

ANALYSIS AND IMPLEMENTATION OF IMAGE ENHANCEMENT TECHNIQUES USING MATLAB

Gurjeet Kaur
 GHSS, Nilokheri,
 Haryana, India

Rajiv Kumar
 Faculty of Computing
 Botho University, Gaborone, Botswana

Abstract- Digital Image Processing have a variety of diverse applications in the area of scientific, agriculture, medical, satellite, entertainment etc. The various steps of digital image processing are acquisition of image preprocessing, image enhancement, image restoration, image compression, image segmentation, object recognition. Image Enhancement used to enhance the images where the images are very dark, noisy or blurred. In this paper, we will analyze the various image enhancement techniques such as point processing, neighboring processing, histogram, filtering & edge sharpening. These various techniques are implemented in MATLAB and results are shown on various images.

Keywords- Digital Image Processing, Filtering, Histogram, Image Enhancement

I. INTRODUCTION

Digital Image processing (DIP) means processing of images which our digital in nature by a digital computer. It is concerned with acquiring and processing of an image. We can perform the number of tasks by DIP Such as improvement of pictorial information for human perception, image processing for autonomous machine application, efficient storage and transmission (Agarwal et al. (2014)). Under Human perception, the following applications of DIP are: Medical imaging, remote sensing, weather forecasting, atmospheric study and astronomy. Under machine vision, the following applications of DIP are: Automated Inspection, Boundary information and video sequencing processing. Finally, image compressions are useful for storage and transmission (Canny (1986)). There are the various steps in image processing are:

Image acquisition: An imaging sensor and the capability to digitize the signal produce by the sensor.

Preprocessing: Enhances the image quality, filtering, contrast enhancement.

Segmentation: Partition an input image into constituent parts of objects.

Description/ Feature selection: Extract description of image objects suitable for further computer processing.

Recognition and Interpretation: Assigning a label to the object based on the information provided by its descriptor. Interpretation assigns meaning to a set of labeled objects.

Knowledge base: It helps for efficient processing as well as inter-module cooperation.

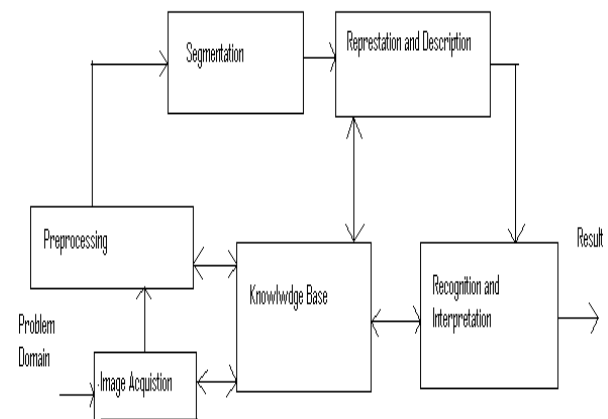


Fig. 1. Steps in Digital Image Processing

A. Image Representation

An image is a 2- D light intensity function $f(x, y)$. It can be consider as a matrix whose rows and columns indices specify a point in the image and the element value identify gray level value at that points. These elements are referred to as a pixel or pels.

$$0 < f(x, y) < \infty \quad (1)$$

$$f(x, y) = r(x, y) * I(x, y) \quad (2)$$

where $0 < r(x, y) < 1$ and $0 < I(x, y) < \infty$ (3)

The $r(x, y)$ is called *reflectance* which shows reflectivity of the surface of the corresponding image point and $i(x, y)$ is called *illuminations* which represent intensity of the incident light. These equations (1) and (2) show that an image is represent by infinite number of points. Each such image point may consist one of the infinitely many possible intensity values needing infinite number of bits (Castleman (1979)). Obviously,

such a representation is not possible by any digital computer. Therefore, there is a need to digitize an image.

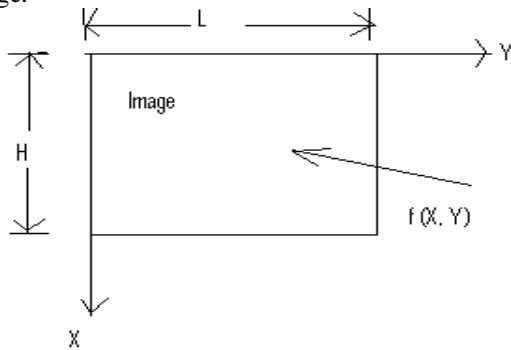


Fig. 2. Digitization of an Image

$$0 \leq X \leq H$$

$$0 \leq Y \leq L \quad (4)$$

$$I_{min} \leq f(x, y) \leq I_{max} \quad (5)$$

$$\text{Where } f(x, y) = r(x, y) * i(x, y)$$

For digitizing an image we need sampling and quantization. Sampling means representation of image by 2-D finite matrix. Quantization means that each matrix element represented by one of the finite set of elements. There are various techniques of digital image processing in which enhancement is very important technique (Kaur et al. (2017)). It is processing of an image to enhance certain features of the image. It is necessary when there is a noise in an image, image is very dark and need to highlight the edge of the object in the image. In that case, we can use image enhancement so that the result is more suitable as compare to original image (Kim (1997)).

The paper is organized as: in section I, introduction about topic has given. The section II described the point-processing techniques. Section III has insight into histogram. Section IV described the mask processing techniques. The results are analyses and implemented in MATLAB Image Toolbox and these results are shown for various images.

II. POINT PROCESSING TECHNIQUES

In case of point processing, the neighborhood size is 1×1 .

$$S = T(r) \quad (8)$$

Where r is the pixel value in the original image, S is the pixel value in the processed image and T works only on the single pixel (x, y) .

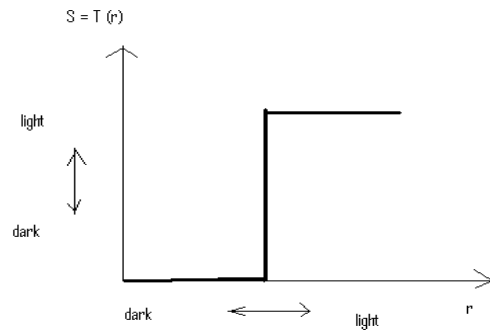


Fig. 3. Point processing in case of 1-D

From figure 3, it is clear that

$$s = 0 \text{ when } r < I \text{ and}$$

$$s = I \text{ when } r > I$$

Point-processing is of following types:

A. Image Complement

The complement of a grey scale image is its photographic negative (Li et al. (2017)). Figure 4 shows the complement function and the result of the commands.

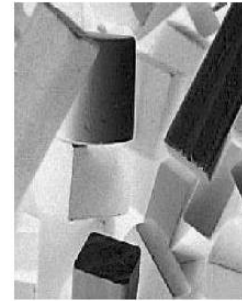
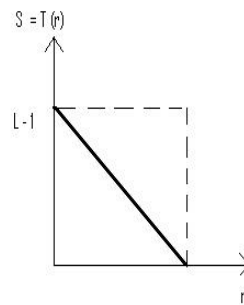


Fig. 4. Image Complement and its result in MATLAB

Here, the value of processed image S will be

$$S = T(r) = L - 1 - r \quad (9)$$

The MATLAB code is

```
>> b=imread('blocks.tif');
>> bc=imcomplement(b);
>> imshow(bc)
```

B. Image adding and subtraction

Figure 5 shows the result of adding or subtracting 128 from each pixel in the image

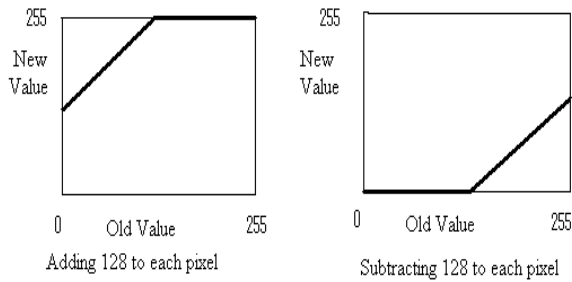


Fig. 5. Adding and subtracting a constant

When we add 128, all grey values of 127 or greater will be mapped to 255. And when we subtract 128, all grey values of 128 or less will be mapped to 0 (Mahashwari et al. (2013)). By looking at these graphs, we observe that in general adding a constant will lighten an image, and subtracting a constant will darken it.

$$y = 255 \text{ if } y > 255, \\ y = 0 \text{ if } y < 0 \quad (10)$$

We can use the Matlab function *imadd* and *imsubtract* which is designed precisely to do this:

```
>> b1=imadd(b,128);
```

```
>> b2=imsubtract(b,128);
```

And now we can view them:

```
>> imshow(b1),figure,imshow(b2)
```

and the results are seen in figure 6.

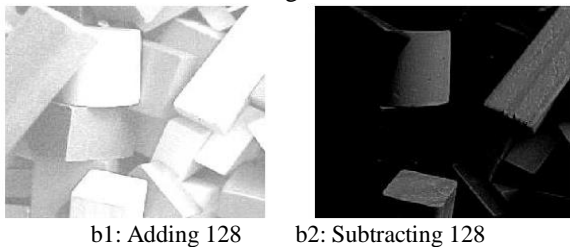


Fig. 6. Arithmetic operations on an image: adding or subtracting a constant

C. Contrast stretching

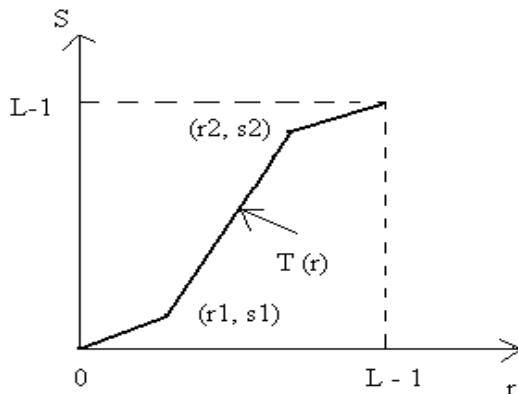


Fig. 7. Contrast stretching

If $r_1 = r_2$ and $s_1 = s_2$, then it is a straight line. For other values of r_1, r_2, s_1, s_2 ; there is variation in proposed image. If $r_1 \leq r_2$ and $s_1 \leq s_2$ then, $T(r)$ will be single valued function and it is monotonically increasing. It means that if the image is dark then the proposed image is also dark and if the image is white then the proposed image is also white. If $r_1 \geq r_2$ and $s_1 \geq s_2$; then, we get totally opposed proposed image. It means that if the image is dark then the proposed image is white and vice-versa (Murni (2000)).

All these techniques, which are described above, do not consider the overall appearance of the image. They simply take the transformation on the original image and correspondingly provide the processed image.

III. HISTOGRAM

It considers the overall appearance of an image. Given a grey scale image, its histogram consists of the histogram of its grey levels; that is, a graph indicating the number of times each grey level occurs in the image (Nnolim (2018)). We can infer a great deal about the appearance of an image from its histogram, for example: In a dark image, the grey levels (and hence the histogram) would be clustered at the lower end. In a uniformly bright image, the grey levels would be clustered at the upper end.

In a well contrasted image, the grey levels would be well spread out over much of the range.

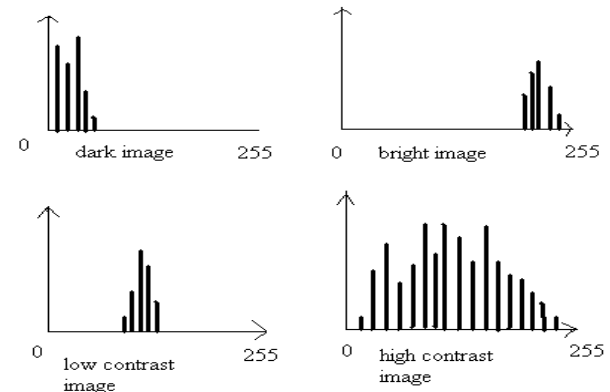


Fig. 8. Histogram

It is clear from the figure 8 that in case of high contrast image, the histogram is uniformly spread.

Histogram is given by following equation:

$$H(r_k) = n_k \quad (12)$$

Where r_k is the k^{th} level of intensity and N_k is the number of pixels in the image having intensity level $H(r_k)$

Histogram is of two types:

- ❖ Histogram equalization
- ❖ Histogram specification

A. Histogram equalization

To apply histogram equalization in Matlab, use the *histeq* function; for example:

```
>> p=imread('pout.tif');
>> ph=histeq(p);
>> imshow(ph),figure,imhist(ph),axis tight
```



Fig. 9. Before equalization and after equalization

Limitations: In case of histogram equalization, the image is fixed, there is no interactive manipulation. It generates only single processed image. The probability density function (PDF) of intensity value of equalized image is ideally uniformly distributed. Theoretically, we may get uniform distribution but practically it is not possible because in discrete image, many of the pixels value may not be present (Rana (2015)). To achieve this, we use another technique which is called *histogram specification* or *histogram matching* or *histogram modification*.

B. Histogram specification

In case of histogram specification, we enhance only certain region of the histogram which is not possible in case of histogram equalization. The image has to be processed such that histogram of the proposed image becomes same as that of target histogram. We have

$$P_r(r_k) = n_k / n \quad \text{-- from given image}$$

$$P_z(z_k) \quad \text{-- Target histogram which is specified}$$

$$s_k = T(r_k) = \sum_{i=0}^k n_i / n = \sum_{i=0}^k p_r(r_i)$$

$$z_k \Rightarrow v_k = G(z_k) = \sum_{i=0}^k p_z(z_i) = s_k$$

$$\Rightarrow Z_k = G^{-1}(s_k)$$

We can find G^{-1} by using iterative process.

Step 1: Obtain histogram of the given image.

Step 2: Precompute a mapped level s_k from each level r

$$s_k = \sum_{i=0}^k n_i / n$$

Step 3: Obtain transformation function G from given $p_z(z)$ using

$$z_k \Rightarrow v_k = G(z_k) = \sum_{i=0}^k p_z(z_i) = s_k$$

Step 4: precompute z_k for each value of s_k using iterative scheme.

Step 5: For each pixel in the original image, if the value of the pixel is r_k , map this to its corresponding level s_k ; then map s_k into the final level z_k using the precomputed values.

$$v_k = G(z_k) = s_k$$

$$G(z_k) - s_k = 0$$

We assume that $z_k \approx \hat{z}$. Therefore,

It should satisfy $G(\hat{z}) - s_k \geq 0$

So, we will increment value of \hat{z} by iteratively unless that condition met.



Fig. 10. Before specification and after specification

IV. MASK PROCESSING TECHNIQUES

In case of mask processing, the neighborhood size is 3×3 , 5×5 , 7×7 and so on. To change the grey level of a given pixel we need only know the value of the grey levels in a small neighborhood of pixels around the given pixel (Sivakumar et al. (2018)).

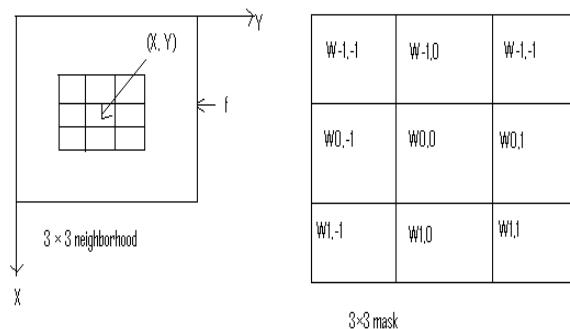


Fig. 11. Mask Processing

The processed image can be calculated as:

$$g(x, y) = \sum_{i=-1}^1 \sum_{j=-1}^1 W_{i,j} f(x+i, y+j) \quad (11)$$

In mask processing techniques, the following filters are:

- ❖ Smooth spatial filter
- ❖ Median filter
- ❖ Sharpening filter

A. Smooth spatial filter

It is also called *low-pass filter* or *image averaging* or *image smoothing*. Image smoothing is the spatial filtering operation where the value at a particular location (x, y) in the processed image is the average of all the pixel values in the neighborhood of (x, y) .

For example, suppose we apply the 3 X 3 averaging filter to an image as follows:

```
>>filter2(filter,image,shape)
>> c=imread('cameraman.tif');
>> f1=fspecial('average');
>> cf1=filter2(f1,c);
```

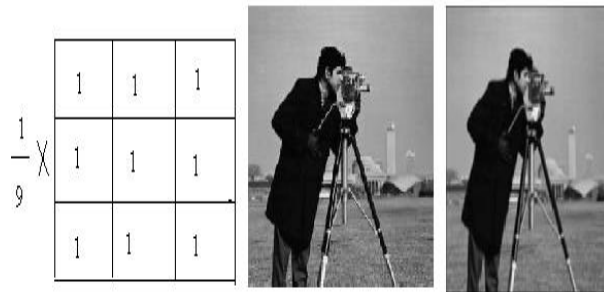


Fig. 12. Box filter, original image and averaging filter

We can get box filter by this equation

$$g(x,y) = \frac{1}{9} \times \sum_{i=-1}^1 \sum_{j=-1}^1 f(x+i, y+j) \quad (12)$$

B. Median filter (non-linear filter)

In non-linear filter, the response is based on the original of pixel values in the neighborhood of point under consideration (Wan et al. (2018)). Suppose we first take 3 X 3 neighborhood and then arrange the order of pixel and take median of these pixels.

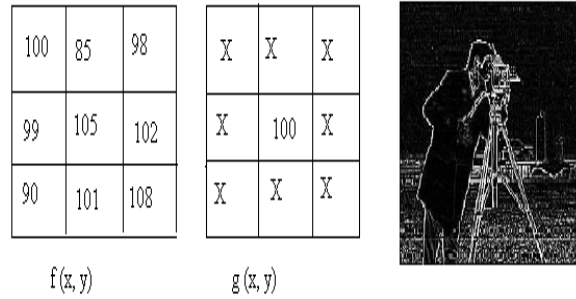


Fig. 13. Original image, processed image and its result

C. Sharpening filter

Spatial filtering can be used to make edges in an image slightly sharper and crisper, which generally results in an image more pleasing to the human eye. The operation is variously called *edge enhancement*, *edge crispening*, or *unsharp masking* (Xiao et al. (2018)). The idea of unsharp masking is to subtract a scaled unsharp version of the image from the original. In practice, we can achieve this affect by subtracting a scaled blurred image from the original.

The unsharp masking can be applied by the following sequence of commands:

```
>> f=fspecial('average');
>> xf=filter2(f,x);
>> xu=double(x)-xf/1.5
>> imshow(xu/70)
```



Fig. 14. Original image and unsharp image

V. CONCLUSION

Image enhancement is used to enhance the image. There are various techniques of image enhancement. Point processing techniques operate on the single pixel while neighboring processing on the adjacent pixels of an image. Image negative darkens those pixels in processed image which are light in original image and vice-verse. Image add and image subtract brighten and lighten the given image respectively. High contrast gives the better look of an image. These techniques do not give the overall appearance of an image. For overall



appearance of an image, we use histogram. The disadvantage of histogram is that it is not uniformly distributed. For uniform distribution of histogram, we use histogram equalization. Histogram equalization enhances the entire region of an image, but if we want to enhance only certain region of an image, then it is not possible. Histogram matching used to enhance the certain region of an image. The result is much better in case of histogram as compare to histogram equalization. All these above techniques operate on the single pixel. For masking, we use filtering operation. Low-pass filter suppress the high-frequency components and allow the low-frequency components while high-pass filter allow the high-frequency components and blocks the low-frequency components. Averaging filter takes the average of all pixels in the processed image while median filter takes the median of all pixels. Unsharp filter highlights the edges of an object in an image. All these above techniques give the better result as compare to original image. These various techniques are implemented using MATLAB and the results are shown for various images. In short, we have a better view of an image by using image enhancement which is not possible otherwise.

VI. ACKNOWLEDGMENT

The first author is gratefully thanks to GHSS for their support throughout the research work. The second author extends his gratitude to the faculty of computing for their support.

VII. REFERENCE

- [1] Agarwal T.K., Tiwari M. and Lamba S.S. (2014). Modified histogram based contrast enhancement using homomorphic filtering for medical images. In 2014 IEEE International Advance Computing Conference (IACC), (pp. 964-968). IEEE
- [2] Canny J. (1986). A computational approach to edge detection. *IEEE Trans. Patt. Anal. Mach. Intell. (PAMI)*. vol. 8, no. 6, (pp. 679-698).
- [3] Castleman K.R. (1979). *Digital Image Processing*. Prentice Hall, Englewood Cliffs, NJ.
- [4] Gonzalez R.C and Woods R.E. (1992). *Digital Image Processing*, Addison-Wesley Publishing Company, Inc, Reading, Massachusetts.
- [5] Kaur R., Chawla M., Khiva N.K., and Ansari M.D. (2017). On Contrast Enhancement Techniques for Medical Images with Edge Detection: A Comparative Analysis. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, vol. 9, (pp. 35-40).
- [6] Kim Y.T. (1997). Contrast enhancement using brightness preserving bi-histogram equalization. *IEEE transactions on Consumer Electronics*, vol. 43, no. 1, (pp. 1-8).
- [7] Li Y., You S., Brown M.S. and Tan R.T. (2017). Haze visibility enhancement: A survey and quantitative benchmarking. *Computer Vision and Image Understanding*, vol. 165, (pp. 1-16).
- [8] Mahashwari T. and Asthana A. (2013). Image enhancement using fuzzy technique. *International Journal of Research in Engineering Science and Technology*, vol. 2, no. 2, (pp. 1-4).
- [9] Murni A. (2000). *Image Processing*, class handouts, Faculty of Computer Science, University of Indonesia, Jakarta.
- [10] Nnolim U.A. (2018). An adaptive RGB colour enhancement formulation for logarithmic image processing-based algorithms. *Optik*, vol. 154, (pp. 192-215).
- [11] Rana S.B. (2015). A Review of Medical Image Enhancement Techniques for Image Processing. *International Journal of Current Engineering and Technology*, vol. 5, no. 2, (pp. 1282-1286).
- [12] Sivakumar R. and Mohan E. (2018). High Resolution Satellite Image Enhancement Using Discrete Wavelet Transform. *International Journal of Applied Engineering Research*, vol. 13, no. 11, (pp. 9811-9815).
- [13] Wan M., Gu G., Qian W., Ren, K., Chen, Q. and Maldague X. (2018). Infrared image enhancement using adaptive histogram partition and brightness correction. *Remote Sensing*, vol. 10, no. 5.
- [14] Xiao B., Tang H., Jiang Y., Li W. and Wang G. (2018). Brightness and contrast controllable image enhancement based on histogram specification. *Neurocomputing*, vol. 275, (pp. 2798-2809).

