Universitat Autònoma de Barcelona

MASTER IN COMPUTER VISION AND ARTIFICIAL INTELLIGENCE
**REPORT OF THE RESEARCH PROJECT**
OPTION: COMPUTER VISION

# Use of Projection and Back-projection Methods in Bidimensional Computed Tomography Image Reconstruction

**Autor: Jorge Bernal del Nozal**
**Data: 14th of September 2009**
**Tutor: Javier Sánchez Pujadas**

Universitat Autònoma de Barcelona

# Use of Projection and Back-projection Methods in Bidimensional Computed Tomography image reconstruction

**Jorge Bernal del Nozal** jbernal@cvc.uab.es
*Computer Vision Center*
*Edifici O, Universitat Autònoma de Barcelona*
*08193, Bellaterra (Spain)*

**Supervisor:** Dr. Javier Sánchez

## Abstract

One of the biggest drawbacks related to the use of CT scanners is the cost (in memory and in time) associated. In this project many methods to simulate their functioning, but in a more feasible way (taking an industrial point of view), will be studied.

The main group of techniques that are being used are the one entitled as 'back-projection'. The concept behind is to simulate the X ray emission in CT scans by lines that cross with the image we want to reconstruct.

In the first part of this document euclidean geometry is used to face the tasks of projection and back-projection. After analysing the results achieved it has been proved that this approach does not lead to a fully perfect reconstruction (and also has some other problems related to running time and memory cost). Because of this in the second part of the document 'Filtered Back-projection' method is introduced in order to improve the results.

Filtered Back-projection methods rely on mathematical transforms (Fourier, Radon) in order to provide more accurate results that can be obtained in much less time. The main cause of this better results is the use of a filtering process before the back-projection in order to avoid high frequency-caused errors.

As a result of this project two different implementations (one for each approach) had been implemented in order to compare their performance.

**Keywords:** Projection, Back-projection, CT scan, Euclidean geometry, Radon transform

## Contents

## List of Figures

## 1. Introduction

One of the main motivations that lead to face this project is being able to simulate by cameras the way CT scans work in order to avoid the high cost associated to this kind of devices. CT is the acronym of Computed Tomography [Feng-rong Sun (2008); David J. Brenner and Eric J. Hall (2008)], which consists on using X ray to generate 2D images of objects (i.e, human body). Images are taken by rotating the equipment 360º round the human body, as it can be seen in Figure 1.


Figure 1: CT scanner

The amount of radiation emitted[David J. Brenner and Eric J. Hall (2008)] is measured by a ring of detectors placed in the gate-shape structure that the patient is introduced in. The image is created from this measures so the internal structure of the human body can be reconstructed from X ray projections. In this kind of devices, data is combined by using a technique called topographic reconstruction. Data is placed inside a matrix in computer's memory, convolving each piece of data with its neighbours using a seed algorithm and FFT, rising each time the resolution of each volume element or voxel. The next step consists on using back-projection [Vu Cong (2003); Sources (2003)], which is the opposite to the previous process, storing results in a array. This data can be shown, pictured or used as input in posterior processes like Multiplanar Reconstruction. In the industrial environment that we are working on, there are several manufacturers [X-View (2009b); Imaging (2009)] that are applying CT techniques with several aims such as detecting failures in pieces or observing certain parts of an object in order to extract useful information. In Figure 2 one example of an industrial CT scan can be observed. In this case we do not have a ring of detectors but a flat moving area [Noel (2008)] which will receive the rays emitted from the X-Ray tube. The problem is that these kind of devices, in order to get a perfect reconstruction of the object needs from 360 to 3600 images (depending on the angular offset) which leads to slow (more than three minutes in some cases to acquire all the necessary images [ray Computed Tomography Facility (2009)] and high-memory consuming processes. For example, the fastest device that the company X-View can provide to a client (in this case, Boeing) takes nearly 2 minutes [X-View (2009a)] to complete the process. While it is not a lot of time compared to hours that the process used to take, it is a considerable time if we want to put a device like this in the production line.

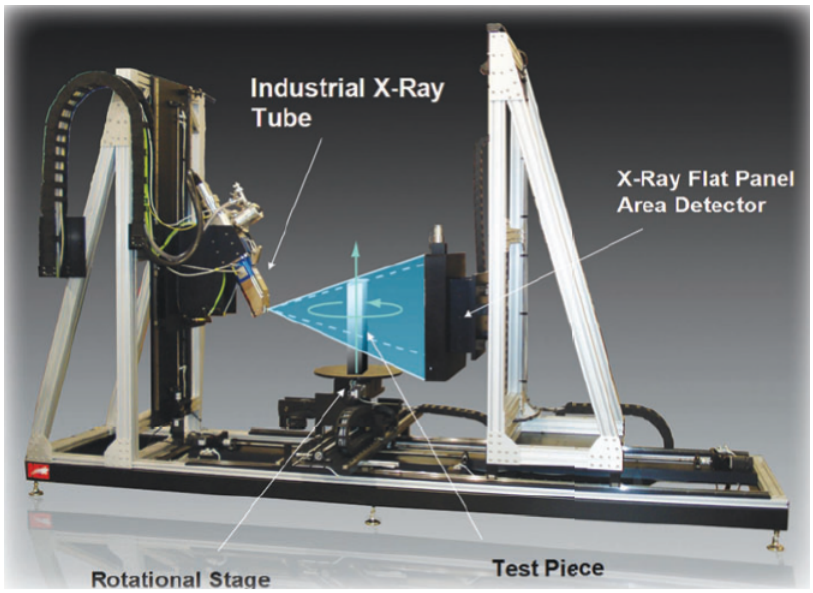



Figure 2: Industrial CT scanner

The objective of this project is to apply projection and back-projection techniques but under an industrial point of view. That means, our objective is not only to reconstruct faithfully an image, but to do this process in a fast and feasible way, trying to find out how many different views are needed in order to reconstruct the image without losing too much accuracy. A reduction in the number of views needed will lead to faster and less memory consuming process which will not affect the overall performance of the system.
In order to simulate the functioning of the industrial CT scanners we have implemented two different methods, an Euclidean Geometry based and a Filtered Back-projection solution, each one following one of the actual research lines in the field, as it will be observed in the Related Work section. After a first implementation of an algorithm based in euclidean geometry is clear that the image is well reconstructed in terms of shape but not in accuracy of grey value. The method also has inconveniences such as long processing time and excessive memory cost. All these facts lead us to search for another solution that keeps the good points of the previous one but improves its weak points. One big group of techniques is the one referred as Filtered Back-projection [Gopi (2004); Pow (2003)]. The majority of this methods are based in the use of the Radon transform [Vu Cong (2003)], because the data represent, directly or after a preprocessing, the Radon transform of the scanned image. It seems pretty clear that the image reconstruction problem now consists in solving how to invert the Radon transform. Enclosed in this group of methods there are some techniques [J.L. Fernandez Marron (2004); Raman P.V. Rao (1995)] that after achieving the back-projection apply a windowed ramp filter to get the final reconstructed image. Taking this new framework into account, in this project it has been implemented a Filtered Back-projection algorithm that uses different types of filters to do the windowing process. In the third section both solutions will be fully detailed along with explanation of how the two implemented algorithms work.

## 2. Related Work

Before starting to present the method we have followed, it is necessary to take a look at what has been done in this field during the last years. This section will be divided into the two main approaches, cone/fan-beam projection and Filtered Back-projection. While it may seem that the second is a refinement of the first one, indeed research is going along both paths in parallel so there have been improvement around both approaches, as it will be explained shortly.

### 2.1 Cone/Fan-beam Projection

This group of methods are all based in the emission of rays taking the shape of a fan, simulating the functioning of a X-ray emitter in CT scanners, as it can be seen in Figure 3

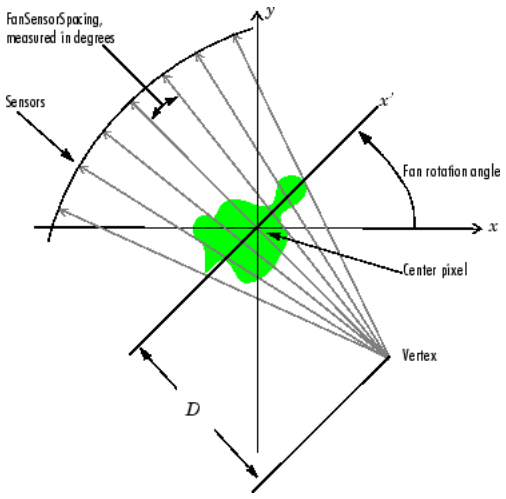



Figure 3: Fan-beam projection schema

As it can be seen in the figure above, a group of beams with the same source point are emitted towards the object that we want to later reconstruct. The beams that act as X-rays cross the image (which represent the object) in several pixel positions. One of the most typical approaches is calculating the integral along the beam, storing the data in a projection matrix which will be used later in the reconstruction process. Once we have the projection data from a source point there are two main alternatives: rotate the object and repeat the projection step from the same source point or use another source point that is placed in the same circumference as the previous one but separated some degrees. Once this step is decided, the main difference between the methods that have been applied throughout the years is how they get and later treat the projection data. Here are presented some of the papers that have inspired somehow the work in this project:

- **'Rectification for Cone-Beam Projection and Back-projection'**: This work [Riddell and Trousset (2006)] is focused on finding a technique that decreases the computational time and increases the accuracy. The rectification method that is presented tries to solve the problem of continuous memory jumps while accessing the projection data, because the projection of a volume cannot be assumed aligned with any of the axes of the volume. With this method all volume planes are accumulated after 2-D mapping

onto an unique plane, as it can be seen in Figure 4. This whole method is applied in a field (Rotational Angiography) that, although is not the same as ours, uses some techniques that can be useful in the project.

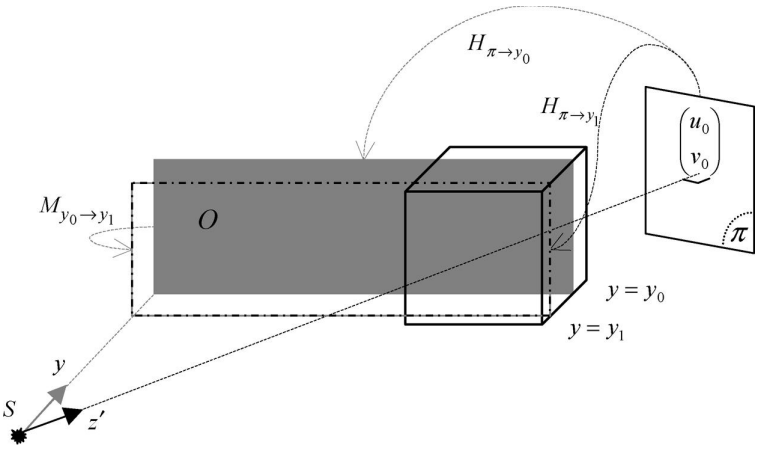



Figure 4: Rectified plane obtained by back-projection of projection plane $\pi$ onto a plane

As it can be seen, to ease the projection step it uses homography matrix as the process is very source-driven and using this kind of matrix can speed up the process. Although it is clear that the rectification method is not needed in this project (as we are in 2-D space), some aspects of this article have been applied in the methods that we will present later:

- In the Introduction of the article there is a mention of how interpolation is key in computational speed, stating that a Nearest Neighbour approach is faster than linear interpolation. In our case, we have applied bilinear interpolation that is slower than Nearest Neighbour approach but is more useful for our concrete problem.
- Also, in the subsection 'Cone-Beam Forward Projection' it presents a local magnification factor associated to each voxel, which has been adapted in our case to ponder how many times a pixel is crossed by a beam. This is necessary to get more accurate reconstructions, because not every pixel position is accessed by the same number of beams.
- Finally, in the 'Oversampling' subsection it states how to use the information of the pondering matrix to get the most accurate grey value for each pixel.

- **'Directional View Interpolation for Compensation of Sparse Angular Sampling in Cone-Beam CT'**: The main aim of this recent article [Bertram et al. (2009)] is to provide a way to generate artificially more views of an object without having to sample more angles. This is done because in beam geometry if the sampling is not balanced or the number of views is too small, aliasing artifacts can appear in the reconstructed image. To solve this problem without having to use more angles as initial sources, a directional interpolation method has been applied. This method 'uses' an intermediate view between two angles not done by making the simple average between the borders' view but using an interpolation method that takes into account the direction of

change in the grey value to predict the position of the object in the intermediate view. It differs from existing methods in that it is customized to the task of sinogram interpolation, taking into account that the structure to be projected only travels a limited distance between views. One key aspect in this work is the use of local orientation that is defined as the direction along which the grey level profile shows maximum average variation within the neighbourhood. As it can be seen in Figure 5, local orientation is represented by the calculation of a structure tensor (which provides information not only on the orientation of structures within a image, but quantifies the directional grey value change along each eigenvector of the structure tensor (the principal eigenvector $\mathbf{e}_1$ specifies the orientation with maximum grey value change).

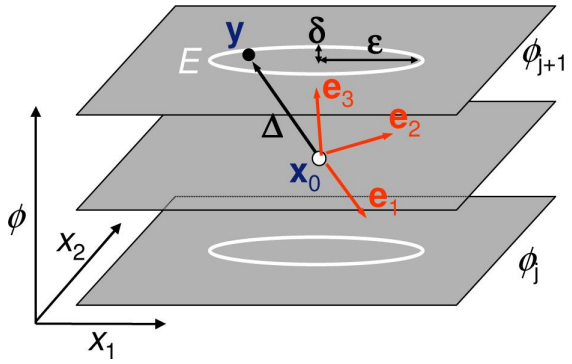



Figure 5: Schematic illustration of the geometry used for directional interpolation

In the figure above it can be see than for the interpolation of a sinogram point $\mathbf{x}_o$ (corresponding to detector pixel at a intermediate viewing angle), only such points $\mathbf{y}$ from adjacent projections are considered to be located within in-view regions E). It is necessary to apply interpolation in order to increase the sampling density of the data along the viewing angle axis. In order to preserve object structures (such as edges and ridges) and avoid blurring, interpolation is conducted in a parallel way. Another important aspect that has to be considered is the elliptic regions (labelled as E) where we will apply the algorithm to calculate the structure tensor. These regions have to be large enough to cover the movement of localized structures within adjacent views.

In the results section of the article it is mentioned that a strong increase in the interpolation error is related with decreasing angular sampling. It is also observed that the use of different form of interpolations (spline, cubic) increase the accuracy of the reconstruction, although the best results are achieved with directional interpolation. Another conclusion that can be extracted from the results is if the projections are extremely sparsely sampled, it is difficult to determine the orientation of the structures so the performance of the algorithm decreases greatly. But the use of interpolated projections help to reduce streaks caused by sparse angular sampling and also reduce the noise level. One of the drawbacks of this method is that the inclusion of interpolated views introduces slight object- and position-dependent blurring of sharp edges in the reconstructed image, which mainly affects objects positioned far from the center of rotation. As a final conclusion of this article, the authors indicate that the quality of the reconstructed image can be improved by combining two separate reconstructions: one

from only the original data and the other from both the original and interpolated views. This will avoid the blur effect that appears by the use of interpolated views so the final method takes the advantages of both trends but at the cost of a higher processing time.

In our case we have not implemented directional interpolation because the method is recent (July 2009) but we think that this approach is very interesting and can help us improving our results by means of getting better accuracy in the reconstruction with less number of views.

- **'A Fast Algorithm for Back-projection with Linear Interpolation'**: This article [Sahiner and Yagle (1993)] presents an algorithm that reduces the time the back-projection step takes by decreasing the number of multiplications in linear interpolation. Although in the article it is applied to a method more in the 'Filtered Back-projection' group, it presents a method to speed up linear interpolation by backprojecting four views at a time. The performance of this method is quite good but we have not implemented it because bilinear interpolation is needed in the projection and back-projection steps of our project, because beams not always cross round pixel positions.

### 2.2 Filtered Back-projection

If there is a group of techniques that have been mainly applied in recent years when reconstructing CT images from projections, is the one that is known as Filtered Back-projection. This groups of methods rely on mathematical transforms (Radon, Fourier) in order to get better results but using the same philosophy behind Cone-beam Projection methods. In both cases beams are emitted from source points and the differences are in the way the grey values of the pixels crossed by the beam are taken and later stored in the projection step. The main difference is in the Back-projection step where a filtering process and different interpolation techniques are applied in order to get rid of some of the drawbacks of the previous group of techniques. Here we present some of the articles that have been studied in order to gain knowledge about this group of methods:

- **'Fourier-Based Forward and Back-Projectors in Iterative Fan-Beam Tomographic Image Reconstruction'** : As it seems logical, one of the biggest concerns in CT reconstruction algorithms is the computational time . In this article [Zhang O'Connor and Fessler (2007)] Fourier-based forward and back-projection methods that can reduce computation are presented. The method proposed relies on Projection-slice Fourier Theorem but extended to fan-beam geometry. The steps that are followed in the projection phase are the following, summarized in Figure 6

  1. *Two-Dimensional Non-Uniform Fast Fourier Transform (NUFFT):* In this first step a 2-D NUFFT is used to evaluate the 2-D Fourier transform in polar coordinates. The input is a chosen patch of the image and the output is polar coordinates samples equally-spaced.
  2. *Approximating Detector Response:* This step is done to avoid the blur effect of the detectors, which are known to be dependant on the distances between each image pixel and the detector elements, and cannot be modelled in the frequency domain. In this article the authors approximate the depth-dependant detector response

by the effective beam-width at the rotation-center (calculated by multiplying the actual detector width by the ratio of the source-to-isocenter distance over the source-to-detector distance).

3. *One-Dimensional Non-Uniform Inverse FFT*
4. *One-Dimensional Shifts Using FFT-Based Interpolations:* This step spaces uniformly projection data along the radial coordinate.
5. *Use of Symmetry Properties:* Because the input image is real, its Fourier transform is Hermitian symmetric so only half of the Fourier samples need to be calculated in the first and third step. This approach reduces computation and ensures that the projection are all real.

   This approach is quite interesting because the use of Fourier transform really speeds up the process with no effect in performance. In the method that we will propose we have used FFT and IFFT in order to have a fast and accurate reconstruction.

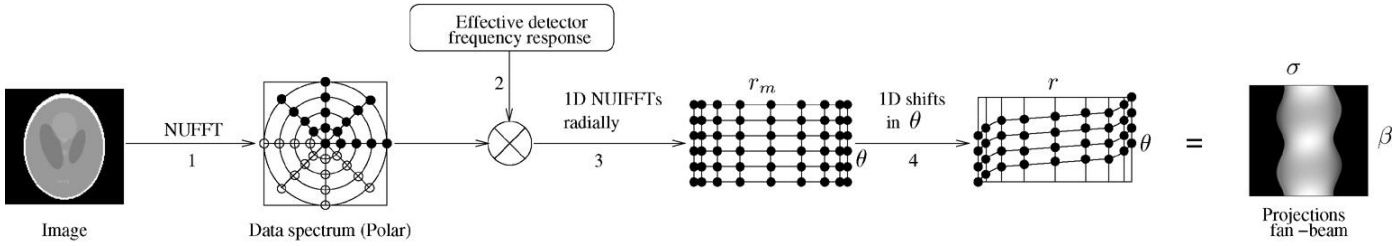



Figure 6: Basic steps of Non-Uniform Fast Fourier Transform in fan-beam CT

- **'Direct Fan-Beam Reconstruction Algorithm via Filtered Back-projection for Differential Phase-Contrast Computed Tomography':** Although the aim of this article [Zhihua Qi (2008)] is not directly related to our project, there is indeed an important aspect that is covered in the Introduction section that must be noticed. For a given X-ray detector with rectangular geometry, the fan angle is determined by the angle defined in this formula:

$$\gamma_m = 2tan^{-1}(\frac{H}{2D}) \quad (1)$$

where H is the detector's width and D the distance from the X-ray local spot to the detector. When D is very long, the divergence of the X-ray beam becomes insignificant so the beams are approximated as plane wave and parallel-beam image reconstruction is sufficient. But using the new phase retrieval method the distance D is significantly reduced. It is important to know that when a larger object is scanned with the same D, the fan angle is increased and the parallel approach becomes less accurate. When the fan angle increases more than $5^o$, the divergent nature of the data acquisition gets no longer negligible so it is necessary to provide a method that can accurately reconstruct CT images at any fan angle. So in our case we will have to place our source points in such a distance from the image that does not make appear divergence effects which may vary the reconstruction results.

- **'A High-Speed Radon Transform and Back-projection Processor':** This article [Eric Shieh and Agi (1990)] is referenced because it is one of the first chronologically that is concerned not only about the quality of the reconstruction but in the overall performance of the method (that includes accuracy or speed as the main factors) in the Radon transform domain.

- **'Combining Image Reconstruction And Image Analysis With An Application To Two-Dimensional Tomography':** Although the objective of this project is to develop a tool that let us reconstruct accurately an image from a series of projections, it is clear that one of the next steps can be the analysis of the image reconstructed in order to find or detect anything that we want (defects, shapes, etc...). In this article [Louis (2008)] the author proposes a method that combines both the reconstruction and analysis steps in order to get as a final result the image both reconstructed and analysed. In this field there have been many attempts to define a working procedure, as the $\Lambda$ computerized tomography or Tikhonov-Phillips regularization. Without entering in many details, because the objective of this article exceeds the final purpose of this project, we will explain briefly the basic mathematical roots of the procedure.

  We have to take into account than in computed tomography the images are produced by applying reconstruction algorithms to the measured data, calculating images which are smoothed versions of the original density distributions. The result can be presented as:

  $$f_\gamma = f * e_\gamma = E_\gamma f \quad (2)$$

  where $f$ is the original object and $e_\gamma$ is a mollifier (like a smoother) depending on the reconstruction method. For example, if we want to do an edge detection algorithm, the next step is to calculate smoothed versions of the image.

  $$f_{\gamma\beta k} = \frac{\partial}{\partial x_k}(G_\beta * f_\gamma) = \frac{\partial}{\partial x_k}(W_\beta f_\gamma) = \frac{\partial}{\partial x_k}(W_\beta E_\gamma f) \quad (3)$$

  where $G_\beta$ represents a mollifier (for example, a Gaussian kernel, that smooths the image) and $\beta$ and $\gamma$ are chosen independently. The main concept behind is that by changing some operations in the reconstruction we can do two steps at a time. The problem behind this approach is that our reconstruction method may be analysis-specific, that is, it will not reconstruct in the same level of accuracy all the images but apply the reconstruction and analysis step at a time. This article will be interesting to study as a first approach to some of the future lines of this project.

- **'Filter design for Filtered Back-projection guided by the interpolation model':** One of the most interesting articles related with the filtering and interpolation steps that have to be applied in Filtered Back-projection algorithms is this one [Horbelt et al. (2002)]. In this case this article is focused on designing a filtering operator having in mind the interpolation model that is applied to the sinogram, combining the ramp filtering and the spline fitting into one single operation.
  The basis of the Spline-based Filtered Back-projection can be summarized in three steps:

1. Filter the sinogram in the Fourier domain with the ideal ramp filter
2. Fit the sinogram with a model that is represented as a linear combination of shifted basis functions (Splines)
3. Calculate the back-projection at the pixel location.

Once this basic schema is presented, the authors propose three different ramp filter implementations. The first approach uses interpolation combining the ramp and direct spline filter into one single FFT filtering operation, while the second one implements an oblique projection of the sinogram into a spline space (it performs better because takes into account the band-limited nature of the sinogram). The last option uses fractional splines to discretize, which produces a spline of reduced degree greatly coherent with the underlying model. In Figure 7 we can see how great the reconstruction is using the three types of processes.

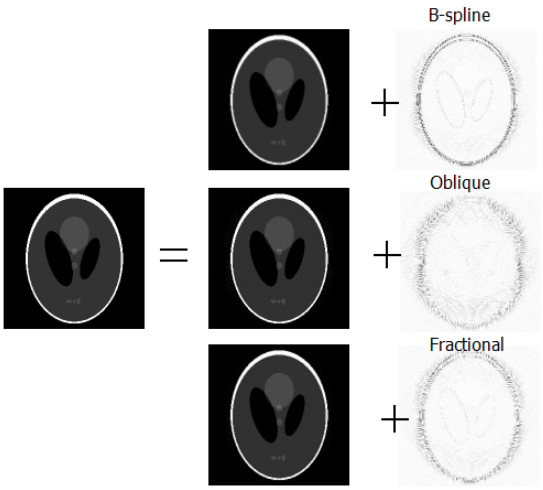



Figure 7: Results of Spline-filtered Back-projection with linear interpolation model and the three different ramp filters

As it can be seen, the method that performs better is the last one because the error image is the less visible of the three (but the oblique approach seems to recover better the inside shape). In our implementation of Filtered Back-projection we have used spline interpolation in order to improve the results, as its adapts better to the shape that we try to recover by dividing the curve into segments instead of trying to fit the curve with a single line.

## 3. Projection and Back-projection Techniques

After discussing the most interesting articles related to the different approaches that are taken to solve our problem, in this section theoretical aspects of the two main alternatives, Euclidean Geometry (Cone-Beam Approach) and Filtered Back-projection, will be discussed starting from the roots of each method and discussing the two final implementations of the algorithms.

### 3.1 Euclidean Geometry Approach

As it has been stated in the Introduction, the process of reconstructing images from CT scans will rely on the use of two different but strongly related: projection and back-projection. Both methods will be fully studied and presented through this section along with graphs and diagrams that may lead to a better understanding of the steps that have been followed.

#### 3.1.1 Projection

**Basic Procedure**

In the framework of this project, projection is defined as the process in which rays are emitted from a certain number of source points. There are, for every source point, as many rays as pixels in the image. The accumulative value of intensity of every pixel that is crossed by a ray is stored in a matrix that will be used as input in the later back-projection process.
The algorithm implemented is the following one:

**Algorithm 1**: Algorithm of the general process

**Input**: *steps* = angular offset between the initial source points, image to reconstruct
**Output**: *results* = projection data, *incidences* = incidence matrix
**begin**
*Define the source points and the opposites of the source points*
*For each source point*
*Calculate the tangent line in the opposite of the source point*
For every pixel of the image
*Calculate the equation of the line between the source point and the pixel*
*Calculate the accumulated grey value:*
*If the pixel position is round, get the exact grey value then add 1 to this pixel position in the incidence matrix*
*else interpolate to get the pondered grey value and add the ponder factor to the incidence matrix in the pixel´s position*
*Store the accumulated grey value in results(position of the pixel,source point)*
`return(results,incidence)`
**end**

The algorithm places, around the image that we want to reconstruct, source points that are equally spaced (with the value the parameter steps indicates) and will play the same part than X-ray emitters. For every source point the tangent line between it and its opposite is generated. This line will receive the rays emitted from it and with destination every pixel of the image. This process can be better seen in Figure 8.

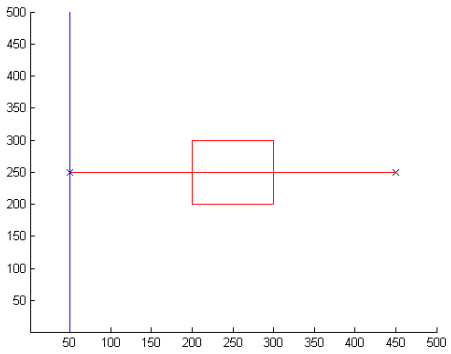



Figure 8: Tangent line that joins the source point with its opposite

Once we have the line, the next step consists on calculating the accumulated grey value that will be stored in the results matrix and will be used in the back-projection process. To do this we simulate the emission of a ray by generating a line between the source point and every pixel position. The value that is going to be stored will contain information of every pixel that is crossed by the line, as it can be seen in Figure 9. We use bilinear interpolation because the ray will not always cross a round pixel position: if the ray touches a whole pixel position value (i.e $(25, 25)$ we take the exact intensity value. But if we touch a pixel in a not whole position value, (i.e $(23.4, 38.5)$) we take the value calculated by a combination of the intensity values of positions $(23, 38), (24, 38), (23, 39)$ y $(24, 39)$ by bilinear interpolation [Sources (2009a)]. The value that is going to be stored is divided by the number of positions of the image that the ray touches.

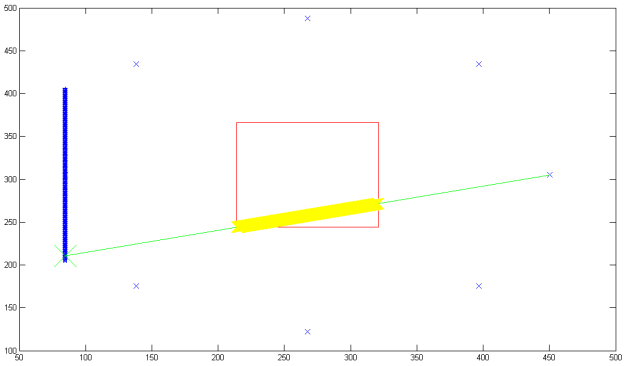


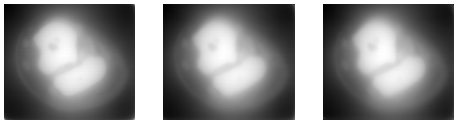
Figure 9: Calculation of the accumulated grey value

Along with storing the accumulated grey value we will keep in another matrix, called incidence matrix, the number of times each pixel's grey value has been taken. This is done to get more accurate reconstruction results, because the pixels in the center of the image will be crossed by more rays so we need to ponder the value to get the correct final value. The results matrix has as many rows as pixels the image has and as many columns as source points.

### Variations to the Basic Procedure

Once the basic procedure has been explained we will explains variations that have been applied in order to optimize the performance of the method.

- **Different Ray Width:** In the basic procedure, we have considered rays of width one but it does not have necessarily to be always like this. The program also accepts as an input parameter the width desired for the ray. It has to be noticed that the result will be assigned in the same place of the results matrix than in case of width one. An example of implementation on ray emission with different value of ray width can be seen in Figure 10



(a) Width 1 (b) Width 2 (c) Width 3

Figure 10: Different width of rays

- **Ray Emission:** Until now, we have assumed that for every source position we only consider one source point. But, in order to explore other possibilities it has also been used a paralell ray emission method based on linear cameras [Sources (2009b)]. This type of cameras emits paralell rays perpendicularly to the device, as it can be seen in Figure 11.

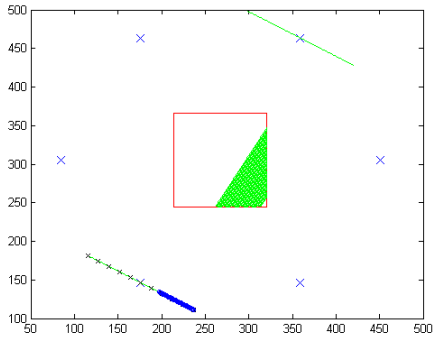



Figure 11: Linear Camera Simulation Example

- **Use of Slots:** As it can be observed, if we want to reconstruct a large image and use many source points, the size of the results matrix can represent a great amount of memory which can cause a decrease in the performance of the algorithm. We have to take into account that we do not have an unlimited number of sensors to receive the rays (and store the accumulated grey value). One way to solve this problem is the use of slots.
  The implementation of the projection admits the use of slots so the accumulative grey

value that is going to be stored is assigned to the position of the results matrix associated to the label of the ray. This label is given depending on the intersection of the ray with the tangent line, using the position of the intersection between the line between the source point and the pixel and the tangent line $(x_cut, y_cut)$. The process may be better understood by observing Figure 12

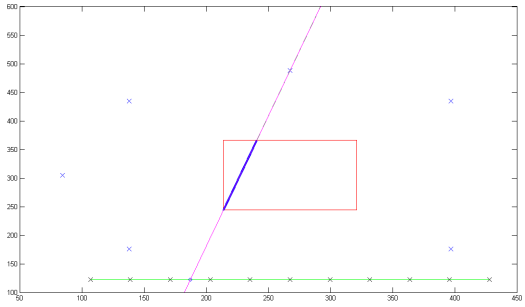


Figure 12: Use of Slots

**Implementation of the algorithm**

In order to make the application more customizable, the algorithm accepts the next input parameters:

- steps: this variable is the responsible to adjust the angular offset between the different ray source points.
- slots: number of slots.
- option: if this variable takes value 1 the program will use bilinear interpolation [Sources (2009a)] to calculate the accumulative grey value. In case it takes value 2 it will only use the destination's grey value as the value to accumulate.
- type: this variable let us switch between the different types of ray emission ('conic' or 'linear').
- rank: range of original source points (only used in linear camera simulations).
- width: width of the ray.

The program gives an output the results and incidence matrix and, if we use slots, the touch matrix and a matrix that contains the different ranks that cover each slot. It also returns a variable that contains the data of the experiment which will be used to ease the back-projection process.

### 3.1.2 Back-projection

As it has been mentioned in previous sections of this document, back-projection [Feng-rong Sun (2008); Sources (2003)] is projection's inverse process. While the objective of the projection is to get data from the image, in the back-projection process the objective is to get

the image from the data that has been calculated in the projection process. So the back-projection process takes as input the results matrix that the projection process returns and also all the data associated to the projection process that may be useful to complete the process. Depending on the type of simulation, back-projection process has some particularities that will be detailed next:

### Conic Projection

- **Without Slots:** In this case back-projection consists on calculating, for each pixel, the mean of the accumulated grey values that have been stored in the row of the results matrix corresponding to each pixel. This process can be better understood looking at Figure 13.

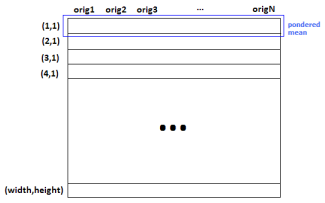



Figure 13: How to calculate the final grey value by using the mean of projections

In this case the incidence matrix is used to get a result that takes into account the number of times a pixel has been touched by rays.

- **With Slots:** In order to complete the back-projection process we have to follow, for every source point and pixel, the next procedure. First, we have to calculate the intersection point of the line formed by each pair (initial position, end position) and the tangent line in the opposite of the source point. Once we have the coordinates of the intersection point and using the information from rank variable we will take the accumulated grey value by doing linear interpolation between the grey value assigned to the borders of the slot the ray has been assigned to. This can be seen graphically in Figure 14

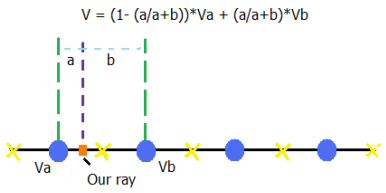



Figure 14: Use of linear interpolation to get the correct grey value

It has to be taken into account that the grey value that has to be taken needs to be weighted by the number of rays that have fallen in this concrete slot, by using the touch matrix.

#### Projection by Paralell Ray Emission

A process similar to back-projection with slots must be followed to reconstruct an image projected by a simulation of paralell emission such as Linear Camera[Sources (2009b)] devices. The difference consists on instead of emitting rays for every pair (source point, image pixel), it only have to be emitted as rays as the linear camera originally emitted. The rest of the process is the same as in back-projection with slots, including steps like label assignment or calculation of the grey value by using linear interpolation.
The grey value that will be assigned to each ray will be associated to every pixel that is touched by the original ray. Inverse Bilinear Interpolation will be used to adjust the value that each pixel will get, as it can be seen in Figure 15

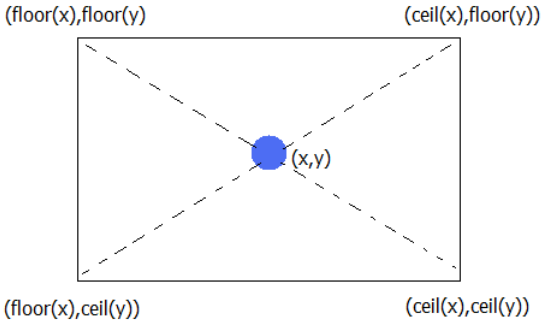



Figure 15: Inverse Bilinear Interpolation

### 3.2 Filtered Back-projection Approach

As it has been said in the Introduction many of the Filtered Back-projection methods are based on the use of the Radon transform. So, in this section this transform and the properties that are interesting to our problem will be studied along with some other aspects (type of filters or interpolation) that are implemented in the final solution.

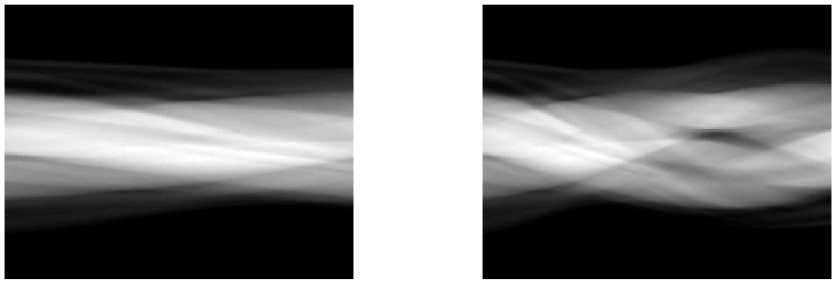
#### 3.2.1 Radon Transform

#### Basic Concepts

Bi dimensional Radon Transform is an integral transformation of a function along a group of lines. As an example, if a line is represented by the equation $xcos\theta + ysin\theta = s$ where $s$ is the minimum distance between the line and the origin and $\theta$ is the angle that the line makes with the x axis, the Radon transform is equal to:

$$R[f](\theta, s) = \int_{-\infty}^{+\infty} \delta(xcos\theta + ysin\theta - s)\, dxdy \quad (4)$$

In a $n$-dimensional space, the Radon transform is the integral of a function on hyperplanes. The integral of a function along a group of lines in the $n$-dimensional space is also known as X-ray transform. In the context of CT scans where we are, Radon transform is also called sinogram because the Radon transform of a delta function has sinusoidal shape. So the graphical representation of the Radon transform of a series of small objects appears as a group of sines with different phases and amplitudes, as it can be seen in Figure 16



(a) 79 (b) 179

Figure 16: Radon transform of an image considering different number of maximum angles

**Projection-slice theorem of the Fourier transform**

As it has been stated, Radon and Fourier transform are strongly related. In this section we will explain how using Fourier transform the process of inverting the Radon transform becomes easier. Fourier transform in one variable domain is equal to:

$$\widehat{f}(w) = \frac{1}{(2\pi)^{\frac{1}{2}}} \int f(x) e^{-ixw}\, dx \quad (5)$$

And, more related to our problem, bi dimensional Fourier transform of $\mathbf{x} = (x, y)$:

$$\widehat{f}(\mathbf{w}) = \frac{1}{(2\pi)} \int f(\mathbf{x}) e^{-i\mathbf{xw}}\, dxdy \quad (6)$$

We will use the next notation:

$$R_\theta [f](s) = R[f](s, \theta) \quad (7)$$

because we will do the Fourier transform of variable $s$. Projection-slice theorem is formulated as:

$$\widehat{R_\theta [f](\sigma)} = \sqrt{2\pi} \widehat{f}(\sigma \mathbf{n}(\theta)) \quad (8)$$

where $\mathbf{n}(\theta) = (cos\theta, sin\theta)$.
By using this property we have an explicit way to invert the Radon transform (and to study whether it is possible or not to invert it). But this method is way too complex and the operations involved can consume a lot of time. In the next section we will discuss another way to take advantage of the Fourier transform to invert the Radon transform.

### 3.2.2 Filtered Back-projection

A computational efficient algorithm in the bi-dimensional domain is the one that is used in this project, Filtered Back-projection. If we take the adjoint operator of R (Radon transform) the equation now takes this value:

$$R^* \widehat{g}(x) = \int_{\theta=0}^{2\pi} g(\theta, \mathbf{n}(\theta)x)\, d\theta \quad (9)$$

This operator is also known as 'backprojector' because it takes the projections along the different lines 'smearing' or projecting them back over the line to produce an image. This operator by no way is the inverse Radon transform.
Now we define the ramp filter $h$ for one variable:

$$\widehat{H[h](w)} = [w]\,\widehat{h}(w) \quad (10)$$

If we apply now the Projection-slice theorem changing the integration variables, we can observe that for $f$, a two variables function, and $g = R[f]$

$$f = \frac{1}{4\pi} R^* H[g] \quad (11)$$

All this means that the original image $f$ can be reconstructed from the sinogram $g$ by applying a ramp filter (over $s$ variable) and then backprojecting. Considering that the filtering step can be implemented in an efficient way and taking into account that the back-projection consists only in accumulating values in the pixels of the image, the final algorithm is efficient enough.

### 3.2.3 Implementation of the Filtered Back-projection algorithm

The solution that has been implemented is based in the use of Matlab [Sources (2009c)] functions *radon* and *iradon* to complete the Filtered Back-projection process. The original functions have been integrated and extended into a final application so the program accepts any kind of image (black and white or color images) and a series of parameters to tune the process and returns the reconstructed image. The different input parameters and the values they may take are detailed next:

#### Input Parameters

The parameters which values can be adjusted are the following ones:

- **File:** File that the user wants to backproject.
- **Angles:** Slider that lets the user vary the number of angular positions (taking as starting point the 0) that act as source of the different rays.
- **Steps:** It is a slider that lets the user adjust the value of a scalar in the range [1,20] that modifies the separation between the initial angles. The default value is 1, the higher the value is, more separated will be the angles (i.e if we have 179 initial angles and the value of variable pasos is 10, the initial angles will be placed in 0, 10, 20 and so on).
- **Interpolation:** Type of interpolation that will be used in the Back-projection process. It can take any of the next values:
  - **'nearest':** Interpolation using the 'Nearest Neighbour' paradigm, taking the value of the nearer position.
  - **'linear':** Linear Interpolation.
  - **'spline':** Form of interpolation where the interpolant is a special type of piecewise polynomial called a spline (that is a polynom that tries to fit each division of the curve).

  - **'pchip':** Shape-preserving piecewise cubic interpolation.
  - **'cubic':** Simple cubic interpolation.
  - **'v5cubic':** Simple cubic interpolation from MATLAB 5 that does not extrapolate data and uses spline method if X axis is not equally divided.

- **Filter:** Type of filter used in the windowing process:

  - **'Ram-Lak':** Ramp filter used by default. As it is very noise sensitive is the least used of the chosen ones. The rest of filters are a combination of this one and windows that multiplied by the Ram-Lak filter decrease the effect of high frequencies.
  - **'Shepp-Logan':** Multiplies ramp filter by a sinc function.
  - **'Cosine':** Multiplies the basic ramp filter by a cosine function.
  - **'Hamming':** In this case the basic ramp filter is multiplied by a Hamming window.
  - **'Hanning':** The basic ramp filter is multiplied by a Hanning window.
  - **'None':** No filter is applied.

  The frequency response of all the filters presented can be observed in Figure 17:

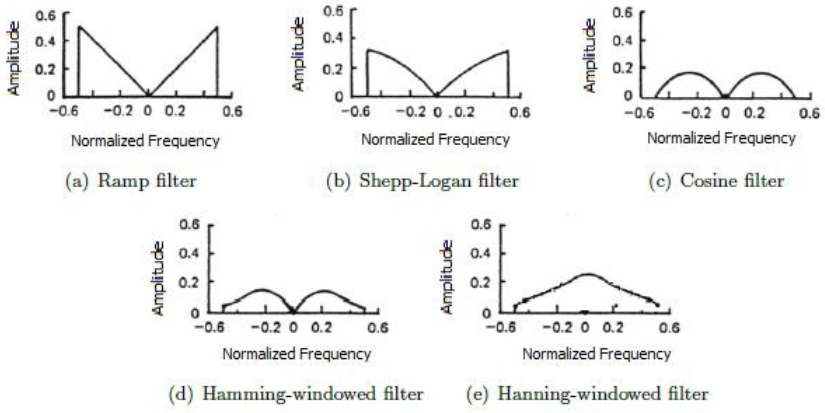



Figure 17: Frequency response of the filters presented

- **Frequency Compression:** It is a scalar in the range (0,1] that modifies the filter by rescaling its frequency axis. The default value is 1, if the value is less than 1, the filter is compressed to fit into the frequency range [0,value], in normalized frequencies.

## 4. Results

After detailing the two different processes and presenting the two implementations of the solution, in this section we present some results of reconstruction. First we present the results separated by each approach, analysing the pros and cons of each solution. Finally we compare both methods in order to decide which one achieves better the objective of the project,that is, reconstructing accurately a bi-dimensional image.

### 4.1 Euclidean Geometry Approach

Once discussed the theoretical aspects of the project related to the techniques implemented, it is time to show the results of several experiments in order to find out how well the theory has been put into practice. The results will be analysed in depth, not only considering the accuracy of the reconstruction but other topics as running time or use of memory. The original image that has been used in the simulations is the one shown in Figure 18

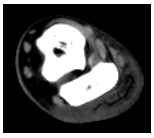


Figure 18: Image to reconstruct

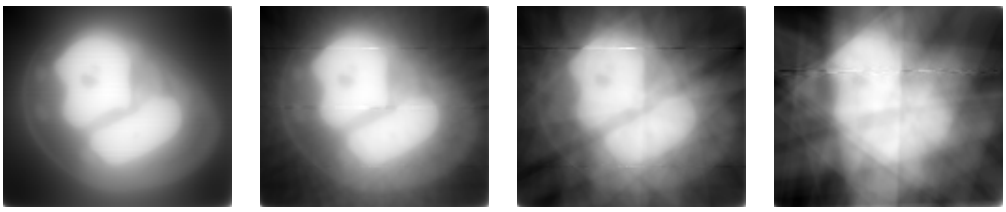
#### 4.1.1 Some Results

**Conic Projection**

#### 4.1.2 Without Slots

- Rays of width 1

  Results can be seen in Figure 19

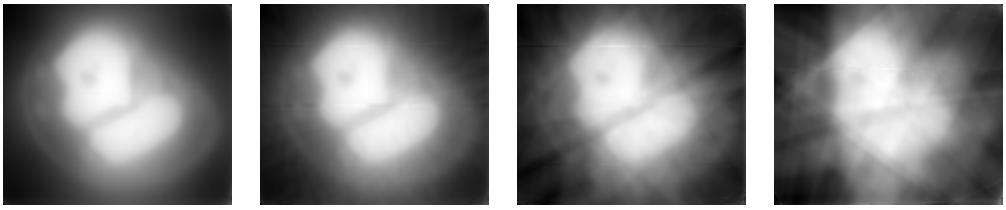


(a) 360 views (b) 36 views (c) 18 views (d) 8 views

Figure 19: Results for several number of views

- Rays of different width

  In this case 3 has been chosen as the width. Some results can be observed in Figure 20

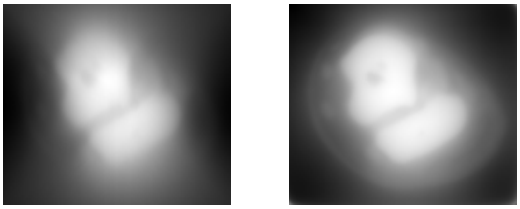


(a) 360 views (b) 36 views (c) 18 views (d) 8 views

Figure 20: Results for several number of views

### 4.1.3 With Slots

Some results using different values of the slot parameter can be seen in Figure 21



(a) Few slots (b) Many slots

Figure 21: Results with Slots

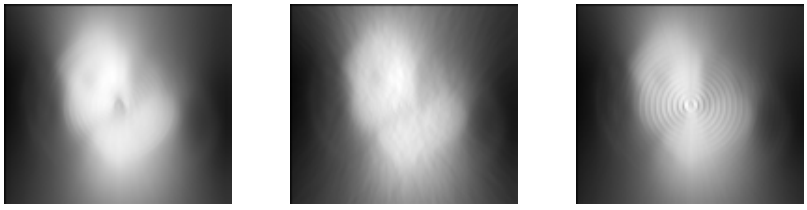
### 4.1.4 Analysis of the Results

By an observation of the results achieved by using a conic ray projection, the following conclusions can be extracted:

- In no case the reconstruction is perfect. But, the error is not in shape but in accuracy of grey value. That means the object can be well distinguished but the reconstructed image is not the same as the original one.
- Results get better the less angular offset we consider (the more views we take, the better results). From 20 $^{o}$ and on, although the shape of the object can be recognized, it is difficult to appreciate concrete details.
- Comparing results using or not slots it can be seen that there is no big difference. This is not bad at all, because the use of slots lead to less use of memory with no significant loss in the reconstruction (with 150 slots the quality of the reconstruction is fair good).

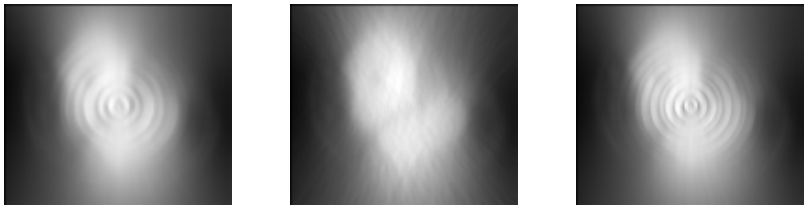
#### Parallel Ray Emission Simulation

Results obtained by simulating a paralell ray emission like what linear cameras do with two different values of the slot variable can be observed in Figure 22 and in Figure 23.



(a) rank 110 (b) rank 160 (c) rank 210

Figure 22: Results with paralell emission (Slots = 160)



(a) rank 170 (b) rank 220 (c) rank 270

Figure 23: Results with paralell emission (Slots = 220)

In this case it can be concluded that the quality of the reconstruction is worse than when we use conic projection. Some reasons behind this fact are the following:

- By simulating a linear camera (by emitting the rays in paralell) we are not taking for every source point information of every image pixel. Instead we only take information of these points that are touched by the rays that are emitted for every source point (which have as origin a point in the tangent line centered in the source point. So we have a concrete loss of information).
- Results do get better (by means of shape reconstruction) as we increase the number of slots. The reconstruction is better if the number of slots and source points are similar.
- An special effect can be observed in some results, simulating an ellipse in the center of the image. This effect is known as Moiré Effect [Sources (2009d)], which consists in the appearance of artifacts because of a difference between the framing of the picture and camera's sensor resolution. it is an interference produced by difference between image's and sensor's frequency.

**Discussion of the Results**

Once results from several simulations have been shown it is necessary to discuss them in depth.

### 4.1.5 Comparison Between the Different Techniques

Two different methods (conic and linear/paralell emission) have been used to simulate the emission of rays. The difference between them relies mainly in the amount of the information

of the original image that is used to calculate the results of the projection.
The implemented conic projection covers a lot of possible cases at the expense of a long running time and high memory cost. Paralell emission simulations, although worse by means of reconstruction quality, take much less time.
In terms of accuracy of the reconstruction the results, as it can be observed in Figure 24, are far from perfect. It has been mentioned before that the results are good in terms of shape reconstruction but not in accuracy of grey value.

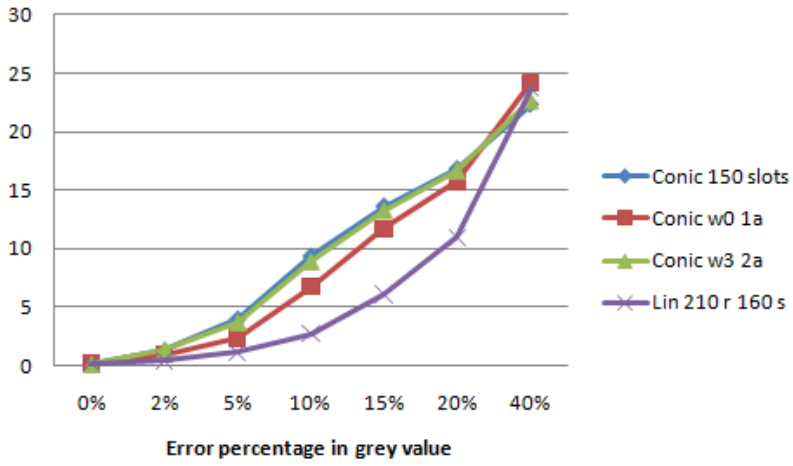



Figure 24: Reconstruction Accuracy

The accuracy results (in terms of perfect reconstruction) are not good. One of the main causes is that when we calculate the accumulative grey value for each pixel we do not only consider this pixel´s grey value but every other one that is touched by the line that simulates the ray. The best results are achieved (considering a five per-cent error) by using conic projection with slots and conic projection with rays of width 3 but without slots (and, in this case, by storing in the results matrix not the accumulative grey value but the mean accumulative grey value).

### 4.1.6 Some Other Things to Consider

There are some other topics that are relevant to understand the results that have been achieved.

- **Running Time:** One of the drawbacks of the several methods implemented is the amount of time it takes to deliver results.
  - In the case of conic projection with angular offset of one degree the program takes 28 hours to give results. Although it is clearly a long time, it has to be considered that it does not only provides results for $1^o$ separation but for every integer possible separation between 1 and 360 in only one simulation. Using slots in this case only increases a little the time it takes (due to labels computation). Back-projection processes are almost instantaneous with no slots and lasts around 20 minutes when we use slots.

  - In linear camera (paralell emission) simulation the projection process takes less time (around 4 hours to do both the projection and back-projection). The main reason is that we emit less rays (less equations and interpolations) which eases the process. Considering that the quality of the reconstruction (in shape) is not too bad, it may be a method to consider.

- **Memory Constraints:** On deciding which technique to use it has to be considered not only the quality of the reconstruction but if this reconstruction is feasible. That means, in conic projection the program returns a 13054x360 matrix as result (with separation of 1 degree) while with 150 slots the size of the results matrix is 150x360. The use of memory in the first case is much higher than in the second one and it can stop the program or even freeze the system. It also has to be said that to do the test it has been used only the central part of the image (size 122x107) because the tests made with the original image takes even more time.

So, taking into account all these topics and analysing the results, it seems that the better option will be to use conic projection with slots (with a high number of slots) or use parallel emission instead of fan-shape projections. The first option leads to better accuracy in the reconstruction but at the expense of higher running time and use of memory. By using paralell projections we lose some accuracy in the results but we get results sooner. Anyway, the running time considerations have to be taken into the context where they belong, because the tests have been made with everyday computers. Maybe using machines with higher processing power the running time of the processes will decrease dramatically and use less memory. Still the code can be improved so it is possible that some of the drawbacks can be overcome.

### 4.2 Filtered Back-projection Approach

In this subsection results from several tests (under different conditions) are shown. The results are separated by the parameter we want to test in each case (type of interpolation, filter used or frequency compression). The original image to recover is again shown in Figure 25

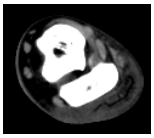
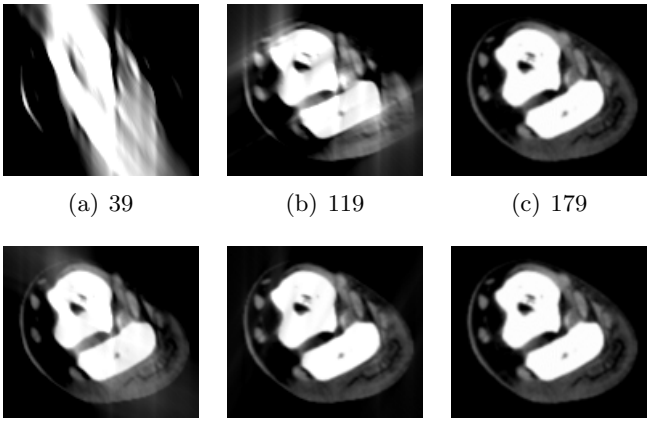


Figure 25: Original Image

#### 4.2.1 Variation of the Angle Value

In Figure 26 it can be seen how the reconstruction is affected by the value of the maximum turn angle that is considered.



(a) 39 (b) 119 (c) 179

(d) 249 (e) 329 (f) 359

Figure 26: Results for different number of initial angles

As it can be observed, the reconstruction of the image is fair good for a number of initial angles equal or higher than 179. it is also true that if the number of angles is higher than 179 the system performance is not as good than with 179. This happens because we are throwing projections from angles that are opposite to ones that we have used before (i.e 189 will cover the same portion of the image that 9 originally covered) so in this case we have a superposition of information that may lead to some errors in the final reconstruction. It can be seen that it we choose 359 as the number of initial angles, the result is the same as with 179 because in this case the superposition's compensate each other.

As it can be seen in the Figure 27 with few initial angles the exact accuracy gives us better results than the euclidean geometry based method and if we use 179 or 359 as values of the variable the exact accuracy is higher than 50 % meaning than more than half the pixels of the reconstructed image have the exact same grey value than the original image.

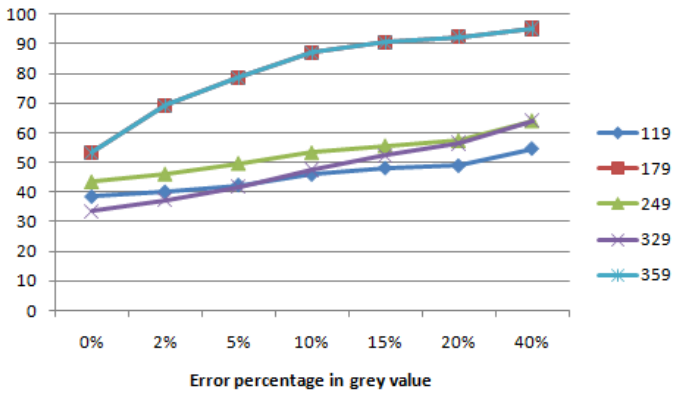
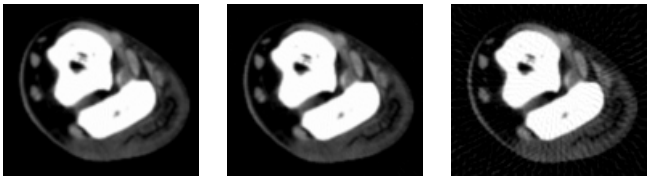


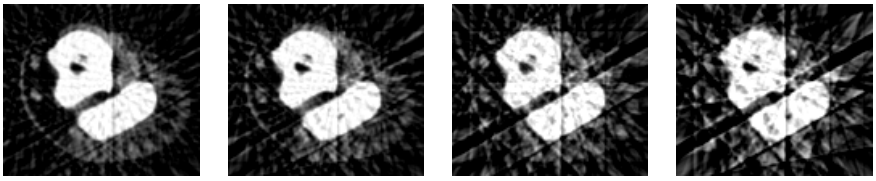

Figure 27: Reconstruction Accuracy

### 4.2.2 Variation of the number of views

We have to take into account what is the final objective of the project, that is, simulate the functioning of the CT scanners but with cameras. If we take a look at the results that have been just presented, we need to take up to 179$^o$ (and one take for each angle) in order to get the reconstruction done. But maybe if we separate the cameras a little bit (take less intermediate views), we can get good enough reconstruction results with less cameras. In Figure 28 we show how the reconstruction result is affected by the number of degrees of separation between angles (we have chosen 179 as the number of initial angles).



(a) 180 views (b) 90 views (c) 45 views



(d) 23 views (e) 18 views (f) 12 views (g) 9 views

Figure 28: Results for several values of separation between initial angles

If we observe the figure above it is obvious that there is difference in the reconstruction as we increase the separation between angles. But if we take a look at Figure 29 we can prove this fact with concrete data:

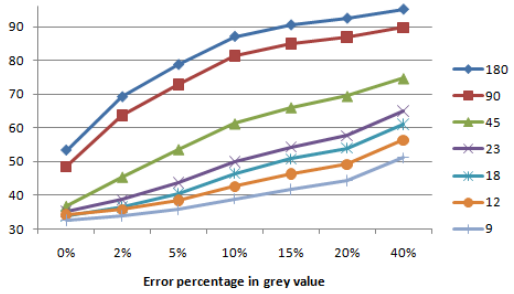


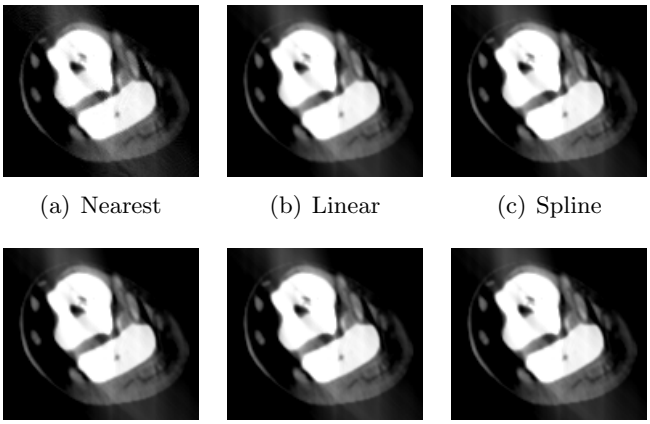

Figure 29: Reconstruction Accuracy

Although we lose some precision in the reconstruction, if we take half the initial angles the loss of exact accuracy is less than 2 % and even taking one fourth of the initial position the reconstruction is fair good in terms of shape and relatively good in terms of accuracy in grey value. It can be seen that the more error percentage that we admit in the reconstruction

most of the times the results are better if we take half the views, but it is due to the error tolerance that we are considering.

### 4.2.3 Variation of the Interpolation Form

In Figure 30 results obtained by varying the interpolation form (using an intermediate initial angles value (229)) are shown:



(a) Nearest (b) Linear (c) Spline

(d) Pchip (e) Cubic (f) v5Cubic

Figure 30: Results using several forms of interpolation

In this case is very difficult to distinguish by a glance which form of interpolation performs better so in order to have enough information to decide we show in Figure 31 the accuracy percentages in the reconstruction:

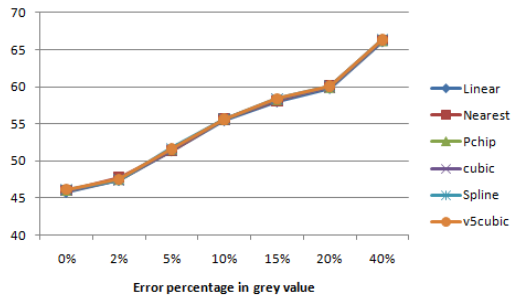
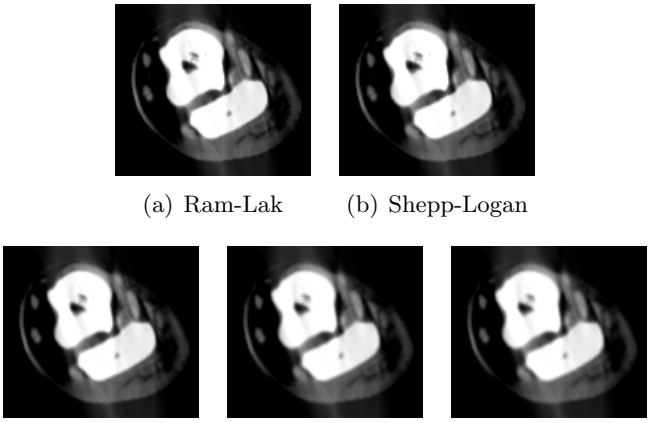



Figure 31: Reconstruction Accuracy

By observing the results in the table we can see that it is difficult to see the difference in the reconstruction results whether we choose one form of interpolation or another. In terms of exact accuracy the form of interpolation that performs better is the spline interpolation (we get the higher accuracy percentage, 45.46) so it will be the option to choose if we want to achieve slightly better results.

### 4.2.4 Effect of the Variation of the Filter used

Results obtained by varying the type of filter applied (using a random number of initial angles (209) and spline interpolation) are shown in Figure 32:



(a) Ram-Lak (b) Shepp-Logan

(c) Cosine (d) Hamming (e) Hanning

Figure 32: Results obtained varying the type of filter

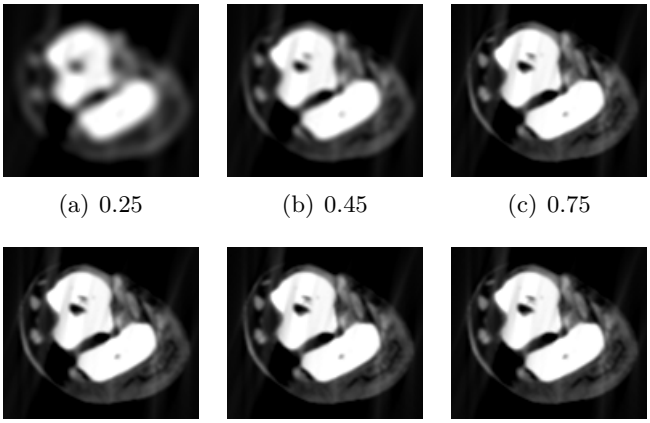
Again it is not clear which type of filter gives us the better results so it is necessary to observe concrete data (which is represented in Figure 33) in order to decide which filter type we have to use.


Figure 33: Reconstruction Accuracy

The results vary a little between the different types of filters but, again, the variation is not high enough to discard any type of filter. In this case the better results are obtained surprisingly with the simplest Ram-Lak filter (accuracy value of 48,39), followed closely by the Shepp-Logan filter (48,27), which is the one that we will use in the tests because it decreases the effect of higher frequencies. The effect the filters have in the performance may not be clearly seen in this example, because the image that we are working with have no noise to filter.

### 4.2.5 Variation of the Frequency Compression

Now it is time to analyse the effect of frequency compression in the reconstruction. Results obtained by using several values of the frequency compression parameter can be observed in Figure 34



(a) 0.25 (b) 0.45 (c) 0.75

(d) 0.85 (e) 0.95 (f) 0.99

Figure 34: Results for several values of frequency compression

This time clear differences can be observed, but in the Figure 35 they can be seen better:

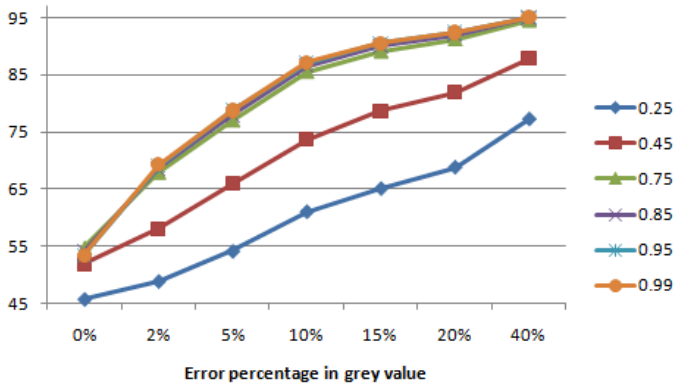



Figure 35: Reconstruction Accuracy

In this case there is indeed difference in the quality of the reconstruction related with the frequency compression chosen. The higher the value of the frequency compression is, the better the reconstruction because we are not eliminating necessary frequencial components.

### 4.2.6 Discussion of the Results

The main facts that can be extracted by taking a glance at the results are the following:

- In every case the reconstruction is better than the one achieved with the euclidean geometry approach. The reason behind this fact is the use of a filtering and interpolation process after the Back-projection.

- The quality of the final reconstruction depends on the number of initial angles that are considered. The results for 179 and 359 are the same because there is redundancy related to having the angles and their opposites as source points.

- The form of interpolation used affects slightly the reconstruction, but the better results are achieved by using spline interpolation.

- One of the key points in the improvement of the results is the use of filters. Filters dim the effect of high frequencies (to the point of deleting very high frequencies) and help in decreasing the blur effect that can be appreciated in the results using the Euclidean approach. Best results are achieved using ramp filter windowed by a sinc function, although the results do not vary a lot if we change the type of filter.

- Another factor that has to be considered is the frequencial compression. The results improve the less we compress the frequencial axis, because we are not deleting any frequencial contribution.

- So, as a kind of first approximation, if we want to get the best reconstruction possible we have to have 179 or 359 initial angles, spline interpolation, sinc windowed ramp filtered and no compression of the frequencial axis.

- Taking into account the framework of the project it has also been studied the effect of separate the initial angles. The effect of a slight separation (2 degrees) has little impact in the overall reconstruction and can lead to save resources.

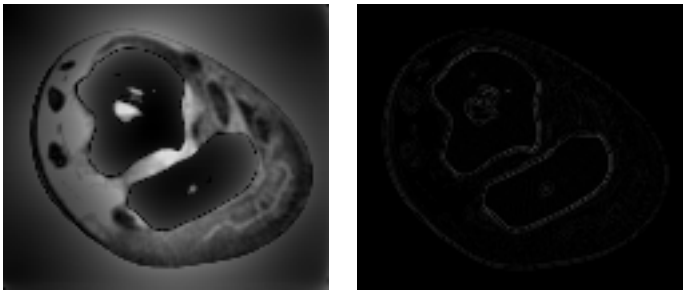
### 4.3 Comparison of the two methods

Once we have presented the results for both methods separately, it seems clear that the Filtered Back-projection outperforms the euclidean approach in both shape reconstruction and accuracy. In this subsection we will compare each method related to several topics such as accuracy of the reconstruction or memory cost.

#### Accuracy in the Reconstruction

It is obvious that the main factor to decide between a method or another is the accuracy in the reconstruction. In this case we have a clear winner, Filtered Back-projection delivers a much more accurate reconstruction, not in terms of shape recovery (where euclidean approach does perform well) but in terms of accuracy in grey value. Notice that here we are comparing our best results with every method.

The difference in the accuracy in the reconstruction can be clearly seen in Figure 36, where is shown the difference between the original image and the reconstructed one. The parts of the image that are black correspond with zones where there is no difference between the original and reconstructed image.



(a) Euclidean Geometry (b) Filtered Back-projection

Figure 36: Difference between original and reconstructed image

As it can be seen in the Figure above, Filtered Back-projection let us recover an image that is very similar to the original one (the difference image is almost black) while the euclidean geometry approach does recover the shape but not the grey value at all. In order to have more concrete data to compare let us take a look at Figure 37:

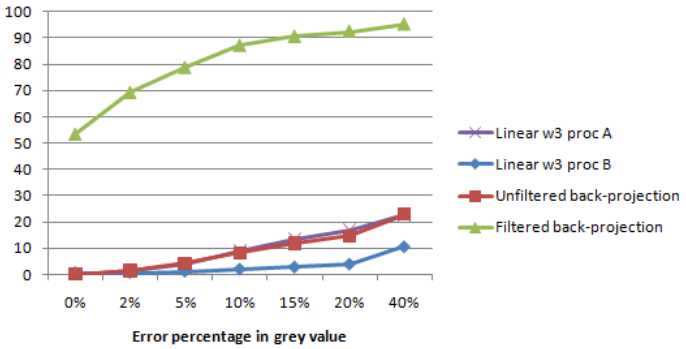



Figure 37: Reconstruction Accuracy

It can be seen that there is a huge difference between the performance of the two methods, being Filtered Back-projection the clear winner. But if we take a more deep look at the results we can see that our euclidean approach outperforms (0.62 versus 0.14) the unfiltered Back-projection in terms of exact accuracy. This means that if we get a way to combine both approach the results could be even better. The reason behind why they are not combined yet will be discussed in the next point of the comparison.

### Runtime length

One of the biggest drawbacks that led to the search of another type of Back-projection method was the excessive runtime of the euclidean approach. It could take more than a day to get the best results and, although considering that with higher processing power maybe the computing time will decrease, it is not admissible that we need so much time to do the whole process. So, what happens with the second method?
Using Filtered Back-projection not only gives us more accurate results but also way faster. It takes no more than 3 seconds to reconstruct the same image that lasted 1 day with the

previous algorithm. It takes more time to reconstruct bigger or color images, but in none of the cases it will last more than 1 hour. It is clear that maybe the euclidean approach could be refined to deliver results sooner because the algorithm is very exhaustive, considering every possible case while Filtered Back-projection one does a similar operation but does not include (at this time) as many variations as the euclidean approach.
So the next step could be to include aspects of the euclidean approach into the Filtered Back-projection in order to improve the exact accuracy.

### Memory Constraints

Not only the euclidean geometry based approach lasts longer but it also uses more amount of memory while doing the whole process. In the case of Filtered Back-projection the algorithm outputs also an image of the same size as the original one, with no intermediate variables so we are saving a lot of memory, but it is in the intermediate step where the difference lies. In the end of the projection step in the euclidean approach ,five different variables (some with high size) needed to be returned and then used in the back-projection step while in Filtered Back-projection the only intermediate variable is the Radon transform which size is fewer than any of the Euclidean approach's variables. The only difference in terms of output is that the euclidean approach by itself outputs an image with the same size as the original one while in the Filtered Back-projection we have to tell the algorithm to reconstruct up to the size of max(width,height) and then taking back only the part of the Filtered Back-projection that we are interested in.

In conclusion, it is clear that at this moment the only option to consider will be using Filtered Back-projection because it is more accurate, faster and uses less memory. But it has to be considered that when we do not use filters, the euclidean approach performs better which is an indicator of a future integration step that could be done.

# 5. Conclusions and Future Work

## 5.1 Conclusions

Several projection and back-projection techniques had been discussed through this document in order to achieve the final objective which consists on getting an accurate reconstruction of an original CT scan image. The main reason to do this is to simulate the way CT scans work with less expensive devices as cameras are.
The first attempt consisted on simulating the functioning of the CT scan by using euclidean geometry, using lines as X-rays. As a first approach to the solution, two different ways to emit the rays were considered. Conic projection throws rays from every source point to each one of the possible image points. This solution gives the better results in terms of accuracy but it is also the one that lasts longer and uses more memory. The results get worse as the angular separation increases and gets better (a little) if the width of the rays is increased.
Another take to this approach considers that the range of values that a camera is able to store is limited. So this range can be divided in slots to store in each of them the average grey value of all the rays that fall between each slots borders. This enhancement gives similar results but uses less memory. The results do improve if a higher number of slots is used.
Besides the conic projection a simulation of a linear camera [Sources (2009b)] (which employs a paralell ray projection) has also been implemented. The concept behind linear cameras is associated to the construction of an image line by line, using a linear sensor so the camera moves with respect to the object to photograph or the object moves with respect to the camera, emitting a series of paralell rays. The results of simulations with this method gives worse results but more efficient in terms of computational cost and are way faster (mainly in the projection step).
Looking at the results it can be observed that, although they are not good by means of accuracy, they are in terms of shape reconstruction. The reason behind the lack of accuracy is because of the method itself, which adds error when it calculates the accumulative grey value, as it has been explained. The source of error has been proved because if we only take for each pair (source point, image point) as grey value the image´s point value, the reconstruction is perfect.
Taking all the drawbacks that had been cited, taking another different approach was straightforward. In the literature most of the recent articles use Filtered Back-projection algorithms. This group of methods are based in the Radon transform to perform a similar process to the one in the euclidean geometry approach but by taking advantage of the relationship between Radon and Fourier transforms. By using Projection-slice theorem of the Fourier transform a faster and more accurate method was achieved and the addition of a filter at the end of the process helps the reconstruction by managing possible problems with higher frequencies.
The results obtained by the use of Filtered Back-projection are clearly better in terms of accuracy (more than 49 % of difference) and also helps to get rid of some drawbacks of the euclidean approach such as the excessive computing time and the use of memory. While it can be improved (maybe by the use of more complex filters or forms of interpolation) the results are fair good.
We do not have to forget that the main objective of this project is to provide an efficient way to reconstruct an image from CT scans in order to develop a technique that let us simulate their functioning but with the less possible number of views needed. The importance of a correct reconstruction is related to the fact that images will be used later to detect anomalies

in the object to observe so it is important to provide an accurate reconstruction (in some cases shape reconstruction may be enough) because errors can lead to wrong diagnostics. That is why we propose the use of Filtered Back-projection algorithms which are more accurate and faster, two constraints that needed to be overcome.

### 5.2 Future Work

The main objective of this project, as it has been said previously, is to develop a procedure that simulates the functioning of CT scanners but with cameras. Once this first objective is achieved it is necessary to adapt the method to the needs of the environment where we will apply our method.
In our case, we want to use this techniques under an industrial point of view. In this kind of environment it seems clear that the accuracy in the reconstruction is the key point but there are some other aspects that should not be dismissed. If we want to introduce a CT scan in the production line the speed of the scanning and reconstruction process is crucial, because a long process could slow the overall performance of the line.
Although the algorithm can be improved and optimized so the process takes less time, there are some other options than can be considered, such as decreasing the number of views needed to reconstruct the image. Recent articles [Bertram et al. (2009)] show that it is possible to create intermediate views without losing much quality, so one of the tasks that will be faced shortly will be exploring these kind of techniques in order to improve our method in terms of speed and number of views needed.
Also, in the Results section, it was proven that a combination of the two approaches could lead to better accuracy results, so another future line could be combining both Euclidean and Radon-based approaches. Both techniques still have room for improvement, so the research in both trends will continue.
But we can not forget that this project can be seen as the beginning or first step of a bigger project where the enterprise may want to simulate the functioning of CT scanners in order to find defects in the product or detect special zones in the image so the next step could also consist on implementing an exploring or detection algorithm in order to find things like what have been said before.

### Acknowledgements

The author would like to thank his supervisor Dr. Javier Sánchez for the help and guidance throughout this Master Thesis. The author would also thank the help and company of his Master mates and the people of the CVC during all this year. This work has been done with the collaboration of the Universitat Autònoma de Barcelona.

## References

Tomografia por emision de positrones, 2003.

M. Bertram, J. Wiegert, D. Schafer, T. Aach, and G. Rose. Directional view interpolation for compensation of sparse angular sampling in cone-beam ct. 28(7):1011–1022, July 2009.

D.Sc. David J. Brenner, Ph.D. and D.Sc. Eric J. Hall, D.Phil. Computed tomography: An increasing source of radiation exposure. 2008.

Paul Hurst Eric Shieh, Wayne Current and Iskender Agi. A high-speed radon transform and backprojection processor. IEEE International Symposium on Circuits and Systems, 1: 234–237, 1990.

Ze Liu Xiao-hong Qin Jun-qing Geng Yan-ping Zhang1 Gui-hua Yao Yun Zhang Feng-rong Sun, Yan-ling Li. Model and simulation for three-dimensional medical image reconstruction of spiral ct. 2008.

Paul Gopi. Reconstruction of an image using the filtered back projection method, 2004.

S. Horbelt, M. Liebling, and M. Unser. Filter design for filtered back-projection guided by the interpolation model. In M. Sonka and J.M. Fitzpatrick, editors, Progress in Biomedical Optics and Imaging, vol. 3, no. 22, volume 4684, Part II of Proceedings of the SPIE International Symposium on Medical Imaging: Image Processing (MI'02), pages 806–813, San Diego CA, USA, February 24-28, 2002. Best poster award.

Bio Imaging. Our industrial ct is more than ndt. 2009.

J.M. Barcala Riveira J.L. Fernandez Marron, J. Alberdi Primicia. Desarrollo de algoritmos de reconstruccion de imagenes en tomografia de capacitancia electrica, 2004.

A.K. Louis. Combining image reconstruction and image analysis with an application to two-dimensional tomography. 1(2):188–208, 2008.

Julien Noel. Advantages of ct in 3d scanning of industrial parts. 2008.

A.Lynn Abbott Calvin J.Ribbens Raman P.V. Rao, Ronald D.Kriz. Parallel implementation of the filtered back projection algorithm for tomographic imaging, 1995.

High-Resolution X ray Computed Tomography Facility. About high-resolution x-ray ct. 2009.

C. Riddell and Y. Trousset. Rectification for cone-beam projection and backprojection. 25 (7):950–962, July 2006.

B. Sahiner and A.E. Yagle. A fast algorithm for backprojection with linear interpolation. 2 (4):547–550, October 1993.

Several Sources. Image reconstruction from projections. Biomedical lmaging and Emerging Technologies, 2003.

Several Sources. Information about bilinear interpolation, *http* : *//en.wikipedia.org/wiki/bilinear*$_i$*nterpolation*, 2009a.

Several Sources. Linear camera information, *http* : *//www.infaimon.com/catalog/catalog.php?cat* = 65, 2009b.

Several Sources. Matlab help 'http://www.mathworks.com/access/helpdesk/help/toolbox/images/index.html?/ 2009c.

Several Sources. Moire effect information, *http* : *//www.molinaripixel.com.ar/notas/moire.htm*, 2009d.

Huynh Quang Linh Vu Cong. 3d medical image reconstruction. 2003.

X-View. Our fastest ct system delivered to boeing. 2009a.

X-View. X-view ct: Industrial computed tomography systems. 2009b.

Y. Zhang O'Connor and J.A. Fessler. Fast predictions of variance images for fan-beam transmission tomography with quadratic regularization. 26(3):335–346, March 2007.

Guang-Hong Chen Zhihua Qi. Direct fan-beam reconstruction algorithm via filtered back-projection for differential phase- contrast computed tomography. X-Ray Optics and Instrumentation, 2008(ID 835172), 2008.