) will convert the histogram (or equivalently, PDF) of the image to a uniform distribution (i.e. constant in the interval of interest) while preserving the mass.

Now the weights of the two parts are given to be α and (1-α) respectively. That is, the area under each of the two parts of the histograms are α and (1-α). Using this, it is easy to see that the resulting histogram will be of the following form:



Now, to get the mean intensity of the resulting histogram h3, we apply the formula of expectation:

Thus

## Part-b

Since *a* is the median of *h(I*), we have

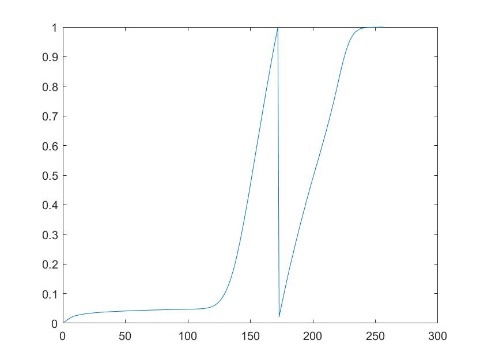
Now the process in the question will preserve the probability mass. Therefore

Using this, and *h3* as above, we get, since the mean and median are the same,

Therefore, using the result of Part-a, we get the average intensity of the resulting image to be .

## Part-c

Such a transformation will be useful to *retain* any peaks present in the histogram of an image. Looking at Part-a, the transformation function that will be applied to the image, which is the CDF of the histogram h3 will look something like:



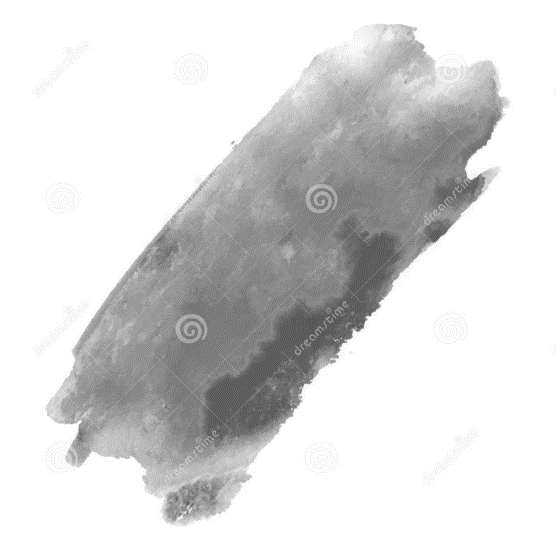
Thus, the peak before the median will be retained, and that after will be equalized as per HE, but mapped to a different range of intensities. The median intensity itself will be darkened, which will also help distinguish this.

Such a transformation would be particularly useful if there is a peak at either corner of the intensity range, for example an image with a white (or dark) background. Such an image, under our transformation, would have this white peak (background) retained with the other part of the image being contrast-enhanced, as opposed to vanilla HE ‘spreading’ the peak intensity over the entire image.

## Part-d

For simplicity, we work directly with a grayscale image. For this, the (saved as) RGB image has been converted to grayscale.

The image chosen is: (<https://thumbs.dreamstime.com/z/abstract-watercolor-grayscale-background-abstract-watercolor-grayscale-background-vector-illustration-grunge-texture-cards-102081554.jpg>). Note – this is not the actual dimension of the image used in the code, however it will work irrespective of that.



### Code for Part-d

Convert the image to grayscale for simplicity

im1 = imread('../data/q3.png');  
im = rgb2gray(im1);  
  
[m,n] = size(im);  
im\_min = double(min(im,[],'all'));  
im\_max = double(max(im,[],'all'));

Get histogram

h = zeros(1,im\_max+1);  
for i=1:m  
 for j=1:n  
 h(im(i,j)+1)=h(im(i,j)+1)+1;  
 end  
end

Normalize the histogram

h = double(h);  
h = h./sum(h);

Let r = point to break histogram('r' fraction of probability mass) Set r=0.5 for median

r=0.5;  
hist\_break = im\_max;  
prob\_sum = h(end);  
while(prob\_sum<r)  
 prob\_sum = prob\_sum + h(hist\_break);  
 hist\_break = hist\_break - 1;  
end

Now break the histogram into two parts a and b

h\_a = h(1:hist\_break);  
h\_b = h(hist\_break+1:end);

Normalize the two histograms Note: this is necessary since the intensity must be mapped to a 'correct' value. To see why this is the case, consider the example of an image with the corner case that h\_b is of length 1, with weight 0.5 at im\_max. If we do not normalize, the peak (in this case just the one value) will be retained but this intensity will get mapped to im\_max/2, which is not what we desire.

h\_a = h\_a./sum(h\_a);  
h\_b = h\_b./sum(h\_b);

Now get CDFs of the two parts

cdf\_a = zeros(1,length(h\_a));  
cdf\_a(1)= h\_a(1);  
i=2;  
while(i<=length(h\_a))  
 cdf\_a(i) = h\_a(i) + cdf\_a(i-1);  
 i=i+1;  
end

cdf\_b = zeros(1,length(h\_b));  
cdf\_b(1)= h\_b(1);  
i=2;  
while(i<=length(h\_b))  
 cdf\_b(i) = h\_b(i) + cdf\_b(i-1);  
 i=i+1;  
end

Final transformation to be applied is obtained by combining the individual CDFs.

cdf = [cdf\_a cdf\_b];

Apply transformation to get the new image

newim = zeros(m,n);  
for i = 1:m  
 for j = 1:n  
 newim(i,j) = 0.005\*im\_max\*cdf(im(i,j)+1);  
 end  
end  
figure()  
subplot(1,2,1),imshow(histeq(im)), title("Vanilla Histogram Equalization");  
subplot(1,2,2),imshow(newim), title("Piece-wise Histogram Equalization");

### Result

### 

Note: the vanilla HE image has been obtained by the inbuilt function *histeq()* for brevity. The same result will be obtained using the code in Question-2.

Clearly, piece-wise HE works better in this image in highlighting the contrast of the part in the foreground, since there is a peak of bright intensities in the backgroung of the image, and we wish to ignore that part while performing equilization.

[*Published with MATLAB® R2020a*](https://www.mathworks.com/products/matlab)