

**IMAGE RESTORATION:  
CONTENT ADAPTIVE MESH MODEL**

BY  
MARC LAIN

Electrical and Computer Engineering  
Illinois Institute of Technology

Approved \_\_\_\_\_  
Advisor

Chicago, Illinois  
October 2010



# ACKNOWLEDGEMENT

This Thesis would not have been possible without the help of Dr. Brankov who was my advisor and guided and encouraged me throughout all my research . I am also very grateful to Dr. Yang who helped me during this task.

I would also like to thank all my family and my girlfriend for they support during all the time I have been working in this Thesis.

Last but not least, I show my gratitude to my colleagues and friends who helped me during my studies by sharing their knowledge and ideas with me.

# TABLE OF CONTENTS

ACKNOWLEDGEMENT.....	iii
LIST OF TABLES.....	vi
LIST OF FIGURES.....	vii
ABSTRACT .....	ix
CHAPTER	
1. INTRODUCTION.....	1
1.1 Image Enhancement and Image Restoration.....	2
2. CLASSICAL IMAGE RESTORATION TECHNIQUES.....	4
2.1 Degradation Model .....	4
2.2 Basic Concepts .....	7
2.3 Image Restoration Techniques .....	12
3. IMAGE RESTORATION WITH CONTENT ADAPTIVE MESH MODELING FILTER .....	22
3.1 Basic Concepts .....	22
3.2 Algorithm .....	24
3.3 Implementation Issues of CAMM .....	39
3.4 Numerical Results .....	45
4. CONCLUSION .....	53
4.1 Summary .....	53
4.2 Future Work.....	53

APPENDIX

A. RELATION BETWEEN HIGH FREQUENCY CONTENT AND SECOND PARTIAL DERIVATIVES.....	54
B. FEATURE MAP COMPUTATION BY THE HESSIAN EIGENVALUES .....	55
C. LEAST SQUARES FITTING DERIVATIONS.....	57
REFERENCES .....	59
OTHER BIBLIOGRAPHY .....	61

# LIST OF TABLES

Table 1. Number of operations for DFT and FFT .....	10
Table 2. Comparison between linear interpolation and least squares fitting for different ratios .....	37
Table 3. Comparison between linear interpolation and least squares fitting when the degraded image is used.....	38
Table 4. PSNR and Quality Index for each method when the original image “Lena” is available .....	48
Table 5. PSNR and Quality Index for each method when the original image “Lena” is not available .....	50
Table 6. PSNR and Quality Index for each method when the original image “Tech” is not available.....	52

# LIST OF FIGURES

Fig. 1 Degraded astronomical image.....	1
Fig. 2 Degraded Computed Tomography image .....	2
Fig. 3 Degraded and restored images from a license plate .....	3
Fig. 4 Degradation Model of an image.....	4
Fig. 5 Two-dimensional Gaussian filter.....	6
Fig. 6 Degradation model in the frequency domain .....	8
Fig. 7 Restored images using Inverse filter .....	14
Fig. 8 Comparison between Inverse filter and Standard Wiener filter.....	16
Fig. 9 Relation between PSNR and K for different values of the noise variance....	18
Fig. 10 Restored images by the Standard Wiener filter when the original spectrum is not available .....	19
Fig. 11 Restored images using the Parametric Wiener filter .....	20
Fig. 12 Graph of relation between PSNR and Gamma.....	21
Fig. 13 Feature Map for different sensitivities.....	26
Fig. 14 Floyd-Steinberg dithering algorithm applied to “Lena” .....	27
Fig. 15 Error diffusion of the Feature Map .....	29
Fig. 16 Node Placement for different number of mesh nodes .....	31
Fig. 17 Delaunay Triangulation.....	32
Fig. 18 Mesh Structure.....	32
Fig. 19 Restored image by Linear Interpolation .....	33
Fig. 20 Contribution of a mesh node to the image.....	34

Fig. 21 Restored images using the nodal values from the original and the degraded images.....	35
Fig. 22 Restored images using Least Squares fitting and Linear Interpolation with the original image.....	37
Fig. 23 Restored images using Least Squares fitting and Linear Interpolation with the degraded image.....	38
Fig. 24 Comparison between the use of the degraded, the filtered, and the original image for the Feature Map extraction .....	41
Fig. 25 Restored images with different values of Gamma.....	43
Fig. 26 Evolution of the MSE between original and restored images .....	45
Fig. 27 Images restored with CAMM and classical methods .....	47
Fig. 28 Images restored with CAMM and Wiener filter .....	49
Fig. 29 Image “Tech” represented with the mesh model.....	51
Fig. 30 “Tech” images restored with CAMM and Wiener filter .....	52

# ABSTRACT

Digital image restoration plays a very important role in many fields such as surveillance or medical imaging, where it can be used to obtain high-resolution images so that a more accurate analysis can be performed. In this study we firstly introduce the reader to classical image restoration techniques, such as the Inverse filter or the Wiener filter. However, the main objective of this study is to evaluate a new approach to digital image restoration, which is based on a mesh model of the image. In order to create the mesh model, the digital image is non-uniformly sampled with the use of an algorithm based on a feature map of the image and the classical Floyd-Steinberg error-diffusion. As a result, the sampling is adapted to the content of the image, so more samples are placed in areas with more image detail (high-frequency areas) and less samples are placed in smooth regions (low-frequency areas). The samples (also called mesh nodes) are then connected using the Delaunay triangulation algorithm in order to form the mesh structure. An iterative least-squares fitting algorithm is then used to calculate the intensity of the mesh elements in order to obtain an accurate approximation of the image. The proposed method is an effective image restoration technique for digital images degraded by blur and noise. Moreover, the use of a content-adaptive mesh model (CAMM) means a compression of the image, because less samples are needed in order to represent the image since they are adapted to the image features. The results obtained demonstrate that the adaptive mesh model method can outperform the classical image restoration techniques presented in this study.

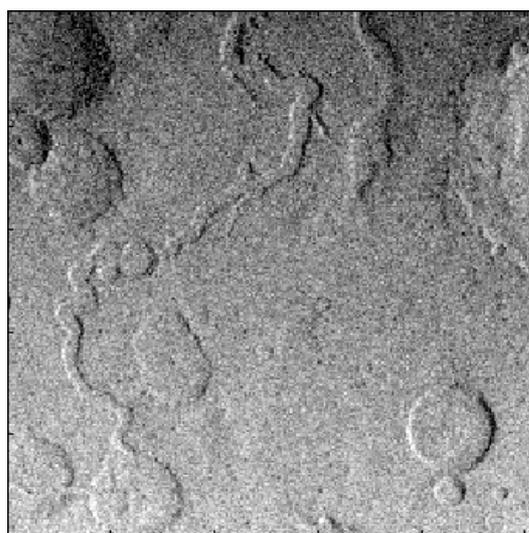
*Keywords*—Digital image restoration, Inverse filter, Wiener filter, mesh model, non-uniform sampling, Floyd-Steinberg error-diffusion, Delaunay triangulation, least-squares fitting.

# CHAPTER 1

## INTRODUCTION

Images are used in many fields in order to record and display useful information. Due to the acquisition process, images can suffer some degradation and, as a consequence, loss of important information. The main causes of image degradation are blur and noise introduced by different systems, devices or operations depending on the nature of the image. The noise can be added by the medium where the image is recorded (scattering), by the imaging system (noise in the sensor), and by the process of quantization of a digital image for its storage. On the other hand, blur can be introduced by vibrations of the imaging system relative to the object, by fluctuations of the width of the beam in a scanning system, and by an optical system (camera) out of focus.

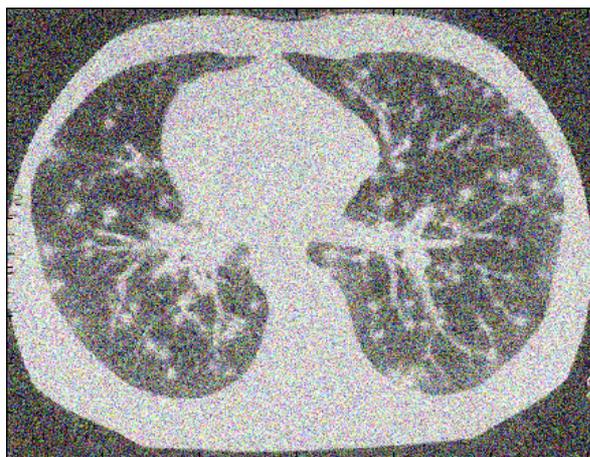
In the field of astronomy [1], images are usually degraded by the addition of Poisson noise [2] caused by the lack of light. Usually, the image is also contaminated by Gaussian noise [3], [4] which arises from the electronic components in the imaging system [4]. Shown in Fig. 1 is an astronomical image degraded by Poisson and Gaussian noise. Another problem in astronomical images is the blur [4] created by the imperfections of the mirror of the telescope, the exposure time of the camera, or changes in the index of refraction of the atmosphere.



*Fig. 1 Image from the Moon degraded by Poisson and Gaussian noise.*

*Image: [www.freeimages.co.uk](http://www.freeimages.co.uk)*

Medical imaging is another scientific field where degraded images are a problem. For example, in magnetic resonance (MR) images we can observe some Rician noise, whereas Gaussian noise can appear in computed tomography (CT) images and Poisson noise in X-ray images and mammograms [5]. Fig. 2 shows a computed tomography image of lungs degraded by additive white Gaussian noise (AWGN).

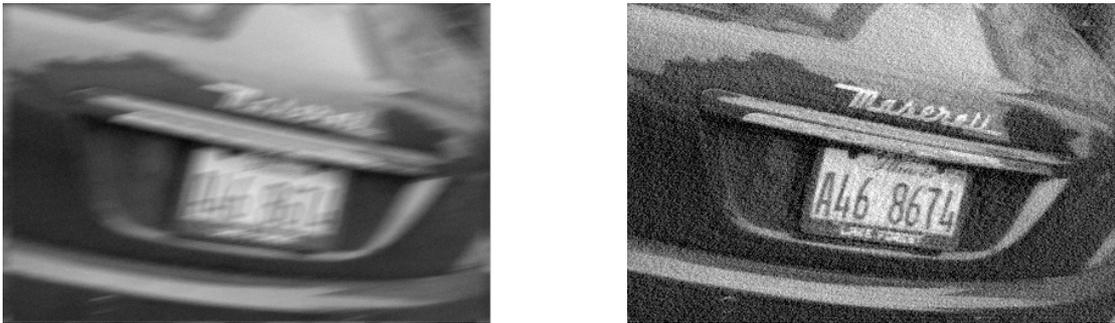


*Fig. 2 Computed tomography image degraded by Gaussian noise.  
Image: Public Health Image Library, Centers for Disease Control and Prevention.*

## **1.1 Image Enhancement and Image Restoration**

Image enhancement and image restoration are two image processing techniques whose goal is to improve the corrupted image characteristics with the help of some mathematic fields such as linear algebra, estimation theory and signal processing techniques. However, image enhancement and image restoration differ in the way they manipulate the image. The former is designed to improve the image features in the sense of subjective visual quality, that is, make it more pleasant to an observer. By contrast, image restoration techniques are oriented toward achieving some criterion of goodness, such as the mean squared error or the peak signal to noise ratio, which yields to an estimation of the non-degraded image. Another difference is that image enhancement techniques sometimes process sets of image parts or regions while in image restoration the full image is always treated. Moreover, image restoration techniques need some *a priori* knowledge about the degrading system to obtain an estimated model of the degrading system and apply the inverse operation in order to recover the non-blurred and noise-free image.

Besides astronomy and medical imaging, there are also other important applications of digital image enhancement and restoration. For instance, these techniques are often used in law enforcement and forensic science. The restoration of corrupted security videotapes or the improvement of license plates images have been crucial in many crime investigations. In Fig. 3(a) we can see an image of a license plate degraded by motion blur and Gaussian noise and in Fig. 3(b) the restored image using the Wiener filter, which will be explained later. We can see the higher quality of the restored image, where the numbers and letters are more clear.



*Fig. 3(a) Image from a license plate degraded by motion blur and Gaussian noise; (b) restored image using the Wiener filter.*

*Image: [www.freeimages.co.uk](http://www.freeimages.co.uk)*

## CHAPTER 2

### CLASSICAL IMAGE RESTORATION TECHNIQUES

#### 2.1 Degradation Model

As said before, image restoration techniques (also known as image deconvolution or de-blurring techniques) try to inverse the process of degradation in order to recover of the original image, thus, it is necessary some *a priori* knowledge about the noise and the blurring filter. Unfortunately, usually just some of their characteristics are known, so it is necessary to estimate them before designing the filter.

The degradation model is represented in Fig. 4. The blurring system  $H$  modifies the original non-degraded image  $f(x,y)$ . The imaging system also introduces additive noise  $n(x,y)$  to the blurred image. The result of the acquisition is the degraded output image  $g(x,y)$ .

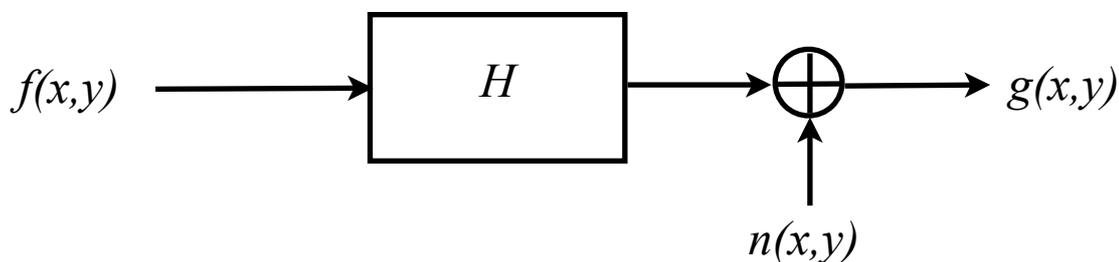


Fig. 4 Degradation model of an image.

This process can also be written mathematically as follows:

$$g(x,y) = H[f(x,y)] + n(x,y). \quad (1)$$

There are many different kinds of noise that can affect images. Some of them are additive, others are multiplicatives and in a few cases they are neither additive nor multiplicative. The most common additive noise is the Gaussian [3], which has a Gaussian distribution (also known as normal distribution) of its values. Speckle [6] is a multiplicative granular noise which uses to degrade synthetic aperture radar (SAR) images used, for instance, in environmental monitoring and in military targeting.

Finally, the salt-and-pepper noise [7] (also called impulse noise) is an example of noise which can't be modeled as an additive either multiplicative noise. This kind of noise is caused by noisy transmission channels or defective hardware during the image acquisition. In this study, we will consider an additive white Gaussian noise (AWGN), i.e., with Gaussian probability density distribution (PDF) and zero mean. The main reason of choosing this kind of noise is that it is a good approximation of the noise found in most of the image restoration problems. This is proved by the central limit theorem [8] which says that the sum of various random variables (such as noise) can be approximated by a random variable with a Gaussian PDF.

$$n(x,y) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2+y^2}{2\sigma^2}}. \quad (2)$$

The blurring system  $H$  is normally modeled by a linear and spatially invariant function, that is, it affects the same way every part of the image. This assumption is taken because it simplifies significantly the mathematical expressions and operations carried out during the filter design.

The most common blur models used in image restoration are the motion blur, the out-of-focus blur and the atmospheric turbulence blur [4]. We will focus on the blur caused by a Gaussian point-spread function (PSF), which can be computed easily.

$$h(x,y) = ke^{-\frac{x^2+y^2}{2\sigma^2}}, \quad (3)$$

where  $k$  is a constant to satisfy that

$$\iint_{\forall x,y} h(x,y) dx dy = 1. \quad (4)$$

Fig. 5 shows a graph of the two-dimensional Gaussian blurring filter with a standard deviation  $\sigma = 0.85$ .

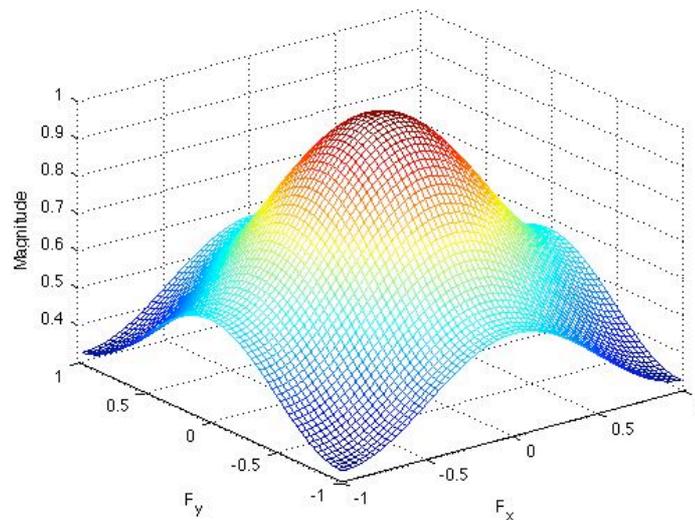


Fig. 5 Two-dimensional Gaussian filter

Taking this into consideration, the equation above can be rewritten as follows:

$$g(x,y) = h(x,y) * f(x,y) + n(x,y), \quad (5)$$

where  $h(x,y)$  is the point-spread function of the blurring system and the operator  $*$  is a two-dimensional convolution. Using the continuous notation and considering that  $h(x,y)$  is a linear and spatially invariant filter, the expression becomes

$$g(x,y) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} h(x-\alpha, y-\beta) f(\alpha, \beta) d\alpha d\beta + n(x,y). \quad (6)$$

However, digital images are non continuous functions, where the samples are the pixels. In this case, the equation above becomes

$$g(x,y) = \sum_{i=1}^M \sum_{j=1}^N h(x-i, y-j) f(i,j) + n(x,y), \quad (7)$$

where  $M$  and  $N$  are the number of pixels in the first and second dimensions respectively.

## 2.2 Basic Concepts

### 2.2.1 Frequency Domain and Fourier Transform

Digital image filters can be implemented in both spatial domain and frequency domain [9] (also known as spectral domain or Fourier domain). However, image restoration filters are usually designed in the frequency domain, since it simplifies the formulations, reduces the computational costs, and allows the implementation of filters for larger images. The Fourier transform [9] is a very important tool in this sense, not only in image restoration but also in other image processing techniques such as image enhancement, compression and analysis because it allows the transformation of spatial functions, like images, to the frequency domain and the use of some of its useful properties for the computation of the algorithms.

The continuous one-dimensional Fourier transform (FT) is defined as follows:

$$FT \{f(x)\} = F(u) = \int_{-\infty}^{+\infty} f(x)e^{-2j\pi ux} dx, \quad (8)$$

where  $f(x)$  is the original function defined in the spatial domain,  $F(u)$  its Fourier transform, and  $u$  is the variable in the frequency domain.

The inverse process, that is, obtaining  $f(x)$  from its Fourier transform  $F(u)$  is called the inverse Fourier transform and is defined as follows:

$$FT^{-1} \{F(u)\} = f(x) = \int_{-\infty}^{+\infty} F(u)e^{2j\pi xu} du. \quad (9)$$

The two-dimensional Fourier transform must be used in the case of functions of two variables. It can be obtained by extending the one-dimensional case:

$$FT \{f(x,y)\} = F(u,v) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x,y)e^{-2j\pi(ux+vy)} dx dy \quad (10)$$

and the inverse transform

$$FT^{-1} \{F(u,v)\} = f(x,y) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} F(u,v)e^{2j\pi(xu+yv)} du dv. \quad (11)$$

When the function is discrete instead of continuous it is necessary the use of the discrete Fourier transform (DFT) [9]:

$$DFT^{-1}\{F(u)\} = f(x) = \frac{1}{N} \sum_{u=0}^{N-1} F(u) e^{2j\pi \frac{ux}{N}}, \quad (13)$$

$$DFT\{f(x)\} = F(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) e^{-2j\pi \frac{ux}{N}}, \quad (12)$$

where  $M$  and  $N$  are the number of samples in the first and second dimensions ( $x$  and  $y$ ), respectively. From now on we will consider  $M$  and  $N$  to be the same number for simplicity.

The use of some of the Fourier transform properties involve an advantage in many image processing operations. One of the most important advantages is the relation between convolution in the spatial domain and product in the frequency domain, and vice-versa. It can be demonstrated that

$$f(x) * h(x) \stackrel{FT}{\Leftrightarrow} F(u)H(u), \quad (14)$$

$$f(x)h(x) \stackrel{FT}{\Leftrightarrow} F(u) * H(u), \quad (15)$$

which helps to simplify most of the equations and operations in many restoration algorithms.

Considering this property, the degradation model defined in (5) can be rewritten as follows:

$$G(u, v) = F(u, v)H(u, v) + N(u, v). \quad (16)$$

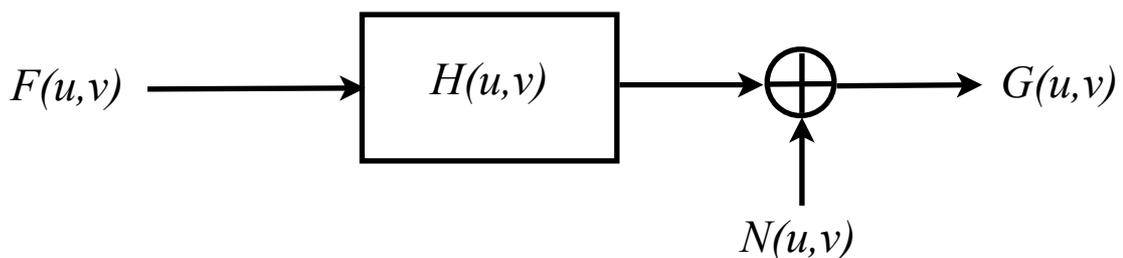


Fig. 6 Degradation model in the frequency domain

Periodicity is another important property of the discrete Fourier transform. It means that the DFT is periodic function with a period of  $N$  samples, that is

$$F(u, v) = F(u + N, v) = F(u, v + N) = F(u + N, v + N). \quad (17)$$

This property will be used later in section 2.2.2 when talking about circulant and block-circulant matrices.

Another important property of the Fourier transform is the relation with the spatial density of an image. For example, the slowest varying frequency component of the Fourier transform  $F(u, v)$  (i.e.,  $u$  and  $v$  equal to zero) corresponds to the average intensity of the image. Thus, the low frequencies correspond to the smooth parts of an image and the high frequencies to the fast changes of intensity of an image such as edges.

When the number of samples of the function increases the computational cost (number of operations) to obtain the discrete Fourier transform increases as well. That is, the calculation of a DFT can be very slow if there is a big amount of samples, so it means a disadvantage. In the case of an image of  $N$  samples the number of operations needed is proportional to  $N^2$ . Fortunately, there are some algorithms, called fast Fourier transform (FFT) algorithms, that help to accelerate this process. By using FFT algorithms the number of operations is reduced to  $N \log_2 N$  which means an advantage of  $N / \log_2 N$ .

Table 1 shows the comparison between the use of the DFT and the FFT. For example, to compute the direct DFT of a function of 256 samples 65,536 operations are needed while using the FFT it is reduced to 2,048 operations which means an improvement of 32.

N	DFT ( $N^2$ )	FFT ( $N\log_2N$ )	Improvement ( $N/\log_2N$ )
2	4	2	2.00
8	64	24	2.67
32	1,024	160	6.40
128	16,384	896	18.29
256	65,536	2,048	32.00

Table 1 Number of operations needed when using the DFT compared to the number needed for the FFT.

### 2.2.2 Degrading Model Using Block-Circulant Matrices

From the discrete equation of the degrading model defined in (7), we can use the matrix notation by converting the  $M \times N$  image matrices to  $MN \times 1$  column vectors

$$\mathbf{g} = \mathbf{H}\mathbf{f} + \mathbf{n}, \quad (18)$$

where  $\mathbf{g}$  is the column vector of the degraded image samples

$$\mathbf{g} = \begin{bmatrix} g(0) \\ g(1) \\ \vdots \\ g(MN - 1) \end{bmatrix}_{MN \times 1}, \quad (19)$$

$\mathbf{f}$  is the column vector of the original image samples

$$\mathbf{f} = \begin{bmatrix} f(0) \\ f(1) \\ \vdots \\ f(MN-1) \end{bmatrix}_{MN \times 1}, \quad (20)$$

the noise  $\mathbf{n}$  has exactly the same structure as  $\mathbf{g}$  and  $\mathbf{f}$ , and finally  $\mathbf{H}$  represents the degrading system, which is an  $MN \times MN$  matrix formed by  $M^2$  blocks of size  $N \times N$ . Every block has the form

$$\mathbf{H}_j = \begin{pmatrix} h(j,0) & h(j,N-1) & \cdots & h(j,1) \\ h(j,1) & h(j,0) & \cdots & h(j,2) \\ \vdots & & \ddots & \vdots \\ h(j,N-1) & h(j,N-2) & \cdots & h(j,0) \end{pmatrix}_{N \times N}, \quad (21)$$

then, the degrading system matrix  $\mathbf{H}$  is constructed as follows:

$$\mathbf{H} = \begin{pmatrix} \mathbf{H}_0 & \mathbf{H}_{M-1} & \cdots & \mathbf{H}_1 \\ \mathbf{H}_1 & \mathbf{H}_0 & \cdots & \mathbf{H}_2 \\ \vdots & & \ddots & \vdots \\ \mathbf{H}_{M-1} & \mathbf{H}_{M-2} & \cdots & \mathbf{H}_0 \end{pmatrix}_{MN \times MN}. \quad (22)$$

Every block matrix is a circulant matrix. It means that its elements are placed in a circulant way. In other words, the last element in the first row is equal to the first element in the second row. The elements of the next rows are placed the same way. This is due to the assumption that the system function  $h(x,y)$  is periodical in both dimensions  $x$  and  $y$ . In addition, the system matrix is a block-circulant matrix, since its blocks are placed in a circulant way as well. Large images can be computed faster thanks to these properties.

## 2.3 Image Restoration Techniques

As said before, image restoration techniques use some *a priori* knowledge about the blurring filter and the noise that degraded the image during the acquisition process in order to obtain a good approximation of the original image from the corrupted one. This study focuses on three classical image restoration techniques: the Inverse Filter [4], [10], the Standard Wiener Filter [4], [11], [12], [13], and the Parametric Wiener Filter [11], [12], [13], [14]. These three methods use the mean square error (MSE) as a criterion of goodness.

### 2.3.1 Inverse Filter

A simple and direct restoration method is the Inverse filter [4], [10]. This method tries to reconstruct the image by applying the inverse of the blurring filter to the degraded image neglecting the additive noise. Using the matrix notation introduced in (8) we can find the estimated image by the minimization of the function

$$J(\tilde{\mathbf{f}}) = \|\mathbf{g} - \mathbf{H}\tilde{\mathbf{f}}\|^2. \quad (23)$$

Assuming that  $\mathbf{H}$  is a square matrix and its inverse exists, it yields to the solution

$$\tilde{\mathbf{f}} = \mathbf{H}^{-1}\mathbf{g}. \quad (24)$$

This equation can be rewritten using the Fourier transform and the diagonalization of block-circulant matrices as described in [4] and [16].

$$\tilde{\mathbf{f}} = \mathbf{H}^{-1}\mathbf{g} = (\mathbf{W}\mathbf{D}\mathbf{W}^{-1})^{-1}\mathbf{g}, \quad (25)$$

where  $\mathbf{W}$  is the matrix containing the eigenvectors of  $\mathbf{H}$  and  $\mathbf{D}$  is the diagonal matrix where its elements are the eigenvalues of  $\mathbf{H}$ .

If we develop this expression as follows

$$\mathbf{W}^{-1}\tilde{\mathbf{f}} = \mathbf{D}^{-1}\mathbf{W}^{-1}\mathbf{g}, \quad (26)$$

we can use the Fourier notation

$$\tilde{F}(u,v) = \frac{G(u,v)}{H(u,v)}. \quad (27)$$

If the degraded image is substituted by the degrading model, the previous equation becomes

$$\tilde{F}(u,v) = F(u,v) + \frac{N(u,v)}{H(u,v)}. \quad (28)$$

It is clear from (28) that the noise contributes negatively to the estimation of the original image. The main problem of the inverse filter method is that if the blurring filter values are close to zero, the noise term is the dominant term in the restored image. This is a likely situation, since the blurring filter is a low-pass filter so its values decrease fast for high frequencies and, unlike the original image, the noise is not null at those frequencies so it is amplified.

Shown in Fig. 7(a) is the degraded image by blur (PSNR = 26.928 dB) and in Fig. 7(b) is the restored image by inverse filtering (PSNR = 29.4481 dB). Fig. 7(c) shows a blurred and noisy image (PSNR = 21.5567 dB) and Fig. 7(d) the restored image using the inverse filter (PSNR = -9.6997 dB). As we can see with the images and also with the values of the Peak Signal-to-Noise Ratio (PSNR), the restoration of a blurred image is successful but, by contrast, the result of the restoration of a blurred and noisy image is not a good approximation.

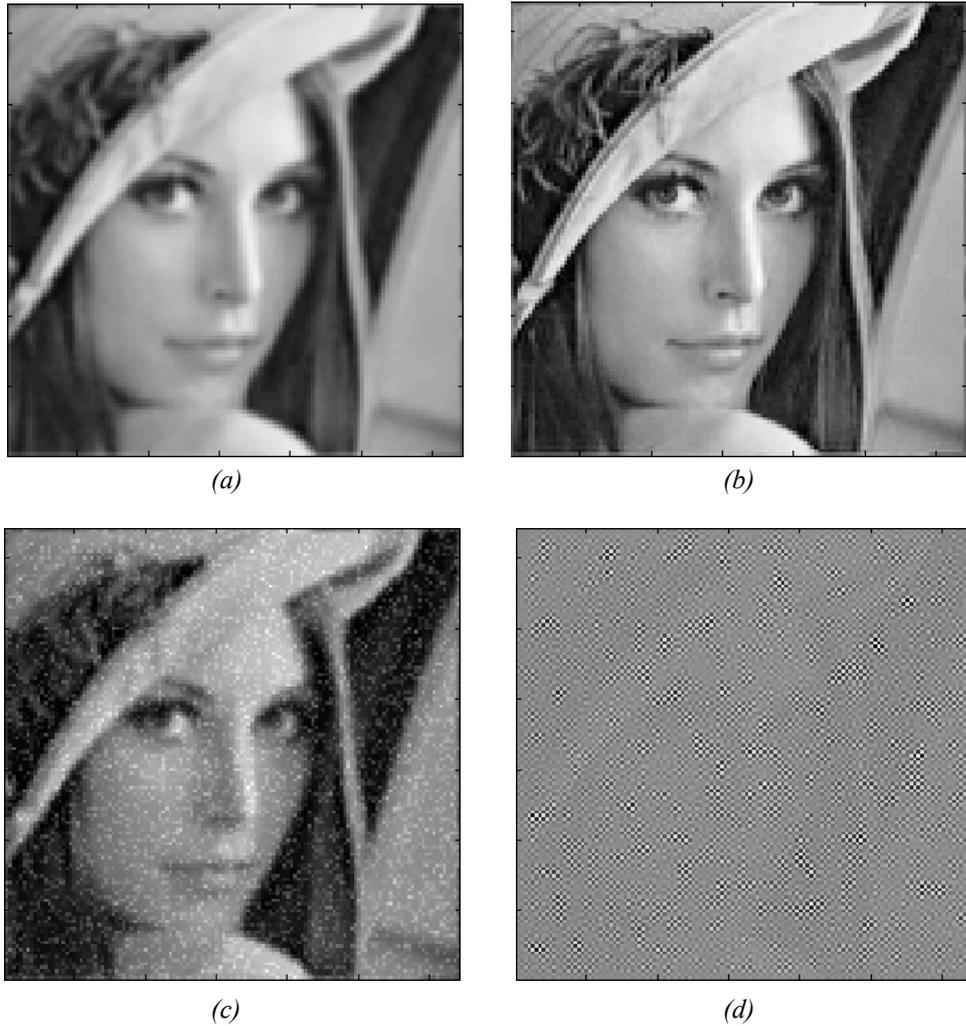


Fig. 7 (a) Image from a section of “Lena” degraded by Gaussian blur; (b) restoration of (a) using the Inverse filter; (c) Image degraded by Gaussian blur and Gaussian noise; (d) restoration of (c).

### 2.3.2 Standard Wiener Filter

The Standard Wiener filter [4], [11], [12], [13] (also known as Least Mean Square filter or Wiener deconvolution) attempts to eliminate the effect of noise amplification in those high frequencies where the noise power is higher than the image power (low signal-to-noise ratio). Thus, the zero values of the blurring filter are not a problem.

The restored image is the result of applying the Standard Wiener filter to the degraded image. In the frequency domain that is

$$\tilde{F}(u,v) = W(u,v)G(u,v), \quad (29)$$

where  $\tilde{F}(u, v)$  is the restored image and  $W(u, v)$  is the Fourier transform of the Standard Wiener filter.

If we substitute the equation

$$G(u, v) = H(u, v)F(u, v) + N(u, v) \quad (30)$$

in (29), then the restored image becomes

$$\tilde{F} = WHF + WN. \quad (31)$$

Note that in (31) the notation used doesn't includes the frequency variables  $u$  and  $v$ , i.e.  $F$  is equivalent to  $F(u, v)$ . In the next expressions this notation is used for simplicity.

The Standard Wiener filter  $W$  is designed as the filter which minimizes the mean square error between the restored image and the original undegraded image, that is

$$W = \arg \min_w E \left\{ \|\tilde{F} - F\|^2 \right\}. \quad (32)$$

If we substitute  $\tilde{F}$  for its equivalent expression (31), this equation becomes

$$W = \arg \min_w E \left\{ \|(WHF + WN) - F\|^2 \right\}. \quad (33)$$

Taking into consideration that the noise  $N$  and the original image  $F$  are uncorrelated and the noise has zero mean, it can be reduced to the following expression:

$$W = \arg \min_w E \left\{ (WH - 1)(WH - 1)^* S_f + WW^* S_n \right\}, \quad (34)$$

where  $S_f$  and  $S_n$  are the frequency spectra of the original image and the noise respectively

$$S_f = E \{ FF^* \}, \quad (35)$$

$$S_n = E \{ NN^* \}. \quad (36)$$

The minimization of the filter  $W$  yields to the solution

$$W = \frac{1}{H} \frac{|H|^2}{|H|^2 + \frac{S_n}{S_f}}, \quad (37)$$

where the ratio  $S_n/S_f$  is equivalent to the inverse of the signal-to-noise ratio (SNR).

We can see that the Standard Wiener filter  $W$  has two terms. When the second term is equal to one, that is, when the image is not degraded by noise ( $S_n$  equal to zero), the Standard Wiener filter is equivalent to the Inverse filter. On the other hand, at those frequencies where the noise power is high (equivalent to a low SNR), the second term is lower than one, so the Standard Wiener filter is able to reduce the effect of the noise amplification suffered in the Inverse filter case.

Fig. 8(a) shows the image degraded by blur and noise (PSNR = 21.5567 dB), Fig. 8(b) is the restored image by inverse filtering (PSNR = -9.6997 dB), and Fig. 8(c) shows a restored image by the Standard Wiener filter (PSNR = 24.1468 dB). As said before, the resulting image after applying the Standard Wiener filter is not affected by the noise amplification effect.

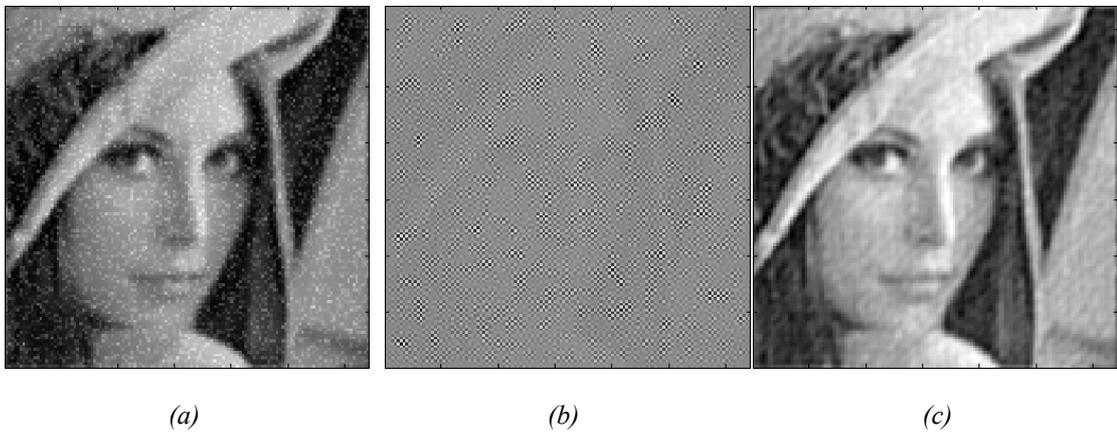


Fig. 8 (a) Image degraded by noise and blur; (b) restored image using the Inverse filter; (c) restored image using the Standard Wiener filter.

The degradation function  $H$  has to be estimated in order to restore the image. There are three different methods to obtain a good estimation of the blurring system: estimation by observation, estimation by experimentation, and estimation by modeling. These methods are discussed with more detail in [17].

In addition, it is clear in the expression of the Standard Wiener filter in (37) that it is necessary to know the power spectra of the noise and the original image. This is *a priori* information which is usually not available. Thus, we need to estimate the power spectra of the noise and the original image.

The noise spectrum is a constant, so it is simple to estimate it. However, the power spectrum of the original image is more difficult to estimate. Usually it is estimated by the frequency spectrum of the degraded image ( $S_g$ ). A more simple approach is to estimate the ratio  $S_n/S_f$  by a constant  $K$

$$W = \frac{1}{H} \frac{|H|^2}{|H|^2 + K}. \quad (38)$$

The value of the parameter  $K$  used must be adapted to the noise that added to the image. The optimal value of  $K$  is the one that maximizes the Peak Signal-to-Noise Ratio (PSNR) between the original image and the image restored with the Wiener filter. Shown in Fig. 9 is a graph where the PSNR is computed for different values of  $K$  as well as different noise variances.

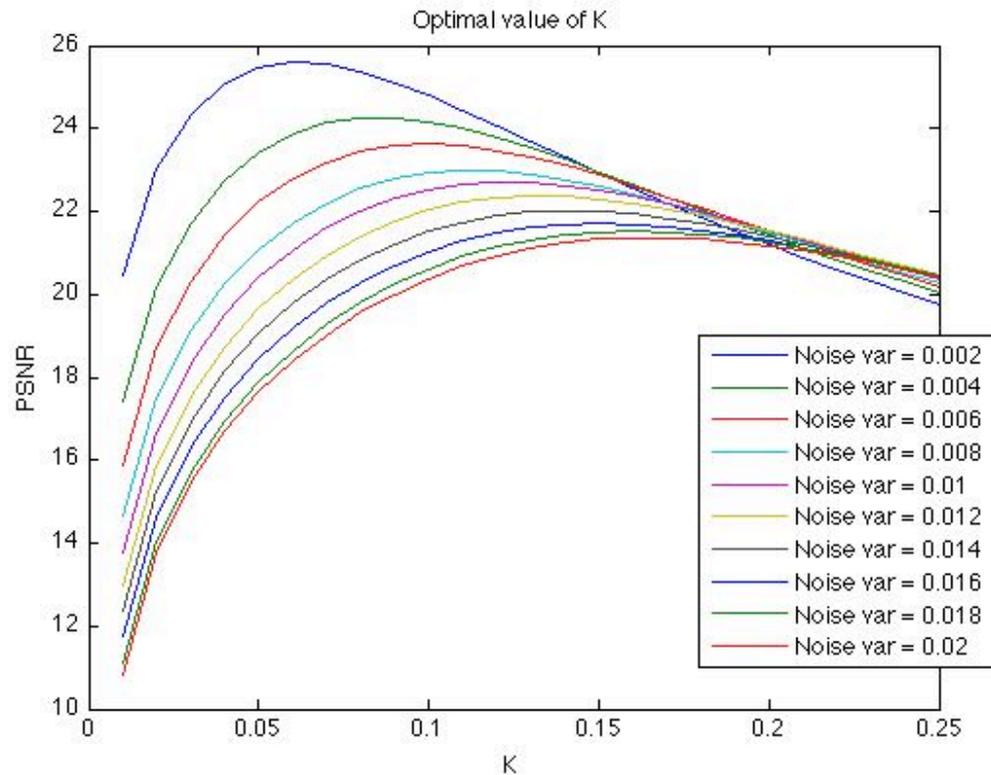
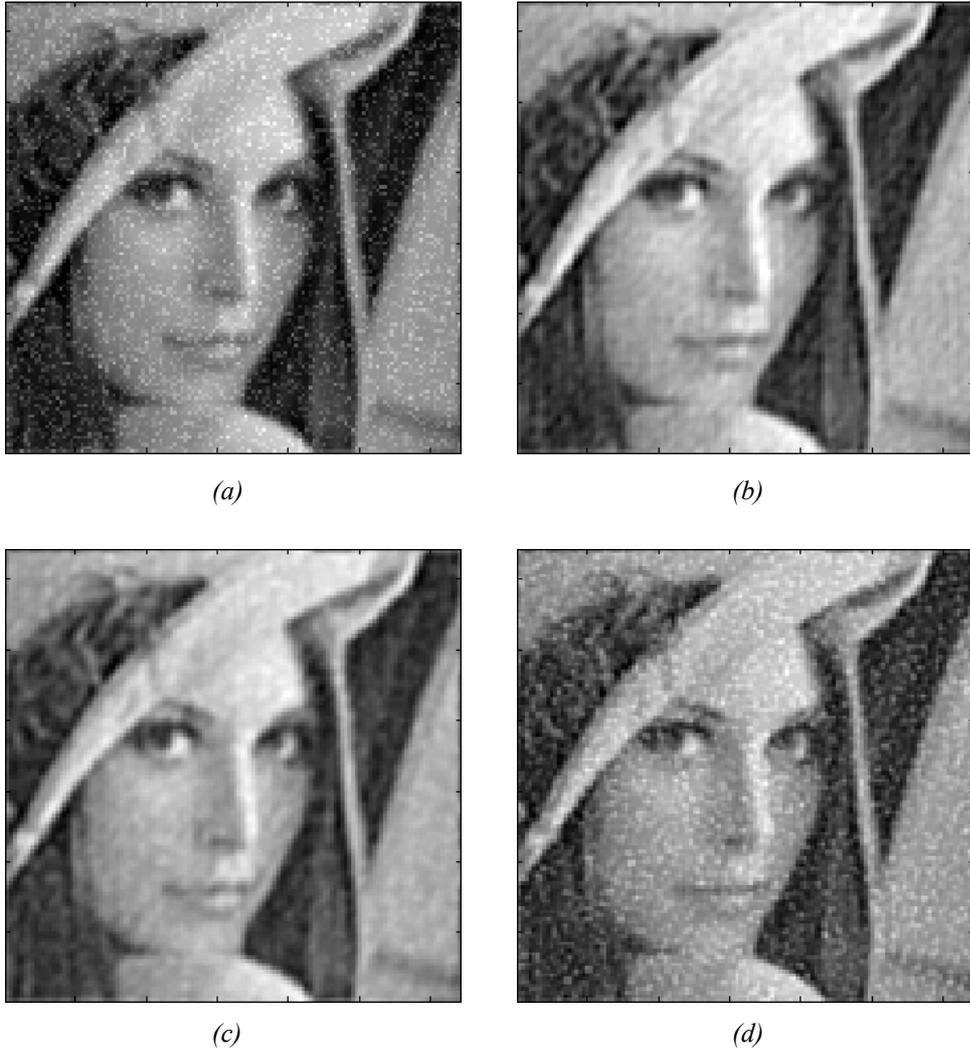


Fig. 9 Relation between PSNR and  $K$  for different values of the noise variance

Fig. 10(a) shows an image degraded by blur and noise with a PSNR = 21.5567 dB. Fig. 10(b) to (d) show the difference between the use of the spectra from the original image (PSNR = 24.1468 dB), the degraded image (PSNR = 23.2888 dB), and a constant (PSNR = 22.0625 dB for  $K = 0.1$ ) respectively in the design of the Standard Wiener filter. The deterioration of the PSNR is clear in both images Fig. 10(c) and (d) where the PSNR is lower than in Fig. 10(b).



*Fig. 10 (a) Image degraded by blur and noise; (b) to (d) restored images with the Standard Wiener filter using the spectrum from the original image, from the degraded image, and a constant.*

### 2.3.3 Parametric Wiener Filter

The Parametric Wiener filter [11], [12], [13], [14] is based on the Standard Wiener filter but with the addition of a parameter *gamma* ( $\gamma$ ) so that the expression of the filter is as follows

$$W = \frac{1}{H} \frac{|H|^2}{|H|^2 + \gamma \frac{S_n}{S_f}}. \quad (39)$$

When either the noise spectrum ( $S_n$ ) or the original image spectrum ( $S_f$ ) are unknown, this parameter tries to adapt the estimated ratio  $S_n/S_f$  so that it is closer to the original ratio. When  $\gamma$  is equal to one, the expression above reduces to the Standard Wiener filter. It can be also seen as an smoothing parameter, that is, higher values of  $\gamma$  mean smoother output images.

Shown in Fig. 11(a) the restored image after applying the Parametric Wiener filter for  $\gamma = 0.5$  with a PSNR = 23.5535 dB, in Fig. 11(b) for  $\gamma = 1$  with a PSNR = 23.773 dB, and in Fig. 11(c) for  $\gamma = 7$  with a PSNR = 23.9496 dB.

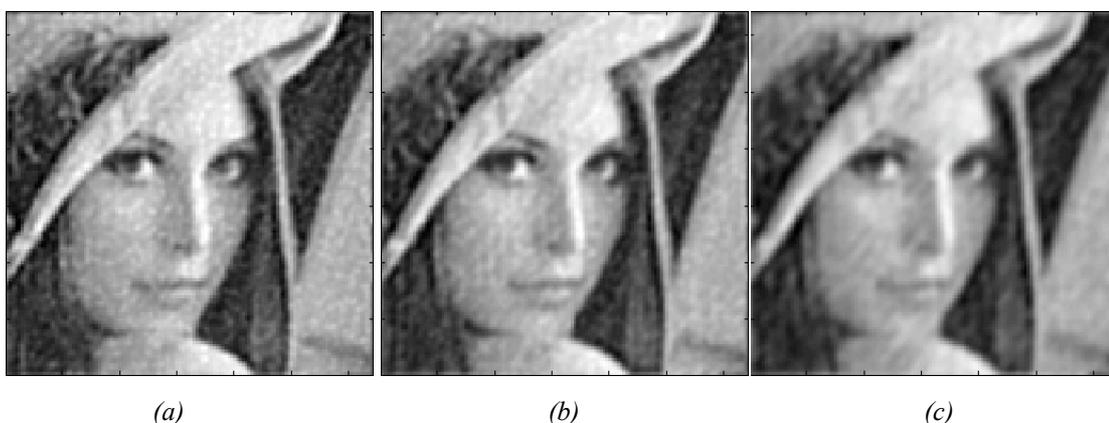


Fig. 11 (a) Restored image using the Parametric Wiener filter with  $\gamma = 0.5$ ; (b) with  $\gamma = 1$ ; (c) with  $\gamma = 7$ .

In addition, in images where the noise power is high and the spectrum of the original image is not available, the Parametric Wiener filter with values of  $\gamma$  higher than one ( $\gamma > 1$ ) yields better results than the Standard Wiener filter. Fig. 12 shows a PSNR-Gamma graph for different values of the noise standard deviation (equivalent to the Gaussian noise power), which demonstrates the statement above.

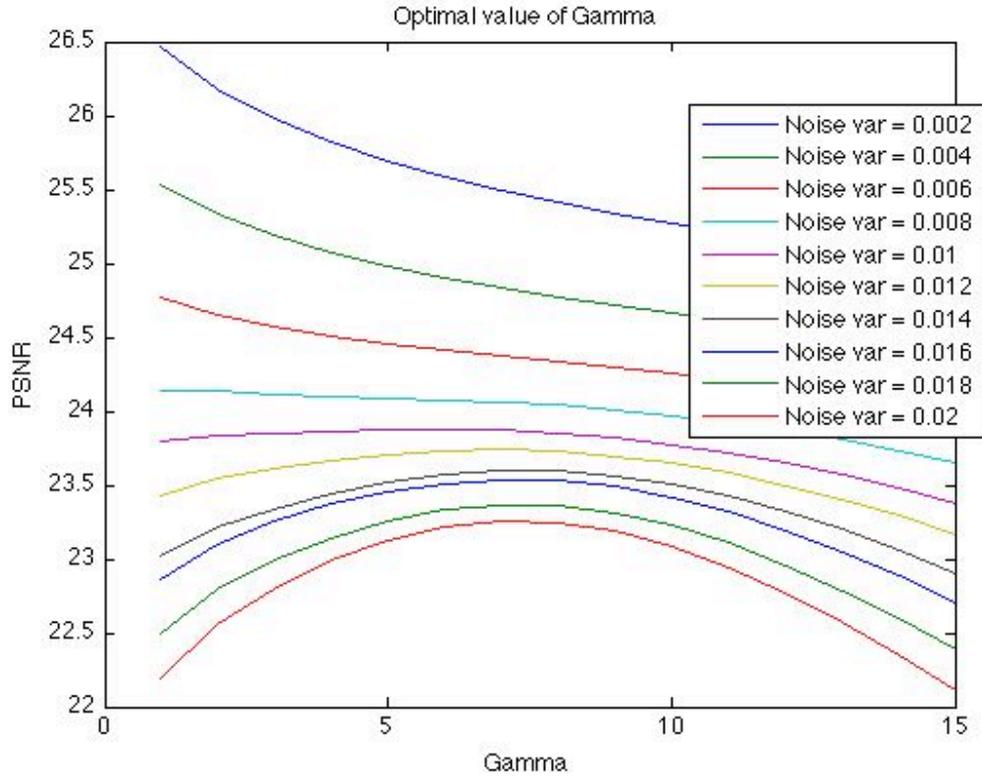


Fig. 12 Relation between PSNR and gamma for different values of the noise variance.

The optimal value of  $gamma$  is computed based on the iterative procedure used in [8], which is described briefly next:

- 1) specify an initial value of  $gamma$ .
- 2) use the Winer filter to obtain the restored image.
- 3) compute the residual vector  $\mathbf{r} = \mathbf{g} - \mathbf{H}\tilde{\mathbf{f}}$ .
- 4) stop if  $\|\mathbf{r}\|^2 = \|\mathbf{n}\|^2 \pm a$ , where  $\mathbf{n}$  is the noise vector and  $a$  corresponds to the desired accuracy. Otherwise, increase the value of  $gamma$  if  $\|\mathbf{r}\|^2 < \|\mathbf{n}\|^2 - a$  and decrease  $gamma$  if  $\|\mathbf{r}\|^2 > \|\mathbf{n}\|^2 + a$  and return to step 2.

## CHAPTER 3

### IMAGE RESTORATION WITH CONTENT ADAPTIVE MESH MODELING FILTER

#### 3.1 Basic Concepts

Digital images are divided or sampled by smaller units or elements called pixels (picture elements). Pixels are usually squared and they form a two-dimensional uniform grid which represents the image. A higher number of pixels used in a digital image usually means a better representation of the original object. The value of the pixel provides the brightness intensity of that sample of the image.

Most of the classical image restoration algorithms, such as Inverse filter and Wiener filter, are pixel-based, that is, they are designed to find the best approximation of the pixel values in order to recover the original non-degraded image. However, the next sections focus on a different approach, a non-uniform mesh model.

##### 3.1.1 Mesh Model Description

Mesh modeling a digital image means dividing the image into non-overlapping polygonal units, called mesh elements that form a grid. Unlike in pixel based images, mesh modeling grids are not necessary uniform, i.e., the mesh elements may have different shape and different size. In mesh modeling, the image is sampled by points, called nodes, which correspond to the vertices of the mesh elements. The intensity of each element is computed by interpolation between its nodes (vertices) values. Next we present the mathematical equations of the mesh model.

Let  $f(x)$  be an image function defined over a domain  $D$ , which in this study is two-dimensional. This domain is divided into  $M$  non-overlapping mesh elements  $D_m$ , with  $m = 1, 2, \dots, M$ . Usually the mesh elements are polygonal. In this study we used triangles. The advantage of triangular elements over, for example, quadrangular elements, is that it is easier to represent an image with no overlap between elements and no gaps between them.

Thus, the image function can be represented over each mesh element as follows:

$$f(x) = \sum_{k=1}^K f(x_k) \varphi_{m,k}(x) + e(x), \quad \text{for } x \in D_m, \quad (40)$$

where  $K$  is the number of nodes defining  $D_m$ ,  $x_k$  is the  $n$ th node belonging to  $D_m$ ,  $\varphi_{m,k}$  is the interpolation basis function associated with this node, and  $e(x)$  is the interpolation error.

As said before, triangular elements are used in this study, that is each mesh element is defined by three nodes ( $K = 3$ ). Each interpolation basis function  $\varphi_{m,k}$  has support only over those mesh elements  $D_m$  attached to the node  $x_k$ . For simplicity, we use the following notation in the next sections:

$$f(x) = \sum_{n=1}^N f(x_n) \phi_n(x) + e(x), \quad \text{for } x \in D, \quad (41)$$

where  $N$  is the total number of nodes all over the image and  $\phi_n(x)$  is the sum of all the interpolation basis functions  $\varphi_{m,k}$  associated to the  $n$ th node. Note that in this equation  $x$  is an element of  $D$ .

Some important image processing applications of mesh modeling are image reconstruction, image compression [15], medical image analysis, and motion tracking and segmentation [16], [17]. In this study we will focus on the use of a Content Adaptive Mesh Modeling (CAMM) [18], [19], [20] as an image restoration technique. The main difference between CAMM and other image reconstruction algorithms is that the image is sampled non-uniformly, placing more nodes (samples) in areas where there is more image content or detail. This provides some advantages in comparison to other methods. For example, the use of CAMM allows the use of a lower amount of mesh nodes than pixels for a similar image representation. This means a compression of the image and, at the same time, faster computation. In addition, CAMM yields better adapted smoothness to the different image areas, while uniform sampling methods yield, in general, an smoother version of the image since the filter is applied to the whole image.

As we will see in future sections, CAMM is able to achieve a very accurate interpolation of the image, so the error due to this process  $e(x)$  is negligible. Thus, the image function can be approximated as

$$\tilde{f}(x) = \sum_{n=1}^N f(x_n)\phi_n(x), \quad \text{for } x \in D, \quad (42)$$

Finally, the CAMM algorithm also allows to restore motion image sequences, since the mesh elements can be deformed in order to be adapted to the image content.

### 3.2 Algorithm

The proposed mesh model of an image is based, as said before, on placing the mesh nodes in those areas with a higher image detail (high frequency areas). That is, it has to place smaller mesh elements (i.e., more nodes) in regions where the second directional derivative of the image is larger and bigger mesh elements (i.e., less nodes) where the second directional derivative is smaller. The proof can be found in Appendix A.

The algorithm used for the content adaptive mesh modeling of an image consists in four steps. The first step is the feature map extraction from the image. This feature map indicates the areas where the second directional derivative of the image is larger, i.e., it shows the high frequency regions of the image. In the second step, the Floyd-Steinberg error-diffusion algorithm [21] is used in order to place the mesh nodes according to the feature map obtained in the previous step. In the third step, the Delaunay triangulation algorithm [16] is applied to create the mesh elements from the nodes placed in the second step by connecting them forming triangular elements. Finally the image intensity is defined for each mesh element by interpolation between its corresponding vertices (mesh nodes) values. In the next section we explain the implementation of this algorithm.

### 3.2.1 Feature Map Extraction

The feature map of the image  $\sigma(x,y)$  is computed by the second directional derivative and it is defined as follows:

$$\sigma(x,y) = \left( \frac{G(x,y)}{A} \right)^\gamma, \quad (43)$$

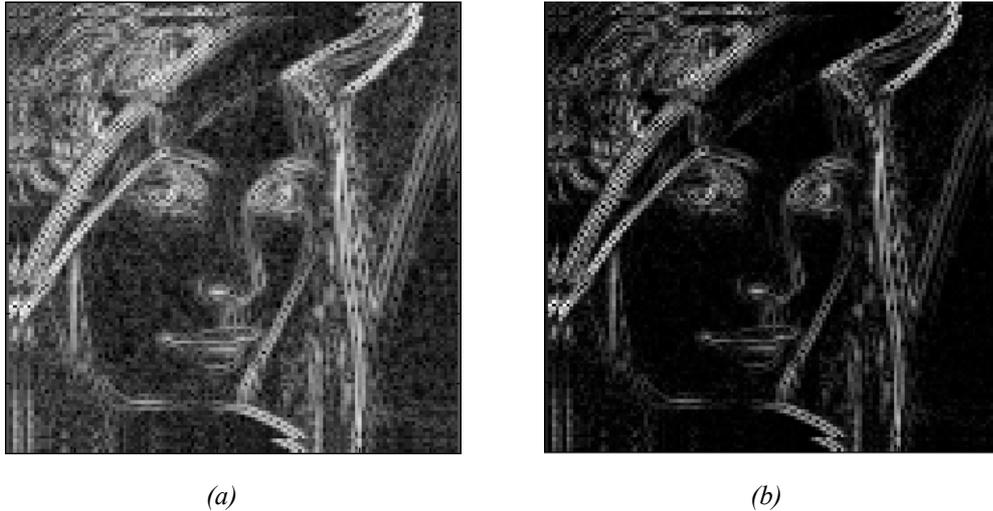
where  $G(x,y)$  indicates the largest magnitude of the second partial derivative of the image  $f(x,y)$ ,  $A$  is the maximum value of  $G(x,y)$  used to normalize the feature map with values between 0 and 1, finally,  $\gamma$  is a constant which corresponds to the sensitivity of the nodes placement to the image edges. High values of  $\gamma$  yield to a feature map with more nodes placed in high frequency regions of the image, such as edges.

$G(x,y)$  is computed by finding the maximum eigenvalue of the image Hessian matrix for each pixel, that is:

$$G(x,y) = \max \{ |\lambda_1(x,y)|, |\lambda_2(x,y)| \}, \quad (44)$$

where  $\lambda_1(x,y)$  and  $\lambda_2(x,y)$  are the eigenvalues of the Hessian matrix of the image  $f(x,y)$ . The proof of this equation can be found in Appendix B.

The extraction of the feature map of an image yields to an image where the high frequency regions (edges) have higher values (close to 1) and the low frequency areas (smooth areas) have lower values (close to 0). Fig. 13(a) and (b) show the feature map of an image with the sensitivity set to  $\gamma = 0.5$  and  $\gamma = 1$ , respectively.



*Fig. 13 Feature map for different sensitivities (a)  $\gamma = 0.5$ ; (b)  $\gamma = 1$ .*

### 3.2.2 Nodes Placement

The Floyd-Steinberg error-diffusion algorithm [21] is a dithering technique used in digital images in order to reduce the effect of color quantization of an image. The algorithm approximates non-available colors in the color palette by diffusing colored pixels so that the human eye perceives the mixture of the colored pixels. For grayscale images that are converted to black and white format, the quantization is based on a threshold in order to decide the value of the pixel (0 or 1). The resulting image after applying the Floyd-Steinberg algorithm can present some graininess.

Fig. 14 shows a black and white image corresponding to a section of “Lena”. The right side of the image corresponds to the original grayscale image and the left side is the result of applying the Floyd-Steinberg dithering algorithm. In the enlarged section on the right we can see the graininess, in form of black and white dots, present in the image.

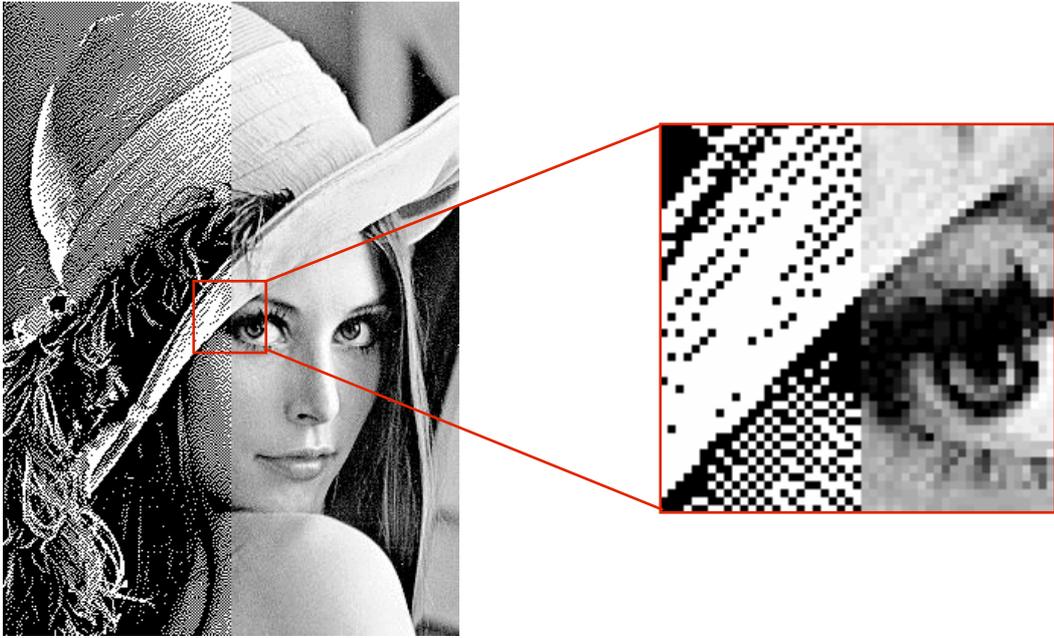


Fig. 14 Section of “Lena” with the left side of the image corresponding to the result of the Floyd-Steinberg dithering algorithm.

Image: Connexions (<http://cnx.org/>)

In this study, the Floyd-Steinberg algorithm is used to place the mesh nodes according to the spatial density defined by the feature map of the image. First, the algorithm scans every pixel of the feature map  $\sigma(i,j)$  and compares its value to a threshold  $q$ , which is defined later in this section, in order to obtain the function

$$b(i,j) = \begin{cases} 1, & \text{if } \sigma(i,j) \geq q \\ 0, & \text{otherwise,} \end{cases} \quad (45)$$

which indicates if a node must be placed in that pixel. If the value of  $b(i,j)$  is one, a mesh node is placed at the pixel  $(i,j)$ .

Then the error of quantization  $e(i,j)$  is computed for every pixel

$$e(i, j) = 2qb(i, j) - \sigma(i, j). \quad (46)$$

Next, the error of each pixel is diffused to its four casual neighbors by scanning the image line by line, from left to right for the odd lines and from right to left for the even lines. This method is called serpentine raster order and the advantage with respect to the traditional left-to-right raster order is that it yields to a better propagation of the error diffusion. The value of the feature map pixels are recalculated as follows:

$$\sigma(i, j+1) = \sigma(i, j+1) + w_1 e(i, j), \quad (47)$$

$$\sigma(i+1, j-1) = \sigma(i+1, j-1) + w_2 e(i, j), \quad (48)$$

$$\sigma(i+1, j) = \sigma(i+1, j) + w_3 e(i, j), \quad (49)$$

$$\sigma(i+1, j+1) = \sigma(i+1, j+1) + w_4 e(i, j), \quad (50)$$

where the weights  $w_1$ ,  $w_2$ ,  $w_3$  and  $w_4$  are the same used in the Floyd-Steinberg algorithm, that is

$$w_1 = \frac{7}{16}, w_2 = \frac{3}{16}, w_3 = \frac{5}{16}, w_4 = \frac{1}{16}. \quad (51)$$

However, the weights from the Floyd-Steinberg algorithm must be readjusted when an image boundary is reached in order to avoid error leakage. If the pixel is on the right boundary, the weights  $w_1$  and  $w_4$  must be zero and the rest of the weights ( $w_2$  and  $w_3$ ) must add to 1. If it is on the bottom boundary the weights  $w_2$ ,  $w_3$ , and  $w_4$  must be zero while  $w_1$  must take the value of 1. Finally, if the pixel is on the left boundary, only the weight  $w_2$  must be zero and the remaining weights ( $w_1$ ,  $w_3$ , and  $w_4$ ) must be recalculated to add to 1.

In this study, the process of the error diffusion is carried out the opposite way, that is, every pixel receives the weighted part of the error from its neighbors. Fig. 15(a) shows the diffusion from the four neighbors to a pixel when it is located in the middle of a picture, so no readjustment of the weights is necessary. Fig. 15(b) corresponds to a pixel on the right boundary of the feature map of the image. Fig. 15(c) and (d) show the cases of a pixel located on the bottom boundary and the left boundary, respectively.

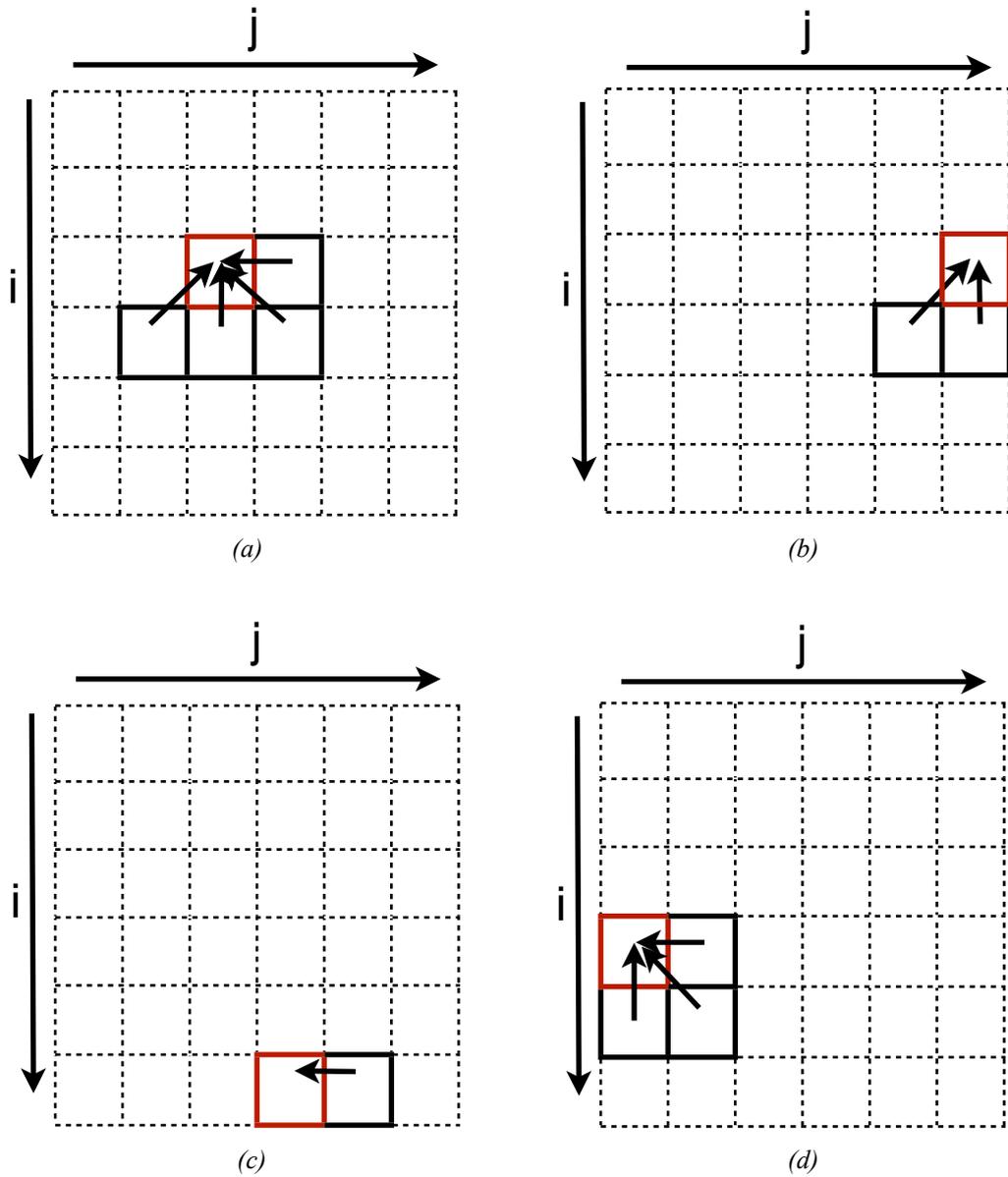


Fig. 15 (a) Error diffusion to a pixel when it is located in the center of the image; (b) when it is at the right boundary; (c) when it is at the bottom boundary; and (d) when it is located at the right boundary.

The error diffusion is equivalent to apply a filter in the spatial domain to the feature map of the image, since it processes every pixel and their neighbors the same way all over the image. It can be modeled as a squared filter  $\mathbf{H}$  of dimensions 4 by 4

$$\mathbf{H} = \frac{1}{16} \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 7 \\ 3 & 5 & 1 \end{pmatrix}. \quad (52)$$

The threshold  $q$  used to compute the indicator function  $b(i,j)$  from the feature map of the image  $\sigma(i,j)$  is inversely proportional to the number of nodes ( $N$ ) used in the mesh structure. Assuming that the image is squared, i.e. the number of pixels in both dimensions is  $M$ , the relation between the number of nodes  $N$  and the threshold  $q$  is defined by

$$N = \frac{1}{2q} \left[ \sum_{i=1}^M \sum_{j=1}^M \sigma(i,j) - e(M,M) \right], \quad (53)$$

where  $e(M,M)$  is the quantization error corresponding to the last pixel.

If we consider that  $e(M,M)$  is much smaller than  $\sum_{i=1}^M \sum_{j=1}^M \sigma(i,j)$ , the expression above

can be approximated by

$$N \approx \frac{1}{2q} \sum_{i=1}^M \sum_{j=1}^M \sigma(i,j) \quad (54)$$

and

$$q \approx \frac{1}{2N} \sum_{i=1}^M \sum_{j=1}^M \sigma(i,j). \quad (55)$$

It is clear that the threshold value can be calculated after the feature map extraction and, what is more advantageous, before the Floyd-Steinberg error diffusion for a pre-defined number of mesh nodes  $N$ . In Fig. 16(a) and (b) we can see the node placement for two different values of the number of nodes used as well as their corresponding threshold.

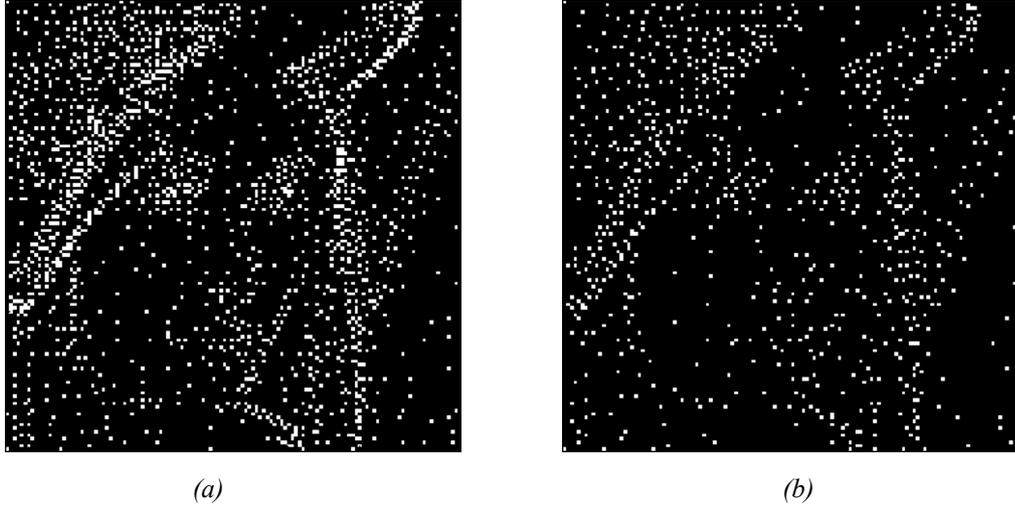


Fig. 16 Images of the nodes placement in (a) using  $N = 1853$  nodes and a threshold  $q = 0.52585$  and in (b) using  $N = 1047$  and  $q = 1.0317$ .

In this study, a desired ratio between the number of nodes and the number of pixels is fixed prior to the computation of the feature map.

$$\text{Ratio} = \frac{\# \text{ pixels of the image}}{\# \text{ number of nodes}}. \quad (56)$$

### 3.2.3 Mesh Elements Structure With Delaunay Triangulation

The formation of the mesh elements is carried out by the Delaunay triangulation algorithm [16]. Given a set of points (mesh nodes) the Delaunay triangulation algorithm connect them forming a triangulation with the condition that no point can be inside the circumcircle of any triangle. If four points are on the same circle (co-circular), the Delaunay triangulation is not unique and yields to two different triangulations, both satisfying the condition mentioned before. Another property of the Delaunay triangulation is that each vertex (node) belongs, on average, to six triangles (mesh elements).

Fig. 17(a) shows an example of an incorrect Delaunay triangulation since there is one node inside each circumcircle. On the other hand, Fig. 17(b) shows a correct triangulation with no point inside any circumcircle.

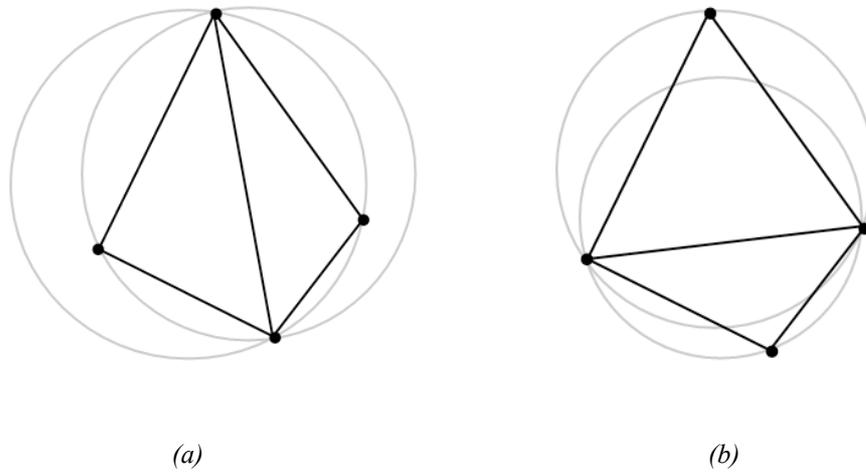


Fig. 17 (a) Incorrect Delaunay triangulation; (b) correct triangulation.

Fig. 18 shows a mesh structure after applying the Delaunay triangulation to the mesh nodes of a section of the image “Lena”. The number of nodes used is  $N = 1853$  (Ratio = 10 pixels/node).

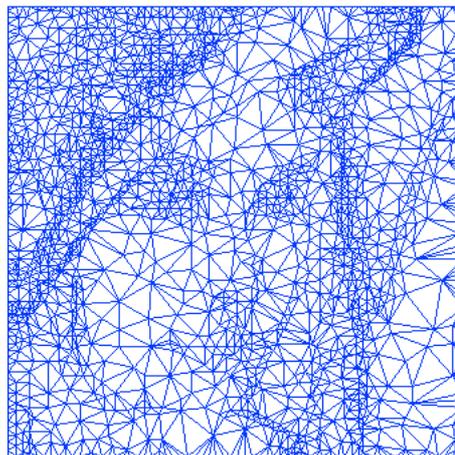


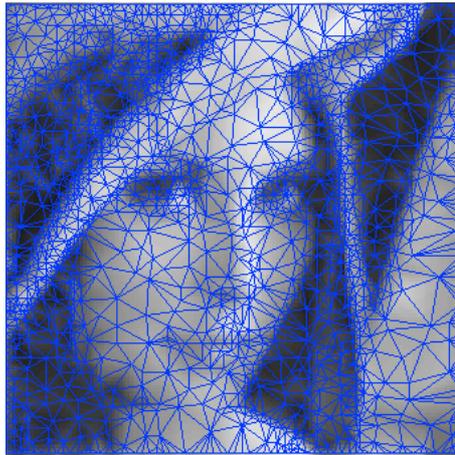
Fig. 18 Mesh structure of a 128x128 section of “Lena” using  $N = 1853$  nodes (Ratio = 10 pixels/node).

### 3.2.4 Direct Interpolation

Considering the mesh representation model described previously in (42), the resulting restored image after applying the CAMM filter can also be written as

$$\tilde{f}(x) = \sum_{n=1}^N f(x_n) \phi_n(x), \quad (57)$$

where  $N$  is the total number of mesh nodes over the image,  $f(x_n)$  is the nodal value, and  $\phi_n(x)$  is the interpolation function corresponding to the node  $x_n$ . Fig. 19 shows how the mesh elements form the restored image after the interpolation of its nodes using the mesh structure from Fig. 19.



*Fig. 19 Image restored by linear interpolation using the mesh structure from Fig. 18*

As said before, every node contributes, on average, to the brightness intensity of six mesh elements. Fig. 20 corresponds to an enlarged part of the mesh structure used before, where we can see the contribution of a mesh node (marked in red) to the image.

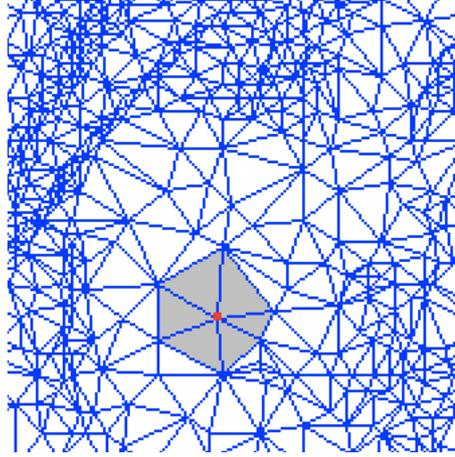


Fig. 20 Contribution of a node to the intensity of the associated mesh elements.

If the original image is available, the restored image  $\tilde{f}(x)$  has the same value as the original image  $f(x)$  at the mesh nodes, that is

$$\tilde{f}(x_n) = f(x_n). \quad (58)$$

Unfortunately, this is an ideal case and usually there is no *a priori* knowledge about the original image, thus an estimation of the image must be used instead. Sometimes this estimated image is the same degraded image. Then the equality above becomes

$$\tilde{f}(x_n) = g(x_n), \quad (59)$$

which yields to a bad approximation of the original image. The difference between the use of the original image and the use of the degraded image is demonstrated in Fig. 21. The mesh structure from the original image (Fig. 18) was used. Fig. 21(a) corresponds to the interpolated image when the nodal values are taken from the original image (PSNR = 27.2341 dB), while Fig. 21(b) is the interpolated image when the values from the degraded image are used (PSNR = 21.3665 dB). The number of nodes used is  $N = 1853$ . The solution for this problem is discussed in the next section.



Fig. 21 Restored images using the mesh structure from Fig. 18 and linear interpolation between the nodal values from the original image in (a); in (b) nodal values are from the degraded image.

### 3.2.5 Least Squares Fitting

Since the original image is not known, the direct interpolation of the nodal values is not the most accurate approximation. In addition, the use of the intensity values from the original image at each node is not the best approximation of the image, since it only ensures the optimal intensity at the mesh nodes, i.e.  $\tilde{f}(x_n) = f(x_n)$ , not to the overall mesh. Thus, the nodal values can be optimized or better approximated using a least squares algorithm [15], [18] which is described next.

The expression used in (57) can be rewritten as

$$\tilde{f}(x) = \sum_{n=1}^N f_n \phi_n(x), \quad (60)$$

where  $f_n$  are the nodal values. If we rewrite this expression using the discrete form it becomes

$$\tilde{\mathbf{f}} = \mathbf{\Phi} \mathbf{n}, \quad (61)$$

where  $\tilde{\mathbf{f}}$  is the column vector formed by all the points ( $M$  pixels) of the restored image,  $\mathbf{n}$  is the column vector formed by all the nodal values ( $N$  mesh nodes), and  $\mathbf{\Phi}$  is the matrix of the interpolation function for each node all over the image, that is

$$\tilde{\mathbf{f}} = \begin{bmatrix} \tilde{f}(0) \\ \tilde{f}(1) \\ \vdots \\ \tilde{f}(M-1) \end{bmatrix}_{M \times 1} \quad \mathbf{n} = \begin{bmatrix} n(0) \\ n(1) \\ \vdots \\ n(N-1) \end{bmatrix}_{N \times 1} \quad \Phi = \begin{pmatrix} \phi_1(0) & \phi_2(0) & \cdots & \phi_N(0) \\ \phi_1(1) & \phi_2(1) & \cdots & \phi_N(1) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_1(M-1) & \phi_2(M-1) & \cdots & \phi_N(M-1) \end{pmatrix}_{M \times N}. \quad (62)$$

The nodal values can be optimized by minimizing the square error between the original image and the interpolated image from the mesh nodes, that is

$$\mathbf{n} = \arg \min_{\mathbf{n}} \|\mathbf{f} - \Phi \mathbf{n}\|^2. \quad (63)$$

The solution of this equation is computed iteratively as follows:

$$\mathbf{n}^{i+1} = \mathbf{n}^i - \alpha \cdot \nabla J(\mathbf{n}), \quad (64)$$

where  $\mathbf{n}^i$  and  $\mathbf{n}^{i+1}$  are the nodal values for the current and the next iterations respectively,  $\nabla J(\mathbf{n})$  is the gradient of the mean square error

$$\nabla J(\mathbf{n}) = -\Phi^T \mathbf{f} + \Phi \Phi^T \mathbf{n}, \quad (65)$$

and  $\alpha$  is the step value. The derivations of this equation can be found in Appendix C.

Fig. 22(a) corresponds to the resulting image after applying the least squares fitting (PSNR = 29.2533 dB) and Fig. 22(b) is the restored image using direct linear interpolation between the nodal values from the original image as in Fig. 21(a) (PSNR = 27.2341 dB). There is an improvement of more than 2 dB, which becomes even higher when a higher ratio (less nodes) is used. This is demonstrated in Table 2

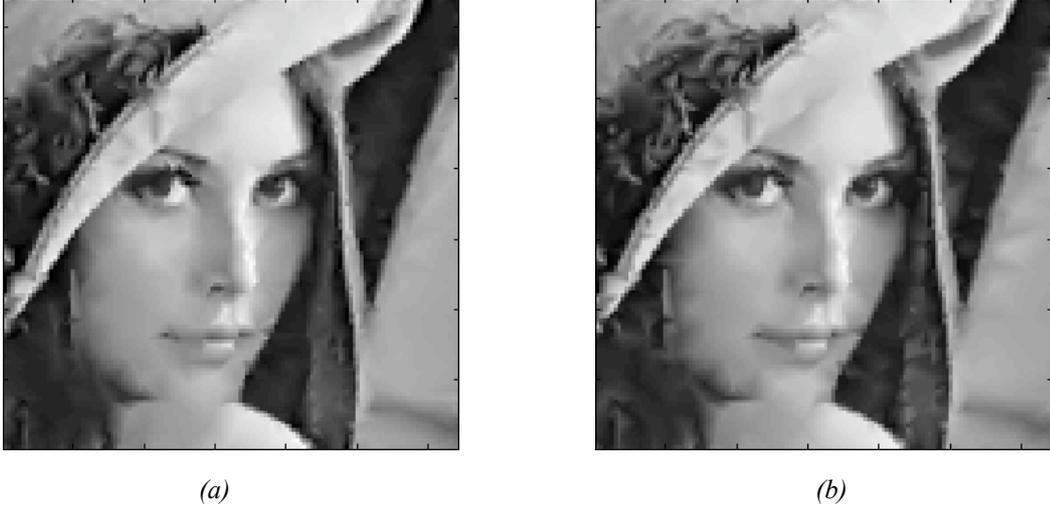


Fig. 22 Restored images using least squares fitting in (a) (PSNR = 29.2533 dB); and linear interpolation in (b) (PSNR = 27.2341 dB). The mesh structure from Fig. 18 was used in both cases.

Ratio	Interpolation (PSNR)	Least squares fitting (PSNR)	Improvement
5	31.3127 dB	33.1037 dB	1.791 dB
10	27.2341 dB	29.2533 dB	2.0192 dB
20	23.3584 dB	25.6828 dB	2.3244 dB

Table 2 Comparison between the use of linear interpolation and least squares fitting for different ratios (pixels/node)

However, this is not a feasible solution since the original image  $\mathbf{f}$  is unknown and the values at the mesh nodes are not available. A more realistic approach is the use of the degraded image  $\mathbf{g}$  instead of the original one. In this case, the interpolated image  $\Phi\mathbf{n}$  must be filtered by the blurring filter  $\mathbf{H}$  as well

$$\mathbf{n} = \arg \min_{\mathbf{n}} \|\mathbf{g} - \mathbf{H}\Phi\mathbf{n}\|^2. \quad (66)$$

The new nodal values are computed iteratively as before, however, in this case the gradient of the mean square error becomes

$$\nabla J(\mathbf{n}) = -(\mathbf{H}\Phi)^T \mathbf{g} + (\mathbf{H}\Phi)(\mathbf{H}\Phi)^T \mathbf{n}. \quad (67)$$

Fig. 23(a) corresponds to the restored image after applying the least squares fitting using the degraded image (PSNR = 23.9847 dB). Fig. 23(b) shows the estimated image when direct linear interpolation between the values from the degraded image is used (PSNR = 21.6092 dB). It is clear the improvement achieved by using least squares fitting to define the intensity of the mesh nodes. Table 3 shows the improvement of least squares fitting compared to linear interpolation for different ratios.

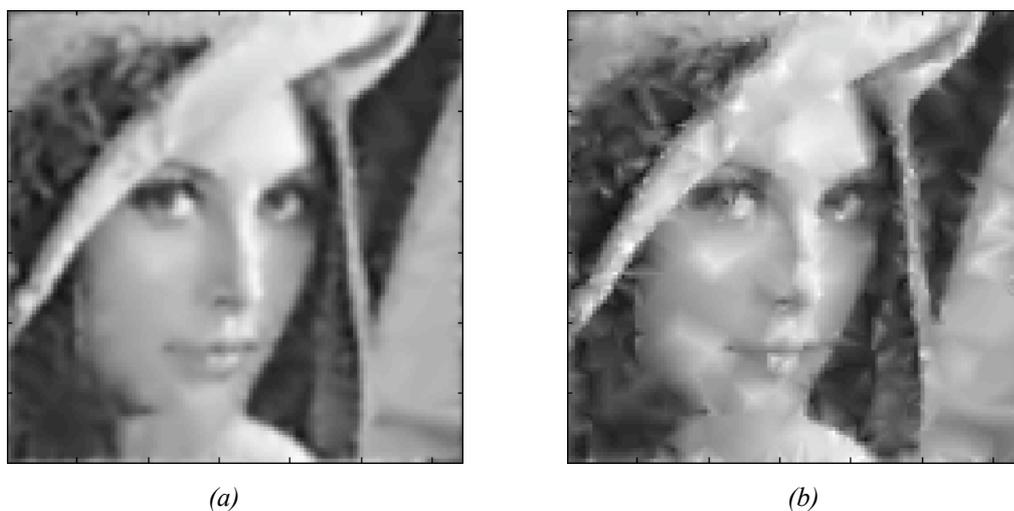


Fig. 23 (a) Restored image using least squares fitting when the original image is not available; (b) restored image by direct interpolation between the nodal values taken from the degraded image.

Ratio	Interpolation (PSNR)	Least squares fitting (PSNR)	Improvement
5	21.8814 dB	23.8121 dB	1,9307 dB
10	21.6092 dB	23.9847 dB	2,3755 dB
20	20.3427 dB	22.9726 dB	2,6299 dB

Table 3 Comparison between linear interpolation and least squares fitting when the degraded image is used.

### 3.3 Implementation Issues of CAMM

Most of the image restoration techniques have some implementation issues and some of the theoretical equations are unfeasible in practice. In this section we are presenting some of this issues that affect to the design of the Content Adaptive Mesh Model, as well as some solutions which try to solve them.

#### 3.3.1 Feature Map

Regarding the feature map extraction of the image, there are some issues which make it impossible or unfeasible to obtain the exactly feature map. As said before, the computation of the feature map is based on finding the maximum eigenvalue of the Hessian matrix of the image. The Hessian matrix  $\mathbf{H}(f)$  of a function of  $N$  variables is defined as the matrix of the second-order partial derivatives, that is

$$\mathbf{H}(f) = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_N} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_N \partial x_1} & \frac{\partial^2 f}{\partial x_N \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_N^2} \end{pmatrix}. \quad (68)$$

However, the function of a digital image is not continuous, so the partial derivatives cannot be computed, they must be estimated. If we consider that an image  $f(x,y)$  is a two-dimensional discrete function, we can estimate its partial derivatives by the values of a pixel  $f(i,j)$  and its neighbors

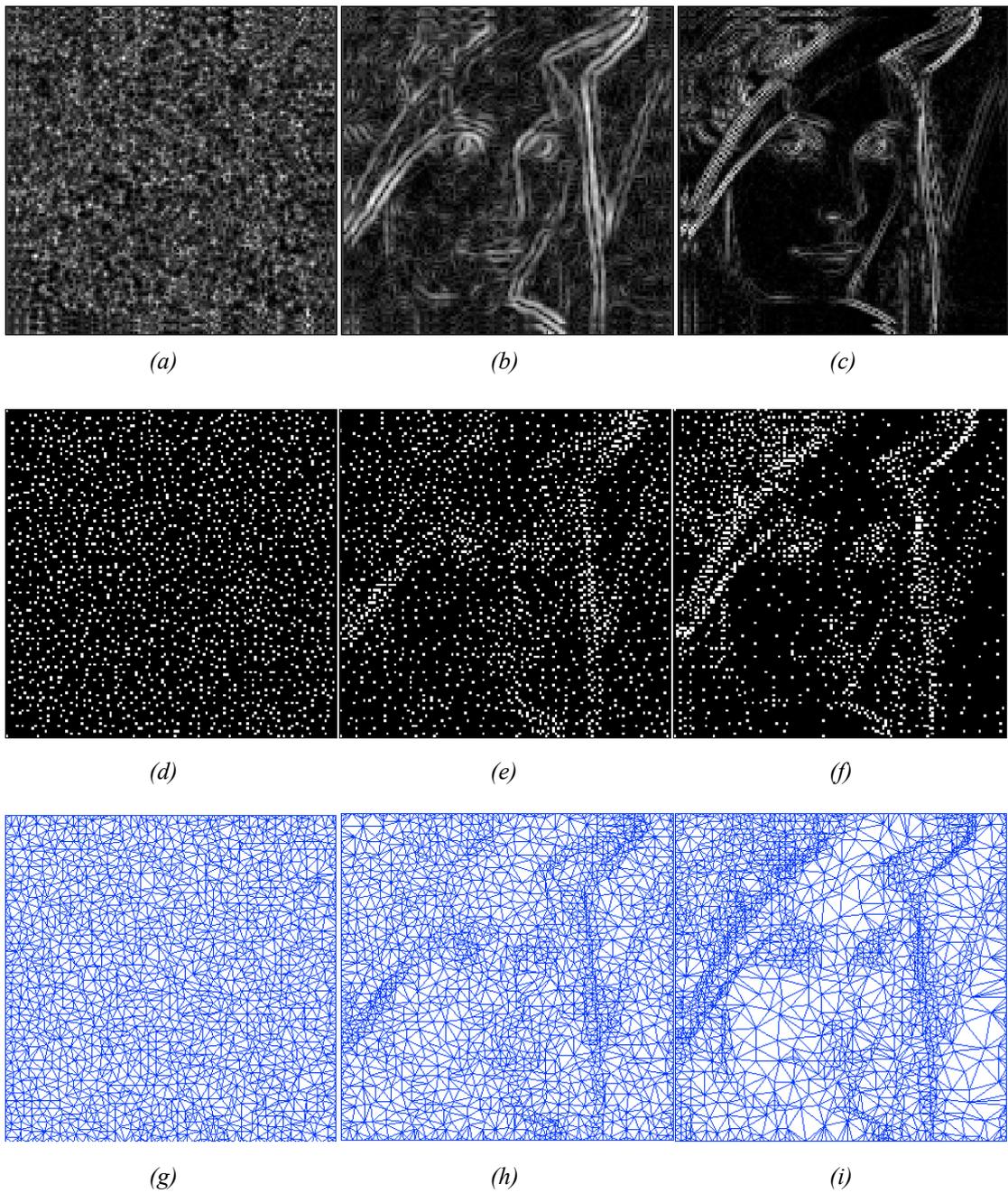
$$\frac{\partial^2 f}{\partial x^2} \approx f(i+1,j) - 2f(i,j) + f(i-1,j), \quad (69)$$

$$\frac{\partial^2 f}{\partial y^2} \approx f(i,j+1) - 2f(i,j) + f(i,j-1), \quad (70)$$

$$\frac{\partial^2 f}{\partial x \partial y} \approx \frac{1}{4} [f(i+1,j+1) - f(i+1,j-1) - f(i-1,j+1) + f(i-1,j-1)]. \quad (71)$$

Another issue related to the feature map is due to the fact that the original image  $f(x,y)$  is unknown, so it must be estimated. In this study we used a Gaussian low-pass filter in order to reduce the noise of the degraded image so it can be used in order to compute de feature map. This issue affects not only to the feature map, but also to the node placement and the generation of the mesh structure, which are computed from the feature map.

Fig. 24(a) shows the feature map extracted from the degraded image and Fig. 24(b) corresponds to the feature map from the degraded image after applying a Gaussian low-pass filter with a length  $L = 9$  samples and a standard deviation  $\sigma = 0.7$ . If we compare them to Fig. 24(c), which is the feature map extracted from the original image, we can see that (b) is closer to the ideal case. Fig. 24(d) to (f) correspond to the resulting nodal placement (number of nodes used  $N = 1853$ , *ratio = 10 pixels/node*) when the feature maps from (a) to (c) are used respectively. We can see that in (d) the nodes are placed more randomly due to the noise. Fig. 24(g) to (i) are the mesh structures after applying the Delaunay triangulation between the nodes from (d) to (f), respectively.

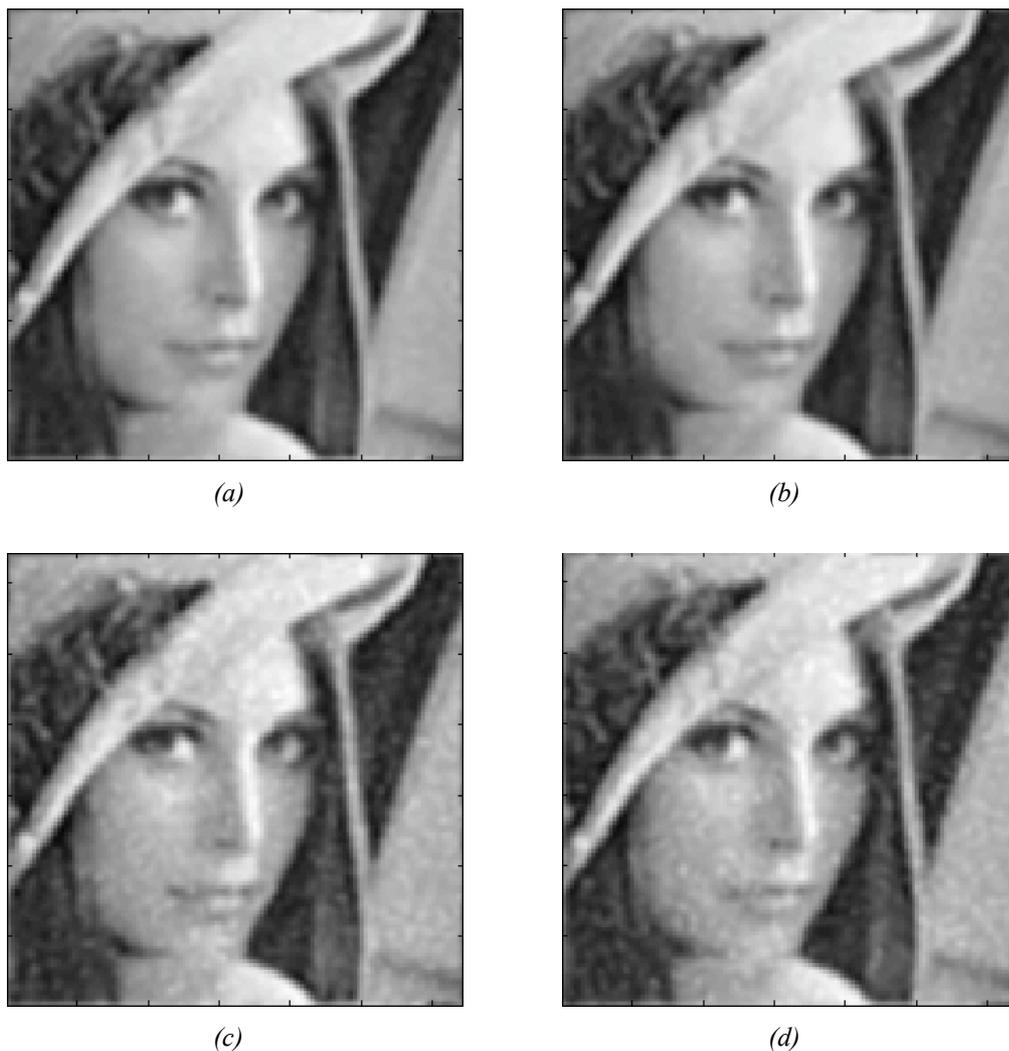


*Fig. 24 (a) to (c) feature maps from the degraded image, the filtered image and the original image, respectively; (d) to (f) nodes placement from the degraded image, the filtered image and the original image, respectively; and (g) to (i) mesh structure from the degraded image, the filtered image and the original image, respectively.*

The value of *gamma* ( $\gamma$ ) used in the computation of the feature map is another important factor to take into account. It has to be adjusted in order to obtain the most accurate approximation of the original image. Thus, it is necessary to find the optimal value of *gamma* for every restoration problem.

For example, when the original image is available for the feature map extraction and the noise which degrades the image has a standard deviation  $\sigma = 0.002$ , the optimal value of *gamma* is  $\gamma = 1$ . However, when we use an estimated image to compute the feature map, this optimal value may differ. The use of a different number of nodes or the changes in the degradation system, such as the noise power, make it necessary to adapt the value of *gamma* in order to obtain a more accurate restored image.

Fig. 25 shows the comparison of the optimal value of *gamma* for different cases of image restoration. Fig. 25 (a) and (b) correspond to the restoration of an image degraded by noise with a standard deviation  $\sigma = 0.002$ . In (a) the optimal value of *gamma*  $\gamma = 1.4$  was used, while in (b) it was  $\gamma = 1$ . In the first case the MSE is lower and the PSNR, which is inversely proportional to the MSE, is higher. On the other hand, in Fig. 25 (c) and (d) the standard deviation of the noise was  $\sigma = 0.01$ . In (c) we used the optimal *gamma* which was  $\gamma = 0.3$  while in (d) we used  $\gamma = 1$ . In all four cases the *ratio* used was of *4 pixels/node* and the degraded image was filtered by a Gaussian filter ( $\sigma = 0.7, L = 9$ ) prior to the feature map extraction in order to reduce the noise.



*Fig. 25 (a) and (b) restored images from a degraded image with a noise standard deviation  $\sigma = 0.002$ . In (a) the feature map was computed with the optimal gamma  $\gamma = 1.4$  (PSNR = 26.9104 dB); in (b)  $\gamma = 1$  (PSNR = 26.7977 dB); (c) and (d) were restored from an image degraded by noise with a standard deviation  $\sigma = 0.01$ . In (c)  $\gamma = 0.2$  (PSNR = 22.6869 dB) while in (d)  $\gamma = 1$  (PSNR = 22.5517 dB). A ratio of 4 pixels/node were used in all the restored images.*

Ideally, this optimal value is found with an iterative algorithm based on finding the minimum mean square error between the original image and the restored image for different values of *gamma*. However, the original image is unknown, so the mean square error must be calculated between the restored image and an estimation of the original image. In this study we used the Standard Wiener filter to estimate the image.

### 3.3.2 Least Squares Fitting

The Least Squares Fitting used is an iterative algorithm whose goal is to minimize the mean square error between the degraded image and the restored image filtered by the blurring system. If the number of iterations used is too large, this solution can yield to a bad approximation. The reason is that the noise from the degraded image is not taken into consideration in the computation of the optimized nodal values, however, they depend on the noise.

From the expression obtained in (67) we can compute the nodal values as follows:

$$\mathbf{n} = (\Phi^T \mathbf{H}^T \mathbf{H} \Phi)^{-1} \Phi^T \mathbf{H}^T \mathbf{g}, \quad (72)$$

$$\mathbf{n} = (\Phi^T \mathbf{H}^T \mathbf{H} \Phi)^{-1} \Phi^T \mathbf{H}^T (\mathbf{H} \mathbf{f} + \mathbf{w}), \quad (73)$$

where  $\mathbf{n}$  is the column vector containing the new nodal values,  $\Phi$  is the interpolation filter,  $\mathbf{H}$  is the blurring filter,  $\mathbf{f}$  is the original image, and  $\mathbf{w}$  is the additive noise.

We can see that the noise  $\mathbf{w}$  is filtered by  $(\Phi^T \mathbf{H}^T \mathbf{H} \Phi)^{-1} \Phi^T \mathbf{H}^T$  which is a high-pass filter. Therefore, for every iteration the noise is amplified relative to the image function and it yields to inaccurate nodal values. Fig. 26 shows the evolution of the MSE between the restored image and the original image relative to the number of iterations and how it increases after some iterations due to the noise amplification. A *ratio* of 10 was used, and the total number of iterations is 200.

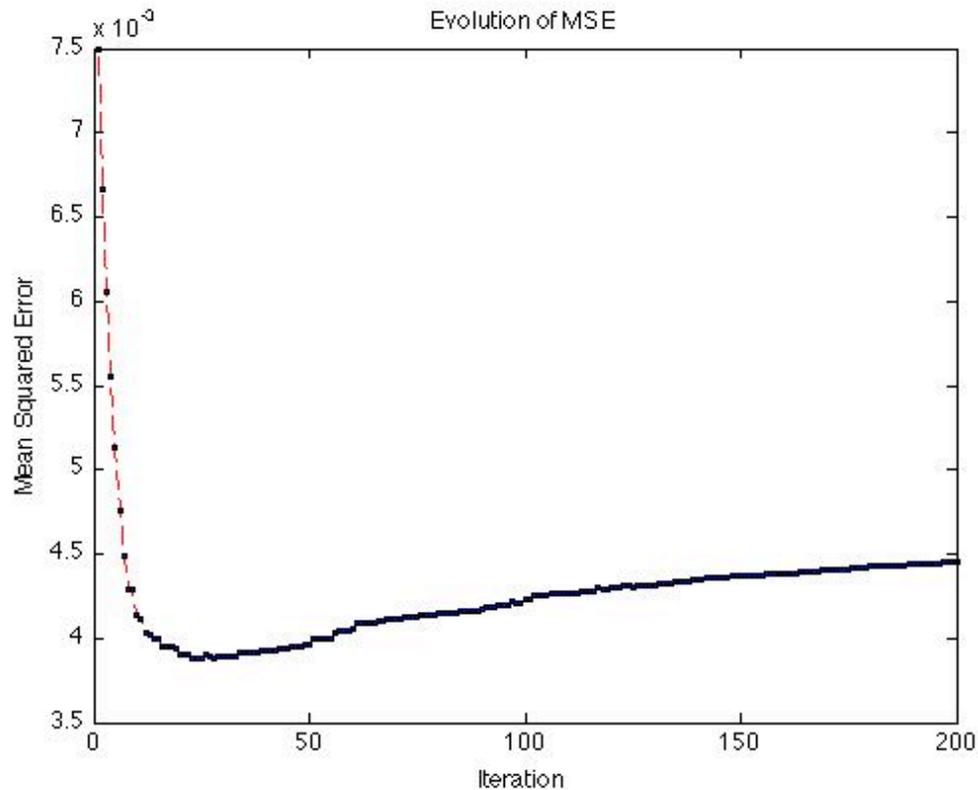


Fig. 26 Evolution of the MSE between the original image and the restored image during a least squares fitting of 200 iterations.

The number of iterations needed to obtain the best approximation of the nodal values with the least squares algorithm also depend on the initial values used for the mesh nodes: if the initial values are, for example, one ( $\mathbf{n}$  equal to a vector of ones), the number of iterations is higher than if the initial values are taken from the degraded image.

## 3.4 Numerical Results

### 3.4.1 Comparison with Classical Methods

In this chapter we have seen that the Content Adaptive Mesh Model is able to restore an image degraded by blur and noise. The main advantage of the algorithm used in CAMM is that it has some parameters which can be adapted in order to obtain a more accurate approximation of the original undegraded image. The main parameters are the *gamma* ( $\gamma$ ) used in the feature map and the *ratio pixels/node* (proportional to the number of nodes used).

Depending on the characteristics and level of degradation of the image, CAMM yields to different results. In order to demonstrate that it works well with degraded images we are going to compare CAMM with some classical methods. The results will be compared with the use of images as well as numerical results that will show the level of accuracy of the restored image. Besides the commonly used criterion of goodness peak signal-to-noise ratio (PSNR), we also used a quality index ( $QI$ ) which compares the images by three different factors: correlation, mean luminance, and contrast. Its expression is defined as

$$QI = \frac{\sigma_{xy}}{\sigma_x \sigma_y} \frac{2\bar{x}\bar{y}}{(\bar{x})^2 + (\bar{y})^2} \frac{2\sigma_x \sigma_y}{\sigma_x^2 + \sigma_y^2}, \quad (74)$$

where  $\bar{x}$  and  $\bar{y}$  are the means, and  $\sigma_x$  and  $\sigma_y$  are the correlation coefficients of the image  $x$  and image  $y$  respectively. The first term measures the level of correlation between the images  $x$  and  $y$ . The second component corresponds to the luminance and finally, the last term measures how close are the contrasts of the two images. This quality index is explained with more detail in the literature [22], [23].

The image to restore is a 128x128 section of ‘‘Lena’’ degraded by Gaussian blur (standard deviation  $\sigma = 0.8511$  and length  $L = 5$ ) and additive white Gaussian noise (standard deviation  $\sigma = 0.002$ ). This image is shown in Fig. 27 (a) and it has a PSNR = 25.396 dB and a QI = 0.76444. Firstly we are going to compare the ideal cases, i.e. the original image is available to design the algorithms, which yield to the best restored images. When the feature map can be extracted from the original image, the image restored with the CAMM algorithm (Ratio = 4 and  $\gamma = 1$ ), shown in Fig. 27 (b), has a PSNR = 27.8645 dB and a QI = 0.87195. Fig. 27 (c) corresponds to the same image restored by the Standard Wiener filter ( $\gamma = 1$ ) when the exactly spectra  $S_n$  and  $S_f$  are used (PSNR = 26.8484 dB and QI = 0.82224). The corrupted image was also reconstructed by a median filter, whose resulting image is Fig. 27 (d) (PSNR = 25.859 dB and QI = 0.81774), and also by an averaging filter in Fig. 27 (e) (PSNR = 23.6347 dB and QI = 0.79051). Thus, in the ideal case CAMM with a ratio of 4 pixels/node and  $\gamma = 1$  yields to a better restored image. In fact, even the use of a higher ratio (less mesh nodes) yields to better results than Wiener, median, and averaging filters. For a Ratio = 10 the image shown in Fig. 27 (f) has a PSNR = 27.0791 dB and a QI = 0.82044. Table 4 shows the numerical comparison of all the methods used to reconstruct the degraded image.

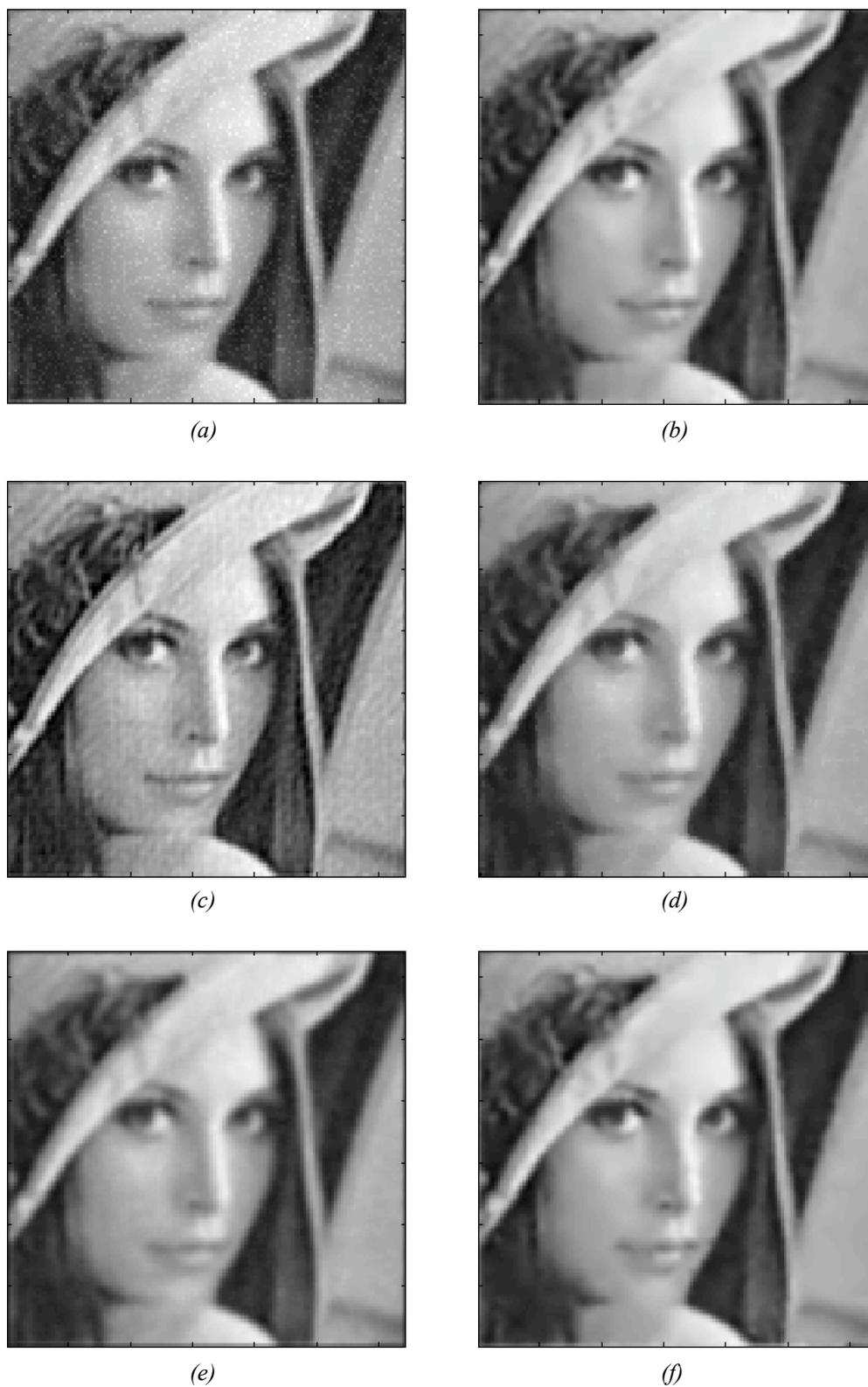


Fig. 27 (a) A 128x128 section of the degraded image from “Lena”; (b) restored image using CAMM with a Ratio = 4 pixels/node and  $\gamma = 1$  and the feature map extracted from the original image; (c) image restored with the Standard Wiener filter; (d) image obtained by median filtering; (e) image obtained by averaging filtering; and (f) restored image using CAMM with a Ratio = 10 pixels/node.

Method	Parameters	PSNR (dB)	QI
CAMM	Ratio = 4 $\gamma = 1$	27.8645	0.87195
CAMM	Ratio = 10 $\gamma = 1$	27.0791	0.82044
Wiener	$\gamma = 1$	26.8484	0.82224
Median	3x3 neighborhood	25.859	0.81774
Averaging	size 3x3	23.6347	0.79051

Table 4 PSNR and Quality Index for each restoration method: Mesh Model with a ratio of 4 pixels/node and  $\gamma = 1$ ; Mesh Model with a ratio of 10 pixels/node and  $\gamma = 1$ ; Standard Wiener filter; Median filter with a 3-by-3 neighborhood; and Averaging filter of a size equal to 3-by-3.

Based on the images and the results showed in Table 4, the mesh model achieves better results than the other methods. However, the design of the CAMM and the Wiener filter are unfeasible due to the use of some unavailable information about the original image, such as its values at some points or its frequency spectrum.

Next, we use the same degraded (blurred and noisy) image (PSNR = 25.396 dB and QI = 0.76444) in order to compare the different methods when there is no *a priori* information about the original image. As in section 3.3.1, the degraded image is processed by a Gaussian low-pass filter ( $\sigma = 0.7$ ,  $L = 9$ ) prior to the feature map extraction in order to reduce the noise. Fig. 28 (a) is the restored image by the CAMM algorithm using a ratio of 4 pixels/node and an optimal *gamma*  $\gamma = 1.3$ , which has a PSNR = 27.0357 dB and a QI = 0.83519. Fig. 28 (b) corresponds to the restored image when the ratio is changed to 10 pixels/node and *gamma* adjusted to  $\gamma = 1.1$  (PSNR = 26.1826 dB and QI = 0.81674). When the Standard Wiener filter is used ( $\gamma = 1$ ) the resulting image corresponds to Fig. 28 (c) (PSNR = 26.3902 dB and QI = 0.83241). If we use the Parametric Wiener filter with  $\gamma = 0.4$ , the resulting image shown in Fig. 28 (d) achieves a PSNR = 26.6195 dB and a QI = 0.82654.

Finally, in Table 5 all the numerical results are compared, included the results from the median filter and the averaging filter used before in Table 4. In this case the Parametric Wiener filter has a higher PSNR than the Standard Wiener filter but, on the contrary, it has lower QI. The reason may be due to the fact that the PSNR is related to the correlation while the QI takes into account two more measures: luminance distortion and contrast distortion. The mesh model using a ratio of 4 pixels/node and  $\gamma = 1.3$  yields to a more accurate image than the Wiener, median, and averaging filters. However, CAMM with a higher ratio (10 pixels/node) obtains worse results (lower PSNR and QI) than the Wiener filter.

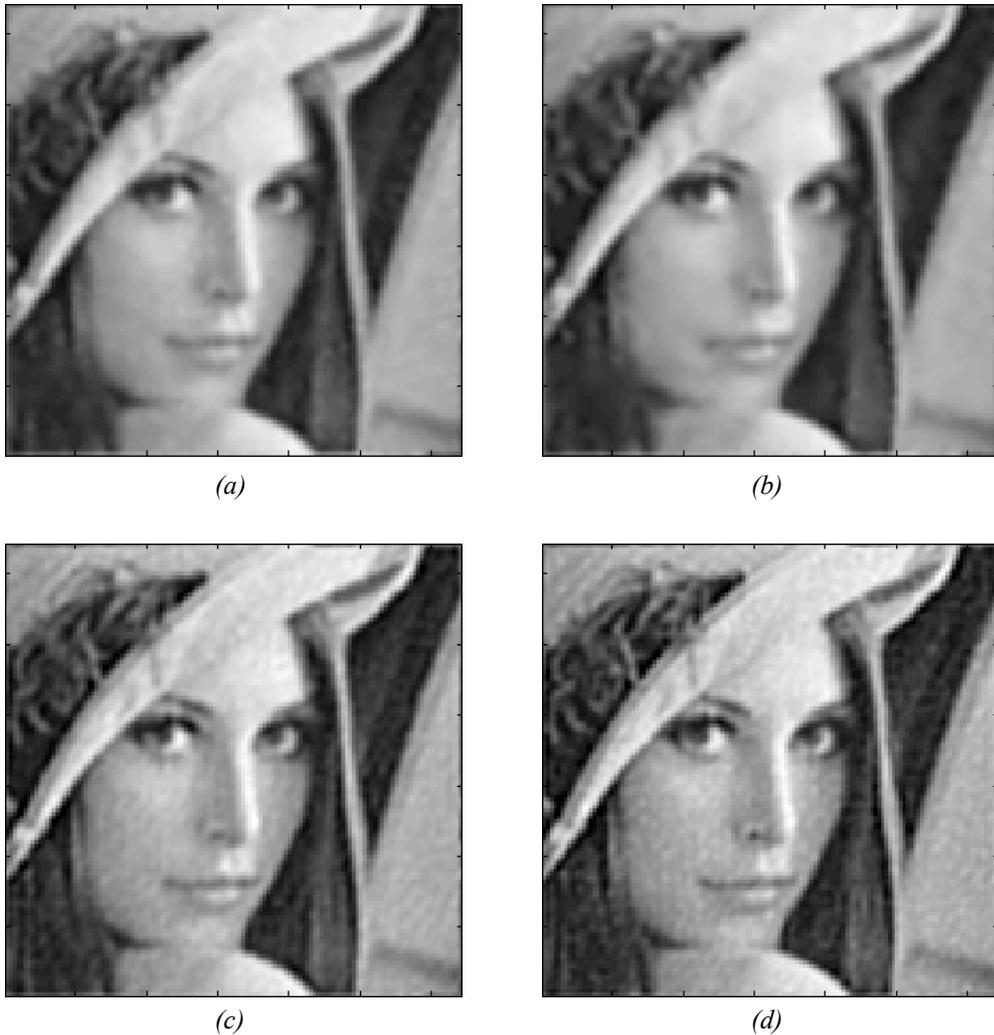


Fig. 28 (a) Restored image using CAMM with a Ratio = 4 pixels/node and  $\gamma = 1.3$ , and the feature map extracted from the degraded image after applying a Gaussian low-pass filter with standard deviation  $\sigma = 0.7$  and length  $L = 9$ ; (b) image restored using CAMM with a Ratio = 10 and  $\gamma = 1.1$ ; (c) resulting image using the Standard Wiener filter ( $\gamma = 1$ ) when the spectrum of the degraded image is used; and (d) restored image obtained from the Parametric Wiener filter ( $\gamma = 0.4$ ) when the spectrum of the degraded image is used as well.

Method	Parameters	PSNR (dB)	QI
CAMM	Ratio = 4 $\gamma = 1.3$	27.0357	0.83519
CAMM	Ratio = 10 $\gamma = 1.1$	26.1826	0.81674
Wiener	$\gamma = 1$	26.3902	0.83241
Wiener	$\gamma = 0.4$	26.6195	0.82654
Median	3x3 neighborhood	25.859	0.81774
Averaging	size 3x3	23.6347	0.79051

*Table 5 PSNR and Quality Index for each method: Mesh Model with a ratio of 4 pixels/node and  $\gamma = 1.3$ ; Mesh Model with a ratio of 10 pixels/node and  $\gamma = 1.1$ ; Standard Wiener filter; Parametric Wiener filter with  $\gamma = 0.4$ ; Median filter with a 3-by-3 neighborhood; and Averaging filter of a size equal to 3-by-3.*

### 3.4.2 Comparison with Other Images

We have seen that the mesh model can restore a section of the image “Lena” degraded by blur and noise with similar or even better results than classical restoration methods such as the Wiener filter. In this section we want to demonstrate that CAMM works well with any kind of image, because the mesh structure is adapted to the image features.

Firstly, a 256-by-256 section of the image “Tech” shown in Fig. 29(a) is used to demonstrate how the mesh structure is perfectly adapted to the image content. The mesh shown in Fig. 29(b) used 16655 nodes ( $Ratio = 4 \text{ pixels/node}$ ) and the corresponding reconstructed image using least squares fitting is shown in Fig. 29(c) (PSNR = 35.0823 dB, QI = 0.91942). It is clear that the mesh representation yields good results for any image.

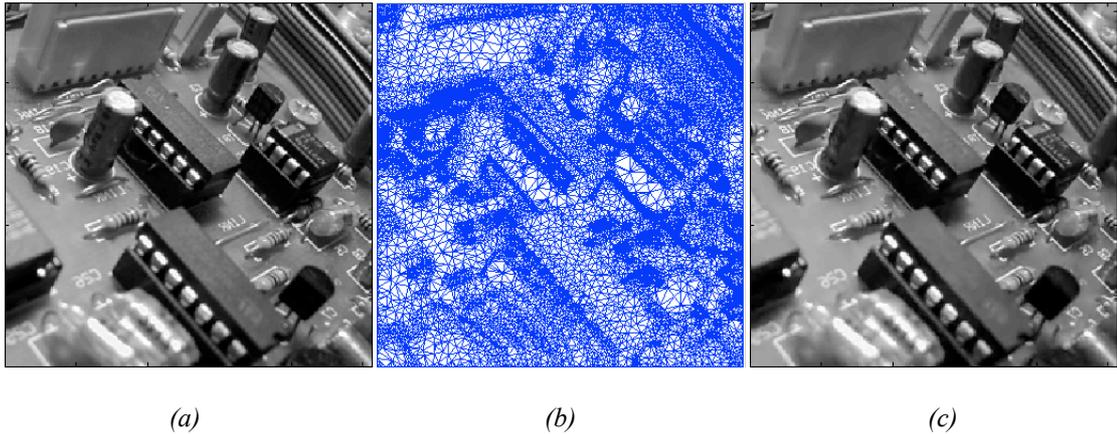


Fig. 29 (a) A 256 x 256 section of the original image “Tech”; (b) mesh structure obtained with CAMM ( $Ratio = 4 \text{ pixels/node}$ ,  $\gamma = 1$ ); and (c) image represented using the mesh from (b) and least squares fitting (PSNR = 35.0823 dB, QI = 0.91942).

Image: [www.freeimages.co.uk](http://www.freeimages.co.uk)

The image used in Fig. 29(a) is now degraded by Gaussian blur and additive white Gaussian noise (PSNR = 23.7448 dB, QI = 0.68143) and it is shown in Fig. 30(a). Shown in Fig. 30(b) is the restored image using the mesh from Fig. 29(b) and least squares fitting (PSNR = 25.7872 dB, QI = 0.82201). Fig. 30(c) corresponds to the image obtained from applying the Standard Wiener filter (PSNR = 25.1982 dB, QI = 0.77066); and (d) is the resulting image obtained with median filtering (PSNR = 24.4036 dB, QI = 0.75424). These results, shown in Table 6, demonstrate again that the CAMM is the method which achieves a more accurate restored image.

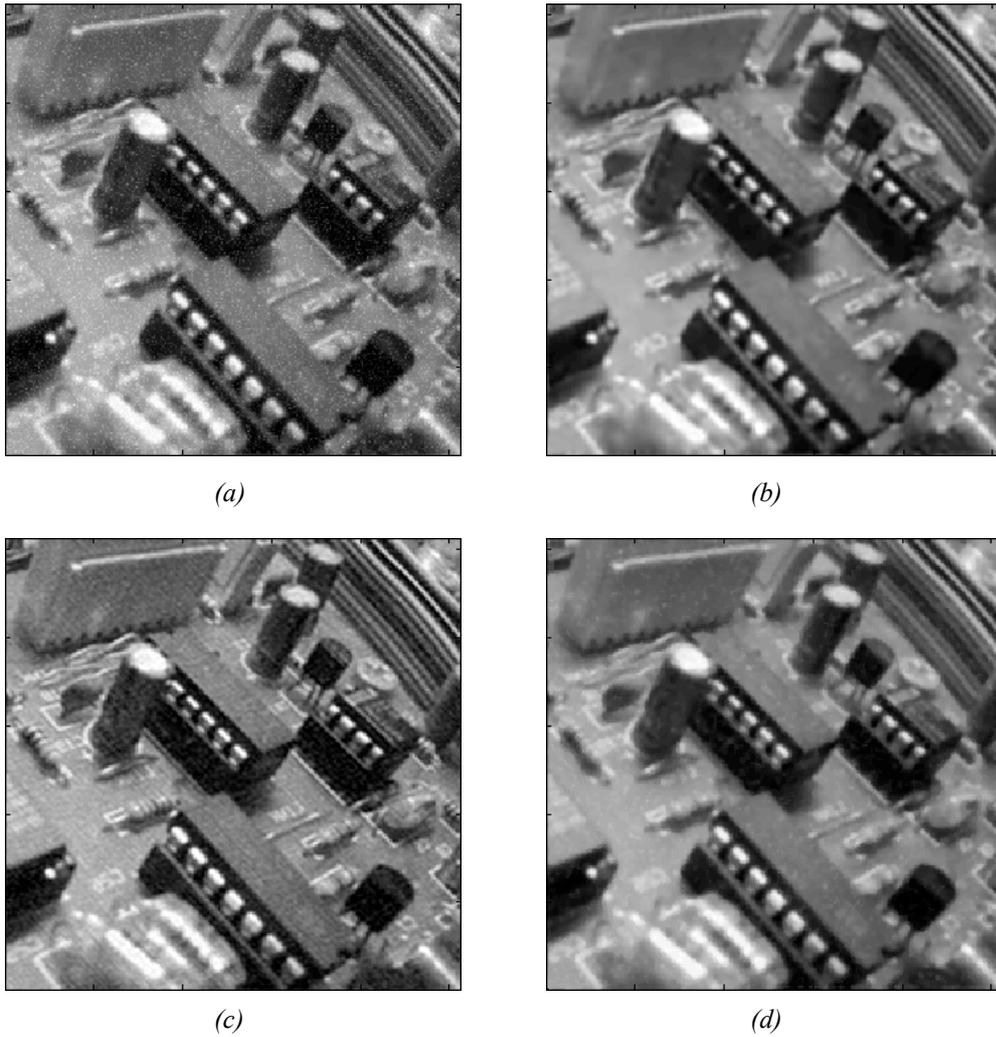


Fig. 30 (a) A 256 x 256 section of the degraded image “Tech”; (b) image obtained with CAMM using the mesh from Fig. 29 (b) (Ratio = 4 pixels/node,  $\gamma = 1$ ) and least squares fitting; (c) image restored with Standard Wiener filter; and (d) image obtained with median filtering.

Method	Parameters	PSNR (dB)	QI
CAMM	Ratio = 4 $\gamma = 1$	25.7872	0.82201
Wiener	$\gamma = 1$	25.1982	0.77066
Median	3x3 neighborhood	24.4036	0.75424

Table 6 PSNR and Quality Index for each restoration method: Mesh Model with a ratio of 4 pixels/node and  $\gamma = 1$ ; Standard Wiener filter; and Median filter with a 3-by-3 neighborhood.

# CHAPTER 4

## CONCLUSION

### 4.1 Summary

In this study, we have developed a new method to represent digital images using a mesh structure. The mesh is adapted to the image features, placing more samples in areas with a high frequency content, and, on the other hand, less samples in regions with a low frequency content. This is, therefore, a nonuniform sampling method, which yields to a more accurate image representation than uniform sampling methods. Using the mesh model we achieve an adapted smoothness of the image. Some classical algorithms were used in order to design the mesh structure: the Floyd-Steinberg algorithm for error diffusion (also used in image dithering) was used for the mesh nodes (samples) placement, and the Delaunay triangulation algorithm was used in order to form the mesh elements from the mesh nodes. In addition, a Least Squares algorithm was also used in order to improve the nodal intensities or values so the resulting image is more accurate. Our experimental results demonstrate that the mesh model proposed can be used, moreover, as an image restoration technique, since it is able to reconstruct images degraded by blur and noise with similar or even better results than other classical technique, such as the Wiener filter.

### 4.2 Future Work

In future work, we will focus on an image reconstruction technique based on the content adaptive mesh model proposed in this study, along with the use of the theoretical basis used in the Wiener filter. With this technique it may be possible to improve the quality of the reconstructed images using the mesh model. In addition, it would be interesting to find other techniques to estimate the original image in some important parts of the design, such as in the future map extraction, where a good estimation of the original image is necessary in order to place the nodes correctly.

## APPENDIX A

### RELATION BETWEEN HIGH FREQUENCY CONTENT AND SECOND PARTIAL DERIVATIVES

The mesh is designed to place more mesh nodes in those areas of the image where there is more detail, i.e. where there is a high frequency image content. The feature map, which indicates where the mesh nodes have to be placed, is computed based on the second partial derivatives. To demonstrate the relation between the high frequency content areas and the second partial derivatives let us consider an image  $f(x,y)$ , whose Fourier transform ( $FT$ ) is denoted by  $F(u, v)$ . Then the Fourier transform of the second partial derivatives of the image  $f(x,y)$  can be computed as follows:

$$FT \left\{ \frac{\partial^2 f(x,y)}{\partial x^2} \right\} = -u^2 F(u,v), \quad (\text{A-1})$$

$$FT \left\{ \frac{\partial^2 f(x,y)}{\partial x \partial y} \right\} = -uv F(u,v), \quad (\text{A-2})$$

$$FT \left\{ \frac{\partial^2 f(x,y)}{\partial y^2} \right\} = -v^2 F(u,v). \quad (\text{A-3})$$

It is clear that the second partial derivatives are directly related to the high frequency content of an image, since the variables  $u$  and  $v$ , which are the frequency variables, appear as a product with the Fourier transform of the image.

## APPENDIX B

### FEATURE MAP COMPUTATION BY THE HESSIAN EIGENVALUES

We have seen that the high frequency content of an image is related to its second directional derivatives. Now we will demonstrate that the feature map of an image, which must indicate the areas with high frequency content, can be computed by finding the maximum eigenvalues of the Hessian matrix of the image.

Recall that the feature map is computed as

$$\sigma(x,y) = \left( \frac{G(x,y)}{A} \right)^\gamma, \quad (\text{B-1})$$

where  $G(x,y)$  corresponds to the largest magnitude of the second directional derivative of an image  $f(x,y)$ , that is,

$$G(x,y) = \max_{\theta \in [0, 2\pi]} |f''_{\theta}(x,y)|. \quad (\text{B-2})$$

The directional derivative of  $f(x,y)$  is defined as

$$f'_{\theta}(x,y) = \mathbf{u}_{\theta}^T \nabla f(x,y). \quad (\text{B-3})$$

If we apply this equation again, we can obtain the second directional derivative

$$f''_{\theta}(x,y) = \mathbf{u}_{\theta}^T \nabla (\mathbf{u}_{\theta}^T \nabla f(x,y)) = \mathbf{u}_{\theta}^T \nabla (\nabla f(x,y))^T \mathbf{u}_{\theta}. \quad (\text{B-4})$$

Now, if we consider the Hessian matrix of the image denoted by  $H(x,y)$  and defined by

$$\mathbf{H}(x,y) = \begin{pmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial y^2} \end{pmatrix}, \quad (\text{B-5})$$

the expression to compute the second partial derivative can be written as

$$f''_{\theta}(x,y) = \mathbf{u}_{\theta}^T \mathbf{H}(x,y) \mathbf{u}_{\theta} \quad (\text{B-6})$$

and

$$|f''_{\theta}(x,y)| = |\mathbf{u}_{\theta}^T \mathbf{H}(x,y) \mathbf{u}_{\theta}| \leq \max\{|\lambda_1(x,y)|, |\lambda_2(x,y)|\}, \quad (\text{B-7})$$

so, finally, the feature map can be compute as follows:

$$G(x,y) = \max_{\theta \in [0, 2\pi]} |f''_{\theta}(x,y)| = \max\{|\lambda_1(x,y)|, |\lambda_2(x,y)|\}. \quad (\text{B-8})$$

## APPENDIX C

### LEAST SQUARES FITTING

As seen in section 3.2.5, the restored image using the mesh model can be written using the discrete form as follows:

$$\tilde{\mathbf{f}} = \Phi \mathbf{n}, \quad (\text{C-1})$$

where  $\tilde{\mathbf{f}}$  is the column vector formed by all the points ( $M$  pixels) of the restored image,  $\mathbf{n}$  is the column vector formed by all the nodal values ( $N$  mesh nodes), and  $\Phi$  is the matrix of the interpolation function for each node all over the image.

We have demonstrated in section 3.2.4 that the linear interpolation between the mesh nodes is not the most accurate way to define the intensity of the mesh elements. Thus, we want to improve the approximation of the image by using a least squares fit. The value of the mesh nodes are recalculated so that the mean square error between the original image and the restored image is minimized, i.e,

$$\mathbf{n} = \arg \min_{\mathbf{n}} \|\mathbf{f} - \tilde{\mathbf{f}}\|^2. \quad (\text{C-2})$$

If we use the expression in the discrete form defined in (C-1), the function to minimize can be written as follows:

$$J(\mathbf{n}) = \|\mathbf{f} - \Phi \mathbf{n}\|^2, \quad (\text{C-3})$$

If we develop this expression it becomes

$$J(\mathbf{n}) = \|\mathbf{f} - \Phi \mathbf{n}\|^2 = (\mathbf{f} - \Phi \mathbf{n})(\mathbf{f} - \Phi \mathbf{n})^T = -2\mathbf{n}^T \Phi^T \mathbf{f} + \mathbf{f}^T \mathbf{f} + \mathbf{n}^T \Phi^T \Phi \mathbf{n}. \quad (\text{C-4})$$

Then, the new nodal values can be computed by an iterative algorithm

$$\mathbf{n}^{i+1} = \mathbf{n}^i - \alpha \nabla J(\mathbf{n}), \quad (\text{C-5})$$

where  $\mathbf{n}^i$  and  $\mathbf{n}^{i+1}$  are the nodal values for the current and the next iterations respectively,  $\alpha$  is the value of the step used in each iteration, and  $\nabla J(\mathbf{n})$  is the gradient of the minimizing function defined as

$$\nabla J(\mathbf{n}) = \frac{\partial J(\mathbf{n})}{\partial \mathbf{n}} = -\Phi^T \mathbf{f} + \Phi^T \Phi \mathbf{n}. \quad (\text{C-6})$$

## REFERENCES

- [1] J. L. Starck and F. Murtagh, "Astronomical image and signal processing: looking at noise, information and scale," in *IEEE Signal Processing Magazine*, March 2001.
- [2] W. Guangxin, W. Zhengming, X. Meihua, and L. Yarui, "Variation-Based Approach to Restoring Blurred Images Corrupted by Poisson Noise" in *8th International Conference on Signal Processing*, Nov. 2006.
- [3] R. C. Gonzalez and R. E. Woods, "Noise Models," in *Digital Image Processing*, 2008, pp. 313-322.
- [4] M. R. Banham and A. I. Katsaggelos, "Digital Image Restoration," in *IEEE Signal Processing Magazine*, March 1997.
- [5] P. Gravel, G. Beaudoin, and J. A. De Guise, "A Method for Modeling Noise in Medical Images," in *IEEE Transactions on Medical Imaging*, vol. 23, No. 10, Oct. 2004.
- [6] J. M. Park, W. J. Song, and W. A. Pearlman, "Speckle filtering of SAR images based on adaptive windowing," in *IEE Proceedings - Vision, Image Signal Processing*, vol. 146, No. 4, Aug. 1999.
- [7] Y. C. Trivedi and L. Kurz, "An Experimental Design Approach to Image Enhancement," in *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 22, No. 4, Jul/Aug 1992.
- [8] S. Engelberg, "The Central Limit Theorem and Low-pass filters," in *Proceedings of the 2004 11th IEEE International Conference on Electronics, Circuits and Systems*, Dec. 2004.
- [9] R. C. Gonzalez and R. E. Woods, "Filtering in the Frequency Domain," in *Digital Image Processing*, 2008, pp. 199-310.
- [10] R. C. Gonzalez and R. E. Woods, "Inverse Filtering," in *Digital Image Processing*, 2008, pp. 251-252.
- [11] T. Berger, J. O. Strömberg, and T. Eltoft, "Adaptive Regularized Constrained Least Squares Image Restoration," in *IEEE Transactions on Image Processing*, vol. 8, No. 9, Sept. 1999.
- [12] L. Guan and R. K. Ward, "Restoration of Randomly Blurred Images by the Wiener Filter", in *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 37, No. 4, Apr. 1989.

- [13] R. C. Gonzalez and R. E. Woods, "Least Mean Square (Wiener) Filter," in *Digital Image Processing*, Sept. 1993, pp. 279-282.
- [14] E. Oja and H. Ogawa, "Parametric Projection Filter for Image and Signal Restoration," in *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-34, No. 6, Dec. 1986.
- [15] I. Álvarez-Cascos and Y. Yang, "Least-Squares Mesh Model for Image Compression," in *International Conference on Image Processing*, 2004.
- [16] J. P. Pons and J. D. Boissonnant, "Delaunay Deformable Models: Topology-Adaptive Meshes Based on the Restricted Delaunay Triangulation," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2007, pp. 1-8.
- [17] J. Lautissier, L. Legrand, A. Lalande, P. Walker, and F. Brunotte, "Object tracking in medical imaging using a 2D active mesh system," in *Proceedings of the 25th Annual International Conference of the IEEE BMBS*, Cancun, Mexico, Sept. 2003.
- [18] Y. Yang, M. N Wernick, and J. G. Brankov, "A Fast Approach for Accurate Content-Adaptive Mesh Generation," in *IEEE Transactions on Image Processing*, Vol. 12, No. 8, August 2003.
- [19] J. G. Brankov, Y. Yang, and M. N Wernick, "Tomographic Image Reconstruction Based on a Content-Adaptive Mesh Model," in *IEEE Transactions on Medical Imaging*, Vol. 23, No. 2, Feb. 2004.
- [20] J. G. Brankov, Y. Yang, and M. N Wernick, "Multi-Modality Tomographic Image Reconstruction Using Mesh Modeling," in *IEEE International Symposium on Biomedical Imaging*, 2002.
- [21] R. W. Floyd and L. Steinberg, "An Adaptive Algorithm for Spatial Grayscale," in *Proceedings of the Society for Information Display*, Vol. 7, No. 2, 1976.
- [22] Z. Wang and A.C. Bovik, "A Universal Image Quality Index," in *IEEE Signal Processing Letters*, Vol. 9, No. 3, pp. 81-84, Mar. 2002.
- [23] Z. Wang and A.C. Bovik, "Mean Squared Error: Love It or Leave It?," in *IEEE Signal Processing Magazine*, Jan. 2009.

## OTHER BIBLIOGRAPHY

- A. C. Bovik, *Handbook of Image and Video Processing*, First Edition, 2000.
- A. C. Bovik, *The Essential Guide to Image Processing*, 2009.
- R. C. Gonzalez and R. E. Woods, *Digital Image Processing Using MATLAB*, 2004.