International Journal of Computing Science and Information Technology, **2018**
ISSN: 2278-9669, April 2018 (http://ijcsit.org)

A Study of Image Enhancement Techniques Using Histogram Processing.

Harish Naik T/ Dr.Suresha M/Amalorpavam G
[1]Assistant Professor, Department of Computer Applications, Presidency College, Karnataka, India.
harishtkola@gmail.com
[2]Assistant Professor, Department of Computer Science, Kuvempu University, Karnataka, India.
srit_suresh@yahoo.com
[3]Assistant Professor, Department of MCA, Sambhram Academy of Management Studies, Karnataka, India.
george.amala7@gmail.com

Abstract

This study primarily centers on Image Processing. The principal objective of enhancement techniques is to process an image so that the result is more suitable than the original image for a specific application, the word specific is important, because it establishes at the outset that the techniques are very much problem oriented. Thus, for example, a method that is quite useful for enhancing x-ray image may not necessarily be the best approach for enhancing pictures of Mars transmitted by a space probe. Image enhancement is a challenging problem in digital image processing because of its usefulness in all image processing applications. Some commonly used image processing techniques are discussed in this study.

1. Introduction

In image enhancement, the goal is to accentuate certain image features for subsequent analysis or for image display. Examples include contrast and edge enhancement, pseudo coloring, noise filtering, sharpening, and magnifying. Image enhancement is useful in feature extraction, image analysis, and visual information display. The enhancement process itself does not increase the inherent information content in the data. It simply emphasizes certain specified image characteristics. Enhancement algorithms are generally interactive and application-dependent. Image enhancement techniques, such as contrast stretching, map each gray level into another gray level by a predetermined transformation. An example is the histogram equalization method, where the input gray levels are mapped so that the output gray level distribution is uniform. This has been found to be a powerful method of enhancement of low contract image.

Image enhancement is among the simplest and most appealing area of digital image processing. Basically, the idea behind enhancement techniques is to bring out detail that is obscured, or simply to highlight certain features of interest in an image. The greatest difficulty in image enhancement is quantifying the criteria for enhancement; therefore a large number of image enhancement techniques are empirical and require interactive procedure to obtain satisfactory result. However, image enhancement remains a very important topic because of its usefulness is virtually all image processing applications.

2. Image enhancement operations

1. Point operations
2. Spatial operations.

2.1 Enhancement by point processing

We begin the study of image enhancement techniques by considering processing methods that are based only on the intensity of single pixels.

2.2 Some simple Intensity Transformation

1. **Contrast stretching (adjusting the brightness)**
Low contrast images can result from poor illumination, lack of dynamic range in the image sensor, or even wrong setting of a lens aperture during image acquisition. The idea behind contrast stretching is to increase the dynamic range of the gray level in the image being processed.

2.3 Gray level slicing
Highlighting a specific range of gray levels in the image often is desired. There are several ways of doing level slicing, but most of them of are variation of two basic themes. One approach is to display a high value for all gray levels in the range of interest and a low value for all other gray levels. The second approach based on the transformation brightness the desired range of gray

levels but preserves the background and gray level tonalities in the image.

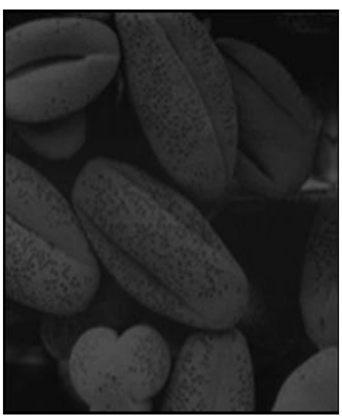
2.4 Image negatives

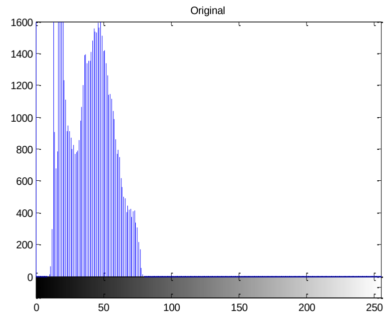
Negatives of digital images are useful in numerous applications, such as displaying medical images and photographing a screen with monochrome positive film with the idea of using the resulting negatives as normal slides. The negative of an image can be obtained by subtraction of all intensity values from the maximum intensity value.

3. Histogram modeling

In an image processing context, the histogram of an image normally refers to a histogram of the pixel intensity values. This histogram is a graph showing the number of pixels in an image at each different intensity value found in that image. For an 8-bitgrayscale image there are 256 different possible intensities, and so the histogram will graphically display 256 numbers showing the distribution of pixels amongst those grayscale values. The operation is very simple. The image is scanned in a single pass and a running count of the number of pixels found at each intensity value is kept. This is then used to construct a suitable histogram. The histogram is used and altered by many image enhancement operators. Two operators which are closely connected to the histogram are contrast stretching and histogram equalization. They are based on the assumption that an image has to use the full intensity range to display the maximum contrast. Contrast stretching takes an image in which the intensity values don't span the full intensity range and stretches its values linearly.

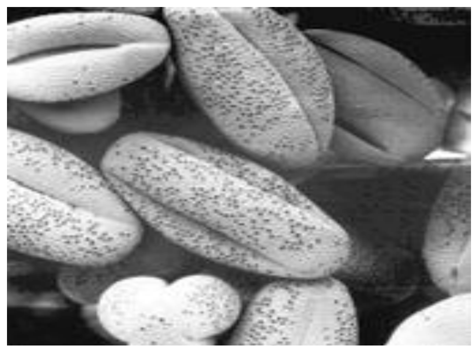
Original

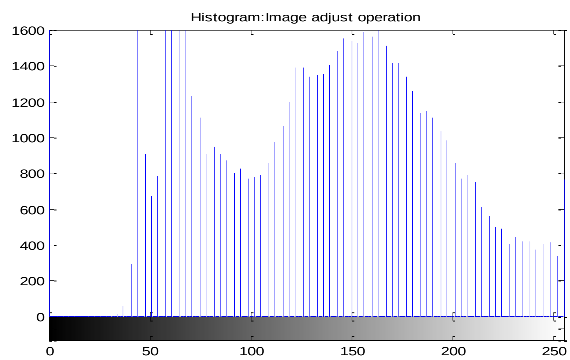




The image has low contrast. However, if we look at its histogram, it shows that most of the pixels values are clustered in a rather small area, whereas the top half of the intensity values is used by only a few pixels. The idea of histogram equalization is that the pixels should be distributed evenly over the whole intensity range, *i.e.* the aim is to transform the image so that the output image has a flat histogram.

The image results from the histogram equalization and is the corresponding histogram.





In this histogram the values are much more evenly distributed than in the original histogram and the contrast in the image was essentially increased.

4.Implementation

Consider a discrete grayscale image {*x*} and let n_i be the number of occurrences of gray level *i*. The probability of an occurrence of a pixel of level *i* in the image is

$$p_x(i) = p(x = i) = \frac{n_i}{n}, \quad 0 \le i < L$$

L being the total number of gray levels in the image, *n* being the total number of pixels in the image, and $p_x(i)$ being in fact the image's histogram for pixel value i, normalized to [0,1].

Let us also define the cumulative distribution function corresponding to p_x as

$$cdf_x(i) = \sum_{j=0}^{i} p_x(j),$$

Which is also the image accumulated normalized histogram.

We would like to create a transformation of the form *y* = *T*(*x*) to produce a new image {*y*}, such that its CDF will be linearized across the value range, i.e.

$$cdf_y(i) = iK$$

for some constant *K*. The properties of the CDF allow us to perform such a transform (see *Cumulative distribution function#Inversee*); it is defined as

$$y = T(x) = cdf_x(x)$$

Notice that the T maps the levels into the range [0,1]. In order to map the values back into their original range, the following simple transformation needs to be applied on the result:

$$y' = y \cdot (\max\{x\} - \min\{x\}) + \min\{x\}$$

Small image

The following is the same 8x8 sub image as used in JPEG. The 8-bit grayscale 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.

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:

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

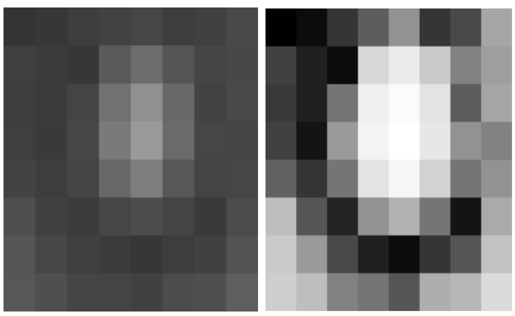
For example, the cdf of 78 is 46. (The value of 78 is used in the bottom row of the 7th column.) The normalized value becomes

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

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.



5. Generating and plotting image histogram in MATLAB.

The histogram of an digital image with L total possible intensity levels in the range[0,G] is defined as the discrete function.

$$h(r_k) = n_k$$

Where r_k is the k^{th} intensity level in the interval [0,G] and n_k is the number of pixels in the image whose intensity level is r_k. The value of G is 255. MATLAB function for dealing with image histogram is imhist, which has the following basic syntax.

h = imhist(f,b)

Where f- input image, h-is the histogram, b-is the number of bins(subdivision of intensity scale) used for forming histogram. (If b is not included the default value is 256).

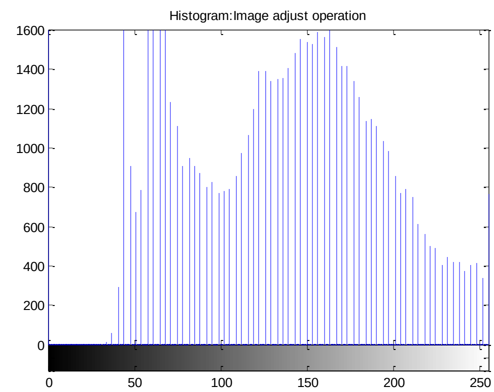
5.1 Histogram Equalization

The process of adjusting intensity values can be done automatically by the histeq function. histeq performs histogram equalization, which involves transforming the intensity values so that the histogram of the output image approximately matches a specified histogram

This example illustrates using histeq to adjust a gray-scale image. The original image has low contrast, with most values in the middle of the intensity range. histeq produces an output image having values **evenly distributed throughout the range**.

```
I = imread('file1.bmp');
J = histeq(I);
imshow(J)
```

The Image after histogram equalization



Histogram

5.2 Adaptive Histogram Equalization

As an alternative to using histeq, you can perform adaptive histogram equalization using the adapthisteq function. While histeq works on the entire image, adapthisteq operates on small regions in the image, called tiles. Each tile's contrast is enhanced, so that the histogram of the output region approximately matches a specified histogram. After performing the equalization, adapthisteq combines neighboring tiles using bilinear interpolation to eliminate artificially induced boundaries.

To avoid amplifying any noise that might be present in the image, you can use adapthisteq optional parameters to limit the contrast, especially in homogeneous areas.

To illustrate, this example uses adapthisteq to adjust the contrast in a grayscale image. The original image has low contrast, with most values in the middle of the intensity range. adapthisteq produces an output image having values evenly distributed throughout the range.

```
I = imread('file1.bmp');
J = adapthisteq(I);
imshow(I);
figure, imshow(J)
```

Adapthisteq

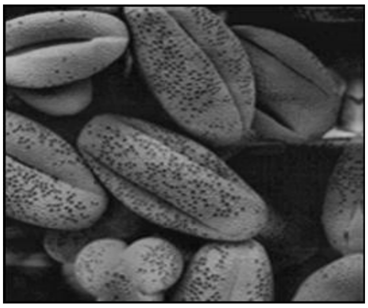


Fig: Image after Adaptive Equalization with Its Histogram

CONCLUSION

In this paper various image enhancement techniques were discussed. Image enhancement techniques are to process an image so that the result is more suitable than the original image for a specific application. The histogram equalization increases the dynamic range of gray levels and, consequently, produces an increase in image contrast.

Histogram method significantly improved the visual appearance of the image. A disadvantage of the method is that it is indiscriminate. It may increase the contrast of background noise, while decreasing the usable signal. Histogram equalization often produce unrealistic effects in photographs, however it is very useful for scientific images like thermal, satellite or x-ray images and biological neural networks. Adaptive histogram equalization method is generalization of this method uses multiple histograms to emphasize local contrast, rather than overall contrast. It is suitable for improving the local contrast of an image and bringing out more detail.

REFERENCES

[1] Rafael C.Gonzalez and Richard E. Woods, "Digital Image Processing," *PPH*, 2008

[2] Rafael C.Gonzalez and Richard E. Woods, "Digital Image Processing using MATLAB," *PPH*, 2008

[3] Jose Meseguer, "Set Theory in Computer Science - A Gentle Introduction to Mathematical Modeling I", 2011.

[4] Glynn Winskel, "Set Theory for Computer Science", 2010.

[5] Anil K. Jain, "Fundamentals of Digital Image Processing".

[6] International Journal of information & knowledge management, July-2011.

[7] International Journal of Information Technology and Knowledge Management June 2012.