Optimization Methods for Multiple-Source Image Fusion

Manikis Georgios

A Thesis submitted for the Degree of Diploma
in the Department of Electronic & Computer Engineering

Technical University of Crete

Supervisory Committee

Professor Michalis Zervakis (Supervisor)
Associate Professor Athanasios Liavas
Associate Professor Evripides Petrakis

May 2004

Acknowledgements

I wish to express my sincere appreciation to Professor Michalis Zervakis for his valuable assistance and excellent cooperation in every single stage of this work.

In addition, special thanks to Associate Professor Athanasios Liavas and to Associate Professor Evripides Petrakis for their participation in the supervisory committee.

Finally, I would like to thank all my friends and especially Iliana, Nikos and Manos for their love and support throughout all the years of my studies.

This thesis is dedicated to my family and to a lovely friend who left very early.

Terminology

NDT	Non Destructive Testing	(Chapter 1)
ME	Maximum Entropy	(Chapter 2)
ML	Maximum Likelihood	(Chapter 2)
MAP	Maximum A Posteriori	(Chapter 2)
POCS	Projection Onto Convex sets	(Chapter 2)
MRI	Magnetic Resonance Image	(Chapter 2, 3, 4)
CT	Computed Tomography	(Chapter 2, 3, 4)
WGN	White Gaussian Noise	(Chapter 2)
MSE	Mean Square Error	(Chapter 2, 3, 4)
ANN	Artificial Neural Network	(Chapter 3)
PSF	Pointwise Spatial Frequency	(Chapter 3)
GD	Gradient Descent	(Chapter 3)
BAYES	Bayesian Approach	(Chapter 3)
SPECT	Single-Photon Emission CT	(Chapter 4)
DFT	Discrete Fourier Transform	(Appendix B)

1 Introduction

Data Fusion is one of the active areas of research in many applications such as non-destructive testing (NDT), geophysical imaging, medical images, radio astronomy, military applications, etc. This technology enables the images from different sensors to be combined into a single image that has more information or information of a better quality than any of the individual images alone. For example, thermal infrared or millimetre-wave radar images can be combined with visual spectrum images to create a system that provides clearer pictures in poor visibility conditions.

Thesis research, with respect to the other fields of image fusion, deals with combining different sources of information from modern biomedical systems. Technological advances in medical imaging in the past two decades have enabled radiologists to create images of the human body and its internal structures with unprecedented resolution and realism. State-of-the-art CT, MR images, and other imaging devices can quickly acquire 3D images, and these images can further be computed to merge into a single volume thus combining the information of all modalities. The available computing power and sophisticated display software allow for the fused images to be captured on screen with both scientific authenticity and aesthetic worth.

Who should have Image Fusion? Not everyone needs image fusion. In most instances the diagnostic information from each individual study is sufficient. On the other hand, the need to register different imaging modalities arises for several reasons. Some anatomical details, especially soft tissues, are more easily seen in MR images than CT images, but the bony structures are better visualised by CT. Medical image fusion from the previous described medical systems has been applied to the diagnosis of breast cancer, colon cancer, cardiac studies, wrist and other injuries, inflammatory diseases and different neurological disorders including brain tumors, Alzheimer's disease and schizophrenia. This method has also been utilized in radiotherapy, mostly for brain tumors, and by cranio-facial surgeons to prepare for and simulate complex surgical procedures.

For that reason, the idea of image fusion processing between the MR and CT images is really seductive. Unfortunately, all the optical systems, electronic and

photometric sources, and generally all the measurement systems that obtain the diagnostic information to the human review, contribute to the degradation. Degradation, which includes the blurring as well as the noise consequence, affects inefficiently the images and generates problems to the characteristics of the objects that are represented in them. Historically, a large-portion of digital image processing activity has been devoted to avoid such a problem. Image restoration methods; aim to bring the image toward what it would have been if it had been recorded without degradation.

Interpolating the image restoration method between the raw data from the measurement systems and the image fusion application, incurs an efficient way to optimize the knowledge of the sources and to prepare the image fusion application with better quality images.

In Chapter 2, image restoration methods are performed in order to release the raw data from the degradation causes. In Chapter 3, the image fusion procedure is evolving, trying to combine images from different sensors into a single image. In Chapter 4, evaluation and discussion of our approach is attempted along with possible future directions. Finally, in Appendix A, we represent all the functions of our code that simulated the proposed image restoration and image fusion models.

Our work can be seen as an extension of Ali Mohammad-Djafari work, referenced in **[8]**, towards probabilistic model based methods for data fusion.

In the following flow chart, we represent the steps that this thesis has performed. In the first step, the main strategy is to remove the degradation from the MR and CT images, given from the two measurement systems, with respect to the image restoration methods. Magnetic resonance and computed tomography images are restored separately with the independent and the joint restoration models, and the resulted images are the input sources for the next step.

In the next step the restored MR images are combined into a single image that has information of a better quality than any of the individual restored images alone. The same approaches occur in the case of the restored CT images. The goal is to have only one good quality CT image as well as a MR image.

In the final step, the fused magnetic resonance and computed tomography images are combined into a single image that contains high quality information about the bony structures and the anatomical data of the object corresponded in the images.

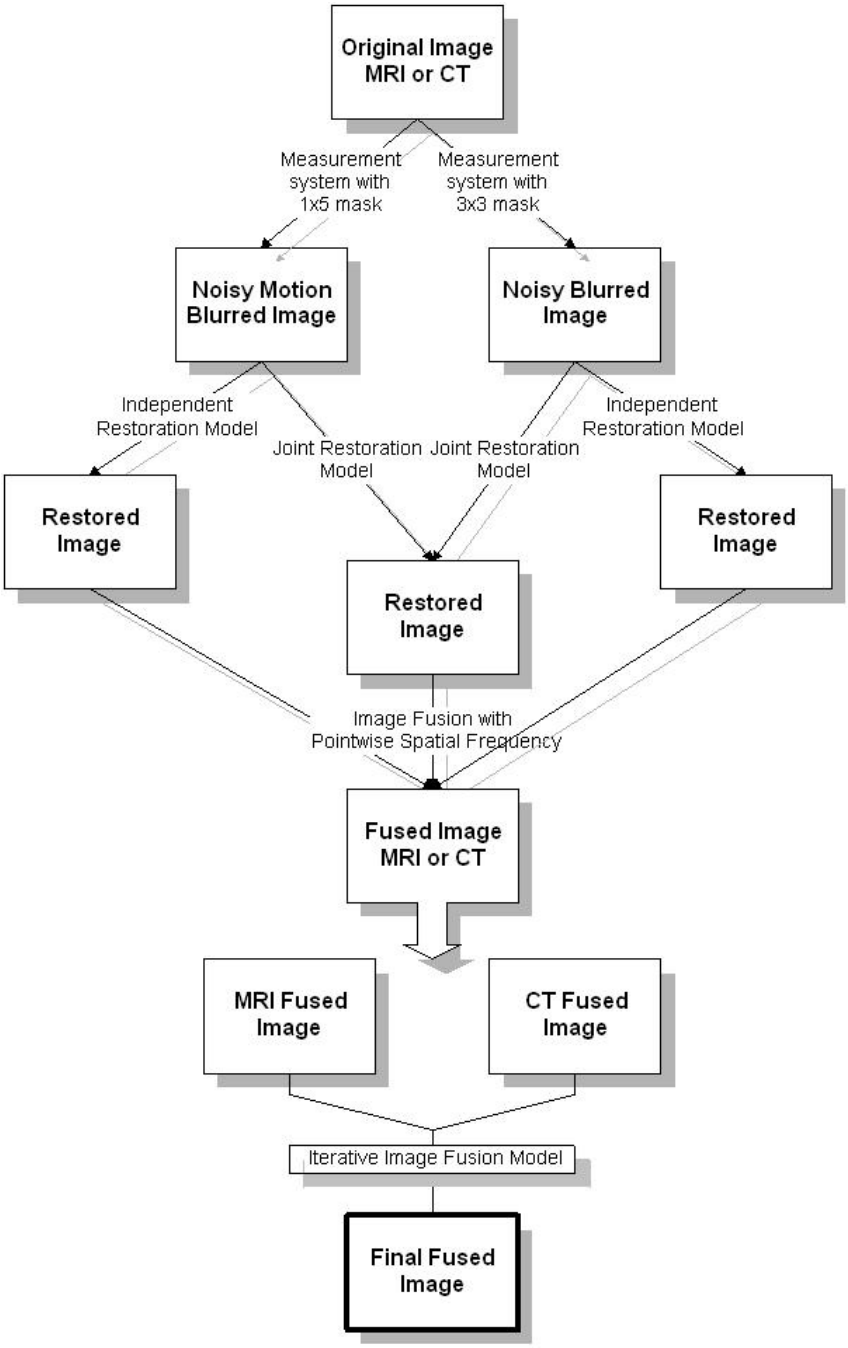


Figure 1:1 Thesis block diagram algorithm

Before we continue with the thesis presentation, it is essential to do a report in the problems that we have faced up. First, the duration of the thesis application has last for approximately one year. Through this year, lots of problems were produced and hopefully for us, have solved! The image fusion field is an open area of research

where lots of experimentations are possible. For that reason, we have speculated with different literature work based on image fusion, trying to upgrade the thesis image fusion procedure. Another problem was the design of our code. Note that the thesis code, written in C with the powerful tool of Microsoft Visual Studio and 100% by our own in the laboratories of the university, had to change a lot during the research in order to be reliable and fast.

2 Image restoration

2.1 Introduction to image restoration

2.1.1 General description

Restoration of single channel images is a relatively mature field, started back in the early 1970's, when digital computers were first making headway into the research laboratories. Since the publication of *Digital Image Restoration* by Hunt and Andrews in 1977 **[1]**, restoration has been widely researched. There are basically two types of approaches to image restoration. Both approaches assume a degradation model as follows:

$$y = h * x + n \tag{1}$$

Matrices **y**, **x** are the observed and original images, **n** is the additive random noise, and **h** is the degradation.

2.1.2 Degradation causes

The degradations may have many causes, but two types of degradations are often dominant: blurring and noise.

Blurring is a form of bandwidth reduction of the image due to imperfect image formation process. It can be caused by relative motion between the camera and the original scene, or by optical system, which is out of focus. When aerial photographs are produced for remote sensing purposes, blurs are introduced by atmospheric turbulence, aberrations in the optical system, and relative motion between the camera and the ground.

In addition to these blurring effects, the recorded image is also corrupted by **noises**. This may be introduced by the transmission medium (e.g. a noisy channel), the recording medium (e.g. film grain noise), measurement errors due to the limited accuracy of the recording system and quantization of the data for digital storage.

The field of image restoration (some times referred to as image de-blurring) is concerned with the reconstruction or estimation of the uncorrupted image from a distorted and noisy one.

2.1.3 Estimation of the degradation

The problem of image restoration can now be mathematically stated as given **y**, find a best estimate of **x**, according to some optimality criterion. Restoration algorithms differ with respect to the amount of knowledge that is assumed known a priori. The simplest algorithms assume **h** is known exactly, as well as the statistics of the noise, **n**. If the degradation is not known, it can be estimated by a couple of different methods.

If the image is an astronomical image, then the blur can be calculated very easily by pointing the telescope at a point source. The observed image will not be a point, due to the degradation. Instead, the point will be blurry, which can then be used to mathematically estimate the degradation. In more general cases, different algorithms have been proposed in the literature to estimate the degradation (or, Point Spread Function, PSF).

2.1.4 Related search for image restoration

The two general approaches for image restoration are stochastic (random) and deterministic (non-random) methods. Stochastic approaches assume that the original image is a realization of a random field, usually Gauss-Markov. Bayesian estimation, Maximum Entropy (ME), Maximum Likelihood (ML), and Maximum A Posteriori (MAP) approaches are specific types of stochastic methods. The other method, deterministic approaches, frequently uses constraints to make the restoration problem tractable. Popular methods are Projection Onto Convex Sets (POCS) and regularization.

2.1.5 Evaluation criterion

In most applications we assume that either a human interpreter or another image processing algorithm, such as segmentation or analysis procedure, will be using the image restoration result. It would be ideal if we had available an evaluation criterion or performance measure that corresponds to either human visual system or the requirements of the subsequent processing steps. Unfortunately, such criteria are hardly available, and the ones that are known are virtually impossible to use within the context of image restoration. Therefore restored images are usually subjectively evaluated by man or it could also be done by using signal-to-noise ratio improvement measures.

2.1.6 Thesis parameters and input data

The processing results of the proposed image restoration methods are based on the quadratic constrained restoration approaches. Indeed, the degradation in our research is known by simulating each time the measurement system that produced the observed image from the original one. According to this, the first measurement system is assumed to have a 3x3 linear blurring response matrix and the second measurement system a 1x5 motion blurred matrix, which are described below. Moreover, the images to be restored are biomedical images from magnetic resonance imaging (MRI) and computed tomography (CT) systems.

Below, we are going to analyze specifically the methods that we have used to restore the input images. Thus, a comparison between the effectiveness of each restoration model is documented. Any given image can be represented as a two-dimensional set of pixels.

2.2 Restoration using gradient estimation

2.2.1 General description

Assume that we are observing an unknown image $\mathbf{x}$ through two different measurement systems and obtain two sets of data $\mathbf{y_1}$ and $\mathbf{y_2}$. Both $\mathbf{y_1}$ and $\mathbf{y_2}$ are

correspond to the same source image but are produced from different measurement systems with linear response matrices $\mathbf{h_1}$ and $\mathbf{h_2}$ respectively. The linear response matrix $\mathbf{h_1}$ corresponds to the 3x3 blurring mask with matrix values shown in equation (2), whereas the linear response matrix $\mathbf{h_2}$ is the 1x5 motion blurred mask with calculated matrix values shown in the below equation (3).

$$blurred\_mask = \begin{bmatrix} \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \end{bmatrix} \tag{2}$$

$$motion\_blurred\_mask = \begin{bmatrix} \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \end{bmatrix} \tag{3}$$

Indeed, assume that we have linear relations both between $\mathbf{y_1}$ and $\mathbf{x}$ and between $\mathbf{y_2}$ and $\mathbf{x}$, and $\mathbf{e_1}$ and $\mathbf{e_2}$ of the measurement systems are their respective errors (modelling, discretization and measurement errors). In order to be quite strict with our method, we have decided the measurement system error to be 15db white Gaussian noise (WGN).

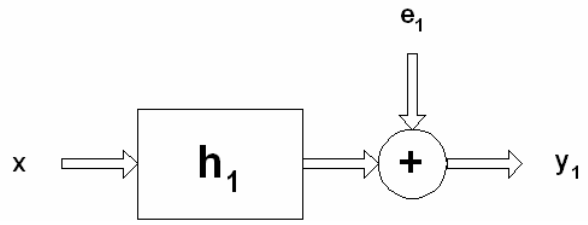


Figure 2:1 the first measurement system with degradation h_1

$$y_1 = h_1 * x + e_1 \tag{4}$$

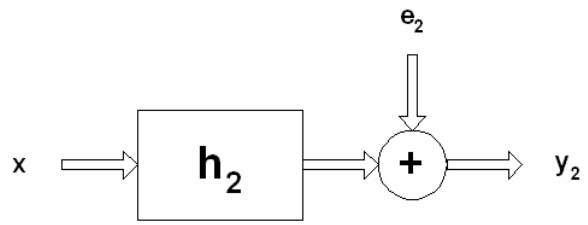


Figure 2:2 the second measurement system with degradation h_2

$$y_2 = h_2 * x + e_2 \tag{5}$$

2.2.2 Input data from the MR and CT systems

Our main purpose is to restore the unknown image **x** by the given images $\mathbf{y_1}$ and $\mathbf{y_2}$. The unknown image **x** is represented by the original image in Figure 2:3 in the case of a magnetic resonance image, and with image in Figure 2:5 in the case of a computed tomography (CT) system. Both MR and CT images are 256 grayscale bitmap images. Besides the figure of each image, this thesis represents its histogram, given by the Adobe Photoshop software tool.

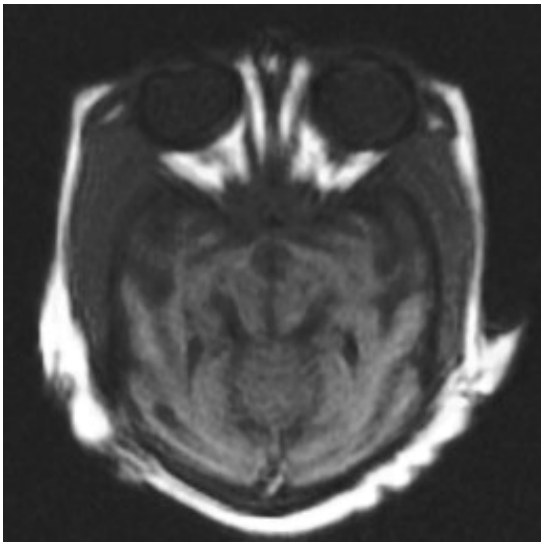


Figure 2:3 Original magnetic resonance image

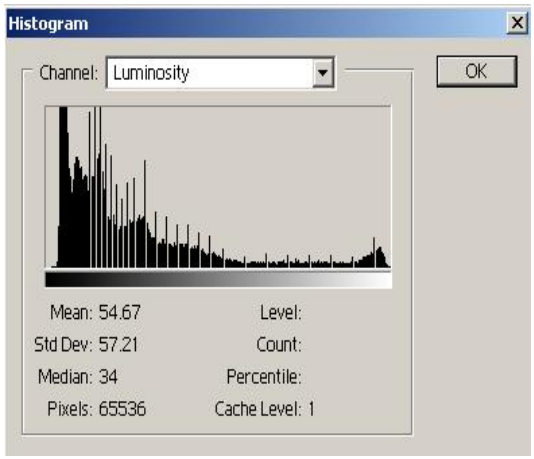


Figure 2:4 Histogram of the original magnetic resonance image

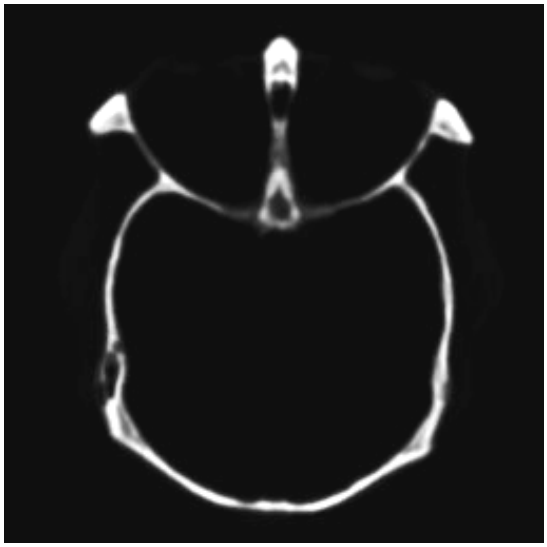


Figure 2:5 Original computed tomography image

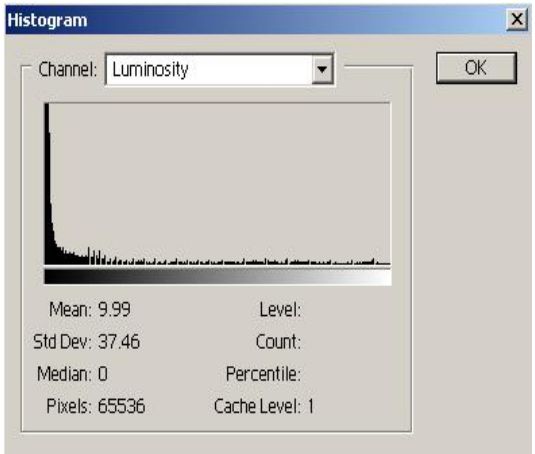


Figure 2:6 Histogram of the original computed tomography image

2.2.3 Restoration models

2.2.3.1 General description of the models

The two general approaches for image restoration, performed in this thesis, are the joint and the independent method based on the quadratic constrained restoration method. Before the description of the constrained restoration model, a brief description of the unconstrained restoration method, referenced in **[12]**, is essential. The noise term in the degradation model is:

$$n = y - h * x \tag{6}$$

In the absence of any knowledge about **n**, a meaningful criterion function is to seek a $\hat{x}$ such that $h * \hat{x}$ approximates **y** in a least squares sense by assuming that the norm of the noise term is as small as possible. In other words, we want to find an $\hat{x}$ such that

$$\|n\|^2 = \|y - h * x\|^2 \tag{7}$$

is minimum. Equation (7) allows the equivalent view of this problem as one minimizing the criterion in equation (8) with respect to $\hat{x}$.

$$J(\hat{x}) = \|y - h * \hat{x}\|^2 \tag{8}$$

Aside from the requirement that it minimizes equation (8), $\hat{x}$ is not constrained in any other way. Minimization of equation (8) is straightforward. We simply find the derivative of function $J(.)$ with respect to $\hat{x}$ and set the result equal to the zero matrix.

$$\frac{\partial J(\hat{x})}{\partial \hat{x}} = 0 \tag{9}$$

One way to alleviate sensitivity of the result to errors in **h** mask, is to base optimality of restoration on a measure of smoothness, such as the second derivative of the image. We will approximate the second derivative (Laplacian) by a matrix **c**. Indeed, we will first formulate the constrained restoration problem and obtain its solution in terms of a general matrix **c**. In constrained image restoration, we choose $\hat{x}$

to minimize $\|c * \hat{x}\|^2$, subject to the constraint $\|n\|^2 = \|y - h * x\|^2$ from equation (7). The augmented objective function $J(\hat{x})$ is then given in equation (10)

$$J(\hat{x}) = \|c * \hat{x}\|^2 + v \cdot \left(\|y - h * \hat{x}\|^2 - \|n\|^2 \right) \tag{10}$$

where **v** is a Lagrange multiplier. Minimization of equation (10) is straightforward. We simply find the derivative of the augmented function $J(.)$ with respect to $\hat{x}$ and set the result equal to the zero matrix.

In the following chapters we are going to describe further the quadratic constrained restoration method, and how we can approximate the derivative of function $J(.)$. In the joint restoration application, assume that the original image **x** is restored at the same time by both images $\mathbf{y_1}$ and $\mathbf{y_2}$ given by the two measurement systems. The known images $\mathbf{y_1}$ and $\mathbf{y_2}$ are participating equally in the same equation. After the restoration process, the restored image has information by both images $\mathbf{y_1}$ and $\mathbf{y_2}$.

On the contrary, in the independent restoration application, assume that the original image **x** is restored separately by the given images $\mathbf{y_1}$ and $\mathbf{y_2}$. The output images from the first and the second measurement system are restored individually. Below, we implement the restoration of the magnetic resonance image and the computed tomography image.

2.2.3.2 Mathematical models

In the next sub-chapter, mathematical equations and rules lead to a deeper and more analytic view of how the joint and the independent constrained restoration models can contribute positively to the restoration issue.

2.2.3.2.1 Mathematical model for joint quadratic constrained restoration method

The model that we have implemented is based on the derivative of the function that includes the unknown image, the degradation of the measurement system, and the input image that is taken by the measurement system. The equation for this model is

given in (11). The equation takes as inputs both outputs of the two measurement systems and restores the unknown image from the two input images.

Note that $\mathbf{y_1}$ is the known noisy blurred image given by the first measurement system with the 3x3 mask, $\mathbf{y_2}$ is the known noisy motion blurred image given by the second measurement system with the 1x5 mask, $\mathbf{h_1}$ and $\mathbf{h_2}$ represent, respectively, the 3x3 degradation mask and the 1x5 degradation mask convolved with the unknown image, and finally, **c** mask is the Laplacian mask which de-blurs the image. Note that the Laplacian mask functions as a 3x3 high-pass filter.

$$J(x) = \alpha \cdot \left\| y_1 - h_1 * x \right\|^2 + \alpha \cdot \left\| y_2 - h_2 * x \right\|^2 + \beta \cdot \left\| c * x \right\|^2 \tag{11}$$

In order to avoid the complex calculations between the degradation matrices, the Laplacian mask and the image to be restored, we have transformed our equation from time domain to the frequency domain due to the mathematical technique of discrete Fourier transform (DFT). The convolution between the matrices is converted into a pixel-by-pixel multiplication between the pixels of the two-dimensional matrices. The transformed form is shown in equation (12). Indeed, assume that we have worked separately with the real and the imaginary parts of the complex numbers.

$$J(X) = \alpha \cdot \left\| \mathrm{Y}_1 - H_1 \cdot X \right\|^2 + \alpha \cdot \left\| \mathrm{Y}_2 - H_2 \cdot X \right\|^2 + \beta \cdot \left\| C \cdot X \right\|^2 \tag{12}$$

Our common de-noising and de-blurring technique is to set the derivative of equation (12) equal to zero as, $\nabla_X J\left(X\right) = 0$. The restored image $\hat{x}$ is then:

$$\hat{X} = \left(\alpha \cdot H_1^t \cdot H_1 + \alpha \cdot H_2^t \cdot H_2 + \beta \cdot C^t \cdot C\right)^{-1} \left(\alpha \cdot H_1{}^t \cdot Y_1 + \alpha \cdot H_2{}^t \cdot Y_2\right) \tag{13}$$

or

$$\hat{X}\left(k,l\right) = \frac{a \cdot H_1^*\left(k,l\right) \cdot Y_1\left(k,l\right) + a \cdot H_2^*\left(k,l\right) \cdot Y_2\left(k,l\right)}{a \cdot \left|H_1\left(k,l\right)\right|^2 + a \cdot \left|H_2\left(k,l\right)\right|^2 + \beta \cdot \left|C\left(k,l\right)\right|^2} \tag{14}$$

Note that matrices $H_1^*\left(k,l\right)$ and $H_2^*\left(k,l\right)$ are, respectively, the complex conjugate matrices $H_1\left(k,l\right)$ and $H_2\left(k,l\right)$, and variables k, l are the pointers of the Height and the Width of the two-dimensional matrix in the DFT domain. Details about the step-by-step calculations for the estimation of matrix $\hat{x}$, can be seen in Appendix B. Indeed, the weights α and β control the relative importance of the terms.

2.2.3.2.2 Mathematical model for independent quadratic constrained restoration method

The equation for the independent constrained restoration model is given in (15). Note that the following equation takes as input separately the output of the first and the second measurement system and restores the unknown image **x**.

Note that **y** is the known image given by the measurement system, **h** represents the degradation mask of the measurement system convolved with the unknown image **x**, and finally, **c** mask is the Laplacian mask which de-blurs the image.

$$J(x) = \|y - h * x\|^2 + \beta \cdot \|c * x\|^2 \tag{15}$$

According to chapter 2.2.3.2.1, the unknown image **x** is estimated by setting the derivative of function $J(X)$ equal to zero. The gradient of function (15) is:

$$\nabla_X J(X) = -2 \cdot H^t \cdot Y + 2 \cdot H^t \cdot H \cdot X + 2 \cdot \beta \cdot C^t \cdot C \cdot X \tag{16}$$

The restored image is now given by the following equation:

$$\hat{X} = \left(H^t \cdot H + \beta \cdot C^t \cdot C\right)^{-1} \left(H_1^t \cdot Y\right) \tag{17}$$

or

$$\hat{X}(k,l) = \frac{H^*(k,l) \cdot Y(k,l)}{\left|H(k,l)\right|^2 + \beta \cdot \left|C(k,l)\right|^2} \tag{18}$$

Note that matrix $H^*(k,l)$ is the complex conjugate of matrix $H(k,l)$, and variables k, l are the pointers of the Height and the Width of the two-dimensional matrix in the DFT domain.

In the following chapters, we are going to simulate this function and to restore the unknown Image **x**, experimenting with the parameter **β**.

2.2.4 Restoration of the magnetic resonance image

2.2.4.1 Input data images

Below, we represent the output MR images from the two measurement systems. In Figure 2:7 and Figure 2:9 we represent, respectively, the output image from the measurement system with degradation the 3x3 blurring mask, and the 1x5 motion blurring mask. The percentage of the deformation that the measurement systems have caused to the original image can be estimated through the mean square error (MSE) coefficient. The mathematical equation that obtained the mean square error of an image, is given in (19).

$$\text{Mean Square Error} = \sqrt{\frac{\sum_{i=1}^{M}\sum_{j=1}^{N}\left(x_{i,j} - \hat{x}_{i,j}\right)^2}{M \cdot N}} \tag{19}$$

Note that $x_{i,j}$ is the pixel of the original image, $\hat{x}_{i,j}$ is the pixel of the examined image, and M, N are respectively the Height and Width of the MXN matrices x and $\hat{x}$. The mean square error is not a criterion of decision. This coefficient is a measure that obtains the efficiency of each method in the whole procedure and compares the resulted images from the following models.

The MSE of an image is a criterion that we can not implement in the experimental part of this thesis, because we don't have any knowledge about the original image. Hopefully, the image processing literature provides effective methods that ensure the correct decision of the parameters. In this case, our attention is to perform an effective model for image restoration and even more, a powerful model for image fusion.

Nevertheless, in the below tables, we can also see an approximate method of decision that is based on the difference between the affected matrix **y** and the restored matrix $\hat{x}$, convolved with the degradation mask **h**. The equation that gives this coefficient is shown below, where the norm is the energy norm represented in equation (21). Variable N is the number of the elements that the vector $\underline{q}$ has, and j is the pointer of vector q.

$$approximate\_difference = \left\| y - h * \hat{x} \right\|^2 \tag{20}$$

$$\left\| q \right\|^2 = \frac{1}{N} \cdot \sum_j q_j^2 \tag{21}$$

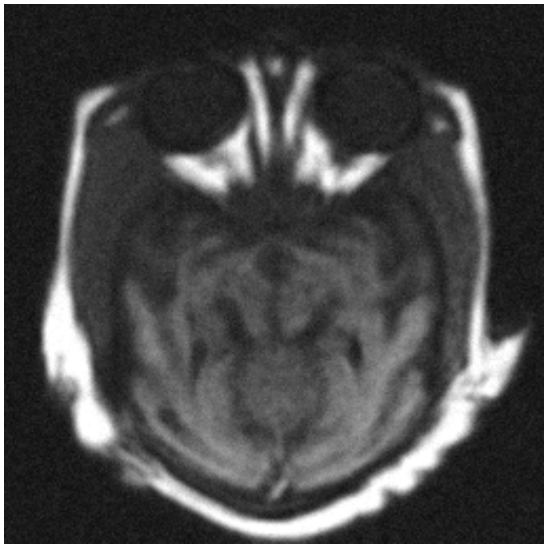


Figure 2:7 Noisy blurred MR image by the 3x3 mask

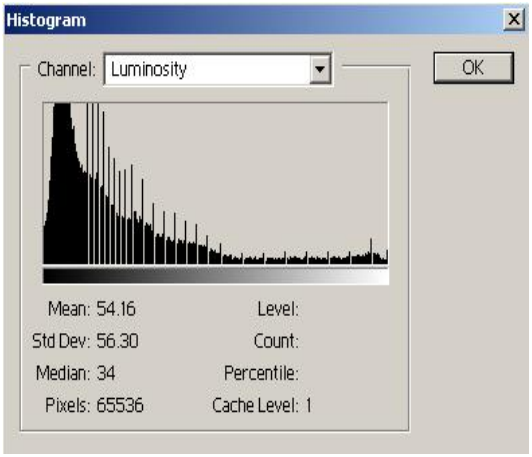


Figure 2:8 Histogram of the noisy blurred MR image by the 3x3 mask

The mean square error (σ) of the noisy blurred image in Figure 2:7 is:

$$\sigma = 15.092118 \tag{22}$$

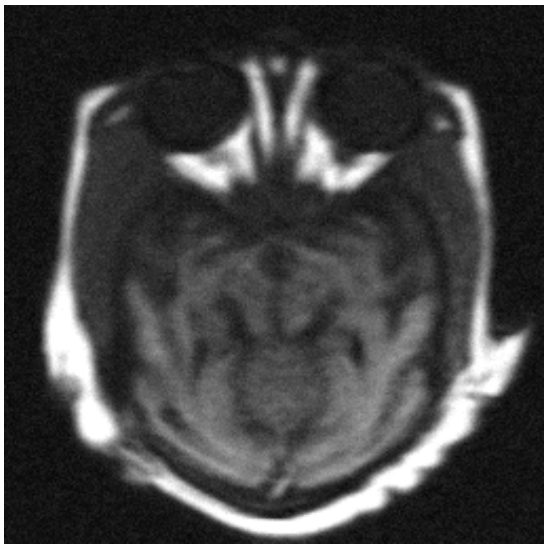


Figure 2:9 Noisy motion blurred MR image by the 1x5 mask

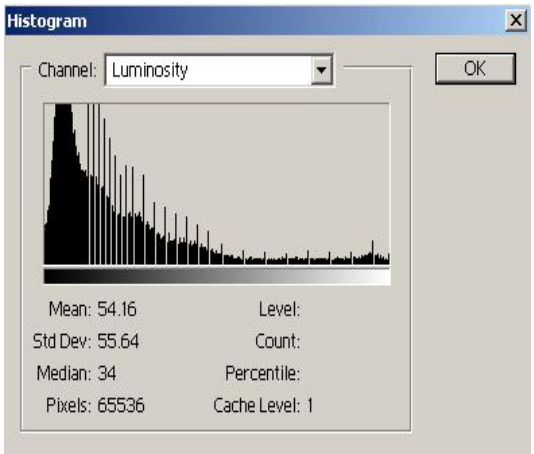


Figure 2:10 Histogram of the noisy motion blurred MR image by the 1x5 mask

The mean square error (σ) of the noisy motion blurred image in Figure 2:9 is:

$$\sigma = 20.097562 \tag{23}$$

2.2.4.2 Independent method for the noisy blurred MR image

According to the model that we have described analytically in chapter 2.2.3.2.2, the equation that performs the restoration of the original image from the noisy blurred magnetic resonance image is the following.

$$J(x) = \left\| y - h * x \right\|^2 + 0.117 \cdot \left\| c * x \right\|^2 \tag{24}$$

Setting the derivative of equation (24) equal to zero, the restored image $\hat{x}$ is given by the following equation:

$$\hat{X} = \left(0.117 \cdot C^t \cdot C + H^t \cdot H\right)^{-1} \cdot \left(H^t \cdot Y\right) \tag{25}$$

Experimenting, through the simulations, with the variable (α) that multiplies the second term in equation (24), we can select the value that maintain the energy norm $\left\| y - h * \hat{x} \right\|^2$ in the lowest value. The results of the simulations are obtained through the following table and chart. Note, that a good range for variable a, is between values 0.1 and 0.2.

Variable α	Mean square error	Approximate difference
a = 0.01	σ = 6.001026	5,0934
a = 0.05	σ = 3.924991	5,0912
a = 0.09	σ = 3.679927	5,1298
a = 0.1	σ = 3.662833	5,1344
a = 0.11	σ = 3.654423	5,1357
a = 0.115	σ = 3.652802	5,1487
a = 0.117	σ = 3.652579	5,1531
a = 0.12	σ = 3.652613	5,1736
a = 0.123	σ = 3.653120	5,2539
a = 0.125	σ = 3.653697	5,3454
a = 0.15	σ = 3.673478	5,4521
a = 0.2	σ = 3.754181	5,4767
a = 0.3	σ = 3.968198	5,5543
a = 0.4	σ = 4.188303	5,7676
a = 0.5	σ = 4.396180	5,8016
a = 0.6	σ = 4.589385	5,8408

Table 2:1 Mean square error for noisy blurred magnetic resonance image restoration

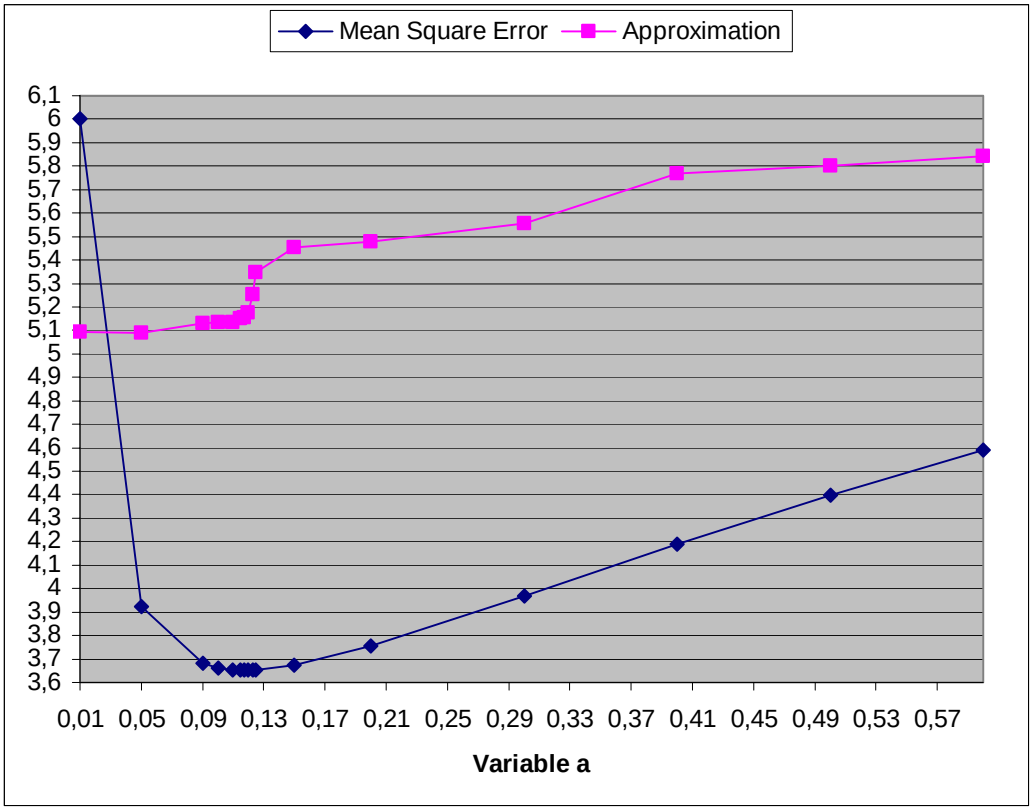


Table 2:2 Chart of noisy blurred magnetic resonance image restoration

Performing the independent restoration method, the resulted image as well as its histogram are the following:

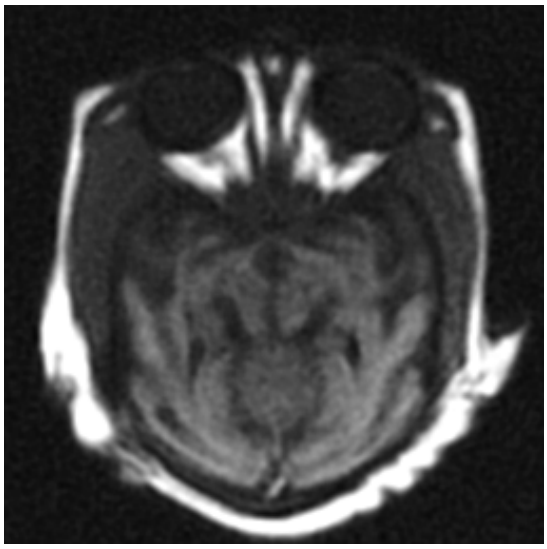


Figure 2:11 Restored MR image by the noisy blurred image

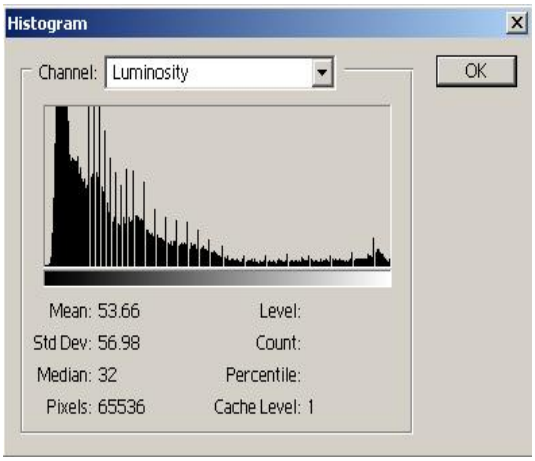


Figure 2:12 Histogram of the restored MR image by the noisy blurred image

A comparison between the output, from the measurement system, affected image and the restored image, is obtained through the mean square error (σ). The MSE of the noisy blurred image is $\sigma = 15.092118$. The mean square error (σ) of the restored MR image in Figure 2:11 is:

$$\sigma = 3.652579 \tag{26}$$

Besides the MSE comparison, this thesis presents the image that obtained by the pixel-by-pixel difference between the original MR image and the restored MR image. Note that, in this thesis all the images that correspond to the pixel-by-pixel differences between images are first inverted in order to be visible by the reader.

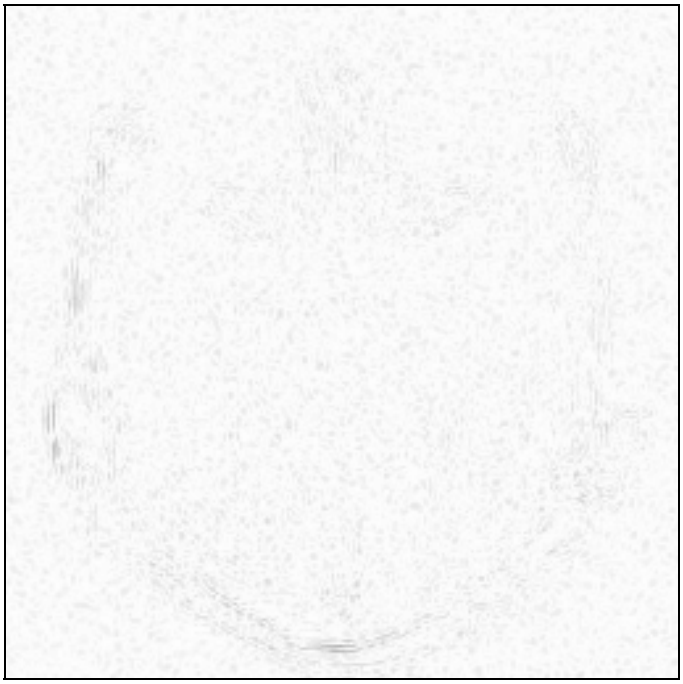


Figure 2:13 difference between the restored MR image by the noisy blurred image and the original MR image

2.2.4.3 Independent method for the noisy motion blurred MR image

The equation that performs the restoration of the original image from the noisy motion blurred magnetic resonance image is:

$$J(x) = \|y - h * x\|^2 + 0.665 \cdot \|c * x\|^2 \tag{27}$$

Setting the derivative of equation (27) equal to zero, the restored image $\hat{x}$ is given by the following equation:

$$\hat{X} = \left(0.665 \cdot C^t \cdot C + H^t \cdot H\right)^{-1} \cdot \left(H^t \cdot Y\right) \tag{28}$$

The results of the simulations are obtained through the following table and chart. Note, that a good range for variable a, is between values 0.6 and 0.75.

Variable α	**Mean square error**	**Approximate difference**
a = 0.1	σ = 40.769902	31.0157
a = 0.2	σ = 10.288084	7.4190
a = 0.3	σ = 8.083379	6.8791
a = 0.4	σ = 7.537364	6.8135
a = 0.5	σ = 7.350917	6.8348
a = 0.6	σ = 7.286261	6.8450
a = 0.65	σ = 7.277273	6.8632
a = 0.66	σ = 7.276952	6.8876
a = 0.665	σ = 7.276938	6.8813
a = 0.67	σ = 7.277057	6.8845
a = 0.675	σ = 7.277285	6.8958
a = 0.7	σ = 7.279807	6.9056
a = 0.75	σ = 7.290672	6.9121
a = 0.8	σ = 7.308391	6.9628
a = 0.9	σ = 7.357633	7.0063
a = 1	σ = 7.419354	7.1293

Table 2:3 Mean square error for noisy motion blurred magnetic resonance image restoration

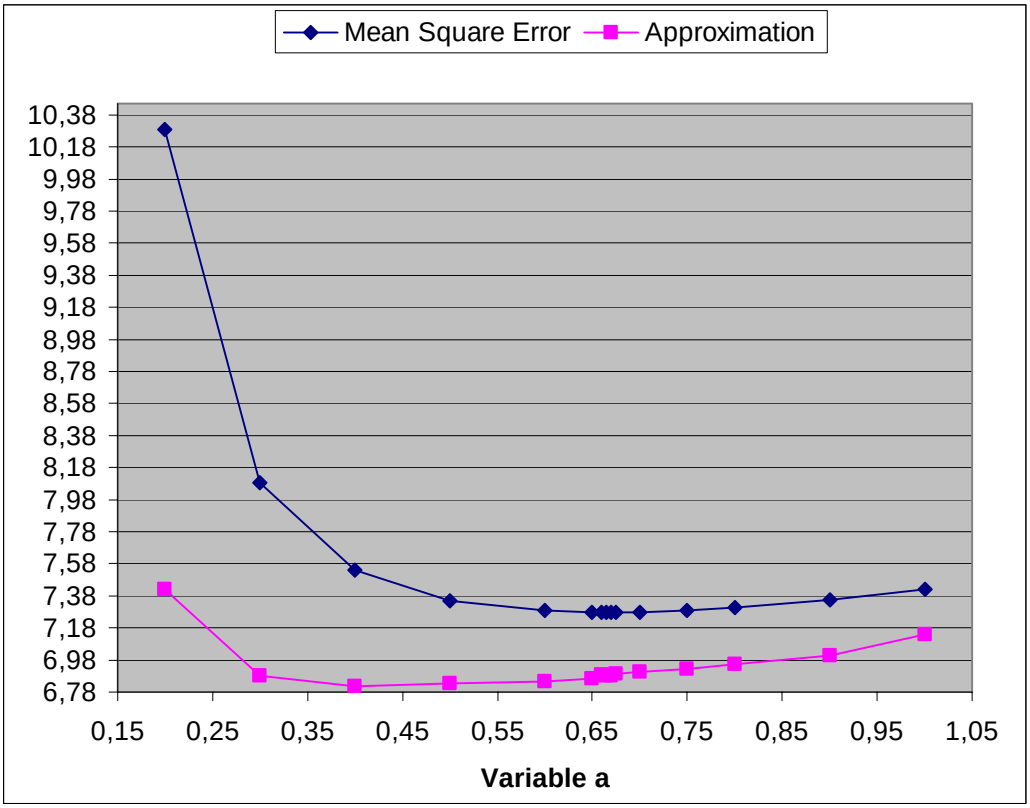


Table 2:4 chart of noisy motion blurred magnetic resonance image restoration

Performing the independent restoration method, the resulted image as well as its histogram are the following:

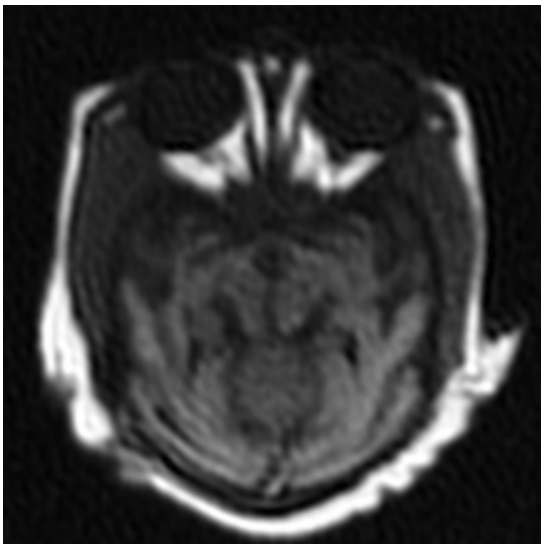


Figure 2:14 Restored MR image by the noisy motion blurred image

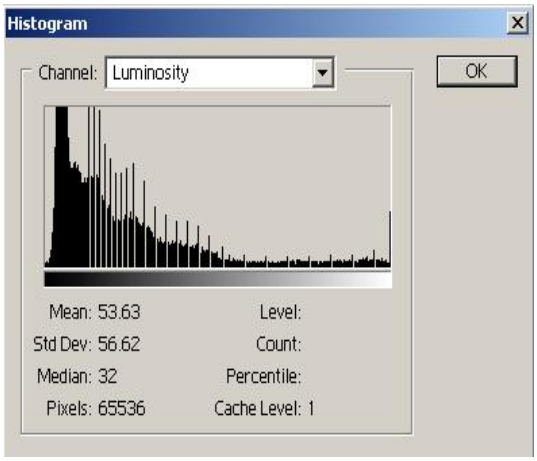


Figure 2:15 Histogram of the restored magnetic resonance image by the noisy motion blurred image

The mean square error (σ) of the noisy blurred image is $\sigma = 20.097562$. The MSE (σ) of the restored magnetic resonance image in Figure 2:14 is:

$$\sigma = 7.276938 \tag{29}$$

The following image corresponds to the pixel-by-pixel difference between the original magnetic resonance image and the restored magnetic resonance image.

Figure 2:16 difference between the restored MR image by the noisy motion blurred image and the original image

2.2.4.4 Joint method for the degraded MR images

We have described analytically the joint constrained restoration model above. According to this model, the equation that performs the restoration of the original image from the noisy blurred magnetic resonance image and the noisy motion blurred magnetic resonance image is the following:

$$J(x) = 0.5 \cdot \left\| y_1 - h_1 * x \right\|^2 + 0.5 \cdot \left\| y_2 - h_2 * x \right\|^2 + 0.137 \cdot \left\| c * x \right\|^2 \tag{30}$$

Note that $\mathbf{y_1}$ is the known noisy blurred image, affected by the 3x3 mask, $\mathbf{y_2}$ is the known noisy motion blurred image, $\mathbf{h_1}$ and $\mathbf{h_2}$ represent, respectively, the 3x3 mask and the 1x5 mask and finally, **c** mask is the Laplacian high-pass filter.

Setting the derivative of equation (30) equal to zero, the restored image $\hat{x}$ is given by the following equation:

$$\hat{X} = \left(0.137 \cdot C^t \cdot C + 0.5 \cdot H_1^t \cdot H_1 + 0.5 \cdot H_2^t \cdot H_2\right)^{-1} \cdot \left(0.5 \cdot H_1^t \cdot Y_1 + 0.5 \cdot H_2^t \cdot Y\right) \quad (31)$$

The results of the simulations are obtained through the following table and chart. Note, that a good range for variable a, is between values 0.1 and 0.2.

Variable α	**Mean square error**	**Approximate difference**
a = 0.05	σ = 5.155579	5.5746
a = 0.1	σ = 4.695298	5.4918
a = 0.125	σ = 4.651121	5.5241
a = 0.13	σ = 4.648435	5.5512
a = 0.134	σ = 4.647346	5.5588
a = 0.135	σ = 4.647200	5.5618
a = 0.136	σ = 4.647135	5.5624
a = 0.137	σ = 4.647109	5.5642
a = 0.138	σ = 4.647131	5.5664
a = 0.14	σ = 4.647309	5.5702
a = 0.145	σ = 4.648571	5.5793
a = 0.15	σ = 4.650891	5.5892
a = 0.16	σ = 4.658237	5.6073
a = 0.17	σ = 4.668531	5.6921
a = 0.2	σ = 4.711969	5.7213
a = 0.3	σ = 4.913472	5.8266

Table 2:5 Mean square error for restored MR image by joint restoration

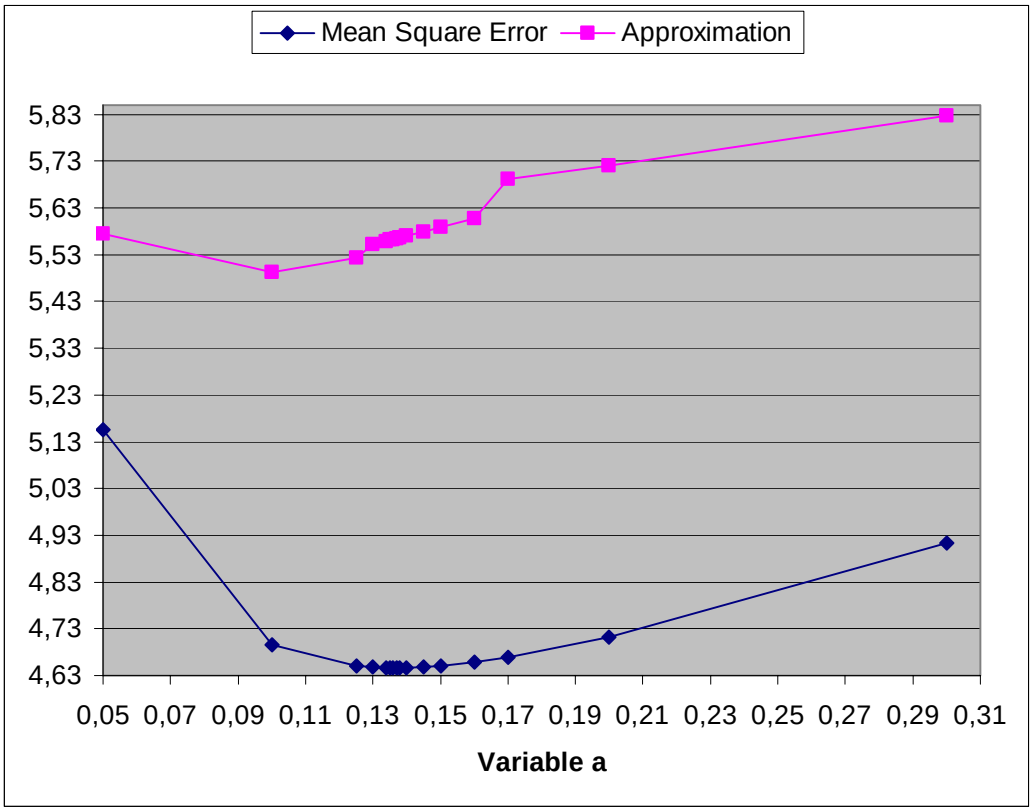


Table 2:6 chart of joint magnetic resonance image rstoration

Performing the independent image restoration method, the resulted image as well as its histogram are the following:

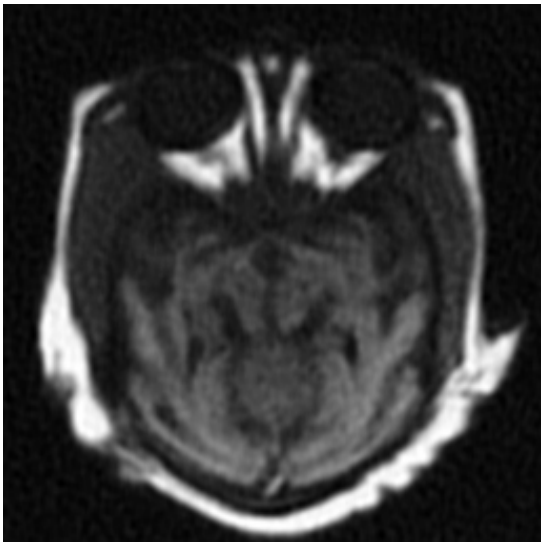


Figure 2:17 Restored magnetic resonance image by the joint restoration

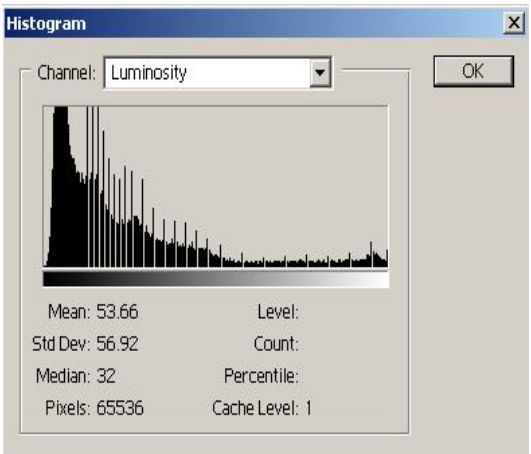


Figure 2:18 Histogram of the restored magnetic resonance image by the joint restoration

The MSE (σ) of the noisy motion blurred image is $\sigma = 20.097562$. The MSE of the noisy blurred image is $\sigma = 15.092118$. Finally, the MSE (σ) of the restored magnetic resonance image in Figure 2:17 is:

$$\sigma = 4.647109 \tag{32}$$

The following image corresponds to the pixel-by-pixel difference between the original magnetic resonance image and the restored magnetic resonance image.

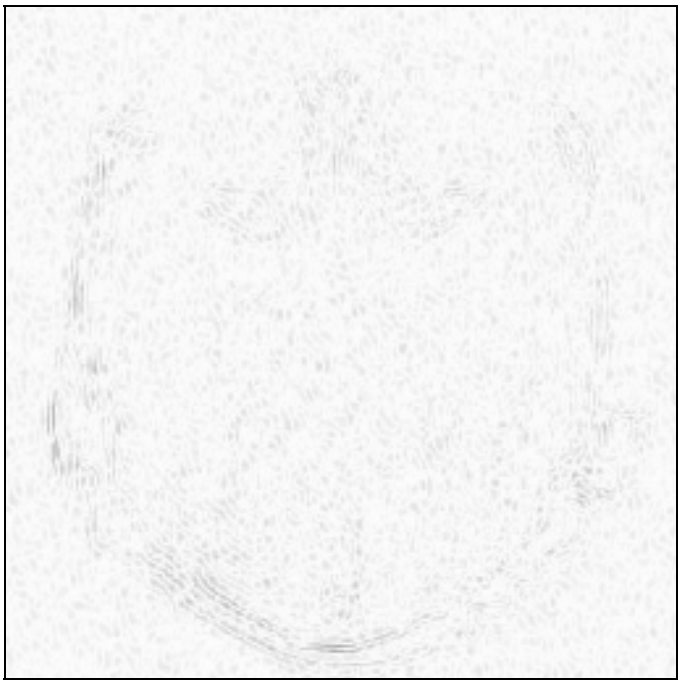


Figure 2:19 difference between the restored MR image by the joint restoration and the original image

2.2.5 Restoration of the computed tomography image

2.2.5.1 Input data images

Below, we represent the two output images from the measurement systems. In Figure 2:20 and Figure 2:22 we represent, respectively, the given images by the measurement system with degradation the 3x3 blurring mask and the 1x5 motion blurring mask. The percentage of the deformation that the measurement systems have caused to the original image can be estimated through the MSE and the approximate difference between the restored and the affected image.

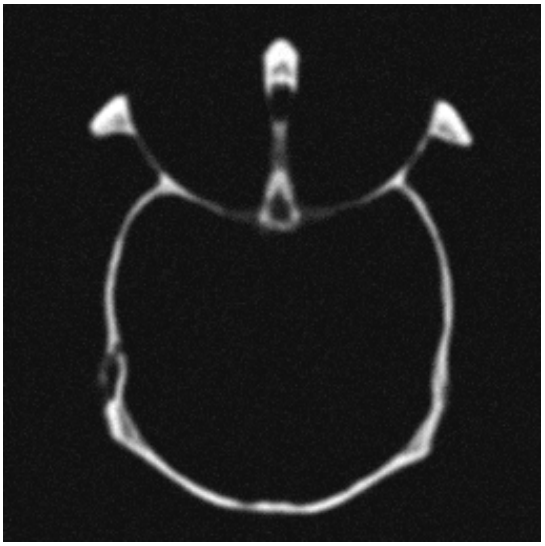


Figure 2:20 Noisy blurred image by the 3x3 mask

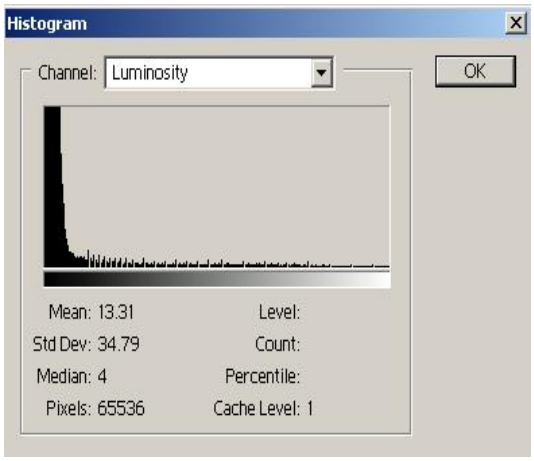


Figure 2:21 Histogram of the noisy blurred image by the 3x3 mask

The mean square error (σ) of the noisy blurred image in Figure 2:20 is:

$$\sigma = 14.905295 \tag{33}$$

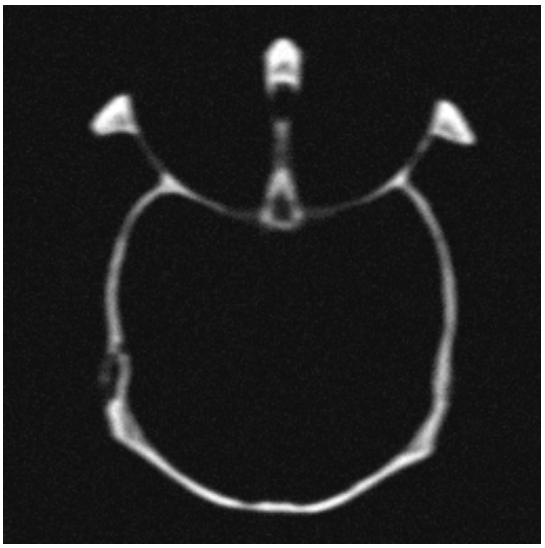


Figure 2:22 Noisy motion blurred image by the 1x5 mask

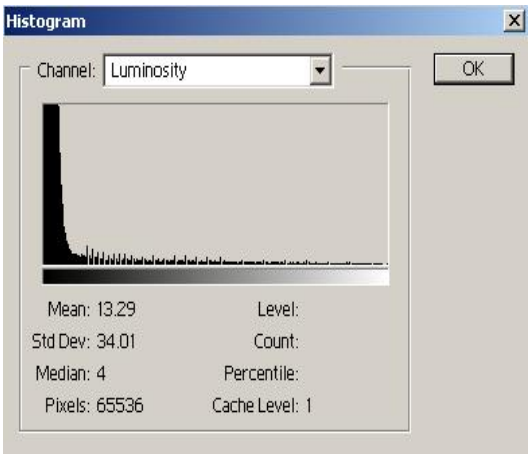


Figure 2:23 Histogram of the noisy motion blurred image by the 1x5 mask

The mean square error (σ) of the noisy motion blurred image in Figure 2:22 is:

$$\sigma = 18.884409 \tag{34}$$

2.2.5.2 Independent method for the noisy blurred CT image

The equation for the independent quadratic constrained restoration method is the following. Note that the known matrix **y** corresponds to the noisy blurred CT image.

$$J(x) = \|y - h * x\|^2 + 0.049 \cdot \|c * x\|^2 \tag{35}$$

Setting the derivative of equation (35) equal to zero, the restored Image $\hat{x}$ is:

$$\hat{X} = \left(0.049 \cdot C^t \cdot C + H^t \cdot H\right)^{-1} \cdot \left(H^t \cdot Y\right) \tag{36}$$

The following table and chart obtain the value of the mean square error of the restored image and the approximate difference due to the alpha variable. Note, that a good range for variable a, is between values 0.03 and 0.1.

Variance α	**Mean square error**	**Approximate difference**
a = 0.01	σ = 5.499821	3.3497
a = 0.02	σ = 4.994994	3.2893
a = 0.03	σ = 4.841860	3.2863
a = 0.04	σ = 4.787582	3.3419
a = 0.045	σ = 4.777989	3.3819
a = 0.047	σ = 4.776279	3.3917
a = 0.048	σ = 4.775833	3.3986
a = 0.049	σ = 4.775587	3.4018
a = 0.05	σ = 4.775610	3.4094
a = 0.051	σ = 4.775809	3.4129
a = 0.052	σ = 4.776244	3.4205
a = 0.06	σ = 4.785520	3.5401
a = 0.1	σ = 4.912756	3.6402
a = 0.2	σ = 5.349675	4.0030
a = 0.3	σ = 5.762711	4.3110

Table 2:7 Mean square error for noisy blurred computed tomography image restoration

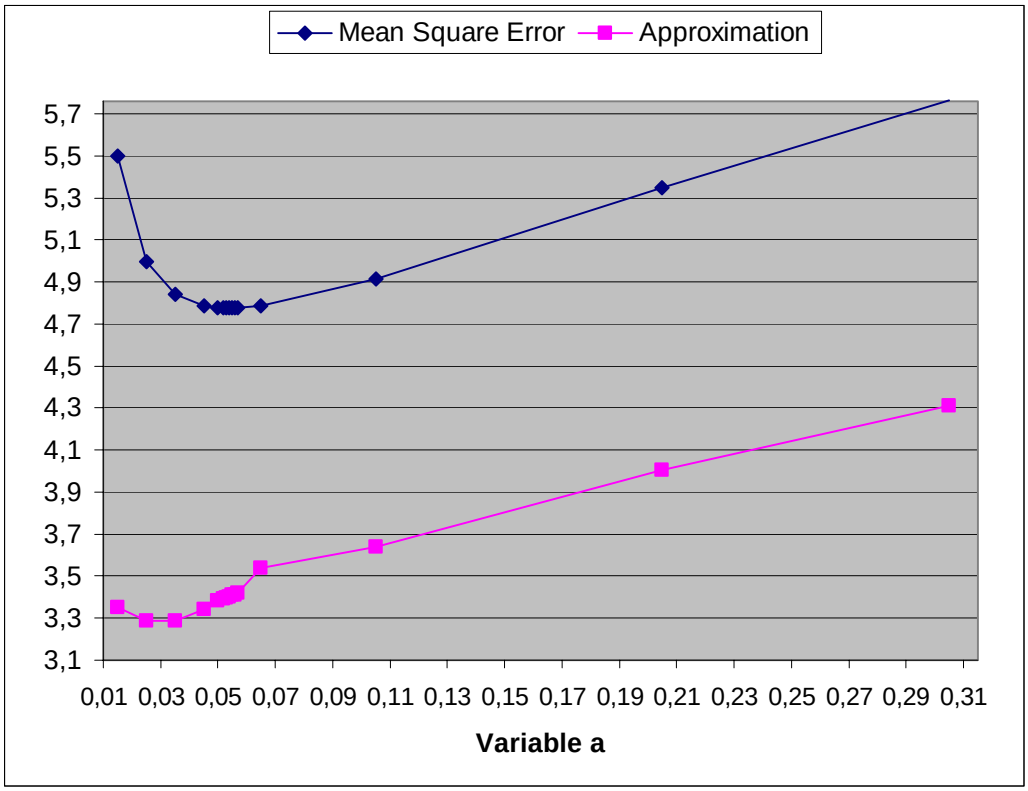


Table 2:8 chart of noisy blurred computed tomography image restoration

Performing the independent image restoration method, the resulted image as well as its histogram are the following:

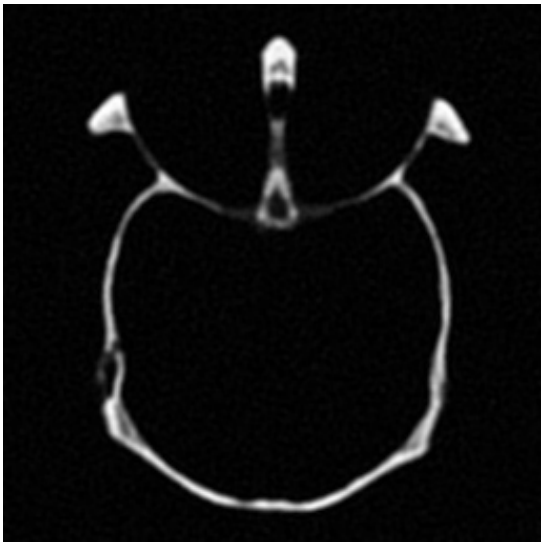


Figure 2:24 Restored CT image by the noisy blurred image

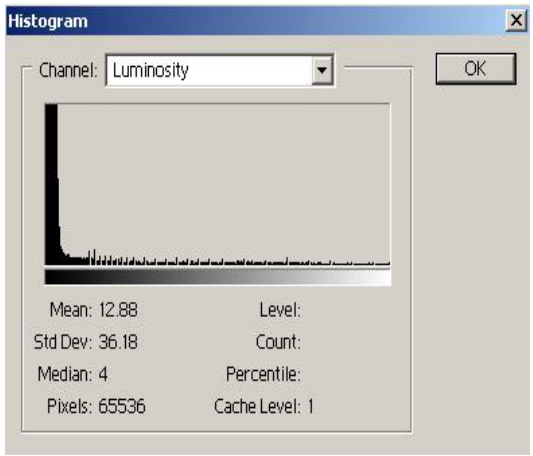


Figure 2:25 Histogram of the restored CT image by the noisy blurred image

The MSE (σ) of the noisy blurred image is $\sigma = 14.905295$. The MSE (σ) of the restored computed tomography image in Figure 2:24 is:

$$\sigma = 4.775588 \tag{37}$$

The image corresponds to the pixel-by-pixel difference between the original computed tomography image and the restored CT image is:

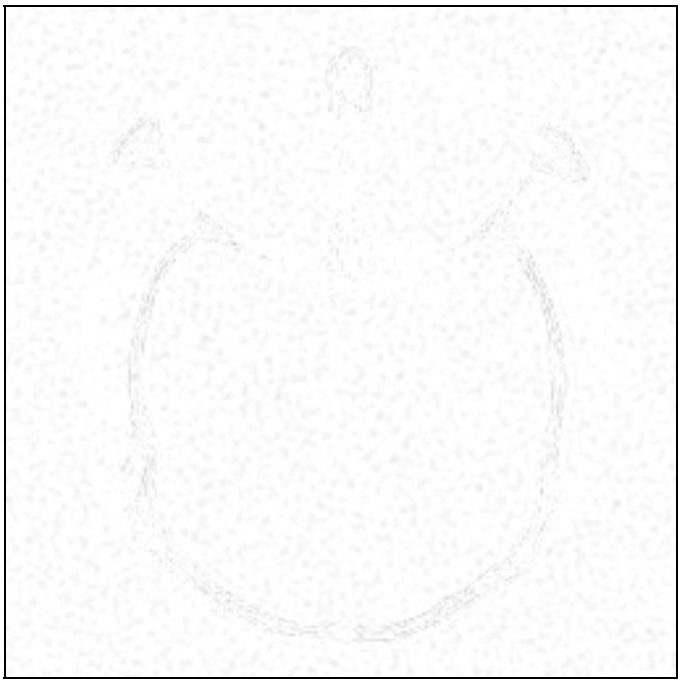


Figure 2:26 difference between the restored CT by the noisy blurred image and the original image

2.2.5.3 Independent method for the noisy motion blurred CT image

The equation for the independent quadratic constrained restoration model, applied to the noisy motion blurred CT image is the following:

$$J(x) = \|y - h * x\|^2 + 0.049 \cdot \|c * x\|^2 \tag{38}$$

Setting the derivative of equation (38) equal to zero the restored image is given by the following equation:

$$\hat{X} = \left(0.049 \cdot C^t \cdot C + H^t \cdot H\right)^{-1} \cdot \left(H^t \cdot Y\right) \tag{39}$$

The following table and chart obtain the value of the mean square error of the restored image and the approximate difference due to the alpha variable. Note, that a good range for variable a, is between values 0.3 and 0.8.

Variance α	**Mean square error**	**Approximate difference**
a = 0.1	σ = 24.138487	17.0564
a = 0.2	σ = 12.357147	8.6360
a = 0.3	σ = 10.717384	7.6595
a = 0.4	σ = 10.501489	7.5892
a = 0.42	σ = 10.494787	7.5695
a = 0.425	σ = 10.494207	7.5581
a = 0.43	σ = 10.494017	7.5782
a = 0.431	σ = 10.494015	7.5601
a = 0.435	σ = 10.494162	7.5626
a = 0.44	σ = 10.494706	7.5671
a = 0.45	σ = 10.496713	7.5692
a = 0.46	σ = 10.499882	7.5812
a = 0.5	σ = 10.522231	7.5994
a = 0.55	σ = 10.565886	7.6241
a = 0.6	σ = 10.621195	7.6864
a = 0.8	σ = 10.888481	7.8990

Table 2:9 Mean square error for noisy motion blurred computed tomography image restoration

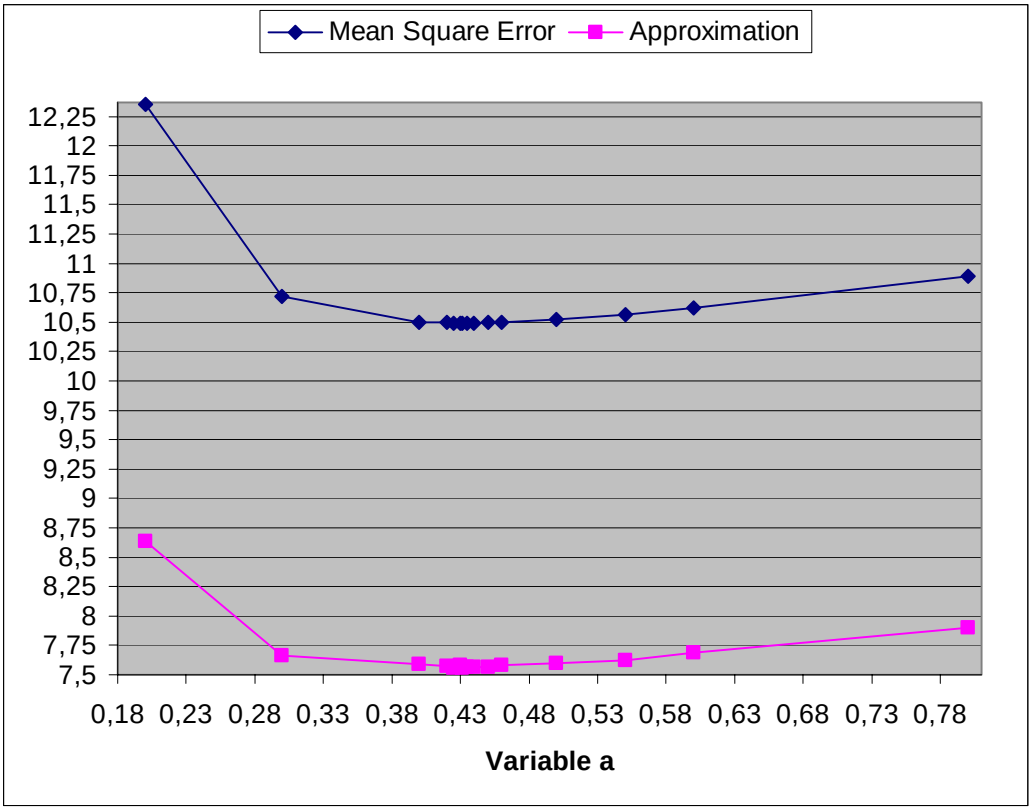


Table 2:10 Chart of noisy motion blurred computed tomography image restoration

Performing the independent image restoration method, the resulted image as well as its histogram are the following:

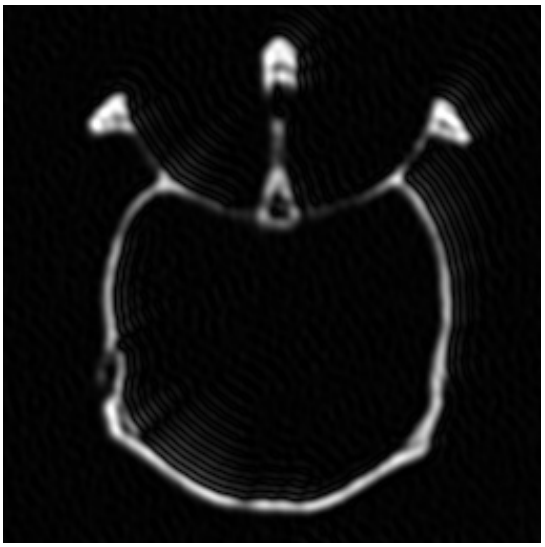


Figure 2:27 Restored CT image by the noisy motion blurred

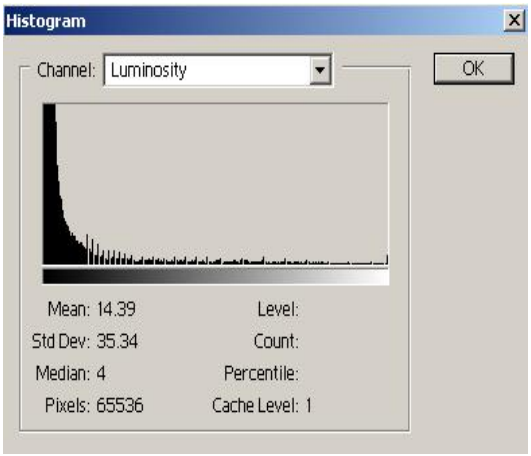


Figure 2:28 Histogram of the restored CT image by the noisy motion blurred image

The MSE (σ) of the noisy motion blurred image is $\sigma = 18.884409$. The mean square error (σ) of the restored computed tomography image in Figure 2:27 is:

$$\sigma = 10.494015 \tag{40}$$

The image corresponds to the pixel-by-pixel difference between the original computed tomography image and the restored computed tomography image is:

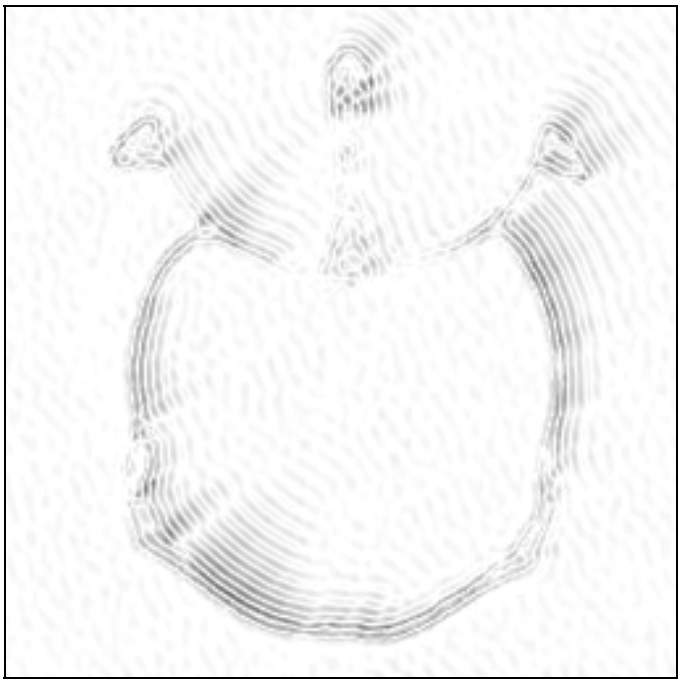


Figure 2:29 difference between the restored CT by the noisy motion blurred and the original image

2.2.5.4 Joint method for the degraded CT images

The joint model, performed to restore the original image from the noisy blurred CT image and the noisy motion blurred CT image is represented in equation (41).

$$J(x) = 0.5 \cdot \left\| y_1 - h_1 * x \right\|^2 + 0.5 \cdot \left\| y_2 - h_2 * x \right\|^2 + 0.048 \cdot \left\| c * x \right\|^2 \tag{41}$$

Note that $\mathbf{y_1}$ is the noisy blurred image affected by the 3x3 mask, $\mathbf{y_2}$ is the known noisy motion blurred image affected by the 1x5 mask, $\mathbf{h_1}$ and $\mathbf{h_2}$ represent, respectively, the 3x3 mask and the 1x5 mask and finally, **c** mask is the Laplacian 3x3 high pass filter.

Setting the derivative of equation (41) equal to zero the restored image is given by the following equation:

$$\hat{X} = \left(0.048 \cdot C^t \cdot C + 0.5 \cdot H_1^t \cdot H_1 + 0.5 \cdot H_2^t \cdot H_2\right)^{-1} \cdot \left(0.5 \cdot H_1^t \cdot Y_1 + 0.5 \cdot H_2^t \cdot Y\right) \quad (42)$$

The MSE of the restored image and the approximate difference are given in the following table. Note, that a good range for variable a, is between values 0.04 and 0.06.

Variance α	**Mean square error**	**Approximation difference**
a = 0.01	σ = 6.793724	3.6823
a = 0.02	σ = 5.833908	3.6774
a = 0.04	σ = 5.460186	3.6897
a = 0.045	σ = 5.445394	3.7190
a = 0.047	σ = 5.443341	3.7347
a = 0.048	σ = 5.443016	3.7406
a = 0.049	σ = 5.443098	3.7479
a = 0.05	σ = 5.443573	3.7652
a = 0.053	σ = 5.447111	3.7734
a = 0.055	σ = 5.451060	3.7889
a = 0.057	σ = 5.456123	3.7987
a = 0.06	σ = 5.463487	3.9231
a = 0.1	σ = 5.691504	4.0522
a = 0.2	σ = 6.353106	4.5395
a = 0.3	σ = 6.901653	4.9237

Table 2:11 Mean square error for restored CT image by joint restoration

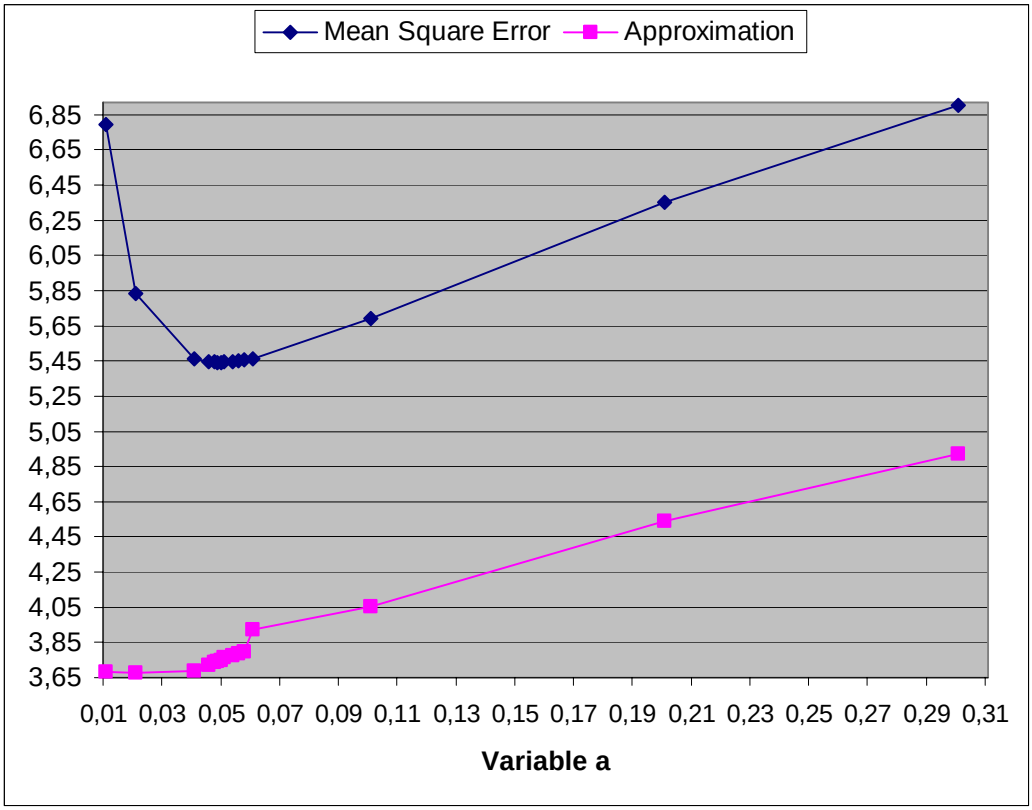


Table 2:12 chart of joint computed tomography image restoration

Performing the independent image restoration method, the resulted image and its histogram are the following:

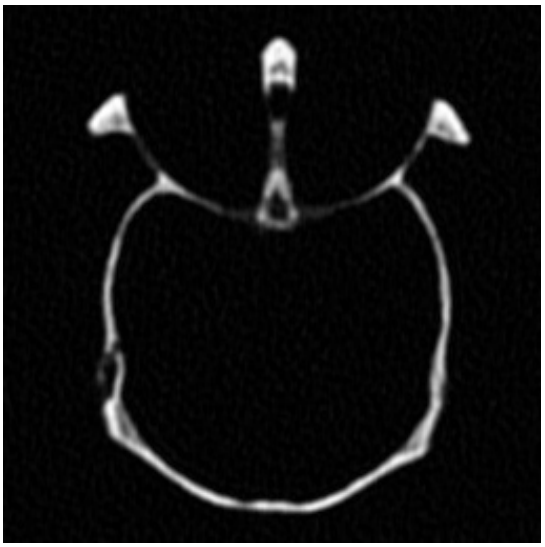


Figure 2:30 Restored computed tomography image by the joint restoration

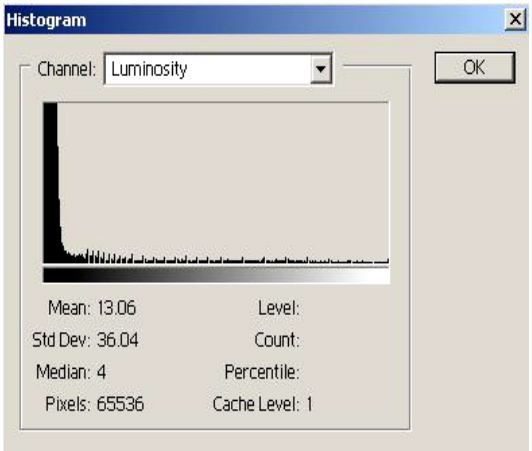


Figure 2:31 Histogram of the restored CT image by the joint restoration

The mean square error (σ) of the noisy motion blurred image is $\sigma = 18.884409$ and the noisy blurred image is $\sigma = 14.905295$. The mean square error (σ) of the restored CT image in Figure 2:30 is:

$$\sigma = 5.443016 \tag{43}$$

The image corresponds to the pixel-by-pixel difference between the original CT image and the restored CT image is:

Figure 2:32 difference between the restored CT by the joint restoration and the original image

2.3 Conclusion of image restoration

We have seen in chapter 2.2 how we can perform the restoration of a magnetic resonance and computed tomography image with the joint and independent quadratic constrained models. As a criterion of knowledge, about the effect that the restoration methods have caused to the restored images, we count on the mean square error (MSE), the approximate difference given in equation (20), and the pixel-by-pixel subtraction between the restored and the original image.

The joint quadratic constrained restoration method has given very good results in both magnetic resonance and computed tomography images. The joint method, functions positively in the restoration of the images without changing the position of the pixels. Moreover, the model removes, as far as possible, the noise and the blurring from the degraded image.

According to the joint restoration method, the independent restoration model performed to the measurement system with degradation, the 3x3 blurring mask has also been effective. The MSE of the restored image is reduced through the restoration process, and the deformation from the degradation model has disappeared.

On the contrary, the independent model, performed to the measurement system with degradation the 1x5 blurring mask has given a good quality restored image but with deformations in some regions of the image. Specifically, the deformation that incurs in the edges of these regions is visible.

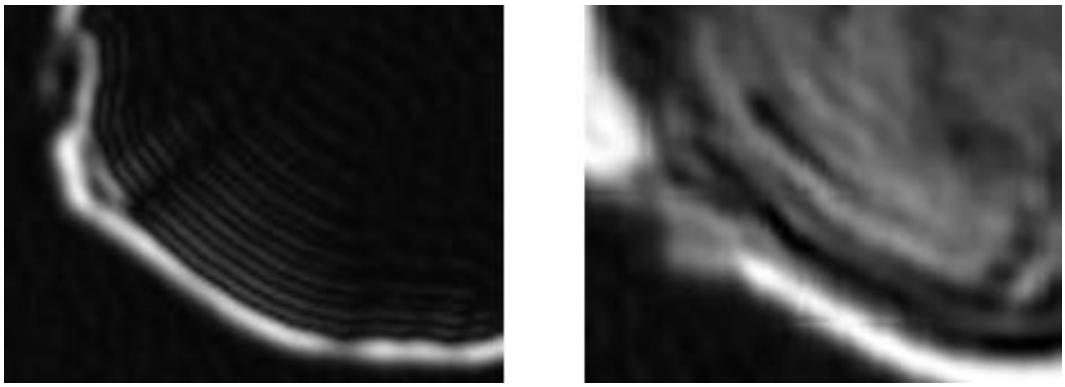


Figure 2:33 Ringing phenomenon

The goal from the previous quadratic constrained restoration models and specifically from the joint restoration model, is an additional restored image with lower MSE, in comparison with the MSE of the restored images by the 1x5 measurement systems. Instead of having a good quality restored image and a quite inefficient image, our knowledge was enhanced with an additional restored image by the joint restoration model with low MSE.

Below we represent all the results obtained by the proposed restoration methods. The joint model decreases the MSE of the restored images, in comparison with the MSE of the restored images by the 1x5 motion blurred mask. Although the

improvement of the joint model, the independent model applied to the system with degradation the 3x3 mask remains the most effective restoration model in this thesis.

Restoration method	**Original image**	**Mean square error (MSE) of input image**	**Mean square error (MSE) of restored image**
Joint restoration	MRI	15.092118	**4.647109**
		20.097562	
	CT	14.905295	**5.443016**
		18.884409	
Independent restoration with 3x3 degradation	MRI	15.092118	**3.652579**
	CT	14.905295	**4.775588**
Independent restoration with 1x5 degradation	MRI	20.097562	**7.276938**
	CT	18.884409	**10.494015**

3 Information fusion

3.1 Introduction to information fusion

Nowadays, information fusion constitutes an important challenging research topic. In “complex” systems, we generally dispose of several sources of knowledge (i.e. data, a priori knowledge, etc.) and the major question we have to deal with is how to relate all these sources in order to achieve a well predefined objective **[3]**.

Information fusion is, relatively, an old research topic that finds its roots while researchers tried to imitate the human “intelligence” or capacities when dealing with a decision-making problem. Two mains approaches have been conducted in order to fulfil this objective **[4-7]**. The first approach is issued from researches dealing with particular physical problems where information fusion methods corresponding to the considered problem “needs” are developed. On the other hand, the second approach is issued from researches in the area of artificial intelligence where the information fusion problem is considered “conceptually” and several solution frameworks are developed.

3.2 Information fusion concepts

In the following subsections, we propose to categorize the information fusion approaches as a function of the conceptual level of the information to be merged by such systems. Three conceptual levels can thus be considered: data, decision and model.

3.2.1 Data fusion [3]

3.2.1.1 Introduction to data fusion

Data fusion is an approach to processing data that has had some notable successes, primarily in the defense sector. Most previous applications of data fusion have concentrated on refining the value produced by data processing.

The general definition of a data fusion system, Figure 3:1, is as follows: “A data fusion system is defined as a system performing information fusion at the data conceptual level, where the data to be merged is assumed to represent all types of information describing an object from the real world scene”. The output of this system is either a decision (a higher conceptual level) or a new data information element. Notice that this definition makes a total abstraction of the data nature (numeric/symbolic, elementary/intermediate, coupled or not, etc.) as well as of the data “origin” (mono/multi sensor) **[3]**.

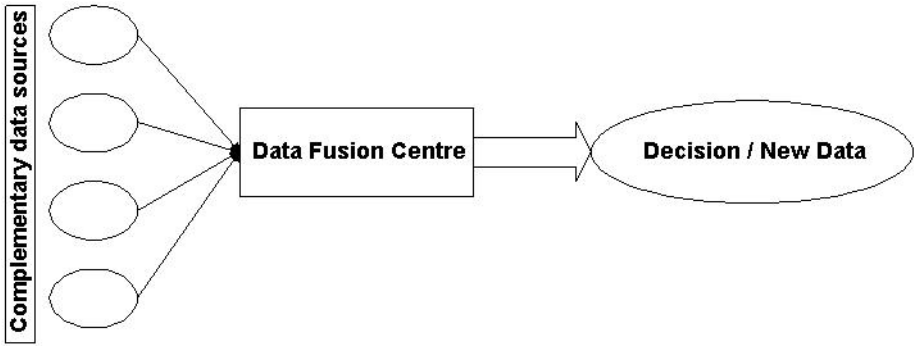


Figure 3:1 Data fusion system [3]

In order to gain an appreciation of data fusion, we have broken down the definition” of data fusion into two parts. We first describe the attributes of a data fusion problem; and then enumerate some of the solution-techniques that data fusion puts at our disposal.

3.2.1.2 The problem

It has been said that data fusion is not a particular algorithm for solving all “fusion” problems, nor is it even a suite of such techniques; it is more an approach to solving problems. However, the problem arises of deciding, which mathematical domain is the correct one in which to work for a complex problem, and which techniques and algorithms in a given domain are suitable for achieving a solution.

Data fusion is the technology whereby often large amounts of diverse data may be combined into a consistent, accurate, and intelligible whole. There are several distinct flavors of data fusion: in some scenarios, for example, the data corresponds to

different attributes of the same object; in others, the data is effectively repeated measurements of the same attribute.

In order to make the definition of data fusion more intuitive, it is useful to enumerate some of the measurement aims for which data fusion is deployed:

- Reliability - to obtain better results than the best individual data source is capable of providing
- Completeness - no direct way of measuring required property
- Improvement - the need to take more factors, or influencing quantities, into account
- Comprehension - the need to reduce information overload

3.2.1.3 The solutions

Having given, above, some indication of how one might recognize a data fusion problem as such, we now go on to describe some of the areas and techniques of mathematics that have been used to tackle problems. In the initial stages of a survey carried out by NPL, we have the following, not necessarily comprehensive, list of areas:

3.2.1.3.1 Physical modelling

In this group are the techniques that exploit a prior understanding of the behaviour of the system being measured, and of the relationships between the attributes being measured. In such circumstances it is difficult to see where data fusion can play a defining role. However, if the prior understanding is not certain or comprehensive, then this physical modelling will have to be supplemented with other mathematical techniques, and the data fusion then becomes less trivial.

3.2.1.3.2 Empirical modelling / curve fitting

In this situation, we have less or no direct physical knowledge of the behaviour of the system. We then must make generic assumptions about the measurements, and fit

the observed data to these assumptions. Typically, these intuitive, it is useful to enumerate some of the assumptions might be that the relationship between measurements is polynomial (up to a certain degree), and different types of curve fitting algorithm are then suitable.

3.2.1.3.3 Probability

Probabilistic methods for data fusion mainly center on a Bayesian approach to probability, and deal with the delicate matter of reconciling prior probabilities with observed measurements. Typically, these involve the use of Bayesian networks, which it might be convenient to think of as a fault-tree analysis of measurement systems; the known, or suspected, modes of failure of the system can be described, and the measurement results are then diagnosed in relation to this known pattern of failure.

Bayesian inference [9]

Bayesian Inference can be used to determine the probability that a diagnosis is correct, given a piece of a priori information. Analytically this process is described as follows:

$$P(f_1 \mid O_n) = \frac{P(O_n \mid f_1) \cdot P(f_1)}{\sum_{j=1}^{n} P(O_n \mid f_1) \cdot P(f_1)} \tag{44}$$

Where:

$P(f_1 \mid O_n)$ = The probability of fault (f) given a diagnostic output (O), $P(O_n \mid f_1)$ = the probability that a diagnostic output (O) is associated with a fault (f), and $P(f_1)$ = the probability of the fault (f) occurring.

Bayes' theorem is only able to analyze discrete values of confidence from a diagnostic classifier (i.e. it observes it or it doesn't). Hence, a modified method has been implemented that uses three different sources of information. A-priori probability of failure at time t, $\left(P_{FO(t)}\right)$, the probability of failure as determined from

the diagnostic classifier $\left(C_{D(i,t)}\right)$ data, and feature reliability which is independent of time $\left(R_{D(i)}\right)$. Care must be taken to prevent division by zero.

$$P_{f(t)} = \frac{\sum_{i=1}^{N} C_{D(i,t)}}{P_{FO(t)} \cdot \sum_{i=1}^{N} R_{D(i)}} \tag{45}$$

3.2.1.3.4 Statistics

Statistical techniques will typically be applicable when there is much more data than knowledge. When this is the case, then we can start to draw conclusions about relations in the data without too many prior assumptions or physical intuition of the system.

We may use traditional statistical techniques for a variety of tasks subsidiary to fusion. For instance, we may use x^2 tests and analysis of variance to help us determine the presence of a faulty sensor within an array of other sensors, or we can use analysis of covariance to help us decide which measurements are interrelated, and which are directly linked to the measurand. This information can then be taken into account when deciding which measurements (or measurement devices) are significant.

3.2.1.3.5 Soft computing

These techniques traditionally used to achieve artificial intelligence. They cover the areas of artificial neural networks (ANN), genetic algorithms, and fuzzy logic, amongst others. ANN's might be used to learn how a series of past inputs to a system related to its outputs, and then use this "knowledge" accurately to fuse present inputs in order to predict an output. Fuzzy logic can be used as a way of representing knowledge of a system that is more qualitative than would be the case in a physical modelling situation.

Fuzzy logic inference [9]

Fuzzy logic inference is a fusion technique that utilizes the membership function approach to scale and combine specific input quantities to yield a fused output. The basis for the combined output comes from scaling the developed membership functions based on a set of rules developed in a rule base. Once this scaling is accomplished, the scaled membership functions are combined by one of various methodologies such as summation, maximum or "Finally, the scaled and combined membership functions are used to calculate the fused output by either taking the centroid, max height or midpoint of the combined function.

Neural network fusion [9]

A well-accepted application of artificial neural networks (ANNs) is data and feature fusion. For the purposes of fusion, a networks ability to combine information in real-time with the added capability of autonomous re-learning (if necessary) makes it very attractive for many fusion applications.

Artificial neural networks (ANN) utilize a network of simple processing units, each having a small amount of local memory. These units are connected by "communication" links, which carry numerical data. The units operate only on their local data, which is received as input to the units via the connections. Most ANN's have some sort of training rule by which the weights of connections are adjusted based on some optimization criterion.

Hence, ANN's learn from examples and exhibit certain capability for generalization beyond the training data (examples). ANN's represent a branch of the artificial intelligence techniques that have been increasingly accepted for data fusion and automated diagnostics in a wide range of aerospace applications. Their abilities to fuse features, recognize patterns and to learn from samples have made ANN's attractive for fusing large data sets from complex systems.

The ANN *structure* is called its architecture, which is an expression of the number of processing units and of the connections among these units. Most processing units are arranged in *layers* (a layer is a collection of the units aligned for the same computational sequence), and the ANN is often referenced by the number of layers and the number of units in each layer.

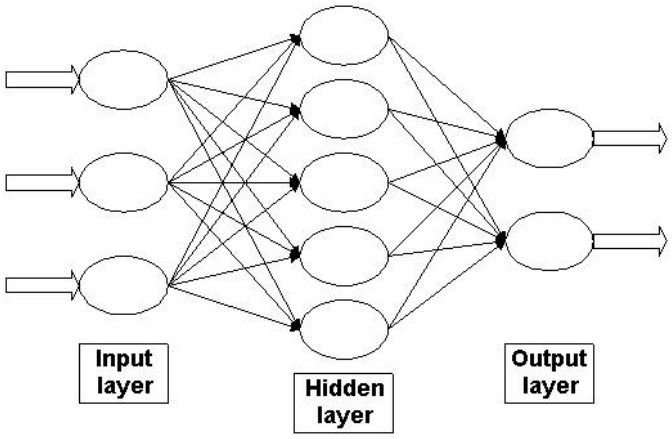


Figure 3:2 Simple neural network fusion architecture [9]

Each solid connection line in Figure 3:2 represents a numerical value called the *weight,* representing the connecting strength between the two inter-connected units. Each circle is a unit and it performs three sequential computations: the first is to multiply the weight by the output of the unit on the other end of the connection; the second is to sum the *weighted outputs* from all connections; and the third is to apply the *weighted sum* to a function (usually nonlinear and bounded) called an *activation* function. One of the most common activation functions is called the sigmoid function and the binary $f(x)$ (0 to 1 input) and bipolar $g(x)$ (-1 to 1 input) versions are given below. They are useful because the simple form of the derivative reduces the computational burden during training.

$$f(x) = \frac{1}{1+e^{-\sigma \cdot x}} \tag{46}$$

$$g(x) = \frac{1-e^{-\sigma \cdot x}}{1+e^{-\sigma \cdot x}} \tag{47}$$

The functional value of the weighted sum is called the *output* of the unit. This sequence of computation is carried out for each unit and for each layer until the outputs layer of the ANN is reached.

Training a neural network for a fusion application involves the process of adjusting the weights and evaluating the activations of the numerous interconnections between the input and output layers. There are two fundamental types of learning methods used for feature fusion applications: *unsupervised* and *supervised*. In the

unsupervised method, learning is autonomous; networks discover properties of the data set and learn to reflect these properties in its output, i.e., it is used to group similar input patterns to facilitate processing of a large number of training patterns. In the supervised method, a "teacher" is present during the training phase who tells the network how well it is performing or what the correct behavior should be, i.e., it is used for specifying what target outputs should result from an input pattern.

3.2.1.3.6 Signal processing

Signal processing could be thought of as a subset of physical or empirical modelling. However, the problems posed by fusing many real-time data streams, such as their synchronization, taken with the fact that these techniques are often not familiar to those outside the signal processing fraternity, has encouraged us to give the subject its own heading. The Kalman filter is one of the best-known signal processing ideas within data fusion. This technique allows us to reconcile many measurements of the state of a dynamic system with predictions of the subsequent state (predicted by some physical model of state evolution), each being weighted by some indication of the source's reliability, in order to achieve the best predictor of the state.

3.2.1.3.7 Novel techniques

Here we place techniques that are difficult to place elsewhere. Over time, it is likely that similarities will be seen, either amongst these techniques, or between these and those in the above categories. An example of a novel technique is Dempster-Shafer theory. This is in many ways an alternative to the usual probabilistic description of uncertainty. Sensors have a prior level of confidence, and provide a measurement result with this confidence.

Multiple sensors can have very different confidence characteristics (for example, sensor X is reliable in the x-coordinate of a two-dimensional measurement but not in the y-coordinate, sensor Y is the converse). These characteristics are taken into account when the measurements are being fused according to the Shafer rule of combination. One of the primary advantages of the possibilistic approach (as it is called) is that it can deal consistently with a sensor that claims to be ignorant of a

component of the measurand, and is useful in diagnosing, and potentially dealing with, conflicting data sources.

Dempster-Shafer method [9]

The Dempster-Shafer method specifically tackles the a priori probability issue by keeping track of an explicit probabilistic measure of the lack of information. The disadvantage of this method is that the process can become impractical for time critical operations in large fusion problems. Hence, the proper choice of method should be based on the specific diagnostic/prognostic issues that are to be addressed.

In the Dempster-Shafer approach, uncertainty in the conditional probability is considered. The Dempster-Shafer methodology hinges on the construction of a set, called the frame of discernment, which contains every possible hypothesis. Every hypothesis has a belief denoted by a mass probability (m). Beliefs are combined in the following manner.

$$Belief\left(H_n\right)=\frac{\sum_{A\cap B=H_n} m_i\left(A\right)\cdot m_j\left(B\right)}{1-\sum_{A\cap B=0} m_i\left(A\right)\cdot m_j\left(B\right)} \tag{48}$$

3.2.2 Decision fusion [3]

A decision fusion system, Figure 3:3, is defined as “a system realizing the fusion of complementary decisions concerning the same object observed in the real world scene”. Notice that this definition means that the decisions to be merged are of the same nature and all concern the same object.

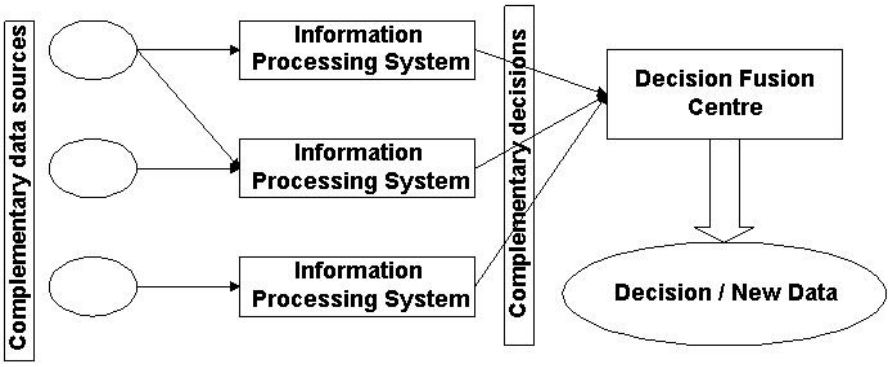


Figure 3:3 Decision fusion system [3]

The reason of stressing this point is that an intuitive question can be raised about the difference between the two concepts of the data and the elements of decision information. The frontier between these two concepts is extremely small. In several applications, an information element given as a decision can be considered as an input data information to another data fusion system. Therefore, and in order to make a clear distinction between the data and the decision fusion systems we assume that in the case of decision fusion systems, all the decisions to be merged are of the same nature in terms of representing the same object.

These decisions are obtained through different information processing systems. Some of these decisions can be obtained through the use of a "previous" data or a decision information fusion system. Several approaches can be used in order to implement such fusion systems. For instance, the multi-expert majority, the probabilistic reasoning approaches, etc.

3.2.3 Model fusion [3]

A model fusion system, Figure 3:4, is simply defined as "any system integrating several complementary technical components (i.e. models) and aims at computing a new information element (a decision or a data)". By complementary we mean that each technical component contributes partially in the construction of the global solution permitting to obtain the output that we are searching for.

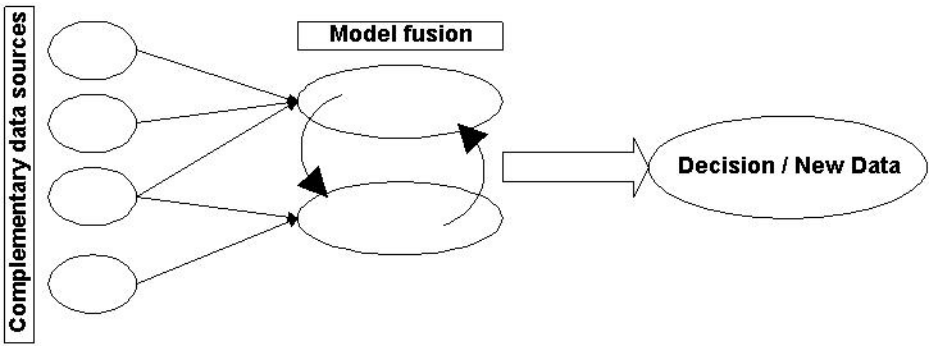


Figure 3:4 Model fusion system [3]

3.3 Information fusion system architectures

This subsection intends to summarize the two major fusion system architectures. In fact, two main architectures of information fusion systems can be distinguished.

3.3.1 Mono-sensor architecture

The mono-sensor architecture, Figure 3:5, is based on the use of a single sensor and by applying different sources of a priori information in order to obtain a new set of information data. Each component of this new set is strongly related to one or several a priori information sources. The use of the probability set theory or the fuzzy set theory is generally performed through this step.

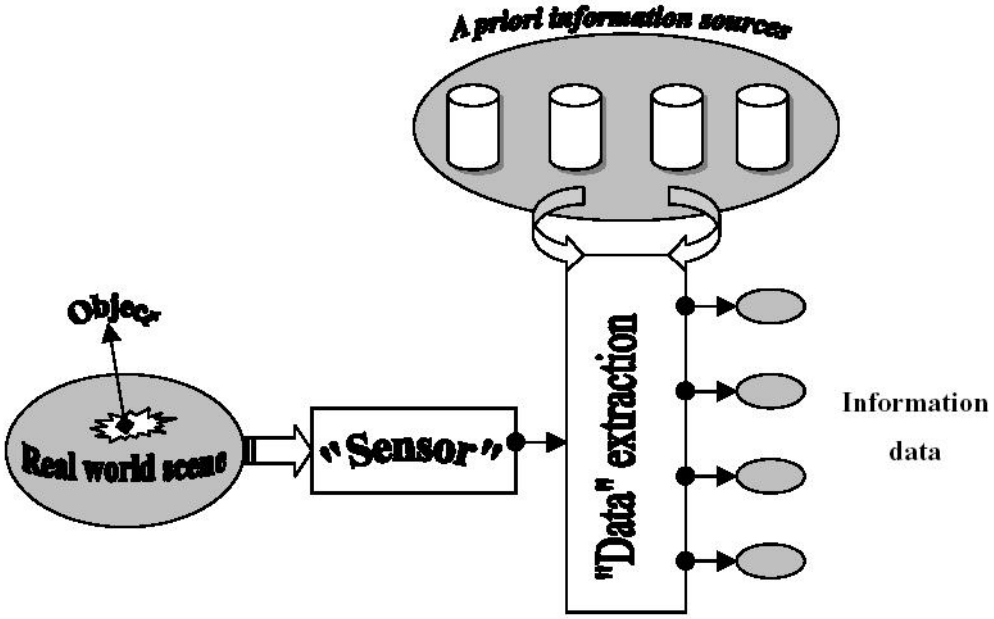


Figure 3:5 Mono-sensor architecture [3]

3.3.2 Multi-sensor architecture

The second system architecture, Figure 3:6, corresponds to the intuitive multi-sensor situation where the "analysed" object is observed through different physical sensors (or the same sensor but with different geometric observation positions as is the case in stereovision).

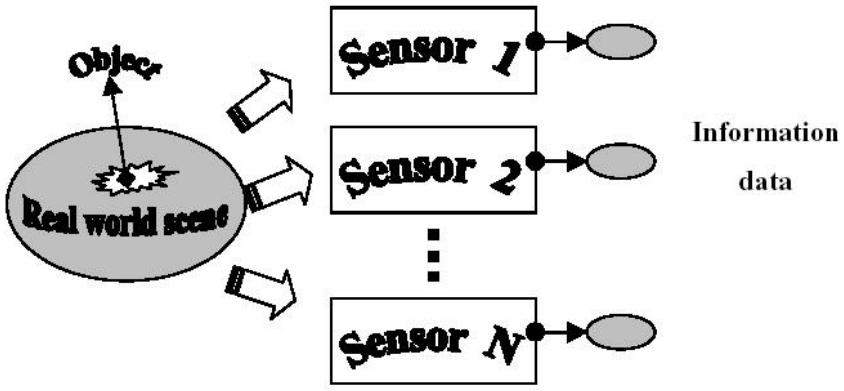


Figure 3:6 Multi-sensor architecture [3]

Multi-sensor is the main architecture used in several applications where the use of different sensors remains an obstacle and where an important amount of knowledge

can be formulated as a priori knowledge sources of information. This is the case, for instance, in medical applications where the processing system can use a huge amount of a priori anatomical and expert-based sources of knowledge while analysing medical Images.

3.4 Information fusion in this thesis

In this chapter, we focus on the image fusion model, applied to the medical images from the restoration models. The multi-sensor architecture system is the system that provides the set of the output data from the two measurement systems. The first, second and the third “sensor”, obtain respectively the output image from the joint restoration, the restored image from the independent restoration applied to the measurement system with degradation the 1x5 mask, and finally the image from the independent method applied to the measurement system with degradation the 3x3 mask.

The proposed image fusion procedure is separated in two parts. In the first part, the image fusion with the pointwise spatial frequency combines separately the restored magnetic resonance and computed tomography sets of images. In the second part, the proposed image fusion model performs the combination of the anatomical data from the restored MR image and the bony structures of the restored CT image into a single image. The image fusion procedure is given in Figure 3:7.

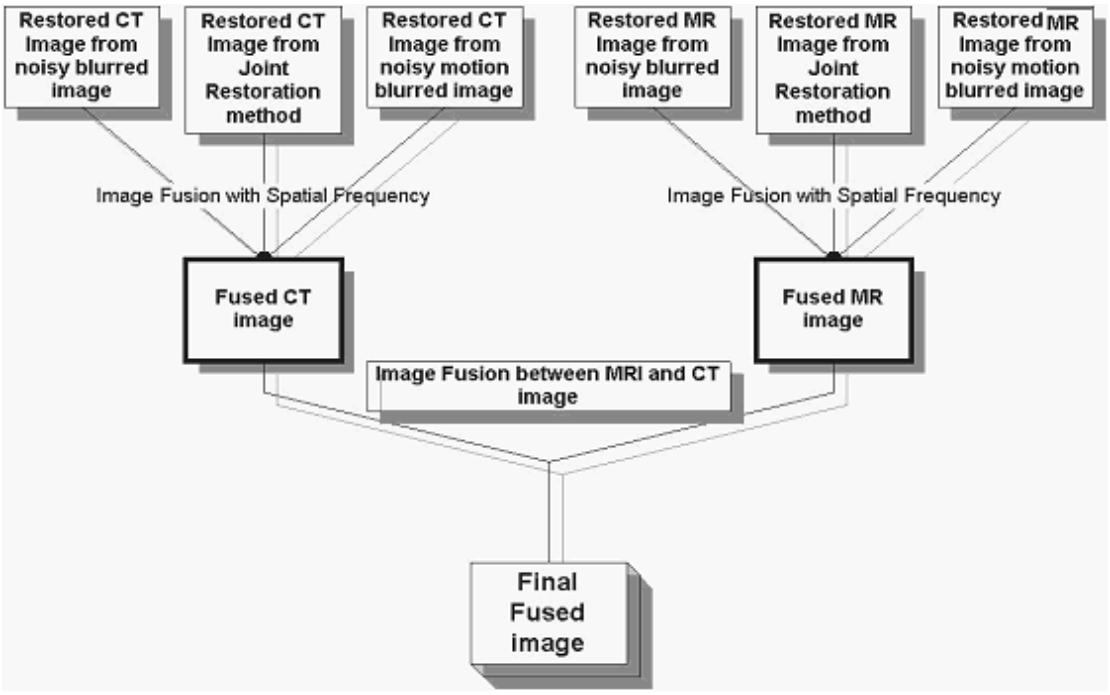


Figure 3:7 Image fusion procedure

3.4.1 Image fusion with multiple images of the same form

In this chapter, the first part of the image fusion model is represented. Note that all the images from both MR and CT sets of input sources, are participating equally in the image fusion procedure.

3.4.1.1 Introduction to image fusion with spatial frequency

The first part of image fusion, is represented in Figure 3:8. In the following text, we represent analytically the original image fusion with the PSF **[10]** and the advanced, by this thesis model, applied to the MR and CT sets of images.

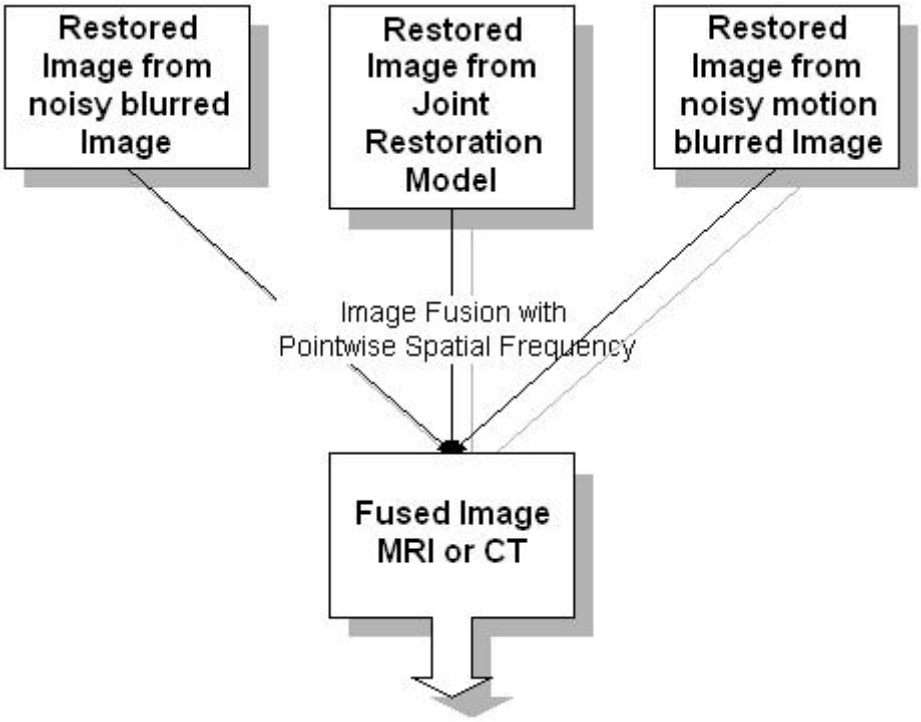


Figure 3:8 Image fusion with pointwise spatial frequency

3.4.1.2 Methodology of the original image fusion with PSF [10]

The criterion for the image fusion application is based on the pointwise spatial frequency (PSF) coefficient. The images of the same form are fused at each output pixel location by comparing the PSF values at that location in all images and selecting the pixel with the highest such value. We define the pointwise spatial frequency (PSF) at a pixel by summing the magnitudes of differences from the center pixel and each other pixel in a 3x3 neighborhood (nbhd).

$$psf_{i,j} = \sum_{k=-1}^{1} \sum_{l=-1}^{1} \left| x_{i,j} - x_{i-k,j-l} \right| \tag{49}$$

where $\mathrm{psf}_{i,j}$ is the pointwise spatial frequency of pixel $\mathrm{x}_{\mathrm{i,j}}$

For finer granularity, this thesis has modified the calculation of the PSF of a pixel, when the examined pixel has less than 8 neighborly pixels in its region. The magnitude of difference exists only when there is a neighbor in the pixel, without filling with zero values the outline of the two-dimensional matrix corresponds to the image.

The algorithm that this method develops, selects the pixel with the highest PSF value among to a set of pixels that correspond to the same location of the image. The comparison among the set of pixels of the same location, and the selection of the pixel according to the PSF coefficient, is effective in non-homogenous, or in bordered, regions. The magnitude of difference from the center pixel and each other pixel in its 3x3 neighborhood in such a region, is really high despite the noise that might participate in the intensity of the pixel. Thus, the probability of a succeed selection is possible.

On the contrary, in homogenous regions, the PSF of a pixel has generally a low value and we should be very careful during the time of selection because the noise intensity is playing an important role in the procedure. For that reason, removing the noise deformation from the image, through the restoration methods, is essential.

In the following figures we represent the PSF values of the MR and CT images, normalized first in region $[0,K\ ,255]$.

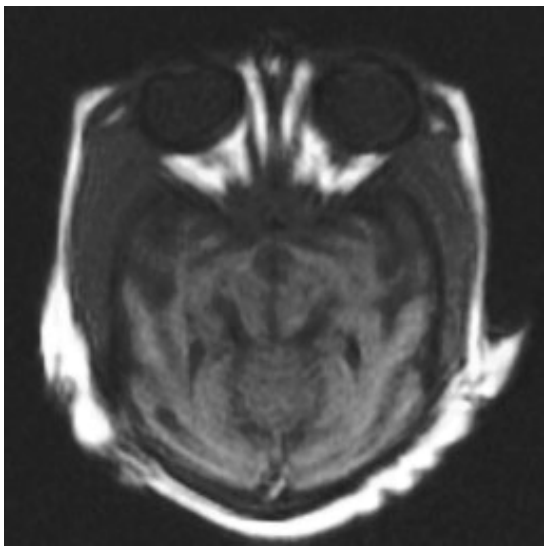


Figure 3:9 Original MR image

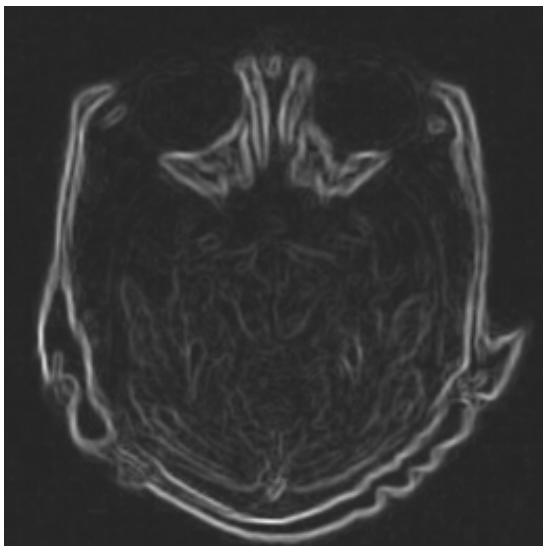


Figure 3:10 PSF of the original MR image

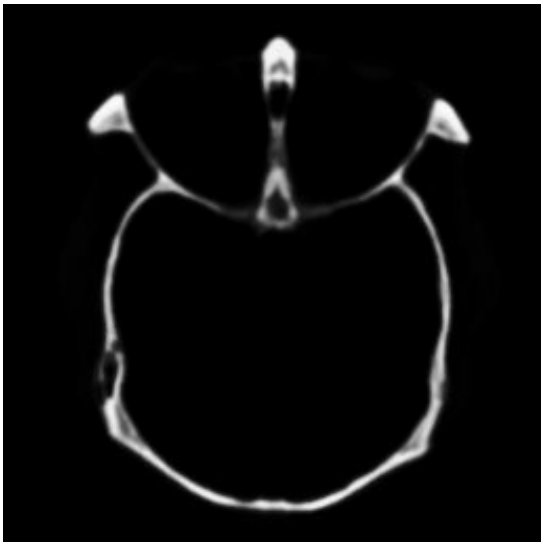


Figure 3:11Original CT image

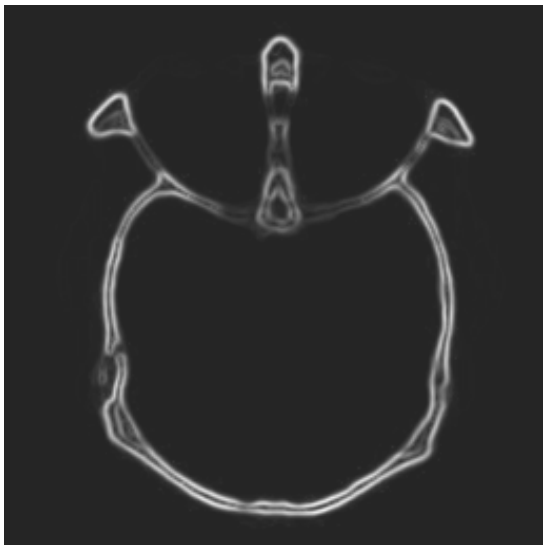


Figure 3:12 PSF of the Original CT image

The general law of this method is to fuse the R images at each pixel location by comparing the PSF values of the R corresponding pixels. For all R pixels x_r at the same location we put $x_{out} = x_{r^*}$ where r* is determined by the following equation.

$$psf\left(x_{r^*}\right) = \max_r \left\{ psf\left(x_r\right) : r = 1,\mathrm{K}\ , R \right\} \tag{50}$$

3.4.1.3 Improvements on the methodology of image fusion with the PSF

Keeping quite the same methodology that the original image fusion with the PSF proposes, this thesis has simulated a more complicated method for image fusion. We have seen in the previous method that the image fusion procedure fuses the input images at each pixel location by comparing the PSF values of the corresponding pixels, and finally selects the pixel with the highest value. With this method, the intensity of a pixel in the fused image is given by only one input source. This decision sometimes might be quite inefficient in a variety of data where measurement systems with different degradations affect the original data.

The measurement systems described in this thesis, affect the original MR and CT images. The degrading elements of noise and blurring affect all the pixels of the original image, changing their intensity and their original position. Besides this, we have seen before that the restoration methods we have applied to the degraded images

cannot totally restore the original image, specifically in the case of the 1x5 motion-blurred mask where the deformation in some regions is visible.

This deformation is quite often in image restoration, specifically when the image is affected by measurement systems with motion blurred degradation. For that reason, selecting the information of the fused pixel from only one source might cause troubles and inaccuracies in the resulted data. Consider the circumstance, where the intensity of the fused pixel is selected to be the intensity in which the deformation exists. The pixel of the restored input image fulfils the terms of the algorithm but unfortunately the pixel does not correspond to the real location of the image.

According to the previous assumptions, we have performed an intermediate step between the restored input images and the fused image as it is shown in Figure 3:13. The restored image by the 1x5 motion blurred image is being updated through this step with the main purpose to eliminate the ringing deformation of the image.

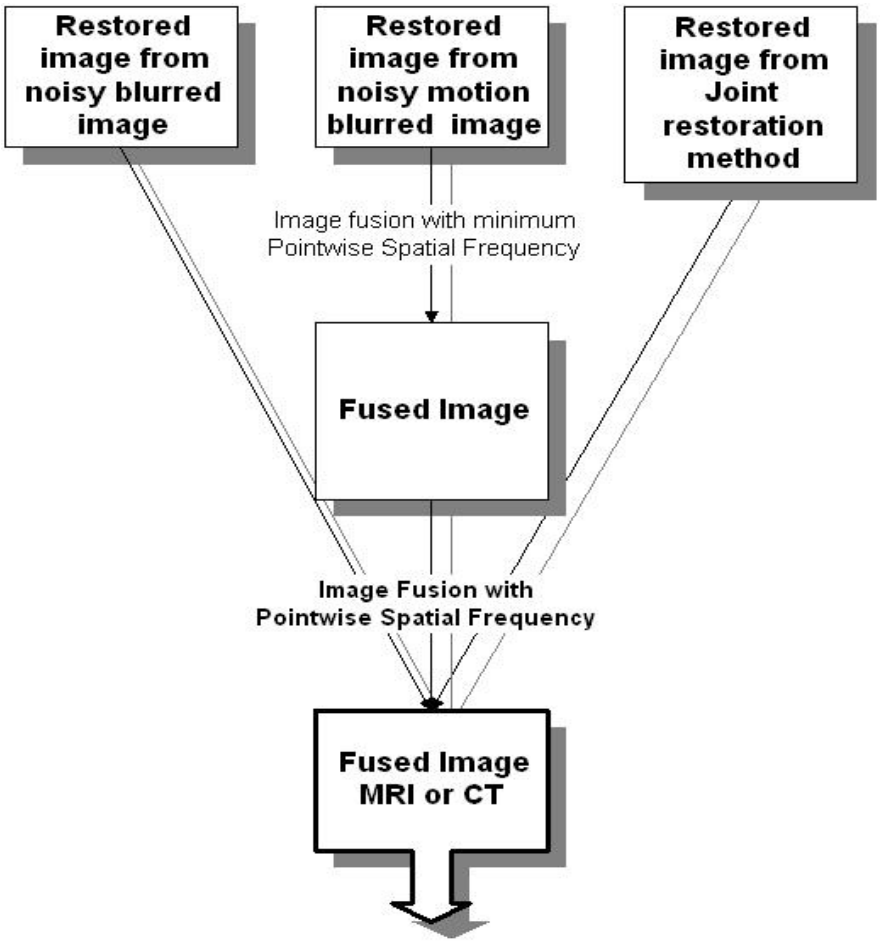


Figure 3:13 Graph of the proposed image fusion with the pointwise spatial frequency

The image fusion model with the PSF is separated in two parts. Note that, in both parts the definition of the pointwise spatial frequency (PSF) at a pixel is the same as in the original image fusion with the PSF. The PSF of a pixel is given in equation (51)

.

$$psf_{i,j} = \sum_{k=-1}^{1} \sum_{l=-1}^{1} \left| x_{i,j} - x_{i-k,j-l} \right| \quad (51)$$

where $\mathrm{psf}_{i,j}$ is the pointwise spatial frequency of pixel $\mathrm{x}_{\mathrm{i,j}}$

The image fusion method in the first part, performed to eliminate the deformation of the restored image given by the measurement system with degradation the 1x5 mask, is based on the selection of the pixel with the minimum pointwise spatial frequency. We have documented before, that the intensity of the SPF is based on the magnitude difference between the examined pixel and its neighbors in a 3x3 region. In other words, when the PSF coefficient is high the discontinuities of the 3x3 neighborhood of the pixel are large.

Thus, if we select the highest PSF among a set of pixels, we automatically select the pixel that belongs in a rough region. Alternatively, if we select the pixel with the minimum PSF value, we prefer to select a smoothing area rather than an area with a visible edge. Applying this feature in regions where we want to eliminate the ringing deformation, smoothing areas with a low value of intensity or background areas with zero intensity are selected to replace the region with the deformation.

According to the previous assumptions, our common reaction is to replace the intensity of the degraded pixels with the optimized pixels adopted by the image fusion procedure in equation(52). The input data are, respectively, the restored MR and CT sets of images and the resulted image is the optimized restored image by the 1x5 degradation mask. In the following step of the proposed image fusion with the PSF, the optimized restored image replaces the restored image by the noisy motion blurred mask.

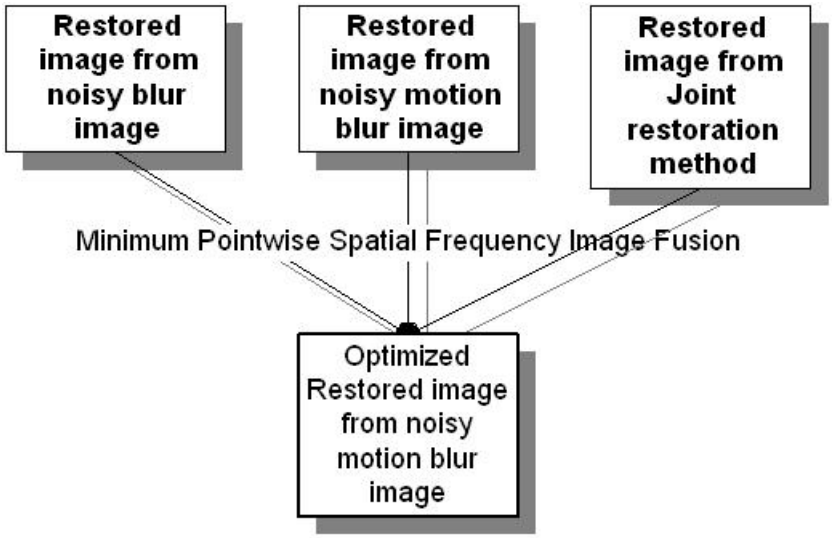


Figure 3:14 Minimum PSF selection

The general law of this method is to fuse the R images at each pixel location by comparing the pointwise spatial frequency (PSF) values of the R corresponding pixels. For all R pixels x_r at the same location we put $x_{out} = x_{r^*}$ where r* is determined by the following equation.

$$psf\left(x_{r^*}\right) = \min_r \left\{ psf\left(x_r\right) : r = 1,\mathrm{K}\ , R \right\} \tag{52}$$

In the second part, the resulted image from the first part and the rest images given by the restoration models are participating in the proposed image fusion with the pointwise spatial frequency. Analytically, depending on the value of the pointwise spatial frequency (PSF), each pixel participates proportionately in the fused image. The equation for this model is,

$$\text{for every fused pixel} : \hat{x} = \frac{\sum_{r=1}^{R}\left(psf\left(x_r\right) \cdot x_r \right)}{\sum_{r=1}^{R} psf\left(x_r\right)}\ , \mathrm{r} = 1,\mathrm{K}\ , R \tag{53}$$

where R are the multiple images to be fused, x_r is the pixel of each image in the examined location, and finally $psf\left(x_r\right)$ is the pointwise spatial frequency of the pixel x_r.

3.4.1.4 Results of the image fusion with the PSF

In the first sub-chapter we represent the results obtained by the original image fusion with the pointwise spatial frequency **[10]**. In the second sub-chapter, the proposed image fusion method is applied to the restored magnetic resonance and computed tomography images. Finally, a comparison between the two methods is documented.

3.4.1.4.1 Results from the original image fusion with PSF

The resulted, from the original image fusion model, magnetic resonance image and its histogram are obtained, respectively, in Figure 3:15 and Figure 3:16.

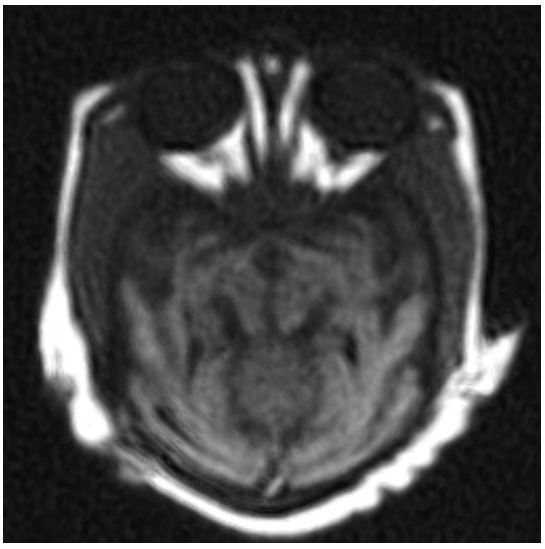


Figure 3:15 Fused MR image by the original image fusion with PSF

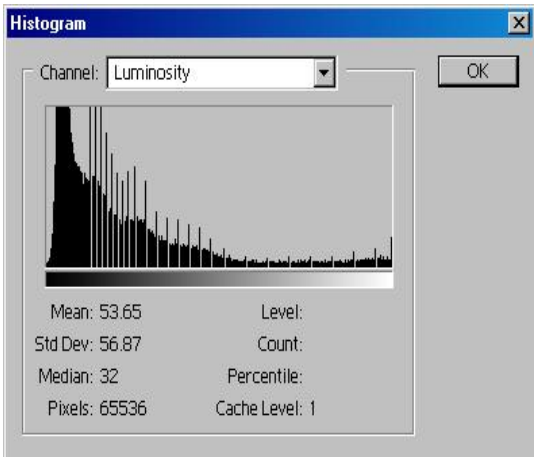


Figure 3:16 Histogram of fused MR image by the original image fusion with PSF

The mean square error (σ) of the fused MR image in Figure 3:15 is:

$$\sigma = 5.869003 \tag{54}$$

In Figure 3:17 we represent the pixel-by-pixel difference between the original MR image and the fused MR image. Unfortunately, the ringing deformation of the fused image is preserved.

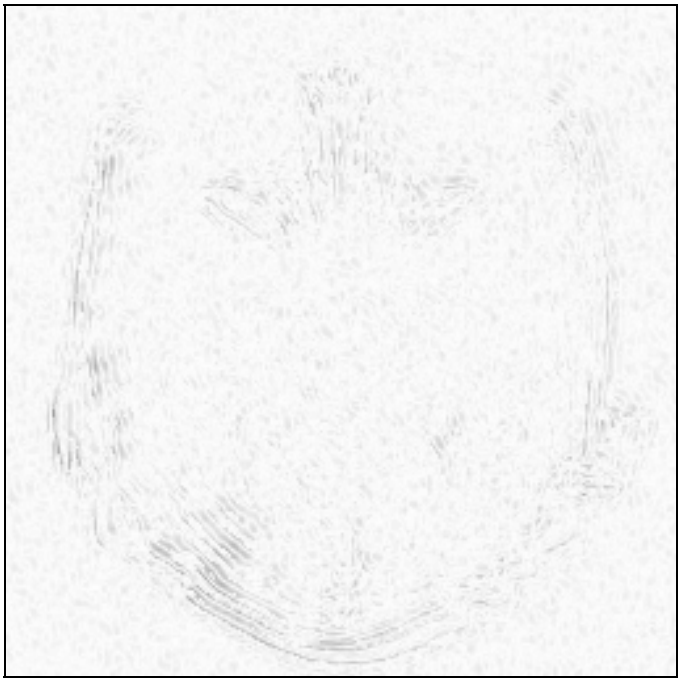


Figure 3:17 difference between the fused MR image by the original image fusion with PSF and the original MR image

The resulted, by the original image fusion model, computed tomography image and its histogram are obtained, respectively, in Figure 3:18 and Figure 3:19.

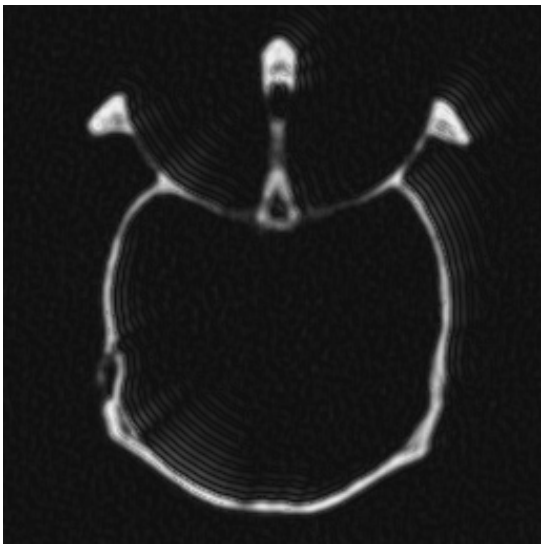


Figure 3:18 fused CT image by the original image fusion with PSF

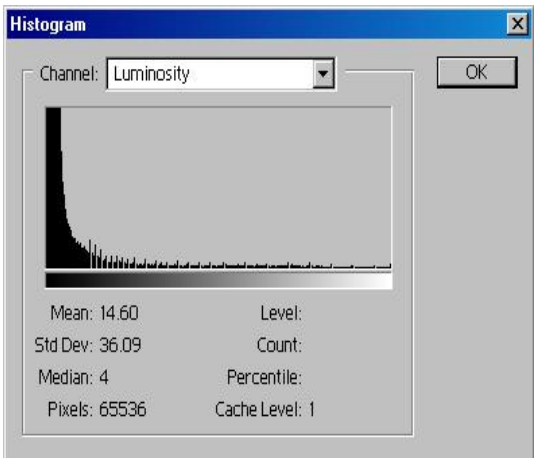


Figure 3:19 Histogram of fused CT image by the original image fusion with PSF

The mean square error (σ) of the fused CT image in Figure 3:18 is:

$$\sigma = 8.618378 \tag{55}$$

Below, we represent the pixel-by-pixel difference between the original CT image and the fused CT image. The ringing deformation of the fused CT image is visible in Figure 3:20.

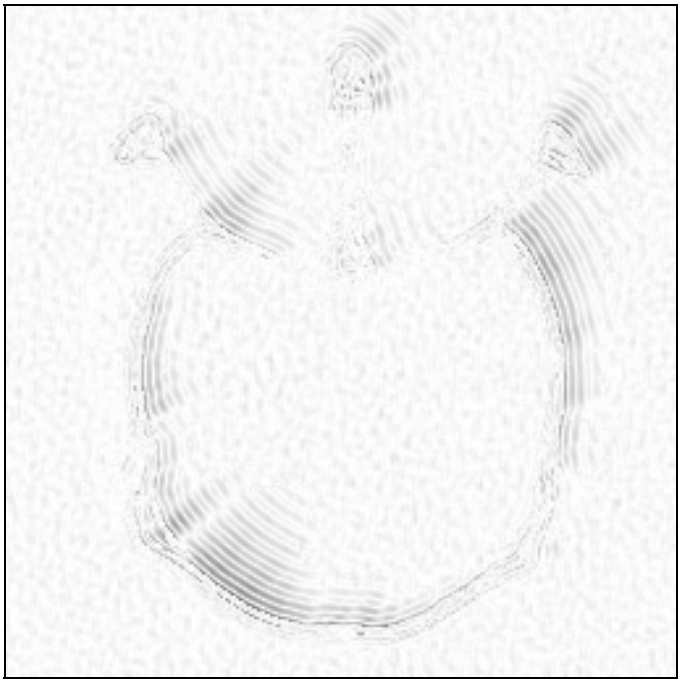


Figure 3:20 difference between the fused CT image by the original image fusion with PSF and the original MR image

3.4.1.4.2 Results from the proposed image fusion with the PSF

In the following text, we represent the results obtained by the first and the second part of the proposed image fusion with the PSF, applied to the MR and CT sets of images.

3.4.1.4.2.1 Elimination of deformation of the restored image by the noisy motion blurred degradation mask.

Note that, the optimized magnetic resonance image has updated its pixels according to equation (52). The benefits of this intermediate step of image fusion are

the following. First, the MSE of the updated restored MR image is lower than the MSE of the original restored image. Moreover, the deformations of the original restored image are eliminated through the first part of the proposed image fusion process. The restored image, affected by the 1x5 motion blur mask, and the optimized restored image are given, respectively, in Figure 3:21 and Figure 3:22.

Note that the mean square error of the MR image, given by the independent restoration method, is $\sigma = 7.276938$. The MSE of the restored MR image after the intermediate step of image fusion is:

$$\sigma = 5.729223 \tag{56}$$

The percentage of pixels that the optimized MR image has taken from each input restored image is given below. Note that the pixels in the 256x256 images are 65536.

- Restored image by the joint restoration model: **9336 pixels** or **14.24%**.
- Restored image by the independent restoration model, applied to the measurement system with the 1x5 degradation mask (image with deformation): **33551 pixels** or **51.19%**.
- Restored image by the independent restoration model, applied to the measurement system with the 3x3 degradation mask: **22649 pixels** or **34.56%**.

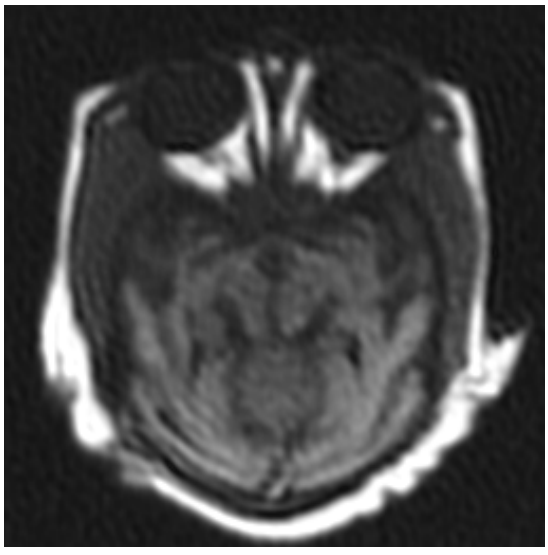


Figure 3:21 Restored MRI Image with deformation

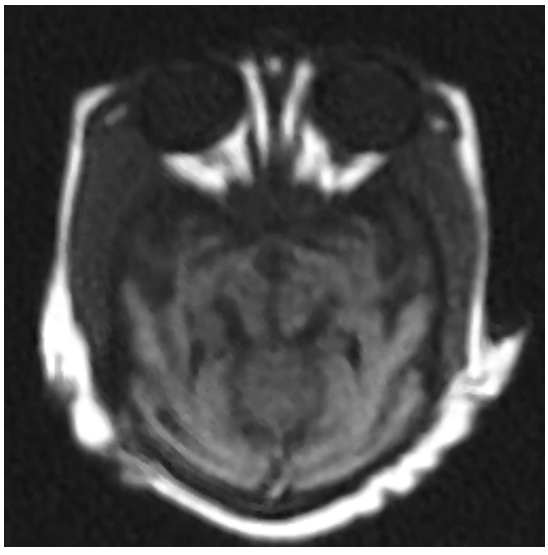


Figure 3:22 Optimized MRI Image

The pixel-by-pixel difference between the restored MR image and the optimized MR image is given in the following figure. It is essential to present this figure in order to see the positive effect that the fusion procedure has caused to the restored image.

Figure 3:23 Pixel-by-pixel difference between the restored and the optimized MR image

Applying the same optimization process to the restored CT image, the benefits of the optimization method are the lower MSE in the optimized restored CT image and the elimination of the deformations. The restored CT image, affected by the 1x5 motion blur mask, and the optimized restored CT image are given, respectively, in Figure 3:24 and Figure 3:25.

Note that the MSE of the restored CT image is $\sigma = 10.494015$. The MSE of the restored CT Image after the intermediate step of image fusion is:

$$\sigma = 7.121319 \tag{57}$$

Moreover, we present the percentage of pixels that the optimized CT image has taken from each restored CT image.

- Restored image by the joint restoration model: **7987 pixels** or **12.18%**.

- Restored image by the independent restoration model, applied to the measurement system with the 1x5 degradation mask (image with deformation): **32771 pixels** or **50.00%**.
- Restored image by the independent restoration model, applied to the measurement system with the 3x3 degradation mask: **24778 pixels** or **37.80%**.

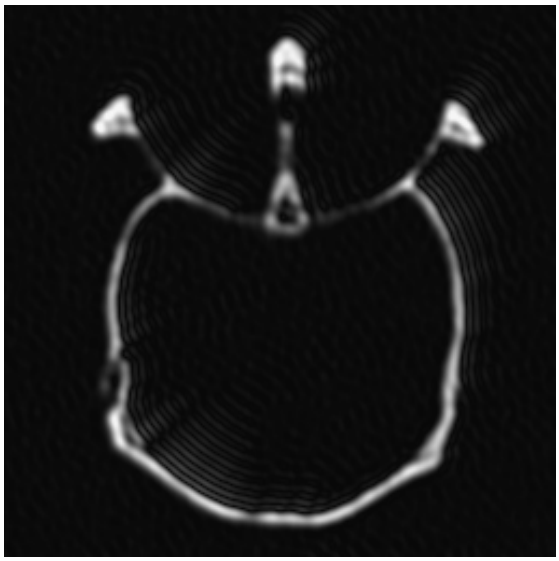


Figure 3:24 Restored CT image with deformation

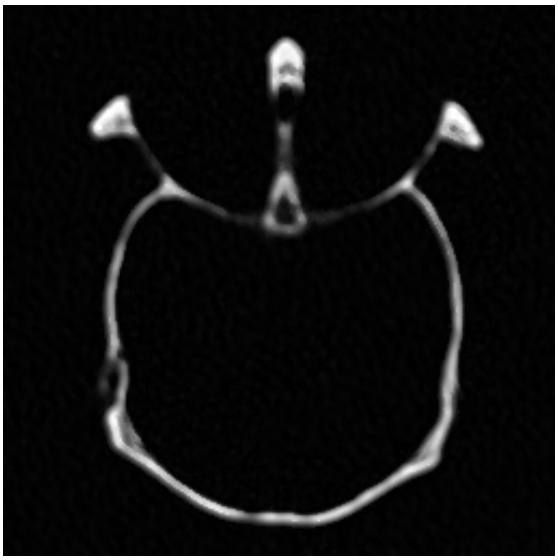


Figure 3:25 Optimized CT image

The pixel-by-pixel difference between the restored CT image and the optimized CT image is given in Figure 3:26. Note that the application of the proposed image fusion to the restored CT image functions positively. The ringing at the edges of the restored image has almost disappeared.

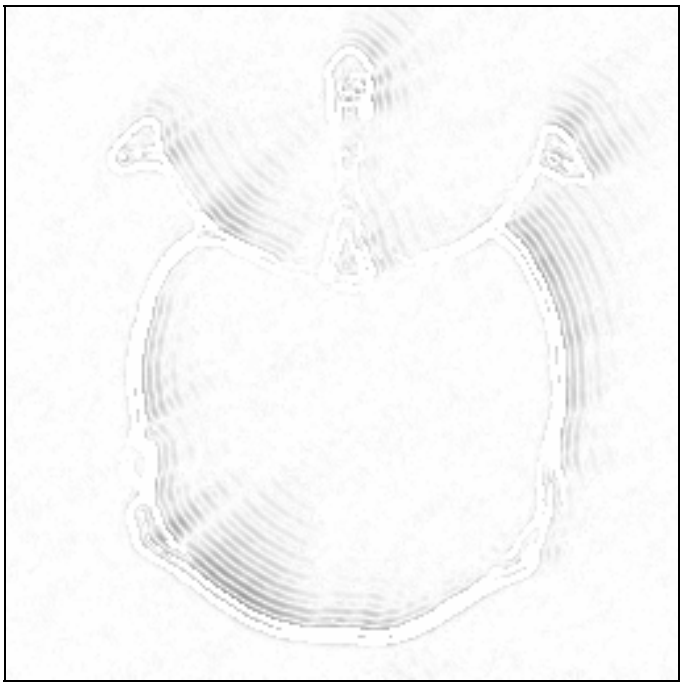


Figure 3:26 Pixel-by-pixel difference between the restored and the optimized CT image

3.4.1.4.2.2 MR and CT image fusion results from the proposed method

The resulted, from the proposed image fusion model, magnetic resonance image and its histogram are obtained, respectively, in Figure 3:27 and Figure 3:28.

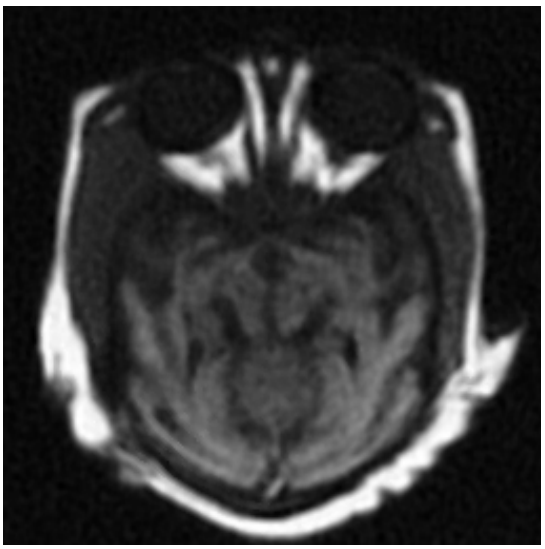


Figure 3:27 Fused MR image by the proposed image fusion with PSF

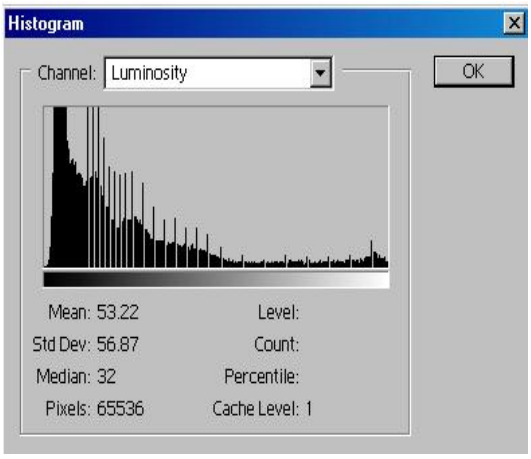


Figure 3:28 Histogram of fused MR image by the proposed image fusion with PSF

The Mean Square Error (σ) of the fused MR image in Figure 3:27 is:

$$\sigma = 4.463830 \tag{58}$$

In Figure 3:29 we represent the pixel-by-pixel difference between the original MR image and the fused MR image. It is obvious that the ringing phenomenon, a problem that we couldn't avoid in the previous method, has eliminated.

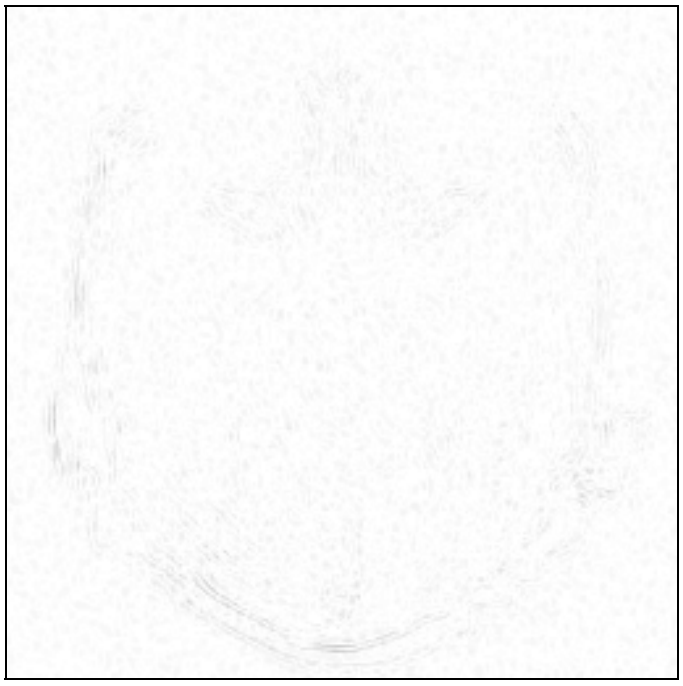


Figure 3:29 difference between the fused MR image by the proposed image fusion with PSF and the original MR image

The resulted, from the proposed image fusion model, computed tomography image and its histogram are obtained, respectively, in Figure 3:30 and Figure 3:31.

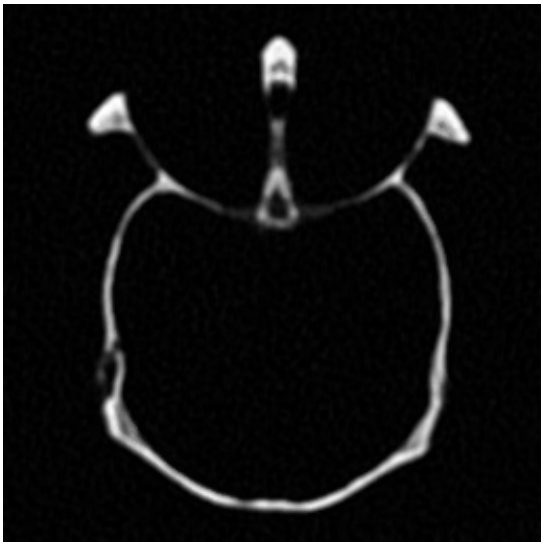


Figure 3:30 Fused CT image by the proposed image fusion with PSF

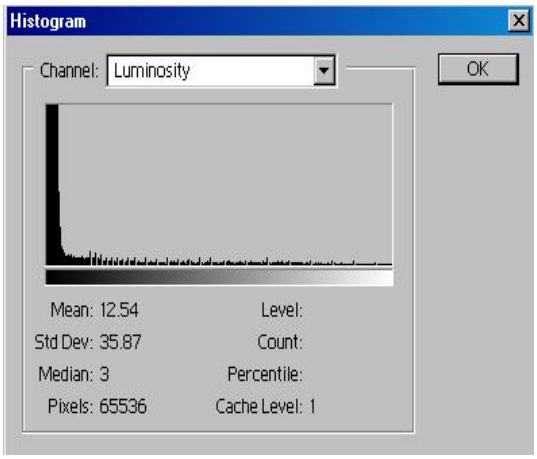


Figure 3:31 Histogram of fused CT image by the proposed image fusion with PSF

The mean square error (σ) of the fused CT Image in Figure 3:30 is:

$$\sigma = 5.172473 \tag{59}$$

In Figure 3:32 we represent the pixel-by-pixel difference between the original CT image and the fused CT image. It is obvious that the ringing phenomenon, a problem that we couldn't overcome in the original image fusion with PSF, has eliminated. The selection of the pixel with the minimum pointwise spatial frequency, performed to the restored image by the measurement system with degradation the 1x5 motion blurred mask, has acted decisively and has "cleaned" the fused image from the ringing deformations.

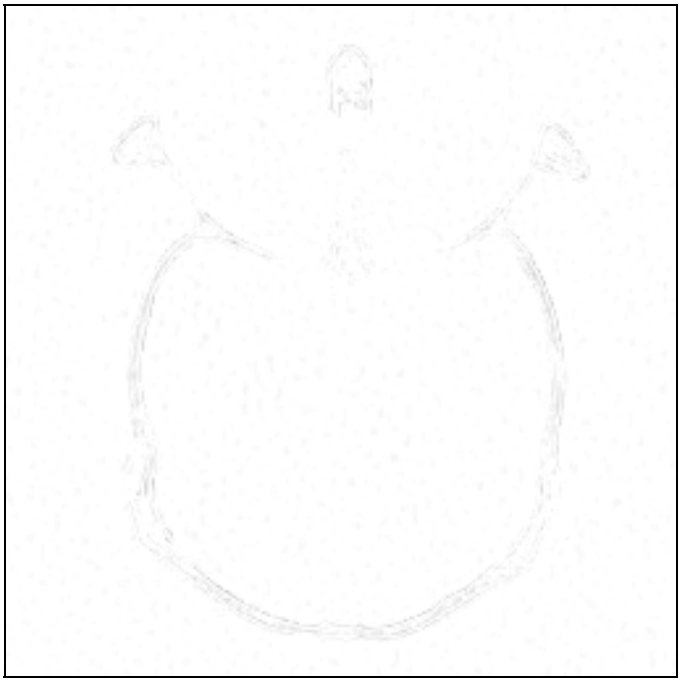


Figure 3:32 difference between the fused CT image by the proposed image fusion with PSF and the original CT image

In the following table, we represent all the results from the image fusion methods. It is visible that the proposed image fusion with pointwise spatial frequency has given better results than the original image fusion method.

Image fusion method	Original image	Mean square error (MSE) of input image	Mean square error (MSE) of fused image
Original image fusion with PSF	MR	4.647109	**5.869003**
		3.652579	
		7.276938	
	CT	5.443016	**8.618378**
		4.775588	
		10.494015	
Proposed image fusion with PSF	MR	4.647109	**4.463830**
		3.652579	
		5.729223	
	CT	5.443016	**5.172473**
		4.775588	
		7.121319	

3.4.2 Image fusion between computed tomography and magnetic resonance Images

In the second part of image fusion we represent the fusion model between the restored CT and MR images.

3.4.2.1 Image fusion procedure between MR and CT images

In this chapter, we illustrate an application of the proposed method for the special case of CT and MR images, where we want to include some geographic information of the fused object. The main function of this model is given in equation (71).

We have seen in Chapter 2 the effective usage of the quadratic constrained restoration methods in the degradation problem. The goal was to minimize the norm of the noise term, given in equation (60).

$$\|n\|^2 = \|y - h * x\|^2 \tag{60}$$

In other words, we wanted to find a vector $\hat{x}$ such that the difference between the known image **y** and the convolved image $h * x$ is minimized. The examined matrix $x(m,n)$ can be computed directly, through a mathematic model with inverse matrices etc, or iteratively through gradient descent methods. The restored image is estimated analytically by setting the derivative of the norm equal to zero. Afterwards, all the restored images by the MR and the CT images are fused with the method of the pointwise spatial frequency into a single MR and CT image.

With almost the same process, our main purpose in this chapter is to combine the fused MR and the fused CT images into a single image that contains the anatomical data and the bony structures of the input sources. Our main idea is to design an algorithm that takes a blank image as an input, and through a number of iterations it updates its elements from the MR and CT images. One of the most effective ways to perform this process is by minimizing the pixel-by-pixel differences between the blank image and the known MR and CT images.

Assume now that, the fused MR and CT images from the image fusion with the PSF model are given, respectively, from the two-dimensional signals $y_1(m,n)$ and $y_2(m,n)$, and the unknown combined image from matrix $x(m,n)$. According to the criterion, the estimation of vector $\hat{x}$ can be performed through the following equation:

$$\hat{x} = \arg\min_x \{J(x)\} \tag{61}$$

with

$$J(x) = \sum_{m,n} \left(y(m,n) - x(m,n)\right)^2 \tag{62}$$

or in a vector form

$$J(x) = \sum_j \left(y_j - x_j\right)^2 \tag{63}$$

Our aim is to find a mixed model that contains the fused MR image and the fused CT image. Receiving the knowledge from both MR and CT images, this model can

update the elements of the unknown matrix $x(m,n)$ from both images, as it is shown in equation (64).

$$\hat{x} = \arg\min_{x} \left\{ J\left(x\right) \right\} \tag{64}$$

with

$$J\left(x\right) = a_1 \sum_{m,n} \left(y_1\left(m,n\right) - x\left(m,n\right) \right)^2 + a_2 \sum_{m,n} \left(y_2\left(m,n\right) - x\left(m,n\right) \right)^2 \tag{65}$$

or in a vector form

$$J\left(x\right) = a_1 \sum_{j} \left(y_{1,j} - x_j \right)^2 + a_2 \sum_{j} \left(y_{2,j} - x_j \right)^2 \tag{66}$$

Variables a_1 and a_2 are equal to 0.5 in order the fused MR and CT images to have a parity participation in the equation. Note that the elements that correspond to the anatomical data of the MR image and the bony structures of the CT image, are participating equally with elements of the background, in the combination of matrix $x(m,n)$. This feature might be inefficient, because the goal of this algorithm is the final combined image to obtain only the anatomical data from the MR image and the bony structures from the CT image, without deformations from the background of the images.

The image segmentation method, a powerful method for identifying or classifying objects into different groups, is the pre-processing model in our image fusion process. According to this method, we can separate the background regions of the image and the regions correspond to the anatomical data of the MR image or the bony structures of the CT image. Assume that the region-segmented images by the MR image and the CT image are represented by matrices $S_1\left(m,n\right)$ and $S_2\left(m,n\right)$. The elements of the matrices $S_1\left(m,n\right)$ and $S_2\left(m,n\right)$ that belong to regions where we have knowledge about the anatomical data or the bony structures of the image, are equal to '1'. On the contrary, elements that belong to the background area have zero value. Applying the region-segmented image in equation (65), the updated equation is taking the following form:

$$J(x) = a_1 \sum_{m,n} S_1(m,n) \cdot \left(y_1(m,n) - x(m,n) \right)^2 + a_2 \sum_{m,n} S_2(m,n) \cdot \left(y_2(m,n) - x(m,n) \right)^2$$

(67)

The scope of equation (67) is that we can select the interest regions from each image to participate 100% to the restoration of matrix $x(m,n)$, cutting the regions of the image that obtain no knowledge, or the knowledge of them isn't necessary for the final fused image. When an element of the segmented image is equal to zero, the algorithm blocks the affect of the term $\left(y_1(m,n) - x(m,n) \right)^2$, in the current position of the segmented element, in the final combined image. Whereas, when an element of the segmented image is equal to one, the algorithm permits the participation of the term $\left(y_1(m,n) - x(m,n) \right)^2$ in the restoration of matrix $x(m,n)$.

An extended version of the proposed algorithm is to compute the percentage of each element that participates in the computation of the unknown matrix $x(m,n)$, relatively to the intensity of this element. This approach uses the real matrices $S_1(m,n)$ and $S_2(m,n)$ in the region $[0,\text{K},1]$ instead of just binary values. To perform such a concept, the proposed algorithm performs a pixel-by-pixel multiplication between the binary-segmented image and the input image, before applying the segmented image in equation (67). Performing such a calculation, the result, is an image that contains pixels with zero values in the background regions, and with intensities up to 255 in the interested areas. Moreover, we have seen that matrices $S_1(m,n)$ and $S_2(m,n)$ function as coefficients in equation (67), with elements that have binary values.

Maintaining this rule, the elements that correspond to the anatomical data of the MR image or the bony structures of the CT image must be normalized their values in region $[0,\text{K},1]$. The above normalization can be done easily in the matrix, by dividing each element with the highest value of the examined matrix. The goal succeed by this scheme is filtering of the knowledge that we receive from the fused MR image and the CT image, together with the computation of the percentage that each element of the fused images is participating in the combined matrix $x(m,n)$.

Finally, the whole procedure can be rewritten in the following form, where function $J(x)$ can estimate the combined image through the model that it will be discussed above. It is important to emphasize that all the matrices in equation (67) are transformed to vectors, with variable j to be the pointer of each element of them.

$$J(x) = \alpha_1 \cdot \sum_j S_{1j} \left(y_{1j} - x_j \right)^2 + \alpha_2 \cdot \sum_j S_{2j} \left(y_{2j} - x_j \right)^2 \tag{68}$$

In the above determination, we have seen the utility of the image segmentation method where background regions are filtered, and the knowledge of the anatomical data and the bony structures is participating without deformations in the combined image. The idea of using anatomical information in computed tomography is not new. Many works on the subject has been done before **[17-20]**. But, combining both region and border information from anatomic data is new. The separation of the background regions from the data regions in both MR and CT images is accomplished through the image segmentation model.

The objective function $J(x)$, can be expanded with an extra term that preserves uniform information on matrix $x(m,n)$ in homogenous region. This can be achieved through the following equation.

$$\lambda \cdot \sum_j \left(1 - \overset{)}{x}_j \right) \cdot \varphi \left(x_j - \overline{x}_j \right) \tag{69}$$

The binary matrix $\overset{)}{x}(m,n)$, reflects the borders (edges) of important regions in the image. This real matrix $\overset{)}{x}(m,n)$, can be considered as a geometrical (borders) data. The computation of this matrix is based on concepts of edge detection. First, all the elements of the binary matrix $\overset{)}{x}(m,n)$ have zero values. Applying the edge detection algorithm to the blank image the result would be a blank image with no edges or borders. While the algorithm is updating the knowledge of the unknown matrix $x(m,n)$, the elements of matrix $\overset{)}{x}(m,n)$ remain equal to zero everywhere, expected in areas of object borders where $\overset{)}{x}(m,n) = 1$. The ϕ function, is a convex function and the parameter λ controls the percentage of the term that participates in the whole procedure. The proposed edge detection method is described analytically in the following chapters.

Note that, when ϕ is chosen to be convex, and when the parameter $\lambda > 0$ is fixed and the rest data are given, this criterion is a convex function of **x**. A simple choice of ϕ is a quadratic $\phi(x) = x^2$ or power form $\phi(x) = x^{\beta}$. This convex function in this thesis has the following form:

$$\phi(x) = \sqrt{1 + x^2} \tag{70}$$

Finally, $\overline{x}_j$ is the mean value of pixels x_j in the 3x3 neighborhood of the pixel j. Based on these assumptions, the final procedure has the following form.

$$J(X) = \alpha_1 \cdot \sum_j S_{1j} \left(g_{1j} - x_j\right)^2 + \alpha_2 \cdot \sum_j S_{2j} \left(g_{2j} - x_j\right)^2 + \lambda \cdot \sum_j \left(1 - \overline{x}_j\right) \cdot \varphi\left(x_j - \overline{x}_j\right) \tag{71}$$

The computation of the unknown matrix $x(m,n)$ in equation (71) can not be performed analytically, as in the quadratic constrained restoration models in Chapter 2. The convex function ϕ, includes the matrix $x(m,n)$ inside the square root and the analytical estimation of $x(m,n)$ is impossible. Moreover, the power of the pixel-by-pixel subtractions in equation (71) sometimes might be a float value than a square power. For that reasons, an iterative method of combination is essential.

There is a variety of restoration methods, and one of them is the iterative detection of the unknown matrix $x(m,n)$ through the gradient descent optimization model. Then, the optimization of the objective function can be done by any gradient based algorithm. The whole reconstruction procedure is the following:

1. Initialize $x = x^{(0)} = 0$.
2. Compute x by optimizing the equation (71), with respect to the gradient based algorithm.
3. Compute the edge detected and the mean valued matrices of x, from x.
4. Return to 2 until convergence.

The graph correspond to the previous steps of reconstruction is shown below.

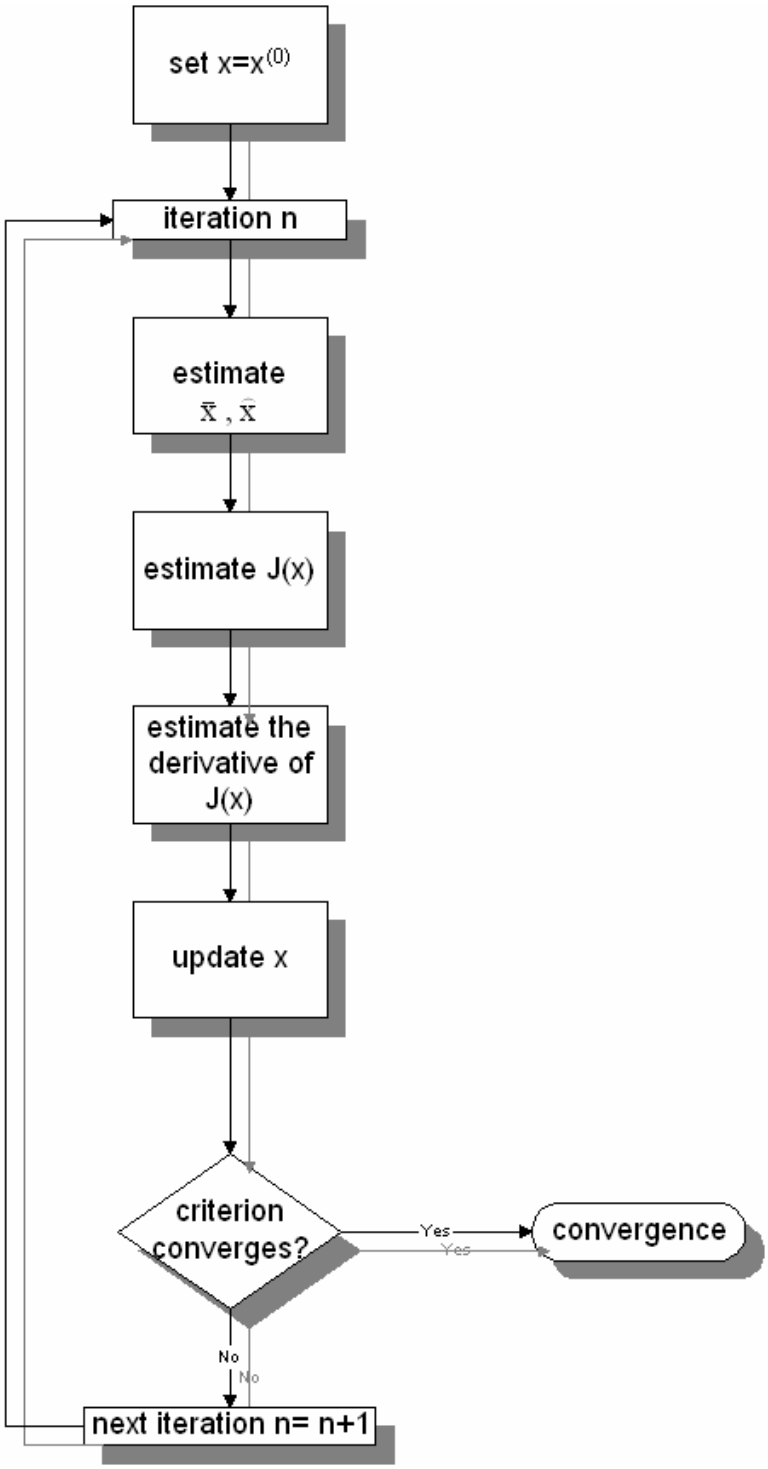


Figure 3:33 Gradient descent algorithm

In the next Chapters, the gradient descent optimization model is represented analytically. Applying the gradient descent model to the fused MR and CT images, with no knowledge about the final combined image, the result is given in Figure 3:34.

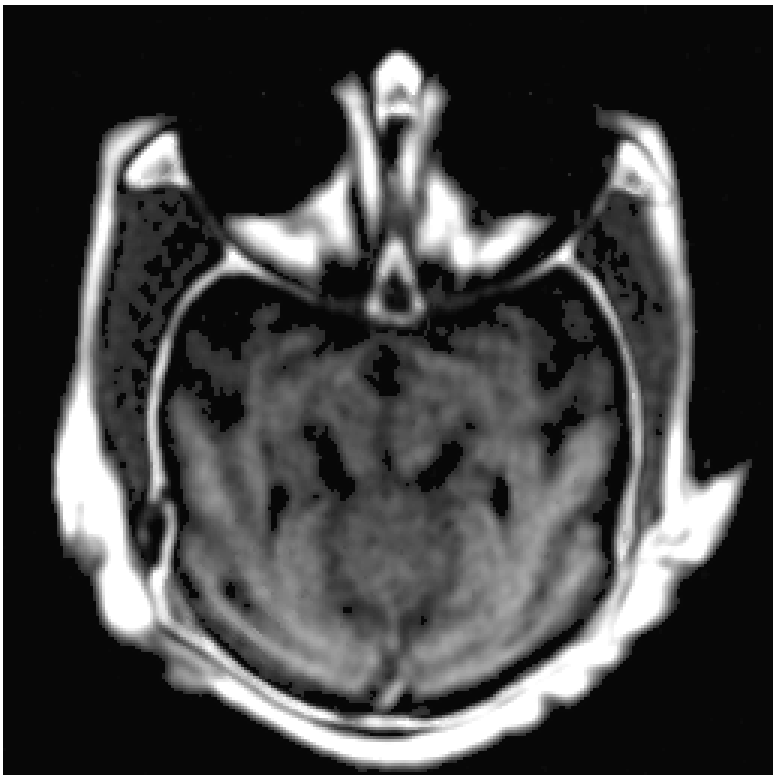


Figure 3:34 Final combined image from MR and CT images

3.4.2.2 Edge detection

For the detection of edges in images, various detection operators are used. Most of these are applied with convolution masks based on differential operations. Differentials for edge detection are used to define color or brightness changes of pixels and their change directions. If there is an abrupt change of brightness in a short interval in an image, it means that in that interval there is an edge with great probability.

Many of the edge operators are negatively affected by noise in the images. Thus, before edge detection, noise should be filtered as much as possible. Instead of this, Frequency domain filters may also be used for edge detection. Only spatial domain methods are being considered in this thesis.

The basic idea of edge detection operators is based on comparison of the brightness values of pixels with their neighbors. For example, if a pixel's value is the same as its neighbors or too close, then with a great probability this pixel will not lie on an edge. If there is difference between pixel and its neighbors then it may be thought that this pixel may lie on an edge. An ideal edge appears with an abrupt gray-

level change. But in practice, instead of such abrupt change, smooth changes occur. In Figure 3:35, an ideal and corresponding practical situation of gray-level changes of an edge has shown.

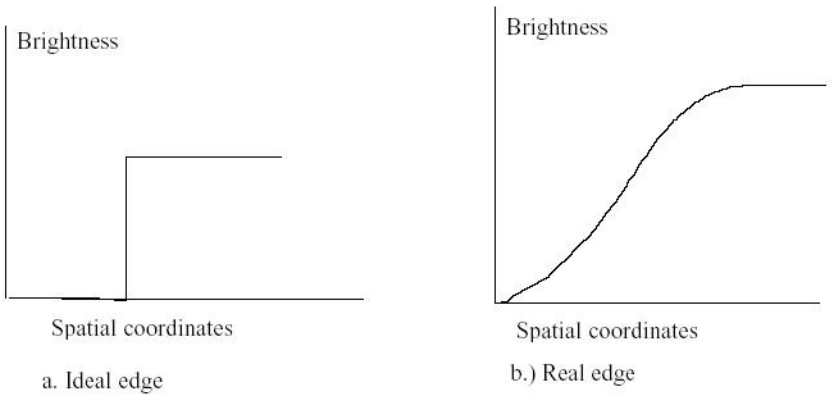


Figure 3:35 Ideal and real edge in an image

In the following sections some edge detection operators have been explained. The method that this thesis has performed is the edge detection with four dimensional Kirsch masks. Note that the description of each edge detection method is according to reference **[21]**.

3.4.2.2.1 Sobel Operators

Sobel edge detection masks detect vertical and horizontal edges separately. And these directional edges are combined finally. Coefficients of 3x3 Sobel masks are:

$$\text{Horizontal Mask} = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} \tag{72}$$

$$\text{Vertical Mask} = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \tag{73}$$

These two masks are convoluted with image separately. The magnitude and direction of an edge is calculated by using convolution results of two masks. These are:

$$S = \sqrt{S_1^2 + S_2^2} \text{ and } a=\tan^{-1}\left[\frac{S_1}{S_2}\right] \quad (74)$$

3.4.2.2.2 Prewitt Operators

Prewitt operators are similar to Sobel, but the mask coefficients are different.

$$\text{Horizontal Mask} = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} \quad (75)$$

$$\text{Vertical Mask} = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix} \quad (76)$$

Magnitude and direction of an edge is found by:

$$S = \sqrt{P_1^2 + P_2^2} \text{ and } a=\tan^{-1}\left[\frac{P_1}{P_2}\right] \quad (77)$$

3.4.2.2.3 Laplace Operators

Laplace operators are used as the ones described above. Laplace operators are symmetric; all changes in all directions affect the result. Sign of the result defines the direction of gray level change. Below Laplacian masks has shown.

$$\begin{pmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{pmatrix} \text{ and } \begin{pmatrix} 1 & -2 & 1 \\ -2 & 5 & -2 \\ 1 & -2 & 1 \end{pmatrix} \quad (78)$$

3.4.2.2.4 Kirsch Operators

The Kirsch edge detector masks can be used to detect edges along different directions. Such masks of size 3x3 are:

$$mask1 = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix} \text{ for "} \leftrightarrow \text{" horizontal edge detection} \tag{79}$$

$$mask2 = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{pmatrix} \text{ for "b " vertical edge detection} \tag{80}$$

$$mask3 = \begin{pmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{pmatrix} \text{ for "\ " diagonal edge detection} \tag{81}$$

$$mask4 = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -1 \end{pmatrix} \text{ for "_ " diagonal edge detection} \tag{82}$$

Edge detection can be performed in the following way. All the Kirsch edge detector masks are applied to each image pixel. The mask that produces the maximal output value is the one that gives the information about each pixel of the image. A schematic and the steps that the algorithm is following are shown below.

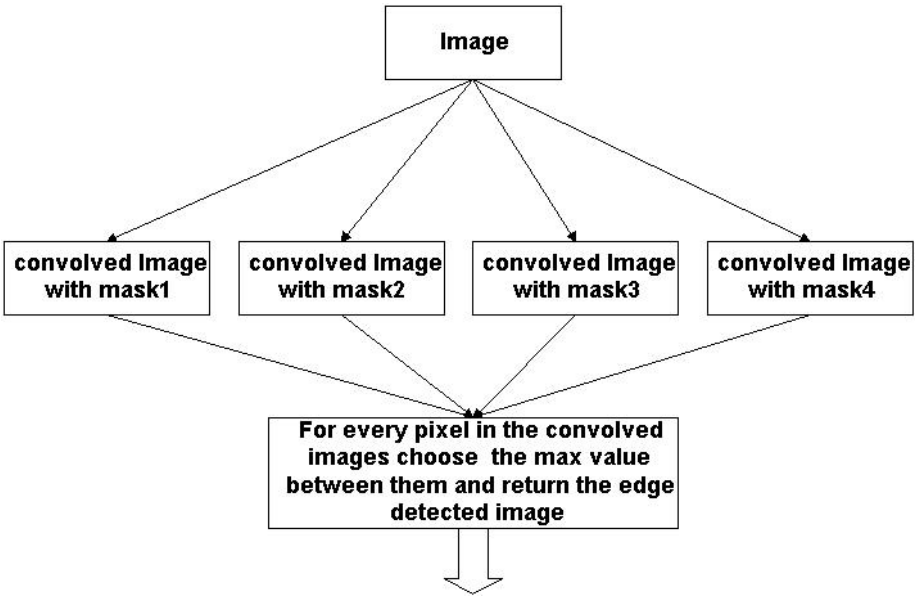


Figure 3:36 Edge detection method

3.4.2.2.5 Results of the edge detection method

After simulating the above methods with the input MR and CT images, we have made the decision to perform the Kirsch edge detection and we present the results of it.

3.4.2.2.5.1 Results from edge detection applied to magnetic resonance image

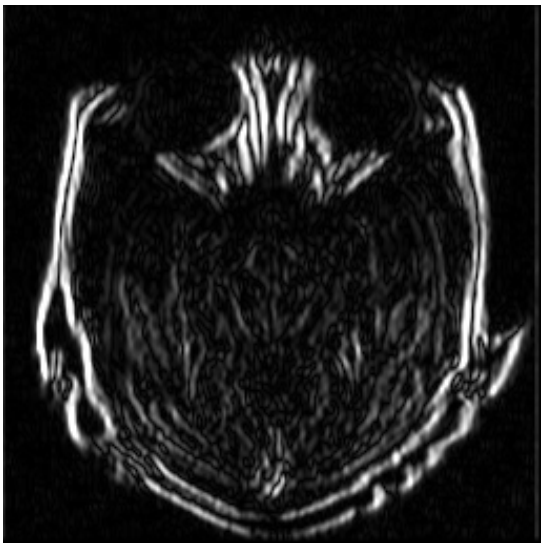


Figure 3:37 Horizontal edge detection of MR image

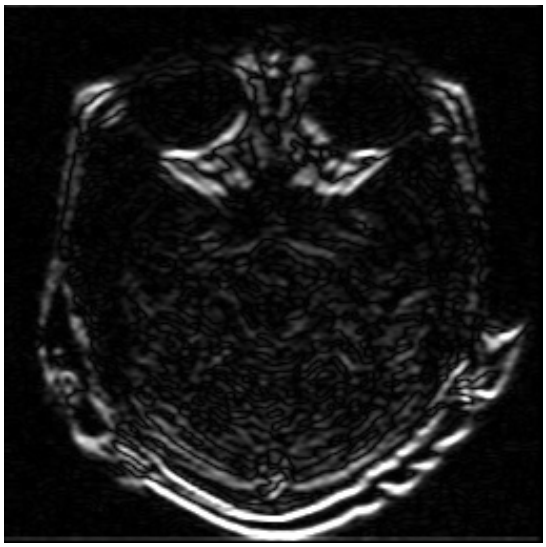


Figure 3:38 Vertical edge detection of MR image

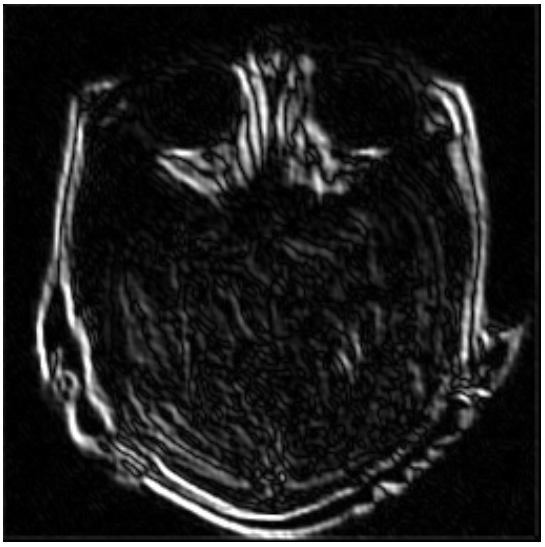


Figure 3:39 Diagonal '/' edge detection of MR image

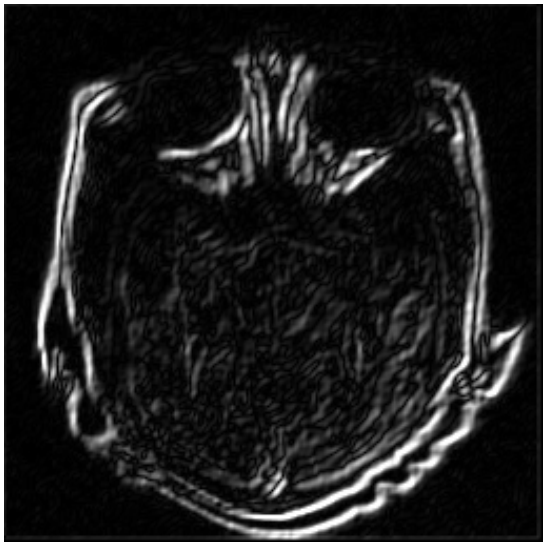


Figure 3:40 Diagonal '\' edge detection of MR image

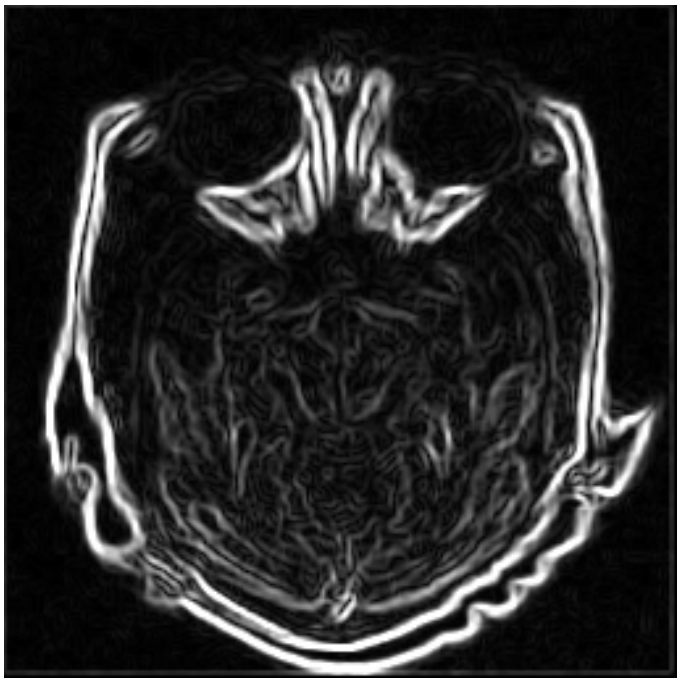


Figure 3:41 Complete edge detection of the MR image

3.4.2.2.5.2 Results from edge detection applied to computed tomography image

Figure 3:42 Horizontal edge detection of CT image

Figure 3:43 Vertical edge detection of CT image

Figure 3:44 Diagonal '/' edge detection of CT image

Figure 3:45 Diagonal '\' edge detection of CT image

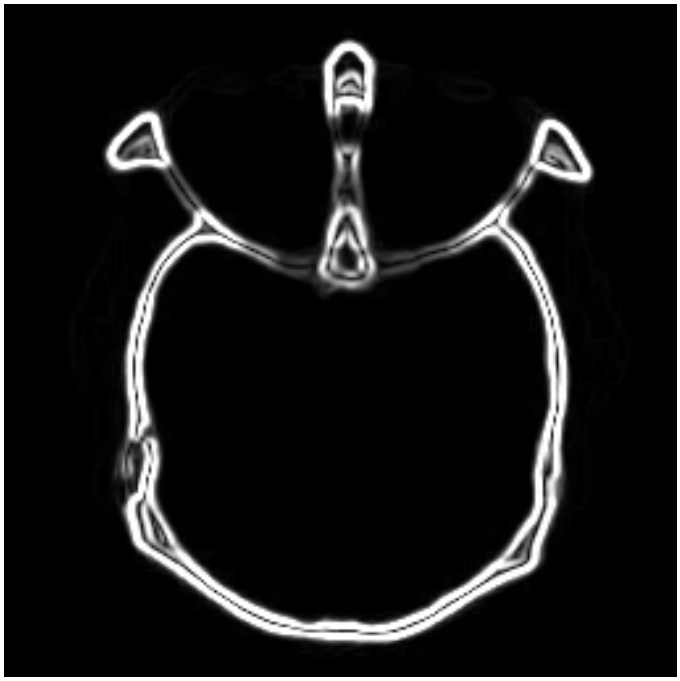


Figure 3:46 Complete edge detection of CT image

3.4.2.3 Image segmentation with thresholding

3.4.2.3.1 Introduction to image segmentation with thresholding

Image segmentation refers to the partition of a digital image into disjoint connected sets of pixels, each of which corresponds to an object or region. Image segmentation can be approached from three different philosophical perspectives **[13]**. In the region approach, one assigns each pixel to a particular object or region. In the boundary approach, one attempts only to locate the boundaries that exist between the regions. In the edge approach, one seeks to identify edge pixels and then link them together to form the required boundaries.

In this thesis, the region approach is the one that we are interesting in and the method that we have performed for region segmentation is the segmentation by thresholding.

Thresholding is a particularly useful region-approach technique for scenes containing solid objects upon a contrasting background. Specifically, in the case of the CT image the contrast between the background and the object is visible. When using a threshold rule for image segmentation, one assigns all pixels at or above the threshold grey level to the object. All pixels with grey level below the threshold fall outside the object. The boundary is then that set of interior points, each of which has at least one neighbour outside the object. The implementations of image segmentation by thresholding are separated in the following applications.

3.4.2.3.1.1 Global thresholding

In the simplest implementation of boundary location by thresholding, the value of the threshold grey level value is held constant throughout the image. If the background grey level is reasonably constant throughout, and if the objects all have approximately equal contrast above the background, then a fixed global threshold will usually work well. The general rule of image segmentation with global thresholding is based on equation(83). Segmentation is accomplished by scanning the image pixel-by-pixel and labelling each pixel as object or background, depending on whether the

grey level of that pixel is greater or less than the value of threshold. Note that, for every pixel (i, j) with $f(i, j)$ the input image and $g(i, j)$ the output image:

$$\begin{gathered} g(i, j) = 1 \text{ ,for } f(i, j) \geq Threshold \\ or \\ g(i, j) = 0 \text{ ,for } f(i, j) < Threshold \end{gathered} \tag{83}$$

The success of this method depends entirely on how well the histogram of the image can be partitioned.

3.4.2.3.1.2 Adaptive thresholding

In many cases, the background grey level is not constant, and the contrast of objects varies within the image. In such cases, a threshold that works well in one area of the image might work poorly in other areas. In these cases, it is convenient to use a threshold grey level that is a slowly varying function of position in the image.

3.4.2.3.1.3 Optimal threshold selection

Unless the object in the image has extremely steep sides, the exact value of the threshold grey level can have considerable effect on the boundary position and overall size of the extracted object. This means that subsequent size measurements-particularly area-are sensitive to the threshold grey level. For this reason, we need an optimal, or at least consistent, method to establish the threshold. The Histogram techniques for image segmentation by thresholding, is the method that this thesis implement.

Suppose that the gray-level histogram corresponds to an image, $f(x)$, composed of light objects in a dark background, in such a way that object and background pixels have gray levels grouped into two dominant modes. One obvious way to extract the objects from the background is to select a threshold T that separates these modes. Then any point (x, y) for which $f(x, y) > T$ is called an object point, otherwise, the point is called a background point. Otsu, in 1979 proposed an algorithm for automatic threshold selection from the histogram of an image **[16]**.

The processing results, applying the previous image segmentation method with histogram thresholding to the MR and CT images, are represented in the following sub-chapter.

3.4.2.3.2 Results from the image segmentation

In the following sub-chapters, we represent the output images given by the image segmentation procedure. Note that we have experimented a lot with the decision of the threshold of each image, selecting the most effective threshold value for the CT and the MR image.

3.4.2.3.2.1 Results from image segmentation applied to MR image

In Figure 3:47, we can see the processing results of the image segmentation process, applied to the MR image. Note again that the choice of the threshold is based on the histogram of the MR image.

Figure 3:47 Segmentation of the MR image

In Figure 3:48, the segmented binary image is applied to the original image with a pixel-by-pixel multiplication between the pixels of the binary image and the pixel of

the original MR image. Performing such a processing in the original image, we automatically cut the regions that we don't want to receive any kind of information from them in later image processing procedures.

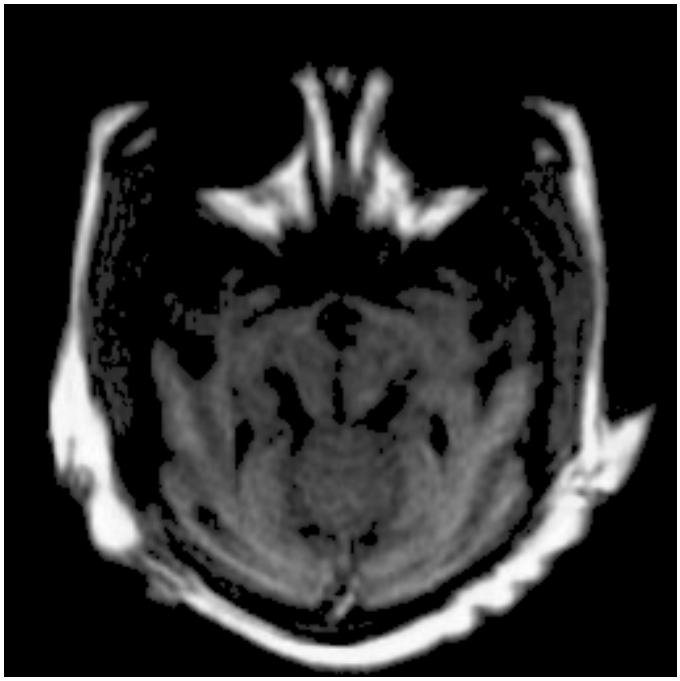


Figure 3:48 Segmentation applied to the MR image

3.4.2.3.2.2 Results from image segmentation applied to CT image

With the same steps as in the MR image, the processing results of the image segmentation algorithm applied to the CT image are the following.

Figure 3:49 Segmentation of the CT image

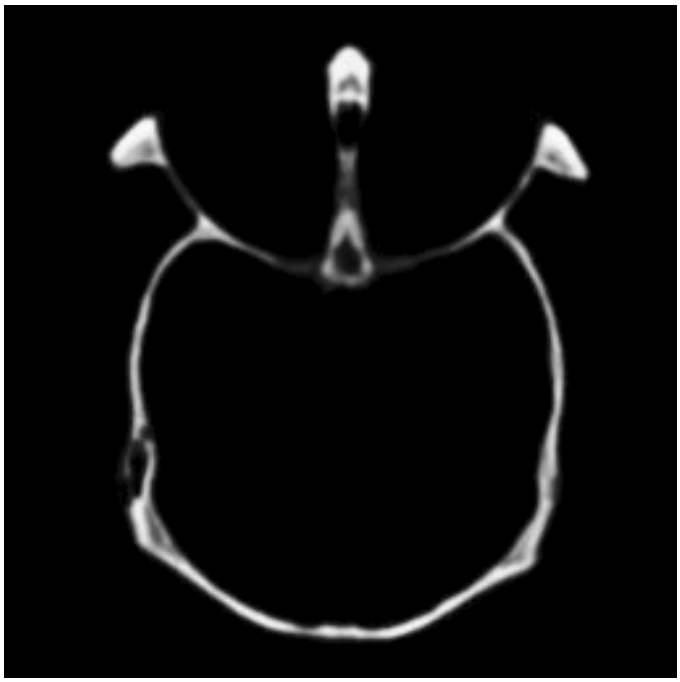


Figure 3:50 Segmentation applied to the CT image

3.4.2.4 Gradient descent method

The general optimization method in the second part of image fusion is the gradient descent algorithm. In the following sub-chapters we present the methodology of gradient descent method and the accuracy of the method in this thesis.

3.4.2.4.1 Methodology of gradient descent method

Gradient descent is a function optimization method, which uses the derivative of the function and the idea of steepest descent **[11]**. The derivative of a function is simply the slope. The slope is just $\frac{\Delta y}{\Delta x}$ in the diagram.

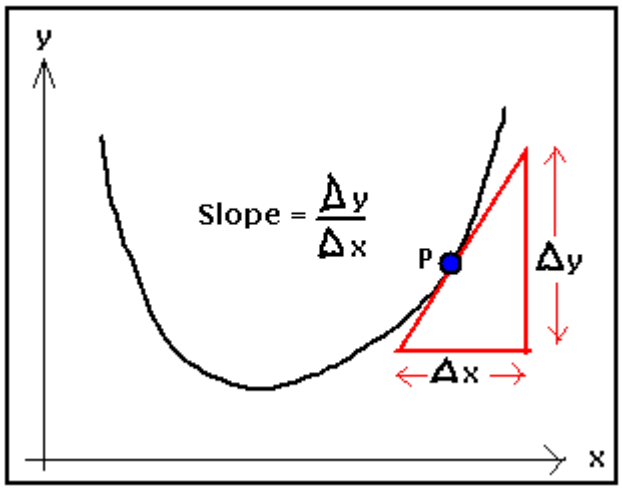


Figure 3:51 slope of the function f(x)

So if we know the slope of a function, then we can move in the negative direction of the slope, and reduce the value of the function. Gradient descent is an iterative method, so the main idea is as follows:

- Compute the derivative of the function with respect to its independent variables. We can denote this derivative as $\nabla f(x)$, where $f(.)$ is the function to be minimized, and **x** is the vector of independent variables.

- Change the value of **x** as follows, $x_{n+1} = x_n - a \cdot \nabla f(x)$, where the subscript n refers to the iteration number, and α is a step size which must be chosen so that we don't take too big or too small of a step. Too big of a step will overshoot the function minimum, and too small of a step will result in a long convergence time.
- Repeat the above two steps until we converge to a local minimum of the function $f(x)$.

Gradient descent is an attractive optimization method in that it is conceptually straightforward and often converges quickly. Its drawbacks include the fact that the derivative of the function must be available, and it converges to a local minimum rather than a global minimum.

The convergence criterion for the gradient descent method in this thesis is based on the changes of the estimated matrix. Specifically, the iterative algorithm stops when the change between the x_{n+1} and the x_n two-dimensional matrix is lower than a definite value. The equation for the above criterion is:

$$\frac{\|x_{n+1} - x_n\|}{\|x_n\|} \geq 0.001 \ , n = 1, \mathrm{K} \ , N \text{ steps} \tag{84}$$

Note that the norm of a two-dimensional matrix is given by the following equation:

$$\|x\| = \sqrt{\sum_{i=1}^{N} \sum_{j=1}^{M} x_{i,j}^2} \tag{85}$$

3.4.2.4.2 Accuracy of the gradient descent method

The main goal of the gradient descent method is to minimize each function $f(.)$ by updating its variable (or variables) **x** with respect to the equation:

$$x_{n+1} = x_n - a \cdot \nabla f(x) \tag{86}$$

Assume that we are simulating function (87). Note again that the fused MR and CT images are represented by the two-dimensional matrices y_{1j} and y_{2j}, S_{1j} and S_{2j} are the extended segmented matrices, and matrix x corresponds to the unknown estimate matrix.

$$J(x) = \alpha_1 \cdot \sum_j S_{1j} \left(y_{1j} - x_j \right)^2 + \alpha_2 \cdot \sum_j S_{2j} \left(y_{2j} - x_j \right)^2 \tag{87}$$

Following the steps of the gradient descent algorithm, and initializing the unknown pixel values of matrix x to be equal to zero (blank Image), it is visible that the value of $J(x)$ is decreasing while the number of the iterations is increasing. The next graph has been taken while gradient descent method was evolving. According to the simulation, i.e. graph in Figure 3:52, the gradient descent method we simulated is reliable and convergent. Note that the study of the algorithm has been performed many times, with different input data.

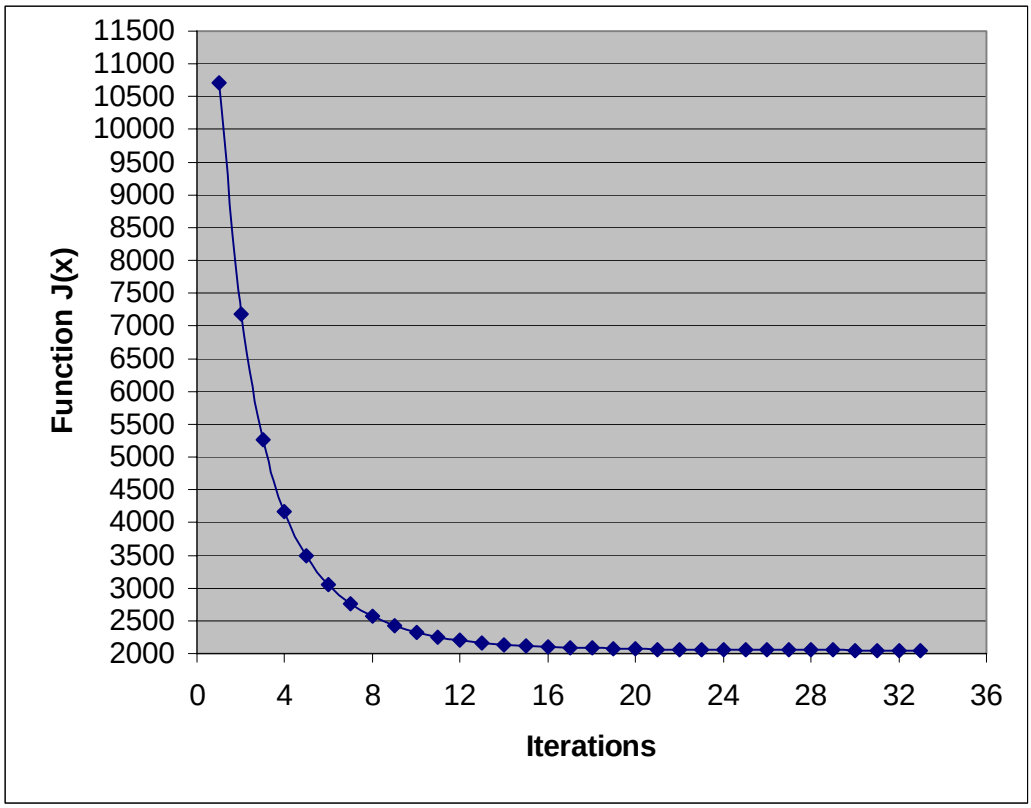


Figure 3:52 Graph of the gradient descent simulation

3.5 Conclusion of information fusion

In this chapter we have proposed the image fusion method between the MR and the CT images. In the first part, image fusion with the pointwise spatial frequency has positively acted on the MR and CT images, so that the deformation that affects the images diminishes. The processing results prove how successful our approach is. The proposed image fusion with the spatial prequency, and specifically the intermediate step we have implemented, has improved the intensity of the edges and the local characteristics of the object.

The first part of the thesis procedure has prepared the field of the joint processing of the two restored MR and CT images. The main objective of this thesis is to focus on how classical probabilistic methods such as Bayesian (BAYES) approaches can be used for data fusion. In the second part of image fusion, we have implemented several versions of this fusion process. The image segmentation method has proved to be a powerful tool in order to select the interesting regions of each input image.

The edge detection method applied to the fused image provides good results on the borders of different regions. The final combined image contains information about the anatomical data and the bony structures of the object, corresponded in the fused image.

4 Discussion and further work

4.1 Discussion

In the second chapter we focus on how restoration methods can function positively in order to overcome the effect of degradation in the raw data. We present an introduction of the problem and a review of related search in order to get a grasp of current trends in image restoration. Different measurement systems with known degradation masks and unknown white Gaussian noise have degraded the magnetic resonance and computed tomography images.

Afterwards, we apply the joint and independent quadratic constrained restoration methods to the images, producing a set of restored data ready for the next part, the image fusion procedure.

In the next chapter, we focus on the image fusion method applied to the restored images. In the first level, the image fusion with the pointwise spatial frequency fuses separately the MR and CT sets of images obtained from the multi-sensor architecture information system. The processing results of the proposed method indicate good success of our approach.

In the next level, the restored magnetic resonance image is processed jointly with the restored computed tomography, providing a new source of knowledge containing information from both images. Despite the fact that the input sources didn't allow us to have a quality database to experiment, the combination of edge detection, image segmentation and the gradient descent method offers a fast and reliable method of image fusion.

In this concluding chapter, we give an evaluation of our ideas and attempt to extend them in order to serve as a starting point for future search.

4.2 Further work

Although good results are obtained, some open issues remain. These are outlined below with possible indications on how they might be solved.

Starting with the image restoration algorithm, it is possible to extend our research into different high-pass filters in order to study how they affect the process.

Moreover, we have seen in this thesis that the independent restoration method applied to noisy motion blurred image has caused a lot of deformations to the restored image. Changing the general method that this thesis ensued is not proposed. The constrained restoration method is a fast and reliable method for image restoration and it should not be changed. Keeping the general form of the constrained method as it is, and enhancing it with extended terms in order to avoid the ringing phenomenon might be really productive.

One possible approach worth of further investigation is the optimization of the image segmentation algorithm. Although the current algorithm works well and the processing results from the MR and CT images prove this fact, an improved image segmentation method would divide the image into small sectors and work separately on them. In the first pass, the boundaries of the image can be defined using a gray-level threshold that is constant within sectors, or might change for the various sectors. The objects so defined are not extracted from the image, but the interior mean gray level of each object is computed. On a second pass, each object defines its own threshold in between its interior gray level and the background gray level of its principal sector.

This thesis has designed and simulated in C code, the Laplacian Pyramid, referenced in **[14]**. The advantage of this code is that it is well suited for many image analysis methods, where the image is expanded in many levels. A future optimization of this thesis is to absorb this technique and to perform the image fusion procedure in many levels of the examined images.

An additional natural extension of our method would be to include functional types of images like single-photon emission computerized tomography (SPECT) to the image fusion procedure. SPECT antibody images can enhance the information provided by either single modality by providing precise anatomical-functional correlation.

Finally, another possible approach should be to extend our research in the 3-D where slices of magnetic resonance images are fused with computed tomography images of the same scene. This kind of research needs a lot of discussion and should be checked separately.

Appendix A

Add_noise.h

In this header file we have created and compiled the following function:

- function wgn (long x_dimension, long y_dimension, float power) which generates a two-dimensional MxN matrix of white Gaussian noise. Variable "power" specifies the power of the output noise in dB. The function calls function "randn" which generates arrays of random numbers whose elements are normally distributed with mean 0, variance $\sigma^2=1$ and standard deviation $\sigma=1$.

Alloc_mem.h

This header file includes the following function:

- function alloc_2D_matrix (long x_dimension, long y_dimension) which allocates memory for a two-dimensional matrix. If the system fails to allocate the memory, a warning appears in the screen and the program terminates.

Bmp1.h

This header file includes structs with all the necessary information about the RGB of an image, its size and its type. Note that the specific program works with 256 grayscale bmp images.

Struct.h

```
struct st
{
    float **real_part; float **imaginary_part; long x_dimension; long y_dimension;
};
```

In our project, we have worked with complex numbers which are produced by the discrete Fourier transform (DFT), applied in an image. In order not to lose any information and to have well organized our data, we have created a struct that splits the complex number and store the imaginary and the real part of it into two two-dimensional matrices. Besides the two-dimensional matrices, variables x_dimension and y_dimension corresponds to the Height and Width of the matrix.

Fft.h

Fourier analysis is extremely useful for data analysis, as it breaks down a signal into constituent sinusoids of different frequencies. For sampled vector data, Fourier analysis is performed using the discrete Fourier transformation (DFT).

The fast Fourier transformation (FFT) is an efficient algorithm for computing the DFT of a sequence; it is not a separate transform. It is particularly useful in image processing, where its uses range from convolution as we have seen and in our work.

As it concerns, in this header file we have created and compiled the following functions:

- function DFT (int direction, long x_dimension, long y_dimension, float **real_part, float **imaginary_part) returns the discrete Fourier transform (DFT) or the inverse discrete Fourier transform (DFT) of matrix X, computed with a fast Fourier transform (FFT) algorithm. The two-dimensional M-by-N DFT and inverse M-by-N DFT relationships are given by the following equations. Note that M and N are the Height and Width of the two-dimensional matrix and X the result discrete Fourier transform (DFT) of matrix x. The function DFT() returns the discrete Fourier transform (DFT) or

the inverse discrete Fourier transform (DFT), depends on the value that the variable “int direction” has. If the “direction”=1, the function returns the discrete Fourier transform (DFT) of the matrix. If “direction”=-1, the function returns the inverse discrete Fourier transform (DFT) of the matrix. The “real_part” and the “imaginary_part ”matrix returns the real part and the imaginary part of the complex result number.

$$X(p,q)=\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}x(m,n)\cdot e^{-j\left(\frac{2\cdot\pi}{M}\right)\cdot p\cdot m}\cdot e^{-j\cdot\left(\frac{2\cdot\pi}{N}\right)\cdot q\cdot n} \tag{88}$$

$$x(m,n)=\left(\frac{1}{M\cdot N}\right)\sum_{p=0}^{M-1}\sum_{q=0}^{N-1}X(p,q)\cdot e^{j\left(\frac{2\cdot\pi}{M}\right)\cdot p\cdot m}\cdot e^{j\left(\frac{2\cdot\pi}{N}\right)\cdot q\cdot n} \tag{89}$$

- function dft2 (int direction, long x_dimension, long y_dimension, float **real_part, float **imaginary_part) gets as input the output of the previous function DFT() and finally returns the two-dimensional discrete Fourier transform (DFT) or the inverse two-dimensional discrete Fourier transform (DFT)of the matrix, Computed with a fast Fourier transform (FFT) algorithm. The result has the same size as the input matrix. As in the previous function, the variable “direction” works as a flag.

Filters.h

This header file includes the following functions:

- function filter1 (long x_dimension, long y_dimension) returns the 3x3 blurring mask of the system. Note that the function takes as inputs the Width and the Height of the image to be convolved, and fill the 3x3 mask with zero values so as to have the same Height and Width with the image to be convolved. Below is the 3x3 mask that we have taken after the simulation of the system that has given the noisy blurred image.

$$blurred\_mask = \begin{bmatrix} \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \end{bmatrix} \quad (90)$$

- function filter2 (long x_dimension, long y_dimension) returns the 1x5 motion blurred mask of the system. Note that the function takes as inputs the Width and the Height of the image to be convolved, and fill the 1x5 mask with zero values so as to have the same dimensions with the image to be convolved. Below is the 1x5 mask that we have taken after the simulation of the system that has given us the noisy motion-blurred image.

$$motion\_blurred\_mask = \begin{bmatrix} \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \end{bmatrix} \quad (91)$$

- function filter3 (long x_dimension, long y_dimension) returns the 3x3 Laplacian mask. Note that the function takes as inputs the Width and the Height of the image to be convolved, and fill the 3x3 mask with zero values so as to have the same dimensions with the image to be convolved. Below is the 3x3 mask that we have used to do our restoration of the original image.

$$Laplacian\_mask = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad (92)$$

Rw.h

This header file includes the following functions:

- function read_bmp (char* filename, long *Height, long *Width) takes as input the filename of a bmp image and returns the intensity of each pixel in the

current image in a two-dimensional matrix. The Height and Width of the matrix are equal with the Height and Width of the image.

- function write_bmp (char *filename, unsigned char** pic, long Height, long Width) takes as inputs the filename where we want to store the bmp image, a two-dimensional matrix corresponds to the intensity of the specific image, and the Height and Width of the matrix.

Math_func.h

This header file includes the following functions:

- function zero_vect (long x_dimension, long y_dimension) which calls the function to allocate memory for a two-dimensional matrix and finally returns a matrix, with Height and Width given by the program, filled with zero values. This function is very useful and practical because we can empty a matrix safety, without causing troubles to any stored values in the next steps of the program.

- function transpose (long x_dimension, long y_dimension, float **data, int choice) gets a two-dimensional matrix and returns it's transpose matrix. Note that we have to convolve every time the images with the masks, described above, and we work with complex numbers, as it concerns from the Fourier transform. The transpose of the two-dimensional complex matrix relationship is given in equation (93). We have split the real and the imaginary part of any number and we work individually with them, with respect to the mathematical laws. Depends on the choice that we give as input to the function, it returns the transpose of the real part of the input vector or the transpose of the imaginary part of the input vector. When the "choice"=1, the function performs the equation (94) and returns the transpose of the real part of the input two-dimensional matrix. When the "choice"=-1, the function performs the equation (95) and returns the transpose of the imaginary part of the input two-dimensional matrix.

$$\begin{pmatrix} a_{11}+i\cdot b_{11} & \text{K} & a_{1n}+i\cdot b_{1n} \\ \text{M} & \text{O} & \text{M} \\ a_{m1}+i\cdot b_{m1} & \text{L} & a_{mn}+i\cdot b_{mn} \end{pmatrix}^{T} = \begin{pmatrix} a_{11}-i\cdot b_{11} & \text{K} & a_{m1}-i\cdot b_{m1} \\ \text{M} & \text{O} & \text{M} \\ a_{1n}-i\cdot b_{1n} & \text{L} & a_{mn}-i\cdot b_{mn} \end{pmatrix} \quad (93)$$

$$\begin{pmatrix} a_{11} & \text{K} & a_{1n} \\ \text{M} & \text{O} & \text{M} \\ a_{m1} & \text{L} & a_{mn} \end{pmatrix}^{T} = \begin{pmatrix} a_{11} & \text{K} & a_{m1} \\ \text{M} & \text{O} & \text{M} \\ a_{1n} & \text{L} & a_{mn} \end{pmatrix} \quad (94)$$

$$\begin{pmatrix} b_{11} & \text{K} & b_{1n} \\ \text{M} & \text{O} & \text{M} \\ b_{m1} & \text{L} & b_{mn} \end{pmatrix}^{T} = \begin{pmatrix} -b_{11} & \text{K} & -b_{m1} \\ \text{M} & \text{O} & \text{M} \\ -b_{1n} & \text{L} & -b_{mn} \end{pmatrix} \quad (95)$$

- function inverse (float ** output_real_part, float ** output_imag_part, float **input_real_part, float **input_imag_part, long x_dimension ,long y_dimension) takes as inputs the real and the imaginary part of a two-dimensional matrix, the Height and the Width of the matrix and returns separately the inversed real and imaginary part of each object of the matrix into a new two-dimensional matrix. Note that the result of the function is not the inverse of the matrix but the inverse of each object of the matrix as we are working pixel-by-pixel in the Frequency domain. The relationships for the inversed real and imaginary part for a complex number $y = \alpha + \imath \cdot \beta$ are given respectively by the following equations:

$$inverse(real\_part) = \frac{\alpha}{\alpha^2 + \beta^2} \quad (96)$$

$$inverse(imaginary\_part) = \frac{-\beta}{\alpha^2 + \beta^2} \quad (97)$$

Mean_value.h

This header file includes the following functions:

- function mean_val (float **data, long x_dimension, long y_dimension) which returns the mean value of each pixel in the 3x3 neighborhood, in a two-

dimensional matrix. The function takes as inputs a two-dimensional matrix where we want to find the mean value of its pixels as wells as its Height and Length. The function has a distinct characteristic when an object of the matrix is belonging to an area where its neighborhood supported by less than 8 objects (as we are calculating the mean value by a 3x3 neighborhood). Then, the equation of the function includes only the neighbor pixels and the result of the function is more optimized than the result of a simple mean value function. The general equation that the function works with is (98). In the special case where the pixel has a 3x3 neighborhood the equation is the following (99).

$$for \text{ every pixel do} : \mu_{i,j} = \frac{\sum neighbor \text{ pixels } \left(x_{i,j}\right)}{number \text{ of neighbor pixels}} \tag{98}$$

$$\mu_{i,j} = \frac{1}{9} \cdot \sum_{k=-1}^{1} \sum_{l=-1}^{1} x_{i-k,j-l} \quad where \ \mu_{i,j} \text{ is the mean value of the pixel } x_{i,j} \tag{99}$$

- function subtraction_func (long x_dimension, long y_dimension, float **subtrahend_matrix, float **subtracter_matrix) which performs a pixel-by-pixel subtraction between the two two-dimensional matrices. The result of the function is a two-dimensional matrix with the same Height and Width as the input matrices have.

Normalization.h

This header file includes the following functions:

- function normalization (float **data, long x_dimension, long y_dimension) which takes as inputs a two-dimensional matrix, its Height and Width and normalizes the matrix's pixels in region 0,K ,1 dividing each pixel with the pixel that has the maximum absolute value. Afterwards, it calculates the subtraction between each pixel and the variable 1 and returns the result into a two-dimensional matrix.

$$for \text{ every pixel do} : \hat{X}_{i,j} = \frac{X_{i,j}}{\max\left(X\right)} \tag{100}$$

$$for \text{ every pixel do} : \hat{X}'_{i,j} = 1 - \hat{X}_{i,j} \tag{101}$$

Variance.h

This header file includes the following functions:

- function Create_Histogram(float **Image, long x_dimension, long y_dimension) which takes as inputs the two-dimensional matrix corresponds to the intensity of each pixel of the image, its Height and Width and returns the histogram of the input image in a vector. The gray-level histogram gives the number of cells having a particular gray-level.

- function Segmentation(int choice, float **Image, long x_dimension, long y_dimension) which performs the histogram-based image segmentation using a manually choice of thresholding between the peak values in image's histogram. The function takes as inputs the image where we want to find its segmentation, its Height and Width and returns in a two-dimensional matrix the segmented image. This function calls the previous function "Create_Histogram" and based on the histogram of the image and selecting a threshold performs the calculations in equation (102) for each pixel (i, j) with $f(i, j)$ the input image and $g(i, j)$ the output image. Afterwards, we apply the segmented image to the input image, with pixel-by-pixel multiplication between the segmented image and the original. With this procedure, we pay attention to parts with semantic meaning in the image, which will lead to interpretation in later analysis. Finally, all the pixels of the two-dimensional image, corresponds to the pixel-by-pixel multiplication, are divided by the intensity of the pixel with the maximum value in order for the intensity of the pixels to have a value between 0 and 1. The input variable "choice" is a helpful variable for selecting whether the segmentation procedure is applied to

the magnetic resonance or to the computed tomography image. It is obvious that the MR and the CT input images have different histograms and the choice of the threshold should be different. If the variable "choice" is equal to 1 the function returns the segmentation of the computed tomography image and when variable "choice" is equal to 2, then the function returns the segmentation of the magnetic resonance image.

$$\begin{gathered} g(i,j)=1 \text{ ,for } f(i,j)\geq Threshold \\ or \\ g(i,j)=0 \text{ ,for } f(i,j)<Threshold \end{gathered} \tag{102}$$

Min_max_fun.h

The specific header file includes the following function:

- function max_value (float **input_matrix_1, float **input_matrix_2, long x_dimension, long y_dimension) which performs a pixel-by-pixel comparison between two input matrices and returns a two-dimensional matrix with values the maximum value of the absolute pixel-by-pixel comparison. The simple calculation that the function performs is given respectively by the following equation:

$$X_{i,j}=\max\left(pixel\_matrix1_{i,j}, pixel\_matrix2_{i,j}\right) \tag{103}$$

Writ_fil_2d.h

The specific header file includes the following function:

- function write_file2 (char *name_of_file, float **data_to_be_stored, long x_dimension, long y_dimension), which takes as input the name of the file that we want to store the input two-dimensional matrix, as well as its Height and

width of the matrix. If the function could not store the matrix, a warning appears in the screen.

Edge_detection.h

This specific header file includes a function that performs edge detection to a two-dimensional matrix-image. More details of the function are given below:

- function MyEdgeDetection(float **input_matrix, long x_dimension, long y_dimension, int choice) takes as inputs a two-dimensional matrix, its Height and Width, and finally the choice for the edge detection that we want to perform. The input matrix is convolved each time with a 3x3 mask, depends on the direction that we want to examine. When the variable of the function "choice" takes the value "1" the function returns the edges of the input Image examined only in the horizontal dimension. When variable "choice" is equal to "2" the function returns the edges of the input Image examined in the vertical dimension etc. In the case where the flag "choice" is equal to "5" then the function performs a pixel-by-pixel comparison between the resulted edges in the four dimensions and selects the maximum value for the final edge of the pixel. Mathematically, equation (108) shows the reported step.

$$mask1 = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix} \text{ for } "\leftrightarrow" \text{ horizontal edge detection} \qquad (104)$$

$$mask2 = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{pmatrix} \text{ for } "b" \text{ vertical edge detection} \qquad (105)$$

$$mask3 = \begin{pmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{pmatrix} \text{ for } "\backslash" \text{ diagonal edge detection} \qquad (106)$$

$$mask4 = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -1 \end{pmatrix} \text{ for } "\_" \text{ diagonal edge detection} \qquad (107)$$

$$final\_edge_{i,j} = \max\left(edge1_{i,j}, edge2_{i,j}, edge3_{i,j}, edge4_{i,j}\right) \quad (108)$$

Anadel_1_input.h

The function of this header file is the one below that returns the unknown image X from the gradient of the equation (109), setting it equal to zero.

$$J(X) = \alpha \cdot \left\| Y - H * X \right\|^2 + \beta \cdot \left\| C * X \right\|^2 \quad (109)$$

- function anadel1_input (int filter_choice, float alpha, float beta, float **Image_matrix, long x_dimension, long y_dimension) which takes as inputs the parameters alpha and beta, the input two-dimensional matrix Y as well as its Height and Width. The function calculates the derivative of the function, sets the gradient function equal to zero, and returns a two-dimensional matrix with the pixels of the unknown matrix X. We have explained analytically in the previous chapter how and why we perform such a calculation. Our main purpose here is to make our code more friendly to the user, explained the use of the functions. The two-dimensional matrix C is the Laplacian mask described in Heading 1.1.6, and the H matrix is the blurring mask which might be the simple blurring mask (90) if the input value of the function "filter_choice" is equal to "1" or the motion blurred mask (91) with "filter_choice" equal to "2". Note also that we have worked in the Fourier dimension because of the convolution between the unknown matrix X and the blurring mask. The output of the function is given by the following steps of calculations:

$$\hat{X} = \arg\min_X \left\{ J\left(X\right) \right\} \Leftrightarrow \quad (110)$$

$$\nabla_X J\left(X\right) = 0 \Leftrightarrow \quad (111)$$

$$\left(\beta \cdot C^t \cdot C + \alpha \cdot H^t \cdot H\right) X = \alpha \cdot H^t \cdot Y \Leftrightarrow \quad (112)$$

$$\hat{X} = \left(\beta \cdot C^t \cdot C + \alpha \cdot H^t \cdot H\right)^{-1} \cdot \left(\alpha \cdot H^t \cdot Y\right) \quad (113)$$

Mul_complex.h

This header file includes the following functions:

- function mult(float **real_part_output, float **imaginary_part_output, float **real_part_vec1, float **imaginary_part_vec1, float **real_part_vec2, float **imaginary_part_vec2, long x_dimension, long y_dimension) which performs a pixel-by-pixel multiplication between the two input matrices with complex numbers as data, and returns the real parts of the pixel-by-pixel multiplication in matrix "real_part_output" and the imaginary parts of the pixel-by-pixel multiplication in matrix "real_part_output". The real part as well as the imaginary part of the result are given respectively by the following equations:

$$\begin{aligned} &\text{with complex numbers : } \alpha+\beta\cdot i \text{ and } \gamma+\delta\cdot \mathrm{i} \\ &\text{real\_part} = \alpha\cdot\gamma-\beta\cdot\delta \\ &\text{imaginary\_part} = \beta\cdot\gamma+\alpha\cdot\delta \end{aligned} \tag{114}$$

Anadel_2_inputs.h

The function of this header file is the one below that returns the unknown Image X from the gradient of the equation (115), setting it equal to zero.

$$J(X)=\alpha_1\cdot\left\|\mathrm{Y}_1-H_1*X\right\|^2+\alpha_2\cdot\left\|\mathrm{Y}_2-H_2*X\right\|^2+\beta\cdot\left\|C*X\right\|^2 \tag{115}$$

- function anadelta (float alpha_1, float alpha_2, float beta, float **Image_matrix1, long x_dimension1, long y_dimension1, float **Image_matrix2, long x_dimension2, long y_dimension2) which takes as inputs the parameters alpha_1 alpha_2 and beta, the input two-dimensional matrices Y_1 and Y_2 as well as their Heights and Widths. As in the previous heading, the function calculates the derivative of the function, sets the gradient function equal to zero, and returns a two-dimensional matrix with the pixels of the unknown matrix X. The blurring masks H_1 and H_2 are respectively the masks in relations (90) and (91). The main target of this function is two take

the two input matrices Y_1 and Y_2, corresponding to the same unknown matrix X, from different systems and to restore the unknown Image X. The output of the function is given by the following steps of calculations:

$$\hat{X} = \arg\min_X \left\{ J(X) \right\} \Leftrightarrow \tag{116}$$

$$\nabla_X J(X) = 0 \Leftrightarrow \tag{117}$$

$$\left(\beta \cdot C^t \cdot C + \alpha_1 \cdot H_1^t \cdot H_1 + \alpha_2 \cdot H_2^t \cdot H_2\right) X = \left(\alpha_1 \cdot H_1^t \cdot Y_1 + \alpha_2 \cdot H_2^t \cdot Y_2\right) \Leftrightarrow \tag{118}$$

$$\hat{X} = \left(\beta \cdot C^t \cdot C + \alpha_1 \cdot H_1^t \cdot H_1 + \alpha_2 \cdot H_2^t \cdot H_2\right)^{-1} \cdot \left(\alpha_1 \cdot H_1^t \cdot Y_1 + \alpha_2 \cdot H_2^t \cdot Y\right) \tag{119}$$

spat_Frequency.h

This header file includes the following functions:

- function max_3_inputs (float a, float b, float c) which returns the maximum input between the three input variables.

- function max_5_inputs (float a, float b, float c, float d, float e) which returns the maximum input between the five input variables.

- function max_8_inputs (float a, float b, float c, float d, float e, float f, float g, float h) which returns the maximum input between the eight input variables.

- function psf (float **input_matrix, long x_dimension, long y_dimension) which takes as inputs a two-dimensional matrix and its Height and Width and returns a two-dimensional matrix with same Height and Width as the input matrix. The function performs pixel-by-pixel calculation between each pixel and its 3x3 neighborhood. Specifically, the function stores in each pixel of the resulted matrix, the sum of the magnitudes of differences from the center pixel and each other pixel in a 3x3 neighborhood. Thus our pointwise spatial frequency (psf) includes all directions and is defined for each pixel rather than for a block in which all pixels are treated the same. The general equation that

the function works with is (120). In the case where the pixel has a 3x3 neighborhood the equation is the following (121):

$$for \text{ every pixel do : } \mathrm{psf}_{i,j} = \sum \left| X_{i,j} - neighbor\_of\_pixel\left(X_{i,j}\right) \right| \qquad (120)$$

$$psf_{i,j} = \sum_{k=-1}^{1} \sum_{l=-1}^{1} \left| x_{i,j} - x_{i-k,j-l} \right| \qquad (121)$$

where $\mathrm{psf}_{i,j}$ is the pointwise spatial frequency of pixel $x_{i,j}$

- function spat_freq (int choice, float **Image1, float **Image2, float **Image3, long x_dimension, long y_dimension) which takes as input three matrices-Images to be fused as well as the their Heights and Widths. When the variable "choice"=2 the Images are fused at each output pixel location, by comparing the pointwise spatial frequency values at that location in the input Images and selecting the pixel with the highest such value. A general description is to fuse the R Images at each pixel location by comparing the pointwise spatial frequency (psf) values of the R corresponding pixels. For all R pixels x_r at the same location we put $x_{out} = x_{r^*}$ where r* is determined by the equation (122). In addition to this way of image fusion, we have performed a more complicated calculation of image fusion where all the images are participating in the fused image. When the input variable "choice" of the equation has the value "1" then the fused Image is the result from the equation (123).

$$psf\left(x_{r^*}\right) = \max_r \left\{ psf\left(x_r\right) : r = 1,\mathrm{K}\ ,R \right\} \Rightarrow$$
$$\text{fused pixel = pixel with the maximum psf} \qquad (122)$$

$$\text{for every fused pixel : } \hat{x} = \frac{\sum_{r=1}^{R} \left(psf\left(x_r\right) \cdot x_r \right)}{\sum_{r=1}^{R} psf\left(x_r\right)} \text{ , r = } 1,\mathrm{K}\ ,R \qquad (123)$$

- function min_spat_freq (int choice, float **Image1, float **Image2, float **Image3, long x_dimension, long y_dimension) which takes as input three

matrices to be fused as well as the their Heights and Widths. When the variable "choice"=2 the images are fused at each output pixel location, by comparing the pointwise spatial frequency values at that location in the two images and selecting the pixel with the minimum such value. A general description is to fuse the R images at each pixel location by comparing the pointwise spatial frequency (psf) values of the R corresponding pixels. For all R pixels x_r at the same location we put $x_{out} = x_{r^*}$ where r* is determined by the equation (124). In addition to this way of fusion, we have performed a more complicated idea of fusion where all the Images are participating in the fused Image. When the input variable "choice" of the equation has the value "1" then the fused Image is the result from the equation (125).

$$psf\left(x_{r^*}\right) = \min_r \left\{ psf\left(x_r\right) : r = 1,\mathrm{K}\ , R \right\} \Rightarrow$$
$$\text{fused pixel} = \text{pixel with the minimum psf} \tag{124}$$

$$\text{for every fused pixel} : \hat{x} = \frac{\sum_{r=1}^{R} \left(psf\left(x_r\right) \cdot x_r \right)}{\sum_{r=1}^{R} psf\left(x_r\right)}\ , \mathrm{r} = 1,\mathrm{K}\ , R \tag{125}$$

Simple_norm.h

This header file includes the following function:

- function norm (float **input_matrix, long x_dimension, long y_dimension) which takes as input a two-dimensional matrix, as well as its Height and Width and returns the norm of the matrix. The norm of the two-dimensional matrix is given by the following equation:

$$norm\left(X\right) = \sqrt{\sum_{i=1}^{i=M} \left(\sum_{j=1}^{j=N} x_{i,j}^2 \right)}\ ,\text{where M,N its Heigth and Width} \tag{126}$$

Orisma2_anadelta.h

This header file includes the following function:

- function ret_orisma (float **normaliz_matrix, float **input_matrix, long x_dimension, long y_dimension) which performs a mixture of calculations and returns a two-dimensional matrix. The general goal, in which the function has been programmed, is to return the deviation of equation (127). Note that the parameter $\overline{x}_j$ is the mean value of pixels in the neighbourhood of the pixel j and $\phi(t)=\left\{|t|^2,\sqrt{1+|t|^2}\right\}$ or any other convex function. The two-dimensional matrix q corresponds to the edge detection of the input matrix variable "normaliz_matrix", normalized first by the function in chapter 1.1.10. Finally, the parameter lambda (λ) can take any real number from 0 to 1. The function allocates memory for a temporary variable that keeps the result of the pixel-by-pixel subtraction between the two-dimensional input matrix "input_matrix" and its mean value matrix. Besides the above calculation, the function allocates memory for a temporary variable that keeps the normalization (as its performed in chapter 1.1.10) of the two-dimensional input matrix "normaliz_matrix". The return of the function is given by the equation (128). During our simulations, we have tested both convex functions depends every time on the input Image-matrix. Our remarks are shown in the previous chapters.

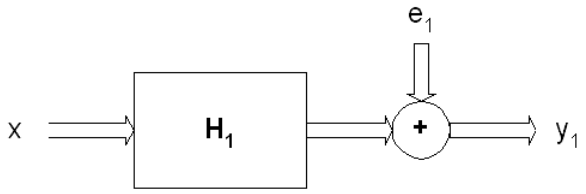
$$J(X)=\lambda\cdot\sum_j\left(1-q_j\right)\cdot\phi\left(x_j-\overline{x}_j\right) \tag{127}$$

$$\begin{cases} 2\cdot\left(1-q_{i,j}\right)\cdot\left(x_{i,j}-\overline{x}_{i,j}\right) & ,\text{if } \phi(\mathrm{x})=|x|^2 \\ \left(1-q_{i,j}\right)\cdot\dfrac{1}{\sqrt{1+\left(x_{i,j}-\overline{x}_{i,j}\right)^2}}\cdot\left(x_{i,j}-\overline{x}_{i,j}\right) & ,\text{if } \phi(\mathrm{x})=\sqrt{1+|x|^2} \end{cases} \tag{128}$$

Convolution.h

This header file includes the following function that simulates the output image from the measurement system.

- function convolution (int choice, long x_dimension, long y_dimension, float **Image, float power) which performs the convolution between the input original Image and the degradation, and finally adds to the blurred two-dimensional matrix corresponds to the Image, white Gaussian noise with mean 0, variance $\sigma^2=1$ and standard deviation $\sigma=1$. When the variable "choice" is equal to "1" then the input Image is convolved with the 3x3 blurring mask described in heading 4.6. On the other hand, when "choice" is equal to "2" the input Image is convolved with the 1x5 motion blurred ask described also in heading 4.6. Finally, variable "power" specifies the power of the output noise in dB. The given output Image $\mathbf{y_1}$, created by the convolution between the input Image **x** and the $\mathbf{h_1}$ mask is the one in the following system:



Fusion_gradient.h

This header file includes the basic functions that perform the image fusion between the magnetic resonance image and the computed tomography image.

- function grad_fusion (float **result, float alpha, float power, float **segmentation_matrix, float **input_Image, float **unknown, long x_dimension, long y_dimension) which returns the derivative of function (129). Note that this procedure is a part of the gradient descent algorithm and it doesn't work separately in the code. The function takes as input the final restored magnetic resonance or computed tomography image represented by

the two-dimensional matrix “input_imag”, the two-dimensional matrix “segmentation_matrix” from function “Segmentation” correspond to the segmentation of the input image and the unknown image where we want to optimize its values through the iterations of the gradient descent algorithm. Moreover, the function takes as inputs the variable “alpha” corresponding to the rate of the participation of the current function $J(x)$, the Height and the Width of the matrices.

$$J(x) = \alpha \cdot \sum_j S_j \left(y_j - x_j \right)^2 \tag{129}$$

- function gradient_descent (float alph1, float alph2, float lamda, float **restored_im1, float **restored_im2, float **unknown_vec, long x_dimen, long y_dimen) which returns the final result from the iterative image fusion algorithm. Function $J(x)$ is the one that updates the values of the unknown matrix **x** is shown in equation (131). Matrices “restored_im1”, “restored_im2” and “unknown_vec” are respectively the matrices $y_{1i,j}$, $y_{2i,j}$ and $x_{i,j}$ in equation (131). Note that this specific function of the header file, just updates the intensity values of matrix “unknown_vec”, send the matrix to functions “grad_fusion” and “ret_orisma” to receive the derivatives of the parts in equation (131) and finally perform the Gradient Descent method, described in Chapter 3. The optimization of the matrix “unknown_vec” is performed as it is shown in equation (132).

$$J(X) = \alpha_1 \cdot \sum_j S_{1j} \left(y_{1j} - x_j \right)^2 + \alpha_2 \cdot \sum_j S_{2j} \left(y_{2j} - x_j \right)^2 + \lambda \cdot \sum_j \left(1 - \overline{x}_j \right) \cdot \varphi \left(x_j - \overline{x}_j \right)$$

(130)

$$x_{n+1} = x_n - a \cdot \nabla f(x) \tag{131}$$

Appendix B

Suppose that we want to find the gradient of the objective function (133). Our main purpose is to set the derivative of function $J(x)$ equal to zero, in order to restore the unknown matrix-image **x**.

$$J(x) = \alpha \cdot \|y_1 - h_1 * x\|^2 + \alpha \cdot \|y_2 - h_2 * x\|^2 + \beta \cdot \|c * x\|^2 \quad (132)$$

Notice that the matrices $y_1(m,n)$, $y_2(m,n)$, $x(m,n)$ as well as the masks $h_1(m,n)$, $h_2(m,n)$ and $c(m,n)$ are two-dimensional matrices with Height equal to variable M and Width equal to variable N. In order to have the same Height and Width to the matrices and the masks we have filled the 3x3 h_1 mask, the 1x5 h_2 mask and the 3x3 Laplacian mask with zero values. Note that m, n are the pointers for the Height and the Width of the matrix.

The main problem in equation (132), is the convolution (given with the symbol $*$) between the masks and the two-dimensional matrix **x** that makes the calculations to be quite complicated. The Discrete Fourier Transform is the solution for this problem, transforming the calculations of the convolution into pixel-by-pixel calculations between the masks and the matrices. Applying the DFT in matrices $y_1(m,n)$ and $y_2(m,n)$, the matrices in the frequency domain take, respectively, the form $Y_1(k,l)$ and $Y_2(k,l)$. Note again that the variables k and l are the pointers for the Height and the Width of the matrix in the DFT domain. Every element of the matrix is a complex number.

The DFT implementation of the convolution between each mask and the matrix **x** is presented analytically in the following equations. Let's take the first part of the equation and assume that:

$$g(m,n) = h_1(m,n) * x(m,n) \quad (133)$$

Matrix $g(m,n)$ with size $M \times N$ can be converted into a vector (with size $M \cdot N \times 1$) with row ordering form in the elements of it as it is shown in equation (135).

$$\begin{pmatrix} g_{11} & \mathrm{K} & g_{1n} \\ \mathrm{M} & \mathrm{O} & \mathrm{M} \\ g_{m1} & \mathrm{L} & g_{mn} \end{pmatrix} \Rightarrow \begin{pmatrix} g_{11} \\ \mathrm{M} \\ g_{1n} \\ \mathrm{M} \\ g_{mn} \end{pmatrix} \tag{134}$$

With the same way, matrix $x(m,n)$ with size $M \times N$ can be converted into a vector (with size $M \cdot N \times 1$) with row ordering form in the elements of it as it is shown in equation (136).

$$\begin{pmatrix} x_{11} & \mathrm{K} & x_{1n} \\ \mathrm{M} & \mathrm{O} & \mathrm{M} \\ x_{m1} & \mathrm{L} & x_{mn} \end{pmatrix} \Rightarrow \begin{pmatrix} x_{11} \\ \mathrm{M} \\ x_{1n} \\ \mathrm{M} \\ x_{mn} \end{pmatrix} \tag{135}$$

In order to compute $g(m,n) = h_1(m,n) * x(m,n)$, we are based on the idea of the circulant matrix. A circulant matrix is a Toepliz matrix with the additional restriction that it must be square and each column is a rotated version of the column to the left of it **[15]**. The obvious generalization is to consider block circulant matrices with circulant blocks. Let H is the $M \cdot N \times M \cdot N$ block circulant matrix having $M \times M$ blocks as it is shown in equation (137), where each block is a $N \times N$ circlant matrix H with elements shown in equation (138).

$$H = \begin{pmatrix} H_0 & H_{M-1} & \mathrm{L} & H_1 \\ H_1 & H_0 & \mathrm{L} & H_2 \\ \mathrm{M} & \mathrm{M} & & \mathrm{M} \\ H_{M-1} & H_{M-2} & \mathrm{L} & H_0 \end{pmatrix} \tag{136}$$

$$H_j = \begin{pmatrix} h_{(j,0)} & h_{(j,M-1)} & h_{(j,M-2)} & \mathrm{L} & h_{(j,1)} \\ h_{(j,1)} & h_{(j,0)} & h_{(j,M-1)} & \mathrm{L} & h_{(j,2)} \\ \mathrm{M} & \mathrm{M} & \mathrm{M} & \mathrm{O} & \mathrm{M} \\ h_{(j,M-1)} & h_{(j,M-2)} & h_{(j,M-3)} & \mathrm{L} & h_{(j,0)} \end{pmatrix} \tag{137}$$

Applying the settings in equation (133), the convolution between the mask and the matrix can be transformed in the following form (139):

$$\underline{g} = H_{BC} \cdot \underline{x} \tag{138}$$

It is known from Linear Algebra that a circulant matrix is diagonalized by the Fourier matrix F. Since all of the blocks H_j are circulant, they can easily be factored into their eigendecomposition which leads to the following factorization of H. Thus, the Block Circulant matrix H_{BC} can be rewritten as:

$$H_{BC} = F^{-1} \cdot \Lambda_H \cdot F \tag{139}$$

Matrix Λ_H is a diagonal matrix with size $M \cdot N \times M \cdot N$ that contains the eigenvalues of H on its diagonal. These eigenvalues are the DFT coefficients of the MxN matrix h in equation (132).

$$\Lambda_H = \begin{pmatrix} H_{(0,0)} & 0 & 0 & 0 \\ 0 & H_{(0,1)} & 0 & 0 \\ \mathrm{M} & \mathrm{M} & \mathrm{O} & \mathrm{M} \\ 0 & 0 & 0 & H_{(0,M \cdot N-1)} \end{pmatrix} \tag{140}$$

Summarizing, equation (133) can be rewritten as:

$$\underline{g} = F^{-1} \cdot \Lambda_H \cdot F \cdot \underline{x} \tag{141}$$

or

$$F \cdot \underline{g} = \Lambda_H \cdot F \cdot \underline{x} \tag{142}$$

Performing the Discrete Fourier Transform in the previous calculations, we have:

$$\underline{G} = \Lambda_H \cdot \underline{X} \tag{143}$$

and finally,

$$G(k,l) = H(k,l) \cdot X(k,l) \tag{144}$$

The diagonal matrix Λ_H can be represented for now on, with the diagonal matrix H. With this simplification, the equation (132) can be rewritten as:

$$J(X) = \alpha \cdot \left\| \underline{Y}_1 - H_1 \cdot \underline{X} \right\|^2 + \alpha \cdot \left\| \underline{Y}_2 - H_2 \cdot \underline{X} \right\|^2 + \beta \cdot \left\| C \cdot \underline{X} \right\|^2 \tag{145}$$

Moreover, for our convenience all the vectors in equation (145) can be rewritten in a simplest form, as in equation (147). It is obvious that all the variables, except the masks of the equation and the parameters, are vectors. In the Frequency domain, all the calculations between the elements of the diagonal masks and the vectors are pixel-by-pixel calculations.

$$J(X) = \alpha \cdot \left\| Y_1 - H_1 \cdot X \right\|^2 + \alpha \cdot \left\| Y_2 - H_2 \cdot X \right\|^2 + \beta \cdot \left\| C \cdot X \right\|^2 \tag{146}$$

To perform the gradient calculation to the above equation, it is recommended to split the equation into two parts and work separately with them. Assume that the first and the second order of the equation have the same gradient equation. Then we have:

$$\left\| Y - H \cdot X \right\|^2 = \left(Y - H \cdot X \right)^t \cdot \left(Y - H \cdot X \right) \tag{147}$$

And

$$\left\| C \cdot X \right\|^2 = \left(C \cdot X \right)^t \cdot \left(C \cdot X \right) \tag{148}$$

Let's take the first order and perform the gradient calculation, with respect to the mathematical rules. To illustrate this approach more in details, it is known from linear algebra that:

- $$\left(A - B \right)^t = A^t - B^t \tag{149}$$
- $$\left(A \cdot B \right)^t = B^t \cdot A^t \tag{150}$$
- $$A^t \cdot B = A \cdot B^t \tag{151}$$

Note that the transpose of each matrix in the frequency domain is the complex conjugant matrix of it. Accordingly to the above mathematical rules and making step-by-step movements the equation is transformed:

$$\left\| Y - H \cdot X \right\|^2 = \left[Y^t - \left(H \cdot X \right)^t \right] \cdot \left(Y - H \cdot X \right) \Leftrightarrow \tag{152}$$

$$\left\| Y - H \cdot X \right\|^2 = Y^t \cdot Y - Y^t \cdot \left(H \cdot X \right) - \left(H \cdot X \right)^t \cdot Y + \left(H \cdot X \right)^t \cdot \left(H \cdot X \right) \Leftrightarrow \tag{153}$$

$$\left\| Y - H \cdot X \right\|^2 = Y^t \cdot Y - Y^t \cdot H \cdot X - X^t \cdot H^t \cdot Y + X^t \cdot H^t \cdot H \cdot X \tag{154}$$

With the same calculations, the equation (148) becomes:

$$\left\| C \cdot X \right\|^2 = X^t \cdot C^t \cdot C \cdot X \tag{155}$$

Afterwards, we perform the gradient estimation to the original function. Using the gradient estimation approach we can restore the unknown image **X** by the known noisy blurred Images $\mathbf{Y_1}$ and $\mathbf{Y_2}$. The equation after the gradient application is:

$$\nabla_X J(X) = \nabla_X \left(\alpha \cdot \left\| Y_1 - H_1 \cdot X \right\|^2 + \alpha \cdot \left\| Y_2 - H_2 \cdot X \right\|^2 + \beta \cdot \left\| C \cdot X \right\|^2 \right) \tag{156}$$

$$\nabla_X J(X) = \nabla_X \left(\alpha \cdot \left\| Y_1 - H_1 \cdot X \right\|^2 \right) + \nabla_X \left(\alpha \cdot \left\| Y_2 - H_2 \cdot X \right\|^2 \right) + \nabla_X \left(\beta \cdot \left\| C \cdot X \right\|^2 \right) \tag{157}$$

$$\nabla_X J(X) = \alpha \cdot \nabla_X \left(\left\| Y_1 - H_1 \cdot X \right\|^2 \right) + \alpha \cdot \nabla_X \left(\left\| Y_2 - H_2 \cdot X \right\|^2 \right) + \beta \cdot \nabla_X \left(\left\| C \cdot X \right\|^2 \right) \tag{158}$$

Accordingly to the previous simplifications that we have done, the gradient estimation of function $J(X)$ is:

$$\begin{aligned} \nabla_X J(X) = {} & \alpha \cdot \nabla_X \left(Y_1^t \cdot Y_1 - Y_1^t \cdot H_1 \cdot X - X^t \cdot H_1^t \cdot Y_1 + X^t \cdot H_1^t \cdot H_1 \cdot X \right) + \\ & \alpha \cdot \nabla_X \left(Y_2^t \cdot Y_2 - Y_2^t \cdot H_2 \cdot X - X^t \cdot H_2^t \cdot Y_2 + X^t \cdot H_2^t \cdot H_2 \cdot X \right) + \\ & \beta \cdot \nabla_X \left(X^t \cdot C^t \cdot C \cdot X \right) \end{aligned} \tag{159}$$

With respect to the previous equations, the gradient estimation of the function $J(X)$ takes the following form:

$$\begin{aligned} \nabla_X J(X) = {} & \alpha \cdot \nabla_X \left(Y_1^t \cdot Y_1 \right) - \alpha \cdot \nabla_X \left(Y_1^t \cdot H_1 \cdot X \right) - \alpha \cdot \nabla_X \left(X^t \cdot H_1^t \cdot Y_1 \right) + \\ & \alpha \cdot \nabla_X \left(X^t \cdot H_1^t \cdot H_1 \cdot X \right) + \alpha \cdot \nabla_X \left(Y_2^t \cdot Y_2 \right) - \alpha \cdot \nabla_X \left(Y_2^t \cdot H_2 \cdot X \right) - \\ & \alpha \cdot \nabla_X \left(X^t \cdot H_2^t \cdot Y_2 \right) + \alpha \cdot \nabla_X \left(X^t \cdot H_2^t \cdot H_2 \cdot X \right) + \\ & \beta \cdot \nabla_X \left(X^t \cdot C^t \cdot C \cdot X \right) \end{aligned} \tag{160}$$

Taking separately all the orders of the equation and finding their derivatives we have:

$$\nabla_X \left(Y^t \cdot Y \right) = 0 \tag{161}$$

$$\nabla_X \left(X^t \cdot H^t \cdot Y \right) = H^t \cdot Y \tag{162}$$

Assume that $J_1 = Y^t \cdot H \cdot X$. This scalar function can also be written as, $Y^t \cdot H \cdot X = \sum_k \left(\sum_l H_{kl} \cdot X_l \right) \cdot Y_k$. The gradient of the equation J_1 is now:

$$\frac{\partial}{\partial X_i} J_1(X) = \frac{\partial}{\partial X_i} \left\{ \sum_k \left(\sum_l H_{kl} \cdot X_l \right) \cdot X_k \right\} = \sum_k H_{ki} \cdot Y_k \qquad (163)$$

for every k we meet Y_i once

And finally

$$\nabla_X J_1(X) = H^t \cdot Y \qquad (164)$$

With the same method, assume that $J_2(X) = X^t \cdot A \cdot X$,with A=$H^t \cdot H$. The function is transformed as, $X^t \cdot A \cdot X = \sum_k \left(\sum_l A_{kl} \cdot X_l \right) \cdot X_k$. The gradient of the equation J_2 is now:

$$\frac{\partial}{\partial X_i} J_2(X) = \frac{\partial}{\partial X_i} \left(\sum_{k \neq i} X_k \cdot A_{ki} \cdot X_i \right) + \frac{\partial}{\partial X_i} \left(\sum_{l \neq i} X_l \cdot A_{il} \cdot X_i \right) + \frac{\partial}{\partial X_i} \left(A_{ii} \cdot X_i^2 \right) \Leftrightarrow$$

(165)

$$\frac{\partial}{\partial X_i} J_2(X) = \sum_{k \neq i} X_k \cdot A_{ki} + \sum_{l \neq i} X_l \cdot A_{il} + 2 \cdot A_{ii} \cdot X_i = \sum_k A_{ki} \cdot X_k + \sum_l A_{il} \cdot X_l \qquad (166)$$

And finally

$$\nabla_X J_2(X) = A^t \cdot X + A \cdot X \qquad (167)$$

Since A=$H^t \cdot H$ and $A^t = A$ the gradient of this order is:

$$\nabla_X J_2(X) = \nabla_X \left(X^t \cdot H^t \cdot H \cdot X \right) = 2 \cdot H^t \cdot H \cdot X \qquad (168)$$

Replacing the term $A = H^t \cdot H$ with $A = C^t \cdot C$, the gradient of the order $X^t \cdot C^t \cdot C \cdot X$ is:

$$\nabla_X \left(X^t \cdot C^t \cdot C \cdot X \right) = 2 \cdot C^t \cdot C \cdot X \qquad (169)$$

After calculating analytically the gradient of all the orders of function $J(X)$, the gradient of function $J(X)$ is:

$$\nabla_X J(X) = -2\cdot\alpha\cdot H_1^t\cdot Y_1 + 2\cdot\alpha\cdot H_1^t\cdot H_1\cdot X - 2\cdot\alpha\cdot H_2^t\cdot Y_2 + 2\cdot\alpha\cdot H_2^t\cdot H_2\cdot X + 2\cdot\beta\cdot C^t\cdot C\cdot X \quad (170)$$

Our common de-noising and de-blurring technique is to set the derivative of equation (12) equal to zero as, $\nabla_X J(X) = 0$, in order to restore the unknown Image **X**. The restored Image **X** is then:

$$\nabla_X J(X) = 0 \Leftrightarrow \quad (171)$$

$$-\alpha\cdot H_1^t\cdot Y_1 + \alpha\cdot H_1^t\cdot H_1\cdot X - \alpha\cdot H_2^t\cdot Y_2 + \alpha\cdot H_2^t\cdot H_2\cdot X + \beta\cdot C^t\cdot C\cdot X = 0 \Leftrightarrow \quad (172)$$

$$\alpha\cdot H_1^t\cdot H_1\cdot X + \alpha\cdot H_2^t\cdot H_2\cdot X + \beta\cdot C^t\cdot C\cdot X = \alpha\cdot H_1^t\cdot Y_1 + \alpha\cdot H_2^t\cdot Y_2 \Leftrightarrow \quad (173)$$

$$\left(\alpha\cdot H_1^t\cdot H_1 + \alpha\cdot H_2^t\cdot H_2 + \beta\cdot C^t\cdot C\right)\cdot X = \alpha\cdot H_1^t\cdot Y_1 + \alpha\cdot H_2^t\cdot Y_2 \Leftrightarrow \quad (174)$$

$$\hat{X} = \left(\alpha\cdot H_1^t\cdot H_1 + \alpha\cdot H_2^t\cdot H_2 + \beta\cdot C^t\cdot C\right)^{-1}\left(\alpha\cdot H_1^t\cdot Y_1 + \alpha\cdot H_2^t\cdot Y_2\right) \quad (175)$$

or

$$\hat{X}(k,l) = \frac{a\cdot H_1^*(k,l)\cdot Y_1(k,l) + a\cdot H_2^*(k,l)\cdot Y_2(k,l)}{a\cdot\left|H_1(k,l)\right|^2 + a\cdot\left|H_2(k,l)\right|^2 + \beta\cdot\left|C(k,l)\right|^2} \quad (176)$$

where k, l are the pointers for the Height and Width of each matrix in the frequency domain. At last, in order to get back the restored Image in the time domain, we apply the inverse Discrete Fourier Transform method in matrix $\hat{X}$.

REFERENCES

[1] H. C. Andrews and B. R. Hunt, *Digital Image Restoration*, Englewood Cliffs, NJ: Prentice-Hall, 1977.

[2] Data Fusion: from Primary Metrology to Process Measurement, Gavin Kelly National Physical Laboratory

[3] Information Fusion: Application to data and model fusion for ultrasound Image Segmentation, B. Solaiman, R. Debon, F.Pipelier and C.Roux
Dépt. Image et Traitement de l'Information
Laboratoire de Traitement de l'Information Médicale (L.A.T.I.M)

[4] L.F. Pau, "Fusion of multisensor data in pattern recognition", Pattern Recognition Theory and Application, pp. 189-201, 1982.

[5] T. Gravey, J. Lowrence and M.A. Fischler, "An inference technique for integrating
knowledge from disparate sources", Inter. Joint Conf. on Artificial Intelligence, 1981.

[6] J.K. Aggarwal and N. Nandharkumar, "Multisensor fusion for automatic scene interpretation", Analyse and Interpretation of Range Images, pp.339-361, Spr.Verlag.

[7] J.J Clark and L. Yuille, "Data fusion for sensory information processing systems", Kluwer Academic Publishers, 1990.

[8] Probabilistic model based methods for data fusion, Ali Mohammad-Djafari, Laboratoire des Signaux et Systems (CNRS-SUPELEC-UPS)

[9] Assessment of Data and Knowledge Fusion Strategies for Prognostics and Health Management, Michael J. Roemer, Gregory J. Kacprzynski, Rolf F. Orsagh,
Impact Technologies, LLC

[10] Image Fusion with Spatial Frequency, Supported by Army Research Office Contract. Lily R. Liang, Carl G. Looney, Computer Science Department/171 University of Nevada, Reno

[11] An Introduction to the Conjugate Gradient Method without the Agonizing Pain Edition $1\frac{1}{4}$ Jonathan Richard Shewchuk August 4, 1994

School of Computer Science Carnegie Mellon University Pittsburgh, PA 15213

[12] Digital Image Processing. Rafael Gonzalez, Richard Woods

[13] Digital Image Processing. Kenneth Castleman

[14] The Laplacian Pyramid as a Compact Image Code. PETER J. BURT, MEMBER, IEEE, AND EDWARD H. ADELSON

[15] G.H Golub and C.F. van Loan. Matrix Computations. The John Hopkins University Press, third edition 1996

[16] N. Otsu, A threshold selection method from grey level histograms, IEEE Transactions on Systems, Man and Cybernetics, vol. 8, pp. 62-66, 1978.

[17] M. Fiani, J. Idier, and S. Gautier, "Algorithmes ART semi-quadratiques pour la re-construction a partir de radiographies" in Actes du 18 colleque GRETSI, (Toulouse), sep. 2001.

[18] G. Gindi, M. Lee, A. Rangarajan, and I.G. Zubal, "Bayesian reconstruction of functional images using anatomical informations as priors" IEEE Trans. Medical Imaging, vol. MI-12, no. 4, pp. 670-680, 1993.

[19] S. Gautier, J. Idier, A. Mohammad-Djafari, and B. Lavayssiere, "Traitement d' echogrammes ultrasonores par deconvolution aveugle" in Actes du 16 colloque GRETSI, (Grenoble), pp. 1431-1434, sep. 1997.

[20] S. Gautier , J. Idier, A. Mohammad-Djafari, and B. lavayssiere, "X-Ray and ultrasound data fusion" in Proc. IEEE ICIP, (Chicago, IL), pp. 366-369, oct. 1998

[21] Image Processing and Edge Detection, Computer Vision © Department of Computing, Imperial College. GZ Yang and DF Gillies.