

**COMPUTED TOMOGRAPHY (CT)
IMAGE RECONSTRUCTION
USING
MATLAB PROGRAMMING**

**Ferdinan Manuel Siahaan
6305230039**



**UNIVERSITAS INDONESIA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
PROGRAM MAGISTER FISIKA
KEKHUSUSAN FISIKA MEDIS
DEPOK
2008**

**COMPUTED TOMOGRAPHY (CT)
IMAGE RECONSTRUCTION
USING
MATLAB PROGRAMMING**

A Thesis submitted in partial fulfillment of
the requirements for the degree of Master of Science

By :

Ferdinan Manuel Siahaan

6305230039



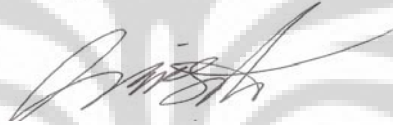
DEPOK

2008

APPROVAL PAGE

TITLE : COMPUTED TOMOGRAPHY (CT) IMAGE RECONSTRUCTION
USING MATLAB PROGRAMMING
NAME : FERDINAN MANUEL SIAHAAN
NPM : 6305230039

APPROVED




Dr. Warsito
COUNSELLOR

COMMITTEE :



Prof. Dr. Djarwani Soeharso
CHAIR



Dr. Mussadiq Musbach
MEMBER



Dr. Freddy Haryanto
MEMBER

Program Studi Magister Fisika

Program Pasca Sarjana FMIPA Universitas Indonesia



Dr. Dedi Sutanto
DIRECTOR

Graduating Date: June 17th, 2008

ACKNOWLEDGEMENTS

Praise the Mighty God because only by His permission this research can be finished and tested. And since conducting and finishing the research is a longterm trial and error work, it is a pride for me to acknowledge great appreciations to following persons:

1. My wife, Imelda, and my little girl, Rahel, whose love and understanding had given major determination and inspiration to me to be able to finish the research and write the thesis.
2. The thesis counsellor, Dr Warsito, who had encouraged me to take the research title and whose recommendations and advices had been leading me to finish the research and write the thesis.
3. Prof. Dr. Djarwani, chair of the thesis comitee, whose experience and leadership had enable the thesis to be tested on the proper dateline.
4. Dr. Mushadiq and Dr. Freddy, members of the thesis comitee, who had spared their precious time to test the thesis and give precious corrections.
5. Dr. Dedi Susanto, Director of Graduate Program of Mathematic and Science Department of University of Indonesia, who had given important permission for me to fullfill the administrative requirements on the proper time.
6. All lecturers in Graduate Program of Medical Physics Specialization who had given important knowledge for me to understand the research basic theory. It is also a pride to especially acknowledge the contribution of (Alm) Dr. Rahmad Widodo Adi, former lecturer in Graduate Program of Medical Physics.
7. Prof. Dr. Abubakar Lubis, my former Director, who before his retirement had major permission to let me taking the graduate program. Mr. Reno Alamsyah, Mr. Dedik Eko Sumargo and Mr. Heddy Krishyana, my Director, Sub Director and former Director respectively, in Badan Pengawas Tenaga Nuklir (BAPETEN), who had let me finishing the thesis on the time limit. Also the appreciations are sent to my friends and colleagues in the office who had given important help.

8. All my friends in Graduate Program of Medical Physics of Year 2005: Wigati, Ardi, Wayan, Jumali, Arsi and Tuti, who with me had spent unforgettable moments conducting the lecture time together. May we will stay in connection and be able to communicate again.
9. Mr. Suparman and Mr. Sarmidi, the administrative officers, who had given important administrative help. Without their contribution, the thesis would not be finished and tested.
10. All persons and sides, who can not be mentioned here one by one, who had given precious contributions for the finishing of the thesis.

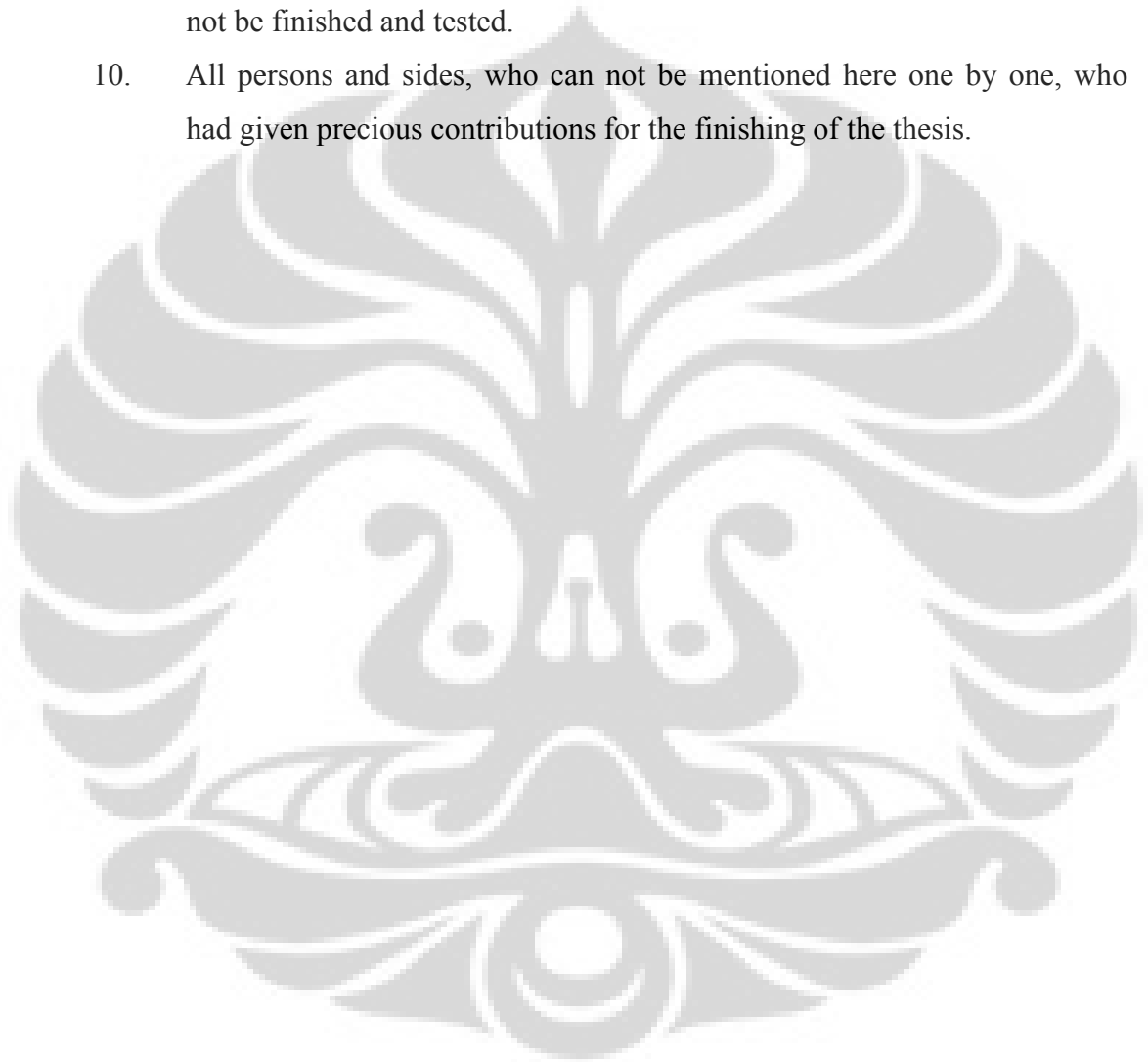


TABLE OF CONTENTS

	page
APPROVAL PAGE	i
ACKNOWLEDGEMENTS	ii
TABLE OF CONTENTS	iv
LIST OF TABLES	vii
LIST OF FIGURES	viii
LIST OF APPENDICES	x
ABSTRACT	xi
CHAPTER	
I. INTRODUCTION	1
II. THEORY	3
II.1. Physical Interpretation of CT Projection	3
II.2. Radon Transform	4
II.2.1. Parallel Beam Projection	6
II.2.2. Fan Beam Projection	7
II.3. Reconstruction Method.....	8
II.3.1. Filtered Back Projection (FBP) Technique	8
II.3.1.1. Projection Data: Unfiltered Sinogram	8
II.3.1.2. Filtered Sinogram	9
II.3.2. Algebraic Reconstruction Technique (ART)	11
II.3.2.1. Projection Matrix	11
II.3.2.2. Backprojection: Initial Guess	13
II.3.2.3. Blurring and Iterative Filtering Technique	14
II.4. Image Quality: Mean Square Error (MSE)	16
III. METHODOLOGY	17
IV. ALGEBRAIC RECONSTRUCTION TECHNIQUE (ART) ALGORITHM	22
IV.1. Third Generation CT-Scanner Projection Plan	22
IV.2. Projection Data Matrix (P)	23
IV.3. Object Matrix (A)	24

IV.3.1. Pixel Numbering	24
IV.4. Projection Function	26
IV.4.1. Ray Linear Equation	26
IV.4.2. Intersection Points	27
IV.4.3. Intersection Shape	28
IV.4.4. Intersection Area	31
IV.5. Initial Quadrant Projections	32
IV.6. Radon Transform: Imaginary Object Matrix Back Rotation	34
IV.6.1. A Quadrant – Object Matrix Back Rotation	35
IV.6.2. A Half Circle – Object Matrix Back Rotation	36
IV.7. Projection Matrix	37
IV.7.1. Initial Projection from A Projection Matrix: Practical Use	38
IV.8. Backprojection	39
IV.8.1. Iterative Filtering Process	40
IV.9. Otomatization : Matlab Programming	41
V. RESULT	43
V.1. Projection Matrix	43
V.2. Parameters Optimization	44
V.2.1. Beam Shape	44
V.2.2. Angular Width	45
V.2.2.1. Beam Angular Width	45
V.2.2.2. Ray Angular Width	46
V.2.3. Reconstruction Time	47
V.2.4. Number of Iterations (j) and Filtering Process	49
V.3. Experiment	50
V.3.1. Qualitative	50
V.3.1.1. Object Matrix of N=32x32 and M=64x64 projections	50

• Changing Parameter: Number of Iterations (j) for lambda (λ) = 2	50
• Changing Parameter: Lambda (λ) for j = 10.000 iterations	51
V.3.1.2. Object Matrix of N=64x64 and M=128x64 projections	51
• Changing Parameter: Number of Iterations (j) for lambda (λ) = 2	51
• Changing Parameter: Lambda (λ) for j = 10.000 iterations	52
V.3.2. Quantitative: Mean Square Error (MSE)	54
V.3.3. Comparison of ART and FBP Technique	58
V.3.4. Comparison of ART and Another Iterative Technique ...	59
VI. SUMMARY	62
REFERENCES	64
APPENDICES	66

LIST OF TABLES

	page
Table 5.1. Backprojection speed (iterations/seconds)	59
Table 5.2. Mean Square Error (MSE) / pixel	66



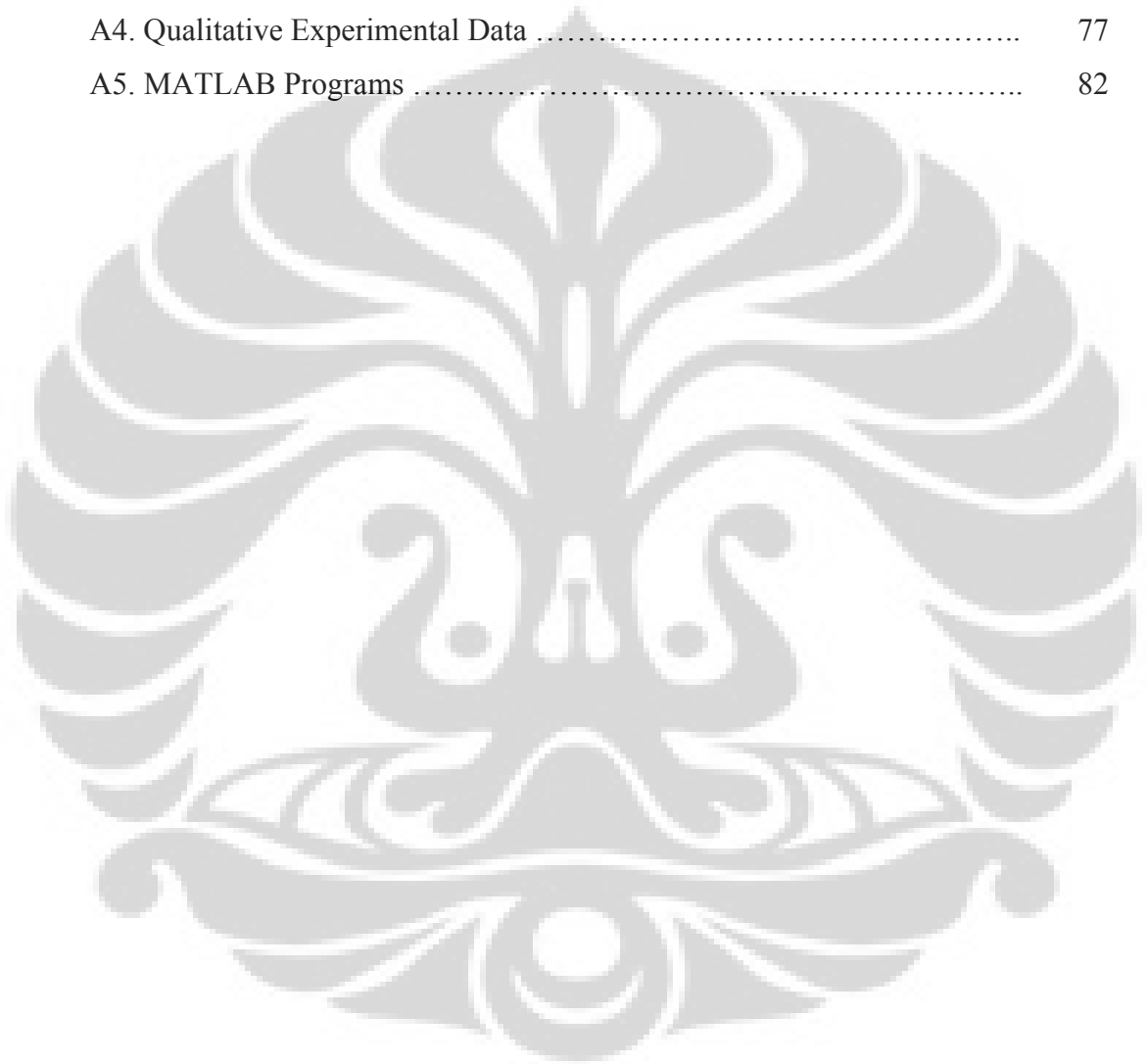
LIST OF FIGURES

	page
1.1. A historic EMI-Scanner	1
2.1. Attenuation process from initial intensity (I_0) to final intensity (I_1) of photons passing a slice of four voxels.....	4
2.2. Radon Transform.....	5
2.3.a. An imaginary circular matrix projected by $P\phi(k)$	6
b. An original rectangular matrix.....	6
2.4. A quadrant projections.....	8
2.5.a. A Sinogram.....	10
b. Ram-Lak Filter.....	10
c. A Filtered Sinogram of (a).....	10
2.6. Three views of parallel beam projection of an EMI scanner	11
2.7. A fan beam consisting of rays.....	12
2.8.a. Projection of four fan beam onto a rod of D diameter.....	14
b. Backprojection of (a)	14
3.1. Projection design of a third generation CT-Scanner.....	17
3.2. Angular width of a beam ($m_\beta, \Delta\beta$) based on matrix width.....	18
3.3. Angular width of a ray ($\Delta\beta$) based on a pixel width.....	19
3.4. A Shepp-Logan phantom of $N = 32 \times 32$	20
3.5. A Shepp-Logan phantom of $N = 64 \times 64$	21
4.1. A Third Generation CT-Scanner.....	22
4.2. A matrix of N pixels, k is the pixel number, $k=1,2,3,\dots,N$	24
4.3. A ray projecting a pixel	26
4.4. A beam direction of $\alpha = 90^\circ$, with three categories of ray direction	28
4.5.a. Direction category 1: $0 < \Phi < (\pi/2 - \Delta\beta)$	29
4.5.b. Direction category 2: $(\pi/2 - \Delta\beta) \leq \Phi \leq (\pi/2)$	29
4.5.c. Direction category 3: $(\pi/2) < \Phi < (\pi - \Delta\beta)$	30
4.6. A representative beam direction of $\alpha = 90^\circ$	33
4.7. Four quadrants positions, Quadrant 1 is the initial quadrant	33
4.8. Rotating a quadrant projections when the object matrix is fixed (left) is the same as rotating object matrix to opposite direction by a factor of 90° when the quadrant projections is fixed (right)	34
5.1.a. Plot of a projection matrix $W_{(M \times N)}$, $M=16 \times 16$, $N=16 \times 16$	43
5.1.b. Plot of a projection matrix $W_{(M \times N)}$, $M=32 \times 32$, $N=32 \times 32$	43
5.2.a. A parallel ray projection	44
5.2.b. A fan ray projection	44
5.3. Three possible reference parts width of a matrix	45
5.4. Possible reference pixels with positions along X-axis is $a_0 - fd$, where f is a changing value	46
5.5. A chart of backprojection speed vs number of pixels (N). The blue line is for M (projections) = N while the red line is for $M=2N$	48
5.6. $j=10.000$ iterations, $\lambda = 2$	48
5.7.a. $\lambda = 5$, $j=10.000$ iter	51
5.7.b. $\lambda = 15$, $j=10.000$ iter.....	51
5.8.a. $\lambda = 2$, $j=1.000$ iter	52

5.8.b. $\lambda = 40, j = 100$ iterations	52
5.9.a. $\lambda = 2, j = 1.000$ iter	53
5.9.b. $\lambda = 40, j = 100$ iterations	53
5.10. $\lambda = 50, j = 1.0000$ iterations	53
5.11.a. A Graphic of MSE vs Iterations for $\lambda = 2$. Object matrix of $N = 32 \times 32$, M = 64×64 projections	56
5.11.b. A Graphic of MSE vs λ for $j = 10.000$ Iterations, Object matrix of $N = 32 \times 32$, M = 64×64 projections	56
5.11.c. A Graphic of MSE vs Iterations, for $\lambda = 2$. Object matrix of $N = 64 \times 64$, M = 128×64 projections	57
5.11.d. A Graphic of MSE vs λ for $j = 100$ Iterations. Object matrix of $N = 64 \times 64$, M = 128×64 projections	57
5.12.a. Unfiltered sinogram	58
5.12.b. Ram-Lak filter	58
5.12.c. Filtered sinogram of (5.11a.)	58
5.12.d. Resulting Image, M = 64×32 , MSE = 0.0115	58
5.13. Resulting image of the ART developed in the research, $\lambda = 15, j = 10.000$ iterations, MSE = $2.09E-08$	58
5.14.a. Resulting Image, M = 64×32 , MSE = 0.0115	59
5.14.b. Resulting Image, M = 128×32 , MSE = 0.011	59
5.15.a. (top) M = 32×32 , N = 32×32 , fan beam, parallel source-detector geometry, $j = 1.000$ iterations (left) and $j = 10.000$ iter. (right).....	60
5.15.b. (bottom) M = 128×128 , N = 32×32 , fan beam, clamshell source- detector geometry, $j = 1.000$ iterations(left) and $j = 40.000$ iterations (right)	60
5.16. Resulting image of the research, $\lambda = 15, j = 10.000$ iterations.....	61

LIST OF APPENDICES

	page
A1. Table of Intersection Points Position	66
A2. Table of Intersection Shapes and Area	67
A3. Resulting Images of Parameters Optimization	70
A4. Qualitative Experimental Data	77
A5. MATLAB Programs	82



COMPUTED TOMOGRAPHY (CT) IMAGE RECONSTRUCTION USING MATLAB PROGRAMMING

By Ferdinan Manuel Siahaan^{*)}

A Thesis submitted in partial fulfillment of the requirements for the degree of
Master of Science at

*Program Studi Magister Fisika
Program Pasca Sarjana FMIPA
Universitas Indonesia, 2008*

(Thesis, 92 + xi Pages, 57 Figures, 2 Tables, Year 2008)

ABSTRACT

X-ray computed tomography (CT) has been playing an important role in current medical practice for diagnostic procedure. Beside its delicate technology, the 'hidden' software of CT image reconstruction has contributed almost half of total cost of a CT-scanner unit. Since Algebraic Reconstruction Technique (ART) is a basic to understand an iterative method of CT image reconstruction algorithm, and since it is difficult to find a clear description of fan beam ART algorithm in university literatures, it is important to develop an own algorithm and to begin a basic systematic research of this iterative method. After a long term of trial and error work, the research had succeeded in developing an ART algorithm for third generation CT image reconstruction. By comparing the result of the research with more popular technique like Filtered Back Projection (FBP), the algorithm has been proved applicable to reconstruct a low dimension object matrix (32x32 and 64x64). By the resulted computer program, then basically a simple and low cost third generation CT-scanner can be designed for medical physics or biomedical imaging research. Finding a way of shortening the massive number of iterations process then, will be able to open the possibility of using the software for higher object matrix dimensions.

Keywords: Iterative method, Algebraic Reconstructive Technique (ART) reconstruction algorithm, Filtered Back Projection (FBP) technique, matrix dimension, Matlab program.

^{*)} f.manuel@bapeten.go.id

CHAPTER I

INTRODUCTION

Background

In the early 1930s the Italian radiologist Alessandro Vallebona proposed a method to represent a single slice of the body on the radiographic film. This method was known as tomography (tomos means slice). The first commercially CT scanner was invented by Sir Godfrey Newbold Hounsfield in Hayes, United Kingdom at EMI (Electric and Musical Industries) Central Research Laboratories using X-rays. Hounsfield introduced his idea in 1967, and it was announced in 1972. Allan McLeod Cormack of Tufts University, Massachusetts, USA independently invented a similar process, and both Hounsfield and Cormack shared the 1979 Nobel Prize in Medicine.

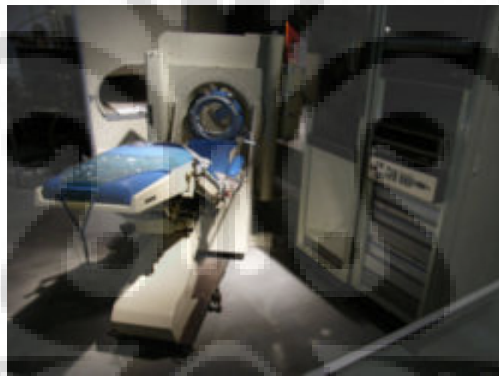


Figure 1.1. A historic EMI-Scanner ^[22]

Questions of The Research

The questions of the research are:

1. How to develop a complete algorithm of image reconstruction from a third generation CT-Scanner projection based on simultaneous linear equations solving (simple backprojection technique)? How to create an efficient computer program based on the algorithm?
2. What are the optimal value of input parameters of the program to achieve an efficient reconstruction process and best resulting image?

3. How is the quality of the resulting image reconstructed by Algebraic Reconstructive Technique (ART) compared to those reconstructed by more popular Filtered Back Projection (FBP) technique.

Purpose of The Research

To answer the questions of the research, the purposes of the research are:

1. To create an efficient computer program for CT image reconstruction that based on Algebraic Reconstructive Technique (ART) using Matlab programming.
2. To optimize main parameters value for the program. The parameters are:
 - i. Beam shape (parallel and fan)
 - ii. Angular width (beam and ray)
 - iii. Reconstruction time
 - iv. Number of iterations (filtering process)
3. To conduct an experiment to implement the optimized parameters of CT image reconstruction using Algebraic Reconstructive Technique (ART) and then comparing the result with Filtered Back Projection (FBP) technique.

Originality of the Research

The originalities of the research are:

1. Development of ART algorithm based on a quadrant rotation technique for fan beam projection.
2. Matlab programming of CT image reconstruction based on ART algorithm.
3. Conducting simulations to provide optimized parameters for CT fan beam projections and experiments to know the performance of the computer program.

CHAPTER II THEORY

II.1. Physical Interpretation of CT Projection

A photon of intensity I_0 that emerges when a narrow beam of monoenergetic photons with energy E passes through a homogeneous absorber of thickness x can be expressed as [1]:

$$I = I_0 \exp [-\mu(\rho, Z, E)x]$$

Where μ , ρ and Z are the linear attenuation coefficient, density of the absorber and atomic number, respectively. In the energy region there most commercial X-ray CT system are existing for medical tomography ($\cong 70$ keV), two types of interaction are dominant. They are photoelectric absorption and Compton scattering.

In photoelectric interaction the X-ray photon is completely absorbed by transferring all of its energy to an element. In Compton scattering, scattered X-rays undergo both a directional and energy change. By directing a monochromatic X-ray beam in the y direction, for instance, the output X-ray intensity $I(x)$ can be written as:

$$I(x) = I_0(x) \exp \left[- \int \mu(x, y) dy \right] \quad (2-1)$$

Where I_0 and $\mu(x, y)$ are the incident X-ray intensity and X-ray attenuation coefficient, respectively. By taking the logarithm from above equation, projection data $p(x)$ can be obtained:

$$p(x) = - \ln \left[\frac{I(x)}{I_0(x)} \right] \quad (2-2)$$

$$= \int \mu(x, y) dy \quad (2-3)$$

Where $p(x)$ is equivalent to a simple integration or summation of the total attenuation coefficient along the X-ray path (i.e., y direction). In digital form, above equation becomes:

$$p(x) = \sum_{i=1}^{i=N} \mu_i(x, y), \quad N = 1, 2, 3, \dots, N \quad (2-4)$$

In X-ray CT the contrast is associated with the different attenuation coefficient of the material involved. Since each set of projection data represents the integral value of the attenuation coefficients along the path, the projection data taken at different views are the basic for tomographic image reconstruction. An example of a projection of four voxels in **x direction** can be seen below:

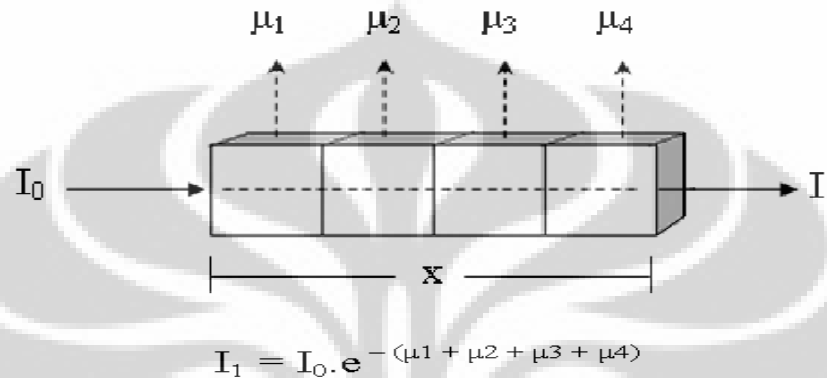


Figure 2.1. Attenuation process from initial intensity (I_0) to final intensity (I_1) of photons passing a slice of four voxels of total length x and attenuation constants μ_1 , μ_2 , μ_3 and μ_4 respectively. Since the voxels thickness are the same, each μ will be the value of a pixel.

II.2. Radon Transform

The mathematical principles of CT were first developed by Radon in 1917. Radon transform has proved that an image of an unknown object could be produced if one had an infinite number of projections through the object, we can understand the basic idea behind tomographic imaging with an example taken from radiography.

Projection function in Radon transform is interpreted as a set of line integrals of straight lines (y') that projecting the object area by angle ϕ . Each line integral, in practice, represents a physical property of a strip with a finite width which is determined mainly by the detector width. At each view a set of line-integral data is obtained. Complete projection data sets can be obtained by repeated assessment of the data round $180'$ or 360° with a specified angular indices.

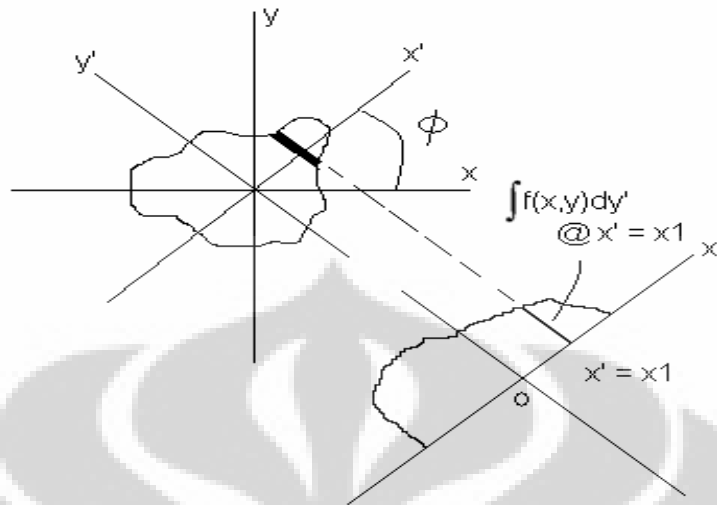


Fig.2.2. Radon Transform

The Radon transform equation is:

$$P\phi(x') \equiv R [f(x,y)]$$

$$= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x,y) \cdot \delta(x \cdot \cos \phi + y \cdot \sin \phi - x') \cdot dx \cdot dy \quad (2-5)$$

$$= \int_{-\infty}^{\infty} f(x' \cdot \cos \phi - y' \cdot \sin \phi, x' \cdot \sin \phi + y' \cdot \cos \phi) \cdot dy' \quad (2-6)$$

Where coordinate (x',y') resulted from rotating coordinate (x,y) by ϕ is:

$$\begin{bmatrix} X' \\ Y' \end{bmatrix} = \begin{bmatrix} \cos \phi & \sin \phi \\ -\sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix} \quad (2-7)$$

So, the initial (x,y) coordinate related to coordinate (x',y') will be:

$$\begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} \cos \phi & -\sin \phi \\ \sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} X' \\ Y' \end{bmatrix} \quad (2-8)$$

Since an object is a set of points in (x,y) coordinate, Radon transform will map all points projected by y' into a new coordinate (ϕ,x') . The result is a sinusoidal graphic which is a characteristic of a projection set of an object.

II.2.1. Parallel Beam Projection

If the projection line is replaced by a parallel beam, and the object is a matrix of pixels, then a pixel can be assumed as a big point with diameter similar as the beam width. In the sinusoidal graphic the parallel beam is now represented by a thin sinusoidal line, and the projection (y') data value is represented by the scaled contrast of the line. The resulting sinusoidal graphic will be a set of multicontrast sinusoidal thin lines. This graphic is a projection data record of a CT projection onto an object, known as a Sinogram. Then, by backprojecting the Sinogram, a resulting image of the object can be achieved.

Since in Radon transform, each pixel identity is determined only by its distance to central point (k), then for easier interpretation, a rectangular object matrix can also be imagined as a circular matrix. A specific distance of pixels to the central point will be represented by a circle line. So the resulting imaginary matrix will be a set of circles with radius k , projected by $P(\varphi, k)$.

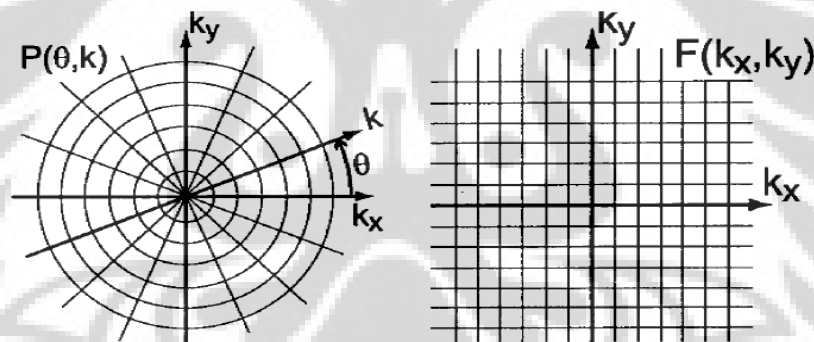


Fig.2.3. An imaginary circular matrix projected by $P\varphi(k)$ (left) and the original rectangular matrix (right).

Because in practical modern CT, thousands of both projections and matrix pixels will need massive computation of matrix operations, it will be unefficient to solve the Radon transform in spatial domain. Therefore the spatial domain coordinate (x, y) is usually transformed first into a frequencial domain coordinate (k_x, k_y). The symbol k is popular as a unit in frequencial domain, where:

$$k = 1/\Delta x \text{ (mm}^{-1}\text{)} \quad (2-9)$$

where: Δx = a spatial distance unit (mm)

II.2.2. Fan Beam Projection

For fan beam projection used in the research, the Radon transform is not able to be applied directly because of two reasons:

- i. A fan beam can not be directly assumed as a line projection.
- ii. The reconstruction technique used in this research is Algebraic Reconstructive Technique (ART) which based on matrix operations where the object matrix is kept as a rectangular geometry in a Cartesian coordinate system.

To answer the first problem, with a set of mathematic details, Cho [1] had suggested a definition of a specified positioned fan beam of direction φ as a parallel beam. According to Cho, all projection rays of the fan beam can be defined as a function of the fan beam axis (a radius line between source and central point $(0_{(0,0)})$). Then all the function of rays can be integrated as a single beam function. By the function, the beam of direction φ can be defined as a parallel beam which then by using Radon transform can be generated to all beam directions. Since they are beyond the scope of the research, the details are not mentioned here. However the idea of defining fan beam as a basic projection before generating it using Radon transform is very useful to simplify the algorithm developed in this research.

Related to the second problem, it is understood that Radon transform can generate projections only if the object matrix is transformed first into a circular shape. By this transformation, all intersection shapes formed between projection beam and the object will be the same and only one integral formula is needed for all beam directions. But for a rectangular object matrix, the intersection shape between projection beam and the object will much depend on the projection direction (φ). There is no general integral formula that can stand for all φ .

To solve the two problems of fan beam projection, the research suggested to use the symmetrical properties of a square matrix. If a square matrix was rotated by a factor of 90^0 , the shape of matrix would have repetition every 90^0 , where ever the rotation starting point was. So by defining a quadrant (90^0) part of a full circle projection (360^0) that projecting a rectangular matrix, the matrix then can be assumed as a circular object. This quadrant part is called here as a quadrant

projections After defining all possible intersection shape between all fan beam located in the initial quadrant projections, a general integral formula can be generated for other three quadrants. An example of the initial quadrant projections can be seen in figure below:

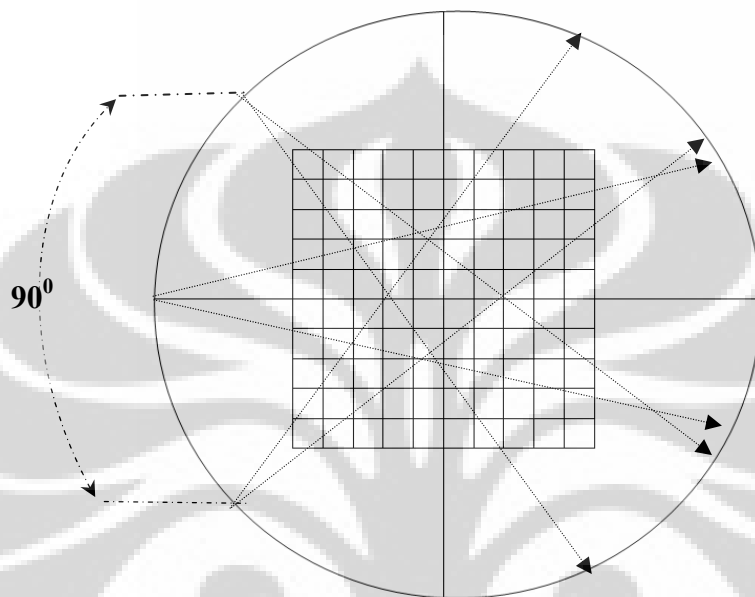


Figure 2.4. A quadrant projections

The quadrant projections will be the main technique of the reconstruction algorithm developed in this research to generate projection function for all directions.

II.3. Reconstruction Method

II.3.1. Filtered Back Projection (FBP) Technique

The Filtered Back Projection (FBP) or convolution backprojection algorithm is the most popular reconstruction method so far and is used in both transmission and emission CT. Its simple procedure of generating the initial projection for all projection angles, makes the reconstruction time relatively very short.

II.3.1.1. Projection Data: Unfiltered Sinogram

As described before, by using the Radon transform to generate an initial projection of $y' = f(x')$ for all projection angle, ϕ , a CT projection data function,

$P(s,\phi)$, is resumed as a sinusoidal graphic called a sinogram, which then can be backprojected to get the reconstructed image.

Since in FBP technique the initial domain of a sinogram is frequencial, direct backprojection of the sinogram will inverse the domain to be spatial. The domain transformation will cause a slight mathematical distortion which will be accumulated to become a blurring in the reconstructed image, called as a $1/r$ blurring. In FBP technique this problem is solved by adding first a filter function to the sinogram before it is backprojected. General equation of the estimated object function, $f(x,y)$, in FBP can be written down as:

$$f(x,y) = 1/\pi \int_0^\pi d\phi \int_{-\infty}^\infty dx' p_0(x') h(x \cdot \cos \phi + y \cdot \sin \phi - x'). \quad (2.10)$$

where $h(x)$ is the filter function.

II.3.1.2. Filtered Sinogram

In frequencial domain the $1/r$ blurring is interpreted as a function of frequencial distance unit ($k = 0,1,2,3,..K$) of the object points position to the center. Since the blurring function is divided in three cathegories of k value (zero, even or odd), Ramachandran-Lakshmirayan [2] suggested an alternative filter function which will add different negative value for each of the three cathegories of k value. This filter function is called as a Ram-Lak filter. By applying the Nyquist sampling criteria with uniformly spaced $\Delta x=1/(2B)$, the sampled (discrete) version of Ram-Lak filter is:

$$h_{RL}(0) = B^2 = 1/(4 \Delta x^2) \quad (\text{if } k=0) \quad (2.11a)$$

$$h_{RL}(k) = 0 \quad (\text{if } k \text{ even}) \quad (2.11b)$$

$$h_{RL}(k) = -4B^2/(\pi^2 k^2) = -1/(\pi^2 k^2 \Delta x^2) \quad (\text{if } k \text{ odd}) \quad (2.11c)$$

By convoluting this filter function with the projection function, then it can be safely generated for all projection angle, ϕ , to get a filtered projection data function, $P_f(s,\phi)$, which is represented in a filtered sinogram. The sinogram then can be safely backprojected to a filtered reconstructed image. Although in the FBP technique, the Ram-Lak filter is simple and very effective to decrease general $1/r$ blurring, it can not completely solve the blurring caused by small density difference of two neighbouring object mediums (eq: tumour and tissue). Other

filter functions then can be used to solve this problem, like Shepp-Logan or Hamming filter function. So in medical application of CT imaging, the choice of filter function to be used in the FBP technique also considering the diagnostic priority of the CT scanning. A series of figures of a sinogram, its Ram-Lak filter and filtered sinogram can be seen below.

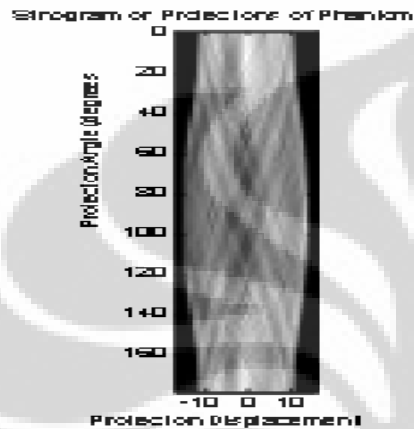


Figure 2.5a. A Sinogram

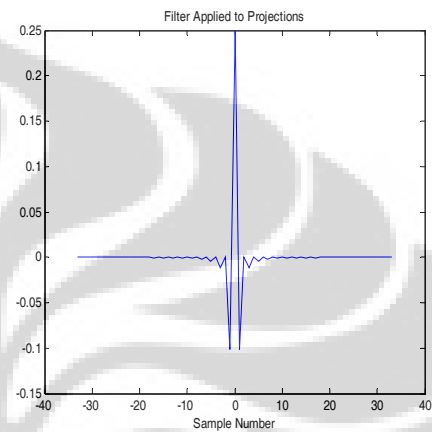


Figure 2.5b. Ram-Lak Filter

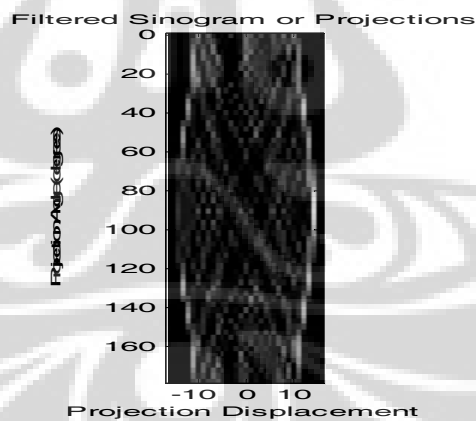


Figure 2.5c. Filtered Sinogram of (a)

Using this filtered sinogram, a backprojection process can directly be conducted without any added filter. This direct backprojection process is a benefit of the FBP since the filtering process has been conducted along with projection function.

II.3.2. Algebraic Reconstruction Technique (ART)

The ART is one of some iterative techniques used in CT image reconstruction. Iterative techniques are often used in image reconstruction as an alternative reconstruction method. The ART was first of its kind based on the iterative procedures and was first used in the EMI brain scanner developed by Hounsfield. Currently the ART algorithm is used only in some specific application, such as the case of limited view angle reconstruction. Its relative clear resulting image can compensate the drawback of its inefficiency computing operations. Therefore it is suitable enough for research and educating experiments which involve limited object matrix dimensions.

II.3.2.1. Projection Matrix

In an EMI scanner a parallel beam which determine a view (direction) of the projection will be measured by a detector array consisting of a number of detectors, which will determine the number of samplings of the beam. Each detector of the array will have its own projection data according to a part of the beam (photon intensity) that succeeded in passing the object medium of its path and strike the detector window. Since the X-ray source can be assumed as a point, each part of the beam that pass a small fan shaped track can be assumed as a small beam itself. A small beam here is simply called as a **ray**. So a projection beam (view) consists of many projection rays (samplings). A figure of three views of parallel beam projection of an EMI scanner can be seen below:

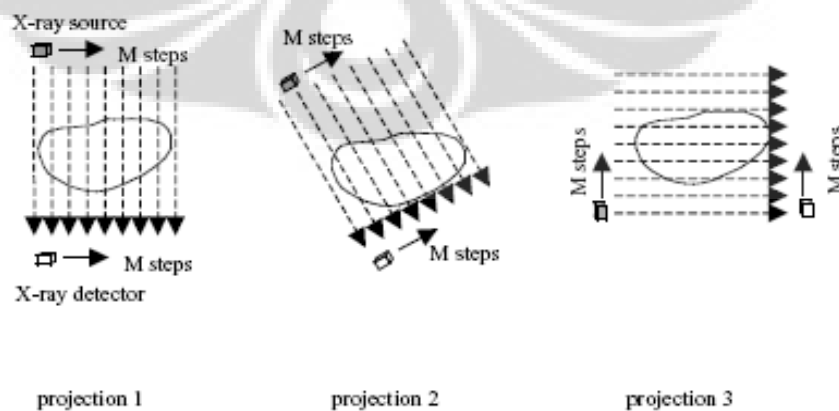


Figure 2.6. Three views of parallel beam projection of an EMI scanner

Since the research uses fan beam projection, a fan beam consisting of rays can also be figured from the above parallel beam description, as can be seen below.

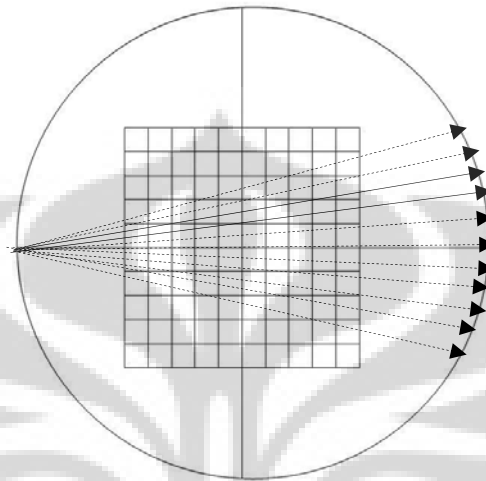


Figure 2.7. A fan beam consisting of rays

A complete CT projection will have a total number of sampling data resulted from total number of projection rays (M). So there will be M known values of sampling data. Since a CT projection is meant to solve N pixels value of an object matrix, then there will be M known data used to solve N unknown pixels value. A matrix operation of a complete CT projection can be shown as below:

$$\mathbf{W}_{(M \times N)} \cdot \mathbf{A}_{(N \times 1)} = \mathbf{P}_{(M \times 1)} \quad (2-12)$$

$$\begin{bmatrix} w_{11} \cdot \mu_1 + w_{12} \cdot \mu_2 + w_{13} \cdot \mu_3 + \dots + w_{1k} \cdot \mu_k + \Lambda + w_{1N} \cdot \mu_N \\ w_{21} \cdot \mu_1 + w_{22} \cdot \mu_2 + w_{23} \cdot \mu_3 + \dots + w_{2k} \cdot \mu_k + \Lambda + w_{2N} \cdot \mu_N \\ w_{31} \cdot \mu_1 + w_{32} \cdot \mu_2 + w_{33} \cdot \mu_3 + \dots + w_{3k} \cdot \mu_k + \Lambda + w_{3N} \cdot \mu_N \\ \Lambda \\ w_{m1} \cdot \mu_1 + w_{m2} \cdot \mu_2 + w_{m3} \cdot \mu_3 + \dots + w_{mk} \cdot \mu_k + \Lambda + w_{mN} \cdot \mu_N \\ \mathbf{K} \\ w_{M1} \cdot \mu_1 + w_{M2} \cdot \mu_2 + w_{M3} \cdot \mu_3 + \dots + w_{Mk} \cdot \mu_k + \dots + w_{MN} \cdot \mu_N \end{bmatrix} = \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ \mathbf{K} \\ P_m \\ \mathbf{K} \\ P_M \end{bmatrix}$$

Where: W = projection matrix as the projection function (line integrals)

A = reshaped object matrix as the object function ($f(x,y)$)

P = projection data matrix of projection data function ($p(x)$)

Since projection data, P , is a measured data matrix and object matrix, A , is unknown, from the above matrix operation it is then understood that the projection matrix, W , will be the main key of CT image reconstruction using the ART. An ART algorithm then must be started by solving the definition of projection matrix, W , before a backprojection process can be conducted. And since there is still no ART algorithm of fan beam projection, the research must start all the ART algorithm development by solving first the definition of projection matrix, W , as can be seen in the Chapter IV.

II.3.2.2. Backprojection: Initial Guess

From the matrix operation above, it can be interpreted as a simultaneous linear equations where there are M equations with N variables. But direct solving of the simultaneous linear equations with a simple backprojection:

$$A_r = W^{-1} \cdot P \quad (2-13)$$

where A_r is the resulting image value, will yield a very noisy resulting image and is very inefficient for a massive matrix operation. Therefore in an iterative reconstruction technique, the direct inverting backprojection is replaced with a more 'step to step' reconstruction process which is started with a nearest initial guess of A_r .

Since an inverse of W ($=W^{-1}$) consists of a matrix determinant and W -transpose ($=W^T$) and since the determinant is a complex parameter which will determine the gradient (difference) between W^T and W^{-1} , the initial guess used in the research are taken by simply replacing the W^{-1} with its transpose (W^T). Then by filtering the first resulting image value, resulted from initial guess, in an iterative process, the final resulting image value A_r , will be achieved. This filtering process is a converging process of adjusting the closest gradient of W^T to W^{-1} , resulting closest value of A_r to its origin value A . So if the initial guess of A_r is A_{r0} , then:

$$Ar_0 = W^T.P \quad (2-14)$$

where Ar_0 is an initial guess of image matrix value and W^T is the transpose of projection matrix W .

I.3.2.3. Blurring and Iterative Filtering Technique

Like the FBP technique, CT image reconstruction using the ART will suffer blurring in its resulting image. The blurring is naturally existing as a consequence of mathematical inverse when backprojection is started. Since in the ART, a backprojection process involved an inverse of projection matrix (W^{-1}), and since W is a spatial domain value, then inverting it will automatically change the domain to be frequencial. The changing of domain will cause a slight difference between resulting value and its origin, and when W^{-1} is used to solved the image value, the value will also suffer distortion from its origin. The accumulation of this distortion will be displayed as a blurring of the resulting image. So, like in FBP technique, the ART also suffer $1/r$ blurring.

Physical interpretation of the blurring function can be explained by a figure below:

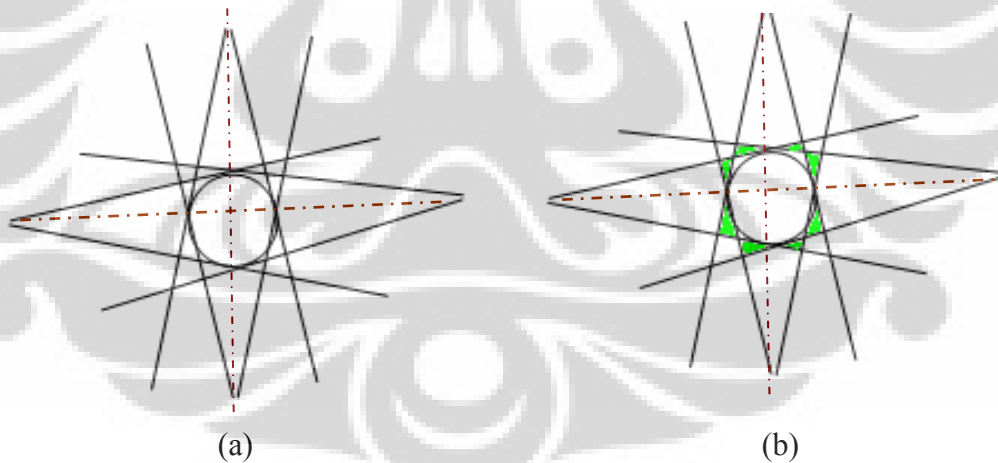


Figure 2.8.a. Four fan beam projecting a cylindrical rod.

b. Backprojection of (a) will cause a star like blurring around the rod boundary

If the object value is defined as A and the resulting image is A_r , then the blurring function $f(b)$, can be written down as :

$$f(b) = (A - A_r) \quad (2.15)$$

Since original object value is unknown, in physical measurement, this $1/r$ blurring value exists as a difference between measured data and first simulated data values:

$$f(b) = \lambda.(P - P_1) \quad (2.16)$$

Where:

$f(b)$ = blurring function

A = initial object value

A_r = resulting object value after backprojection process

P = measured projection data

P_1 = first simulated data

λ = an adjusted parameter

And from both above definitions of blurring function, the filtering process can be written down as:

$$A_{r1} - A_{r2} = \lambda. (P_1 - P_2) \quad (2.17)$$

$$A_{r2} = A_{r1} - \lambda.(P_1 - P_2) \quad (2.18)$$

And since from projection operation it is known that $P = W.A$, then :

$$A_{r2} = A_{r1} - \lambda.(W.A_1 - P_2) \quad (2.19)$$

For general number of j iteration, it will be:

$$A_{r_{j+1}} = A_{r_j} - \lambda.(W.A_j - P_{j+1}) \quad (2.20)$$

This is the iterative operation for solving filtered image value, but above equation can not be solved if W is not a square dimension projection matrix (views \neq samplings). To solve the problem, Van Cittert [10] suggested a 'reblurred' multiplier W^T (W -transpose) to bringback the dimension of resulting matrix ΔP to be the same with dimension of the initial matrix A_r , and adjust a new λ value. So the equation then can be generalized for any dimension of projection matrix W , as:

$$A_{r_{j+1}} = A_{r_j} - \lambda. W^T.(W.A_j - P_{j+1}) \quad (2.21)$$

This is the general iterative equation to achieve a filtered image started from initial guess value where $j = 0$. The λ value will determine the convergence speed and stability of the process. A too small value of λ will lead to slow convergence and too many iterations needed, while a very large value of λ will make the process unstable with a wide range among successive iterations. By adjusting the right λ value, a more converging value of A_r will be gained without

too many iterations. The initial guess of image value, λ adjustment and the number of iterations needed are the main parameters of an iteration operation.

This iterative filtering process has been the main issue of CT imaging. Since a filtering function is an iterative process and the iterations number can reach thousands to achieve an optimal resulting image, the duration of a filtering process will play a major part of the whole reconstruction process. The duration of a filtering process will mostly determine the reconstruction time. So beside its benefit in image filtering process, an iterative technique will suffer a drawback of longer reconstruction time than another technique like Filtered Back Projection (FBP) which doesn't need to conduct an iterative process.

II.4. Image Quality: Mean Square Error (MSE)

Since the research used simulated objects, to know the performance of the computer program, it can be analysed from both qualitative and quantitative point of view. Qualitatively, the analysis is conducted by simply observing blurring level of the resulting images and comparing the look of important small objects in the images. However as compared images are almost similarly blurred, then they will also need a quantitative value as a comparing parameter. A popular quantitative value in simulated CT imaging is Mean Square Error (MSE)/pixel which can tell relative error level of the resulting image. MSE can be counted from difference between image pixels value (ar) and object pixels value (a), as:

$$\Delta a_k = a_k - ar_k \quad (2-22)$$

Where : Δa_k = difference between k-pixel values of object and image matrix.

a_k = k-pixel value of object matrix

ar_k = k-pixel value of resulting image matrix

and Mean Square Error (MSE)/pixel will be:

$$\text{MSE/pixel} = (1 / N) \sum_{k=1}^N | \Delta a_k |^2 \quad (2-23)$$

where: MSE/pixel = Mean Square Error

N = Number of object pixels

k = pixels number = 1, 2, 3, ..., N

CHAPTER III

METHODOLOGY

To achieve the purpose of the research mentioned in the introduction, the methodology consists of:

1. Matlab 6,0 Programming:

- 1.1. Algorithm: to develop the algorithm of a third generation CT image reconstruction based on Algebraic Reconstructive Technique (ART).
- 1.2. Programming: to otomatize the reconstruction using Matlab programming based on the developed algorithm.

2. Parameter Optimization

Conducting image reconstructions of a simulated object to optimize input parameters for best performance of the computer program. All simulations were processed by: a notebook computer, processor: Intel Core(TM)2 Duo CPU T7300@2.00 GHz, RAM 1014 MB. All projections based on a Third (3rd) generation CT projection design as figured below:

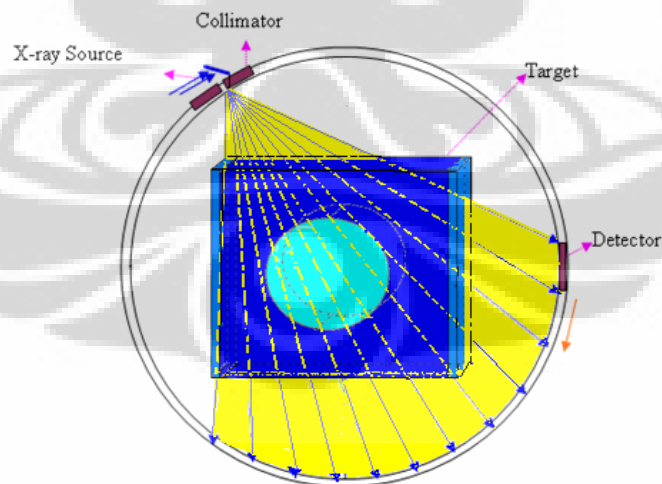


Figure 3.1. Projection design of a third generation CT-Scanner

Simulations was conducted on four main parameters, which are:

2.1. Beam shape

Changing parameter : beam shape

i. Parallel beam of $M=20 \times 20$ projecting object of $N=20 \times 20$

ii. Fan beam of $M=16 \times 16$ projecting object of $N=16 \times 16$

2.2. Angular width

i. Beam : $(m_\beta \cdot \Delta\beta) = 2 \cdot \arctan(a_0 / (R + a_0))$ (3-1)

where:

m_β = number of rays per beam

$\Delta\beta$ = angular width of a ray

a_0 = half width of square object matrix

R = radius of CT system

Changing parameter = m_β

- $m_\beta = 16$

- $m_\beta = 32$

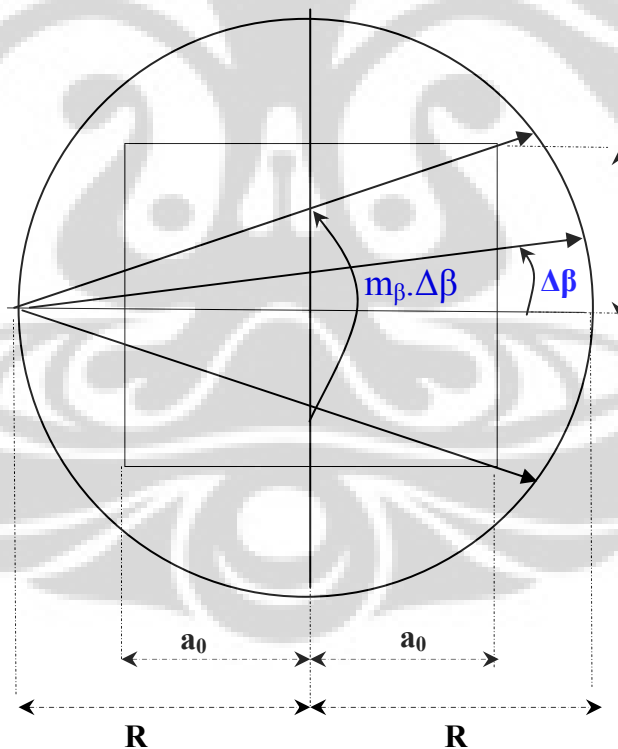


Figure 3.2. Angular width of a beam ($m_\beta \cdot \Delta\beta$) based on matrix width

ii. Ray : $\Delta\beta = \arctan(d / (R + (a_0 - f \cdot d)))$, (3-2)

where:

$\Delta\beta$ = angular width of a ray

a_0 = half width of a simulated square object matrix

R = radius of CT system

d = width of a pixel

f = a changing integer

Changing parameter = f

- Object of $N = 16 \times 16$ was projected for $f = 0; 1; 2; 3; 4$ and 5
- Object of $N = 32 \times 32$ was projected for $f = 1$ and $f = 8$
- Object of $N = 64 \times 64$ was projected for $f = 1$

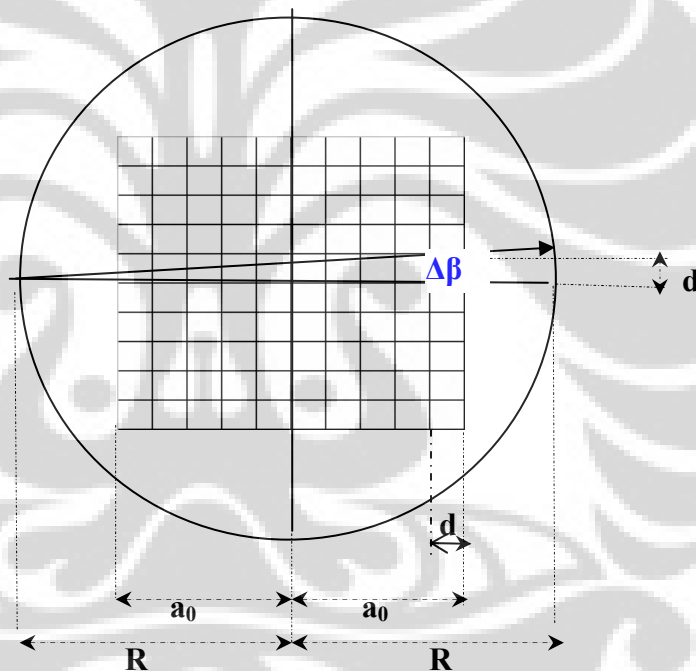


Figure 3.3 Angular width of a ray ($\Delta\beta$) based on a pixel width

2.3. Reconstruction time (\bar{t})

Reconstruction time (\bar{t}) was measured by a timer which is started as the backprojection program was runned and stopped as the resulting image had been displayed on the screen..

Changing parameter: number of pixels (N) and number of projections (M)

i. $N = 16 \times 16$ and

- $M = 16 \times 16$
 - $M = 32 \times 32$
- ii. $N = 32 \times 32$ and
- $M = 32 \times 32$
 - $M = 64 \times 64$
- iii. $N = 64 \times 64$ and $M = 64 \times 64$

2.4. Number of iterations (j)

Changing parameter: $j = 0; 10; 100; 1000; 10.000$ and 20.000

3. Experiment

3.1. Implementation of the optimized parameters to reconstruct an image of a Shepp-Logan phantom of:

- i. $N = 32 \times 32$ using $M = 64 \times 64$ projections

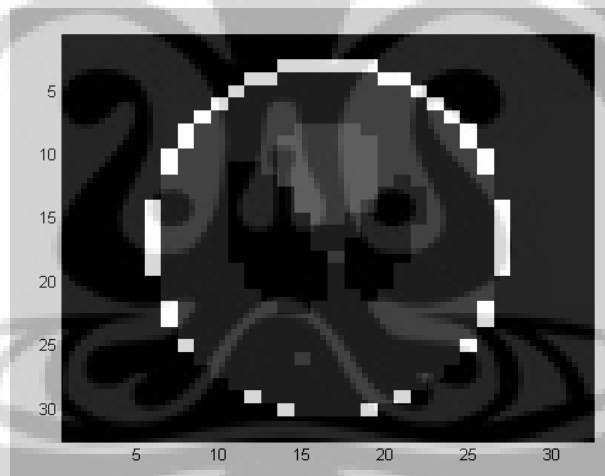


Figure 3.4. A Shepp-Logan phantom of $N = 32 \times 32$

Changing parameters:

- Convergence speed (λ)
- Number of iterations (j)

- ii. $N = 64 \times 64$ using $M = 128 \times 64$ projections

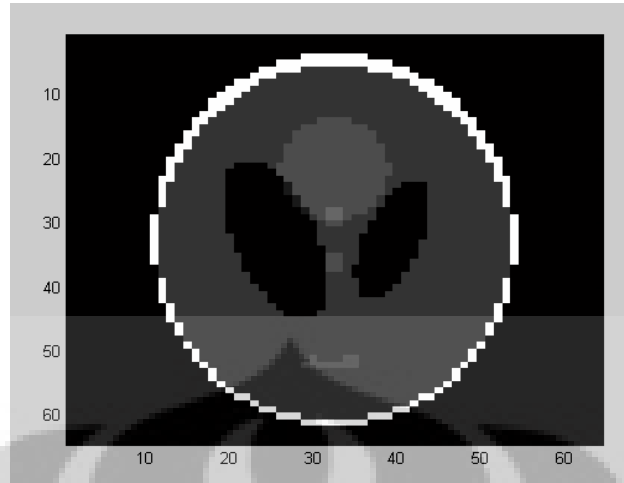


Figure 3.4. A Shepp-Logan phantom of $N = 64 \times 64$

Changing parameters:

- Convergence speed (λ)
- Number of iterations (j)

3.2. Comparison of Algebraic Reconstruction Technique (ART) with Filtered Back Projection (FBP) technique.

Comparison parameter:

- Qualitative: blurring level
- Quantitative: Mean Square Error (MSE) / pixel

3.3. Comparison of Algebraic Reconstruction Technique (ART) with another iterative technique with different initial guess method (from [10]).

CHAPTER IV

ALGEBRAIC RECONSTRUCTIVE TECHNIQUE (ART)

ALGORITHM

IV.1. Third Generation CT-Scanner Projection Plan

In a simplified projection mechanism of a third generation CT, a wave of high energy electron (≈ 70 keV) emitted by the cathode of an X-ray tube is struck into the anode (made of heavy atomic number medium, eq. Tungsten). The physical interaction will cause the emission of a wave of X-ray (photon) beam with straight direction to a detector array that consists of a number of detectors (m_β). This photon beam is first shaped by a hole, called as a focal spot, producing a wide fan shape photon beam with a specific angle ($30^\circ - 60^\circ$). The fan beam produced from the focal spot then shaped again by a collimator to limit the thickness (3^{rd} dimension side) of the target slice.

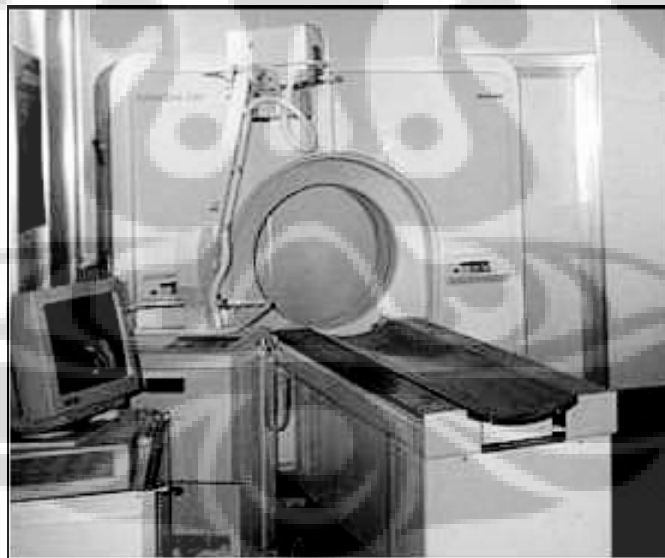


Figure 4.1. A Third Generation CT-Scanner^[22]

If succeeded in passing the target, the fan beam will be measured by a detector array. The measured data of the non projecting part of the fan beam will become a reference intensity of photon beam (I_0) while the measured data of the projecting part will become a projection data (I_1). The length of each detector of

the detector array are the same and will determine the angular width of a projection ray ($\Delta\beta$).

A position of the source-detector couple for one projection is identified as a projection direction/view (α). After measured by the detector array and a projection data array is processed and saved, then the position of source-detector couple is shiftly turned by a small angle ($\Delta\alpha$) to get another projection data array. This shifting process is then repeated several times to get total number of projection direction from 0^0 to 360^0 (m_α), and all saved array data will become total projection data (sampling data) of the target slice. A computer program of image reconstruction can be executed after all sampling data is completed.

IV.2. Projection Data Matrix (P)

All resulting data arrays are simultaneously put into a total data array, $P_{(M \times 1)}$, with M is the total number of sampling data, where:

$$M = m_\alpha \cdot m_\beta$$

where: m_α = total number of views (beam projection directions)

m_β = total number of samplings (ray projection directions in a beam)

Each sampling data is a logarithmic function of the relative photon intensity of m-projection ray.

$$P_{(M \times 1)} = \begin{bmatrix} P_1 = \ln I_0 / I_{\text{beam}_1 \text{ray}_1} \\ P_2 = \ln I_0 / I_{\text{beam}_1 \text{ray}_2} \\ P_3 = \ln I_0 / I_{\text{beam}_1 \text{ray}_3} \\ \vdots \\ P_m = \ln I_0 / I_{\text{beam}_i \text{ray}_j} \\ \vdots \\ P_M = \ln I_0 / I_{\text{beam}_{m_\alpha} \text{ray}_{m_\beta}} \end{bmatrix} \quad (4-1)$$

where: $P_{(M \times 1)}$ = a projection data array of M samplings

P_m = a projection data of sampling m

I_0 = initial photon intensity (measured by reference detector)

- $I_{\text{beam-}i \text{ ray-}j}$ = measured photon intensity of beam- i and ray- j
 i = counter for beams (views) = 1, 2, 3, ..., m_α
 j = counter for rays (samplings) in a beam = 1, 2, 3, ..., m_β
 m = counter for total samplings = 1, 2, 3, ..., M

IV.3. Object Matrix (A)

The target object is divided into many slices depending on the prior target size and diagnostic need. Every slice is put exactly at the center of an imaginary table that also put at the center of the source-detector circle. The table is divided by n rows and n columns forming a square matrix ($n \times n$) consisting of N pixels. The CT system is then assumed to be a Cartesian coordinate system.

Each pixel of the matrix is a square resulted from intersection between four lines, which are: $Y = y_s$, $Y = y_{s+1}$, $X = x_r$ and $X = x_{r+1}$, with :

$$\begin{aligned}
 d = \text{pixel width} &= x_{r+1} - x_r && (r = 1, 2, 3, \dots, M), \\
 &= \text{pixel height} = y_{s+1} - y_s && (s = 1, 2, 3, \dots, N)
 \end{aligned}$$

IV.3.1. Pixel Numbering

Since in CT projection, the object matrix consists of pixels, identification of each pixel is the basic definition of an object function. The numbering order of pixel-1 to N can be figured as below:

n	2n	3n	N
.
.
.	.	.	.	k	.	.	.
.
3	n+3	2n+3
2	n+2	2n+2
1	n+1	2n+1	n

Figure 4.2. A matrix of N pixels, k is the pixel number, $k=1,2,3,\dots,N$

To identify each pixel position in the Cartesian coordinate system, first the pixel number (k) must be defined as a function of its row (r) and column (s) number, and a definition of k is:

$$k = (n-1).r + s$$

where: r = column number of pixels in object matrix

$$= 1,2,3,\dots, n$$

s = row number of pixels in object matrix

$$= 1,2,3,\dots,n$$

n = number of rows or columns of the object matrix

In Matlab programming, an alternative definition of k is used. A short syntax using 'round' function has been effective to number matrix pixels. The short syntax is:

```
=====
...
for k=1:N
    r=round(k/n-0.5001)+1;
    s=rem(k,n);
    if s==0
        s=r-1;
        s=n;
    end
end
...
=====
```

Here the 0.5001 constant instead of 0.5 is used to make sure that all non-integer k values will be rounded down to smaller integer.

Although this Matlab syntax is already effective, the previous definition of k as $f(r,s)$ is very important because later it will be the key of object matrix back rotation method used in the research.

If the width of a square matrix is $2a_0$ and the width of a pixel is d then the coordinate of pixels can be defined as:

$$X(r) = -a_0 + d.(r-1)$$

$$Y(s) = -a_0 + d.(s-1)$$

IV.4. Projection Function

IV.4.1. Ray Linear Equation

A beam is emitted by a point source (S) which distance from central point $O_{(0,0)}$ is \overline{OS} (radius). The beam direction is α which is defined as the angle between line \overline{OS} and \overline{OY} . The beam consists of rays which each is shaped by two lines, defined as $Y_1 (=f(x)_1)$ and $Y_2 (=f(x)_2)$. The direction of a ray is β , defined as the angle between \overline{OS} and Y_1 . Figure of a ray projecting a pixel can be seen below

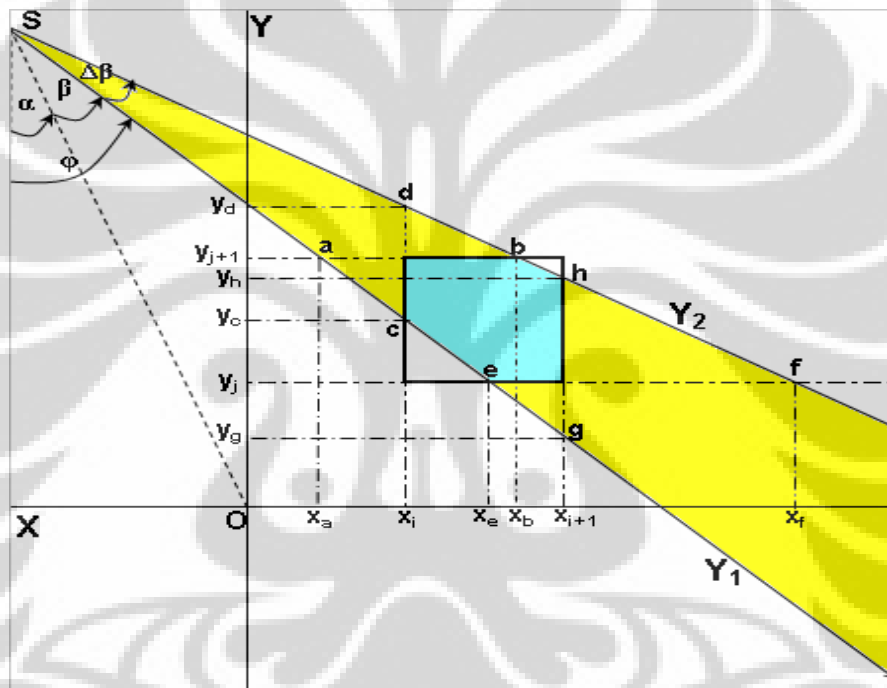


Figure.4.3. A ray projecting a pixel

The general equation of Y_1 and Y_2 are :

$$Y_1 = m_1X + y_{01} \text{ and } Y_2 = m_2X + y_{02},$$

where:

$$\begin{aligned} m_1 &= -\cotg(\alpha + \beta) \\ &= -\cotg(\varphi) \end{aligned}$$

and :

$$\begin{aligned} m_2 &= -\cotg((\alpha + \beta) + \Delta\beta) \\ &= -\cotg(\varphi + \Delta\beta) \end{aligned}$$

And if

$$\overline{OS} = R,$$

then :

$$y_{01} = R.(\cos(\alpha) + m_1.\sin(\alpha))$$

$$y_{02} = R.(\cos(\alpha) + m_2.\sin(\alpha))$$

where :

y_{01} = ordinate of intersection point between Y_1 and $X=0$

y_{02} = ordinate of intersection point between Y_2 and $X=0$

IV.4.2. Intersection Points

The position of intersection points (a,b,c,d,e,f,g and h) will determine the intersection shape, where:

- a (x_a, y_{s+1}) : intersection point between Y_1 and $Y = y_{s+1}$
- b (x_b, y_{s+1}) : intersection point between Y_2 and $Y = y_{s+1}$
- c (x_r, y_c) : intersection point between Y_1 and $X = x_r$
- d (x_r, y_d) : intersection point between Y_2 and $X = x_r$
- e (x_c, y_s) : intersection point between Y_1 and $Y = y_r$
- f (x_d, y_s) : intersection point between Y_2 and $Y = y_r$
- g (x_{r+1}, y_g) : intersection point between Y_1 and $X = x_{r+1}$
- h (x_{r+1}, y_h) : intersection point between Y_2 and $X = x_{r+1}$

And definition of intersection points can be solved as:

$$x_a = 1/m_1 (y_{s+1} - y_{01})$$

$$x_b = 1/m_2 (y_{s+1} - y_{02})$$

$$x_e = 1/m_1 (y_s - y_{01})$$

$$x_f = 1/m_2 (y_s - y_{02})$$

$$y_c = m_1 x_r + y_{01}$$

$$y_d = m_2 x_r + y_{02}$$

$$y_g = m_1 x_{r+1} + y_{01}$$

$$y_h = m_2 x_{r+1} + y_{02}$$

IV.4.3. Intersection Shape

The most important step of defining a projection function is the identification of all possible intersection shapes between a ray and a pixel. Since it is the ray direction that will determine the shape and since the direction of a ray is a combination of beam direction (α) and ray direction referenced to the beam (β), so the direction of ray will be determined by the addition of both angle ($\varphi = \alpha + \beta$). So the ray direction is identical with φ and it will be categorized by the angle.

Since for each quadrant, the most complete ray directions of a beam is existing for beam direction of $\alpha = 0^\circ, 90^\circ, 180^\circ$ and 270° , then one of these four beam direction will be able to represent all ray directions. For practical reason, the beam direction of $\alpha = 90^\circ$ is taken here as the reference to categorize ray directions, as seen in figure below:

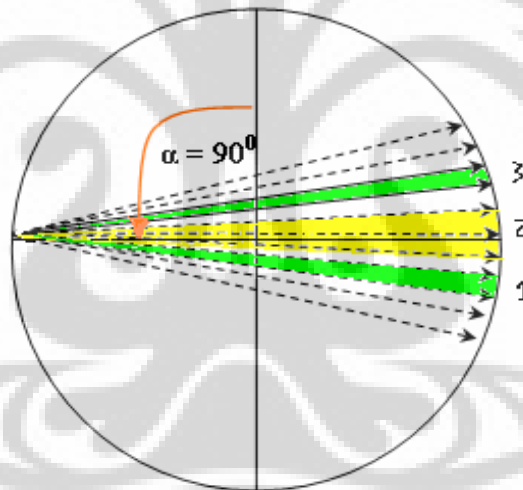


Figure 4.4. A beam direction of $\alpha = 90^\circ$, with three categories of ray direction

As seen in above figure there are three ray direction categories of:

1. $0 < \Phi < (\pi/2 - \Delta\beta)$
2. $(\pi/2 - \Delta\beta) \leq \Phi \leq (\pi/2)$
3. $(\pi/2) < \Phi < (\pi - \Delta\beta)$

Each category can be seen clearly in three separated figures below:

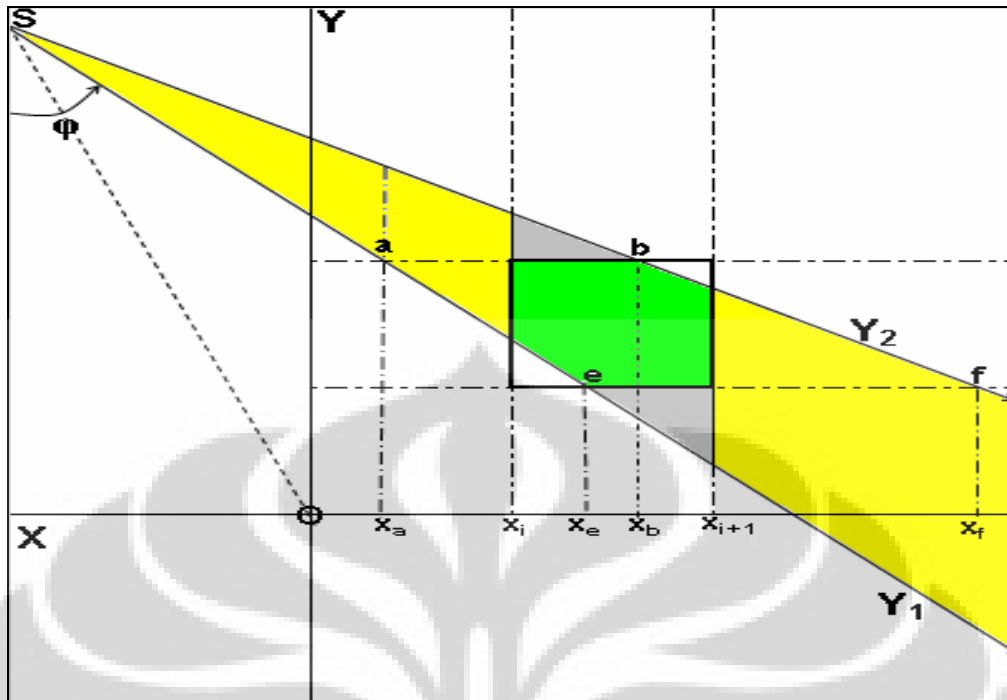


Figure 4.5a. Direction category 1: $0 < \Phi < (\pi/2 - \Delta\beta)$

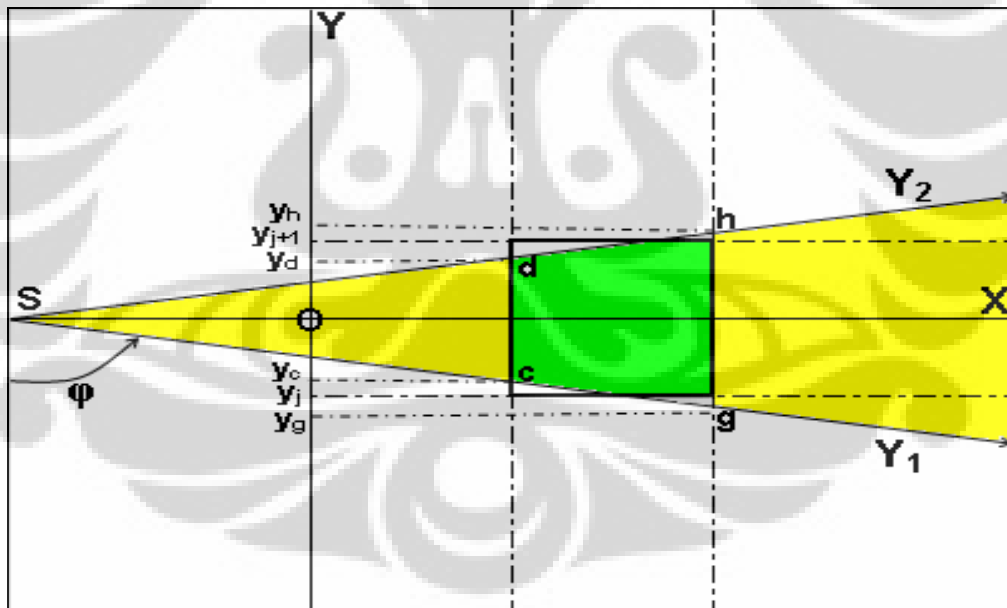


Figure 4.5b. Direction category 2: $(\pi/2 - \Delta\beta) \leq \Phi \leq (\pi/2)$

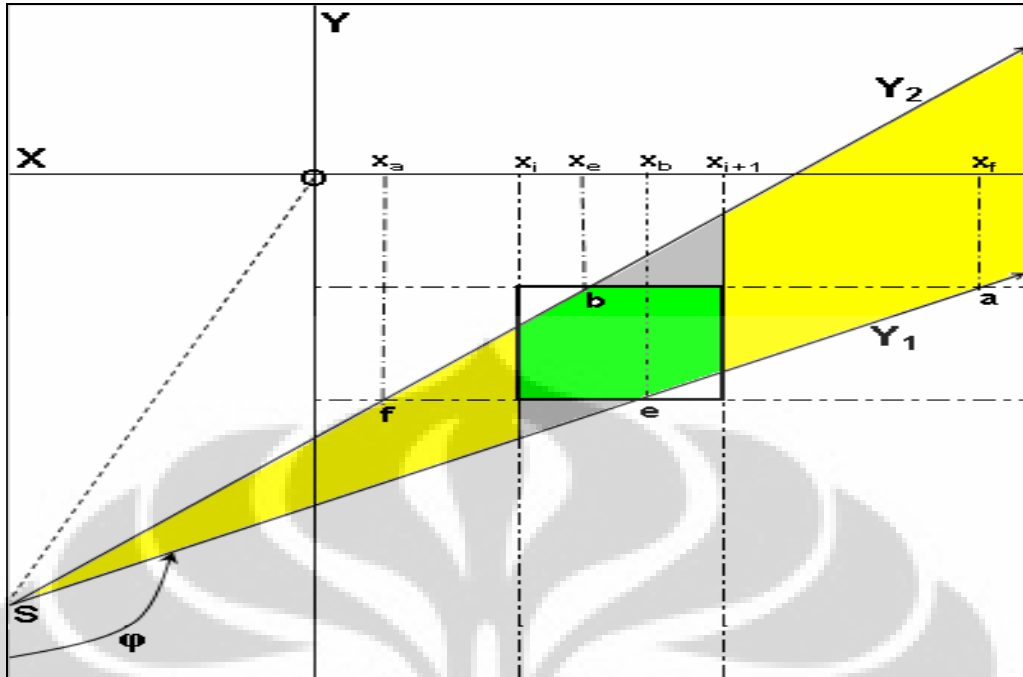


Figure 4.5c. Direction category 3: $(\pi/2) < \Phi < (\pi - \Delta\beta)$

Each direction of ray will determine the slope of the ray which then can be used to define the position of intersection points. As can be seen in above figures, the intersection points of a, b, e and f respectively will have three possible positions referenced to range of $x = x(r) \rightarrow x(r+1)$, which are :

- a. $x_n = x_1$, if $x_n < x(r)$
- b. $x_n = x_2$, if $x(r) < x_n < x(r+1)$
- c. $x_n = x_3$, if $x(r+1) < x_n$

By combining all possible positions of the four intersection points, then total number of all possible combinations will be:

$$\begin{aligned}
 X_{a1-3}; X_{b1-3}; X_{e1-3}; X_{f1-3} &= X_{an} \cap X_{bn} \cap X_{en} \cap X_{fn} \\
 &= 3 \times 3 \times 3 \times 3 \\
 &= 81 \text{ combinations}
 \end{aligned}$$

Then by implementing slope conditions to each direction category, these combinations can be sorted to get final combinations of the intersections points position that fulfill the conditions. An example for the resulting tables of the sorting for category 1 can be seen in the table A1 at Appendices. Using this tables, all intersection shapes of the three categories can be defined. By illustrating the shapes figures from their definitions, the area of the shapes then

can be solved as described in the next section. All resulting shapes illustration figures and their area for each category can be seen in table A2 at Appendices.

IV.4.4. Intersection Area

The area of intersection shape of figure 4.3 can be solved by subtracting the main area (w_U) by both top and under rest area (w_a and w_b respectively):

$$\begin{aligned}
 w_U &= \int_x^{x_{i+1}} (Y_2 - Y_1) dx = \int_x^{x_{i+1}} \{(m_2x + y_{02}) - (m_1x + y_{01})\} dx \\
 &= \{1/2 (m_2 - m_1)(x_{r+1}^2 - x_r^2) + (y_{02} - y_{01})(x_{r+1} - x_r)\} \\
 w_a &= \int_{x_r}^{x_b} (Y_2 - Y_{s+1}) dx = \int_{x_r}^{x_b} \{(m_2x + y_{02}) - y_{s+1}\} dx \\
 &= 1/2 (m_2)(x_b^2 - x_r^2) + (y_{02} - y_{s+1})(x_b - x_r) \\
 w_b &= \int_{x_e}^{x_{r+1}} (Y_r - Y_1) dx = \int_{x_e}^{x_{r+1}} \{y_s - (m_1x + y_{01})\} dx \\
 &= -1/2 (m_1)(x_{r+1}^2 - x_e^2) + (y_s - y_{01})(x_{r+1} - x_e)
 \end{aligned}$$

And the projection function element of a ray projecting a pixel can be solved as the intersection area between the two as :

$$w = w_U - (w_a + w_b)$$

Since intersection shape of a hexagonal geometry is the most complex shape with top and bottom rest area, this shape will become the reference shape for solving a general area integral formula. Other intersection geometry will be less complete. So from the resulting area integral formula above, a general formula can be solved by simply changing all area integral limits to general limits.

And the general area integral formula in definitive form will be:

$$\begin{aligned}
 w_U &= 1/2 (m_2 - m_1).(x_{2U}^2 - x_{1U}^2) + (y_{02} - y_{01}).(x_{2U} - x_{1U}) \\
 w_a &= 1/2 (m_2).(x_{2a}^2 - x_{1a}^2) + (y_{02} - y_{s+1}).(x_{2a} - x_{1a}) \\
 w_b &= 1/2 (m_1).(x_{2b}^2 - x_{1b}^2) + (y_s - y_{01}).(x_{2b} - x_{1b})
 \end{aligned}$$

And

$$w = w_U - (w_a + w_b)$$

Where:

- x_{1U} = first integral limit of w_U
- x_{2U} = second integral limit of w_U
- x_{1a} = first integral limit of w_1
- x_{2a} = second integral limit of w_1
- x_{1b} = first integral limit of w_1
- x_{2b} = second integral limit of w_U

By defining the area integral limits of all possible intersection shapes of the three projection direction categories (1,2 and 3 respectively), this general area integral formula becomes a general projection function for the three categories.

IV.5. Initial Quadrant Projections

Since a square matrix has four symmetrical properties (horizontal, vertical and two diagonals), then by dividing the matrix to four quadrants, it can be assumed as a circle consisting of four equal shape geometries. Then by solving the definition of all projections in the initial quadrant, it can be generated for other three quadrants. The question is which part of the projection circle will be the initial quadrant with minimal number of definitions.

To answer the question, the reference parameter used here is the beam direction. By choosing a right beam direction of any quadrant, the ray directions of the chosen beam should be able to represent all ray directions existing in a quadrant. This is the representative requirement of the beam. This represented quadrant then will become the initial quadrant.

By a simple logic it can be found that the representative requirement is met by four beam directions of $\alpha = 0^0, 90^0, 180^0$ and 270^0 . For practical reason the beam direction of 90^0 is chosen in the research. The beam direction can be seen as below:

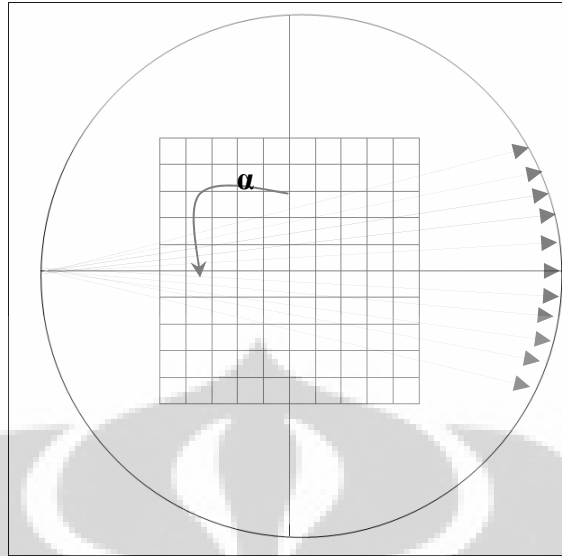


Figure 4.6. A representative beam direction of $\alpha = 90^{\circ}$

With this position, the beam has all three ray directions (φ) of category 1, 2 and 3 mentioned before. And as can be seen in the figure 2.4, these direction categories can represent all possible ray directions of the quadrant limited by diagonals of $\alpha = 45^{\circ}$ to $\alpha = 135^{\circ}$. This quadrant will be the initial quadrant (Quadrant 1) and following α direction the next quadrants will be quadrant 2, 3 and 4 respectively. All quadrants position can be seen in figure below.

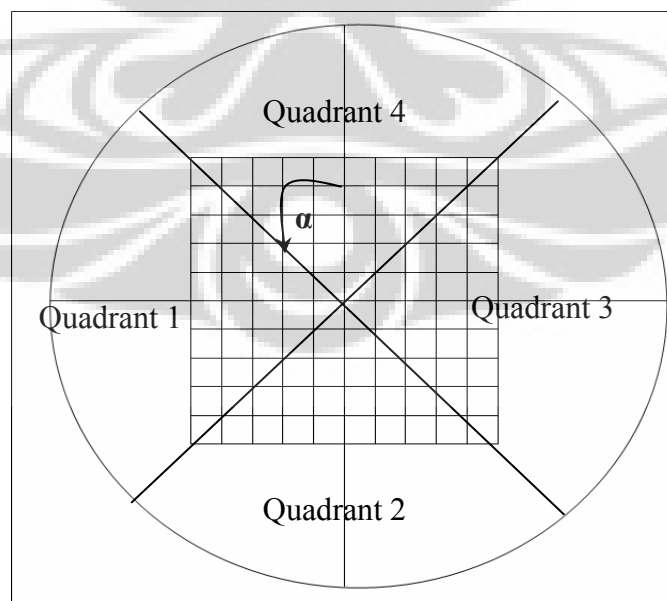


Figure 4.7 Four quadrants positions, Quadrant 1 is the initial quadrant.

From the figure, the angular width of each quadrant will be:

- i. Quadrant 1 : $45^0 \leq \alpha < 135^0$
- ii. Quadrant 2 : $135^0 \leq \alpha < 225^0$
- iii. Quadrant 3 : $225^0 \leq \alpha < 315^0$
- iv. Quadrant 4 : $315^0 \leq \alpha < 45^0$

After the initial quadrant is solved, then all ray projection directions of this quadrant can be categorized into the three categories (1, 2 and 3) mentioned before. So the general area integral formula of these categories will become projection function of the initial quadrant.

IV.6. Radon Transform: Imaginary Object Matrix Back Rotation

After solving the projection function of Quadrant 1, according to Radon transform the area integral formula now can be generated by simply rotating the initial quadrant beams to next quadrant, but even though the geometry of resulting intersection shapes are the same, the coordinate of intersection points will change. It means the intersection points (a,b,c,d,e,f,g,h) will be different, so the result of area integral formula will be different and it is fail to be generated.

The answer to the problem is an imaginary object matrix back rotation while the quadrant projections stay still. Since the rectangular object matrix now can be assumed as a circle, then rotating the quadrant beams to one direction while the object matrix stays still is the same as rotating the object matrix by a factor of 90^0 to opposite direction while the quadrant beams stay still.

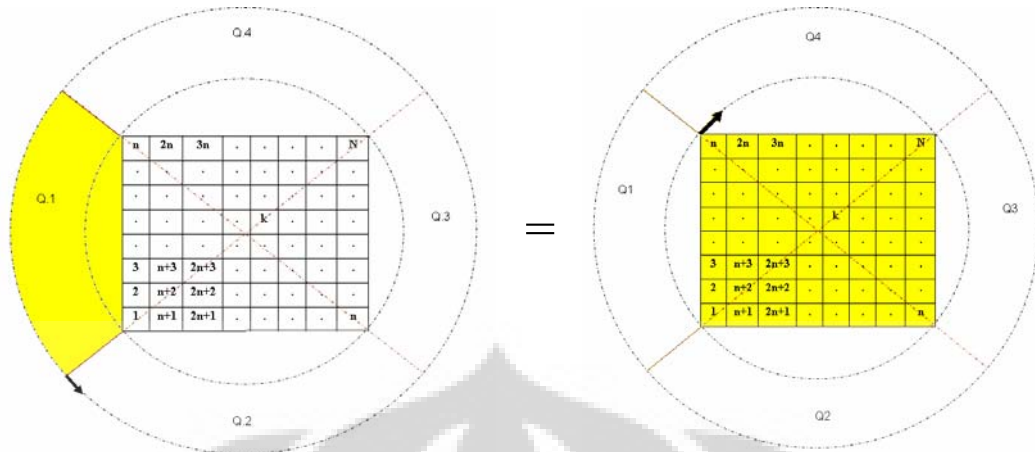


Figure 4.8. Rotating a quadrant projections when the object matrix is fixed (left) is the same as rotating object matrix to opposite direction by a factor of 90° when the quadrant projections is fixed (right)

If the initial coordinate of object matrix is the same as the resulting projection coordinate (X', Y') and the resulting coordinate of ϕ rotated object matrix is the same as the initial projection coordinate (X, Y) then according to Radon Transform:

$$\begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} \cos \phi & -\sin \phi \\ \sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} X' \\ Y' \end{bmatrix}$$

And since the coordinate of pixels is $(X(r), Y(s))$, the imaginary new pixels coordinate will be:

$$X(r) = X'(r) \cdot \cos(\phi) - Y'(s) \cdot \sin(\phi)$$

$$Y(s) = X'(r) \cdot \sin(\phi) - Y'(s) \cdot \cos(\phi)$$

And since the resulting projection coordinate (X', Y') is fixed in Quadrant 1, the resulting projection coordinate will be always be the same as $(X(r_1), Y(s_1))$.

And the equation can be rewritten as:

$$X(r) = X(r_1) \cdot \cos(\phi) - Y(s_1) \cdot \sin(\phi)$$

$$Y(s) = X(r_1) \cdot \sin(\phi) - Y(s_1) \cdot \cos(\phi)$$

And now by using this new pixels coordinate, new quadrant pixels number of k_q ($q = 1, 2, 3, 4$) can be defined and the ϕ rotated object matrix coordinate now has been adjusted.

IV.6.1. A Quadrant – Object Matrix Back Rotation

Rotating all beams of Quadrant 1 to Quadrant 2 is the same as rotating the object matrix to opposite direction by $\varphi = -90^0$ when the beams are fixed. The previous columns/rows of the matrix in Quadrant 1 will become the rows/columns on Quadrant 4. And the new coordinate of pixels will be:

$$X(r_4) = X(r_1).\cos(-90^0) - Y(s_1).\sin(-90^0)$$

$$Y(s_4) = X(r_1).\sin(-90^0) - Y(s_1).\cos(-90^0)$$

:So: $X(r_4) = Y(s_1)$

$$Y(s_4) = -X(r_1)$$

resulting:

$$r_4 = s_1$$

$$s_4 = -r_1$$

where: r_4 = imaginary columns number in Quadrant 4

s_4 = imaginary rows number in Quadrant 4

s_1 = original rows number in Quadrant 1

r_1 = original columns number in Quadrant 1

And by defining $k = f(r,s)$, pixels number in Quadrant 4 (k_4) can be easily defined as a function of initial (r_1,s_1) as:

$$k_4 = (n.s_1 - r_1) + 1$$

where: n = number of rows/columns.

k_4 = imaginary pixels number in Quadrant 4

IV.6.2. A Half Circle – Object Matrix Back Rotation

After projection function of Quadrant 4 have been solved, the object matrix now can proceed the rotation to Quadrant 3 and 2 and pixels number of the quadrants can be defined as in previous process.

But more efficient way is suggested here by a half circle rotation of object matrix. By putting back both the object matrix and the two quadrant beams to its original position in Quadrant 1, the two quadrant (1 and 2) projections now become a half circle projections. And by the same logic as the quadrant projections rotation, the object matrix coordinate for the other half circle can be solved by 180^0 back rotation of the object matrix. And using again the Radon transform for $\varphi = -180^0$, the new coordinate ($X(r),Y(s)$) of object matrix will be:

$$X(r_{2,3}) = X(r_{1,4}) \cdot \cos(-180^0) - Y(s_{1,4}) \cdot \sin(-180^0)$$

$$Y(s_{2,3}) = X(r_{1,4}) \cdot \sin(-180^0) - Y(s_{4,1}) \cdot \cos(-180^0)$$

So:

$$X(r_{23}) = - X(r_{14})$$

$$Y(23) = - Y(r_{14})$$

resulting :

$$r_{2,3} = - r_{1,4}$$

$$s_{2,3} = - s_{1,4}$$

And by defining k_{23} as a function of the initial $(r_{1,4}, s_{1,4})$ and since $k_{1,4}$ is the same as the initial k_1 then it can be found that:

$$k_{23} = - k_{14}$$

And since it is impossible for pixels number k to be a negative value, it can be interpreted that pixels number of k_{23} are taken by reversing order of k_{14} . So since $k_{14} = 1, 2, 3, \dots, N$, then $k_{23} = N, (N-1), (N-2), \dots, 1$, from which the definition of k_{23} as a function of k_{14} will be:

$$k_{23} = N - k_{14} + 1$$

where: N = number of pixels

k_{23} = imaginary pixels number of Quadrant 2 and 3

k_{14} = original pixels number (= k_1) of Quadrant 1 and 4

IV.7. Projection Matrix

After solving all the resulting pixels number of the imaginary ϕ rotated object matrix, then a projection function for all quadrants have been solved. It will be in form of a three dimension projection matrix $W_{(m_\alpha \times m_\beta \times N)}$, which elements are $w_{i,j,k}$ where:

i = counter for beam directions (α)

$$= 1, 2, 3, \dots, m_\alpha$$

j = counter for ray directions in a beam (β)

$$= 1, 2, 3, \dots, m_\beta$$

k = pixels number

$$= 1, 2, 3, \dots, N$$

And: m_α = total number of beam directions

m_β = total number of ray directions in a beam

To be used in a backprojection matrix operation, the 3D projection matrix, $W_{(m_\alpha \times m_\beta \times k)}$, must be first reshaped to a 2D matrix, $W_{(m_\alpha \cdot m_\beta \times k)}$, and the element notation is redefined from $w_{i,j,k}$ to $w_{ij,k}$. The resulting 2D matrix will be :

$$W_{(m_\alpha \times m_\beta \times k)} = W_{(m_\alpha \cdot m_\beta \times k)}$$

$$= \begin{bmatrix} w_{111} & w_{112} & w_{113} & \dots & w_{11k} & \dots & w_{11N} \\ w_{121} & w_{122} & w_{123} & \dots & w_{12k} & \dots & w_{12N} \\ w_{131} & w_{132} & w_{133} & \dots & w_{13k} & \dots & w_{13N} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \Lambda & K & K & \dots & K & \dots & K \\ w_{ij1} & w_{ij2} & w_{ij3} & \dots & w_{ijk} & \dots & w_{ijN} \\ K & K & K & \dots & K & \dots & K \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ w_{m_\alpha m_\beta 1} & w_{m_\alpha m_\beta 2} & w_{m_\alpha m_\beta 3} & \dots & w_{m_\alpha m_\beta k} & \dots & w_{m_\alpha m_\beta N} \end{bmatrix}$$

And since each element $w_{ij,k}$ of above matrix represents a j-ray of a i-beam projecting a k-pixel, then it can be rewritten as an element $w_{m,k}$ which represents a m-ray of total M-ray in CT system, projecting a k-pixel.

$$W_{(m_\alpha \cdot m_\beta \times k)} = W_{(M \times k)}$$

$$= \begin{bmatrix} w_{11} & w_{12} & w_{13} & \dots & w_{1k} & \dots & w_{1N} \\ w_{21} & w_{22} & w_{23} & \dots & w_{2k} & \dots & w_{2N} \\ w_{31} & w_{32} & w_{33} & \dots & w_{3k} & \dots & w_{3N} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \Lambda & K & K & \dots & K & \dots & K \\ w_{m1} & w_{m2} & w_{m3} & \dots & w_{mk} & \dots & w_{mN} \\ K & K & K & \dots & K & \dots & K \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ w_{M1} & w_{M2} & w_{M3} & \dots & w_{Mk} & \dots & w_{MN} \end{bmatrix}$$

Where:

$$w_{m,k} = w_{ij,k}$$

$$m = \text{row number}$$

$$= f(i,j)$$

$$= (i-1)m_\alpha + j$$

$$= \text{counter for ray directions } (\varphi) \text{ in CT system}$$

$$= 1, 2, 3, \dots, M$$

IV.7.1. Initial Projection from A Projection Matrix: Practical Use

From the resulting projection matrix $W_{(M \times k)}$ above, it is unavioded that the initial projection of the CT projection plan will be the beam of $i = 1$. So the initial projection of the research will be the beam of $\alpha_1 = 45^0$. But since in practical use, it will be mechanically difficult to start the projection exactly from $\alpha_1 = 45^0$, and since the rays of initial beam is represented in the matrix by the first rows of $i = 1$, then by choosing the right i , more comfortable position to start the projection can be achieved.

The equation for renumbering the rows of a projection matrix to a new row number can be written down as :

$$m_n = m + (\alpha_{1n} - \alpha_1) / \Delta\alpha$$

Where:

m_n = new row number

m = previous row number

α_{1n} = new initial projection direction

α_1 = previous initial projection direction

$\Delta\alpha$ = indices of beam direction

For example if an initial projection direction (α_1) of the research want to be replaced from 45^0 to 90^0 and if indices ($\Delta\alpha$) is 5^0 , then a new row number will be:

$$\begin{aligned} m_p &= m + (\alpha_{1n} - \alpha_1) / \Delta\alpha \\ &= m + (90^0 - 45^0) / 5^0 \\ &= m + 9 \end{aligned}$$

where: m_p = practical row number

m = research row number

The result of above renumbering operation is a new projection matrix which elements and their columns number are exactly the same as the previous, but with different rows number. From the example, changing the starting position from $\alpha_1 = 45^0$ to 90^0 will change the initial row of the projection matrix from row 1 to row 10, while the order of the rows is still the same.

IV.8. Backprojection

Since the research used simulated objects, A , with known pixels value, then a first simulated projection data matrix, P_0 , can be solved by projecting A with projection matrix, W , as:

$$P_{0(M \times 1)} = W_{(M \times N)} \cdot A_{(N \times 1)}$$

Where:

$P_{0(M \times 1)}$ = simulated projection data matrix

$W_{(M \times N)}$ = projection matrix

$A_{(N \times 1)}$ = simulated object matrix

Then using this first simulated projection data matrix, P_0 , a first resulting image matrix, Ar_0 , can be achieved by an initial guess:

$$Ar_{0(N \times 1)} = W^T_{(N \times M)} \cdot P_{(M \times 1)}$$

Where :

$Ar_{0(N \times 1)}$ = resulting first image matrix

$W^T_{(M \times N)}$ = transpose of projection matrix W .

IV.8.1. Iterative Filtering Process

From the theory it has been mentioned that a backprojection process will cause a $1/r$ blurring which in the image is represented by the difference between each resulting image pixel value and its origin. An iterative filter function then can be started by determining an initial guess of the resulting image value, Ar_0 , using the W -transpose (W^T) replacing the direct W^{-1} . This very blurred initial guess is then filtered (subtracted) by adjusted convergence speed parameter, λ , multiplied by difference between first simulated projection data, P_0 , and second simulated data, P_1 , resulting a first filtered image value, Ar_1 , which is then filtered again by the same process resulting a second filtered image value, Ar_2 . This iterative filtering procedure is then repeated again in a sufficient number of iterations to achieve an optimal resulting image, Ar . The step to step filtering process can be described as follows:

1. Initial guess:

$$Ar_0 = W^T \cdot P_0$$

2. First simulated projection data (P_1) will be:

$$P_1 = W \cdot Ar_0$$

Blurring :

$$\begin{aligned} f(b) &= Ar_0 - Ar_1 \\ &= \lambda \cdot (P_0 - P_1) \end{aligned}$$

So : $Ar_1 = Ar_0 - \lambda \cdot (P_0 - P_1)$

4. Second simulated projection data (P_2) will be:

$$P_2 = W \cdot Ar_1$$

Blurring :

$$\begin{aligned} f(b) &= Ar_1 - Ar_2 \\ &= \lambda \cdot (P_1 - P_2) \end{aligned}$$

So : $Ar_2 = Ar_1 - \lambda \cdot (P_1 - P_2)$

And for ($j+1$) iterations, a general filtering operations can be written as:

$$Ar_{j+1} = Ar_j - \lambda \cdot (W \cdot Ar_j - P_{j+1})$$

And as mentioned in the theory, for M (projections) $\neq N$ (pixels), above equation can not be solved, therefore according to Van Cittert, the ΔP should be 'reblurred' by W^T (W -transpose) to equalize the dimension with ΔA , while the λ value can be readjusted. And the resulting iterative filtering function will be:

$$Ar_{j+1} = Ar_j - \lambda \cdot W^T \cdot (W \cdot Ar_j - P_{j+1})$$

Where λ is a new adjusted convergence speed of the iterations. The number of iterations, j , is tightly influenced by the chosen λ value. Accurate choosing of λ value will make Ar value converging fast and stable with not too many iterations needed. In this research λ will be simulated from 0.5 to 50 to know its influence to image quality and number of iterations needed.

IV.9. Otomatization: Matlab Programming

Using the definitions resume of reconstruction algorithm, a MATLAB computer program for CT image reconstruction can be created. The MATLAB version used in the research is MATLAB 6.0. The program consists of three (3) sub-programs which can be used independently. They are:

1. Projection function program which output is a projection matrix ($W_{(M \times N)}$).

2. Simulated projection data ($P_{(M \times 1)}$) program, replacing the practical measured data. $P_{(M \times 1)}$ is simulated from a matrix operation of:

$$P_{(M \times 1)} = W_{(M \times N)} \cdot A_{(N \times 1)}$$

where $A_{(N \times 1)}$ is the reshaped object matrix..

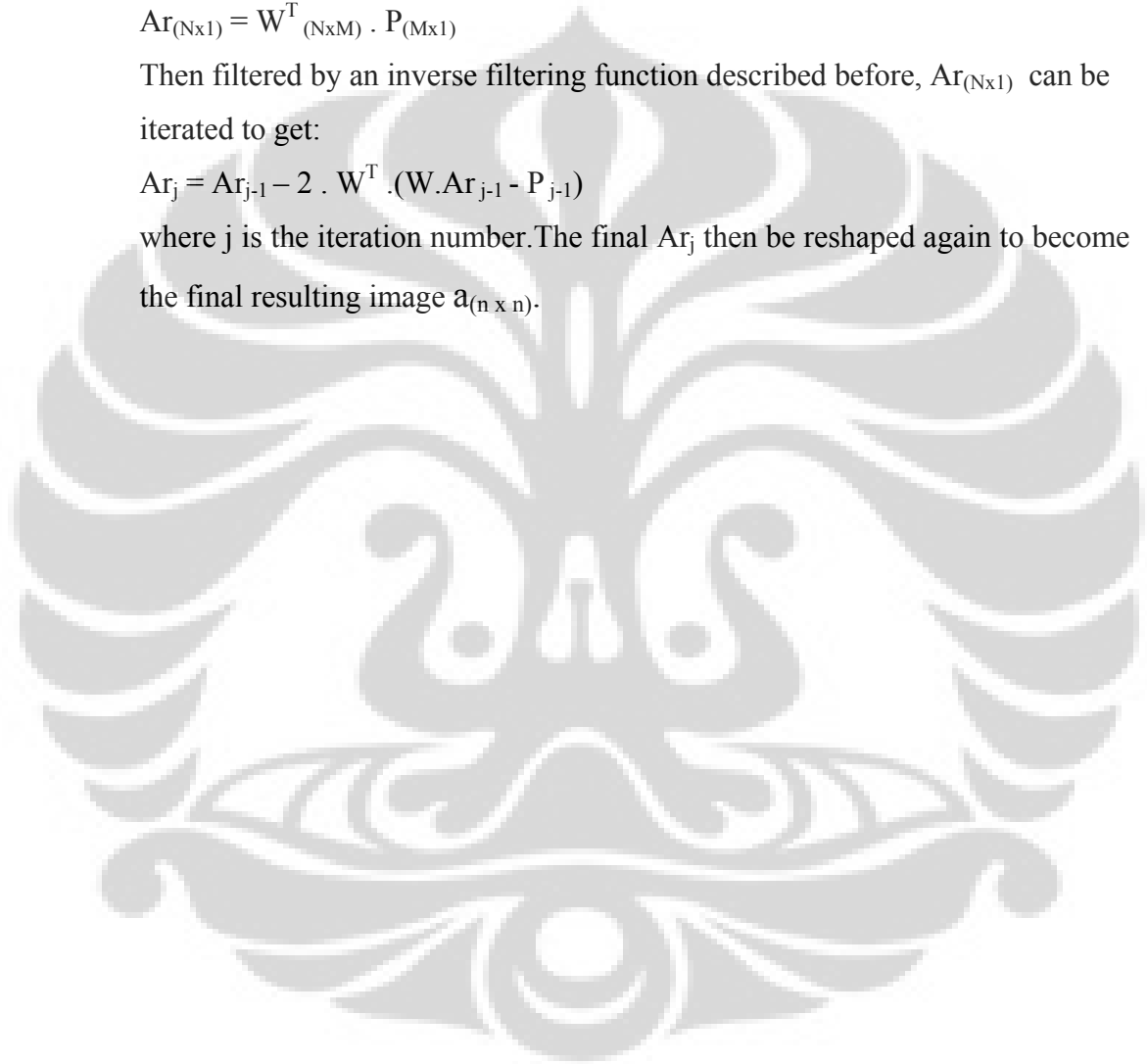
3. Backprojection program which output is a first resulting image ($Ar_{(N \times 1)}$). The algorithm is a matrix operation of:

$$Ar_{(N \times 1)} = W^T_{(N \times M)} \cdot P_{(M \times 1)}$$

Then filtered by an inverse filtering function described before, $Ar_{(N \times 1)}$ can be iterated to get:

$$Ar_j = Ar_{j-1} - 2 \cdot W^T \cdot (W \cdot Ar_{j-1} - P_{j-1})$$

where j is the iteration number. The final Ar_j then be reshaped again to become the final resulting image $a_{(n \times n)}$.



CHAPTER 5

RESULT

V.1. Projection Matrix (W)

Each simulation on a simulated object of dimension $N_{(n \times n)}$ conducted will need a projection matrix $W_{(M \times N)}$. Each projection matrix can be runned independently, so it can be used to project any object matrix of the same N value.

To give graphic value of a projection matrix elements, two examples of projection matrix plot of $W(256 \times 256)$ and $W(1024 \times 1024)$ are displayed below.

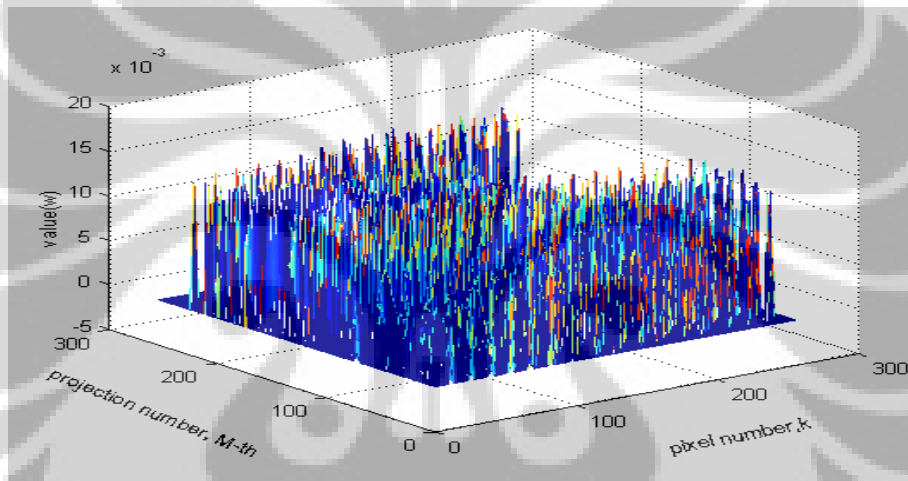


Figure 5.1.a. Plot of a projection matrix $W_{(M \times N)}$, $M=16 \times 16$, $N=16 \times 16$

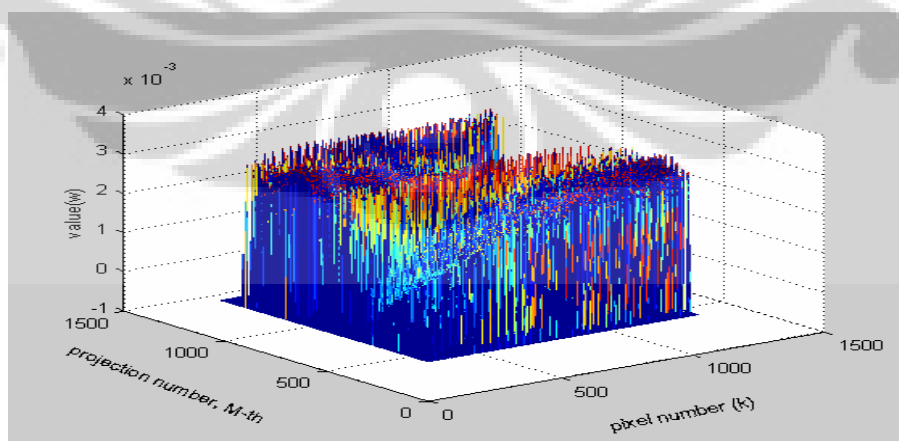


Figure 5.1.b. Plot of a projection matrix $W_{(M \times N)}$, $M=32 \times 32$, $N=32 \times 32$

V.2. Parameters Optimization

V.2.1. Beam Shape

By comparing the images resulted from a parralel beam of 20x20x400 and a fan beam of 16x16x256 projections, it is obvious that resulting image of the fan beam is much sharper and the existing boundary line inside the image is clear compared to blurred resulting image of the parallel beam projection.

By understanding the spatial-frequential domain concept mentioned in the literature, it's not difficult to make analysis of this result. Since each element value of $W_{(M^{th},k)}$ is the area of intersection between a m-projection ray and a k-numbered object pixel, the value is in mm^2 unit (spatial domain). Inversing W to W^{-1} will automatically change the domain to frequencial. So the value of image pixels resulted from $W^{-1}xP$, where P is a no-unit projection data value, will be also in frequencial domain.

A parallel beam will intersect bigger area of an object than a fan will. Bigger area will result in a lower frequency which in the turn will result in a smaller value of the image pixels which will cause darkened blurring in the image.

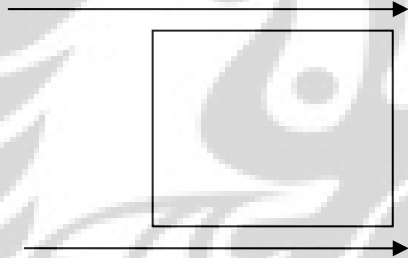


Fig.5.2a. A parallel ray projection

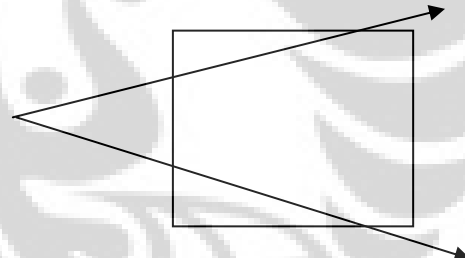


Fig.5.2b. A fan ray projection

If the analysis go further to a ray-pixel scale, it can be predicted that sharper projection fan ray will intersect smaller area of an object pixel which will make higher frequency of W^{-1} . If enough number of sharp rays are projected onto the pixel, then the pisel is defined enough to be reconstructed to be a clear and sharp resulting image pixel. But then there are two problems existing of the above analysis. Both problems are of how sharp the fan ray should be and what minimal number of fan rays will be needed to well define a pixel. Next simulation result is an alternative answer of those two problems.

V.2.2. Angular Width

From the reference it is recommended that the width of a beam projecting an object is somewhat similar with the width of the object.. The recommendation is easy to be applied for a parrallel beam projection. But to apply it in the fan beam projection is rather confusing since fan beam is diverging, so the projection width must be an angular value. The angular width will need a reference part of the object which width will be a reference width. And since a beam consists of rays and an object matrix consists of pixels, the angular width of beam based on object matrix width can be compared with the width of ray based on pixel width.

V.2.2.1. Beam Angular Width

If a beam direction is taken along X-axis, as seen in figure 5.3, there are three possible part of matrix that can become a reference width, which are:

1. Near side boundary
2. Middle line of the matrix
3. Far side boundary

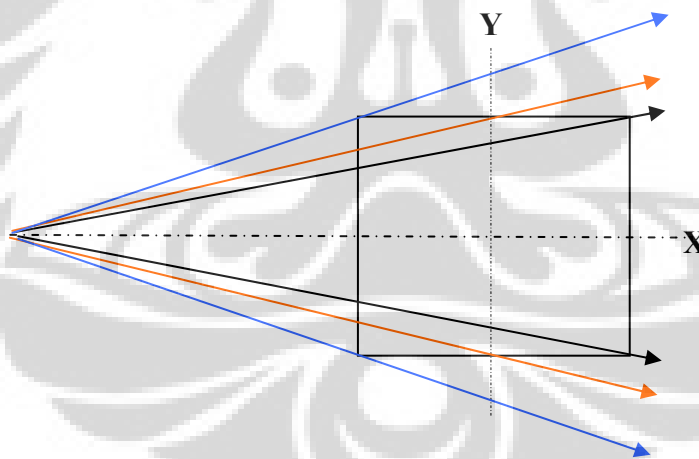


Figure 5.3. Three possible reference parts width of a matrix

From the beam shape simulation it is known that sharper beam means higher frequency and it will result in a more optimal resulting image. For that reason it is decided to conduct simulations only on highest frequency projection which is the beam number 3, which reference angular width is the width of far side boundary of the matrix.

From both resulting images of $N=16 \times 16$ and $N=32 \times 32$, it is obviously seen that both images are very blurred and can not be applied in practical used. Especially for $N=32 \times 32$, the circle shape boundary are not very clear while the rectangular shape boundary at both edge of the matrix is already disappear. Therefore it can be concluded that this beam angular width based on object matrix is not recommended.

V.2.2.2. Ray Angular Width

If a beam direction is taken along X-axis, as the object matrix is put on the center of the CT system and the number of detectors is even, one line of each central rays of the beam as seen in figure 5. will go straight along the X-axis line. Unlike the beam-matrix case, there are more choices of pixels which width can become a reference width of the ray. And since rays width will be accumulated to be a beam width, the highest frequency rule can not be applied here.

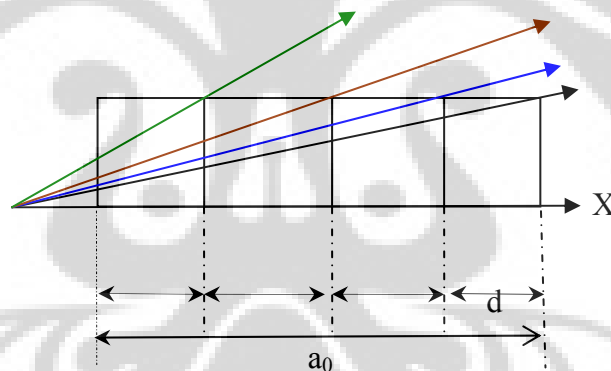


Figure 5.4. Possible reference pixels which positions along X-axis is $a_0 - fd$, where f is a changing value

Due to efficiency priority, number of projection rays in a beam is limited. Therefore too sharp rays will cause a too sharp beam which then will reduce the field of view of CT system and there is a possibility that some object area will not be defined enough. By observing resulting image of the changing f ($= 1, 2, 3, 4$ and 5) which can be seen in appendices, it is clear that among the five resulting images, there are two images that are obviously sharp. Both images are of :

1. $f = 1$

2. $f = 4$

If an analysis is taken from the highest frequency rule, it is rather confusing that image of $f = 0$ (most far side pixel boundary) is more blurred than the two images of less frequency ($f = 1$ and 4). As mentioned in the previous result, the answer of this phenomena is a possibility of less defined area of the object matrix. Using the most far side pixel boundary mean the ray is becoming too sharp and accumulating also into a too sharp beam. And due to limited number of views, this will lead to the existing less defined area of object matrix that will also cause blurring.

Back to the two optimal image, it can be seen further that the central blurred pattern of image of $f = 4$ is even more systematic than the central blurred pattern of image of $f = 1$. Since for object matrix of $N=16 \times 16$, the pixel of $f=4$ is located at the central half of the matrix and boundary of the circle shape figure existing in the image also located at this position, then by this coincidence an analysis can be made.

Due to the same boundary of the circle and the pixel of $f=4$, the inner side of the ray is smaller than the pixel of $f=1$, meaning higher frequency area in the circle and lower frequency outside the circle. By a combination between enough number of projection and rather high frequency, the blurred at circle inner side of the image of $f=4$ is more systematic than it is of $f=1$.

How ever it is difficult in practical use to know where the boundary of an important target organ will be located in a CT object matrix. For example, in a human head scanning it is almost impossible to know where exactly the target brain boundary will be in an imaginary object matrix of a CT system. So in this angular width simulations, it will be safe to recommend the pixel of $f=1$ as a reference width of the ray angular width.

V.2.3. Reconstruction Time

The simulation of the reconstruction time is purposed to know the influence of both doubling object matrix dimensions and number of projections to the duration of reconstruction process. Both the resulting reconstruction time data and the resulting image of M projections onto N pixels can be seen in the

appendices. Here, to be easily analysed, the time data are transformed to backprojection speed value, since in an iterative reconstruction technique, the duration of backprojection process will take the major part of all reconstruction time.

Table 5.1. Backprojection speed (iterations/seconds)

M x N (projections x pixels)	10.000 / \bar{t} (iterations/seconds)
256 x 256	222,22
1024 x 256	83,33
1024 x 1.024	15.15
4096 x 1024	2,32
4096 x 4096	0,56

And from table 5.1, a simple chart can be made.

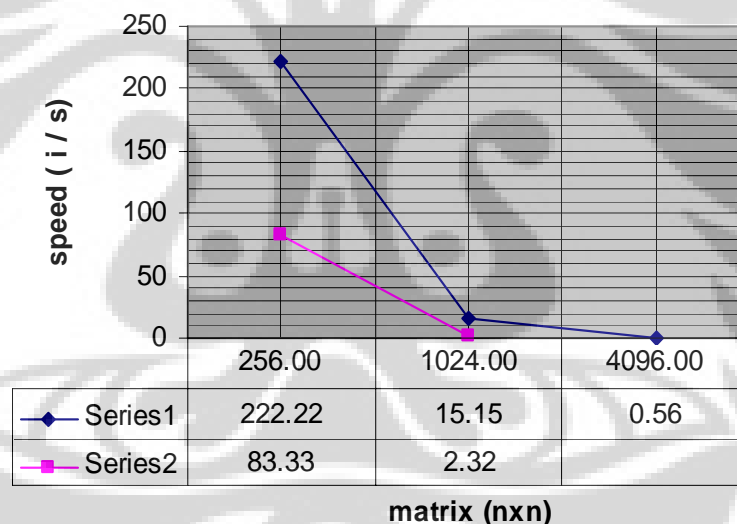


Figure 5.5 A chart of backprojection speed vs number of pixels (N). The blue line is for M (projections) = N while the red line is for M=2N.

From the chart, a brief analysis can be made. Since doubling an object matrix dimension from $n \times n$ to $2n \times 2n$ will automatically need the doubling of M projections dimension from $m \times m$ to $2m \times 2m$, due to $M \geq N$ requirement, the number of both matrix elements involved in an iteration process will be raised to 4 x 4 (= 16) times from the original number. And if the M projections need to be doubled again from $2m \times 2m$ to $4m \times 4m$ to get a better resulting image, total matrix elements now will be raised again to 4 x 4 x 4 (= 64) times from the origin.

This will explain why the chart has a logarithmic trend. Of course with more data for other dimension expansions, more accurate chart will be gained. However the above chart can give a prediction of how long a reconstruction of a doubled dimension object matrix will take.

Beside the time conclusion, from the resulting images another conclusion also can be made. By doubling again the projection number, consequently the reconstruction process will be much slower but more spatial and contrast resolution will be achieved resulting a much sharper image. So designing a CT projection plan will include a decision of choosing the priority between better resulting image or shorter reconstruction time which each will have its own benefit and drawback. Conducting these parameters optimization simulations will give better combination of designing parameters value of a CT projection plan.

V.2.4. Number of Iterations (j) and Filtering Process

This simulation is conducted to know the process of image filtering and the maximal number of iterations needed to give major contribution to the image quality. From the $j = 0$ iteration, it is proved that it's not possible to conduct a reconstruction without filtering process. Since the research use an iterative technique, a step by step filtering is needed.

From the plot of resulting pixel value, it can be observed that significant result of the filtering process starts at $j=1000$ and raise again to maximal value at $j = 10.000$. More iterations then will not make significant result to the image. As it can be seen for $j = 20.000$, the image quality has no significant difference with the image quality of $j = 10.000$.

So from above parameters simulations, it is recommended that optimal combinations of parameters value are:

1. To choose detector size which adjust the ray angular width to $f=1$,
2. To double both number of detectors and views to get: $M = 4N$, and
3. To conduct filtering process up to 10.000 iterations.

V.3. Experiment

For an obvious clear or blurred image, it is easy to review the result qualitatively, but for almost similar image quality, the effective way to review is by using a quantitative standard. The quantitative tool to review an image quality is the Mean Squared Error (MSE)/pixel. However, since the value only represents mean error of an image matrix, it can not tell the details of an image, and since a CT image is also meant to be used by a non-technical person, a qualitative review by naked human eye is still recommended.

V.3.1. Qualitative

V.3.1.1. Object Matrix of $N=32 \times 32$ and $M=64 \times 64$ projections

- **Changing Parameter: Number of Iterations (j) for $\lambda = 2$**

From the experiment results of $N=32 \times 32$, it can be seen that for $\lambda = 2$, the converging process was rather slow and it was still very blurred at 100 iterations. The image got much better at 1000 iterations and needed 10.000 iterations to gain an optimal resulting image. It can also be observed that the spatial resolution of the image was very good since the characteristic shape of Shepp-Logan phantom was still matched with the original shape.

By careful observation, it can be seen that a minor blurring still existed, even at 10.000 iterations. The minor blurring located at two positions. The first blurring was inside the brain area and the second blurring was outside the head boundary. Related to the small tumours of the head phantom, the tumours shape was clear but the contrast was almost the same with their surrounding tissue

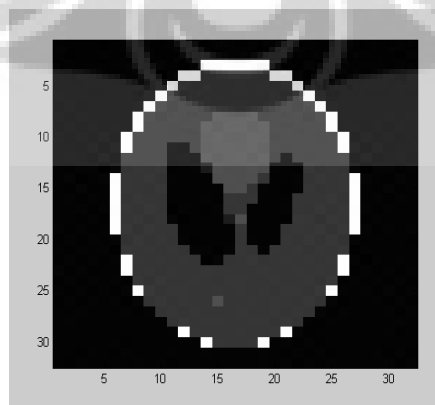


Figure 5.6. $j = 10.000$ iterations, $\lambda = 2$

How ever from the general point of view, it can be concluded that the resulting image for $\lambda = 2$ was clear enough to be recommended.

- **Changing Parameter: Lambda (λ) for $j = 10.000$ iterations**

From the experiment results, it is obvious that the image of $\lambda = 1$ was blurred while the head shape was still very good. The inner blurring got missing along with the raising value of λ , but the outer blurring still existed and even mostly expanded at $\lambda = 5$. Raising again the λ to 10 made both inner and outer blurring totally disappear and the contrast resolution got much better. Finally at $\lambda = 15$, the contrast of the image was almost perfect and somewhat similar with the original phantom.

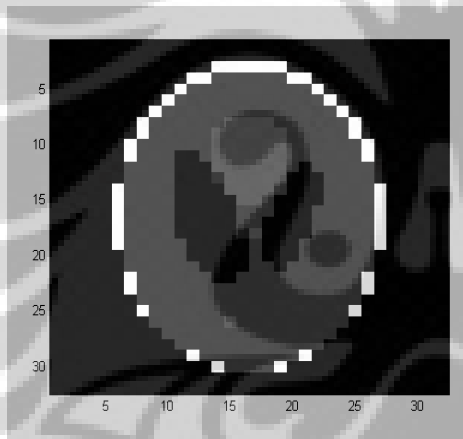


Figure 5.7.a. $\lambda = 5, j = 10.000$ iter.

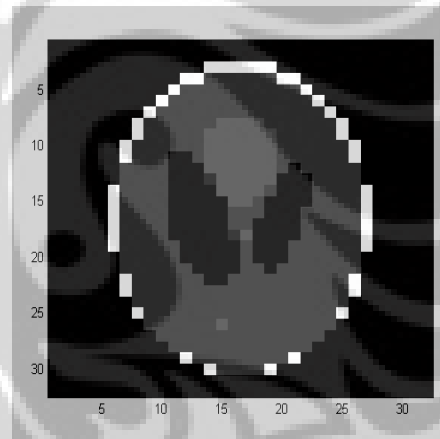


Figure 5.7.b. $\lambda = 15, j = 10.000$ iter.

So it can be concluded that minimal recommended convergence speed for this experiment will be at $\lambda = 15$.

V.3.1.2. Object Matrix of $N=64 \times 64$ and $M=128 \times 64$ projections

- **Changing Parameter: Number of Iterations (j) for lambda (λ) = 2**

From the experiment results it can be observed that the spatial and contrast resolution of the image was very poor until 1.000 iterations. At 10.000 iterations the spatial resolutions was even worse, due to broken lower and upper boundary

of the head shape, but the contrast resolution was much better, due to clear contrast difference between organs and background tissue. The small tumours shape were also clearly seen.

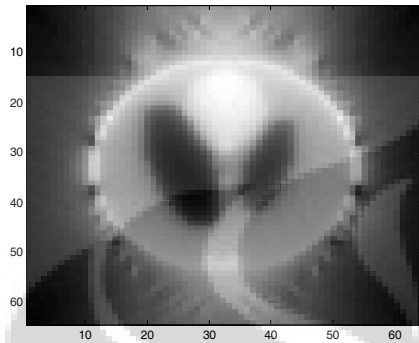


Figure 5. 8.a. $\lambda = 2, j = 1.000$ iterations

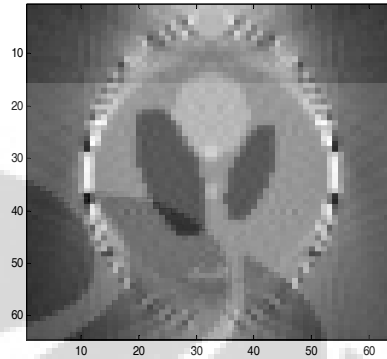


Figure 5. 8.b. $\lambda = 2, j = 10.000$ iterations

The poor spatial resolution was predictable since there were not enough samplings for the phantom. Compared to object of $N=32 \times 32$ with $M=64 \times 64$ projections/samplings, which means $M=4N$, this object of $N=64 \times 64$ was only projected by 128×64 projections/samplings, which means $M=2N$. So the phantom was not sufficiently defined. The above analysis become obvious at 10.000 iterations. Instead of becoming better, the vertical head shape boundary was getting broken, while the horizontal boundary was not. Back to optimizing parameters simulation analysis, it can also be explained that since the head phantom was elliptical (the horizontal boundary was closer to the center than the vertical), the ray width of $f=1$ could only maximally define the center part of the object matrix. The broken boundary showed that some area of vertical head boundary which located near the matrix edge was poorly projected. The above analysis can also explain why the center organs of the phantom got better contrast resolution at $j=10.000$. It also proves the reference theory, that an object image reconstructed by a fan beam projection will have better quality at center area.

- **Changing Parameter: Lambda (λ) for $j=100$ iterations**

From the experiment results it is obvious that the spatial and contrast resolution change due to the raising λ was almost similar with the iterations

experiment above. For higher value of λ , it can be seen that spatial resolution was getting worse but the contrast resolution was getting better. The analysis then will be the same with previous experiments. The concern here is that even the number of iterations only 100, for $\lambda = 40$ the tumours shape was even better than it was for $\lambda = 2$ at 1000 iterations.

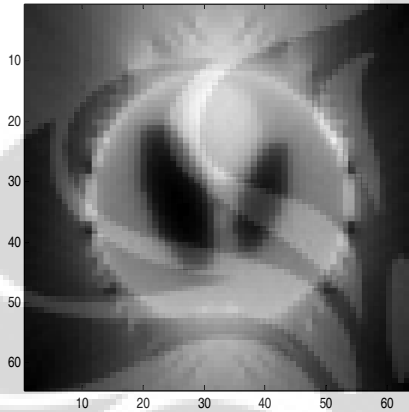


Figure 5.9.a. $\lambda = 2, j = 1000$ iter.

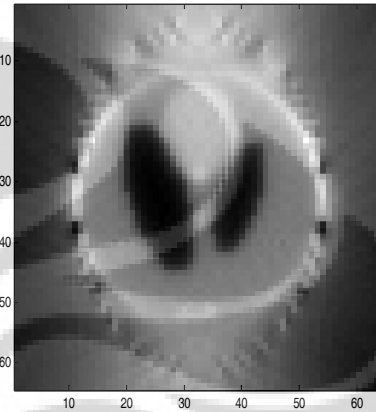


Figure 5.9.b. $\lambda = 40, j = 100$ iter.

From above result, it can be concluded that for object matrix of $N = 64 \times 64$, the convergence speed of $\lambda = 2$, is still too small and it is better to adjust $\lambda = 40$. By this result, a further experiment was then conducted to know the result if value of λ was raised again to 50 and the number of iterations was maximalized to 10.000. Below is the resulting image of the experiment:

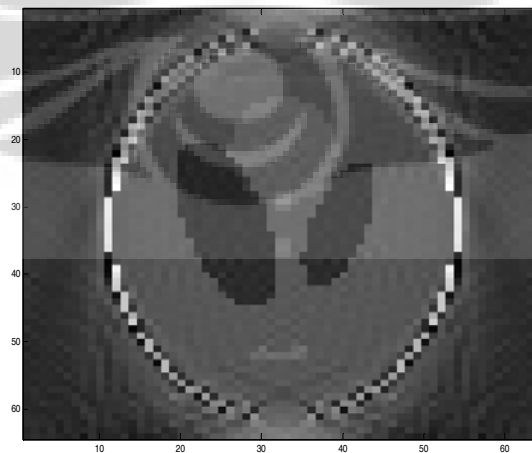


Figure 5.10. $\lambda = 50, j = 10.000$ iterations

The result was predictable. For $\lambda = 50$ and $j=10.1000$ iterations, the contrast resolution was very good and the head organs boundary was very clear, included the three small tumours in the head phantom. However it is also seen that the spatial resolution was still poor, especially at the vertical head boundary, even though the broken was obviously reduced. From this experiment result, it is then recommended to adjust convergence speed to $\lambda=50$ for object matrix of $N=64 \times 64$ and $M=128 \times 64$.

V.3.2. Quantitative: Mean Square Error (MSE)

To give quantitative explanations of the qualitative image data, then it will need a standard parameter to represent the quality of the images. Below is a complete data table of the Mean Square Error / pixel of each resulting images relative to the original phantom. By using the data table, four MSE graphics can be made to get better understanding of the influence of raising the convergence speed, λ , or the number of iterations to the image quality. Both the MSE table and graphics can prove the previous qualitative data analysis

From equation (2.21), the MSE is defined as:

$$\text{MSE/pixel} = (1 / N) \sum_{k=1}^N |\Delta a_k|^2 \quad (2-21)$$

where: MSE/pixel = Mean Square Error

N = Number of object pixels

k = pixels number = 1, 2, 3, ..., N

And from a simple computation (which can be worked by a simple Matlab program), table 5.1 is achieved as:

Table 5.1. Mean Square Error (MSE) / pixel

N=32X32, M=64X64		N=64X64, M=128X64	
LAMBDA = 2		LAMBDA = 2	
ITERATIONS	MSE	ITERATIONS	MSE
100	0.0306	10	0.058
500	0.0150	20	0.0554
1000	0.0086	30	0.0532
2000	0.0037	40	0.0513
3000	0.0020	50	0.0497
4000	0.0012	100	0.0444
5000	0.0008	1000	0.0356
10000	0.0002	10000	0.0286
ITERATION = 10.000		ITERATION = 100	
LAMBDA	MSE	LAMBDA	MSE
1	7.58E-04	2	0.0444
2	2.00E-04	5	0.0397
5	9.00E-06	10	0.0376
10	2.91E-07	20	0.0355
15	2.09E-08	30	0.0344
		40	0.0335

And from the above table, graphics of MSE vs iterations (j) and MSE vs lambda (l) can be made to get better overview of the quantitative result. The graphics can be seen below:

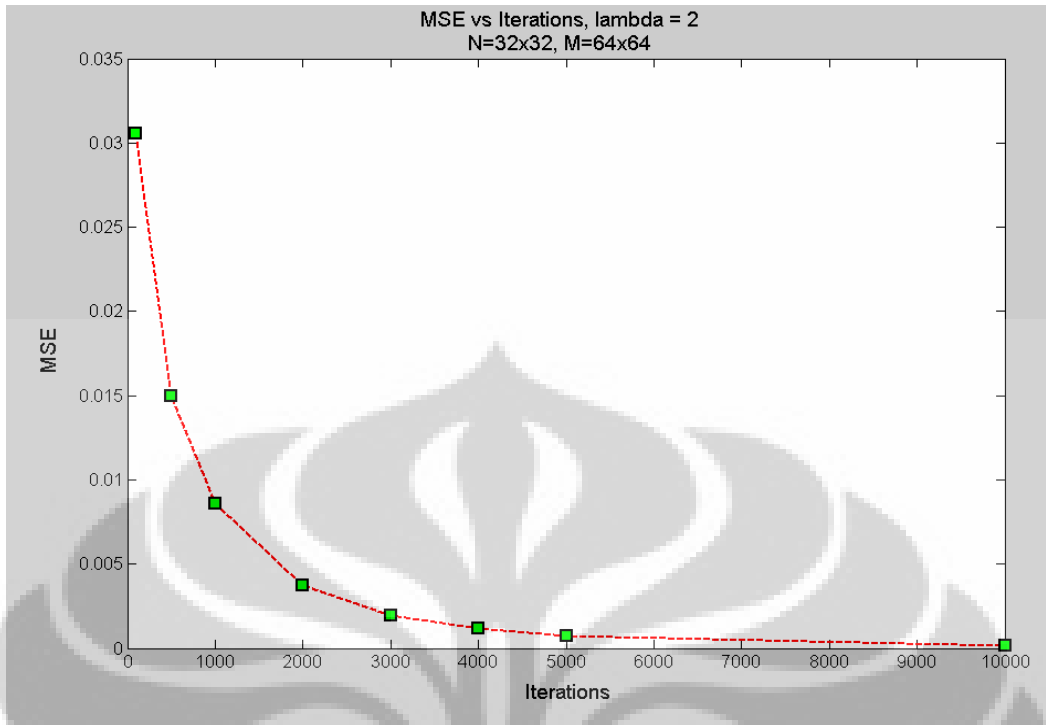


Figure 5.11.a. A Graphic of MSE vs Iterations for $\lambda = 2$.
Object matrix of $N=32 \times 32$, $M=64 \times 64$ projections

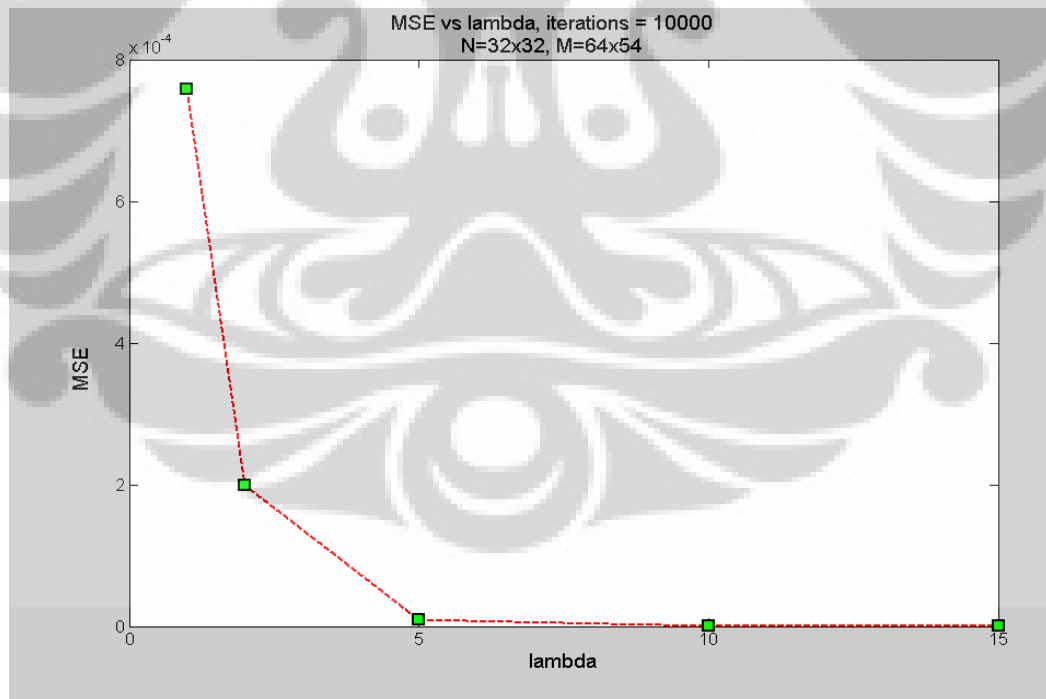


Figure 5.11.b. A Graphic of MSE vs λ for $j = 10,000$ Iterations,
Object matrix of $N=32 \times 32$, $M=64 \times 64$ projections

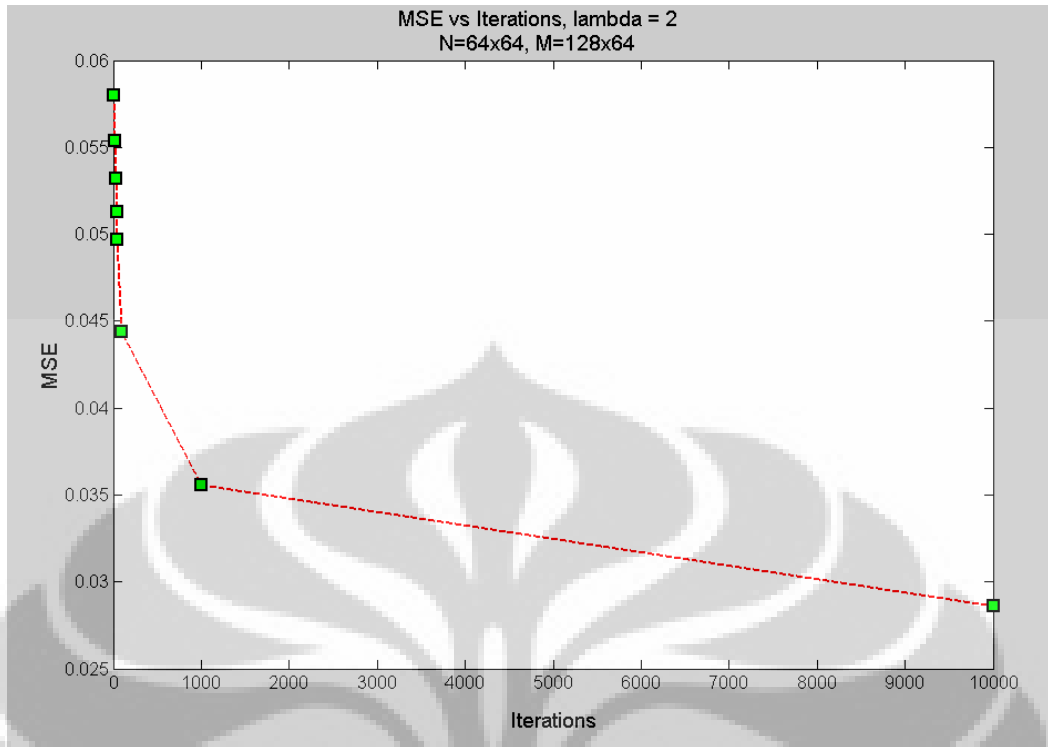


Figure 5.11.c. A Graphic of MSE vs Iterations, for $\lambda = 2$
Object matrix of N=64x64, M=128x64 projections

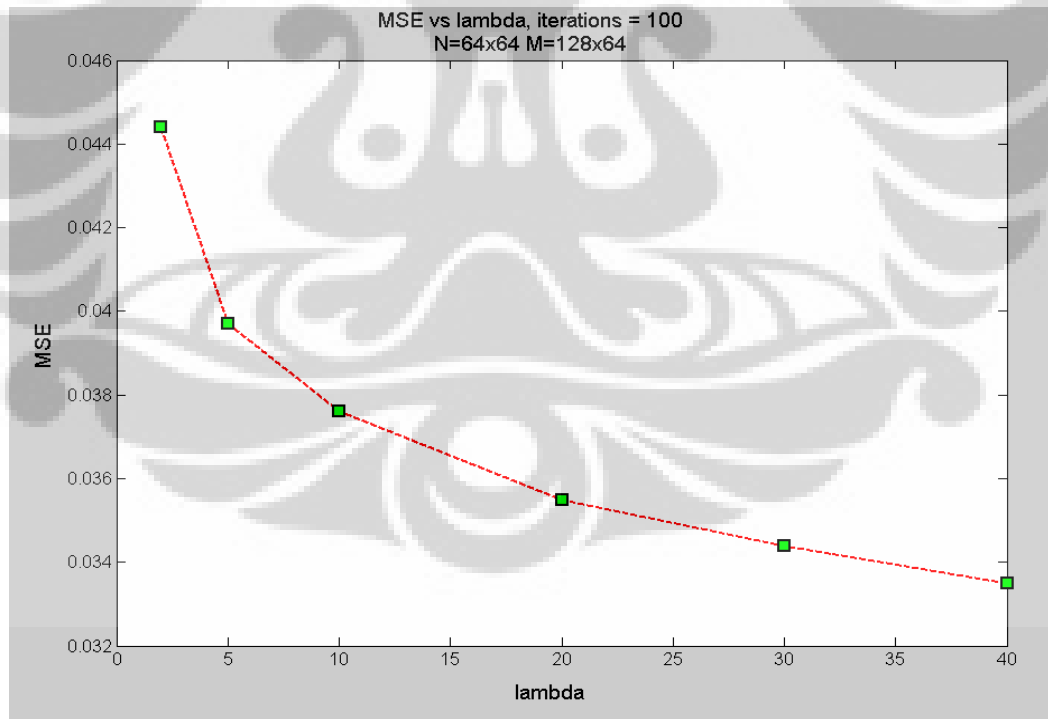


Figure 5.11.d. A Graphic of MSE vs λ for $j = 100$ Iterations,
Object matrix of N=64x64, M=128x64 projections

From above graphics, a general conclusion can be made that the decrease of MSE in MSE vs λ graphic is faster than the decrease of MSE in MSE vs iterations graphic. It means that adjusting an accurate convergence speed, λ , is more effective than conducting so many iterations.

V.3.3. Comparison of ART and FBP Technique

Resulting Images of the FBP technique along with filtering function and resulting images of the ART developed in the research.

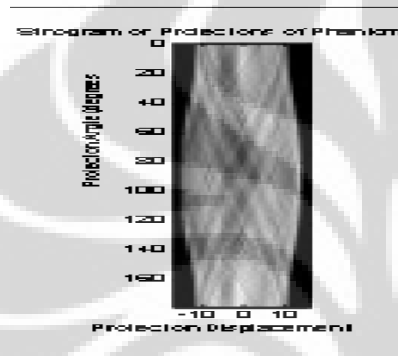


Figure 5.12.a. Unfiltered sinogram

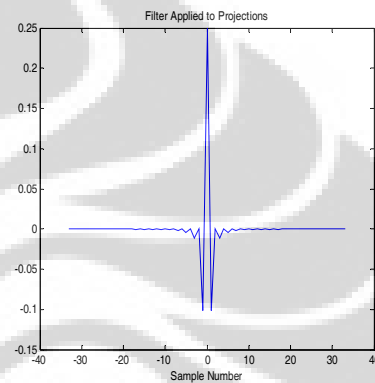


Figure 5.12.b. Ram-Lak filter

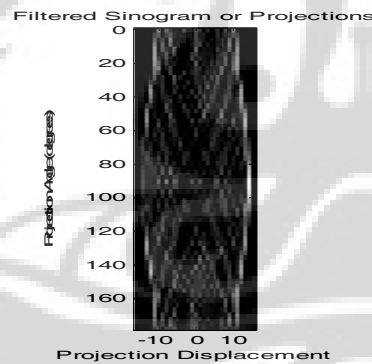


Figure 5.12.c. Filtered sinogram of (5.11.a.)

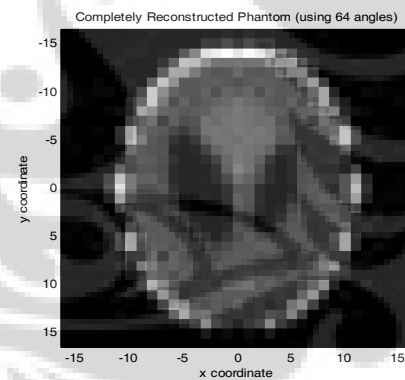


Figure 5.12.d. Resulting Image, $M=64 \times 32$, $MSE = 0.0115$

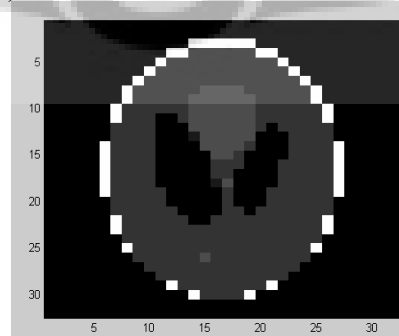


Figure 5.13. Resulting image of the ART developed in the research, $\lambda=15$ $j=10,000$ iterations, $MSE=2.09E-08$

It is obvious from the result that the image quality, both spatial and contrast resolutions, of the ART was much better than the quality image of the FBP. The analysis is not difficult, because the FBP here used a parallel beam projection and it much depended on the object matrix dimension, N , instead of number of views like in the ART.

And since in FBP, the filtering function is tightly related with the spatial resolution of the object matrix, and since parallel beam view is only limited in the range of 0^0 to 180^0 , then adding the view of the beam to an unlimited number is not effective any more to the image quality if the object matrix dimension is not upgraded. It seems that maximal effective number of views relative to the object dimension is $M=2N$. More than that relative number, the Ram-Lak filter is not effective anymore, because unlike ART which more projection views means more definitions to pixels for more accurate initial guess in backprojection process, in the FBP where filtering process is conducted in forward projection, if the definitions of pixels are already enough, more projection views only mean more task for the Ram-Lak filter.

Good example of the above analysis can be seen in the figure below, where the projection views of object matrix of $N=32 \times 32$ is even upgraded from $M=64 \times 32$ to $M=128 \times 32$. The result is similarly blurred with the previous but from the MSE value, it is known that the image quality is even decreased.

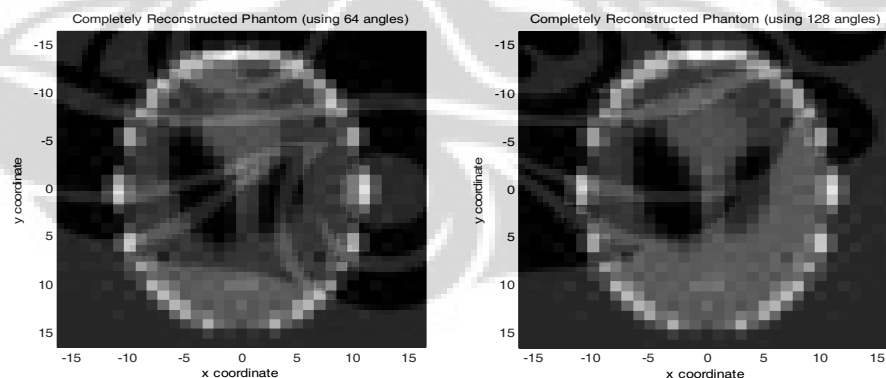


Figure 5.14.a. Resulting Image, $M=64 \times 32$, MSE = 0.0115

Figure 5.14.b. Resulting Image, $M=128 \times 32$, MSE = 0.0116

V.3.4. Comparison of ART and Another Iterative Technique

The comparing images are taken from the result of a mastering degree research conducted by Abbey Mukkanancherry [10] using different iterative

reconstruction technique with different initial guess method. The iterative technique used by Mukkanancherry also used convergence speed, λ . But since the initial guess method was different, the convergence speed is adjusted at $\lambda = 0,5$.

Since all resulting images are reconstructed by a fan beam, it is obvious that both compared methods have the same purpose to achieve most effective value of convergence speed, λ , and its influence to the number of iterations needed. Below is the figure of resulting images worked by Mukkanancherry compared with resulting images of the research. The figure is taken from electronic copy of the thesis publication.

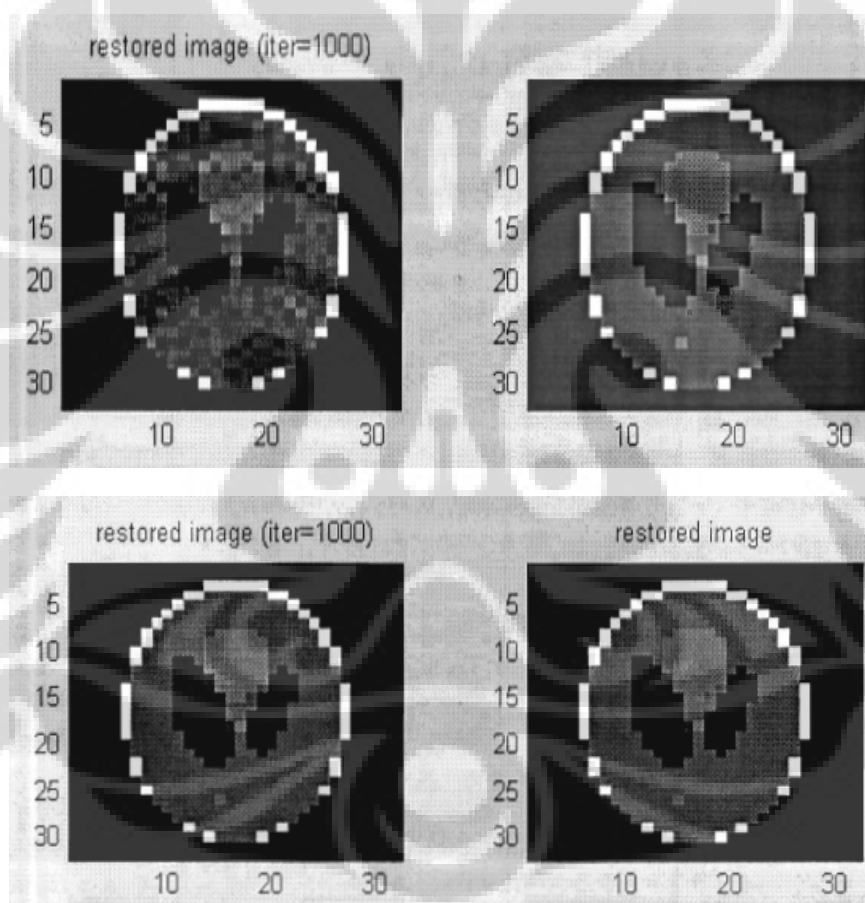


Figure 5.15.a. (top) $M=32 \times 32$, $N=32 \times 32$, fan beam, parallel source-detector geometry, $j = 1.000$ iterations (left) and $j = 10.000$ iterations (right). (from [10])

Figure 5.15.b. (bottom) $M=128 \times 128$, $N=32 \times 32$, fan beam, clamshell source-detector geometry, $j = 1.000$ iterations(left) and $j = 40.000$ iterations (right). (from [10])

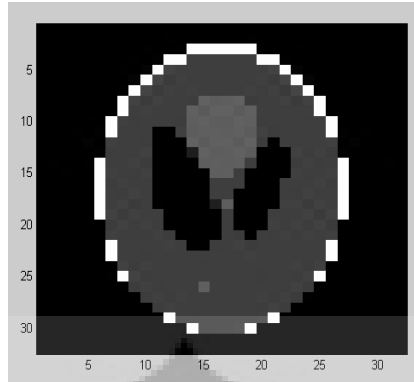


Figure 5.16. Resulting image of the research
 $\lambda = 15$ $j = 10.000$ iterations.

First resulting image of the comparing method in figure 5.13a. are very blurred due to lack of filtering process while the second is much better after the filtering process were added. And from figure.5.13b, where the source-detector geometry is called clamshell, the number of projections M is 16 times of the number of object pixels. The changing parameter here is the number of iterations. Mukkananchery here wanted to show that by using the geometry, it needs only 1.000 iterations to gain a sharp resulting image, compared to 10.000 iterations needed by the previous (parallel geometry). While maximal sharp of the image then was reached by 10.000 iterations as shown in the second image.

Finally from the fig.5.13c. it is obvious that the resulting image of the ART developed in this research is proved to be the clearest image.

CHAPTER VI

SUMMARY

From the result analysis of both the parameter optimization and the experiment, general conclusions can be taken, which are:

1. The MATLAB program based on Algebraic Reconstruction Technique (ART) algorithm developed in the research has been able to be used as a simple CT image reconstruction program.
2. A fan beam is much more effective to be used as a projection beam than the parallel beam.
3. To get maximal performance of the ART algorithm developed in the research, the recommended optimized parameters of the research are:
 - a. To choose the ray angular width of $f=1$,
 - b. To double both number of detectors and views to get: $M = 4N$, and
 - c. To conduct filtering process up to 10.000 iterations.
4. Since the research used the ART which is an iterative technique, it is concluded that:
 - a. For object matrix of $N=32 \times 32$ and $M=64 \times 64$ samplings then a minimal effective value of convergence speed is $\lambda = 15$
 - b. For object matrix of $N=64 \times 64$ and $M=128 \times 64$ samplings then a minimal effective value of convergence speed is $\lambda = 50$
 - c. Adjusting the value of λ is not easy since taking too small value will make the convergence too slow but taking too big value will make the convergence not stable, which can result wide fluctuation of resulting image quality.
 - d. Adjusting an accurate λ is much more effective than conducting too many iterations to get an optimal resulting image.
5. The ART developed in the research has proved to be an effective reconstruction method for limited dimensions of object matrix. Compared to FBP technique and another iterative technique the quality of its resulting images were even better.

6. The main drawback of the ART is the limited projections/samplings, M , can be conducted due to its massive matrix operations in iterative process. To optimize the projection of an object matrix of $N= n \times n$, the computer must be able to process projections/samplings of $M= 2n \times 2n$.
7. A maximal effort should be done to find a mathematical solution for much shorter iterative process.



REFERENCES

Books and Manuals

1. Z.H. Cho, M.Singh, J.P.Jones, Foundations of Medical Imaging, John Wiley & Sons (1993), p:21-103; 148-162.
2. J.T Buhberg, J.A Seibert, E.M. Leidholdt, Jr., J.M. Boone, The Essential Physics of Imaging, 2nd Ed., Lippincott William and Wilkins (2002), p: 327-372.
3. Brown, R.H. Smallwind, P.C. Barker, P.V. Lawford and D.R Hose, Medical Physics and Biomedical Imaging, Institute of Physics Publishing (1999), p: 382-386; 315-339.
4. H.E Johns, O.C., J.R. Cunningham, The Physics of Radiology, Thomas Publisher, Springfield, 4th Ed., (1983), p: 557-664.
5. R.K. Hobbie, Intermediate Physics for Medical and Biology, 3rd Ed. Springer Verlag, New York Inc. (1997).
6. D. Hanselman, B. Littlefield, Mastering Matlab 7, Pearson Prentice Hall (2005).
7. G. N. Hounsfield. A method of and apparatus for examination of a body by radiation such as X-ray or gamma radiation. British Patent No. 1283915. London (1972).
8. Thesis Manual for Graduate School, Colorado State University (2003 revised).

Papers/Publications/Matlab Work

9. Kak A.C., Slaney M., Principles of Computerized Tomographic Imaging, (1999) http://cobweb.ecn.purdue.edu/~malcolm/pct/CTI_intro.pdf
10. A. Mukkananchery, Iterative Methods for The Reconstruction of Tomographic Images with Unconventional Source-Detector Configuration, Master Thesis, Virginia Commonwealth University, Richmond Virginia (2005), http://etd.vcu.edu/theses/available/etd_10262005131008_unrestricted/mukkanancherya_thesis.pdf

11. V.H.Tessa, W. Sarah, etc, The Implementation of Iterative Reconstruction Algorithms in MATLAB, Master Thesis, Department of Industrial Sciences and Technology, University College of Antwerp, Belgium (2007)
12. Warsito, M. Ohkawa, N. Kawata, S. Uchida, Cross Sectional Distributions of Gas and Solid Holdup in Slurry Bubble Column Investigated by Ultrasonic Computed Tomography, Chemical Engineering Science 54 (1999), p:4711-4728.
13. W. Warsito, L.S. Fan, Neural Network Multi-Criteria Optimizing Image Reconstruction Technique (NN-MOIRT) for Linear and Non-linear Process Tomography, Chemical Engineering and Processing 42 (2003), p:663-674, www.elsevier.com/locate/cep
14. M. B. Utomo, W. Warsito, T. Sakai, S. Uchida, Analysis of Distribution of Gas and TiO₂ particles in Slurry Bubble Column Using Ultrasonic Computed Tomography, Chemical Engineering Science 56 (2001), p:6073-6079.
15. Dr. Warsito, Matlab Work for Parallel Beam Projection of 20x20x400 (2002)
16. B.S. Gede, Focusing Computed Tomography, Ph.D.Thesis, Monash University (1999)
17. R. H. Magnus, S.Edward, Methods of Conjugate Gradients for Solving Linear Systems, Journal of Research of the National Bureau of Standards, Vol.49, No.6 (1952).
18. G. Dan, Parallel ART for Image Reconstruction in CT using Processor Arrays, The International Journal of Parallel, Emergent and Distributed Systems, Vol.21, No.5 (2006), p: 365-380.
19. G.H. Chena, A new framework of image reconstruction from fan beam projections), Department of Medical Physics, University of Wisconsin, Madison, Wisconsin 53792 (2003)
20. ART/SART/SIRT, http://cobweb.ecn.purdue.edu/~malcolm/pct/CTI_Ch07.pdf
21. CTSim - The Open Source Computed Tomography Simulator, www.ctsim.org
22. Computed Tomography, http://en.wikipedia.org/wiki/Computed_tomography

APPENDICES

A1. Table of Intersection Points Position

CATHEGORY 1: $0 < \Phi < (\pi/2 - \Delta\beta)$

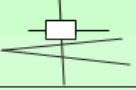



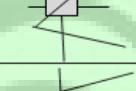




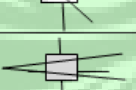
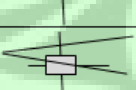
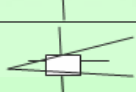
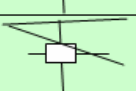
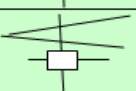

No	X _e	X _f	X _a	X _b
1	Xe1	Xf1	Xa1	Xb1
2	Xe1	Xf1	Xa1	Xb2
3	Xe1	Xf1	Xa1	Xb3
4	Xe1	Xf1	Xa2	Xb1
5	Xe1	Xf1	Xa2	Xb2
6	Xe1	Xf1	Xa2	Xb3
7	Xe1	Xf1	Xa3	Xb1
8	Xe1	Xf1	Xa3	Xb2
9	Xe1	Xf1	Xa3	Xb3
10	Xe1	Xf2	Xa1	Xb1
11	Xe1	Xf2	Xa1	Xb2
12	Xe1	Xf2	Xa1	Xb3
13	Xe1	Xf2	Xa2	Xb1
14	Xe1	Xf2	Xa2	Xb2
15	Xe1	Xf2	Xa2	Xb3
16	Xe1	Xf2	Xa3	Xb1
17	Xe1	Xf2	Xa3	Xb2
18	Xe1	Xf2	Xa3	Xb3
19	Xe1	Xf3	Xa1	Xb1
20	Xe1	Xf3	Xa1	Xb2
21	Xe1	Xf3	Xa1	Xb3
22	Xe1	Xf3	Xa2	Xb1
23	Xe1	Xf3	Xa2	Xb2
24	Xe1	Xf3	Xa2	Xb3
25	Xe1	Xf3	Xa3	Xb1
26	Xe1	Xf3	Xa3	Xb2
27	Xe1	Xf3	Xa3	Xb3
28	Xe2	Xf1	Xa1	Xb1
29	Xe2	Xf1	Xa1	Xb2
30	Xe2	Xf1	Xa1	Xb3
31	Xe2	Xf1	Xa2	Xb1
32	Xe2	Xf1	Xa2	Xb2
33	Xe2	Xf1	Xa2	Xb3
34	Xe2	Xf1	Xa3	Xb1
35	Xe2	Xf1	Xa3	Xb2
36	Xe2	Xf1	Xa3	Xb3
37	Xe2	Xf2	Xa1	Xb1
38	Xe2	Xf2	Xa1	Xb2
39	Xe2	Xf2	Xa1	Xb3
40	Xe2	Xf2	Xa2	Xb1
41	Xe2	Xf2	Xa2	Xb2
42	Xe2	Xf2	Xa2	Xb3
43	Xe2	Xf2	Xa3	Xb1
44	Xe2	Xf2	Xa3	Xb2
45	Xe2	Xf2	Xa3	Xb3
46	Xe2	Xf3	Xa1	Xb1
47	Xe2	Xf3	Xa1	Xb2
48	Xe2	Xf3	Xa1	Xb3
49	Xe2	Xf3	Xa2	Xb1
50	Xe2	Xf3	Xa2	Xb2
51	Xe2	Xf3	Xa2	Xb3
52	Xe2	Xf3	Xa3	Xb1
53	Xe2	Xf3	Xa3	Xb2
54	Xe2	Xf3	Xa3	Xb3
55	Xe3	Xf1	Xa1	Xb1
56	Xe3	Xf1	Xa1	Xb2
57	Xe3	Xf1	Xa1	Xb3
58	Xe3	Xf1	Xa2	Xb1
59	Xe3	Xf1	Xa2	Xb2
60	Xe3	Xf1	Xa2	Xb3
61	Xe3	Xf1	Xa3	Xb1
62	Xe3	Xf1	Xa3	Xb2
63	Xe3	Xf1	Xa3	Xb3
64	Xe3	Xf2	Xa1	Xb1
65	Xe3	Xf2	Xa1	Xb2
66	Xe3	Xf2	Xa1	Xb3
67	Xe3	Xf2	Xa2	Xb1
68	Xe3	Xf2	Xa2	Xb2
69	Xe3	Xf2	Xa2	Xb3
70	Xe3	Xf2	Xa3	Xb1
71	Xe3	Xf2	Xa3	Xb2
72	Xe3	Xf2	Xa3	Xb3
73	Xe3	Xf3	Xa1	Xb1
74	Xe3	Xf3	Xa1	Xb2
75	Xe3	Xf3	Xa1	Xb3
76	Xe3	Xf3	Xa2	Xb1
77	Xe3	Xf3	Xa2	Xb2
78	Xe3	Xf3	Xa2	Xb3
79	Xe3	Xf3	Xa3	Xb1
80	Xe3	Xf3	Xa3	Xb2
81	Xe3	Xf3	Xa3	Xb3

A2. Table of Intersection Shapes and Area

CATHEGORY 1 : $0 < \Phi < (\pi/2 - \Delta\beta)$

INTESECTION SHAPE					TRAPEZOID (Wu) =	TOP REST (Wa) =		BOTOM REST (Wb) =				
No.	COMBINATION $x_a \cap x_e \cap x_b \cap x_f$				Illustration	$x_{a1} \cap x_{e1} = x_{e1}$, $x_{b1} \cap x_{f1} = x_{f1}$, $x_{a3} \cap x_{e3} = x_{a3}$, $x_{b3} \cap x_{f3} = x_{b3}$,	x_{2u} $\int_{x_{1u}} (Y_2 - Y_1) dx$	x_{2a} $\int_{x_{1a}} (Y_2 - Y_{j+1}) dx$	x_{2b} $\int (Y_j - Y_1) dx$			
	x_a	x_e	x_b	x_f		RESUME	x_{1u}	x_{2u}	x_{1a}	x_{2a}	x_{1b}	x_{2b}
1	1	1	1	1		x_{f1}	-	-	-	-	-	-
2	1	1	1	2		$x_{e1} \cap x_{b1} \cap x_{f2}$	x_i	x_f	-	-	x_i	x_f
3	1	1	2	2		$x_{e1} \cap x_{b2} \cap x_{f2}$	x_i	x_f	x_i	x_b	x_i	x_f
4	1	1	1	3		$x_{e1} \cap x_{b1} \cap x_{f3}$	x_i	x_{i+1}	-	-	x_i	x_{i+1}
5	1	1	2	3		$x_{e1} \cap x_{b2} \cap x_{f3}$	x_i	x_{i+1}	x_i	x_b	x_i	x_{i+1}
6	1	1	3	3		$x_{e1} \cap x_{b3}$	x_i	x_{i+1}	x_i	x_{i+1}	x_i	x_{i+1}
7	1	2	1	2		$x_{a1} \cap x_{e2} \cap x_{b1} \cap x_{f2}$	x_i	x_f	-	-	x_e	x_f
8	1	2	2	2		$x_{a1} \cap x_{e2} \cap x_{b2} \cap x_{f2}$	x_i	x_f	x_i	x_b	x_e	x_f
9	2	2	2	2		$x_{a2} \cap x_{e2} \cap x_{b2} \cap x_{f2}$	x_a	x_f	x_a	x_b	x_e	x_f
10	1	2	1	3		$x_{a1} \cap x_{e2} \cap x_{b1} \cap x_{f3}$	x_i	x_{i+1}	-	-	x_e	x_{i+1}
11	1	2	2	3		$x_{a1} \cap x_{e2} \cap x_{b2} \cap x_{f3}$	x_i	x_{i+1}	x_i	x_b	x_e	x_{i+1}
12	1	2	3	3		$x_{a1} \cap x_{e2} \cap x_{b3}$	x_i	x_{i+1}	x_i	x_{i+1}	x_e	x_{i+1}
13	2	2	2	3		$x_{a2} \cap x_{e2} \cap x_{b2} \cap x_{f3}$	x_a	x_{i+1}	x_a	x_b	x_e	x_{i+1}
14	2	2	3	3		$x_{a2} \cap x_{e2} \cap x_{b3}$	x_a	x_{i+1}	x_a	x_{i+1}	x_e	x_{i+1}
15	1	3	1	3		$x_{a1} \cap x_{e3} \cap x_{b1} \cap x_{f3}$	x_i	x_{i+1}	-	-	-	-
16	1	3	2	3		$x_{a1} \cap x_{e3} \cap x_{b2} \cap x_{f3}$	x_i	x_{i+1}	x_i	x_b	-	-
17	1	3	3	3		$x_{a1} \cap x_{e3} \cap x_{b3}$	x_i	x_{i+1}	x_i	x_{i+1}	-	-
18	2	3	2	3		$x_{a2} \cap x_{e3} \cap x_{b2} \cap x_{f3}$	x_a	x_{i+1}	x_a	x_b	-	-
19	2	3	3	3		$x_{a2} \cap x_{e3} \cap x_{b3}$	x_a	x_{i+1}	x_a	x_{i+1}	-	-
20	3	3	3	3		x_{a3}	-	-	-	-	-	-

CATEGORY 2: $(\pi/2 - \Delta\beta) \leq \Phi \leq (\pi/2)$

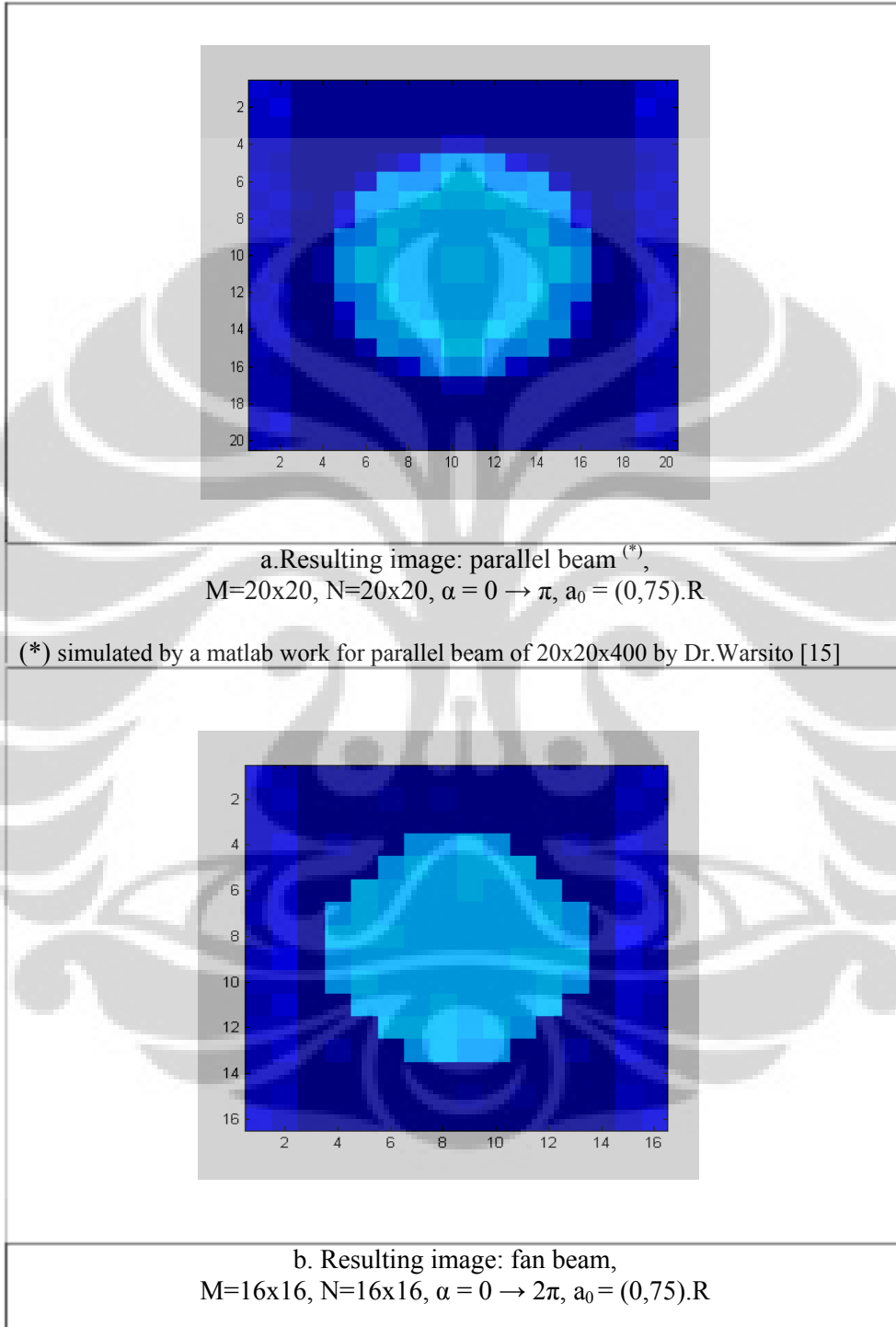
INTERSECTION SHAPE						TRAPEZOID (Wu) =		TOP REST (Wa) =		BOTTOM REST (Wb) =		
No.	COMBINATION $yc \cap yg \cap yd \cap yh$				Illustration	$yc1 \cap yg1 = yc1,$ $yd1 \cap yh1 = yh1,$ $yc3 \cap yg3 = yg3,$ $yd3 \cap yh3 = yd3,$	x_{2u} $\int_{x_{1u}} (Y_2 - Y_1) dx$		x_{2a} $\int_{x_{1a}} (Y_2 - Y_{j+1}) dx$		x_{2b} $\int_{x_{1b}} (Y_j - Y_1) dx$	
	yc	yg	yd	yh			RESUME	x_{1u}	x_{2u}	x_{1a}	x_{2a}	x_{1b}
1	1	1	1	1		yh1	-	-	-	-	-	-
2	1	1	1	2		yc1 ∩ yd1 ∩ yh2	x_f	x_{i+1}	-	-	x_f	x_{i+1}
3	1	1	2	2		yc1 ∩ yd2 ∩ yh2	x_i	x_{i+1}	-	-	x_i	x_{i+1}
4	2	1	2	2		yc2 ∩ yg1 ∩ yd2 ∩ yh2	x_i	x_{i+1}	-	-	x_e	x_{i+1}
5	1	1	1	3		yc1 ∩ yd1 ∩ yh3	x_f	x_{i+1}	x_b	x_{i+1}	x_f	x_{i+1}
6	1	1	2	3		yc1 ∩ yd2 ∩ yh3	x_i	x_{i+1}	x_b	x_{i+1}	x_i	x_{i+1}
7	1	1	3	3		yc1 ∩ yd3	x_i	x_{i+1}	x_i	x_{i+1}	x_i	x_{i+1}
8	2	1	2	3		yc2 ∩ yg1 ∩ yd2 ∩ yh3	x_i	x_{i+1}	x_b	x_{i+1}	x_e	x_{i+1}
9	2	1	3	3		yc2 ∩ yg1 ∩ yd3	x_i	x_{i+1}	x_i	x_{i+1}	x_e	x_{i+1}
10	3	1	3	3		yc3 ∩ yg1 ∩ yd3	x_a	x_{i+1}	x_a	x_{i+1}	x_e	x_{i+1}
11	2	2	2	2		yc2 ∩ yg2 ∩ yd2 ∩ yh2	x_i	x_{i+1}	-	-	-	-
12	2	2	3	3		yc2 ∩ yg2 ∩ yd3	x_i	x_{i+1}	x_i	x_{i+1}	-	-
13	2	2	2	3		yc2 ∩ yg2 ∩ yd2 ∩ yh3	x_i	x_{i+1}	x_b	x_{i+1}	-	-
14	3	2	3	3		yc3 ∩ yg2 ∩ yd3	x_a	x_{i+1}	x_a	x_{i+1}	-	-
15	3	3	3	3		yg3	-	-	-	-	-	-

CATHEGORY 3: $(\pi/2) < \Phi < (\pi - \Delta\beta)$

INTERSECTION SHAPE					TRAPEZOID (Wu) =	TOP REST (Wa) =		BOTTOM REST (Wb) =				
No.	COMBINATION $x_a \cap x_e \cap x_b \cap x_f$				Illustration	$x_{a1} \cap x_{e1} = x_{a1}$, $x_{b1} \cap x_{f1} = x_{b1}$, $x_{a3} \cap x_{e3} = x_{e3}$, $x_{b3} \cap x_{f3} = x_{f3}$, RESUME	x_{2u} $\int_{x_{1u}} (Y_2 - Y_1) dx$		x_{2a} $\int_{x_{1a}} (Y_2 - Y_{j+1}) dx$		x_{2b} $\int_{x_{1b}} (Y_3 - Y_1) dx$	
	x_a	x_e	x_b	x_f			x_{1u}	x_{2u}	x_{1a}	x_{2a}	x_{1b}	x_{2b}
1	3	3	3	3		x_{f3}	-	-	-	-	-	-
2	3	3	3	2		$x_{e3} \cap x_{b3} \cap x_{f2}$	x_f	x_{i+1}	-	-	x_f	x_{i+1}
3	3	3	3	1		$x_{e3} \cap x_{b3} \cap x_{f1}$	x_i	x_{i+1}	-	-	x_i	x_{i+1}
4	3	3	2	2		$x_{e3} \cap x_{b2} \cap x_{f2}$	x_f	x_{i+1}	x_b	x_{i+1}	x_f	x_{i+1}
5	3	3	2	1		$x_{e3} \cap x_{b2} \cap x_{f1}$	x_i	x_{i+1}	x_b	x_{i+1}	x_i	x_{i+1}
6	3	3	1	1		$x_{e3} \cap x_{b1}$	x_i	x_{i+1}	x_i	x_{i+1}	x_i	x_{i+1}
7	3	2	3	2		$x_{a3} \cap x_{e2} \cap x_{b3} \cap x_{f2}$	x_f	x_{i+1}	-	-	x_f	x_e
8	3	2	3	1		$x_{a3} \cap x_{e2} \cap x_{b3} \cap x_{f1}$	x_i	x_{i+1}	-	-	x_i	x_e
9	3	1	3	1		$x_{a3} \cap x_{e1} \cap x_{b3} \cap x_{f1}$	x_i	x_{i+1}	-	-	-	-
10	3	2	2	2		$x_{a3} \cap x_{e2} \cap x_{b2} \cap x_{f2}$	x_f	x_{i+1}	x_b	x_{i+1}	x_f	x_e
11	3	2	2	1		$x_{a3} \cap x_{e2} \cap x_{b2} \cap x_{f1}$	x_i	x_{i+1}	x_b	x_{i+1}	x_i	x_e
12	3	2	1	1		$x_{a3} \cap x_{e2} \cap x_{b1}$	x_i	x_{i+1}	x_i	x_{i+1}	x_i	x_e
13	3	1	2	1		$x_{a3} \cap x_{e1} \cap x_{b2} \cap x_{f1}$	x_i	x_{i+1}	x_b	x_{i+1}	-	-
14	3	1	1	1		$x_{a3} \cap x_{e1} \cap x_{b1}$	x_i	x_{i+1}	x_i	x_{i+1}	-	-
15	2	2	2	2		$x_{a2} \cap x_{e2} \cap x_{b2} \cap x_{f2}$	x_f	x_a	x_b	x_a	x_f	x_e
16	2	2	2	1		$x_{a2} \cap x_{e2} \cap x_{b2} \cap x_{f1}$	x_i	x_a	x_b	x_a	x_i	x_e
17	2	2	1	1		$x_{a2} \cap x_{e2} \cap x_{b1}$	x_i	x_a	x_i	x_a	x_i	x_e
18	2	1	2	1		$x_{a2} \cap x_{e1} \cap x_{b2} \cap x_{f1}$	x_i	x_a	x_b	x_a	-	-
19	2	1	1	1		$x_{a2} \cap x_{e1} \cap x_{b1}$	x_i	x_a	x_i	x_a	-	-
20	1	1	1	1		x_{a1}	-	-	-	-	-	-

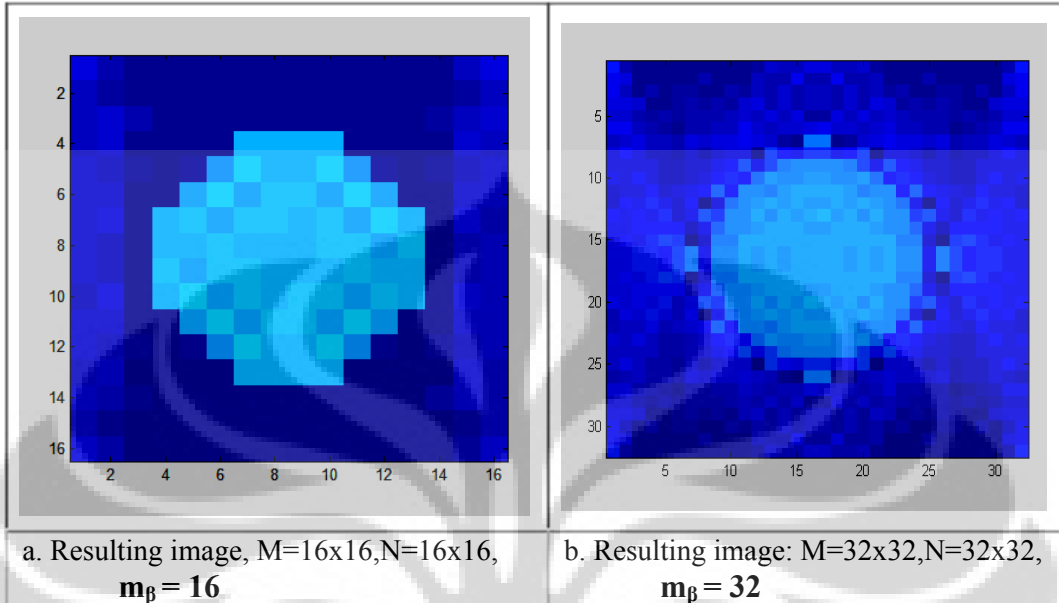
A3. Resulting Images of Parameters Optimization

1 Beam Shape



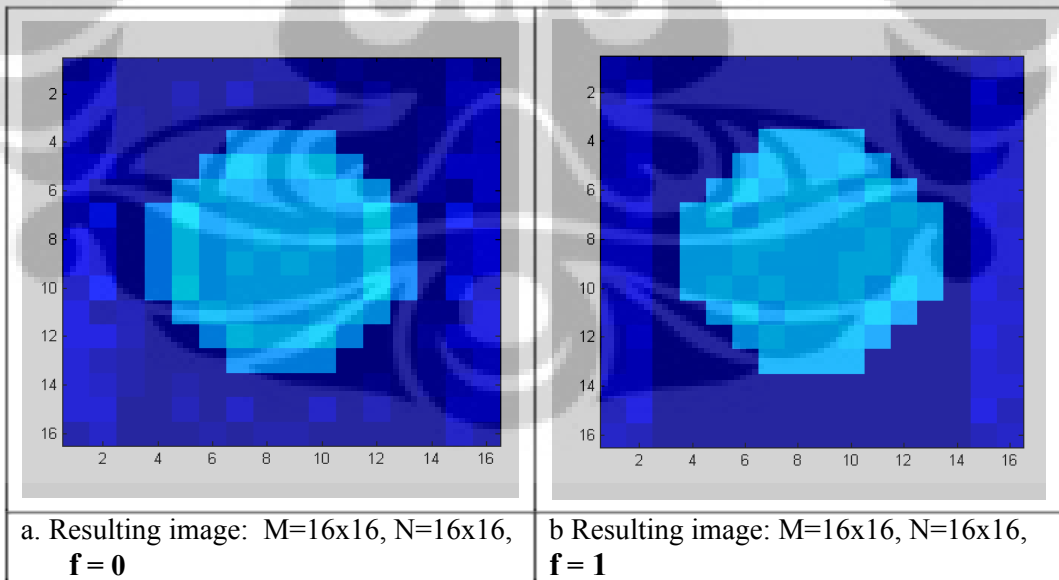
2. Angular Width, $a_0 = (0,5).R$

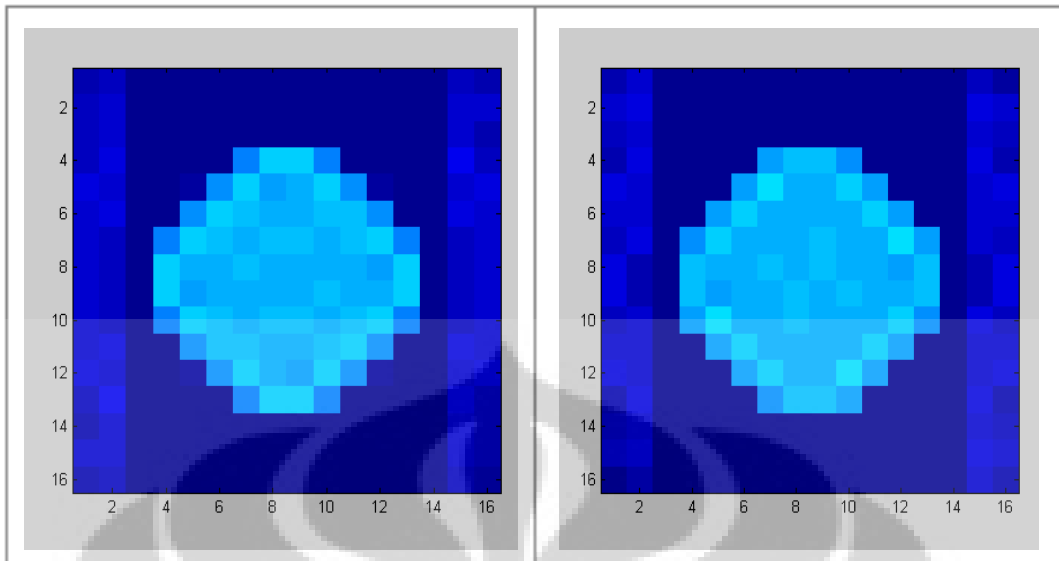
i. Fan Beam Angular Width ($m_\beta \cdot \Delta\beta$), where $(m_\beta \cdot \Delta\beta) = 2 \cdot \arctan(a_0 / (R + a_0))$



ii. Fan Ray Angular Width ($\Delta\beta$), with $\Delta\beta = \arctan(d / (R + a_0 - f \cdot d))$

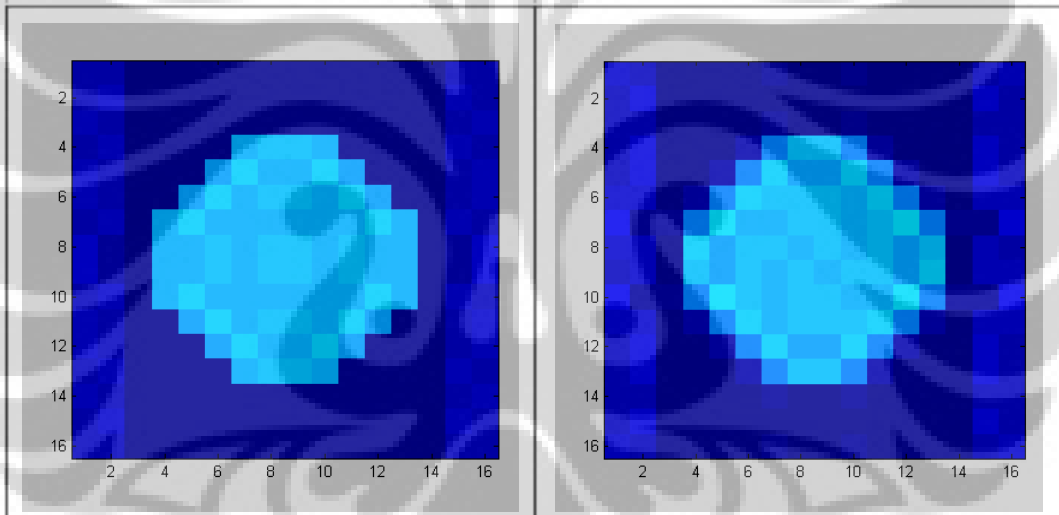
$N = 16 \times 16$





c. Resulting image: $M=16 \times 16$, $N=16 \times 16$, $f = 2$

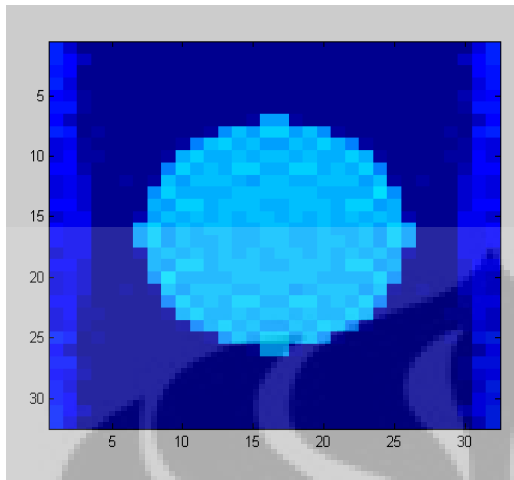
d. Resulting image: $M=16 \times 16$, $N=16 \times 16$, $f = 3$



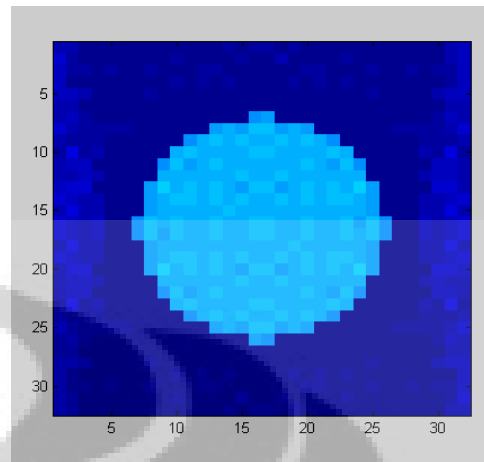
e. Resulting image: $M=16 \times 16$, $N=16 \times 16$, $f = 4$

f. Resulting image: $M=16 \times 16$, $N=16 \times 16$, $f = 5$

$N = 32 \times 32$

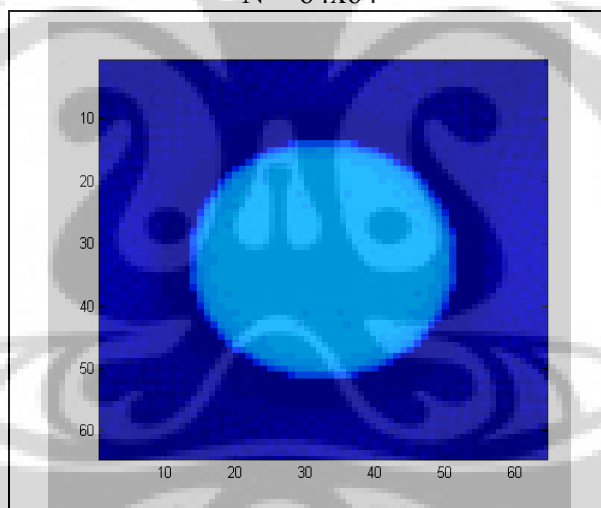


a. Resulting image: $M=32 \times 32$, $N=32 \times 32$,
 $f = 1$



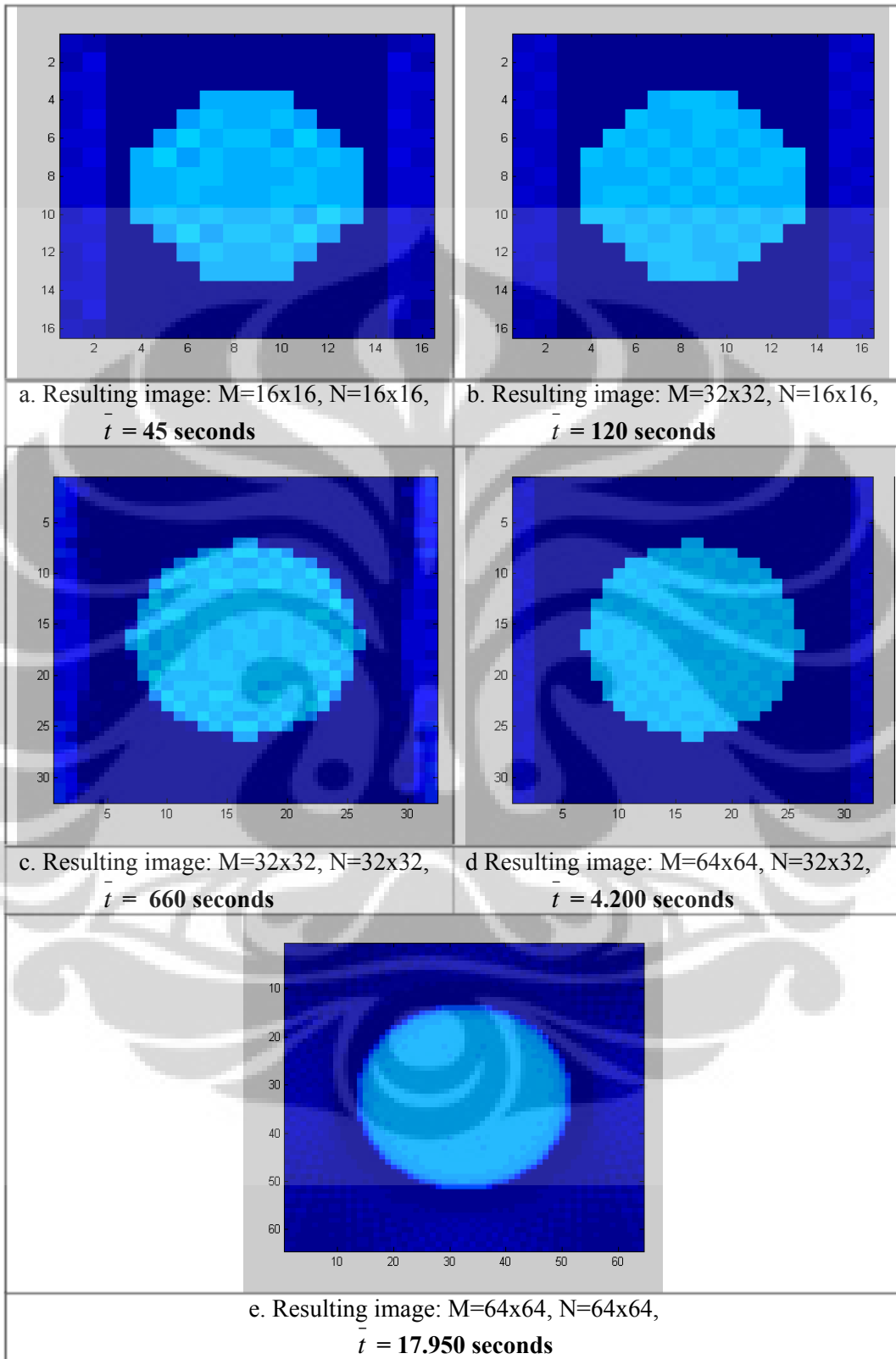
b. Resulting image: $M=32 \times 32$, $N=32 \times 32$,
 $f = 8$

$N = 64 \times 64$

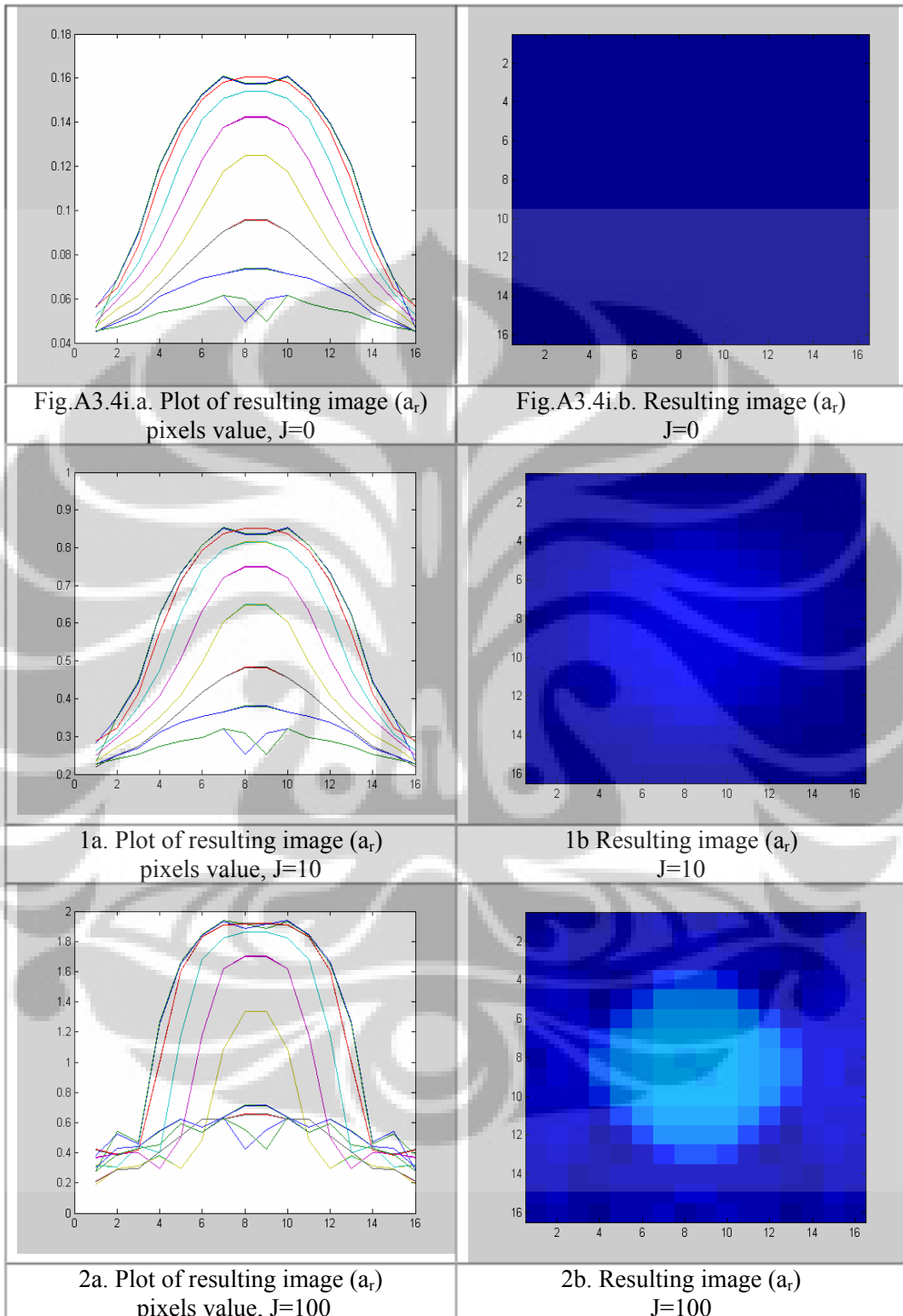


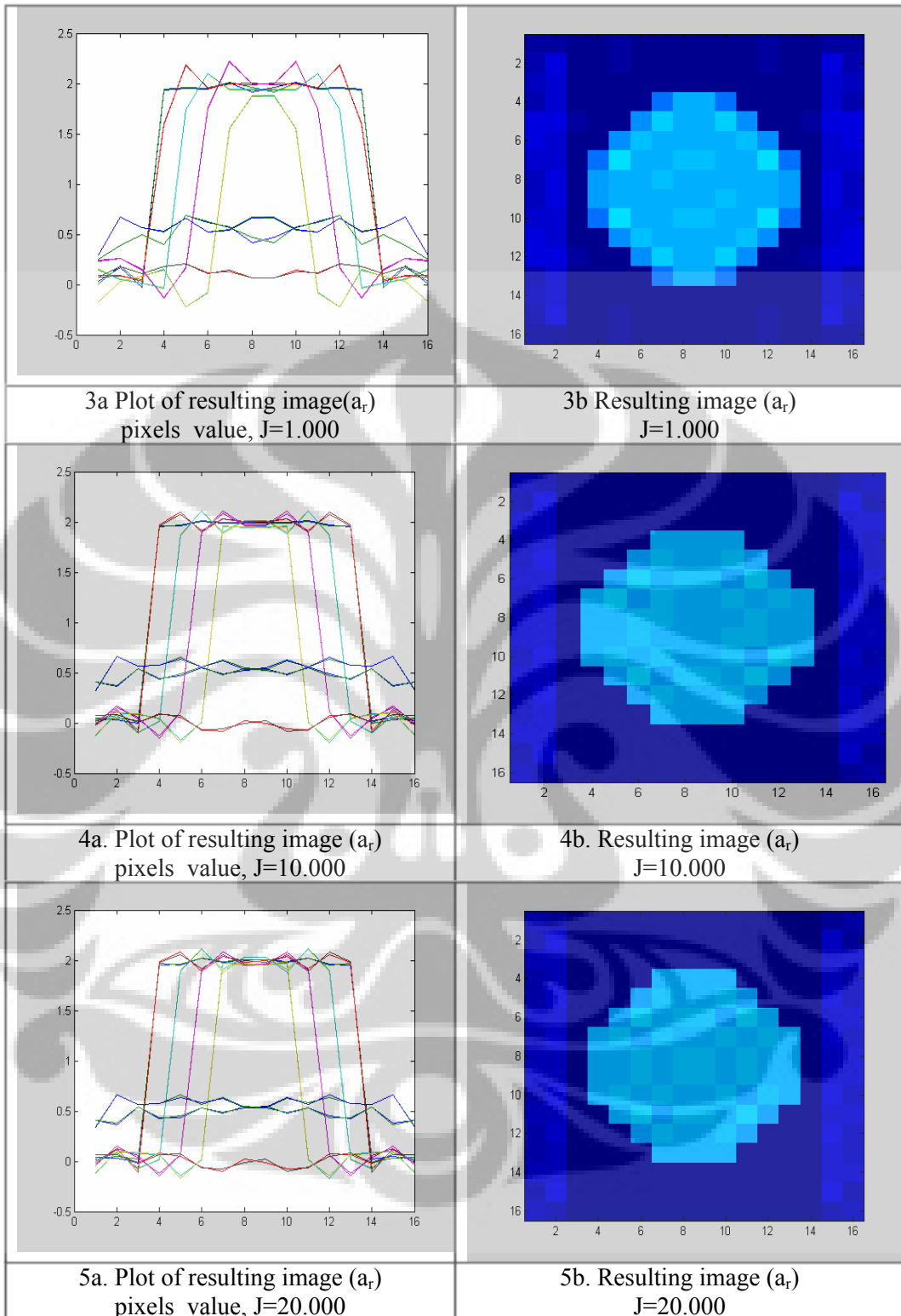
Resulting image: $M = 64 \times 64$, $N=64 \times 64$,
 $f = 1$

3. Reconstruction Time, $a_0 = (0,5).R$



4. Number of Iterations (j) : Filtering Process, $a_0 = (0,5).R$

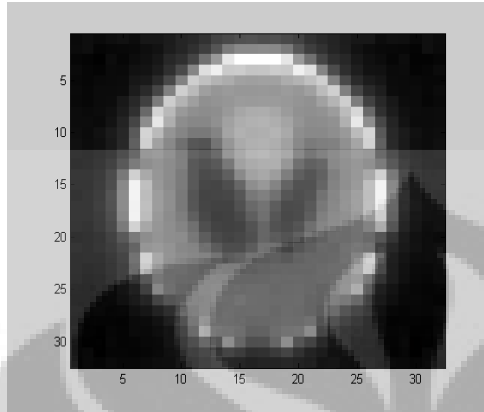




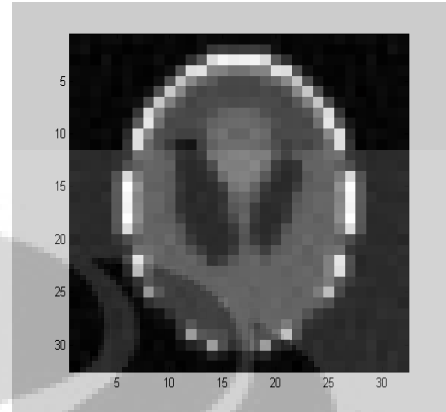
A4. Qualitative Experimental Data

1. $N=32 \times 32$, $M=64 \times 64$

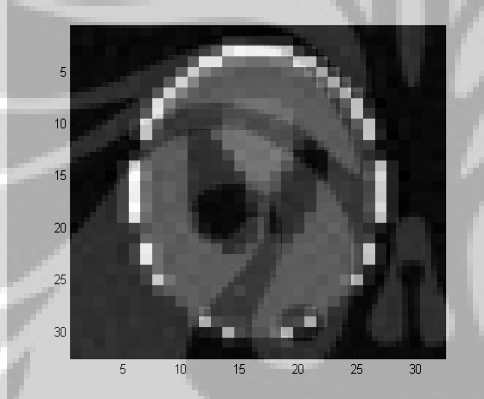
$$\lambda = 2$$



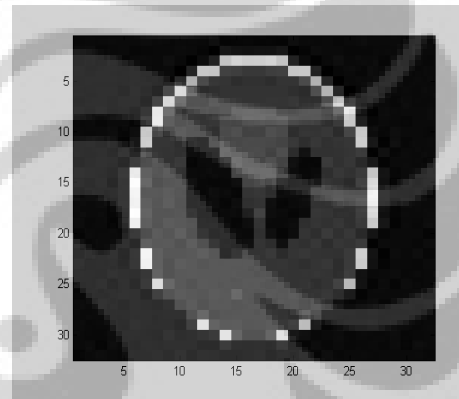
j = 100 iterations



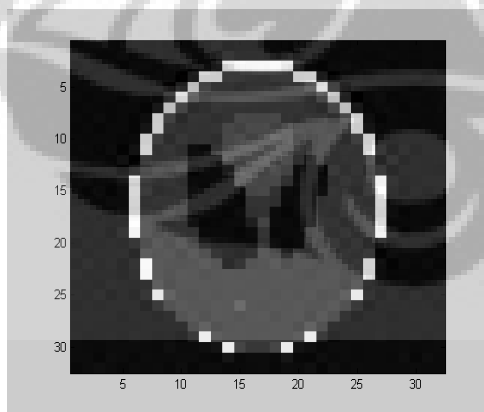
j = 500 iterations



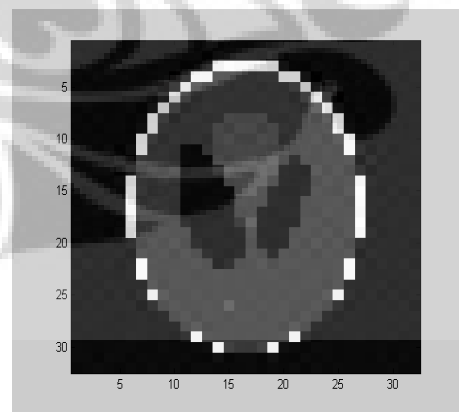
j = 1000 iterations



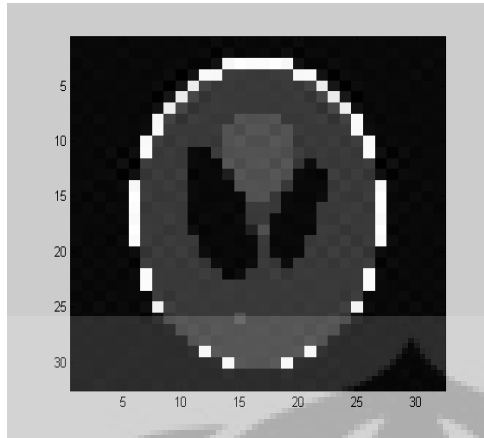
j = 2000 iterations



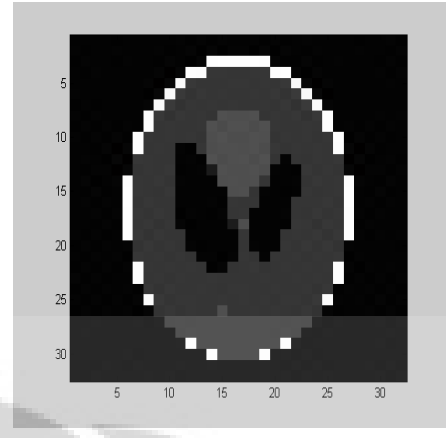
j = 3000 iterations



j = 4000 iterations

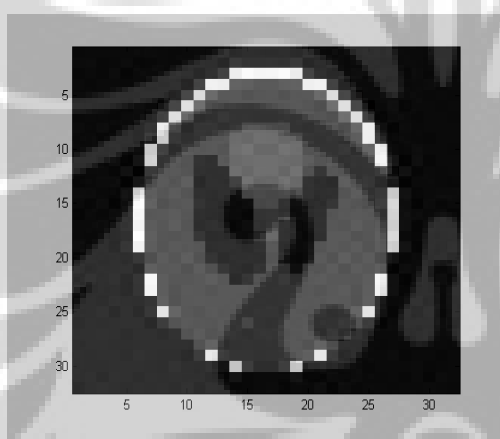


j = 5000 iterations

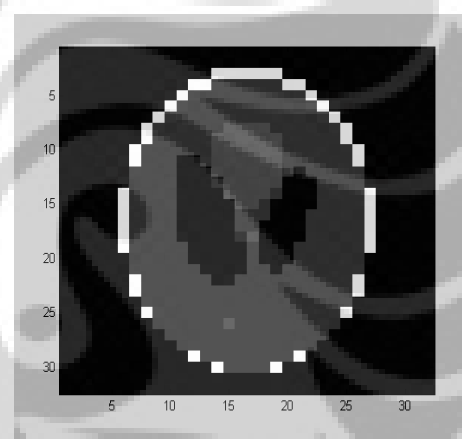


j = 10000 iterations

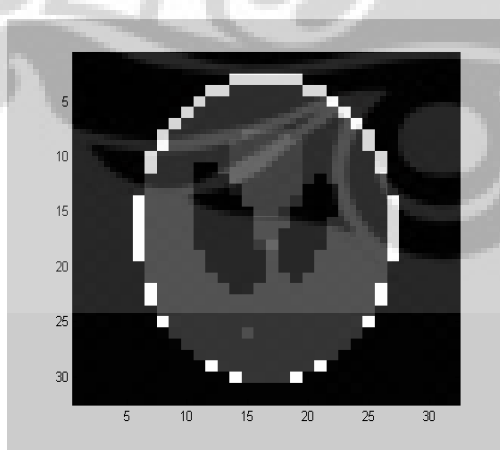
j = 10.000 iterations



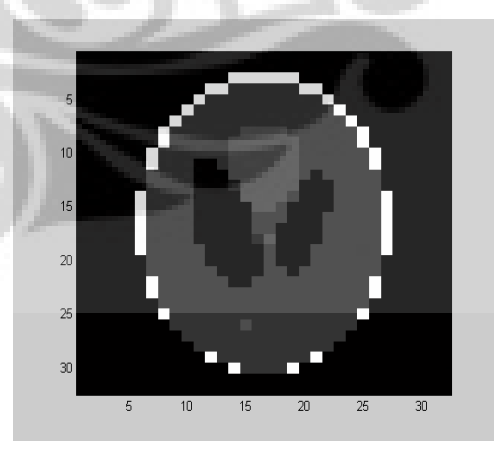
$\lambda = 1$



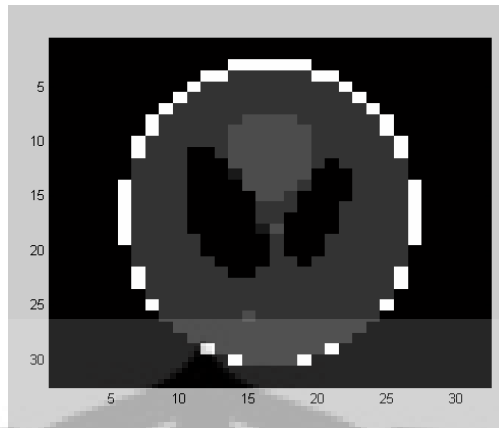
$\lambda = 2$



$\lambda = 5$



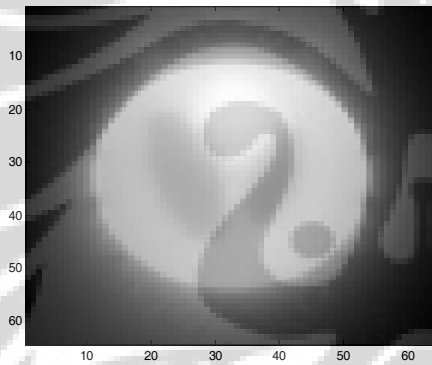
$\lambda = 10$



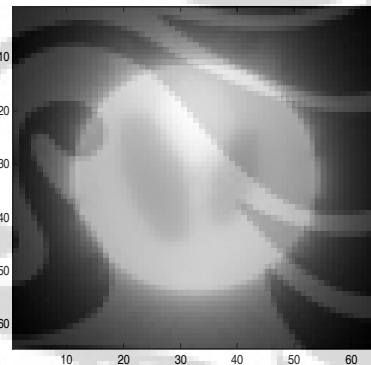
$\lambda = 15$

2. $N=64 \times 64$, $M=128 \times 64$

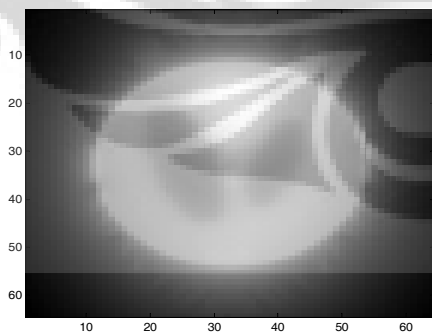
$\lambda = 2$



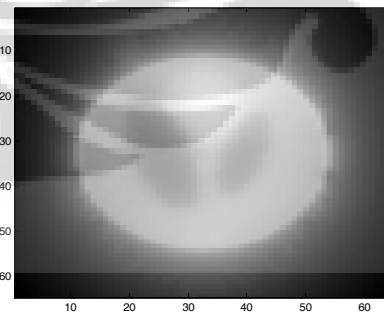
j = 10 iterations



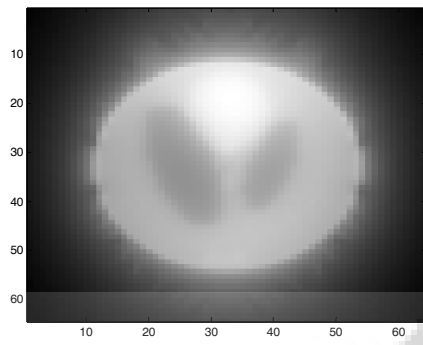
j = 20 iterations



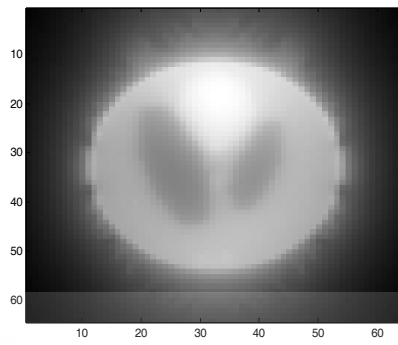
j = 30 iterations



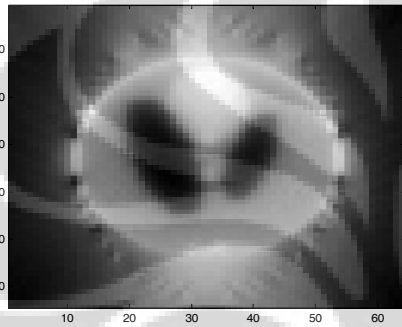
j = 40 iterations



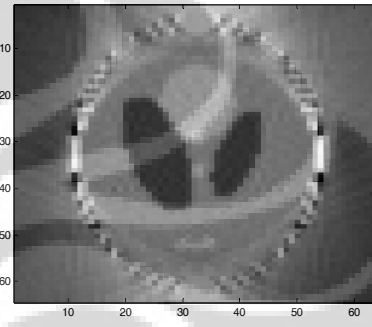
j = 50 iterations



j = 100 iterations

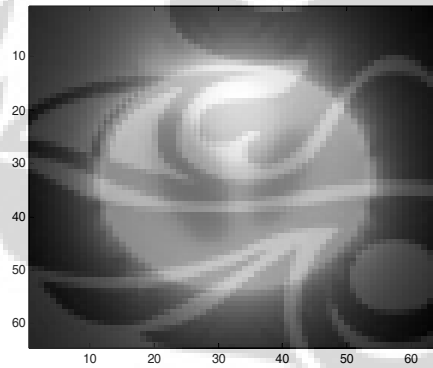


j = 1000 iterations

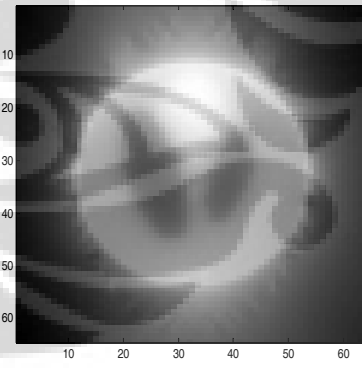


j = 10000 iterations

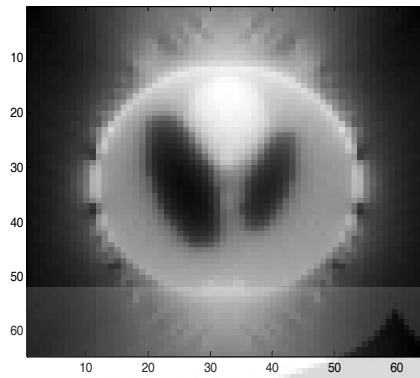
j = 100 iterations



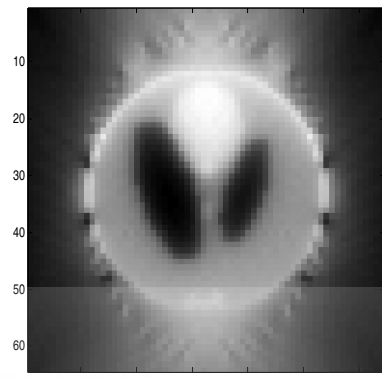
$\lambda = 2$



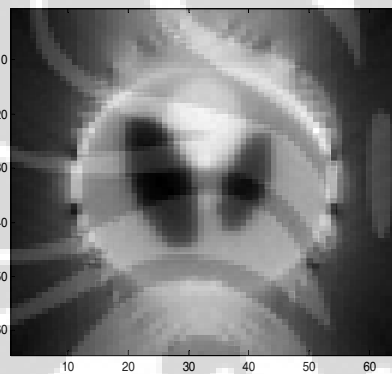
$\lambda = 5$



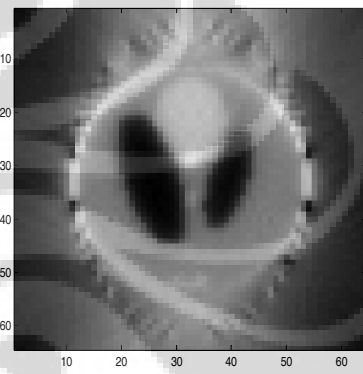
$\lambda = 10$



$\lambda = 20$



$\lambda = 30$



$\lambda = 40$

A5. MATLAB Programs

```
% PROJECTION MATRIX - THESIS
% CT PROJECTION PLAN
R = 2; % CT-radius
mb = 32; % number of rays in a beam
ma = 32; % number of beams
dalpha= 2*pi/ma; % indices of beam direction
N = 1024; % number of pixels of object matrix
% OBJECT MATRIX
n = sqrt(N); % number of pixels in x or y direction
a0 = R/2; % half width of matrix
d = 2*a0/n; % pixel width

%dbeta = 2/mb*atan(a0/(R+a0));% width of a beam
dbeta=atan(d/(R+a0-d)); % width of a ray

% CARTESIAN COORDINATE SYSTEM
x = [-a0:d:a0]; % x-axis,
y = [-a0:d:a0]; % y-axis,
beta = (-mb/2*dbeta):dbeta:(mb/2*dbeta-dbeta); % ray directions in a beam

% PROJECTION FUNCTION
M=0;
i=0;
for alpha = pi/4:dalpha:2*pi+pi/4-dalpha % beam direction
    i = i+1; % counter for alpha
    for j=1:mb % counter for beta
        M=M+1; % counter for pi
        phi(j) = alpha+beta(j); % ray directions in CT system
        % RAY LINEAR EQUATION
        m1(j) = -1/tan(phi(j)); % ray-Y1 gradient
        m2(j) = -1/tan(phi(j)+dbeta); % ray-Y2 gradient
        b1(j) = R*(cos(alpha)+m1(j)*sin(alpha)); % y(x=0)ordinate of Y1
        b2(j) = R*(cos(alpha)+m2(j)*sin(alpha)); % y(x=0)ordinate of Y2
        for k=1:N % pixels numbering
            r=round(k/n-0.5001)+1;
```

```

s=rem(k,n);
if s==0
    s=r-1;
    s=n;
end
x(r)= -a0+d*(r-1);
y(s)= -a0+d*(s-1);

% MATRIX BACK ROTATION, RADON TRANSFORM
% QUADRANT 1 TO 4
k4 = n*s-r+1;           % pixels number in Quadrant 4
% QUADRANT 1&4 TO 3&2
k23 = N-k+1;           % pixels number in Quadrant 2&3

% INTERSECTION POINTS
xa=1/m1(j)*(y(s+1)-b1(j)); % intersection: Y1 and Y= y(s+1)
xb=1/m2(j)*(y(s+1)-b2(j)); % intersection: Y2 and Y= y(s+1)
xe=1/m1(j)*(y(s) -b1(j)); % intersection: Y1 and Y= y(s)
xf=1/m2(j)*(y(s) -b2(j)); % intersection: Y2 and Y= y(s)
yc=m1(j)*x(r) +b1(j); % intersection: Y1 and X= x(r)
yd=m2(j)*x(r) +b2(j); % intersection: Y2 and X= x(r)
yg=m1(j)*x(r+1)+b1(j); % intersection: Y1 and X= x(r+1)
yh=m2(j)*x(r+1)+b2(j); % intersection: Y2 and X= x(r+1)

% SLOPE CONDITIONS (xa;xb;xe;xf)
xa1=(xa<x(r));
xa2=((x(r)<=xa)&(xa<=x(r+1)));
xa3=(x(r+1)<xa);
xb1=(xb<x(r));
xb2=((x(r)<=xb)&(xb<=x(r+1)));
xb3=(x(r+1)<xb);
xe1=(xe<x(r));
xe2=((x(r)<=xe)&(xe<=x(r+1)));
xe3=(x(r+1)<xe);
xf1=(xf<x(r));
xf2=((x(r)<=xf)&(xf<=x(r+1)));
xf3=(x(r+1)<xf);
% SLOPE CONDITIONS (yc;yd;yg;yh)
yc1=(yc<y(s));

```



```

yc2=((y(s)<=yc)&(yc<=y(s+1)));
yc3=(y(s+1)<yc);
yd1=(yd<y(s));
yd2=((y(s)<=yd)&(yd<=y(s+1)));
yd3=(y(s+1)<yd);
yg1=(yg<y(s));
yg2=((y(s)<=yg)&(yg<=y(s+1)));
yg3=(y(s+1)<yg);
yh1=(yh<y(s));
yh2=((y(s)<=yh)&(yh<=y(s+1)));
yh3=(y(s+1)<yh);

% PROJECTION FUNCTION
if (pi/4<=alpha)&(alpha<3/4*pi)
    x1u=0;x2u=0; x1a=0;x2a=0; x1b=0;x2b=0;
    % CATHEGORY 1
    if (0<phi(j))&(phi(j)<pi/2-dbeta)
        if xa1&xe1&xb1&xf1
            w(i,j,k)=0;
            %W(M,k)=w(i,j,k);
        end
        if xa1&xe1&xb1&xf2
            x1u=x(r);x2u=xf; x1b=x(r);x2b=xf;
        end
        if xa1&xe1&xb2&xf2
            x1u=x(r);x2u=xf; x1a=x(r);x2a=xb; x1b=x(r);x2b=xf;
        end
        if xa1&xe1&xb1&xf3
            x1u=x(r);x2u=x(r+1); x1b=x(r);x2b=x(r+1);
        end
        if xa1&xe1&xb2&xf3
            x1u=x(r);x2u=x(r+1); x1a=x(r);x2a=xb; x1b=x(r);x2b=x(r+1);
        end
        if xa1&xe1&xb3&xf3
            w(i,j,k)=d0^2;
            %W(M,k)=w(i,j,k);
        end
        if xa1&xe2&xb1&xf2
            x1u=x(r);x2u=xf; x1b=xe;x2b=xf;

```

```

end
if xa1&xe2&xb2&xf2
    x1u=x(r);x2u=xf; x1a=x(r);x2a=xb; x1b=xe;x2b=xf;
end
if xa2&xe2&xb2&xf2
    x1u=xa;x2u=xf; x1a=xa;x2a=xb; x1b=xe;x2b=xf;
end
if xa1&xe2&xb1&xf3
    x1u=x(r);x2u=x(r+1); x1b=xe;x2b=x(r+1);
end
if xa1&xe2&xb2&xf3
    x1u=x(r);x2u=x(r+1); x1a=x(r);x2a=xb; x1b=xe;x2b=x(r+1);
end
if xa1&xe2&xb3&xf3
    x1u=x(r);x2u=x(r+1); x1a=x(r);x2a=x(r+1); x1b=xe;x2b=x(r+1);
end
if xa2&xe2&xb2&xf3
    x1u=xa;x2u=x(r+1); x1a=xa;x2a=xb; x1b=xe;x2b=x(r+1);
end
if xa2&xe2&xb3&xf3
    x1u=xa;x2u=x(r+1); x1a=xa;x2a=x(r+1); x1b=xe;x2b=x(r+1);
end
if xa1&xe3&xb1&xf3
    x1u=x(r);x2u=x(r+1);
end
if xa1&xe3&xb2&xf3
    x1u=x(r);x2u=x(r+1); x1a=x(r);x2a=xb;
end
if xa1&xe3&xb3&xf3
    x1u=x(r);x2u=x(r+1); x1a=x(r);x2a=x(r+1);
end
if xa2&xe3&xb2&xf3
    x1u=xa;x2u=x(r+1); x1a=xa;x2a=xb;
end
if xa2&xe3&xb3&xf3
    x1u=xa;x2u=x(r+1); x1a=xa;x2a=x(r+1);
end
if xa3&xe3&xb3&xf3
    w(i,j,k)=0;

```

```

        %W(M,k)=w(i,j,k);
    end
    % CATEGORY 2
    else if (pi/2-dbeta<=phi(j))&(phi(j)<=pi/2)
        if yc1&yg1&yd1&yh1
            w(i,j,k)=0;
            %W(M,k)=w(i,j,k);
        end
        if yc1&yg1&yd1&yh2
            x1u=xf;x2u=x(r+1); x1b=xf;x2b=x(r+1);
        end
        if yc1&yg1&yd2&yh2
            x1u=x(r);x2u=x(r+1); x1b=x(r);x2b=x(r+1);
        end
        if yc2&yg1&yd2&yh2
            x1u=x(r);x2u=x(r+1); x1b=xe;x2b=x(r+1);
        end
        if yc1&yg1&yd1&yh3
            x1u=xf;x2u=x(r+1); x1a=xb;x2a=x(r+1); x1b=xf;x2b=x(r+1);
        end
        if yc1&yg1&yd2&yh3
            x1u=x(r);x2u=x(r+1); x1a=xb;x2a=x(r+1); x1b=x(r);x2b=x(r+1);
        end
        if yc1&yg1&yd3&yh3
            w(i,j,k)=d0^2;
            %W(M,k)=w(i,j,k);
        end
        if yc2&yg1&yd2&yh3
            x1u=x(r);x2u=x(r+1); x1a=xb;x2a=x(r+1); x1b=xe;x2b=x(r+1);
        end
        if yc2&yg1&yd3&yh3
            x1u=x(r);x2u=x(r+1); x1a=x(r);x2a=x(r+1); x1b=xe;x2b=x(r+1);
        end
        if yc3&yg1&yd3&yh3
            x1u=xa;x2u=x(r+1); x1a=xa;x2a=x(r+1); x1b=xe;x2b=x(r+1);
        end
        if yc2&yg2&yd2&yh2
            x1u=x(r);x2u=x(r+1);
        end
    end

```

```

if yc2&yg2&yd3&yh3
    x1u=x(r);x2u=x(r+1); x1a=x(r);x2a=x(r+1);
end
if yc2&yg2&yd2&yh3
    x1u=x(r);x2u=x(r+1); x1a=xb;x2a=x(r+1);
end
if yc3&yg2&yd3&yh3
    x1u=xa;x2u=x(r+1); x1a=xa;x2a=x(r+1);
end
if yc3&yg3&yd3&yh3
    w(i,j,k)=0;
    %W(M,k)=w(i,j,k);
end
% CATHEGORY 2
else if (pi/2<phi(j))&(phi(j)<pi-dbeta)
    if xa3&xe3&xb3&xf3
        w(i,j,k)=0;
        %W(M,k)=w(i,j,k);
    end
    if xa3&xe3&xb3&xf2
        x1u=xf;x2u=x(r+1);x1b=xf;x2b=x(r+1);
    end
    if xa3&xe3&xb3&xf1
        x1u=x(r);x2u=x(r+1);x1b=x(r);x2b=x(r+1);
    end
    if xa3&xe3&xb2&xf2
        x1u=xf;x2u=x(r+1); x1a=xb;x2a=x(r+1); x1b=xf;x2b=x(r+1);
    end
    if xa3&xe3&xb2&xf1
        x1u=x(r);x2u=x(r+1); x1a=xb;x2a=x(r+1); x1b=x(r);x2b=x(r+1);
    end
    if xa3&xe3&xb1&xf1
        w(i,j,k)=d0^2;
        %W(M,k)=w(i,j,k);
    end
    if xa3&xe2&xb3&xf2
        x1u=xf;x2u=x(r+1); x1b=xf;x2b=xe;
    end
    if xa3&xe2&xb3&xf1

```

```

    x1u=x(r);x2u=x(r+1); x1b=x(r);x2b=xe;
end
if xa3&xe1&xb3&xf1
    x1u=x(r);x2u=x(r+1);
end
if xa3&xe2&xb2&xf2
    x1u=xf;x2u=x(r+1); x1a=xb;x2a=x(r+1); x1b=xf;x2b=xe;
end
if xa3&xe2&xb2&xf1
    x1u=x(r);x2u=x(r+1); x1a=xb;x2a=x(r+1); x1b=x(r);x2b=xe;
end
if xa3&xe2&xb1&xf1
    x1u=x(r);x2u=x(r+1); x1a=x(r);x2a=x(r+1); x1b=x(r);x2b=xe;
end
if xa3&xe1&xb2&xf1
    x1u=x(r);x2u=x(r+1); x1a=xb;x2a=x(r+1);
end
if xa3&xe1&xb1&xf1
    x1u=x(r);x2u=x(r+1); x1a=x(r);x2a=x(r+1);
end
if xa2&xe2&xb2&xf2
    x1u=xf;x2u=xa; x1a=xb;x2a=xa; x1b=xf;x2b=xe;
end
if xa2&xe2&xb2&xf1
    x1u=x(r);x2u=xa; x1a=xb;x2a=xa; x1b=x(r);x2b=xe;
end
if xa2&xe2&xb1&xf1
    x1u=x(r);x2u=xa; x1a=x(r);x2a=xa; x1b=x(r);x2b=xe;
end
if xa2&xe1&xb2&xf1
    x1u=x(r);x2u=xa; x1a=xb;x2a=xa;
end
if xa2&xe1&xb1&xf1
    x1u=x(r);x2u=xa; x1a=x(r);x2a=xa;
end
if xa1&xe1&xb1&xf1
    w(i,j,k)=0;
    %W(M,k)=w(i,j,k);
end

```

```

        end
    end
end
wu(i,j,k) = [0.5*(m2(j)-m1(j))*(x2u^2-x1u^2)]+[(b2(j)-b1(j))*(x2u-x1u)];
wa(i,j,k)= [0.5*m2(j)*(x2a^2-x1a^2)]+[(b2(j)-y(s+1))*(x2a-x1a)];
wb(i,j,k)= [-0.5*m1(j)*(x2b^2-x1b^2)]+[(y(s)-b1(j))*(x2b-x1b)];
w (i,j,k)= wu(i,j,k)-wa(i,j,k)-wb(i,j,k);

else if (3/4*pi<=alpha)&(alpha<5/4*pi)
    w(i,j,k)= w((i-(ma/4)),j,k4);

else if (5/4*pi<=alpha)&(alpha<2*pi+pi/4)
    w(i,j,k)= w((i-(ma/2)),j,k23);
end
end
end
W(M,k) = w(i,j,k);
end
end
disp('Save projection matrix with filename projmat_thesis_1024.m');
save(['projmat_thesis_1024'],'W')
=====

```

% SIMULATED PROJECTION DATA

```
load objmat_thesis -mat
load projmat_thesis -mat
A=reshape(a,1024,1); % reshaping object_matrix of 32x32 to 1024x1
P=W*A;
disp('Save projdat matrix with filename projdat_thesis');
save (['projdat_thesis'],'P')
```

% BACKPROJECTION

```
load objmat_thesis -mat
load projmat_thesis -mat
load projdat_thesis -mat
Ar=W' *P;
lambda=2;
iter=10000;
for j=1:iter;
j
Ar=Ar-lambda*W' *(W*Ar-P);
end
ar=reshape(Ar,32,32);
disp('Save image matrix with filename immat_thesis');
save (['immat_thesis'],'ar')
imagesc (ar);
colormap (gray(256));
```

```

%=====
% MEAN SQUARE ERROR GRAPHIC
% ITERATIONS OF LAMBDA = 2

clear
% OBJECT MATRIX
a=phantom('Modified Shepp-Logan',32);
k=1:1024;
for i=1:8
    as(i,k)= reshape (a,1,1024);
end

% INPUT IMAGE MATRIX DATA
load IMMAT_thesis_32phantom_i100rayb_a2 -mat;
ars(1,k)=reshape (ar,1,1024);
load IMMAT_thesis_32phantom_i500rayb_a2 -mat;
ars(2,k)=reshape (ar,1,1024);
load IMMAT_thesis_32phantom_i1000rayb_a2 -mat;
ars(3,k)=reshape (ar,1,1024);
load IMMAT_thesis_32phantom_i2000rayb_a2 -mat;
ars(4,k)=reshape (ar,1,1024);
load IMMAT_thesis_32phantom_i3000rayb_a2 -mat;
ars(5,k)=reshape (ar,1,1024);
load IMMAT_thesis_32phantom_i4000rayb_a2 -mat;
ars(6,k)=reshape (ar,1,1024);
load IMMAT_thesis_32phantom_i5000rayb_a2 -mat;
ars(7,k)=reshape (ar,1,1024);
load IMMAT_thesis_32phantom_i10000rayb_a2 -mat;
ars(8,k)=reshape(ar,1,1024);

% MSE
das= abs(as-ars)
darv=das.^2;
for i=1:8
    sumdarv(i)=sum(darv(i,1:1024));

```



```
mse(i)=sumdarv(i)/1024
end
save(['mse32_M32x32_lambda2'],'mse')

% GRAPHIC
iter =[100 500 1000 2000 3000 4000 5000 10000]
plot(iter,mse)
plot(iter,mse,'--rs','LineWidth',2,...
      'MarkerEdgeColor','k',...
      'MarkerFaceColor','g',...
      'MarkerSize',10)
%-----
```

