Image Processing Lab

Batch 9:

Ajay Viswanathan [09EC3509]

Sourin Sutradhar [09EC3514]

Experiment 7

Histogram Equalization

Problem Objective:

- Write C++/Image- modular functions to perform histogram equalization on the 512 × 512 grayscale Lena image. Perform the same on any two low-contrast and dark colored images each.
- Display the histograms of the original image and the enhanced images and document the observations.

Brief Theory:

This method usually increases the global contrast of many images, especially when the usable data of the image is represented by close contrast values. Through this adjustment, the intensities can be better distributed on the histogram. This allows for areas of lower local contrast to gain a higher contrast. Histogram equalization accomplishes this by effectively spreading out the most frequent intensity values.

The method is useful in images with backgrounds and foregrounds that are both bright or both dark. In particular, the method can lead to better views of bone structure in x-ray images, and to better detail in photographs that are over or under-exposed. A key advantage of the method is that it is a fairly straightforward technique and an invertible operator. So in theory, if the histogram equalization function is known, then the original histogram can be recovered. The calculation is not computationally intensive**. A disadvantage of the method is that it is indiscriminate. It may increase the contrast of background noise, while decreasing the usable signal.**

In scientific imaging where spatial correlation is more important than intensity of signal (such as separating DNA fragments of quantized length), the small signal to noise ratio usually hampers visual detection.

Histogram equalization often produces unrealistic effects in photographs; however it is very useful for scientific images like thermal, satellite or x-ray images, often the same class of images that user would apply false-color to. Also histogram equalization can produce undesirable effects (like visible image gradient) when applied to images with low color depth. For example, if applied to 8-bit image displayed with 8-bit gray-scale palette it will further reduce color depth (number of unique shades of gray) of the image. Histogram equalization will work the best when applied to images with much higher color depth than palette size, like continuous data or 16-bit gray-scale images.

There are two ways to think about and implement histogram equalization, either as image change or as palette change. The operation can be expressed as P(M(I)) where I is the original image, M is histogram equalization mapping operation and P is a palette. If we define a new palette as P'=P(M) and leave image I unchanged then histogram equalization is implemented as palette change. On the other hand if palette P remains unchanged and image is modified to I'=M(I) then the implementation is by image change. In most cases palette change is better as it preserves the original data.

Generalizations of this method use multiple histograms to emphasize local contrast, rather than overall contrast. Examples of such methods include adaptive histogram equalization and contrast limiting adaptive histogram equalization or CLAHE.

Histogram equalization also seems to be used in biological neural networks so as to maximize the output firing rate of the neuron as a function of the input statistics. This has been proved in particular in the fly retina.

Histogram equalization is a specific case of the more general class of histogram remapping methods. These methods seek to adjust the image to make it easier to analyze or improve visual quality (e.g., retinex)

Example:

The following is the same 8x8 subimage as used in JPEG. The 8-bit greyscale image shown has the following values:

$$\begin{bmatrix} 52 & 55 & 61 & 66 & 70 & 61 & 64 & 73 \\ 63 & 59 & 55 & 90 & 109 & 85 & 69 & 72 \\ 62 & 59 & 68 & 113 & 144 & 104 & 66 & 73 \\ 63 & 58 & 71 & 122 & 154 & 106 & 70 & 69 \\ 67 & 61 & 68 & 104 & 126 & 88 & 68 & 70 \\ 79 & 65 & 60 & 70 & 77 & 68 & 58 & 75 \\ 85 & 71 & 64 & 59 & 55 & 61 & 65 & 83 \\ 87 & 79 & 69 & 68 & 65 & 76 & 78 & 94 \end{bmatrix}$$

The histogram for this image is shown in the following table. Pixel values that have a zero count are excluded for the sake of brevity.

Value	Count	Value	Count	Value	Count	Value	Count	Value	Count
52	1	64	2	72	1	85	2	113	1
55	3	65	3	73	2	87	1	122	1
58	2	66	2	75	1	88	1	126	1
59	3	67	1	76	1	90	1	144	1
60	1	68	5	77	1	94	1	154	1
61	4	69	3	78	1	104	2		
62	1	70	4	79	2	106	1		
63	2	71	2	83	1	109	1		

The cumulative distribution function (cdf) is shown below. Again, pixel values that do not contribute to an increase in the cdf are excluded for brevity.

Value	cdf	Value	cdf	Value	cdf	Value	cdf	Value	cdf
52	1	64	19	72	40	85	51	113	60
55	4	65	22	73	42	87	52	122	61
58	6	66	24	75	43	88	53	126	62
59	9	67	25	76	44	90	54	144	63
60	10	68	30	77	45	94	55	154	64
61	14	69	33	78	46	104	57		
62	15	70	37	79	48	106	58		
63	17	71	39	83	49	109	59		

This cdf shows that the minimum value in the subimage is 52 and the maximum value is 154. The cdf of 64 for value 154 coincides with the number of pixels in the image. The cdf must be normalized to [0,255]. The general histogram equalization formula is:

$$h(v) = \mathrm{round}\left(\frac{cdf(v) - cdf_{min}}{(M \times N) - cdf_{min}} \times (L-1)\right)$$

Where cdf_{min} is the minimum value of the cumulative distribution function (in this case 1), M × N gives the image's number of pixels (for the example above 64, where M is width and N the height) and L is the number of grey levels used (in most cases, like this one, 256). The equalization formula for this particular example is:

$$h(v) = \mathrm{round}\left(\frac{cdf(v) - 1}{63} \times 255\right)$$

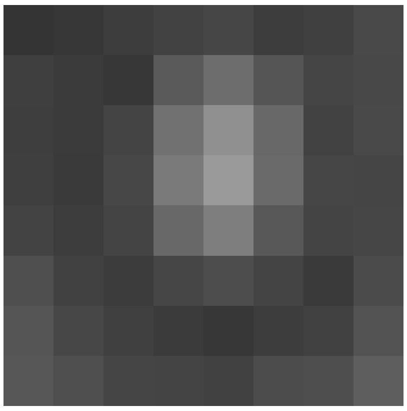
For example, the cdf of 78 is 46. (The value of 78 is used in the bottom row of the 7th column.) The normalized value becomes

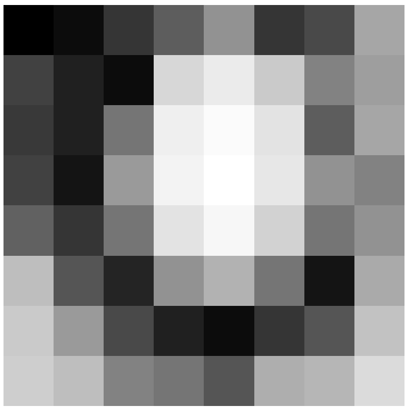
$$h(78) = \mathrm{round}\left(\frac{46-1}{63} \times 255\right) = \mathrm{round}\left(0.714286 \times 255\right) = 182$$

Once this is done then the values of the equalized image are directly taken from the normalized cdf to yield the equalized values:

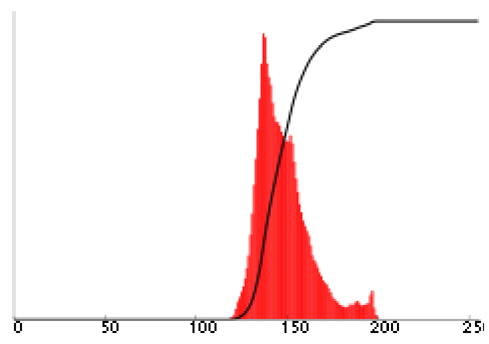
$$\begin{bmatrix} 0 & 12 & 53 & 93 & 146 & 53 & 73 & 166 \\ 65 & 32 & 12 & 215 & 235 & 202 & 130 & 158 \\ 57 & 32 & 117 & 239 & 251 & 227 & 93 & 166 \\ 65 & 20 & 154 & 243 & 255 & 231 & 146 & 130 \\ 97 & 53 & 117 & 227 & 247 & 210 & 117 & 146 \\ 190 & 85 & 36 & 146 & 178 & 117 & 20 & 170 \\ 202 & 154 & 73 & 32 & 12 & 53 & 85 & 194 \\ 206 & 190 & 130 & 117 & 85 & 174 & 182 & 219 \end{bmatrix}$$

Notice that the minimum value (52) is now 0 and the maximum value (154) is now 255.

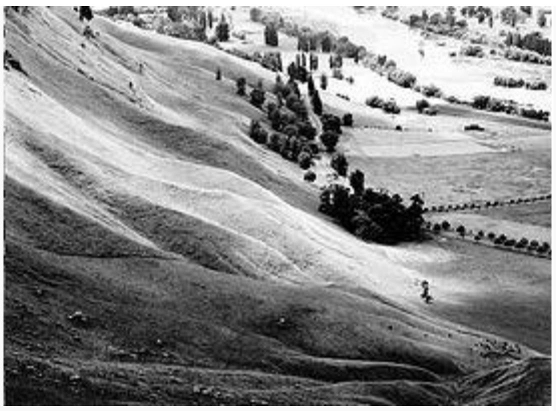

Original


Equalized

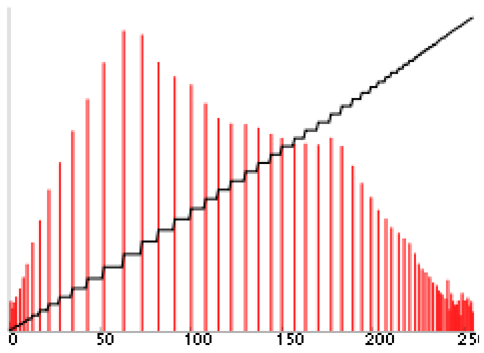
An unequalized image



Corresponding histogram (red) and cumulative histogram (black)



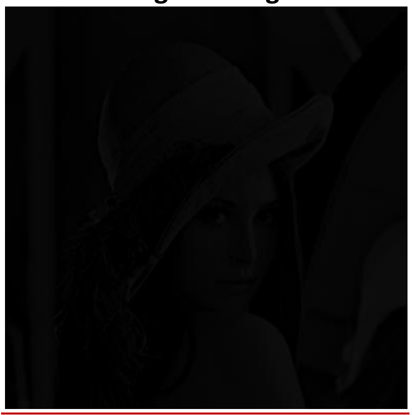
The same image after histogram equalization



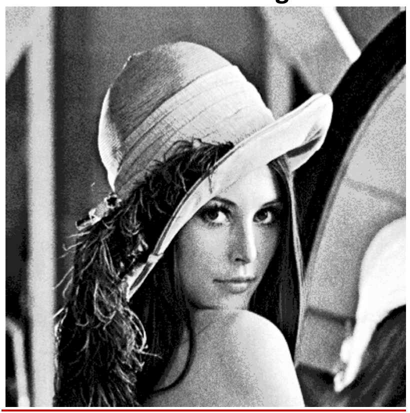
Corresponding histogram (red) and cumulative histogram (black)

Results:

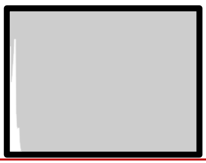
Original Image

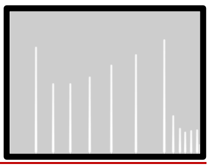


Enhanced Image

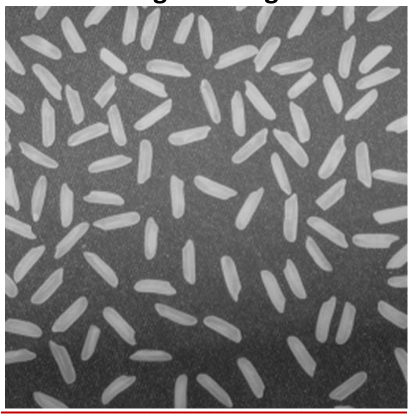


Histogram

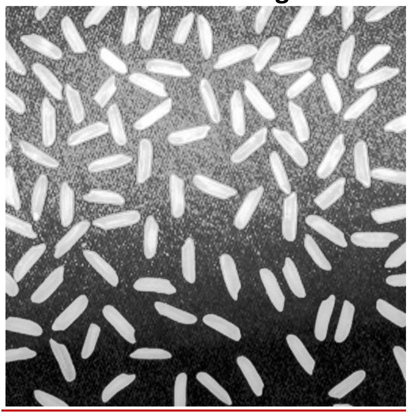




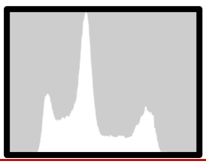
Original Image

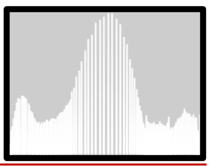


Enhanced Image

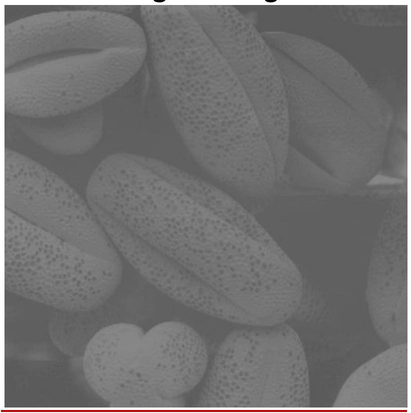


Histogram

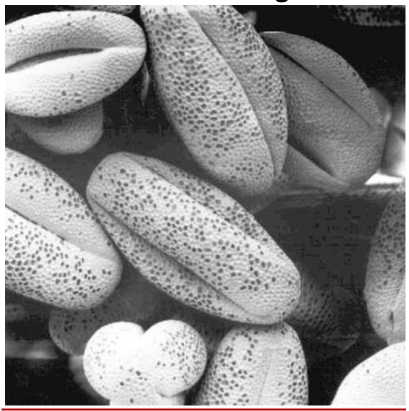




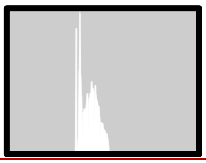
Original Image

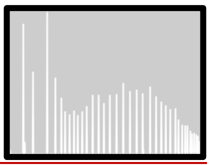


Enhanced Image

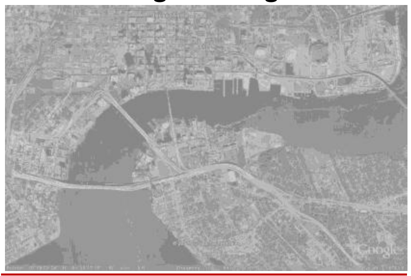


Histogram

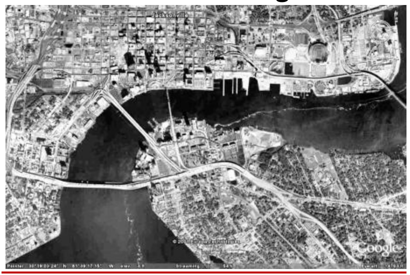




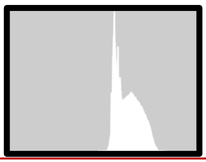
Original Image

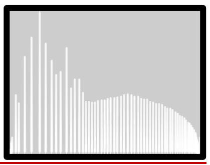


Enhanced Image



Histogram





Discussion and Inference:

Histogram modeling is usually introduced using continuous, rather than discrete, process functions. Therefore, we suppose that the images of interest contain continuous intensity levels (in the interval [0,1]) and that the transformation function *f* which maps an input image $A(x,y)$ onto an output image $B(x,y)$ is continuous within this interval. Further, it will be assumed that the transfer law (which may also be written in terms of intensity density levels, *e.g.* $D_B = f(D_A)$) is single-valued and monotonically increasing (as is the case in histogram equalization) so that it is possible to define the inverse law $D_A = f^{-1}(D_B)$. An example of such a transfer function is illustrated in Figure 1.

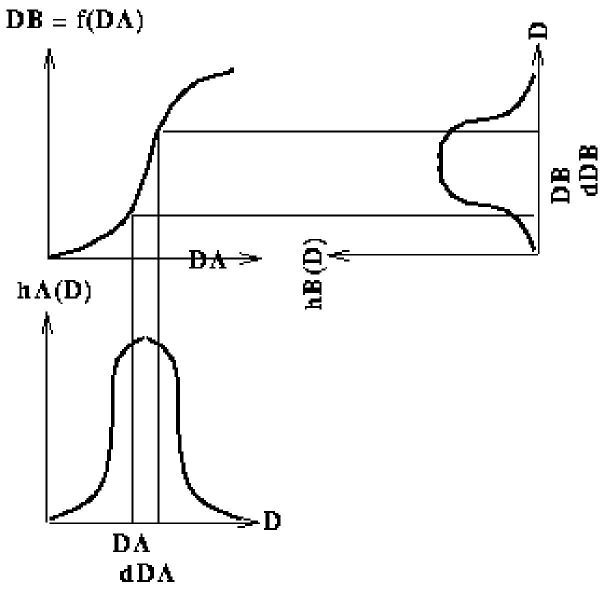


Figure 1 A histogram transformation function.

All pixels in the input image with densities in the region D_A to $D_A + dD_A$ will have their pixel values re-assigned such that they assume an output pixel density value in the range from D_B to $D_B + dD_B$. The surface areas $h_A(D_A)dD_A$ and $h_B(D_B)dD_B$ will therefore be equal, yielding:

$$h_B(D_B) = h_A(D_A) \div d(D_A)$$

where $d(x) = \frac{df(x)}{dx}$.

This result can be written in the language of probability theory if the histogram *h* is regarded as a continuous probability density function *p* describing the distribution of the (assumed random) intensity levels:

$$p_B(D_B) = p_A(D_A) \div d(D_A)$$

In the case of histogram equalization, the output probability densities should all be an equal fraction of the maximum number of intensity levels in the input image D_M (where the minimum level considered is 0). The transfer function (or point operator) necessary to achieve this result is simply:

$$d(D_A) = D_M * p_A(D_A)$$

Therefore,

$$f(D_A) = D_M \int_0^{D_A} p_A(u)du = D_M * F_A(D_A)$$

where $F_A(D_A)$ is simply the cumulative probability distribution (*i.e.* cumulative histogram) of the original image. *Thus, an image which is transformed using its cumulative histogram yields an output histogram which is flat!*

A digital implementation of histogram equalization is usually performed by defining a transfer function of the form:

$$f(D_A) = max(0, round[D_M * n_k/N^2)] - 1)$$

where *N* is the number of image pixels and nk is the number of pixels at intensity level k or less.