DEMOCRATIC AND POPULAR ALGERIAN REPUBLIC

MINISTRY OF SUPERIOR EDUCATION AND SCIENTIFIC RESEARCH

Dissertation submitted to **E**l-hadj **L**akhdar **U**niversity of **B**atna
Engineering **S**ciences **F**aculty
for the degree of
Master in **E**lectronics

Entitled

Forward and Inverse Halftoning using LUT approach Applied to Grey-Level Images

By

ATAMNA Nora

Option μ-waves for **T**elecommunications

Jury members

President : BENATIA Djamel	M.C,. University of Batna
Reporter : DIBI Zohir	C.C,. University of Batna
Co-reporter : BOURIDANE Ahmed	Reader,. Queen's University Belfast, UK
Examiner : NEKHOUL Bachir	Professor,. University of Jijel
Examiner : FORTAKI Tarek	M.C, University of Batna

November 2005.

Forward and Inverse Halftoning using LUT approach Applied to Grey-Level Images

ATAMNA Nora
Electronics Department
Elhadj Lakhdar's University Batna

Abstract

Digital halftoning is the process of rendering continuous tone images into relatively small number of different output levels. For example a grey scale halftoning converts a continuous 8 tone image into a lower resolution (e.g., 1 bit per pixel) for printing or display purposes. This is achieved by generating a pattern of textures that, when perceived by the human viewer, has the appearance of a constant grey value. These patterns are then locally modulated to represent detail in the image. The process of halftoning and its inverse are usually computationally expensive and algorithms that speed up this process are often desired. It is the purpose of this research project to propose efficient algorithms in terms of speed while maintaining sufficient image quality for the inverse halftoning problem.

To achieve these goals we propose to use an algorithm based on the Look up Table principle for both forward halftoning and the inverse problems. The main feature of algorithms proposed is that no computation is required in the fetching of the output values thus making it a fast method. In this work, an investigation to accelerate the algorithms proposed in [17a] and [17b] but with an additional emphasis on showing how the LUT method for inverse halftoning has been carried out. The algorithms proposed have been compared to two existing and similar ones and the results obtained have shown that our algorithms perform favourably in terms of speed and image quality.

Acknowledgements

J'ai obtenu mon ingéniorat en 2001 et avec les circonstances de cette époque, je n'avais aucune ambition de continuer mes études; j'ai même fait la 'bêtise' d'enseigner autre chose que ma spécialité.
Puis, un jour, j'ai accompagné ma sœur à un séminaire où on abordait la sensibilisation vis à vis du sujet du cancer et l'apport de la recherche dans la lutte contre cette maladie, et depuis, j'ai décidé de m'investir dans la recherche scientifique en m'inscrivant dans la post graduation, promotion 2004.
Finissant ma première année, J'ai eu la chance de choisir un thème récent et qui n'a pas été abordé auparavant au niveau de l'université de Batna et encore moins dans les autres universités du pays.
Je ne remercierai jamais assez Dieu qui m'a permis et aidé à arriver où je suis.

D'abord, je dois présenter ma sincère gratitude à quelqu'un qui a merveilleusement su m'apprendre à « *pêcher le poisson au lieu de me le donner* » : Dr Bouridane Ahmed, de l'Université de Belfast.
J'ai puisé la motivation et la volonté d'aller toujours de l'avant dans la personne du Dr Dibi Zohir qui n'a ménagé aucun effort et qui a saisi toutes les occasions pour me conseiller.

Sincères remerciements au Dr BENATIA Djamel de l'Université de Batna, qui a accepté d'assumer la présidence du jury, au Professeur NEKHOUL Bachir de l'Université de Jijel et au Dr FORTAKI Tarek de l'Université de Batna, tout deux membres du jury, de bien vouloir accepter de juger cet humble travail.

Je tiens, en outre à exprimer tous mes remerciements à mes enseignants du département d'électronique notamment mon professeur de 'La Théorie de Signal' Mr Koulougli Abdel Malek, Dr Saïdi Lamir, le Dr Louchène Ahmed, qui a bien voulu mettre à notre disposition son laboratoire de Microprocesseur et le Professeur Abdessmed Rachid du département d'éléctro-technique pour son encouragement et sa générosité.

Je ne pourrai faire passer cette occasion sans remercier Mr Khelifi Fouad de l'Université de Belfast et Mr Umair F. Siddiqi de l'Université d'El Dhahran pour leur précieuse aide.

Merci infiniment à vous tous.
A. Nora

Contents

1st chapter Background and previous work

1.1 Introduction

Images are very useful means to convey beliefs. In our image saturated society, our cultural myths and beliefs are daily reinforced through the numerous photographs we come across – from advertisements, to the news, to family-vacation snapshots. The manipulation of images is well accepted method used as a window on the world rather than as a highly selective filter, placed there by a specific hand and mind.

Optically recorded images such as photographs and video imagery consist of minute, indeterminately arranged components. While the digital photograph looks like its conventional counterpart, when examined very closely, it reveals itself to be composed of discrete elements called pixels, which are assigned precise numerical values. Each pixel in the image has a determined Cartesian horizontal and vertical location value and a specific colour-intensity value. It is this relationship of modular units with definite values that makes it totally controllable. Thus, when we speak of the digital photographic image, we are referring to a simulated photographic representation, achieved through any combination of a mechanical lens, or a database filtered by mathematical language.

Digital halftoning is the process of rendering continuous tone images (such as printing or display) into relatively small number of different output levels. For example grey scale halftoning converts a continuous 8 tome image (say 8 bits per pixel) to a lower resolution (e.g., 1 bit per pixel) for printing or display purposes. This is achieved by generating a pattern of textures that, when perceived by the human viewer, has the appearance of a constant grey value. These patterns are then locally modulated to represent detail in the image. Therefore, the purpose of digital halftoning algorithm is to generate an image that can be printed in a stable manner with the intended printing device (laser, dot-matrix, … etc.) and that, to the viewer, also appears as similar as possible to the original continuous tone image.

1.2 Overview on displaying

Image rendering technologies can be divided into two major categories: electronic displays and printers. Electronic displays can be further sub-divided into emissive and non-emissive types. Emissive displays are those in which the image-forming element also serves as the source of light, while non-emissive displays modulate some aspect of an extrinsic illumination source.

There are currently a large number of display technologies for rendering an electronic image, but two types dominate the market: the cathode ray tube (CRT) is the dominant emissive technology while the liquid crystal display

(LCD) is the pervasive non-emissive technology. Printing is a non-emissive rendering technology.

A digital frame buffer controls most displays. The intensities emitted by the three primaries comprising each pixel are specified by three digital values, (R, G, B). The potential scope of a complete characterization is enormous. The industry standard for color applications allocates 8 bits of intensity control for each display primary and a total of $2^{8(3)}$ or approximately 16.8 million combinations. Multiplied by roughly a million pixels on the screen, and taking into account interactions between pixels, makes it impossible to perform an exhaustive characterization. Instead, characterizations are always based on simple models of the device that make powerful assumptions about the relationships between the display primaries and the spatial interactions between pixels.

The primary roles of the frame buffer are the storage, conditioning and output of the video signals that drive the display device. The industry standard for color applications allocates 8 bits of intensity control for each display primary or approximately 16.8 million discretely addressable colors. The match between the sampled values and human color sensitivity is imperfect, consequently, not all displayed colors can be discriminated from one another, and many colors that differ by a single bit in their digital representation are significantly above the threshold discriminability of the eye. This results in various types of color artifacts, such as contour artifacts on shaded graphics. High quality rendering and other demanding applications, such as psychophysical measurements, can require finer (10 or 12 bits) control over the primary intensity level.

Many of the features of the frame buffer are determined by cost considerations, and the primary costs relate to the size and speed of the frame buffer memory. Consider a display system with 1280 x 1024 addressable pixel resolution and each pixel controlled by a 24-bit value. This system requires 4 million bytes of (fast) memory to represent the frame. An economical alternative is the look-up table (LUT) architecture. In this design, the intensity levels of the primary colors are controlled by a list of entries in a look-up table. Hence, a single number represents the three voltage levels that control the primary intensities, say between 0 and 255. At display time, the system retrieves the primary values from the LUT. In this way, each pixel is represented by one 8-bit quantity. While the precision of intensity control is established by the 8- bit resolution of the digital-to-analog converters (DACs), the number of entries in the LUT limits the total number of colors available for simultaneous display.

One benefit of a LUT design is to reduce image memory. There are other benefits as well, for certain types of display conditions. For example, LUTs provide an efficient means for implementing image operations that depend only on display value, and not on display position. To alter the image luminance or contrast one can re-write the 256 LUT entries rather than the entire image buffer. In this way various image processing operations, spanning the control of drifting and flickering patterns, can be implemented by controlling only the LUT.

1.3 Digital printing

The digital information revolution has brought about profound changes in our society and our lives. The many advantages of digital information have also generated new challenges and new opportunities for innovation. A digital, numeric-based structure is by definition a statistical representation where the degree of accuracy is dependent on the amount of information that can be processed within a given space and time. The greater the memory, the richer the image's degree of resolution. When the volume of statistical data surpasses the threshold of our physiological capabilities to perceive change, the illusion of total simulation is achieved. A digital image does not represent an optical trace such as a photograph but provides a logical model of a visual experience. In other words, it describes not the phenomenon of perception but rather the physical laws that govern it, manifesting a sequence of numbers stored in computer memory. Its structure is one of language: logical procedures or algorithms through which data is orchestrated into visual form. Even though both may look the same on the surface, a digital image may be said to differ from its analog counterpart in terms of a verifiable past and a possible future.

Because of the computational load of dealing with images containing millions of pixels, digital image processing was largely of academic interest until the 1970s, when dedicated hardware became available that could process images in real time, for some dedicated problems such as television standards conversion. As general-purpose computers became faster, they started to take over the role of dedicated hardware for all but the most specialized and compute-intensive operations.

Digital image processing allows the use of much more complex algorithms for image processing, and hence can offer both more sophisticated performance at simple tasks, and the implementation of methods which would be impossible by analog means. The digital world presents a vaster field of use, compared to the analog one owing to the fact that it offers:

§ Exactness: it provides both a perfect reproduction without degradation and perfect duplication of processing result;

§ Convenient and powerful computer-aided processing: by performing rather sophisticated processing through hardware or software;

§ Easy storage and transmission: it's well known that one CD can store thousands of informations including photographs; moreover, through network, it allows paperless transmission of high quality photos within seconds .

Digital halftoning is similar to traditional halftoning in which an image is decomposed into a grid of halftone cells. Elements (or dots that halftoning uses in simulates shades of grays) of an image are simulated by filling the appropriate halftone cells. The more number of black dots in a halftone cell, the darker the cell appears. For example, a tiny dot located at the center, in figure (1.1) is simulated in digital halftoning by filling the center halftone cell; likewise, a medium size dot located at the top-left corner is simulated by filling the four cells at the top-left corner. The large dot covering most of the area in the third image is simulated by filling all halftone cells.

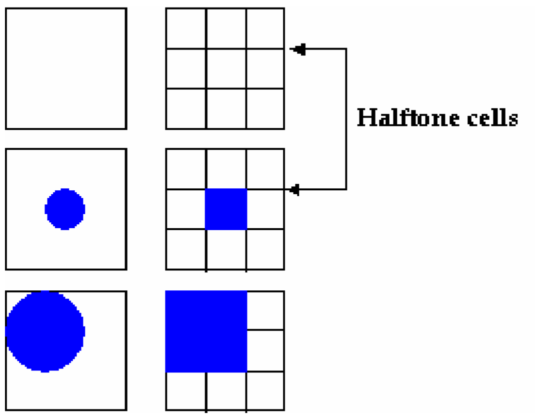


figure (1.1) Traditional (Photographic) verses digital halftonig

1.4 AM and FM halftoning methods

To build a halftoned image two main halftoning methods, AM (Amplitude Modulated, also called conventional) and FM (Frequency Modulated, sometimes referred to as stochastic) can be used. In the AM method the size of the dots is variable while their spacing is constant. The single dot within the halftone cell grows larger as the tone value becomes darker and smaller as the tone value becomes lighter. The distance from the centre of one halftone cell to the next is the same, at least in normal cases when the halftone angle is 0o or 45o. The screen ruling, also referred to as lines per inch (lpi) or frequency, controls the spacing of dot placement. The higher frequency the more continuous an image will appear. For example, halftone dots will be invisible to the naked eye at 150 lpi. Laser printers use a matrix of imaging elements to create the halftone dot. The size of the matrix is determined by dividing the device resolution, or dots per inch (dpi), of the laser printer by the intended frequency. For example, a 300-dpi printer resolution combined with a 100 lpi halftone would use a matrix of 3 by 3 image elements per halftone dot and a 1200 dpi laser printer with a 150 lpi would use a halftone cell containing 8 by 8 imaging elements. The number of image elements per inch that a device can produce is known as the device resolution. As the device resolution increases so does the quality of the halftone dot.

In summary, it can be stated that, unlike the AM method, in the FM technique the size of the dots is kept constant while their spacing varies. The number of micro dots within the halftone cell increases as the tone value becomes darker and decreases as the tone value becomes lighter.

1.5 Hybrid halftoning methods

Comparing the quality of the halftone images produced by these two previous methods, the FM methods are often considered to superior, particularly for heavily textured images. They reproduce the details of the original continuous-tone image much better than AM methods. However, in images with smoothly varying tones this superiority is not so clear. Actually, in some respects the AM method can be superior for images with slowly varying tones. FM methods also suffer much more from dot gain than AM methods do. An ordinary image often consists both of details and also some homogeneous parts. The details are reproduced better by using a FM method and the other parts of the image by using an AM method. The hybrid threshold matrix should therefore have the same characteristic as a FM threshold matrix in those areas where the original image has its details, and as an AM threshold matrix elsewhere. In order to find where the original continuoustone image has its details we high-pass filter it. The absolute value of this high-pass version will have its largest density values in high pass regions (the details of the original image). By thresholding the absolute value of this high pass version with a fixed threshold we can find a mask with the help of which we can combine the two threshold matrices for AM and FM methods.

1.6 Iterative halftoning methods

Another type of halftoning techniques, which is becoming more and more popular due to the fast increase of computer power, is iterative halftoning. Unlike point-by-point and error diffusion techniques, which operate pointwise and on a local neighborhood respectively, iterative techniques operate on the entire image. In many of these techniques one begins with a binary image and measures some error based on the difference between the binary image and the original continuous-tone image. Since the eye acts as a low-pass filter, the difference between the low-pass versions of these images is often used as the error measure. After having a suitable error measure one tries to minimize or at least decrease the error by changing the initial binary image iteratively. The process is stopped when no change in the binary image can be obtained or a given condition is fulfilled. Many different iterative halftoning techniques have been introduced in the literature and often yield to images of very high quality compared to that produced by other methods. The main shortcoming of these techniques is that they are much more time demanding than the techniques discussed previously.

1.7 Previous work in halftoning and inverse halftoning

Many efforts were carried out in order to satisfy the human dream; that is the way offered to simulate real images by analysing image information content. Popular halftoning approaches can be generally distinguished, according to their processing structure and their corresponding output image characteristics, into three categories; screening, iterative and error diffusion.

Screening whose common noun is dithering is the classical method used for image halftoning. Its low complexity makes it an attractive technique in many applications. Dithering is based on the use of screens (matrices) to binarize the multi-level image. Therefore, many screens were developed. Bayer’s dispersed dot screen [3] is the most popular one. Various approaches were recently applied in order to generate screens for dithering. According to whether they are generated by directly looking for the best spatial provision of dither matrix thresholds or by level-by-level design where for every grey level, the dot profile function is designed. Allebach and Stradling in [12], looks for the suitable order of the thresholds in the dither screen in such way to minimize the cost function. With Lin he had also presented a screen design using the direct binary search (DBS) in [13]. In the same context, Rolleston and Cohen had introduced an algorithm that attempts to satisfy both constraints on the dither matrix in space domain and the spectrum of the dither matrix in the frequency domain [14].

In the second class, methods [9], [10] exploit the Human Visual System (HVS) model to minimize the residual error between the original and the perceived halftone image. Even though their requirements in terms of computation, these methods such as DBS are able to produce halftones of high quality [11].

Error diffusion algorithms, first introduced by Floyd and Steinberg [22], are more computationally intensive methods. They require diffusing the error to neighbours defined with the filter. Obviously, this method used to be serial until Knuth has proposed his algorithm and removes these limitations [7], this algorithm has further been improved by Zhang and Webber [23]. Another algorithm discussing the parallelism of error diffusion technique appears in [24]. Diffusing the error with neural networks can be found in [25].

It can happen that we need to process the continuous image rather than the bilevel image. Therefore, we have to recover it from the halftone version. This is what the inverse halftoning principle aims to.

Inverse halftoning has also found applications in many situations that involve processing halftone images. Image processing operations cannot be directly applied to halftone images because of their binary nature. Therefore, the inverse halftoning procedure is one way to solve this problem by transforming the binary image into a continuous-tone , which once processed , it will be rehalftoned.

A variety of methods are used to achieve this goal. Generally, they are based on a simple low_filtering; however, this could cause a loss of certain useful information.

In fact, each approach is typically designed to work for a specific type of halftone images. For error diffused halftones, Wong [29] had used an iterative filtering method while Hein and Zahkor [27] had applied the projection onto convex sets method. This method was also used by Analoui and Allebach [26] but for ordered dither images. Fan [28] attempts to inverse halftone dithered images with a method known as logical filtering. Wavelet was also used for error diffused images in [31]. Another interesting technique is the fast method which leads to images of very good quality which was introduced by Kite *et al.* [32].

After being introduced for the first time by Netravali *et al.* in order to display ordered dithered images, the look-up-table (LUT) approach was used by Ting and Riskin in halftone compression. Mese and Vaidyanathan have proposed their approach for the inverse halftoning problem. They had also presented the template selection method for inverse halftoning. Although both methods follow the same goal, this method in the inverse problem is based on the correlation between halftone value and contone value rather than minimizing the mean square error between the inverse halftoned and corresponding contone images in the halftoning process. The pixels used in the prediction are added in an adaptive way to the template in order to reach the greatest possible quality.

1.8 Thesis organisation

Our work presented in this thesis is divided into five parts. In chapter2, we will first introduce the most popular concepts used by the printing community; giving briefly their definitions, their main drawbacks and relation. This will be demonstrated by implementing these methods using *MATLAB* to illustrate their features. Then, the proposed algorithm for forward image halftoning based on LUT principle is detailed in chapter 3. The 4th chapter deals with the inverse halftoning proccess in order to recover the contone image using also the LUT approch. Conclusion summarising our main findings including remarks are finally given in the last chapter. We will crown our thesis with future works that can be an extension of this modest work.

We took as a starting point the theory of Murat *et al* to develop our algorithm as well for the halftoning as for the inverse halftoning process. It is the purpose of this work to propose efficient algorithms in terms of speed and image quality for the inverse halftoning problem.

2nd chapter Existing Methods for Digital Image Halftoning

2.1 Introduction

How to represent a particular colour with a peripheral which cannot visualize it? This was an interesting question which led to splendid innovations. This can be quite simply carried out by juxtaposing several different colours to obtain an average colour which is nearest to the wished colour. It can be observed that to obtain the average, one may need to access and manipulate many pixels; this usually tends to mix the contributions of those pixels, which in fact leads us to the halftoning subject.

Halftoning is one of the oldest applications of image processing, since it is essential requirement for the printing process. With the evolution of computers and their gradual introduction to typesetting, printing, and publishing, the field of halftoning that was previously limited to the so-called halftoning screen evolved into its successor—digital halftoning. Today, digital halftoning plays a key role in almost every discipline that involves printing and displaying. All newspapers, magazines, and books are printed with digital halftoning. It is used in image display devices capable of reproducing two-level outputs such as scientific workstations, laser printers, and digital typesetters. It is also important for facsimile transmission and compression.

Typically, halftoning process is used in both printing and displaying colour images, however, our work is rather focused on printing domain.

The majority of printing devices are able to print only in mode all or anything. Either a point of the output device printed or it is not. Although this can be seen as limitation, these devices are able to reproduce images comprising of the nuances of grey. The method used consists in reproducing the illusion of grey levels; this is what is called halftoning. However, this is simply produced by juxtaposing black and white dots. The percentage of white printable points in a pattern gives its average intensity level. Thus, when a pattern is made only of white points, a maximum of luminous intensity is reflected by the paper. On the other hand, if it comprises only black dots, the incidental light intensity is entirely absorbed. All of the intermediate values of intensity are obtained by an appropriate choice of the number of black (or white) points in the corresponding patterns. If the individual points of the pattern are very small, they cannot be perceived by the human eye from normal reading distance, moreover, only the average intensity generated by the combination of individual points black and white is perceived. Starting from a certain value, the individual points start to be perceived, while contributing to form with their neighbors the desired average intensity. Pattern generation algorithms aim at obtaining the desired intensity while minimizing the harmful effects generated by combinations of individual points.

In continuous tone problems, printing the printed page is covered with a very fine array of ink drops. The droplet density, but not the position, is controlled by the printing method. Hence, the control of continuous tone printing is conceptually similar to controlling the appearance of overlaid sheets of coloured transparencies, a method called subtractive reproduction. In contrast, during halftone printing the ink drops are larger and the printing process controls their position and size. The colour appearance of the print is controlled by the position and size of the dots in the array and these dots do not often overlap. Thus, the dots from different inks form several spatial mosaics, and colour reproduction is more akin to an additive process: the reflected light is the sum of light scattered from the several mosaics.

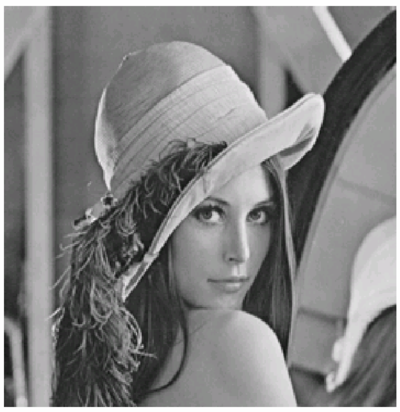
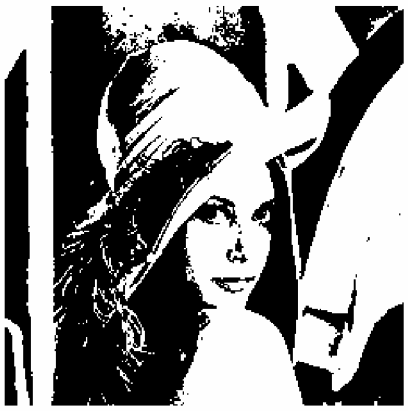
2.2 The classical methods used for digital halftoning

There are a number of different methods by which the digital image halftoning can be accomplished. In this section, and in order to familiarize the reader with the principle of halftoning, we had implemented some traditional methods. A brief discussion of the results obtained is then given to demonstrate the techniques. Generally, we can analyse the performances of the various halftoning algorithms according to different parameters: colour, shape, simplicity of the implementation and speed.

In the following a number of methods are discussed.

2.2.1 Simple thresholding

It is the generic term of any attempt to binarise a given image. This basic technique is fast, but leads to poor halftones caused primarily by a loss of fine details as indicated by figure (2.1). It operates by keeping only the pixels whose intensity is higher (or lower) than a value of reference; the threshold. It can be slightly modified by the addition of a second threshold, often termed as multi-thresholding. One can also threshold according to a pattern, which amounts to using different thresholds on the image. Finally, thresholding with respect to image of noise through the concept of random thresholding.

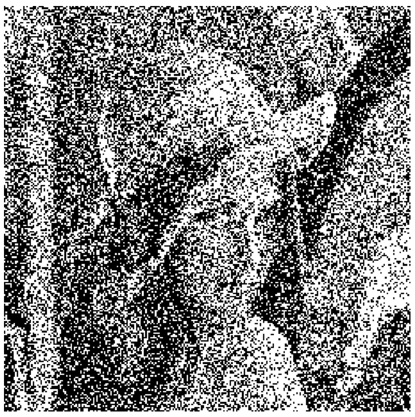
	
The original continuous-tone image	A halftone version

becomes after simple thresholding (thresh=127)

figure (2.1) Demonstration of the thresholding process

2.2.2 Random halftoning

Since grey level images have 8 bit depth (i.e., 256 values), the process of halftoning in this case can be seen as an approximation process so that the rsulting pixel values are either 255 (while) or zero (black). Random halftoning consists on taking a value between 0 and 255 and compares it with the intensity of the at hand so as to set it as either white (if it is higher) or black otherwise.

	
The original continuous-tone image	A halftone version

becomes after random halftoning

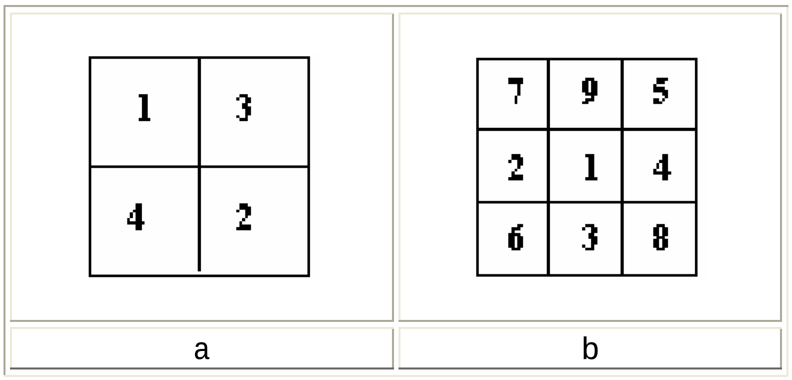
figure (2.2) Demonstration of the random halftoning process

2.2.3 Dithering with matrices

Dithering means the addition of some kind of noise prior to the quantization of a signal which, in our case, is an image. The amount of noise to be added is simply determined by the order of the pixel, i.e., its spatial coordinates. The ordered dithering techniques are attractive in the sense that they are very simple to implement, especially in parallel architectures, and that they are computationally inexpensive. This is because they involve a two-stage process that can be performed independently for every pixel. However, their performance is poor when compared to the error diffusion technique as will be discussed later.

This method works on the principle that the input continuous tone image is divided into small squares (or matrices) of preset size. Then, inside each square, we apply different threshold value preset for each box from the square. For squares of size 2x2 and 3x3, preset values are shown on figure (2.3.a;b).

It is necessary to adapt the coefficients to the problem at hand. Notice that these matrices are introduced by Bayer [3] who gave a formula governing the coefficients of a matrix from those of a matrix twice smaller, which makes it possible to reach rather significant matrices.



1	3
4	2

7	9	5
2	1	4
6	3	8

figure (2.3.a) The 2x2 dither matrix   figure (2.3.b) The 3x3 dither matrix

In the following, we will show the results of dithering with the two different matrices shown above when applied on Lena image.

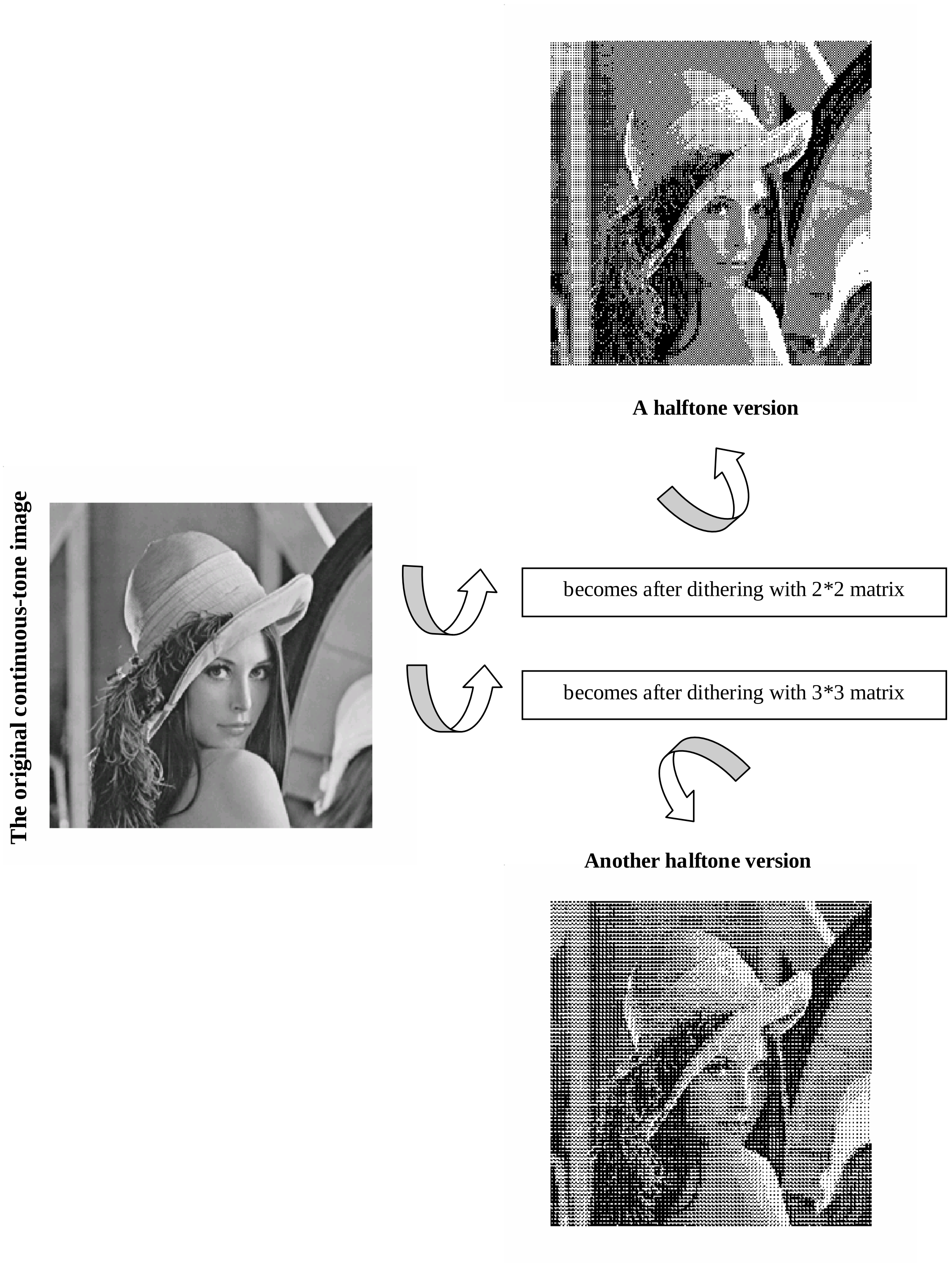

A halftone version
The original continuous-tone image
becomes after dithering with 2*2 matrix
becomes after dithering with 3*3 matrix
Another halftone version

figure (2.4) Demonstration of dithering process

2.2.4 Error diffusion

At low print resolutions, the best halftoning results are obtained using an adaptive algorithm in which the halftoning depends upon the data in the image itself. Floyd and Steinberg (1976) introduced the basic principals of adaptive halftoning methods in a brief and fundamental paper. Their algorithm is called *error diffusion*.

Unlike dithering, error diffusion is a neighbourhood operation since it operates not only on the input pixel, but also its neighbours. The idea is to initiate the halftoning process by selecting a binary output level closest to the original intensity. This binary level will differ substantially from the original one. The output pixel value of this first location will be 255 if the pixel value is higher than the threshold; otherwise, a 0 will be set. Then, the difference between the input pixel value and the outputted value is spread among neighbouring pixels according to the matrix shown graphically in figure(2.5), proposed by Floyd and Steinberg.

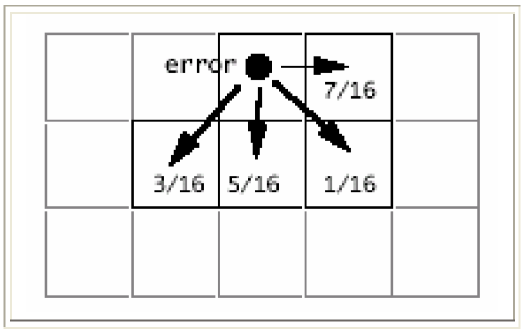


figure (2.5) Floyd & Steinberg error diffusion

Further, and in order to preserve better visual quality of halftoned images, other matrices for error diffusion were developed which allow diffusing the error;

say the difference between the pixel's original greyscale value and the threshold, onto a larger area making its effect negligible.

For example, the matrices proposed by Jarvis, Judice and Ninke and Stucki diffuse the error over the 12 neighbouring cells while Floyd and Steinberg matrix diffuses the error over only 4 neighbors.

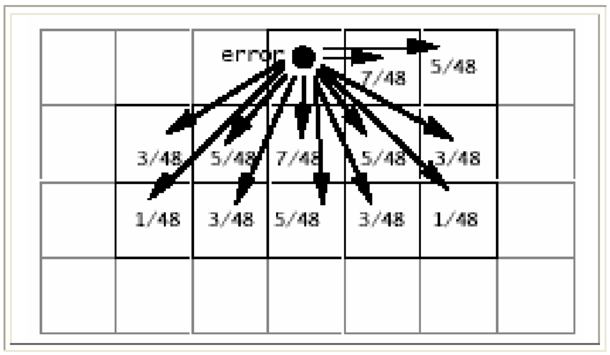


figure (2.6,a) Jarvis, Judice & Ninke error diffusion filter

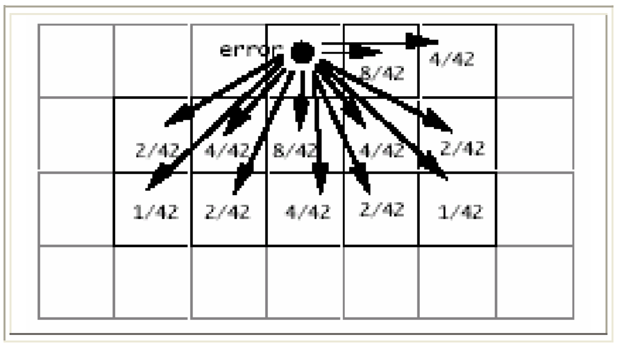


figure (2.6,b) Stucki error diffusion filter

In fact, all these proposed matrices differ only in the fractions of the diffused error as we can see it on figure (2.6.a) and (2.6.b).

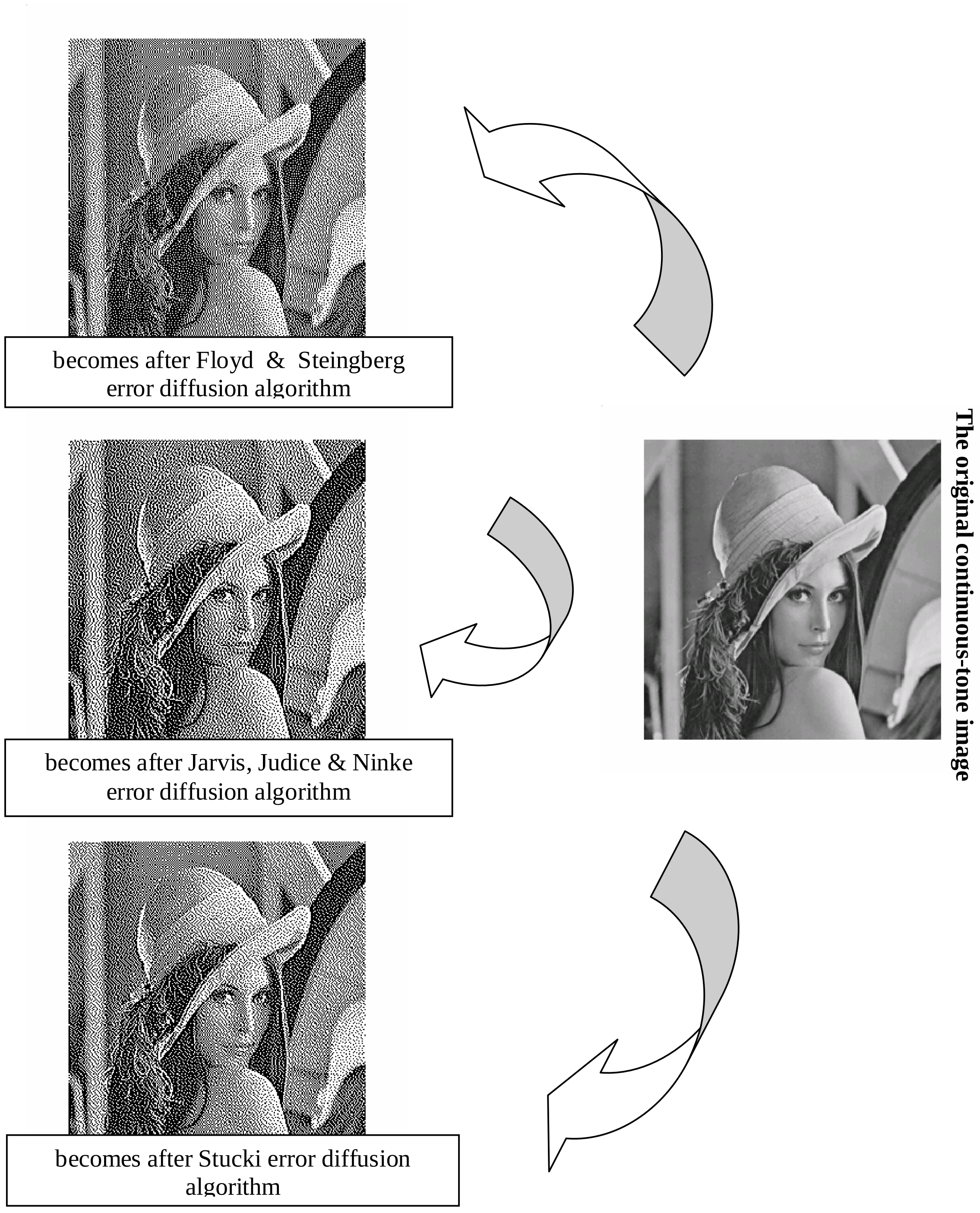

becomes after Floyd & Steingberg error diffusion algorithm
The original continuous-tone image
becomes after Jarvis, Judice & Ninke error diffusion algorithm
becomes after Stucki error diffusion algorithm

figure (2.7) Demonstrations of error diffusion process using different algorithms

2.3 Discussion of the implementation results

Theoretically, the more the dither matrix is large, the better are the returned shades, but the greater is the blur. Effectively, when compared to a 2x2 dither matrix, a 3x3 dither matrix has a better result with respect of shades; on the other hand, it introduces a blurring on the edge of the shapes. This can be seen with the appearance of a squaring which makes the halftone less aesthetic.

Visually, it can be clearly observed that an error diffusion algorithm with all of the used filters yields the best results; shades are strictly respected without deteriorating the shapes too much. In fact, the main factor affecting the halftone's quality in this case is the threshold value.

2.4 Conclusion

Converting greyscale images to halftoned black and white images remains an important issue since the advent and the proliferation of low-cost bi-level printers. Halftoning techniques using precomputed threshold matrices have an advantage in their speed of operation while neighbouring techniques show more visually pleasing results but have very slow operation.

Significant progress in terms of quality seems to be brought by the error diffusion algorithms. Moreover, efforts are centred on the development of this method although there are several different methods address this task. In the next Chapter, the Look up Table technique is exploited for halftoning problems and, as we will show, the implementations lead to excellent results when compared with diverse error diffusion algorithms known as the finest to give the best halftone quality.

3rd chapter ____________________ Forward Halftoning with Look up Table

3.1 Introduction

Halftoning is an encoding method used to reduce the number of quantization levels per pixel in a digital image while maintaining the grey appearance of the image at normal viewing distance. Halftoning is widely employed in the printing and display of digital images. The need for halftoning encoding arises either because the physical processes involved are binary in nature or the processes have been restricted to binary operation for reasons of cost, speed, memory or stability in the presence of process fluctuations.

After giving a short background on printing and halftoning by describing a few of the existing and known halftoning algorithms, we will in this section present the proposed algorithm for halftoning grey scale images.

An image is represented by a two-dimensional array where each element of the array represents a pixel (the basic picture element) value. The value stored in each array element is encoded by using a certain number of bits determining the grey-level or the intensity of the pixel.

Inside the computer, a gray-scale image is represented by assigning, to every pixel p(x,y), the intensity I(x,y) of the grey-level at this pixel. Usually in the case of grey level images, 8 bits are used to store the intensity, so we have $2^8 = 256$ possible intensity levels for each pixel.

Almost any printer cannot print the points of different intensity; at any pixel, it either prints a black (or a colored) dot, or it does not print anything at all. Therefore, when printing an image, it must first be transformed into such a form where at every pixel p(x,y), only 0 (black) or 1 (white) are needed. This transformation from the original continuous image to the two-level image is called halftoning and is the purpose of this work. Thus, the level of intensity at a pixel is represented by the relative frequency of black spots around it: any level between absolute black and absolute white means that some pixels in the neighborhood are black and some are white; the larger the intensity, the more black pixels there is.

Different algorithms, provided in the literature may perform the same task with a different set of instructions. However, they are carried out in more or less time, space, or quality and even efforts than others.

We have investigated the use of LUT method to estimate the halftone grey-scale image given a continuous one. The main advantage of using a LUT to halftone any image is that the process does not involve any computation at all. The key point here is that a halftone value at any pixel in the input image is simply obtained from a set of precomputed values in LUT. Another advantage of this method is that an LUT based approach is very fast since the reconstruction of each pixel involves only the fetching task from the table look up.

To design the Look Up Table, one typically uses a set of n training images and generates the corresponding halftones using any halftoning algorithm. After that, one collects the statistics of the mapping between the values of template contents and the corresponding grey-level values from the grey-scale images.

3.2 halftoning with LUT

The concept of the LUT requires, initially, the choice of a collection of pixels which is to as the 'template' in literature referred. These pixels are used to estimate the halftone value: *the prediction.*

The selection of this neighborhood is capital owing to the fact that it affects both the quality as well as the speed of the process. In addition, it is the basic idea exploited further in our proposed algorithm.

In their algorithm [17b], the authors had shown first, how to design the LUT for a given template; the halftone value should be estimated from the causal neighborhood as closely as possible. Because of the requirements in terms of memory constraints, then they had proposed a method to select the best templates in order to increase the quality of the halftones. This technique has been used in [30] where the suitable template selection was discussed in the compression context.

Before presenting our purpose, let's first give an overview of the algorithm introduced in [17b] for forward halftoning which provides a basis for our work through this section.

3.2.1 Basic principles used for halftoning

In the LUT method, the estimated value is a non-linear combination of the surrounding halftone pixels (neighbors) indicated by the template presented on figure(3.1).

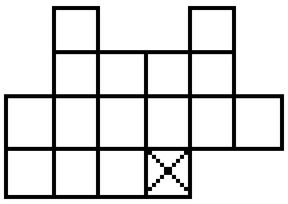


figure (3.1) The 16pls template used in [17b] for halftoning

The table that we hope to design will store all of the predicted halftone values of the different possible patterns. On the other hand, the output image is bi-level, the LUT to conceive should thus return a value $\mathrm{LUT}(p_0, p_1, \ldots, p_{N-1}, ci)$ for each pattern $(p_0, p_1, \ldots, p_{N-1,ci})$ at any input pixel $\in \{0.1\}$, where N is the number of pixels in the neighborhood.

At this stage, both contone and halftone images are required; therefore , one selects a set of images and halftone them with any halftoning algorithm (we had used Floyd and Steighnberg error diffusion in this analysis). We can summarize the design phase in the following:

Let us count the number of occurrences of the pattern $(p_0,p_1,\ldots,p_{N-1},c)$, notice that it can occur any number of times in any image of the sample set.

Two cases can take place:

1. $\underline{k(p_0,p_1,\ldots,p_{N-1}) \neq 0}$ that means that the pattern exist.

The halftone values of the pattern $(p_0,p_1,\ldots,p_{N-1},c)$ at any pixel are denoted by

$$h(p_0,p_1,\ldots,p_{N-1},c,i)\text{ , for } i = 0,1,\ldots,\ k_{p_0,\ldots,p_{N-1},c}-1$$

where $k(p0,p1,\ldots,pN-1,c)$ are the occurrences of the pattern $(p_0,p_1,\ldots,p_{N-1},c)$

According to whether the average value $m(p_0,p_1,\ldots,p_{N-1},c)$ of these values is higher or lower than a certain threshold value, the halftone value of the pattern will be decided and then assigned to its corresponding position in the table, i.e.,

$$LUT(p_0,p_1,\ldots,p_{N-1},c) = \begin{cases} 1 & \text{if } m \geq 0.5 \\ 0 & \text{if } m < 0.5 \end{cases}$$

where

$$m\,(p_0,p_1,\ldots,p_{N-1},c) = \frac{\sum\limits_{i=0}^{k(p_0,p_1,\ldots,p_{N-1})-1} h(p_0,p_1,\ldots,p_{N-1},i)}{k(p_0,p_1,\ldots,p_{N-1},c)}$$

2. $\underline{k(p_0,p_1,\ldots,p_{N-1}) = 0}$ it is the case of the nonexistent pattern.

The estimation of this pattern (its halftone value) should be performed in different ways. One consists in the following:

- $\left(p_{i,0}p_{i,1},\ldots,p_{i,N-1},c_i\right)$, for $i=0,1,\ldots,M\text{-}1$ represent all of the existing patterns in the sample halftone images (for which K≠ 0).

- The best linear estimator will be the least squares solution to an over-determined system A x = B, which is

$$x=(A^T A)^{-1} A^T B$$

where

- A : the pattern matrix
  - $A(i,j)=P_{i,j}$ for $\begin{matrix} i=0,1,\ldots,M\text{-}1 \\ j=0,1,\ldots,N\text{-}1 \end{matrix}$;
  - $A(i,N)=c_i$ for $i=0,1,\ldots,M\text{-}1$;
- $B(i)=\mathrm{LUT}\left(p_{i,0}p_{i,1},\ldots,p_{i,N-1},c_i\right)$ for $i=0,1,\ldots,M\text{-}1$: vector of the halftone values of the existing patterns (obtained in the first case).

- Define $y=[\,p_0,p_1,\ldots,p_{N-1},c\,]\ x$, where $\left(p_0,p_1,\ldots,p_{N-1},c\right)$ is the non-existent pattern.

- Finally and for each pattern non-existent, we obtain the halftone value as follows

$$\mathrm{LUT}\left(p_0,p_1,\ldots,p_{N-1},c\right)=\begin{cases}1 & \text{if } y\geq 0.5\\ 0 & \text{if } y<0.5\end{cases}$$

3.2.2 The proposed algorithm principle

The halftoning process using the LUT concept necessitates firstly a training phase where sample images are used to construct the data structure. This phase is extremely important to achieve the desired goals.

Let τ denotes the template for a halftone image $imh_{x,y}$, N will be the number of pixels in this template. For binary-valued halftones a template of halftone pixels corresponds to 2^N possible patterns. Hence if we have an LUT that contains all possible patterns and each pattern is mapped to either 1 or 0, then the halftoning procedure is simply $imh_{x,y} = LUT(im_{x,y})$, where imh_{xy} is the bi-level image.

We had chosen a 3×3 template which is denoted as 9 pixels (9pls) template shown on figure (3.3); it is a rectangular symmetric template. Any pixel in the template will be used in the prediction of the estimated pixel denoted by "×". Designing the LUT with 3×3 template gives the advantage to produce 2^9 different shades (grey levels) simulated by 2^9 patterns. In our work, we have not used the whole possibilities but only the 10 patterns shown in figure (3.2).

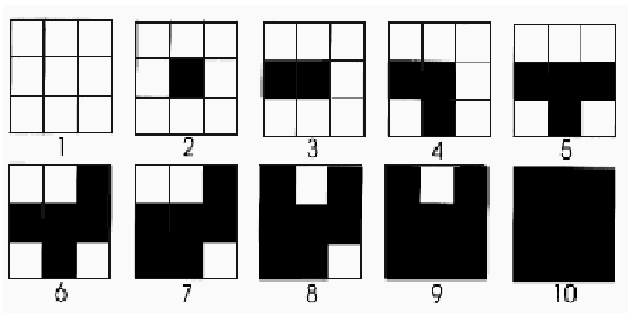


figure (3.2) The ten patterns obtained with 9pls template

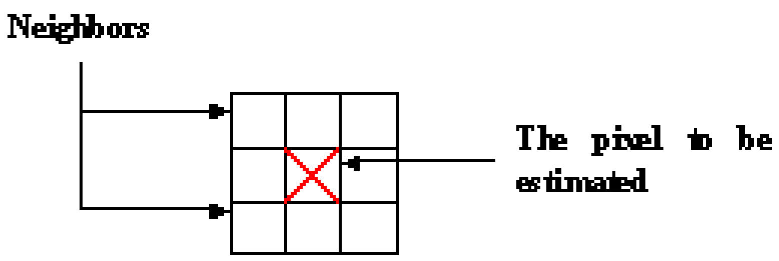


figure (3.3) The 9pls template used in our algorithm for both forward and inverse halftoning

It is worth mentioning here that our training set used in the construction of the tables is composed of 40 images of size 50×50 pixels rather than 256×256 pixels as used by Murat and Vaidyanathan. This is due to our system resources constraints. However, this is the first main factor affecting the quality.

3.2.2.1 The LUT design

For each image from the training set do:

1st step: Read the input image;

2nd step: Halftone it with Floyd & Steignberg error diffusion algorithm as follows:

- *binarize the current pixel;*
- *compute the quantization error;*
- *diffuse the error on neighboring pixels using weights as in* figure (2.5);
- *move to the next pixel (in a raster-scan order);*

3rd step: In a raster scan order, for each pixel in the halftone image resulted in the 2nd step, define the neighbors according to our 9pls template;

4th step: Compute the vector K denoting the occurrences of all the patterns;

5th step: Calculate the halftone value $\mathrm{LUT}(p_0,p_1,\ldots,p8,c)$ of any pattern $(p_0,p_1,\ldots,p8,c)$ according to whether it exists or not.

3.2.2.2 The halftoning phase

The input is a 256×256 continuous image and the output will be therefore bi-level.

1st step: Read the halftone result of the input image obtained in step 2 of (3.2.2.1);

2nd step: In a raster scan order, define for each pixel the neighbours according to the template used.

3rd step: The output pixel will be assigned the halftone value assigned to the binary pattern at this location (step 2).

3.3 Experiment results

We had used various images for our experiments and for visual comparison images in figure (3.6) are given. The corresponding halftone images shown in figure (3.5) are obtained with the Floyd and Steigenberg error diffusion algorithm evoked in the second chapter.

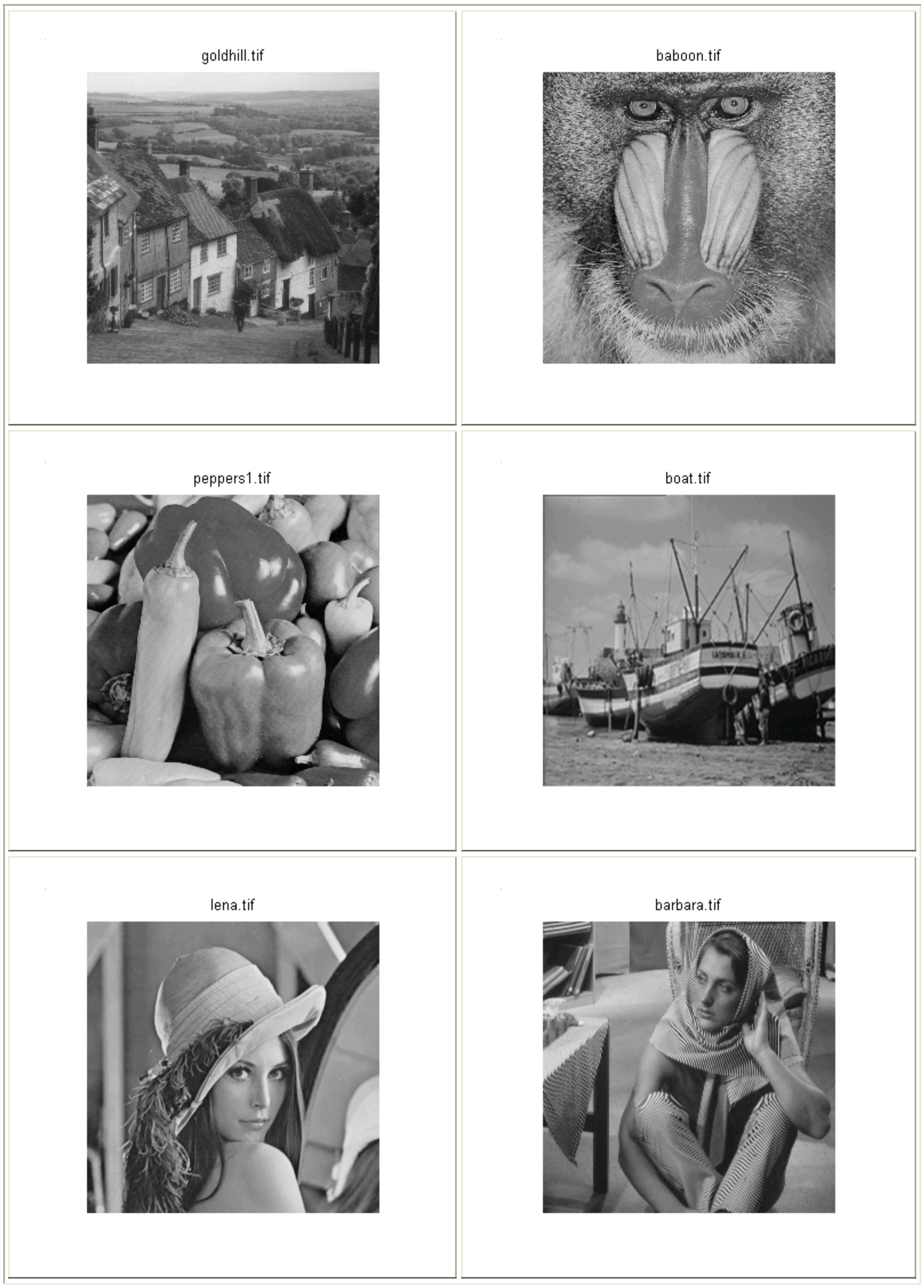


figure (3.4) The original images

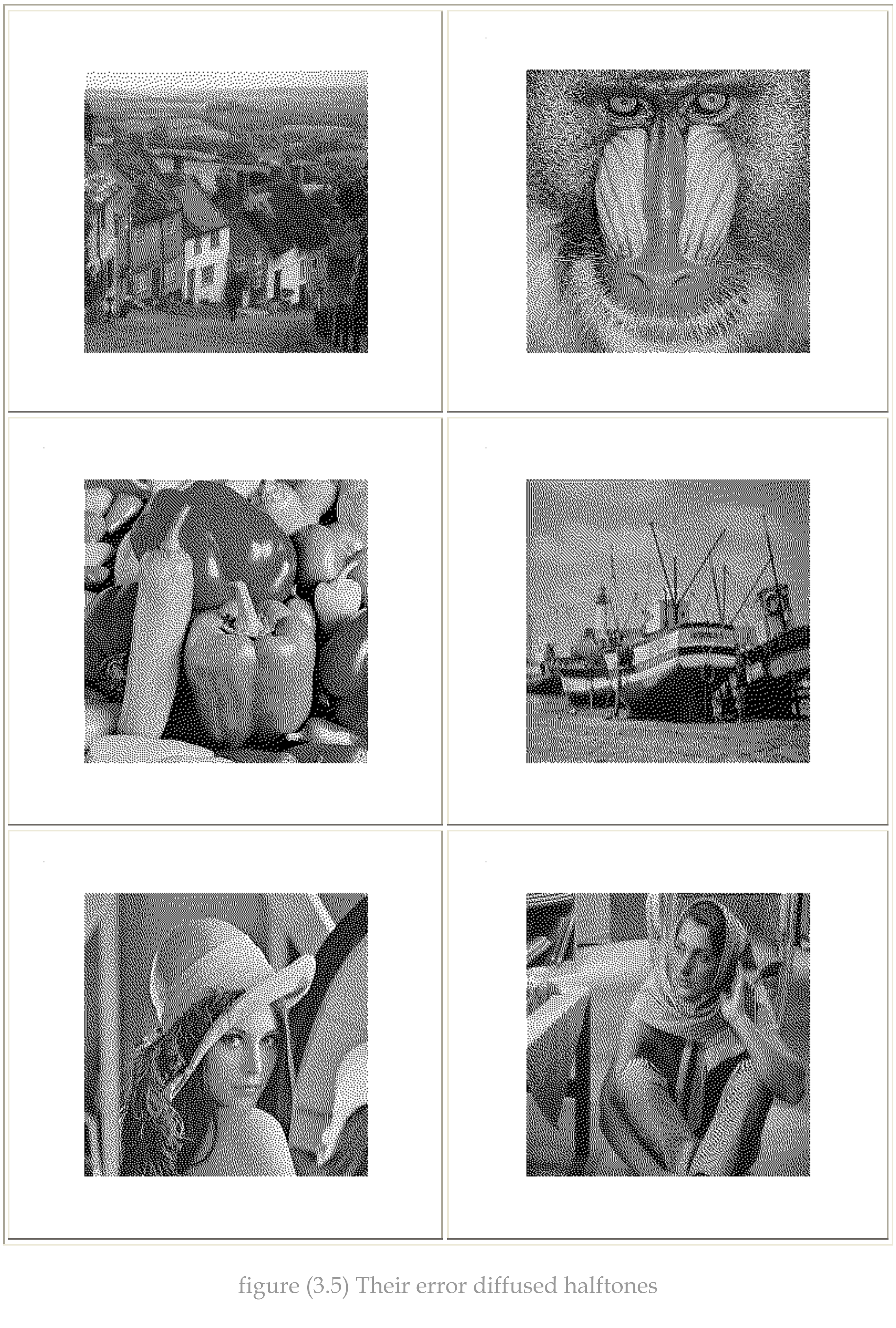


figure (3.5) Their error diffused halftones

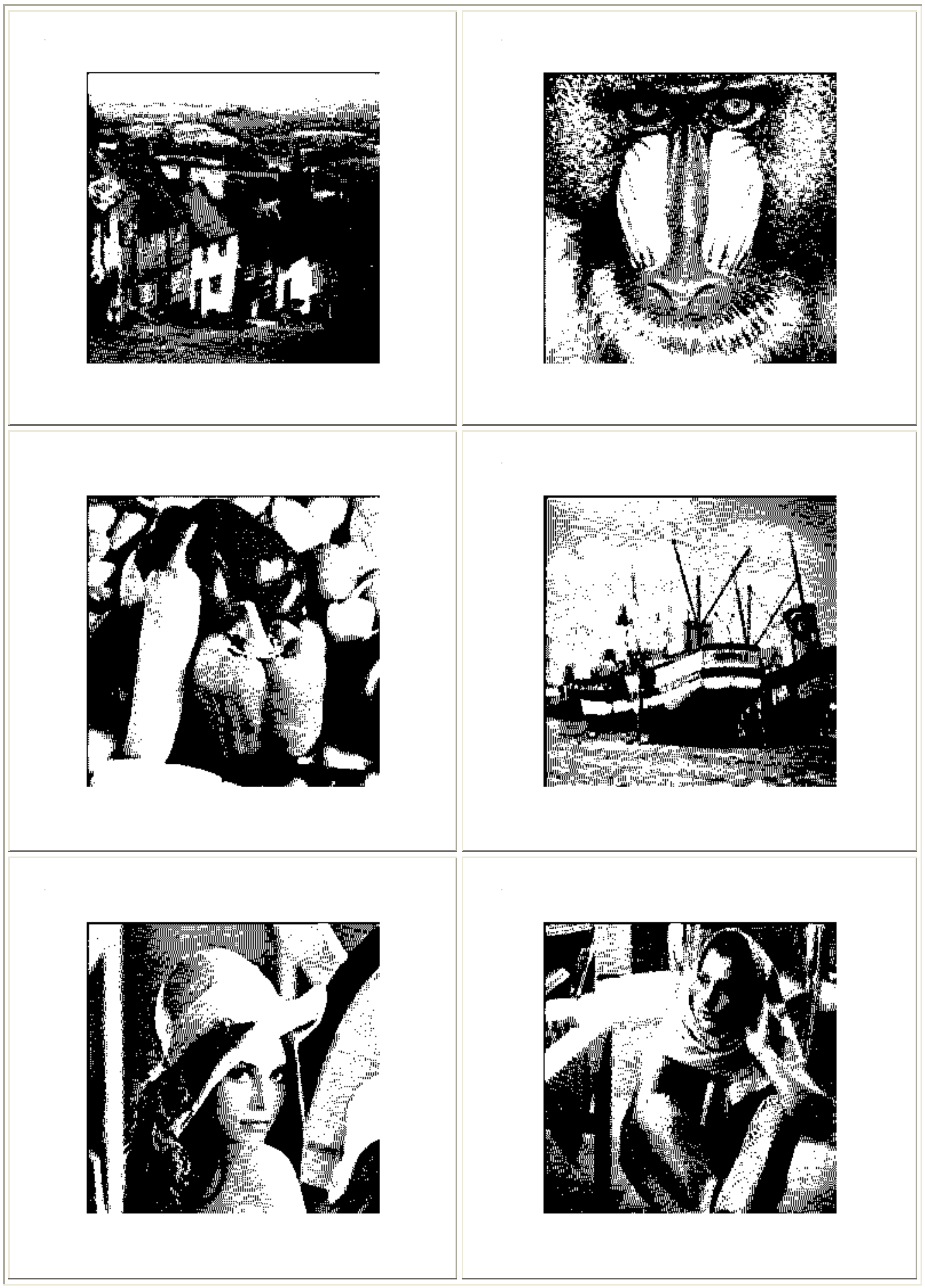


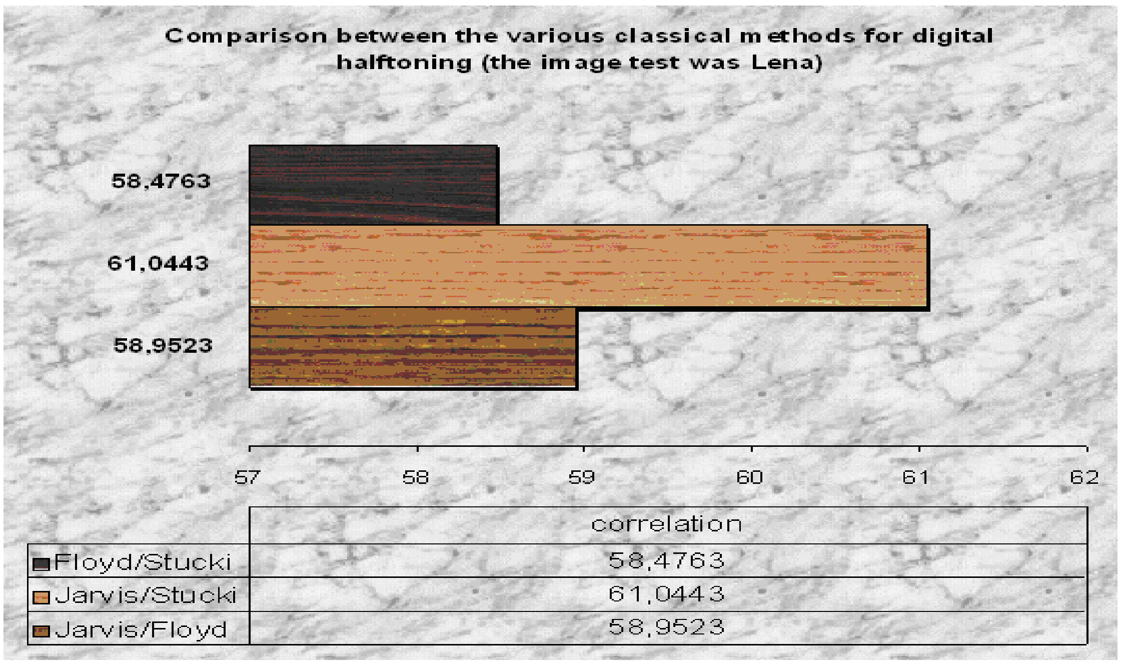
figure (3.6) Their corresponding halftones with the proposed algorithm

3.4 Discussion of our algorithm's achieved features

3.4.1 Evaluation of the performance

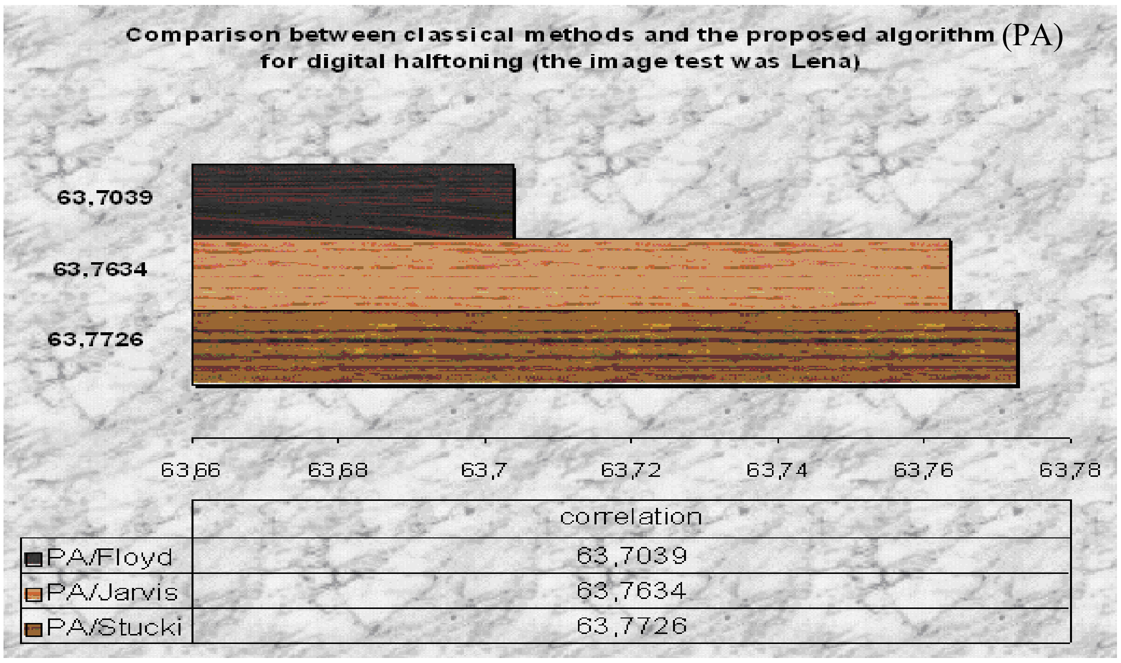
It was mentioned in the last chapter that error diffusion algorithm with all of the considered filters gives the best halftones compared to the other techniques with a reasonable time. Therefore, we thought it is worth making a comparative study to demonstrate the efficiency of the proposed assumption when compared to this algorithm known as the most opted technique for image halftoning [6], [7].

The image test was first trained by different error diffusion filters and by our algorithm. The difference is evaluated by calculating correlations between the halftones obtained. The famous Lena and Peppers images are used in our experiments and results are reported in the following diagrams:



	correlation
■ Floyd/Stucki	58,4763
■ Jarvis/Stucki	61,0443
■ Jarvis/Floyd	58,9523

figure (3.7)



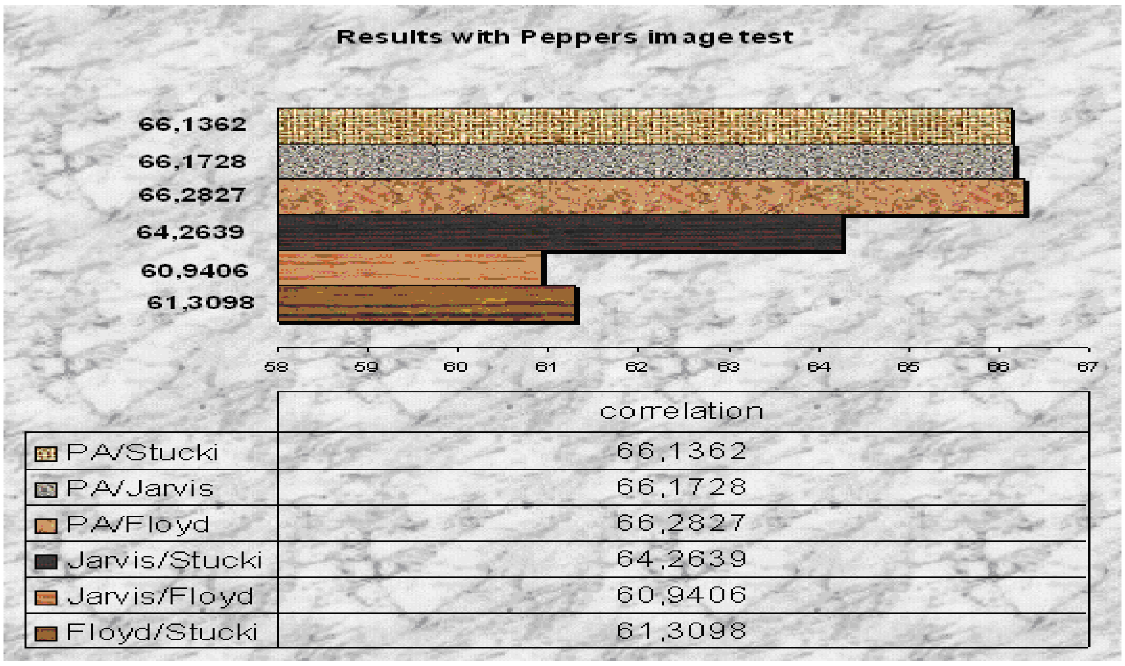
	correlation
■ PA/Floyd	63,7039
■ PA/Jarvis	63,7634
■ PA/Stucki	63,7726

figure (3.8)

The graphs shown above clearly demonstrate that the quality of the halftone images obtained with our algorithm are at least as good as the error diffused ones.

The correlation values between our images are close to those of the corresponding error diffusion halftones with the different filters. Our assumption gives very good results insofar as the maximum of correlation is reached by comparing our images with almost all of the three filters used.

Note here that one of the big problems with improving good halftoning algorithms is that some of them works fine with only certain kind of images and gives results of low quality for others so they are not universal for any kind of images in addition of the fact that there is not any specific definition for the quality. This was observed in our analysis as demonstrated by the results of figure (3.9).



	correlation
▨ PA/Stucki	66,1362
▨ PA/Jarvis	66,1728
▨ PA/Floyd	66,2827
■ Jarvis/Stucki	64,2639
▨ Jarvis/Floyd	60,9406
■ Floyd/Stucki	61,3098

figure (3.9)

3.4.2 Evaluation of the rapidity

In order to examine the speed of our algorithm, we have made a comparative study between error diffusion and our algorithm. We chose Floyd & Steinberg filter owning to the fact that it uses the minimum of neighbors and consequently requires the smallest time of execution among the other filters given before on figure (2.6.a) and (2.6.b).

We have also analyzed the contribution provided by the adoption of our 9pls template carries out by comparing the CPUtime achieved during the execution of our algorithm with that presented in [17b]. Another template has been used in order to investigate the template size's effect on the speed of the operation.

Template	Average CPUtime (s)
Floyd & Steinberg filter	6.5300
Our assumption with 19pls	8.3000
Our assumption with 16pls used in [17b]	7.5200
Our assumption with 9pls	5.2700

Table (4.1) Effect of the size of template on halftoning speed
(the average CPUtime with 256*256 Goldhill image)

The template used in the halftoning process effectively influences its speed. Table (4.1) above (the last three lines) demonstrates that the template including less pixels leads to the fasted solution. Indeed, using 9pls the template accelerates the fetching procedure with a factor of 36.51% with respect to results obtained with 16pls template used in [17b] and shown on figure (3.1). In addition, our algorithm with 9pls template is faster than Floyd & Steinberg error diffusion algorithm. The improvements achieved in terms of speed are summarised in Table (4.2) below.

Speed improvement	Our assumption with 9pls	36.51 %
	Our assumption with 16pls used in [17b]	
	Our assumption with 9pls	19.30 %
	Floyd & Steinberg filter	

Table (4.2) Improvement in terms of speed achieved by our algorithm (using 9pls template) with respect to Floyd & Steinberg error diffusion algorithm

3.5 Conclusion

The halftone images obtained with our algorithm are visually acceptable owing to the fact that they are intended for limit use in which quality does not have really a great importance. The main drawback of the proposed algorithm is that, sometimes, it yields images with less quality. This is primarily due to the smaller LUT size used with 10 patterns corresponding by the 9pls template and also to the small size of images used in the training set which is limited as stated earlier by our resources constraints.

Though we have achieved a bit less in terms of image quality, the proposed algorithm is much faster and as such it can be very useful in those applications where speed is the overriding priority while the quality if not the most important requirement (i.e., fax machines, scanned data,...).

By considering the 9pls temlplate, the speed has been improved by 19.30% and 36.51% when compared the algorithms proposed by Floyd *et al* and Murat *et al*, respectively.

In next Chapter, we will investigate the same template in the inverse problem and show the contribution achieved when using this template.

4th chapter ______________________ Inverse halftoning with Look up Table

4.1 Introduction

Halftone images are often difficult to process without causing severe degradation contrary to grayscale images. That gives the purpose to improve many techniques to reconstruct from their corresponding halftones the continuous images that permits a wide range of operations during transmission or processing, for example, halftones can be compressed efficiently [1], [2]. Inverse halftoning or descreening is the process of retrieving a grayscale image, with a typical wordlength of eight bits, from a halftone, with a wordlength of one bit. Since halftoning is a many-to-one process, there is no unique contone image for a given halftone image. Hence, extra image properties have to be used in inverse halftoning. The simplest inverse halftoning method is to use a simple low-pass filter. Even though it would produce grey levels, it will also tend to blur edges and to destroy fine details. The Look up Table (LUT) is one among several different methods used to achieve this task of inverse halftoning.

In this section, and in order to recreate a greyscale image, with a typical word length of eight bits, from a halftone, we had also introduced the LUT approach. The basic idea was inspired from Murat’s algorithm. However, obtaining high quality was not the aim in this work. Discussion on the experiments results on our algorithm were presented comparing to those achieved by Murat's one in terms of complexity and performance.

Similarly to halftoning, the inverse halftonig algorithm used here is based on a simple table consultation; therefore, it seems to be one among many fast methods used in the literature. The table design phase requires the choice of a template of size N. All of pixels included in this template are used in the prediction to estimate the contone value. Remind us that this template has a significant and incredible effect as well on requirements in memory capacity as on the obtained quality. For binary-valued halftones, a template of halftone pixels corresponds to 2^N possible patterns. The inverse halftoning procedure is im_{xy} = LUT(imh_{xy}), where im_{xy} is the reconstructed continuous_tone image.

In [17a] is proposed an algorithm that uses LUT for inverse halftoning, a tree structure (TLUT) principle is then given. By introducing this last structure, they succeeded in increasing the quality in spite of the enormous requirements in memory capacity. In our everyday life, several situations arise where the factor time has a determining extent such as in network applications; therefore, we can neglect the quality factor.

4.2 inverse halftoning with LUT

The principle with which our table is built will first specified, which will further be adapted to our new considerations.

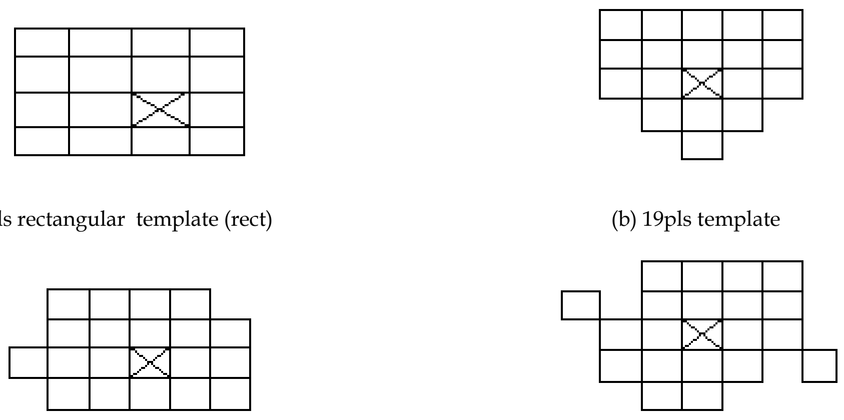
Once the LUT, which is used to store the contone values assigned to corresponding patterns positions, is built, the pixels in the indicated neighbourhood are arranged in a specific order which defines the corresponding pattern then, the contone value at that pixel will be that stored in the appropriate LUT pattern position.

As it has been already mentioned, we need a template to construct our LUT. The same template shown on figure (3.3) is used in this section. The patterns for which we want to predict the contone value, in the table design phase, are well those used in the forward halftoning process. Beyond these considerations, we are interested more to accelerate the process; therefore, we had to take into account when discussing results the obvious output images degradation which will reveal to less Peak Signal to Noise Ratios compared to those achieved by Murat *et al*'s templates, where quality is saved at the expense of increased computational complexity.

Owning to the fact that our algorithm for inverse halftoning is a new version of Murat *et al*'s one, an overview of it is necessary to give.

4.2.1 Basis used for inverse halftoning

An algorithm for the inverse problem using LUT method was exposed in [17a]. Many templates which differ in the number N of pixels were thus used. These templates are shown below.



(a) 16 pls rectangular template (rect)

(b) 19pls template

(c) 19 pls optimal template for Error Diffused images

(d) 19 pls optimal templatefor Ordered Dither images

figure (4.1) Neighborhood used in inverse halftoning

In the next is showing the LUT construction phase that aims to obtain the expected contone value for each pattern in order to assign it to its corresponding position in the LUT.

In its design phase, the LUT for inverse halftoning is obtained from the histogram gathered from a sample set that contains continuous images and their corresponding halftones obtained with error diffused algorithm. This time, and since the original image is an eight-bit image, the LUT to conceive should return a value $LUT(p_0,p_1,...,p_{N-1})$ for each pattern $(p_0,p_1,...,p_{N-1})$ at any input pixel $\in \{0,1,...,255\}$, where N is the number of pixels in the neighborhood defined by the template.

Let us assume that $k(p_0,p_1,...,p_{N-1})$ is the occurrences of the pattern $(p_0,p_1,...,p_{N-1})$ which can occur any number of times in any image of the sample set. The contone values of the pattern $(p_0,p_1,...,p_{N-1})$ at any pixel are denoted by

$$C(p0,p1,...,pN-1,i)\ ,\ \text{for}\ \ i = 0,1,\ldots.., k_{p_0 \ldots\ldots p_{N-1}} - 1$$

We are in front of two cases:

1. $\underline{k(p_0,p_1,...,p_{N-1})\neq 0}$ it means that the pattern exists, the corresponding halftone value $LUT(p_0,p_1,...,p_{N-1})$ of that pattern is indicated by

$$LUT(p0,p1,...,pN-1)=\frac{\sum_{i=0}^{k(p0,p1,...,pN-1)-1} C(p0,p1,...,pN-1,i)}{k(p0,p1,...,pN-1)}$$

2. $\underline{k(p_0,p_1,...,p_{N-1}) = 0}$ it is the case of the non existent pattern

The halftone value attributed to this pattern could be calculated by three methods.

- Low-pass filtering;
- Hamming distance;
- Best linear estimator.

The last method seems to provide the best results in terms of achieved quality and of which the principle could be summarized in the following steps:

- Let M be the number of all the existing patterns in the taken sample halftone images; those for which $k(p_0,p_1,...,p_{N-1})\neq 0$.

- Define the pattern matrix A of which the elements are

$$A(i,j)=p_{i,j} \quad \text{for} \quad \begin{matrix} i=0,1,\ldots,M-1 \\ j=0,1,\ldots,N-1 \end{matrix}$$

- The LUT vector b with elements

$$b(i)=LUT(p_{i,0},p_{i,1},\ldots,p_{i,N-1}) \quad \text{for } i=0,1,\ldots,M-1$$

- Solve the set of equations:

$$\begin{matrix} \text{pattern } \equiv 0 \\ \text{pattern } \equiv 1 \\ \vdots \\ \vdots \\ \text{pattern } \equiv M-1 \end{matrix} \quad \Rightarrow \quad \begin{bmatrix} p_{0,0} & p_{0,1} \cdots\cdots & p_{0,N-1} \\ p_{1,0} & p_{1,1} \cdots\cdots & p_{1,N-1} \\ \vdots \cdots & \cdots\cdots \cdots & \\ \vdots \cdots & \cdots\cdots \cdots & \\ p_{M-1,0} & p_{M-1,1} \cdots\cdots & p_{M-1,N-1} \end{bmatrix} \times \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ \vdots \\ x_{M-1} \end{bmatrix} = \begin{bmatrix} b(0) \\ b(1) \\ \vdots \\ \vdots \\ b(M\text{-}1) \end{bmatrix}$$

the best linear estimator will be the least squares solution to x. It is given by

$$x = (A^T A)^{-1} A^T b \qquad \text{and} \qquad y = \lfloor p_0, p_1, \ldots, p_{N-1} \rfloor x$$

- For each pattern $(p_0,p_1,\ldots,p_{N-1})$ for which $k(p0,p1,\ldots,pN-1)=0$, the contone value $LUT(p_0,p_1,\ldots,p_{N-1})$ will be decided as follows:

$$LUT(p0,p1,\ldots,pN-1) = \begin{cases} 0, & \text{if } y<0 \\ 255, & \text{if } y>255 \\ \text{round}(y), & \text{otherwise} \end{cases}$$

In this analysis, the number of nonexistent patterns depends on the template as well as the halftoning method used to obtain the halftones in the training set. In addition, when adding more pixels to the template the quality of the output images gets better but this makes the LUT bigger. Besides, the neighbourhood to be used in the prediction should kept relatively small.

4.2.2 The proposed algorithm principle

As it has already evoked, our purpose, through this work, is to decrease the time of execution of the algorithm while keeping an acceptable quality. what we seek, in other words, is to make a compromise quality / speed.

To achieve the goal above, we had thought in both the design of the LUT and during the inverse halftoning process to use the template of size N=9 shown in figure (3.3). Consequently, we have 2^9 instead of 2^{19} different patterns used in [17a]. It was remarked that the use of only 10 patterns are sufficient to reach an acceptable visual quality.

The LUT design algorithm can be summarised in the following.

4.2.2.1 LUT design

For each image from the training set do:

1st step: Read the input image;

2nd step: Halftone it with Floyd & Steignberg error diffusion algorithm.

3rd step: In a raster scan order, for each pixel define the neighbors according to our 9pls template.

4th step: Compute the vector K denoting the occurrences of all the patterns.

At this stage, and using the 9pls template, it has remarked that all of the patterns existed at least one time in the whole predefined training set images. Therefore, we were only in front of the first case presented in 4.2.1, where $k(p_0,p_1,\ldots,p_{N-1}) \neq 0$.

5th step: Calculate the halftone value $LUT(p0,p1,\ldots,p8)$ of any pattern $(p0,p1,\ldots,p8)$ by

$$LUT(p0,p1,\ldots,p8) = \frac{\sum_{i=0}^{k(p0,p1,\ldots,p8)-1} C(p0,p1,\ldots,p8,i)}{k(p0,p1,\ldots,p8)}$$

where $C(p0,p1,\ldots,p8)$, for $i = 0,1,\ldots.., (k_{p_0\ldots.p_8}-1)$ denotes the contone value of any pattern $(p0,p1,\ldots,p8)$ at the processed pixel.

In the following is illustrated the inverse halftoning procedure when the designed table is used. The input is an image with only two levels; black and white and the output is a continuous-tone image.

4.2.2.2 The inverse halftoning phase

This phase involves the following steps:

1st step: Read in such a raster scan order the halftone of the input image which is obtained in step2. (*In fact, this step is not necessarly done since in practice the input is already a halftone image*).

2nd step: Identify for each input pixel at the location (x,y), the neighbors $\left(p_0, p_1, \ldots, p_{N-1}\right)$ according to the used template (in our case N=9);

3rd step: The corresponding output pixel is the value stored in the defined pattern (step 2), say

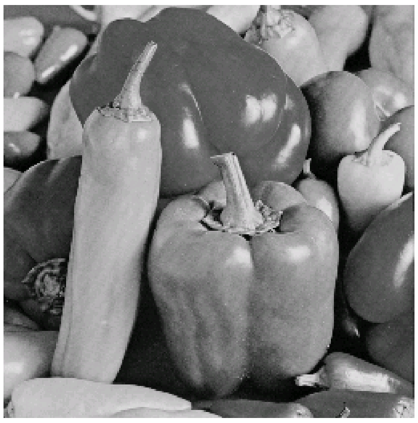
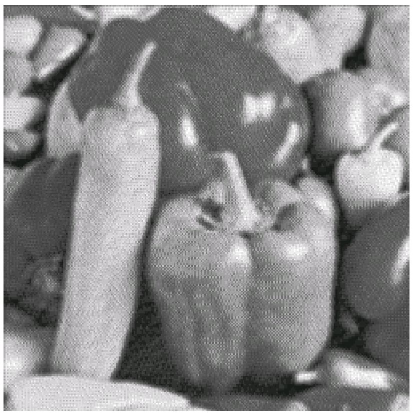
$$\text{output}(x,y) = \text{LUT}\left(p0, p1, \ldots, pN-1\right) \big|_{x,y}$$

We thought of investigating the template effect in order to reinforce the assumption used here; therefore, we had studied tow cases; 19pls and 9pls where in both of them, the new statement is considered.

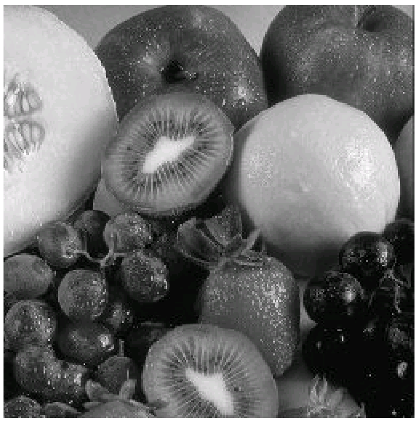
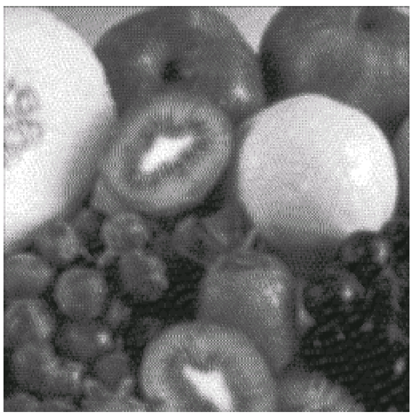
4.3 Experiment results

In order to demonstrate the efficiency of our assumption, we had applied the proposed algorithm on several error diffused images. We will first expose the results obtained when using 19 pixels in the neighbourhood; therefore, the table constructed and then used contains 19 contone values assigned to 19 different patterns.

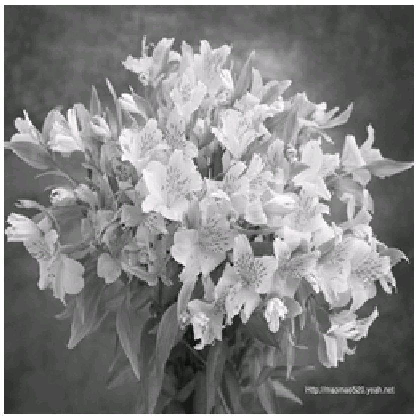
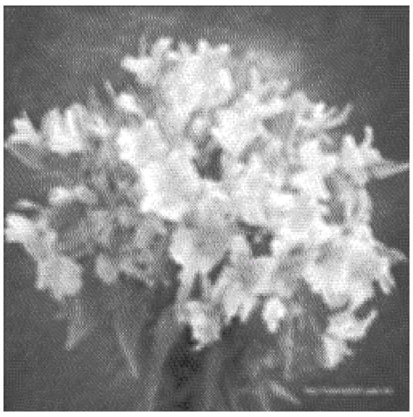
4.3.1 Experiment results with the 19pls template

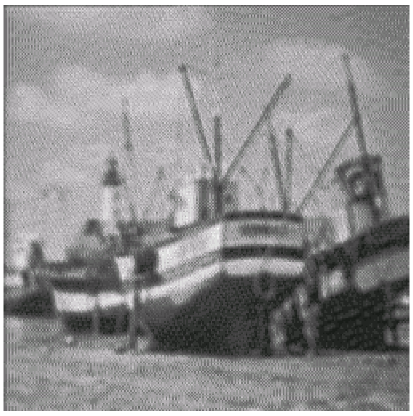
(a) psnr = 20.5788 (b)

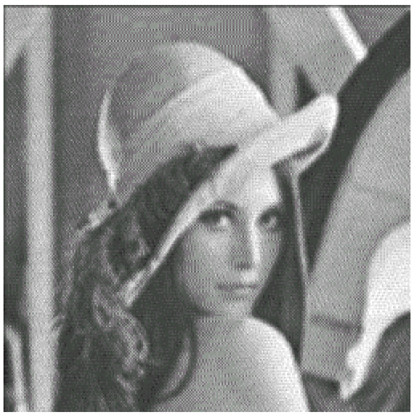
(a) psnr = 20.0661 (b)

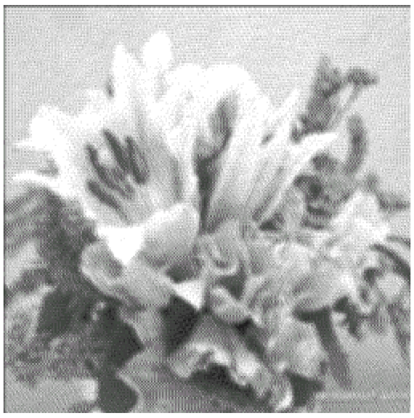
(a) psnr = 20.9987 (b)



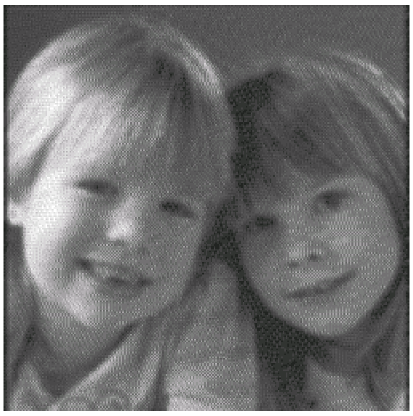
(a) psnr = 20.4554 (b)



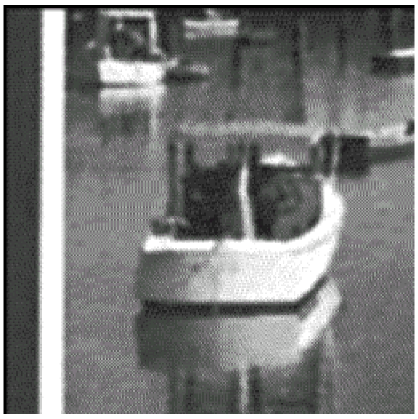
(a) psnr = 20.8494 (b)



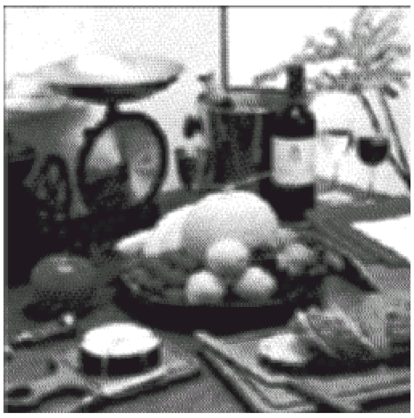
(a) psnr = 20.0081 (b)



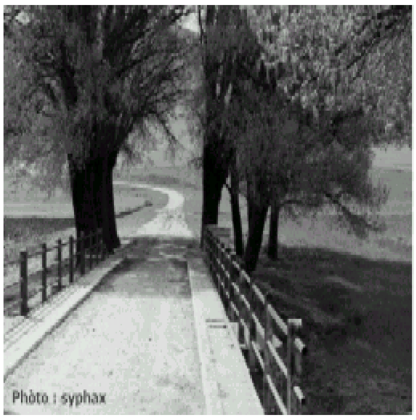
(a) psnr = 21.5370 (b)

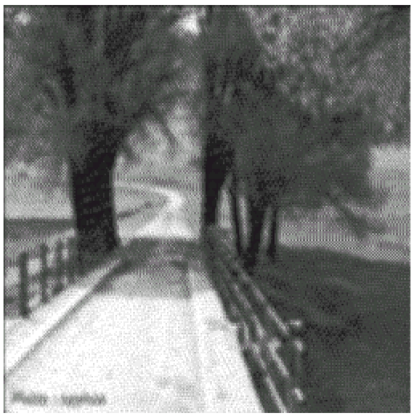


(a) psnr = 21.5265 (b)

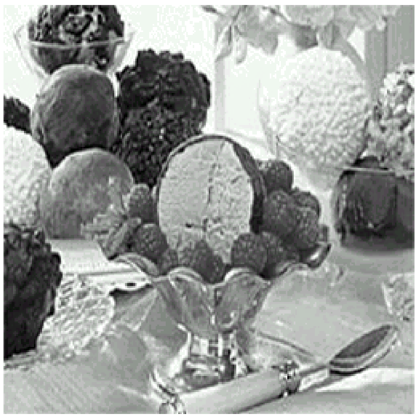


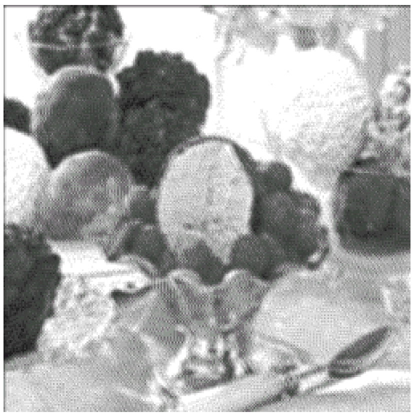
(a) psnr = 18.0772 (b)



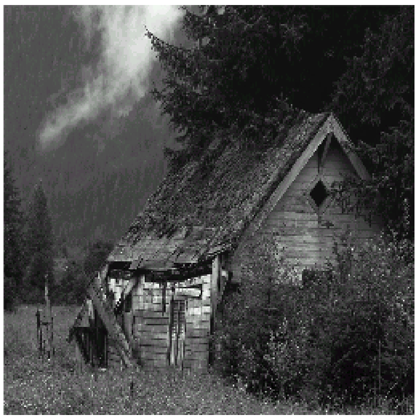


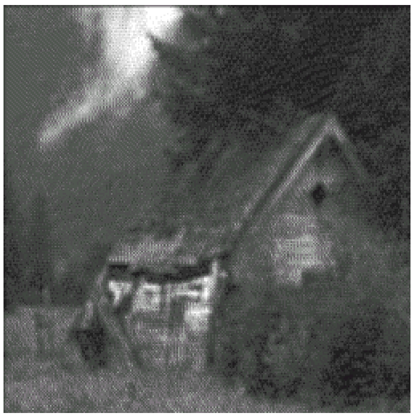
(a) psnr = 20.0786 (b)



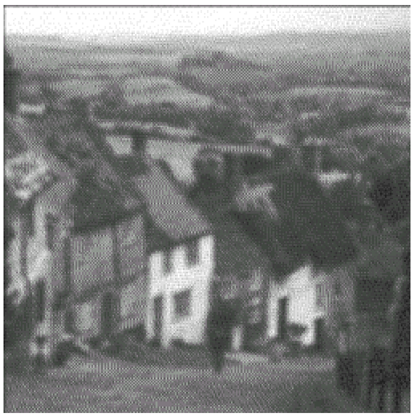


(a) psnr = 19.4148 (b)

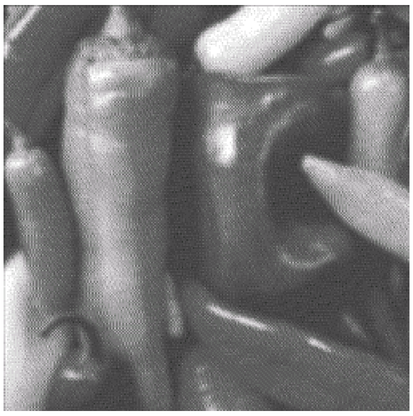




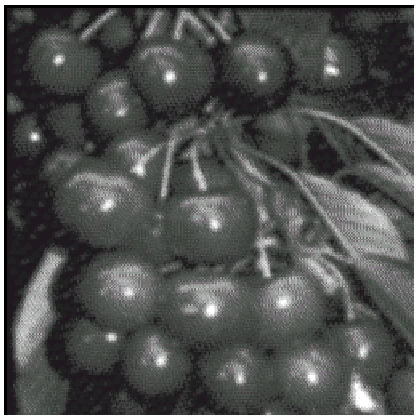
(a) psnr = 19.5124 (b)



(a) psnr = 20.6285 (b)



(a) psnr = 22.1847 (b)



(a) psnr = 20.5495 (b)

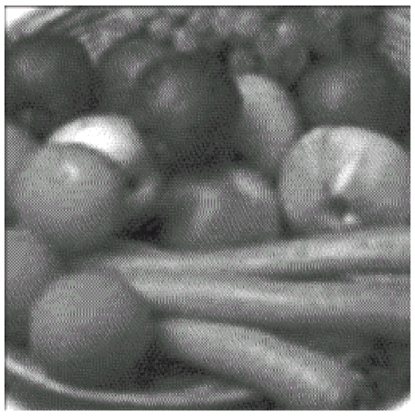
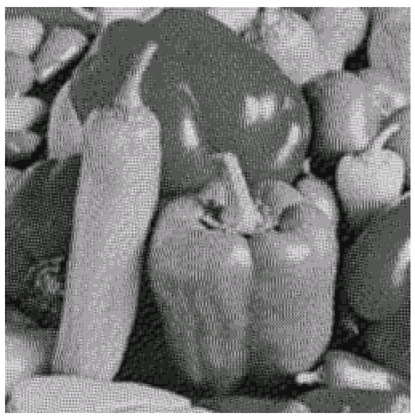
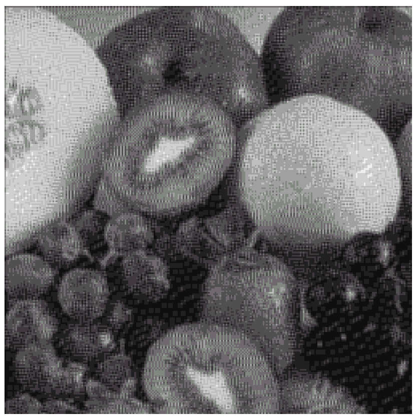
	
(a) psnr = 22.1662	(b)

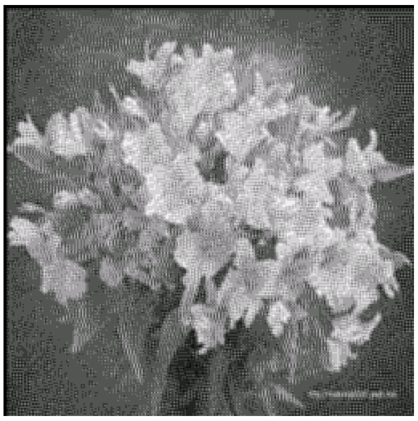
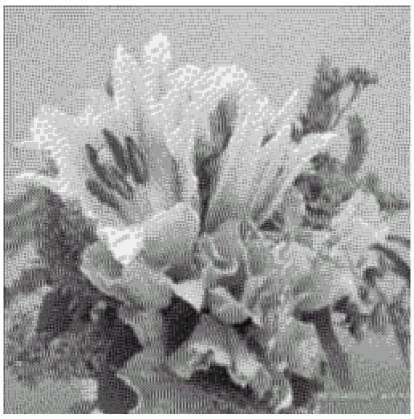
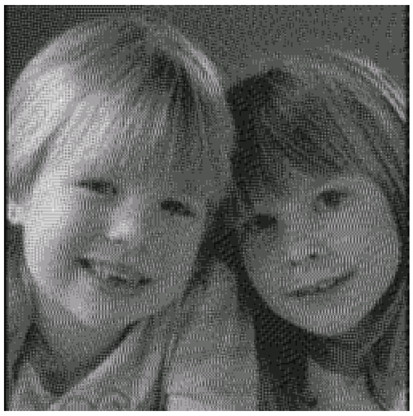
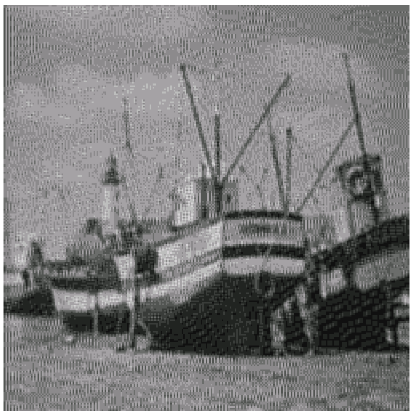
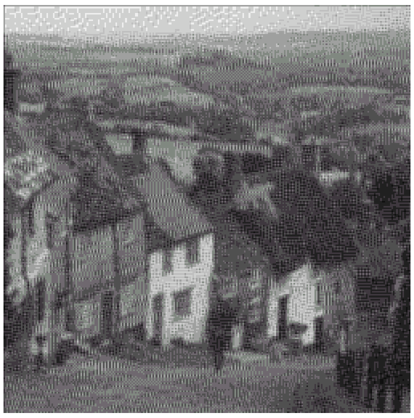
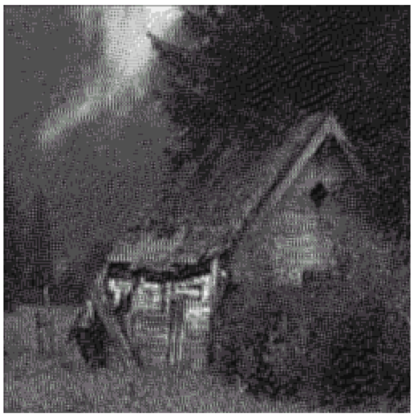
figure (4.2) Results obtained with 19pls template: considering only 19 patterns.
(a) The original images (b) Their inverse halftones

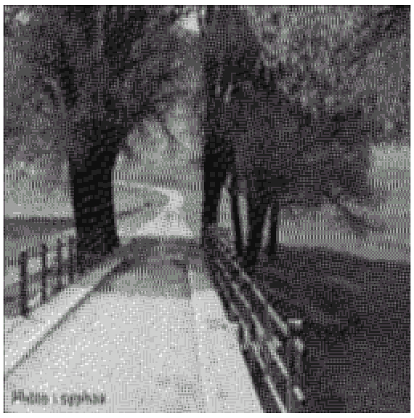
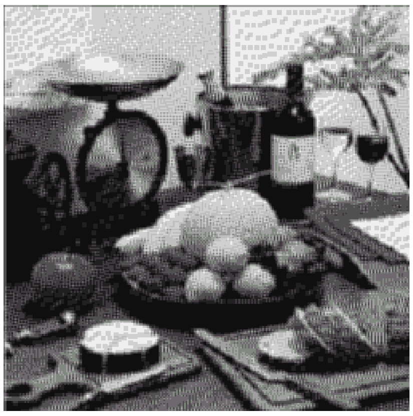
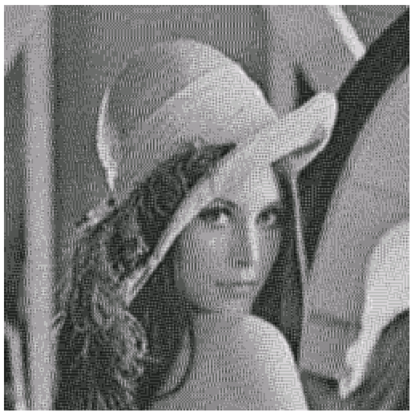
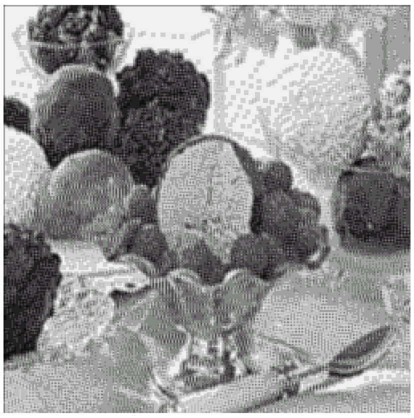
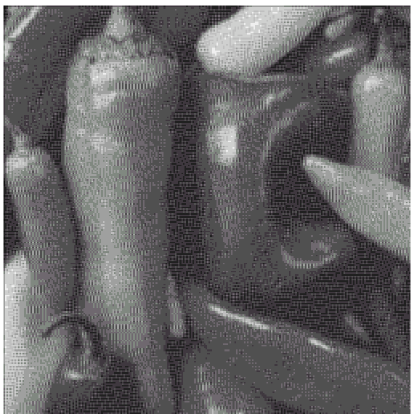
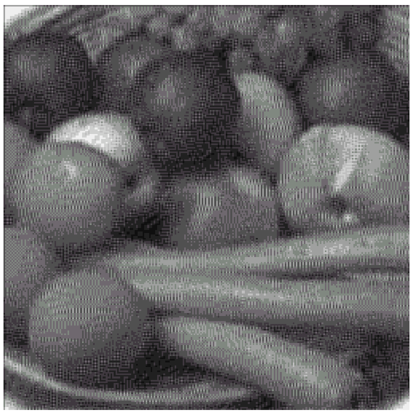
4.3.2 Experiment results with the 9pls template

The table constructed this time contains 10 contone values assigned to 10 different patterns.

For visual comparison we now show the inverse halftone images for the same sample taken in the last passage.

	
psnr =22.3593	psnr = 21.7866

	
psnr = 22.5947	psnr = 21.6588
	
psnr = 22.9915	psnr = 22.8954
	
psnr = 21.8604	psnr = 21.0821

	
psnr = 22.2288	psnr = 19.6746
	
psnr = 22.7175	psnr = 21.4438
	
psnr = 23.5178	psnr = 23.7158

psnr = 23.1692	psnr = 23.2991

figure (4.3) The inverse halftoned images obtained with the proposed algorithm

4.4 Discussion and analysis

The proposed algorithm has two main strengths allowing its rapidity. The first one comes from the reduction of number of search location owning to the fact that the constructed tables in both halftoning and inverse halftoning processes have only 10 cells instead of 2^9 would be considered in [17a]. The second one comes from the reduction of the number of pixels by reducing the size of the template which is in fact a consequence of the first. All these considerations lead to accelerating the algorithm of fetching the contone value during the inverse halftoning process. In the contrast, these strengths led unfortunately to its drawback; the decline of the output image quality as it will be indicated next.

4.4.1 Performance

Digital image and video processing systems are generally involved with signals that are meant to convey reproductions of visual information for 'human consumption'. Tradeoffs between system resources and the visual quality are typically involved in designing such systems, and accurate quality measurement algorithms are needed in order to make these tradeoffs efficiently. The obvious way of measuring quality is to solicit the opinion of human observers. However, such subjective evaluations are not only cumbersome and expensive, but they also cannot be incorporated into automatic systems that adjust themselves in real-time based on the feedback of output quality.

Traditionally, researchers have focussed on measuring quality by quantifying the similarity between a distorted (or test) image and a reference image that is assumed to have perfect quality. The Mean Squared Error (MSE), which gives difference between the test and the reference image, is widely used to quantify (the loss of) visual quality.

In our analysis the performance was measured by using peak-signal-to-noise-ratio function which is given by

$$PSNR = 10\log\left(\frac{255^2}{(1/n\times m)\times\sum_{0}^{n-1}\sum_{0}^{m-1}(original\_image - inverse\_halftone)^2}\right)$$

where (n,m) represent the size of the original image.

Template	Average PSNR
Our assumption with 19pls: only 19 patterns are considered	20.0732
Our assumption with 9pls	21.7459
Rect (result obtained from [17a])	26.50
19pls (result obtained from [17a]):where all of the 2^{19}patterns are considered	26.61

Table (4.1) Quantitative quality evaluation: comparison of the achieved average PSNRs

The Table (4.1) (*the second and last lines*) above shows a bit drop of quality of our images with respect to results given in [17a]; nevertheless, images obtained by our new assumption resemble the original ones extremely well. This is mainly caused by the number of patterns considered in each case. The fact of increasing the number N of pixels to be used in the prediction phase of each contone value (*the table design phase*) that implies the increase of the template's size, gives implicitly the chance to any pattern among all the 2^N patterns to be present in the table with its particular value what enriches this designed table. Moreover, our table with the considered assumption cannot capture the various patterns; they could have different values even though they define the same halftone pattern. Indeed, a table with 2^N different values is richer and gives obviously more precise results than that only comprising N values (20.0732 *dB and* 26.61 *dB with the same template*).

Methods producing better images are usually more computationally intensive and time consuming. The complexity in terms of computation time or number of arithmetic operations required could be the main factor in many applications. In our analysis, the algorithm complexity is rather given by the first since the technique used here does not involve any computation at all. This thought is investigated and reported in the following.

4.4.2 Rapidity evaluation

The proposed algorithm, as well as all other produced, has value only after having established its features. Our purpose is based on the fact that decreasing the number of pixels used in the prediction allows us to decrease the processing time that involves the algorithm to obtain the inverse halftone of a given image.

The evaluation of the speed of our algorithm was of primary importance in order to prove its efficiency. We thought of making an implementation hardware but it was decided, for lack of time and material, to follow simplest but less reliable method which consists in making a comparative study between the proposed algorithm and that of Murat *et al* implemented under the same conditions; i.e. considering only 19 patterns instead of 2^{19}. We had simulated in *Matlab* both of them and estimated the inverse halftoning time given by the CPU time.

Algorithm	Average CPUTime (s)
Our assumption with 19 pls template	8.1600
Our assumption with 9 pls template	4.8650

Table (4.2) The average CPUtime (256*256 Peppers image)

In fact, the second column shows the time required by Matlab process during the inverse halftoning phase, which implies that, for a given image, the only factor affecting this time, and making the difference between the two studied cases, is the number of pixels which the template comprises, in other words, even when the whole patterns (i.e., 2^N) are considered, the time processing of the same image will be identical when one considers only N patterns.

The Table (4.2) shows an uncontested speed up improved by our algorithm. In the other hand, when compared to results given by Murat *et al*, it should however be admitted that we lose from the quantitative point of view by considering the 9 pls if compared to 19 pls in [17a], but not in an alarming way, because visually, this algorithm offers images of good quality, close to the original ones. This lost quality is largely compensated, in contrast, by the decreasing of the time processing, the speediness of the process is therefore achieved, as demonstrated in the following summary table.

LUT size	**Our table with 9 pls**	10
	Table constructed with 19 pls	2^{19}
Memory requirements	**Our table with 9 pls**	$10*2^8$
	Table constructed with 19 pls	$2^{19}*2^8$
Loss in quality		**18.2800 %**
Speed improvement		**40.3800 %**

Table (4.3) Comparison between our algorithm and Murat's *& al* in terms of memory requirements, quality and rapidity

According to this table, it should come out that the inverse process has been accelerated with a speed increase of 40.38% when compared to the algorithm proposed by Murat *et al* [17a]. Furthermore, the memory capacity requirements are considerably reduced (10 bytes compared to 1 Kbyte which could have been considered if one had taken all the 2^9 patterns considered by Murat *et al*'s algorithm). However, this has resulted in a degradation of about 18.28% in quality of the produced images, through visually speaking this still remains an imperceptible impairment.

To overcome this drawback, it was observed that by increasing the number of images in the training set PSNR could attain suitable extents approaching better more values reached by other best algorithms; it is worth mentioning that this solution has no effect on the process rapidity since it is performed at the LUT design phase.

We will now briefly show how the inverse halftone quality is affected by a range of factors according to our experiments. This includes the training set contents, i.e., the number and kind of halftones used in it, the image test itself on which the LUT is applied and then concluding what kind of images is more suitable to our algorithm.

4.4.3 Effect of the training set

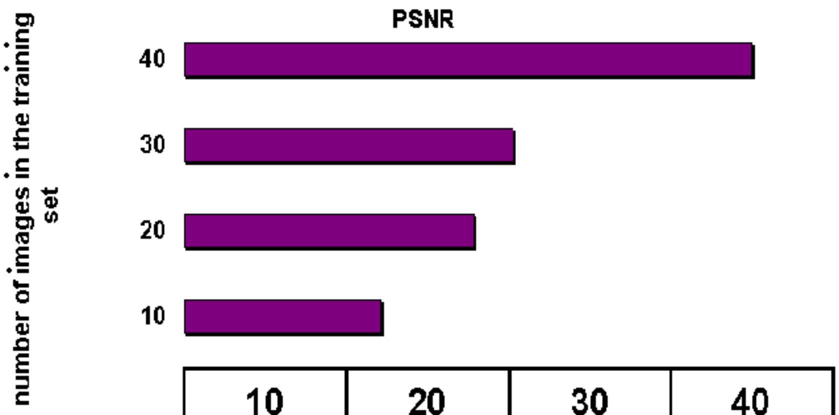
The training set, as well-known includes a number of continuous images and their corresponding halftones.

4.4.3.1 The algorithm used to obtain the halftones

We had examined tow cases; first using Floyd & Steinberg error diffused images then dithered ones to construct the table. It is found out, as envisaged, that error diffused halftones gives the best improvements.

4.4.3.2 The number of images in the training set

We had mentioned that with a properly chosen training set in terms of type would give suitable results. Another mean to go so lies in using much images in this sample set. It is the appropriate way to regain the quality lost with respect to our novel assumption. The following figure clarifies well again this adequate solution.



	10	20	30	40
PSNR	22.8425	22.9562	23.0047	23.2991

figure (4.4) Increasing PSNR by increasing the number of images in the sample set

4.4.4 Effect of the test image type

It was noticed that improvements in PSNR are when applying our algorithm on error diffused rather then dithered halftones. This implies that the inverse halftone quality depends on starting halftones, i.e., better halftones allow to reach the better inverse halftone quality.

4.5 Conclusion

Experimental results show that images obtained with our algorithm and the original images are virtually indistinguishable and the resemblance, which appears, becomes increasingly obvious when it is applied on images initially of good quality.

Even though the new assumption has come the cost of small decline of the output's quality revealed by less average PSNR, the results were visually quite acceptable. On the other hand, we had successfully achieved our aim in accelerating the fetching procedure.

It is the obvious price to pay in order to accelerate the inverse halftoning procedure. The results have efficiently shown a speed up of the halftone value fetching; however this comes with some deterioration of the image quality measured in terms of the PSNR which is further can be removed or at least lowered by using more images in the sample set.

5th chapter General conclusion

Halftoning and inverse haftoning represent increasingly interesting areas which are still developing to satisfy several application requirements in industry, in particular those intended for viewing and printing.

The demand for fast image reproduction increases every day. Much effort and research has been directed towards developing new methods and models for different tasks in the printing and viewing problems. Since quality and speed are generally conflicting, it is usually advantageous to come up with intermediate techniques that provide better images in less time.

In this thesis, we have investigated both forward and inverse halftoning using the Look up Table technique (LUT). The algorithms proposed and their analysis have shown that the LUT method is very fast and the results obtained are of acceptable and good quality. It was remarked that the quality of the final produced image is very much dependent on the starting training set used for the construction of these tables. The outcome quality is mainly affected by the characteristics of the halftoning method used to obtain halftones in the sample set as well in the forward halftoning as in the inverse context. In addition, the size of these images is also important since the richer is the designed table the best will be the output image quality. In contrast, the method presented in this thesis requires a powerful computer to process the images within a reasonable time. This is what we lack considering the new assumption that involves using images of lower size and, unfortunately is reflected on our results in term of quality.

The main work here has been particularly focused on presenting the inverse halftoning algorithm and showing the results produced by the proposed algorithm. Using the 9pls template had effectively resulted in an increase of the speed of processing as demonstrated by much less CPUtime.

The main contribution of this research work is also related to the acceleration of the process of halftoning of grey level images. The use of the 9 pixel template has effectively resulted in increasing the speed of the computation.

The evaluation of the speed of the algorithm was of primary importance in order to prove its efficiency. It was felt that a hardware implementation can be a solution but it has been decided, for lack of time and specialist equipment, to follow the simplest but less reliable method which consists in making a comparative study between the algorithm proposed and that of Murat *et al* which was implemented under the same conditions. This thought is sufficient as judgement to determine which of the two cases is indeed fastest owning to the fact that, for a same given image, the only factor affecting the processing time, and making the difference between the two studied cases, is the number of pixels of the template used. The results have shown a speed up of the process execution; however this comes with some deterioration of the image quality measured in terms of the PSNR. It was also observed that by increasing the number of images in the training set could be an adequate solution to overcome this drawback; it is worth mentioning that this solution has no effect on the speed of the algorithm.

The programs used for producing the images in this thesis are all written in *Matlab6*.5, which has the advantage of providing very useful predefined toolboxes but the programs written using such a language are usually executed slower than those written in other native languages such as C or C++.

One suggested future work will to use write the programs in C or C++ with a view to further increase the speed of the processes. Another important possibility for speeding up the programs is to find a way of dividing the image into a number of sub-images and process them simultaneously without less quality drop.

To achieve this, an idea, which is under investigation consists in using both 19pls and 3pls templates: the first to design the tables (that assures achieving the best quality) while the second will be used in the output fetching phase (to speed up processes).

Another interesting area can be a direct application of the halftone results with Look up Table in practical field such as watermarking. In fact this point is under investigation where one can apply watermarking techniques using visual cryptography to bi-level images obtained by our halftoning algorithm.

An expansion for color images could also be considered.

References

[1] D. Neuhoff and T. Pappas, “Perceptual coding of images for halftone display,” *IEEE Trans. Image Processing*, vol. 3, pp. 1–13, Jan. 1994.

[2] Y. Kim, G. Arce, and N. Grabowski, “Inverse halftoning using binary permutation filters,” *IEEE Trans. Image Processing*, vol. 4, pp. 1296–1311, Sept. 1995.

[3] B. E. Bayer, “An optimum method for two level rendition of continuous-tone pictures,” in *Proc. IEEE Int. Conf. Commun., Conf. Rec., 1973, pp. 26-11 – 26-15.*

[4] J. Luo, R. de Queiroz, and Z. Fan, “A robust technique for image descreening based on the wavelet transform,” *IEEE Trans. Signal Processing*, vol. 46, pp. 1179–1194, Apr. 1998.

[5] R. A. Ulichney, *Digital Halftoning.* Cambridge, MA: MIT Press, 1987.

[6] R. A. Ulichney, “Dithering with blue noise,” in *Proc. IEEE,* vol. 76,pp. 56–79, Jan. 1988.

[7] D. E. Knuth, “Digital halftones by dot diffusion,” *ACM Trans. Graphics,*vol. 6, pp. 245–273, Oct. 1987.

[8] S. Kollias and D. Anastassiou, “A unified neural network approach to digital image halftoning,” *IEEE Trans. Signal Processing,* vol. 39, pp.980–984, Apr. 1991.

[9] M. Analoui and J. Allebech, “Model_based halftoning using direct binary search,” in *Proc. 1992 SPIE/IS&T Symp. Electronic Imaging Science Technology*, vol . 1666, San Jose, CA, Feb. 1992, pp. 96-108.

[10] T. N. Pappas and D. L. Neuhoff, “Least-squares model-based halftoning,” in *Proc. 1992 SPIE/IS&T Symp. Electronic Imaging Science Technology*, vol. 1666, San Jose, CA, Feb. 1992, pp. 165-176.

[11] D. Lieberman and J. P. Allebach, “Efficient model based halftoning using binary search,” in *Proc. 1997IEEE Int. Conf. Image Processing*, Santa Barbara, CA, Oct. 1997.

[12] J. P. Allebach and R. N. Stradling, “Computer-aided design of dither signals for binary display of images,” *Appl. Opt.*, vol. 18, pp. 2708-2713,Aug. 1, 1979.

[13] J. P. Allebach and Q. Lin, “FM screen design using DBS algorithm,” in *Proc. 1996 IEEE Int. Conf. Image Processing*, vol. II, Lausanne, Switzerland, Sep. 1996, pp. 549-552.

[14] R. Rolleston and S. J. Cohen, “Halftoning with random correlated noise,” *J. Electron. Imag.,* vol. 1, pp. 209_217. Apr. 1992.

[15] T. Mitsa and K. J. Parker, “Digital Halftoning Technique Using a Blue Noise Mask,“ *J. Opt. Soc. Am. A*, Vol. 9, No. 11, pp. 1920-1929, November 1992.

[16] A. N. Netravali and E. G. Bowen, "Display of dithered images," *Proc. SID* , Vol. 22, no. 3, pp. 185-190, 1981.

[17a] M. Murat and P.P. Vaidyanathan, "Look Up Table (LUT) Inverse Halftoning," *Proceedings of ISCAS*, 2000.

[17b] —, "Look Up Table Method for Image Halftoning,", *Proceedings of ICIP*, 2000.

[18] —, "Template Selection for LUT Inverse Halftoning and Application to Color Halftones," *Proceedings of ICASSP*, 2000.

[19] —, "Tree-structured method for improved LUT inverse halftoning,", *Proceedings of EUSIPCO*, 2000.

[20] M. A. Seldowitz, J. P. Allebach, and D. E. Sweeney, "Synthesis of digital holograms by direct binary search" Appl. Opt Vol. 26, pp. 2788-2798, 1987.

[21] M. Schulze and T. Pappas, "Blue noise and model-based halftoning,"in *Proc. SPIE, Human Vision, Visual Processing and Digital Display V*,1994, vol. 2179, pp. 181–194.

[22] Robert W. Floyd and Louis Steinberg, "An Adaptive Algorithm for Spatial Grayscale," *Proceedings of the Society for Information Display* 17 (2) 75-77, 1976

[23] Yuefeng Zhang, "Line Diffusion: A Parallel Error Diffusion Algorithm for Digital Halftoning," *The Visual Computer,* 12 (1) 40-46, 1996.

[24] P. Takis Metaxas, "Method and System for Parallel Error-Diffusion Dithering," *US Patent* 6,307,978 (Oct. 23, 2001).

[25] S. Kollias and D. Anastassiou, "A unified neural network approach to digital image halftoning," *IEEE Trans. Signal Processing,* vol. 39, pp. 980–984, Apr. 1991.

[26] M. Analoui and J. P. Allebach, "New results on reconstruction of continuous–tone from halftone," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing,* vol. 3, 1992, pp. 313–316.

[27] S. Hein and A. Zakhor, "Halftone to continuous–tone conversion of error–diffusion coded images," *IEEE Trans. Image Processing,* vol. 4, pp. 208–216, Feb. 1995.

[28] Z. Fan, "Retrieval of images from halftones," in *Proc. IEEE Int. Symp. Circuits Systems, May 1*992, pp. 313–316.

[29] P. W. Wong, "Inverse Halftoning and kernel estimation for error diffusion," *IEEE Trans. Image Processing,* vol. 4, pp. 486–498, Apr. 1995.

[30] M. Ting and E. Riskin, "Error-diffused image compression using a binary-to-grayscale decoder and predictive pruned tree-structured vector quantization," *IEEE Trans. Image Processing*, vol. 3, pp. 854–858, Nov. 1994.

[31] Z. Xiong, M. T. Orchard and K. Ramchandran "Inverse halftoning using wavelets," *IEEE Trans. Image Processing*, vol. 8, pp. 1479–1482, Oct. 1999.

[32] T. D. Kite, N. Damera-Venkata, B. L. Evans, and A. C. Bovik, "A fast, high quality inverse halftoning algorithm for error diffused halftones," *IEEE Trans. Image Processing*, vol. 9, pp. 1583–1592, Sep. 2000.

ملخص النتائج

تركزت أعمالنا أولا على عرض تحويل صورة تحتوي تدرج اللون الرمادي الى أخرى رقمية ذات لونين فقط (أسود '0' وأبيض'1') ثم العملية العكسية. و في كلتا الحالتين استخدمنا تقنية لوحة الترميز (الترقين) التي تعطي نتائج جيدة بالاضافة الى كونها سريعة باعتبارها لا تحتاج الى عمليات حسابية. الطريقة المتبعة تحتاج أساسا إلى اختيار شبكة تحدد العدد N و موقع عناصر الصورة المعنية والتي تستعمل عند مرحلتي بناء اللوحات و أثناء استعمالها أين تطبق اللوحات على الصورة المراد تحويلها للحصول على الصورة المرافقة.

بعد تحليلهما الخوارزميتان المقدمتان في [a17] و[b17] عرضتا بشكل مختلف بادخال الاعتبارات التالية :

- باعتبار التأثير الهام للشبكة السالفة الذكر على سرعة البحث عن القيم المرافقة لعناصر الشبكة المستعملة , استعملنا فقط 9 عناصر عوضا عن 16 و 19 عنصرا اخذا على التوالي في المرجعين السابقين.
- و بدلا من أخذ كل احتمالات عناصر الشبكة المعبر عنها ب 2^N ,اعتبرنا فقط N+1.

ان هذه الاعتبارات الجديدة أدت فعلا الى تسريع العمليتين بالنحو التالي :

- حسنت سرعة البحث عن الصورة ثنائية اللون بمقدار *36,51%* و *19,30%* بالنسبة الى **[b17]** و ***Floyd et al*** على التوالي.
- نجحنا أيضا في تسريع العملية العكسية بمقدار *40.38%* مقارنة بنتائج **[a17]** , كما تمكنا أيضا من تقليص الاحتياجات في الذاكرة من *10 octets* الى. *1 Koctet* و هذا بالرغم من خسارتنا ل *18.28%* من ناحية جودة الصور, ولكن هذا، لحسن الحظ لا يكاد يلاحظ عليها.

Résumé de résultats

Le travail effectué a été concentré, en particulier sur la présentation des algorithmes de tramage et de tramage inverse en utilisant l'approche de la table de transcodage (LUT), ce qui a conduit à de très bon résultats. Cette technique exige principalement le choix d'une trame (the template) qui désigne le voisinage utilisé, d'une part lors de la construction des tables et d'autre part lors de l'utilisation de ces tables. Les tables ainsi conçues sont appliquées sur l'image d'entrée afin d'obtenir l'image de sortie qui lui correspond avec une simple consultation de la table. Ce qui fait d'elle une méthode rapide où aucun calcul n'est requis.

En réalité, nous avons analysés et modifiés les algorithmes présentés par Murat *et al.* pour le tramage et le tramage inverse en prenant de nouvelles considérations:

- Le nombre de pixels que désigne la trame utilisée dans cette technique a un rôle déterminant dans les deux procédés analysés du fait qu'il affecte principalement la rapidité. A cet effet, nous n'avons utilisé que 9 pixels au lieu de 16 et 19 pixels respectivement dans le tramage et le tramage inverse.
- Au lieu de prendre toutes les possibilités engendrées par la trame choisie (c.à.d. 2^N ; où N est le nombre de pixels dans cette trame), on n'en considère que N éléments de trames (patterns). Notons que dans le cas où la trame est de forme rectangulaire on considère N+1 patterns.

Ces nouvelles considérations en utilisant la trame dénotée par 9pls a eu efficacement comme résultat l'augmentation de la vitesse des deux procédures de telle manière que:

- *Dans le cas du tramage, la vitesse a été améliorée par 19,30% et 36,51% une fois comparée aux algorithmes proposés par Floyd et al et Murat et al respectivement.*
- *On a réussi d'accélérer la rapidité de la procédure d'inverse halftoning de 40.38% par rapport à l'algorithme de Murat et al. De même nous avons réduit considérablement les exigences en espace mémoire (10 octets par rapport à 1 Koctet qui aurait pu être nécessaire, si l'on avait pris tout les 2^9 patterns considérés par Murat et al). Par contre, nous n'avons perdu que 18.28% en terme de qualité ce qui n'a pas affecté visuellement les images obtenues.*