Università degli Studi di Padova

Dipartimento di Ingegneria dell'Informazione
Corso di Laurea Magistrale in Bioingegneria

TESI DI LAUREA MAGISTRALE

Geometric Misalignment Calibration and Detector Lag Effect Artifact Correction in a Cone-Beam Flat Panel micro-CT System for Small Animal Imaging

Laureando:
Lorenzo DI SOPRA

Relatore:
Alfredo RUGGERI

Anno Accademico 2015/2016

DEGREE PROJECT IN MEDICAL ENGINEERING 120 CREDITS, SECOND CYCLE

STOCKHOLM, SWEDEN 2015

Geometric Misalignment Calibration and Detector Lag Effect Artifact Correction in a Cone-Beam Flat Panel micro-CT System for Small Animal Imaging

LORENZO DI SOPRA

KTH ROYAL INSTITUTE OF TECHNOLOGY

SCHOOL OF TECHNOLOGY AND HEALTH

A Mamma e Papà,
a Silvia e a Giulia

Abstract

The cone-beam flat panel micro-CT is a high definition imaging system. It acquires projections of an object or animal to reconstruct a 3D image of its internal structure. The device is basically composed by a radiation tube and a detector panel, which are fixed to a gantry that rotates all around the test subject. The micro-CT system is affected by several imperfections and problems, that might lead to serious artifacts that deteriorate the quality of the reconstructed image. In particular, two issues have been discussed in the present work: the source-panel geometric misalignment and the detector lag effect. The first problem concerns the consequences of systems where the different elements are not perfectly aligned to each other. The second issue regards the residual signal, left in the detector's sensor after a projection acquisition, which affects the following frames with ghost images. Both these arguments have been investigated to describe their characteristics and behaviour in a typical acquisition protocol. Then two correction methods have been presented and tested on a real μ-CT device to verify their effectiveness in the artifacts compensation. In the end, a comparison between images before and after the corrections is provided and future prospects are discussed.

Abstract

La micro-CT è uno strumento per l'acquisizione di immagini ad alta risoluzione. È caratterizzata da rilevatori a pannello piatto e da un fascio conico di raggi X. Essa acquisisce numerose proiezioni di un oggetto o animale da angolazioni diverse, con l'obiettivo di ricostruirne un'immagine 3D della struttura interna. L'apparecchio è essenzialmente costituito da un tubo radiogeno e da un pannello fotosensibile che, solidali ad una struttura circolare mobile, ruotano attorno all'oggetto di interesse. Tale sistema è affetto da molti problemi e imperfezioni che possono provocare gravi artefatti con conseguente deterioramento della qualità dell'immagine ricostruita. In particolare, in questo studio vengono affrontate due diverse problematiche: da una parte il disallineamento geometrico esistente fra pannello e sorgente radiogena, dall'altra il lag effect prodotto dal sensore. Il primo problema riguarda le conseguenze di sistemi CT dove i diversi elementi non sono perfettamente allineati gli uni agli altri, provocando incongruenze nella forma tridimensionale dell'oggetto ricostruito. Il secondo riguarda invece un segnale residuo che, rimanendo intrappolato nei sensori del detector anche dopo l'acquisizione di una proiezione, causa la presenza di macchie e ombre nei frame successivi. Innanzitutto, entrambi gli artefatti sono stati studiati al fine di descrivere le loro caratteristiche e comprendere le modalità con le quali si presentano durante un tipico protocollo di acquisizione. In secondo luogo vengono proposti due metodi di correzione, i quali sono testati su un apparecchio micro-CT per verificarne l'efficacia nella compensazione degli artefatti osservati. Infine, viene proposto un confronto fra le immagini che si ottengono prima e dopo l'applicazione delle correzioni, e vengono discussi limiti e prospettive future delle strategie descritte.

Acknowledgements

During the six months I spent at School of Technology and Health at KTH, numerous people helped me and considerably contributed to this thesis project in many different ways, and I would like to express my deep gratitude to all of them.
First and foremost, I would like to sincerely thank Prof. Massimiliano Colarieti-Tosti. I am extremely grateful for the remarkable guidance, the kind support and all the precious time he dedicated to me; without him, this wonderful and enriching experience wouldn't have been possible.
I want to direct special thanks to Dr. Paolo Bennati and M.Sc. David Larsson for their indisputable help and for all the time spent working together. I would also like to thank Prof. Andras Kerek and M.Sc. Tim Nordenfur for their encouragement and valuable advices.
Thank you to my friends Massimo Terrin, Alexander Lundahl and Jacob Trowald, for their hard work and the important contributions to this project, as well as for the wonderful time spent together at KTH.
Finally, special thanks go to my friends Gioia, Stefano, Renato and Annaclaudia, you made this experience unforgettable for me.

Contents

1 Introduction **1**

2 Geometric misalignment calibration **5**

2.1 Background 5

2.2 Method 11

2.2.1 LabVIEW procedure 11

2.2.2 GATE Simulation 17

2.2.3 Image acquisition 22

2.2.4 Image reconstruction and correction 25

2.3 Results 27

2.4 Discussion 36

3 Detector lag effect artifact correction **39**

3.1 Background 39

3.2 Method 47

3.2.1 Lag effect visualization 47

3.2.2 Signal processing 49

3.2.3 Model identification 53

3.2.4 Correction tests 57

3.3 Results 59

3.4 Discussion 80

4 Conclusion **85**

A Appendix **93**

A.1 GATE simulation *macros* 93

A.2 GATE to LabVIEW conversion code 103

A.3 Acquired data to LabVIEW conversion code 106

A.4 Reconstruction misalignment correction code 108

A.5 Model identification 109

Chapter 1

Introduction

Computed Tomography is a widely diffused non destructive medical imaging technique. It is fundamentally based on the capacity of X-rays photons to pass through soft matter. Every tomographic device is based on two fundamental elements. The first one is the *radiation tube*, which is the source of the X-ray beam. The photons released by the tube are shot towards the test subject, and then captured by the second essential component, that is the *detector panel*. The sensor's purpose is to collect the photons that passed through the analyzed object. Within this framework, the most important feature is the capability of the matter to selectively attenuate the X-ray beam, according to its characteristics of composition, density and volume. As a consequence, the radiation beam received by the detector will be a projection of the object structure. In a CT device, this acquisition procedure is repeated over a large number of different angles around the object of interest. All the collected projections are then elaborated by a computer to synthesize a tomographic reconstruction of the internal shape of the test subject. Moreover, this technique allows to eliminate the superposition of anatomical structures that affects the static radiography [1].

The physical phenomenon, that makes possible the creation of projected images over the detector panel, is the attenuation accomplished by the atoms of the object's material on the X-rays. The photons shot by the source are subjected to different kind of interaction with the matter: the most important are the *absorption* and the *scattering*, that capture the X-rays or deflect them from their original direction. The direct consequence of these events is the removal from the beam of a certain fraction of the photons originally produced by the tube. This property of the matter can be resumed by a characteristic called *linear attenuation of the material* (usually expressed as cm^{-1}), defined in condition of monoenergetic beam and per unit of thickness.

With this coefficient, it's now possible to describe a relationship between the quantity N of primary photons transmitted by the matter and those N_0 initially received:

$$N = N_0 e^{-\mu x} \tag{1.1}$$

where x stands for the thickness of material the X-ray goes through. The evaluation of that coefficient is the fundamental step to achieve a tomographic reconstruction of the object, since the final image will actually be a spatial map of the attenuation distribution [2–4].

The geometry of the system represents one of the fundamental aspects for a CT scanner. The radiation tube and the detector's panel are typically fixed to a circular gantry, that moves around a bed holding the test subject. This configuration keeps the tube and the panel always in a constant relative position to each other. During the 360-degree rotation, the system collects projected images at every specific angle.
The detector's structure and the acquisition protocol can assume different configurations. In this particular case, it was employed a panel constituted by a square matrix of micrometric pixels distributed over a flat surface. The X-rays coming from the source hit the whole area simultaneously, since the radiation is shaped as a full cone beam. In this situation, each X-ray received by the detector corresponds to a specific line that links the radiation tube to the position of a particular pixel of the matrix. That captured ray corresponds to a measurement of the linear attenuation along that direction, and then an indirect description of the object the photon passed through. Moreover, since the bank of detectors doesn't follow the curvature of the gantry rotation, each pixel will be characterized by its own constant distance from the source. This is one of the fundamental parameters the reconstruction algorithm has to consider to correctly synthesize the final image [2,4].

In the end, the reconstruction algorithm is the process that transforms the acquired raw data in a CT image. In this case, the collected projections are a set of several 2D images, each of them describing the object from a different point of view over 360 degrees. The reconstruction step combine all of them to produce a unique 3D image, which can be visualized from several different cross-section that cut the entire volume. As the detector has two dimensions, the field of view might encompass the whole test subject, and then a single rotation of the gantry would be enough to reproduce the entire object. As a consequence, there would be no need to gradually move the bed in the axis of rotation direction.
The task of reconstructing an image from a set of its projections can be

performed with different algorithms, according to the geometry of the image acquisition system. Anyway, from a general point of view, those methods are different versions based on the same mathematical theory. First of all, the measured data are transformed with a logarithmic function to compensate the attenuation produced by the interaction of the beam with the matter, which has an exponential description. Then, since the problem is now linearized, the projected signal received by the detector can be considered as a linear sum of the coefficients μ_i, which are the values we want to estimate. This reverse problem can be solved with a *back-projection* approach: the measured values, coming from points of view caught at different angles around the test subject, are smeared back to compute a matrix that reproduces an image of the object. This idea comes from the *projection theorem*: if an infinite quantity of projected images all around the object were available, then a perfect reconstruction of its shape could be possible. In particular, a solution of this inverse problem can be achieved employing the inverse of the *Radon transform*, which is the fundamental step for the image reconstruction [2,5].

According to the framework just illustrated, an ideal tomographic device is first of all characterized by a perfectly defined geometry, where all the components are positioned in the exact place they are expected to be. Secondly, in a perfect acquisition system, each projected image is strictly independent from each other, that is a measured value is by no means influenced by the past history of the detector.
On the contrary, a real CT device cannot be considered as an ideal system. The radiation tube and the flat panel cannot be aligned on the gantry with absolute precision, and the past measures acquired by the sensor necessarily condition the outcome of the following projected values. These two types of imperfections can considerably affect the results of a reconstruction in an imaging device. This is especially true in conditions of extremely high spatial definition, as those that characterize this micro-CT system, where the pixel's dimension is just 50-by-50 μm. These two conditions, if they were neglected and not conveniently compensated, could lead to serious artifacts. These undesired effects might deteriorate the reconstructed image quality, and maybe hide some tiny but important details, wasting the high definition capabilities of the detector panel.

The purpose of this study is to analyze the presence, behaviour and consequences of the two aforementioned problems, that might affect the micro-CT device: these are the *detector's misalignment* and the *lag effect*. Then, two correction algorithm will be employed and tested on some real acquisitions to verify if the resulting artifacts can be effectively compensated and deleted from the reconstructed image.

Chapter 2

Geometric misalignment calibration

2.1 Background

The quality of a tomographic image considerably relies on how precise the geometry of the acquisition system is. This is especially true when the detector used to collect the projections has an extremely high resolution as in our specific case, where the pixel dimension is 50-by-50 μm. That means that even a misalignment of a fraction of a millimeter of the detector from its ideal position might cause some serious artifacts on the reconstructed image. That's why it is of crucial importance to adopt an efficient calibration method: that should be able to evaluate the spacial parameters of the scanner's position with the highest precision possible. This information will then be used to apply the necessary corrections to the projected images in a post-processing step. The method that will be further discussed, firstly introduced by *von Smekal* [6] and based on a previous work of *Gullberg* [7], has been specifically designed for a tomographic acquisition system with a flat panel area detector and a cone shaped X-ray beam.

The method that will be described is based on the analysis of the projected trajectories of some ball shaped objects. Indeed, a phantom composed of several point-like spheres, according to the rotating movement of the CT system, produces different orbits whose geometrical properties can be exploited to estimate the misalignment parameters of the detector panel [8]. In fact, the method analyzes the low spacial-frequency components of those trajectories, and it doesn't even need the knowledge of the initial position of the spheres, which represent a great advantage of this approach [6,9].

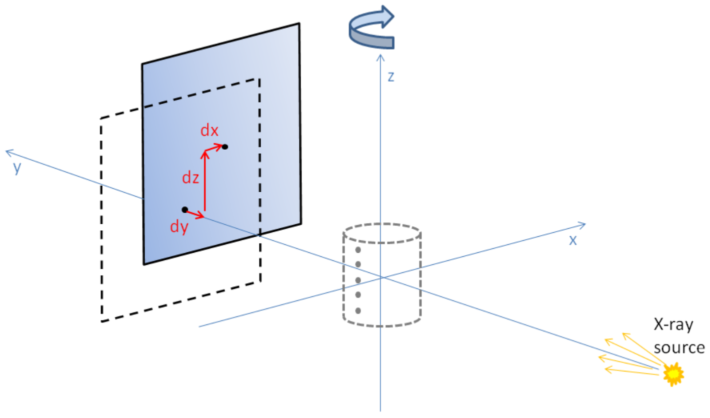


Figure 2.1: Spatial framework with 3 translations of the panel.

The spatial framework used to describe the calibration method is illustrated in Figure 2.1. The source-detector system is built around the z-axis, which is the center of rotation of the whole CT gantry; in this case, without loss of generality, we consider the probe at rest, with the X-ray tube and the flat panel moving. Against this background, $\boldsymbol{R}$ is defined as the distance from the source to the center of the perfectly aligned detector. Similarly, the distance of the source from the axis of rotation is called $\boldsymbol{R_F}$. Both those ideal distances lie on the y-axis. In other words, the source is located in $-(R_F)\vec{\boldsymbol{y}}$, while the center of the ideally positioned panel is $(R - R_F)\vec{\boldsymbol{y}}$. Relative to that point, the real detector will be displaced by a vector $\boldsymbol{d} = \{d_x, d_y, d_z\}$; in addition, $\boldsymbol{R'} = (d_x, R + d_y, d_z)$ specifies the real position of the detector if referred to the source. In the same way, instead of being exactly contained in the (x, z)-plane, the panel will be rotated around the coordinate axes by 3 unknown angles [6,9]. These rotations, with particular attention to the order of application, are shown in the Figures 2.2, 2.3 and 2.4.

The first one to be defined is the *skew* η, which sets the angle around the y-axis. The second one is the *tilt* θ, that rotates the plane around the new x-axis, that is the one already skewed. At last there is the *slant* ϕ, that rotates the detector out of the skewed and tilted plane, moving it from left to right around the new z-axis. Each of these three movements can be defined by a different rotation matrix. Then, these matrices can be combined to obtain a single orthogonal matrix $\mathbf{O}$:

$$\mathbf{O} = \begin{pmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} \cos\eta & 0 & -\sin\eta \\ 0 & 1 & 0 \\ \sin\eta & 0 & \cos\eta \end{pmatrix}$$

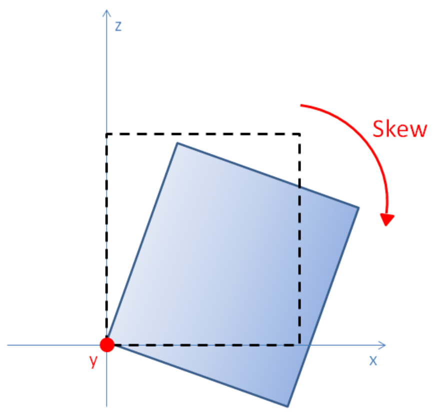


Figure 2.2: Example of the panel rotating around the y-axis (skew).

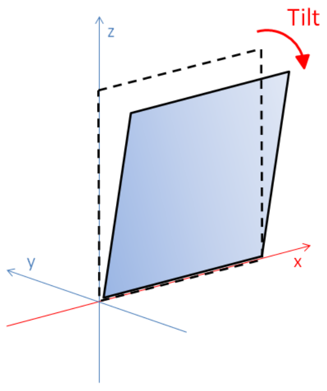


Figure 2.3: Example of the panel rotating around the x-axis (tilt).

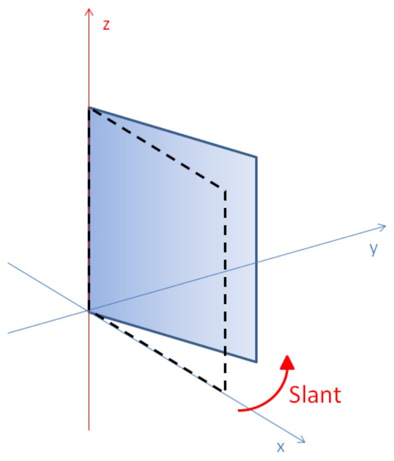


Figure 2.4: Example of the panel rotating around the z-axis (slant).

that produces the geometrical transformation of the detector from its ideal position to the real one [10].

Therefore we have introduced the 6 misalignment parameters, three rotations and three translations corresponding to all the degrees of freedom of the detector, that we will need to estimate:

$$\{\phi, \theta, \eta, d_x, d_y, d_z\} \tag{2.1}$$

As it is above mentioned, the calibration method needs a set of projected point-objects, acquired over a complete rotation around the phantom. Each point position of the elliptical orbit, which is composed by a uniform collection of samples, is then transformed in a discrete real Fourier series. That is the trajectory is expanded in a weighted sum of Fourier coefficients, each one related to different frequency components of the curve. The only coefficients that will be considered in the following are the lowest ones, usually the first 3 of them. By doing so, all the high frequency fluctuations (that mostly describe the measurement errors and random noise) are discarded, keeping only those that actually form the elliptic trajectory [9]. Moving on from this Fourier analysis, it is possible to get the c'_{ij} and the d'_{ij} coefficients. These will be the values we can use to define five formulas that provide an analytic solution to the misalignment problem [6].

The first equation the method introduces is quite general and independent from all other parameters:

$$\tan(\eta) = -\frac{c'_{11}}{c'_{01}} \tag{2.2}$$

that allows a direct calculation of the skew. Secondly, with different combinations of the perspective coefficients and the estimated value of η, five constants are introduced:

$$A = \sin(\eta)c'_{00} + \cos(\eta)c'_{10} \tag{2.3a}$$
$$B = \cos(\eta)c'_{01} - \sin(\eta)c'_{11} \tag{2.3b}$$
$$C = \cos(\eta)c'_{00} - \sin(\eta)c'_{10} \tag{2.3c}$$
$$E = \sin(\eta)d'_{02} + \cos(\eta)d'_{12} \tag{2.3d}$$
$$F = \cos(\eta)d'_{02} - \sin(\eta)d'_{12} \tag{2.3e}$$

Those constants are then used to evaluate the angle ϕ through the formula

$$\tan\phi = \frac{C - F}{\sin\theta(A - E) \pm B} \tag{2.4}$$

where the $\pm$ symbol considers if the rotation sense of the system is clockwise or counterclockwise respectively. In the end, with all the previous coefficients, we can find the translation parameters as

$$d_x = \sin\phi(A\sin\theta \pm B) - C\cos\phi \quad or \quad d_x = \sin\phi\sin\theta E - \cos\phi F \tag{2.5}$$

$$R'_y = R + d_y = -\cos\phi(A\sin\theta \pm B) - C\sin\phi \tag{2.6}$$

$$d_z = -A\cos(\theta) \tag{2.7}$$

This method allows to find an estimate of the pool of parameters from a single point object. This means that using all the trajectories available it is possible to get several estimates: with all those values we might then compute the averages over all the orbits considered. Obviously it won't be possible to find all the six values since we lack one equation. One of the possible solutions, that has been adopted in the first of the two methods employed, assumes the tilt equal to zero. Despite this hypothesis necessarily introduces an amount of error, it has been observed that even a significant tilt of the detector produces a negligible effect on the reconstructed image [6]. On the other hand, if we do not want to quit the possibility of evaluating the complete set of misalignment parameters, it's necessary to consider at least two point objects at the same time. This second calibration method here discussed requires more than a single trajectory, and exploits the point-dependence of some coefficients to define a new equation. In fact, considering a set of $k = 1, ..., K$ different objects, we rename the coefficients

$$E \to E_k \qquad F \to F_k \qquad \frac{z_0}{R_F} \to z_k$$

to highlight their dependence on the initial position of the point they relate to. Considering now the different values, the averaged counterpart of those coefficients is indicated as

$$\bar{E} = \frac{1}{K}\sum_{k=1}^{K} E_k \qquad \bar{F} = \frac{1}{K}\sum_{k=1}^{K} F_k$$

that automatically leads to the updated version of equation (2.5)

$$d_x = \sin\phi\sin\theta\bar{E} - \cos\phi\bar{F} \tag{2.8}$$

At this point, since E_k and F_k are point-dependent but the d_x parameter is not, for all $k = 1, ..., K$ with $k \neq j$, we can write

$$0 = \sin\phi\sin\theta(E_k - E_j) - \cos\phi(F_k - F_j) \tag{2.9}$$

If this equation is combined with the (2.4) we finally obtain the missing formula that estimates the tilt value and complete the calibration algorithm:

$$\sin\theta = \frac{\pm B(F_k - F_j)}{(E_k - E_j)(C - F_k) - (F_k - F_j)(A - E_k)} \tag{2.10}$$

The main limit of this second approach is the following condition: the (2.10) can be applied to calculate the tilt θ only if the slant ϕ doesn't vanishes. Indeed, if $\phi = 0$ we can clearly verify from equation (2.4) that $F_k = C$ for every single trajectory considered. In that particular case the (2.10) becomes of the not solvable form 0/0, and it's no more possible to find an estimate for θ, as well as all the other parameters except from the skew, since they all depend on the tilt value. In the end, a way to calculate the ideal source-detector distance is shown below

$$R'^2 = A^2 + B^2 + C^2 \pm 2\sin(\theta)AB \tag{2.11}$$

or the equivalent

$$R'^2 = c'^2_{00} + c'^2_{10} + c'^2_{01} + c'^2_{11} \pm 2\sin(\theta)AB \tag{2.12}$$

while the real counterpart (an alternative to the (2.6) equation) is

$$R'_y = \sqrt{R'^2 - d_x^2 - d_z^2} \tag{2.13}$$

2.2 Method

2.2.1 LabVIEW procedure

Moore implemented the algorithm described in section 2.1 for the FaCT adaptive micro-CT of the University of Arizona [9,11]. We are grateful to Lars Furenlid for sharing the code they have used for the FaCT micro-CT with us. The programming language chosen was LabVIEW, and the code came in two different versions. In both cases the visual interface of the program is represented by a main function called respectively *RunFullCalculation* and *RunFullCalcWTilt*. Those two codes fundamentally work in the same way, since they both require as input:

- the reference to a folder containing a set of projection images of the calibration phantom (whose shape will be further discussed). Those files need to be in a *raw* format;

- the number of projections employed. They have to be placed over 360 degrees, separated by a constant angle;

- the quantity of point objects, that is the number of trajectories, considered by the algorithm.

Similarly, once the code is running, it is required to draw a rectangular window on a summed image of the complete set of projections. That procedure is used to select all and only the number of balls needed, whose orbits should never intersect. An example of that step is showed in Figure 2.5.
The codes then return several sets of parameters estimates, as many as the quantity of objects considered; for each parameter it is also calculated the mean value. We are now going to examine in depth the modes of operation and the most significant differences that exist between the two LabVIEW implementations.

RunFullCalculation

The first code we are going to analyze is a simplified version of the calibration algorithm described in the previous section. Actually, in this case a double assumption on the misalignment parameters is made: the hypothesis is that the tilt θ and the slant ϕ are both negligible [6]. That means they are set to zero, which reduces not only the number of parameters we have to estimate but also the complexity of the equations to employ. The following formulas

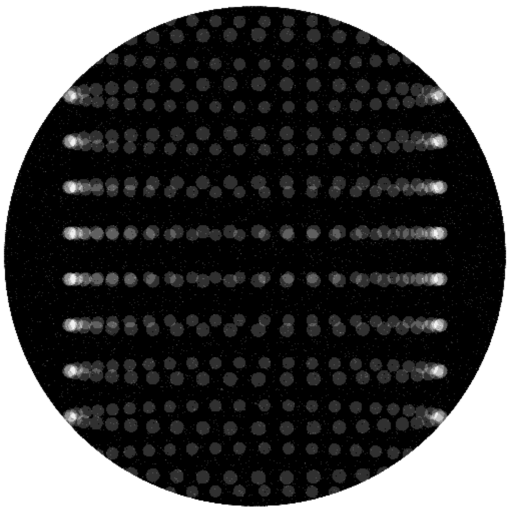


Figure 2.5: Example of some ball trajectories produced by the summed image of a 360 degrees acquisition.

are the revised versions of equations (2.2), (2.5), (2.6), (2.7) and (2.12):

$$\eta = \arctan(-\frac{c'_{11}}{c'_{01}}) \tag{2.14}$$

$$d_x = -F \tag{2.15}$$

$$R'_y = \mp B \tag{2.16}$$

$$d_z = -A \tag{2.17}$$

$$R' = \sqrt{c'^2_{00} + c'^2_{10} + c'^2_{01} + c'^2_{11}} \tag{2.18}$$

To this set of equations one supplementary is added:

$$z_k = \frac{E + d_z}{R'_y} = \frac{E - A}{\mp B} \tag{2.19}$$

that estimates the initial position of each ball along the z-axis.

This code was already perfectly working and, even if it doesn't calculate the whole set of parameters, it has been successfully used by *Moore* to verify the effectiveness of the misalignment correction.

In Figure 2.6 it is illustrated a diagram that reproduces the structure of the *RunFullCalculation*; each box represents a different function and the wires collecting two boxes suggest that the higher function is calling the lower one.

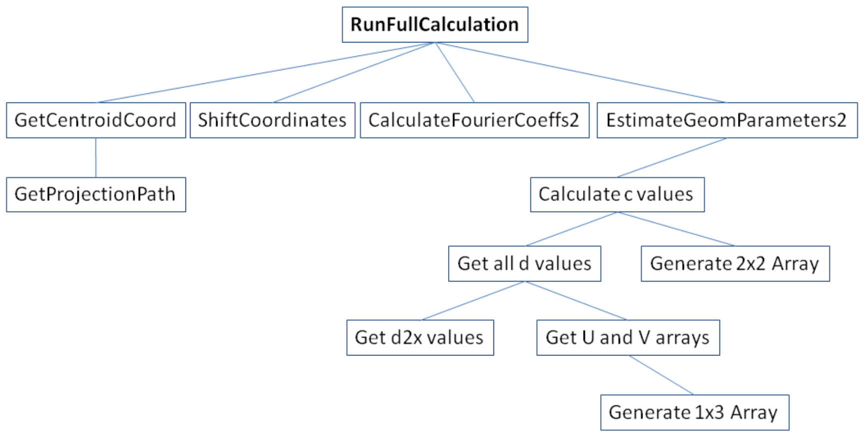


Figure 2.6: Hierarchical diagram of the *RunFullCalculation* LabVIEW system. Each link means that the upper function calls the lower one.

RunFullCalcWTilt

This second code we are now introducing is the complete implementation of the calibration algorithm. As it was previously described, in this case the tilt θ is the first parameter estimated (right after the skew η), since it is then used to calculate almost all the others unknown quantities. That's why one of the main differences with the *RunFullCalculation* code is the introduction of new functions, such as *CalculateAllThetas* and *CalculateTheta*, that implement the following algorithm

$$\theta = \arcsin\left[\frac{1}{N-1}\sum_{k\neq j}\frac{\pm B(F_k - F_j)}{(E_k - E_j)(C - F_k) - (F_k - F_j)(A - E_k)}\right] \tag{2.20}$$

that considers all the possible couples made by two different trajectories. Then, to estimate the values of η, ϕ, d_x, R'_y, d_z and R' the code uses equations (2.2), (2.4), (2.5), (2.6), (2.7) and (2.12) respectively. In the end, the points' initial z-coordinates (in units of R_F) are found out with

$$z_k = \frac{(E - A)\cos\theta\cos\phi}{(E - A)\sin\theta \mp B} \tag{2.21}$$

Contrary to the first code, the *RunFullCalcWTilt* was not working and not even completed, since some parts of the algorithm were wrong or missing. To complete the code, first of all it has been necessary to modify the wiring and the output structure of the *CalculateAllThetas* function. Moreover, the calculation of the R_F (the real distance of the source from the center of rotation) and the *Magnification* (the ratio of the source-detector to the source-center distances) were logically wrong as well.

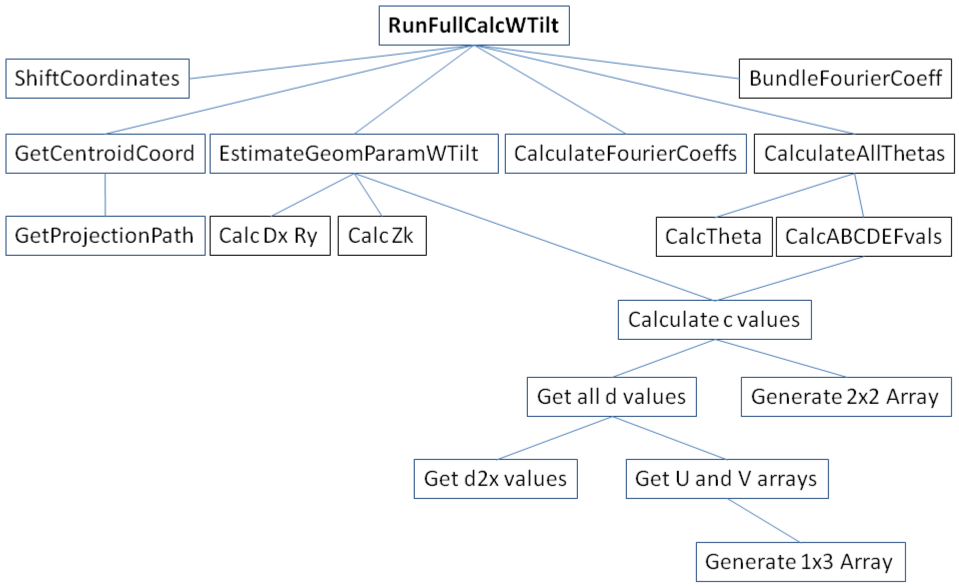


Figure 2.7: Hierarchical diagram of the *RunFullCalcWTilt* LabVIEW system. Each link means that the upper function calls the lower one.

Codes Comparison

In addition to the 6 parameters that describe the geometrical misalignment of the detector panel, another variable we already mentioned is actually estimated: that's the z_k, an information about the initial position of the point objects that are used to perform the calibration method. In this regard, the particular phantom used for our purpose, illustrated in Figure 2.8, is now briefly discussed [6]. Since the algorithm requires the presence of a certain number of distinct trajectories on the summed image, an appropriate shape for the phantom is a set of metal balls mounted on a plastic scaffold. Moreover, that probe should present some particular characteristics:

- since we are interested in getting the brightest image possible of the balls, they should have an high contrast, whereas the scaffold should be almost transparent to the X-rays. That's why metal and plastic are the chosen materials;
- the balls should have a small dimension, so that it will be easier to find their center and then the trajectories they go through;
- the balls should be placed more or less in a linear array along a straight line parallel to the axis of rotation (but not coincident!).

- the distance left between two consecutive balls should be always the same. That will turn out to be useful, as discussed in the next paragraph.

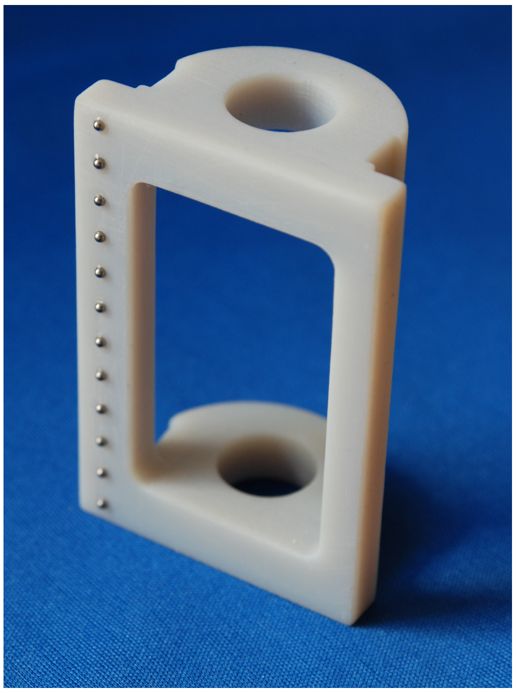


Figure 2.8: Calibration phantom with 12 metal balls. The phantom was 3D-printed for us by Lars Furenlid, UoA.

Since a particular phantom of this kind was employed, we now have some further information we can use to get a few more characteristics of the gantry. Indeed, putting together some data as the number of point objects considered (N), the distance between two consecutive balls (D) and the z-coordinates, in units of R_F, of the first and the last sphere (z_k^1 and z_k^N), it is possible to estimate the R_F distance as follows

$$z_0^1 - z_0^N = D(N-1) \qquad \rightarrow \qquad \frac{D(N-1)}{z_0^1 - z_0^N} = 1$$
$$\rightarrow \qquad \frac{D(N-1)}{z_0^1 - z_0^N} R_F = R_F \qquad \rightarrow \qquad \frac{D(N-1)}{\frac{z_0^1}{R_F} - \frac{z_0^N}{RF}} = R_F$$
$$\rightarrow \qquad \frac{D(N-1)}{z_k^1 - z_k^N} = R_F \tag{2.22}$$

Furthermore, with this new information on R_F and the already calculated R'_y, it's now possible to evaluate the magnification factor *AvgMagn* of the current gantry set-up.

In the end, it has been necessary to modify some parameters and coefficients in the LabVIEW code with the aim of matching the characteristics of the system we are actually developing. In particular, the values we have to check before running the calibration algorithm are:

- the size of one pixel of the detector, expressed in mm (functions *RunFullCalculation* and *RunFullCalcWTilt*)
- the number of pixels each matrix, containing the projected image, has to be shifted to get the center of coordinates exactly in the middle of the rectangle (function *ShiftCoordinates*)
- the range of all the different levels of gray that form the image; you have to specify the lower and upper bound of the interval that contains the shades of the balls (front panel of function *GetCentroidCoord*)
- the lengths (number of pixels) of the sides of the images given as an input to the code (block diagram of function *GetCentroidCoord*)
- the numerical format of the matrices containing the images (block diagram of function *GetCentroidCoord*)

2.2.2 GATE Simulation

After the examination, modification and correction of the LabVIEW code that implemented the calibration method, it was necessary to evaluate how well the algorithm works. To this end, a simulation of the whole image acquisition system is an excellent way to test the performance of the misalignment correction, before the application to the real CT device. In particular, a simulation can reproduce different misalignment situations, and this allows us to evaluate and compare the two algorithms, to verify their effectiveness, precision, consistence in the estimate of the parameters, and also to find out the range of applicability of both the codes.
With this aim, all the main characteristics of the system's geometry and behaviour were recreated in GATE, a software dedicated to numerical simulations in medical imaging and radiotherapy. The GATE code is composed by a few macro (reported in the Appendix A.1) where all the geometric and physical aspects are specified. These features should accurately reproduce the CT characteristics, but as we will see, the computational complexity (and the time needed with it) might rise really fast. That's why in this simulation framework a compromise will have to be reached, and all the decisions will be discussed with this purpose. We are now going to describe the simulation structure following the code flow contained in the macros *CT_test*, *CT_test_phantom* and *xrayspectrum*.

World

The world, that is the volume that has to contain all the structures and all the phenomena that we want to describe, was defined as a cube with a length of 100 cm for each side. The geometric framework we work in is based on a specific system of coordinates, which is different from the one considered in the theoretical presentation of the calibration algorithm. That's an important issue we have to consider when we want to analyze the estimated parameters and compare the simulated system with the theoretical and the real ones.

CT scanner system

From a computational point of view, the structure of the detector here described is one of the critical aspects. The scanner dimensions are 64x64 mm^2 (with a depth of 1 mm) which are much smaller than the real ones 120x120 mm^2, but large enough to record the complete trajectories of 6 or 7 point objects. Furthermore, the size of each pixel is set to 0.125x0.125 mm^2, that is more than 6 times bigger than the real one (0.05x0.05 mm^2). This also implies a

different total number of pixels that form the panel: in the real one there are 2400x2400 pixels, while in the simulated one they are only 512x512. That means a significant loss in terms of resolution, but also a great reduction of the total duration of the simulation.
Similarly, the positioning of the scanner is of crucial importance for the misalignment properties we want to investigate. Indeed, the panel is placed 40 mm away from the center of the system of coordinates (which is in the middle of the $World$) along the z-axis. That length defines the $R - R_F$ distance. Starting from that position, the scanner is then translated in the other two directions ($\vec{x}$ and $\vec{y}$) by an arbitrary length. Those two shifting distances are exactly the misalignment parameters d_x and d_z we want to estimate with our algorithm. At this point the panel is rotated by three angles about the three axis, paying attention to the correct order (skew - tilt - slant), since every rotation has to be performed around the respective axis in his current position, that is the one already moved by the previous rotations. The implementation of these movements in the GATE environment (that is the intentional misalignment of the detector) causes some additional issues in the framework construction. In particular, if the simulation of a multiple rotation is needed, it is necessary to define a specific subvolume for each rotation desired. This expedient allows to correctly rotate the panel with respect of the previous movements. Otherwise, the simulation software will automatically consider the *World* system of coordinates as the default reference for all the rotations.

Phantom

The phantom here described only reproduces the main features of the real one. In particular, we are not interested in the plastic scaffold since it doesn't contribute to the calibration. On the contrary, it is of crucial importance to have a bright image of the point objects. For those reasons the simulated phantom was defined by a cylinder made of air (the scaffold) with an array of perfectly aligned lead balls in it. That metal was chosen due to his high contrast characteristics. For what concerns the probe dimensions, it was decided to match those of the real one. For that reason the height and the radius of the scaffold were set to 59.5 mm and 23.5 mm respectively. The array of metal balls was placed in the $\vec{y}$ direction: each one is 5 mm away from the previous one, and its diameter is 1.5 mm. The line they stay in is 15.75 mm in the x-direction and 12.75 mm in the z-direction, and then approximately 20.26 mm away from the y-axis.
To reproduce the tomographic movement, it was decided to rotate the phantom around his own axis of symmetry, instead of moving the whole source-

detector system as happens in the real machine. That solution is perfectly equivalent to his counterpart, and avoid some rotation problems caused by the detector's misalignment. The chosen speed was 8 or 2 degrees per second, that means a total of 45 or 180 projections acquired respectively, since GATE considers one second for every single image. In Figure 2.9 the simulated phantom and detector panel are illustrated in the GATE environment.

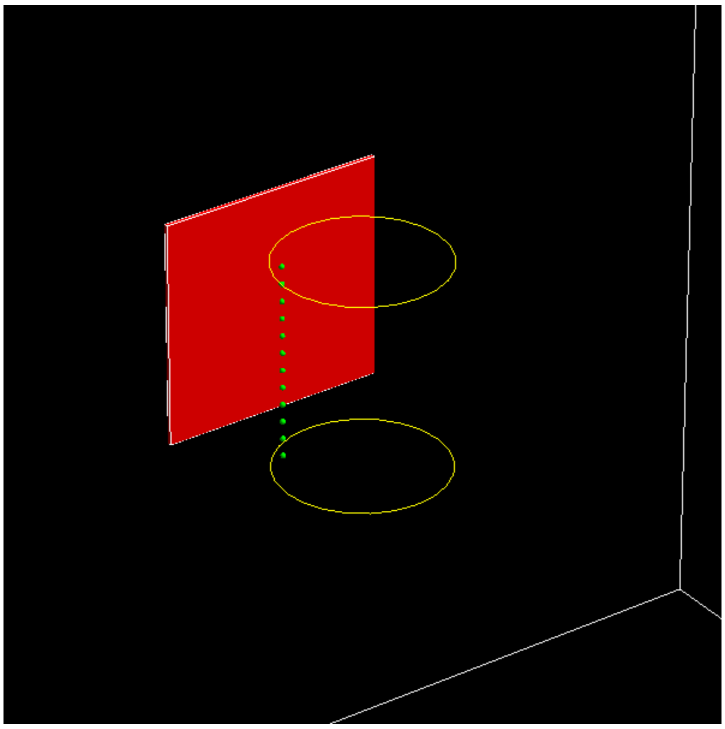


Figure 2.9: Gate simulation visualization of the detector panel (red square) and calibration phantom (cylinder with yellow borders and 12 green spheres). One corner of the *World* is visible (white lines). The X-ray source is out of the field of view.

Source

The source of X-rays was positioned 284 mm away from the center of rotation, along the z-axis. That means that R_F is 284 mm, while R, the ideal source-detector distance, was equal to 324 mm. The simulated source was created with no dimension, meaning that all the X-rays start from that single origin point. This is an unrealistic hypothesis (since the X-ray tube anode has physical dimensions which are obviously different from zero), but helps to avoid some randomness, and then a possible source of error, during the following calibration step. The beam was shaped as a cone and the X-rays are spreading uniformly over that space. The cone beam is defined by its

summit angle, which was set to 5.5 degrees, that is just sufficient to encompass the 6 or 7 trajectories we are interested in.
At this point, it was necessary to simulate a realistic distribution of X photons with different energies. The chosen tube voltage range spread from 9 kV up to 40 kV, producing a beam with an equivalent mean energy of approximately 27.3 keV. The simulated X-ray tube was composed by a tungsten anode and a 1 mm thick aluminium filter. The resulting distribution is illustrated in the diagram of Figure 2.10.

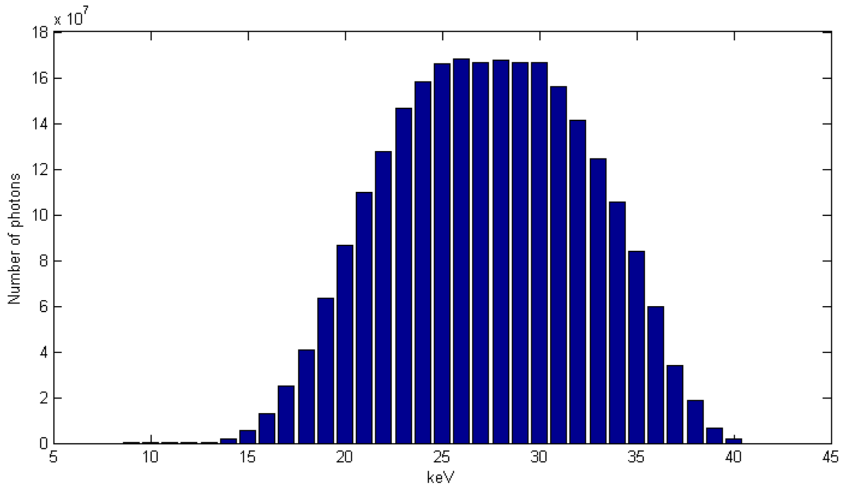


Figure 2.10: Simulated energy distribution for an X-ray beam with mean energy equal to 27.3 keV

In the end, it was necessary to set the intensity of the source, which is one of the main features that affect the computational complexity. The intensity is measured by the quantity of photons produced by the simulated X-ray tube in a single shot. In our case it was chosen to set this parameter to 1.5 million. This implies that in a single image each pixel receives about 7.3 photons on average, which is a really low quantity, but it came out to be sufficient to highlight the desired trajectories.

Output

To get the output data from the simulation and to convert them in a format the LabVIEW code can read is the last step we have to accomplish. The GATE code returns several files containing a list of data concerning the history of every single simulated photon. These files may contain a large amount of collected information, but most part of it was suppressed to reduce the running time, since it wasn't strictly necessary. Then a Matlab code was

created to organize the data in the format desired. The file is showed in the Appedix A.2.
The greatest issue solved by that code was to recreate the projection images using the information implicitly contained in the GATE output files. Moreover, all the images were modified to better satisfy the goal they have to reach. Each frame's contrast was enhanced, their dynamic range was modified (and the numeric format with it) and negative images were produced; the resulting projections exhibit white objects on a dark background, which are finally suitable for the LabVIEW calibration code.

2.2.3 Image acquisition

After the simulation of the whole CT acquisition system to check the effectiveness of the calibration algorithm, the next step was the test of the real device to find out the actual misalignment parameters. To fulfill this task, the kind of phantom used is the first issue to be discussed. The probe we employed is the one we tried to reproduce in the simulation framework: it is composed of an array of 12 metal balls attached to a plastic scaffold. All the observations about the materials the probe is made of are quite the same we made for the simulated one. The phantom used was already shown in Figure 2.8. Actually, three phantoms of the same kind were available, the three of them having the same shape but different dimensions. Anyway, from the calibration point of view, the most important difference among the phantoms is the distance between two consecutive balls. That length is 4 mm, 5 mm and 6 mm, depending on the case. All the other dimensions are not essential since the alignment algorithm doesn't need to know the initial position of the point objects. In this case the biggest phantom was chosen, because a higher distance between two consecutive balls implies more separated trajectories, which is a great advantage for the proper functioning of the algorithm. The phantom was then fixed to the carriage with some adhesive tape, roughly in the middle of the gantry (that means close to the center of rotation).

At this point the acquisition protocol is going to be discussed. That concerns all the settings and the parameters we have to choose in order to get the correct images to perform the calibration algorithm. Actually this acquisition situation reveals a lot of advantages that make the choice of the protocol easier. Firstly, the duration of the whole tomographic acquisition does not represent a problem, since the probe is obviously not moving, and the calibration has to be done only once. Secondly, we are not concerned about the dose of X-rays received by the phantom: that means there is no drawback about making use of a high flux of photons. That allows us to set a high current in the X-ray tube and a long pulse width, so we are sure to have a bright and high-contrast image. Moreover, it is possible to acquire a great number of projections to get high sampling rate trajectories.
The detailed protocol will be displayed in the *Results* section.

Once all the projection images have been acquired, it is necessary to modify them in order to make them suitable for the LabVIEW calibration code. However, that is not just a matter of numeric format: an enhancement of the images and a transformation of the system of coordinates is needed as well. In this regard, we will refer to the Matlab code provided in Appendix

A.3. This code applies a geometric transformation to all the projections, with the aim of bringing the acquired images in the system of coordinates of the calibration framework. The situation is illustrated in Figure 2.11 and 2.12: the first image shows the real position of the detector against the CT system, while the second one reproduces the ideal condition according to the misalignment algorithm theory. The red letters A-B-C-D in the four corners of the panel show the corresponding points in the two situations. Also, in the detail of Figure 2.13 it is described how the images are displayed once we read the output files: the A stands for the upper-left corner, while the D is for the bottom-right one.

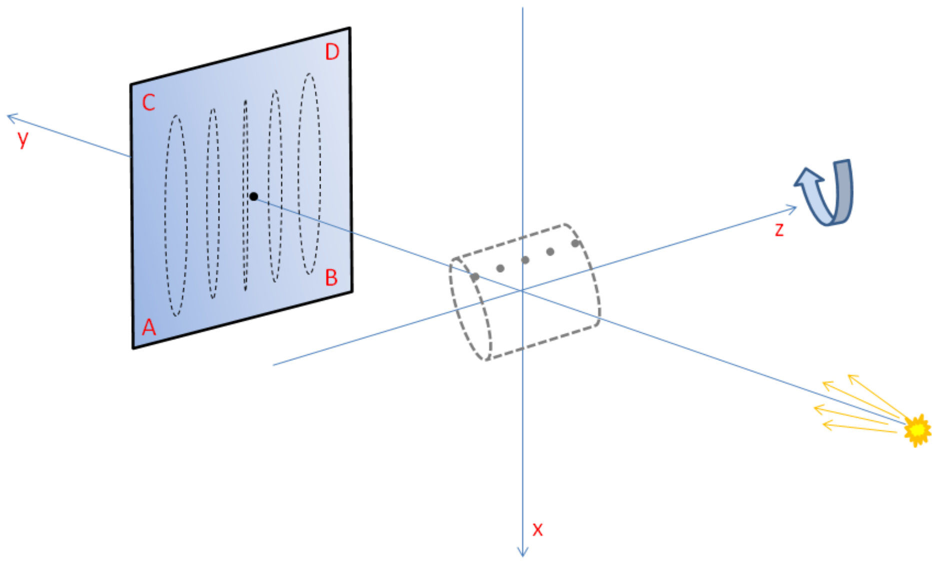


Figure 2.11: Simplified representation of the real CT device geometry. The blue arrow suggests the gantry sense of rotation. The letters ABCD specify the spatial orientation of the detector, with respect to the condition illustrated in Figure 2.13

From this comparison it is possible to define the correct transformation: the matrices need to be flipped upside down, and then rotated clockwise of 90 degrees.

The images are now ready for the next step; they are transformed in *uint16* files and their contrast has been enhanced with a logarithmic transformation function, that improves the grey level difference between the balls shadows and the background. Lastly, they are modified to get the negative images and then converted to the well-known *raw* files.

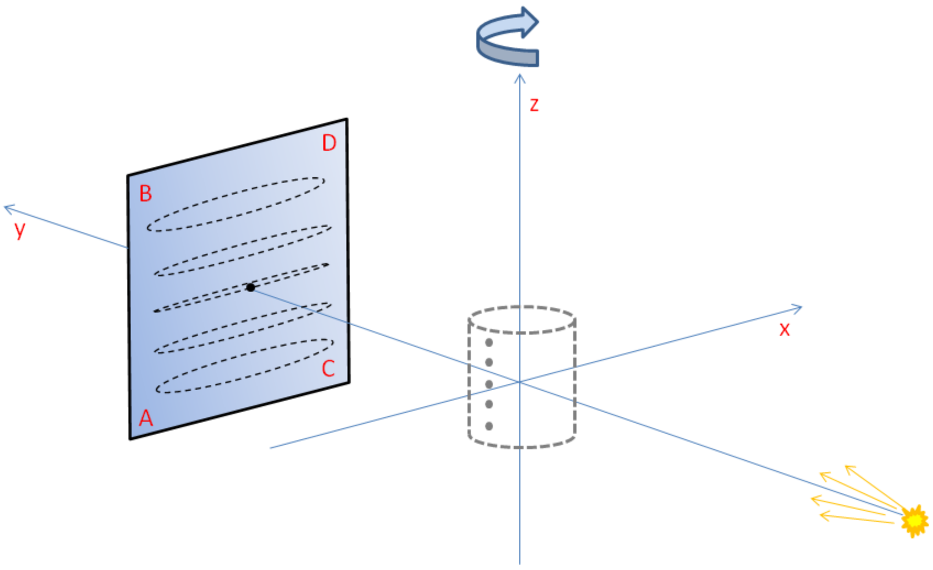


Figure 2.12: Simplified representation of the ideal CT device geometry as it is illustrated in the theoretical background and LabVIEW framework. The blue arrow suggests the gantry sense of rotation. The letters ABCD specify the spatial orientation of the detector, with respect to the condition illustrated in Figure 2.13

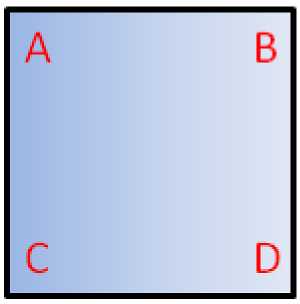


Figure 2.13: Reference of the detector position for Figures 2.11 and 2.12.

2.2.4 Image reconstruction and correction

Once the calibration algorithm is performed and the misalignment parameters are found, it is time to test the estimated corrections and verify if they are effective or not. For this purpose, an acquisition with a new specific phantom is needed; then, a tomographic reconstruction has to be performed with and without the introduction of the alignment corrections, in order to check the possible effects on the image quality.

The phantom that was used for this purpose is illustrated in Figure 2.14.

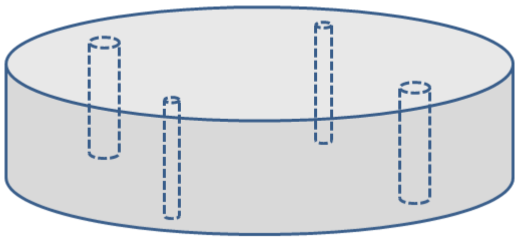


Figure 2.14: Representation of the cylindrical plastic phantom with four holes used for the reconstruction tests.

It is a cylinder made of plastic, with two pairs of holes of different sizes: their diameters are 1 mm and 2.3 mm respectively. Even the holes have a cylindrical shape, they pierce the plastic from the top to the bottom, perpendicularly to the flat surfaces. Also, none of them is closed, that means they are full of air. The phantom was placed on the carriage, fixed to it with some adhesive tape. It was positioned roughly in the middle of the gantry, with its axis of symmetry parallel to (and not so far from) the axis of rotation. At this point, a 360-degrees acquisition was started. All the settings are provided in the *Results* section.

The final step we developed is the cone-beam reconstruction of the phantom volume from its projection images. To this end, a *Feldkamp* algorithm was employed. To introduce the alignment calibration, this algorithm relies on the *ApplyCorrections* function (reported in Appendix A.4): that code was developed to introduce some of the misalignment parameters previously estimated, in addition to the *Dark Current* and *Beam profile* corrections. In this particular case the only parameters we considered are the *skew* and the two shifts in the $\vec{x}$ and $\vec{z}$ directions. That decision will be justified within the results *Discussion* section. To apply the first modification it was simply exploited the *imrotate* function, that rotates the image keeping the matrix dimensions unchanged and realizing a nearest neighbor interpolation. In regard to the translation parameters d_x and d_z, once we know their estimates it is possible to convert those numbers in a quantity of pixels. Those quantities represent the amount of rows and columns we have to remove from one side of the matrix and attach to the opposite side. This operation has the effect of centering the matrix, according to the position of the ideal system of coordinates. Obviously it is necessary to know in which direction we have to "move" the image; to do that we refer to the system's comparison already showed in Figures 2.11 and 2.12.

In Figure 2.15 a photograph of the flat surface of the test phantom is reported.

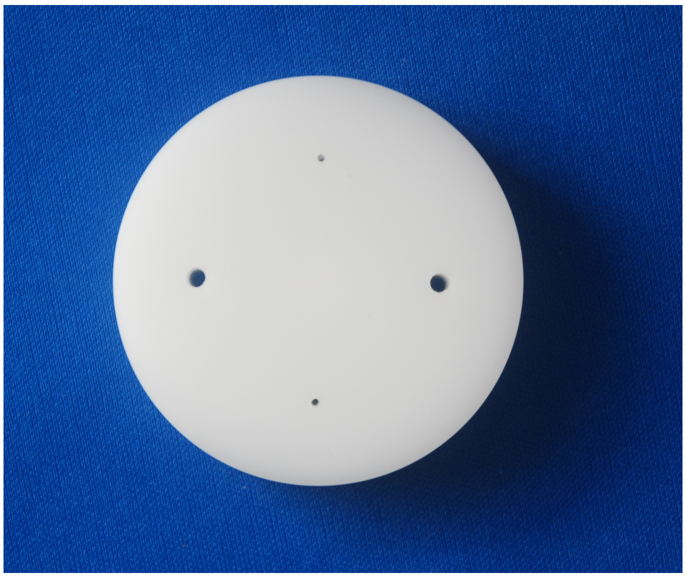


Figure 2.15: Photograph of the flat surface of the cylindrical plastic phantom used for the reconstruction tests. All the four holes are visible.

2.3 Results

The following section presents the results obtained from the *Gate* simulated system, the identification of the misalignment parameters from the real CT device, as well as the corrections applied to the reconstructed images.

GATE Simulation

The first examples we are going to consider are two borderline cases specifically conceived to test the effectiveness of the simulated system and the precision of the identification codes. The first framework presents a perfectly aligned detector, which means the X-ray source can be found on the straight line that perpendicularly intersects the panel surface in its center. As a consequence, all the calibration parameters (both translations and rotations) are set to zero. On the other hand, the second system was conceived as a worst case scenario, where all the misalignment parameters are substantially different from zero. The detector was translated in both d_x and d_y directions (which correspond to d_x and d_z in the theoretical system) of approximately half a millimeter; then it was rotated around all the axes of 1 degree each time. In order to have comparable results, all the other geometric specifications, described in the *Method* paragraph, were kept constants for all the following cases. Furthermore, in these two particular experiments, 45 projections were produced over 360 degrees, meaning an image is collected every step of 8 degrees.

The comparison between the two sets of elliptical trajectories are shown in Figures 2.16 and 2.17.

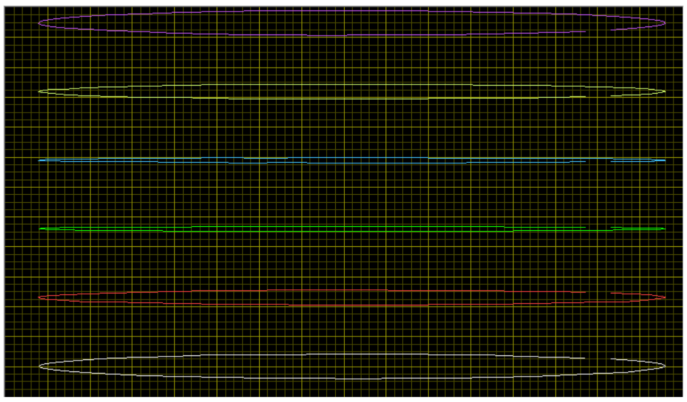


Figure 2.16: The orbits used by the LabVIEW code to identify the calibration parameters. Perfectly aligned simulated case.

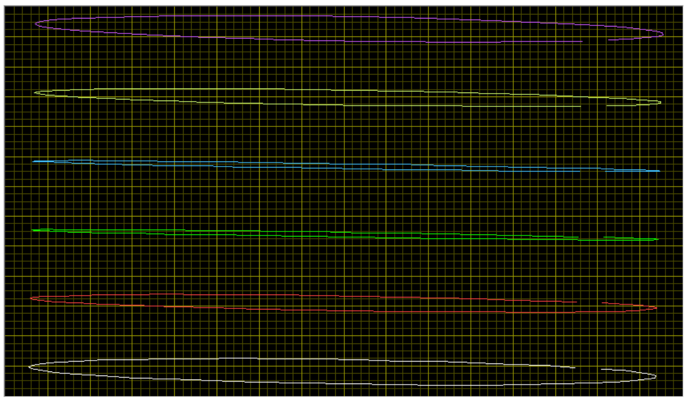


Figure 2.17: The orbits used by the LabVIEW code to identify the calibration parameters. Misaligned simulated case.

These orbits are created by the *LabVIEW* code as an intermediate stage to the parameters estimate. The images present the trajectories covered by the projected shadow of each metal ball, when the simulated gantry rotates around the cylindrical phantom. In fact, these orbits are calculated from the summed image of the whole set of projections. These images, corresponding to the previous Figure 2.16 and 2.17, are illustrated in the following Figure 2.18.

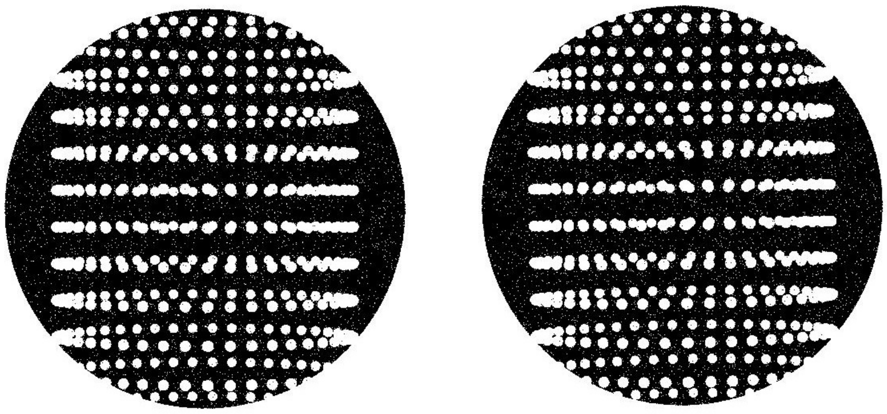


Figure 2.18: Summed images of 45 simulated projections of the calibration phantom: perfectly aligned case (left) and misaligned case (right).

Considering again the Figures 2.16 and 2.17, a more accurate observation can show a slight difference between the two sets. Actually, the ellipses in the first one appear to have a higher symmetry and to better follow the underlying grid. On the contrary, the elliptical trajectories of the second one seem to be more unbalanced in the left-right direction, showing a certain clockwise rotation with respect to the first image. This asymmetry is caused by the misalignment introduced in the detector's position. In particular we can assume the skew is the most visible effect in this case, since it's the movement that produces the rotation around the axis orthogonal to the detector's surface.

Tables 2.1 and 2.2 present the misalignment parameters identification results from the two simulations just discussed. The first row reports the geometrical settings of the system (*Ground Truth*), while in the second and third row the estimated values are shown, using the *RunFullCalculation* and the *RunFullCalcWTilt* respectively. Every parameter's estimate was calculated separately for the 6 trajectories considered from a single phantom. Then, these 6 sets were averaged to obtain the mean values reported in the tables. All the quantities related to a distance (R_F, R', R'_y, d_x and d_z) are expressed in mm, while the three rotations are considered with a degree notation.

Table 2.1: Calibration parameters identified: perfectly aligned system

	R_F	R'	Magn.	R'_y	dx	dz	η	θ	ϕ
G. Truth	284.0	324.0	1.14	324.0	0	0	0	0	0
RFC	283.8	323.9	1.14	323.9	-0.00	0.00	0.00	-	-
RFCWT	285.9	323.9	1.13	323.9	-0.00	0.00	0.00	-1.01	-0.08

Table 2.2: Calibration parameters identified: misaligned system

	R_F	R'	Magn.	R'_y	dx	dz	η	θ	ϕ
G. Truth	284.0	324.0	1.14	324.0	**0.45**	**0.55**	**1.00**	**1.00**	**1.00**
RFC	283.9	323.7	1.14	323.7	0.44	0.59	1.02	-	-
RFCWT	284.2	323.7	1.14	323.7	0.44	0.58	1.02	-2.22	-0.95

Another group of similar experiments will be now presented. These tests were performed to better investigate the capability of the two codes to identify the three misalignment rotations. Furthermore, it was of particular interest to better understand the relationship between each rotation and the estimate of all the other parameters. This was especially true for the *RunFullCalcWTilt* code, where all the parameters relies on the *tilting* angle *theta*. For these reasons, 4 different simulations were produced: three of them present a spatial rotation around a single axis (every time a different one was chosen), while the last one considers the three of them all together. For all of the cases presented, a constant translation movement in the d_x and d_z directions was considered. All the other experimental specifications were exactly the same as in the previous simulations, except for the higher sampling rate used:

in this case an image was collected every 2 degrees, meaning a total of 180 projections.

Table 2.3: Calibration parameters identified: skew rotation

	R_F	R'	Magn.	R'_y	dx	dz	η	θ	ϕ
G. Truth	284.0	324.0	1.14	324.0	**-1.31**	**0.81**	**1.00**	0	0
RFC	284.0	323.9	1.14	323.9	-1.31	0.82	1.00	-	-
RFCWT*	287.3	323.9	1.13	324.0	-1.31	0.82	1.00	0.31	0.06

Table 2.4: Calibration parameters identified: tilt rotation

	R_F	R'	Magn.	R'_y	dx	dz	η	θ	ϕ
G. Truth	284.0	324.0	1.14	324.0	**1.31**	**0.81**	0	**1.25**	0
RFC	283.8	324.2	1.14	324.2	1.31	0.83	0.00	-	-
RFCWT*	284.6	324.2	1.14	324.3	1.31	0.80	0.00	6.82	0.09

Table 2.5: Calibration parameters identified: slant rotation

	R_F	R'	Magn.	R'_y	dx	dz	η	θ	ϕ
G. Truth	284.0	324.0	1.14	324.0	**1.31**	**0.81**	0	0	**1.25**
RFC	283.8	324.1	1.14	324.1	1.30	0.82	-0.00	-	-
RFCWT	283.9	324.1	1.14	324.1	1.30	0.82	-0.00	0.00	-1.29

Table 2.6: Calibration parameters identified: completely misaligned system

	R_F	R'	Magn.	R'_y	dx	dz	η	θ	ϕ
G. Truth	284.0	324.0	1.14	324.0	**1.31**	**0.81**	**0.75**	**1.25**	**1.00**
RFC	283.8	323.8	1.14	323.7	1.30	0.84	0.77	-	-
RFCWT	283.9	323.8	1.14	323.7	1.30	0.84	0.77	-1.50	-1.04

The tables 2.3, 2.4, 2.5 and 2.6 collect the results from the *LabVIEW* identification in the four different conditions discussed. An important feature needs to be highlighted: when the RFCWT row is marked with an asterisk, that means one or more of the 6 parameter's estimates failed. This outcome is caused by a miscalculation in the equation 2.10, the one that should estimate the value of *theta*. In particular, the error originates from an argument of the *arcsine* greater than 1. In these cases the code simply returns a *NaN* instead of a number for the θ, as well as for all the parameters that relies on it to be estimated. Then, in these particular cases, the mean values shown were calculated considering all the other trajectories correctly estimated.

Device parameters identification

Once the parameters' identification codes were tested through the simulated system, they were ready to be applied to the real CT device. With this aim, a 360° acquisition of the calibration phantom was performed: 60 projected images were collected, that means one every 6 degrees. The *tube voltage* was set to 45 kV, the *tube current* to the consistent value of 800 μA and the *frame time* to the standard value of 1000 ms, with half of it employed by the radiation *pulse*. These specification where chosen to provide well-contrasted images, which represents a great advantage in the following trajectory analysis. A few examples of what an enhanced projected image of the calibration phantom looks like are provided in Figure 2.19.

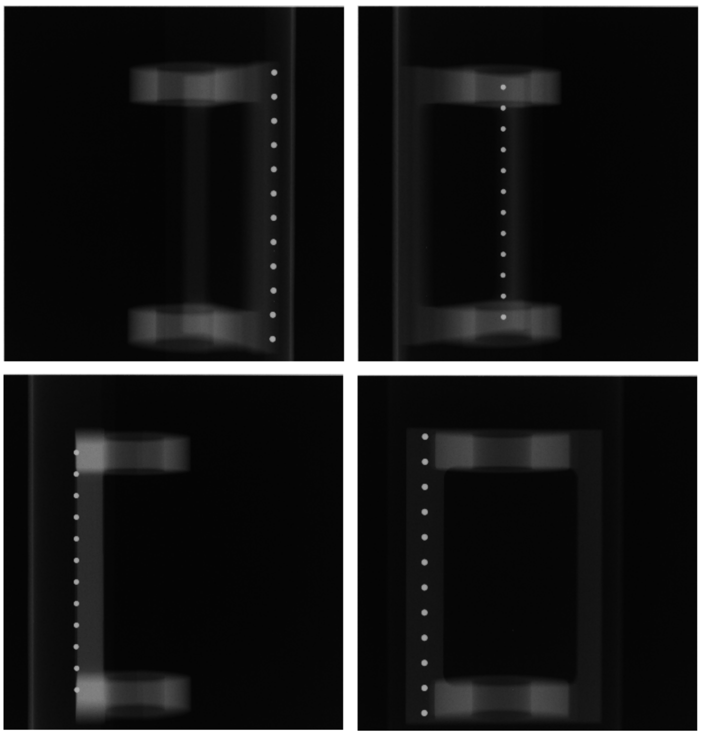
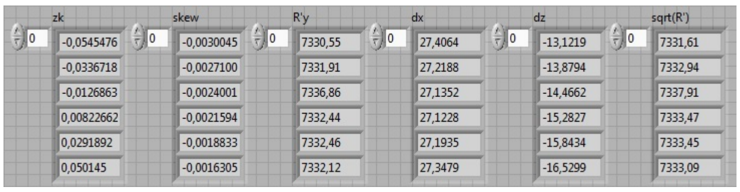


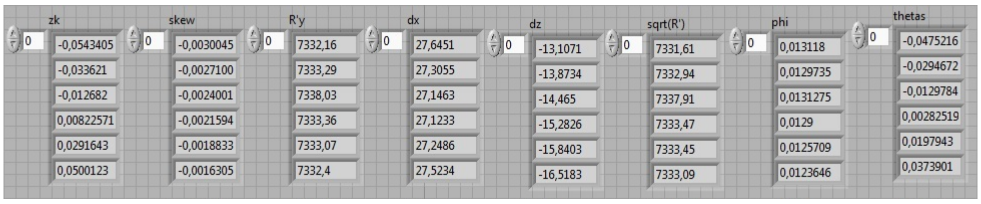
Figure 2.19: 4 projected images of the calibration phantom.

The trajectories of 6 balls, extracted from this set of images and provided to the two *LabVIEW* codes available, produce the following parameters estimates, illustrated in Figures 2.20 and 2.21.



```
     zk                skew              R'y             dx               dz               sqrt(R')
0    -0,0545476   0    -0,0030045   0    7330,55    0    27,4064    0    -13,1219    0    7331,61
     -0,0336718        -0,0027100        7331,91         27,2188         -13,8794         7332,94
     -0,0126863        -0,0024001        7336,86         27,1352         -14,4662         7337,91
     0,00822662        -0,0021594        7332,44         27,1228         -15,2827         7333,47
     0,0291892         -0,0018833        7332,46         27,1935         -15,8434         7333,45
     0,050145          -0,0016305        7332,12         27,3479         -16,5299         7333,09
```

Figure 2.20: LabVIEW output of the RunFullCalculation code for the real CT device geometrical calibration.



```
     zk                skew              R'y             dx              dz               sqrt(R')        phi               thetas
0    -0,0543405   0    -0,0030045   0    7332,16    0    27,6451   0    -13,1071    0    7331,61    0    0,013118     0    -0,0475216
     -0,033621         -0,0027100        7333,29         27,3055         -13,8734         7332,94         0,0129735         -0,0294672
     -0,012682         -0,0024001        7338,03         27,1463         -14,465          7337,91         0,0131275         -0,0129784
     0,00822571        -0,0021594        7333,36         27,1233         -15,2826         7333,47         0,0129            0,00282519
     0,0291643         -0,0018833        7333,07         27,2486         -15,8403         7333,45         0,0125709         0,0197943
     0,0500123         -0,0016305        7332,4          27,5234         -16,5183         7333,09         0,0123646         0,0373901
```

Figure 2.21: LabVIEW output of the RunFullCalcWTilt code for the real CT device geometrical calibration.

In these tables, each line corresponds to a single orbit. Moreover, it should be noted that all the distance values are pixel-size based. That means they have to be multiplied by 0.05 mm (the side length of a pixel) to be expressed in mm. Similarly, all the rotation angles need to be converted from radians to degrees. The averaged results of the identification are illustrated in Table 2.7.

Table 2.7: Misalignment parameters of the real CT device, averaged from the 6 elliptical trajectories considered and expressed in *mm* and degrees.

	R_F	R'	Magn.	R'_y	dx	dz	η	θ	ϕ
RFC	286.6	366.7	1.28	366.6	1.36	-0.74	-0.13	-	-
RFCWT	287.5	366.7	1.28	366.7	1.37	-0.75	-0.13	-0.21	0.73

Reconstruction correction

The results just presented have been tested on a reconstructed image to check the effect of the misalignment artifact and the efficacy of the correction employed. In particular, the three parameters d_x, d_z and η were considered. Taking into account the relationship between the different systems of coordinates, the two translations consists in a movement towards the upper side of the matrix of 27 pixels, and towards the left side of 15 pixels. This operation should bring the center of the image back to its ideal position. But, before the translations, a counterclockwise rotation of 0.13° was performed. These corrections were individually applied to every frame acquired, before the reconstructed image was created.

The acquisition protocol performed presents a high sampling rate (400 projected images over 360°, with steps of 0.9°) and a high tube current, equal to 800 μA. These specifications were chosen to maximize the reconstructed image contrast and spatial resolution. The phantom employed is the cylindrical one presented in the *Method* section and already illustrated in Figures 2.14 and 2.15.

In Figures 2.22 and 2.23 two complete 3D reconstructions are presented. Both of them have a spatial resolution of 480x480x480 voxels, and three cross-sections according to the system of coordinates are showed. The first reconstruction was performed without any alignment correction, while in the second one the three parameters discussed were applied.

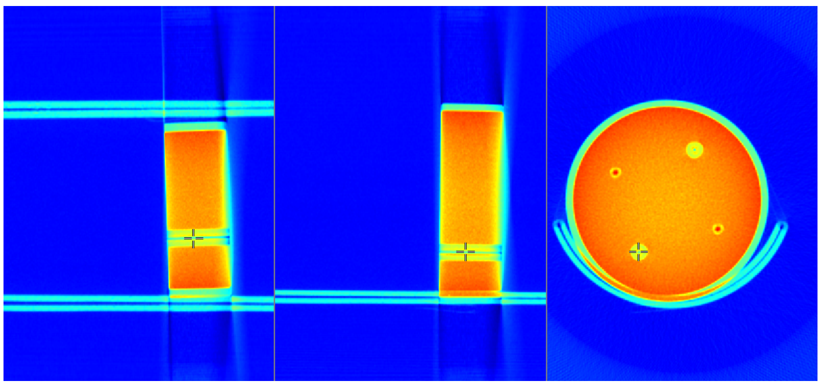


Figure 2.22: Cross-sections of the 3D reconstruction without any misalignment correction, displayed with a 480x480 pixels matrix.

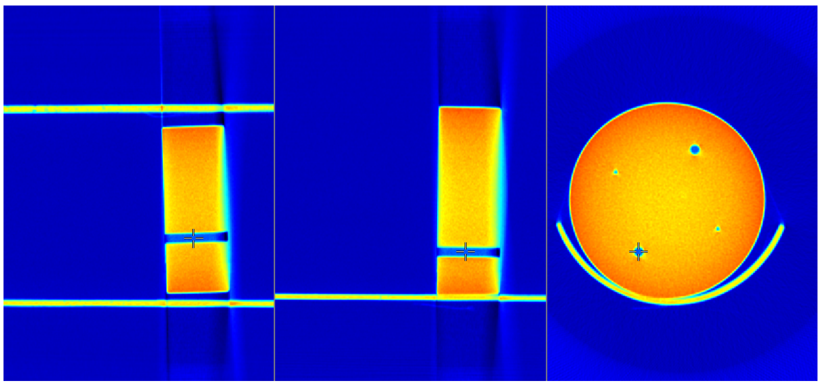


Figure 2.23: Cross-sections of the 3D reconstruction with the d_x, d_z and skew corrections, displayed with a 480x480 pixels matrix.

In the following figures, a more detailed view of the same phantom is illustrated. All of these images report the same cross-section, which is a surface perpendicular to the axis of rotation. Moreover, they present a spatial resolution of 1200x1200 pixels. The Figure 2.24 shows the first case, where no corrections are applied.

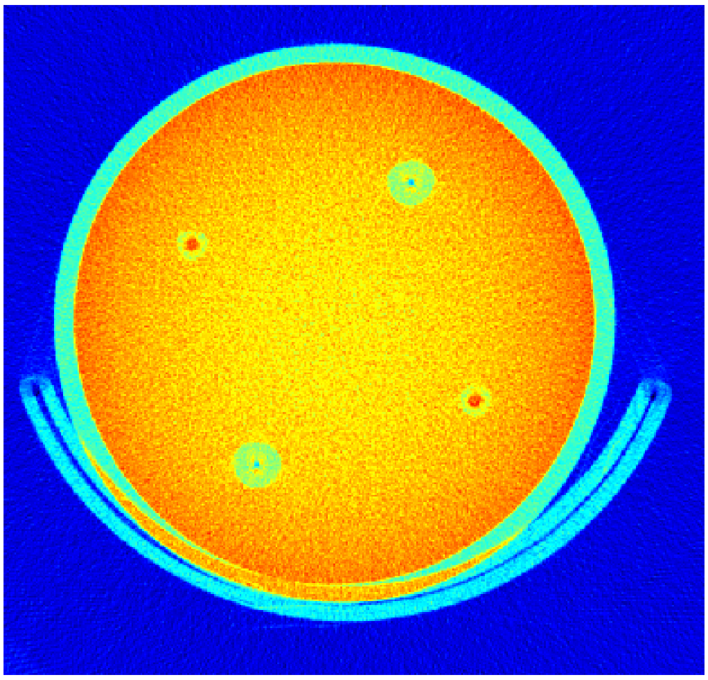


Figure 2.24: Cross-section of the phantom 3D reconstruction, without any misalignment correction applied.

In Figure 2.25 three partial corrections are considered. From left to right: only the d_x, only the d_z and both the translations are respectively employed to correct the images.

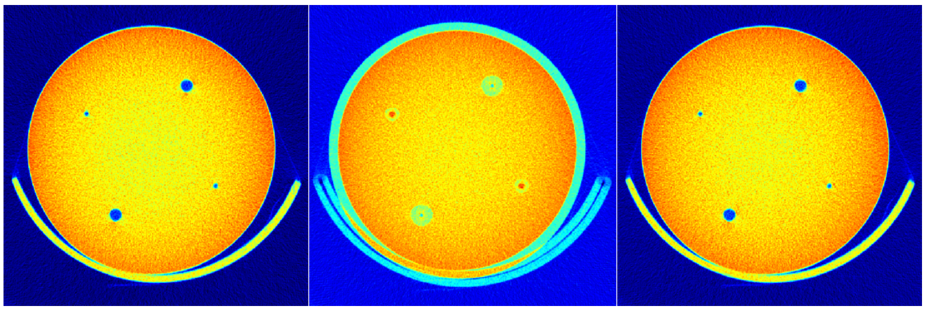


Figure 2.25: Three different reconstruction from the same acquisition. From left to right, the corrections applied are: d_x translation, d_z translation, both translations.

In the end, the complete set of geometric corrections is applied to reconstruct the section of Figure 2.26.

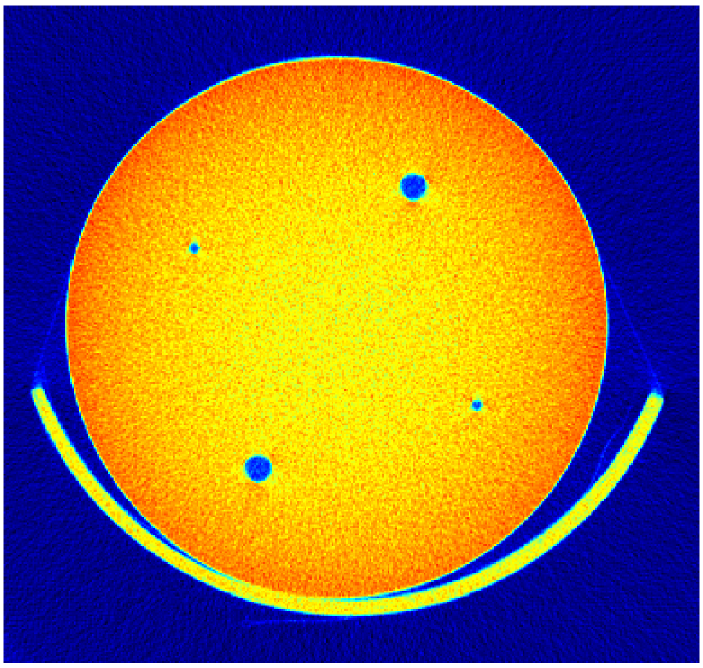


Figure 2.26: Cross-section of the phantom 3D reconstruction, corrected with the d_x, d_z and η estimated parameter.

2.4 Discussion

Some of the steps that led to the results just illustrated need a deeper evaluation to better understand what is the real effect of the misalignment correction performed, which is the best protocol to apply in a future calibration and which are the weak points of the method employed.

The first discussion concerns the results coming from the simulated tests. The estimated parameters seem to be sufficiently close to those set by the simulation software in all the misalignment situations considered. In particular, the two codes (*RunFullCalculation* and *RunFullCalcWTilt*) always supply very similar values for all the parameters they share, which include the linear translations, the skew and the source-center-detector distances. Anyway, a slight lack of precision doesn't allow to obtain a perfect matching with the expected values. This variance might find several explanations to be justified, and most of them are caused by computational issues of the simulation. First of all, the projected images are shot with a very low exposure: the contrast, in terms of photons collected, between a pixel hit by the beam and one shadowed by a phantom's ball is just of 7.3 photons on average. Secondly, the spatial definition of the simulated detector (defined by the pixel's dimensions), is quite poor, actually more than 6 times lower than the real panel. Thirdly, the number of projections acquired is modest, especially in the first two test performed, with only 45 images collected over 360 degrees. All these issues summed together might contribute to cause a poor definition of the balls' projected shadows. As a consequence, it is more difficult for the calibration software to reliably find the elliptical trajectories, and then to correctly estimate the misalignment parameters. Fortunately, most part of these issues are related only to the simulation framework, and do not happen in the real device, that exhibits much higher contrast and spatial resolution.

The second issue to be discussed concerns the two additional parameters estimated by the RFCWT calibration code, that are the *tilt* θ and the *slant* ϕ. First of all, we consider their sign: in almost all the tests where those two rotations were set to an angle different from zero, the output values appear to have the opposite sign. The regularity of this outcome, together with the matching of some *slant* absolute values, suggest that the sign of those parameters was simply inverted. That might be caused by a mismatch of the two systems of coordinates, that affect the transition from the simulated to the theoretical one. Anyway, since the real device has its own system of co-

ordinates (which was already completely described at the end of the *Method* section) this mistake shouldn't have implications for the successive use in the real case.
In addition to that issue, the precision of these two estimates seems to be lower if compared with all the other parameters. In particular, the estimated ϕ exhibits a deviation from the real value of maximum 5% (when the parameter was set to a non-zero number) and appears to oscillate between -0.1 and 0.1 when it was set to zero. On the other hand, the θ estimate exhibits an unreliable behaviour, with values sometimes completely wrong. Furthermore, those estimates are affected by another problem already outlined in the *Background*. When the *slant* value is really close or equal to zero, the algorithm might be unable to produce an estimate for *theta*. Indeed, this happened in the two tests where ϕ was set to zero: some of the results from the RFCWT code were NaN, revealing the miscalculation.
Anyway, the poor reliability of the *tilt* evaluation does not seem to affect the parameters' estimates that rely on it. Indeed, even when the value of θ is completely wrong (as in Table 2.4 for example), the calibration succeeded with all the other parameters. This fact, together with the negligible effect of the *tilt* itself on the trajectory shape, seem to legitimize the effectiveness of the simplification adopted in the *RunFullCalculation* algorithm, where θ is forced to zero.

For many reasons, in the following calibration of the reconstructed images, not all the available misalignment parameters are employed. First of all, we might easily assume that the tilting angle of the real panel is, to a good approximation, really close to zero. Secondly, since also the slanting angle can be considered almost null, we know that the algorithm will probably produce, for the aforementioned reasons, an unreliable estimate of θ. Thirdly, it was demonstrated that, even for large misalignment angles, the tilt and slant have a negligible impact on the elliptical trajectories [6]. Lastly, the implementation of these two corrections would need some matrix transformations much more complex than those employed for the skew rotation. For all these reasons, the only calibration parameters to be actually applied were the linear translations and the skew (in addition to the device dimensions R_F and R'_y).

We are now finally ready to discuss the correction implementation and evaluate the effects on the reconstructed images. We will especially focus on the cross-sections showed in Figures 2.24, 2.25 and 2.26, since they exhibit the highest spatial definition available. Comparing the first image, which was produced without any geometric correction, with the final reconstruction, the

misalignment artifact and the calibration effect are quite evident. The wrong placement of the detector clearly produces an inefficient reconstruction of the phantom shape. In particular, the detection of all the edges appears to be highly imprecise, with the presence of a large stripe (coloured with a light blue) where a sharp border is instead expected. This phenomenon can be observed around the phantom as well as where the bed is located. Moreover, the misalignment artifact affects the detection of the four holes: their location appears to be ambiguous, and their dimensions and shape are badly defined. Lastly, in the lower part of the frame, the two volumes of the cylinder and the carriage cross each other, producing an unreal intersection.
On the contrary, the correction applied to obtain the final reconstruction shown in Figure 2.26 seems to fix most part of the aforementioned artifacts. The cross-section illustrated exhibits sharp edges, precise volumes and a high contrast of the objects against the background. Moreover, many tiny details appear to be much more well-defined, as the accurate shape of the four holes and even the distinguishable presence of the adhesive tape that links the carriage to the phantom.
The intermediate steps of the reconstruction correction illustrated in Figure 2.25 show the partial effects of each single parameter on the final result. It is of particular interest to observe that the enhancement of the image quality is mostly due to the d_x translation. Indeed, even applying only that parameter, the resulting reconstruction is quite similar to the final one. Moreover, the d_z and the η seem to just slightly improve the sharpness of the edges and the global contrast of the image. After all, for what concerns the skew, its estimated value is quite small: 0.13 degrees of rotation in the center of the matrix correspond to less than 3 pixels of translation on the border of the image.
In the end, a dimensional evaluation of the reconstructed image can help to verify the effectiveness of the R_F and R'_y identified parameters. In particular the size of two holes and the diameter of the cylinder were checked; the reconstructed length are 10 pixels, 23 pixels and 510 pixels respectively. Since the visualization set a 2x2 binning for a total of 1200 pixels for each side, each pixels has a side dimension of 0.1 mm. That means the previous length are actually 1 mm, 2.3 mm and 51 mm, and all of them prove to coincide with the real dimensions of the phantom.

Chapter 3

Detector lag effect artifact correction

3.1 Background

Among all the artifact sources that corrupt the reconstructed image quality, the cone-beam computed tomography (CBCT) realized with a CMOS digital flat panel is affected by a characteristic phenomenon known as *Lag Effect*. This detector lag consists of some residual signal which is revealed in the actual frame even if it was created in a previous one [12]. That leads to some wrong pixel counts that might continue for many frames successive to the one they should be related to. This effect produces some dark and bright shades on the projections, that can cause a range of severe artifacts in the reconstructed image.

This problem is strictly related to different aspects [13]. First of all, the kind of flat panel adopted by the CT system, meaning its particular X-ray detection technology, plays a fundamental role in the lag creation and evolution. Furthermore, the protocol employed during an image acquisition in standard conditions might considerably influence the phenomenon behaviour. These aspects will be discussed in the following sections.

Detector structure and acquisition protocol

The flat panel sensor employed (*Hamamatsu C7942CA-22*) is an indirect photon detector panel; that means a scintillator crystal is coupled to a two-dimensional photodiode array [14]. In this configuration the scintillator plate is made of cesium iodide (CsI) grown with a needle shape onto a glass surface, that is a low X-ray absorption material. The crystal's role is to detect the incident X-rays and to convert them into light photons. The fluorescence is then channeled through the needle where the scintillation happened and then transmitted to the corresponding pixel. This fiber-like crystal structure allows an efficient light propagation. That particular arrangement is called *flipped scintillator plate* (FSP) and a cross-section of that device is illustrated in figure 3.1.

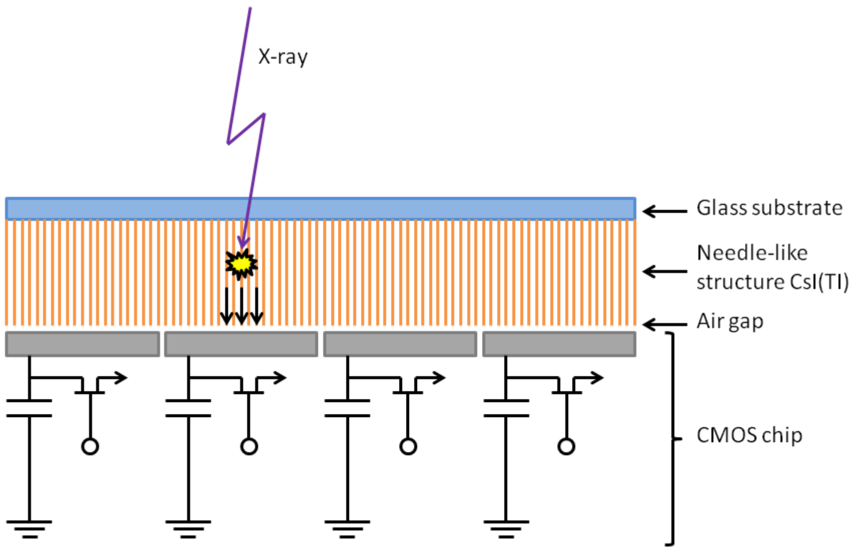


Figure 3.1: Cross-section of a flipped scintillator plate

Under the needle structure, we find a solid-state image sensor made of a photodiode matrix, which is implemented in a standard CMOS process [15]. In each pixel, when the silicon is hit by a light photon, an electric charge is collected in the junction capacitance. That accumulation is proportional to the scintillation strength, and then to the incident x-ray intensity. In particular, the energy supplied by the light photon creates a electron-hole pair in the photodiode silicon substrate [12]. When the acquisition is completed, the circuit chip is ready for the readout step where the capacitance is cleaned and the information organized by the underlying control board. The analog data is now converted into a 12-bit digital signal and transferred outside the flat panel.

Considering a standard acquisition protocol, the operation period is divided into well-defined cycles [16]. These intervals' length is called *frame time*, and it represents the temporal unit of an acquisition procedure. Each period is repeated several times, and all the acquisition tasks have to be accomplished within that limited interval. In particular, the second half is dedicated to the panel read-out, which means the electric charge, collected during the X-ray exposure, is removed by the control circuit and the pixel sensor is *almost* emptied. As a consequence, only the first half of the period is free to be employed to collect the signal. This interval's length is called *pulse width*, and it corresponds to the time the X-ray beam is kept active.

Lag effect origin

As it was previously introduced, the effect we are going to discuss originates from the particular behaviour of the silicon layer in the flat panel sensor. In a pure crystalline silicon detector, the atomic structure of the material exhibits a rigid disposition of the molecules with stable chemical bonds between them [15]. That characteristic structure heavily conditions the sensor behaviour. Indeed, the band energy diagram assumes the particular shape shown in Figure 3.2. The diagram represents the energy states distribution of the silicon electrons: in this ideal condition, all the available states in the *valence band* (the one with lower energy) are occupied, while all the states in the higher energetic *conduction band* prove to be empty. In the middle there is a gap that clearly divides the two energetic bands. The electrons contained in the valence band are tightly held by the rigid atomic structure; anyway, when one of them receives a sufficient amount of energy (that means at least equal to the energy gap), it can *jump* to the valence band, where there are no bonds to hold them, and then they can move freely [12]. An important aspect concerns the origin of the energy supplied: the previous phenomenon does not depend on the particular energy source, that can be either thermal or coming from a scintillation photon.

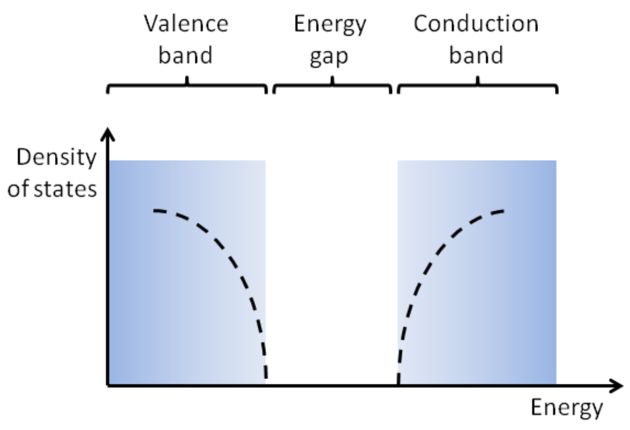


Figure 3.2: Energy states distribution of the electrons in the ideal crystalline silicon.

Anyway, the real detector does not present an ideal behavior, since its atomic composition is far from the perfectly rigid network previously discussed. Indeed, many defects and wrong bonds between neighbor silicon molecules introduce in the material structure a certain level of randomness. This condition modifies the electron energy states diagram. The more chaotic distribution of the atomic bonds creates a greater variety of energetic states. That means the valence and conduction band are now not so well separated: both of them present a spread towards the energy gap zone. As a consequence, we now find a certain amount of energy states which can be occupied by electrons, and that do not belong neither to the valence nor the conduction bands. These particular energetic states that the electrons can assume are called *traps*. In Figure 3.3 it is illustrated the new energy states diagram, where both bands show a characteristic tail and there is a defect states distribution approximately in the middle of the energy gap. During the device operations, a certain amount of charge can eventually fill those trap states, and then reduce the performance of the detector and originate the lag effect [12].

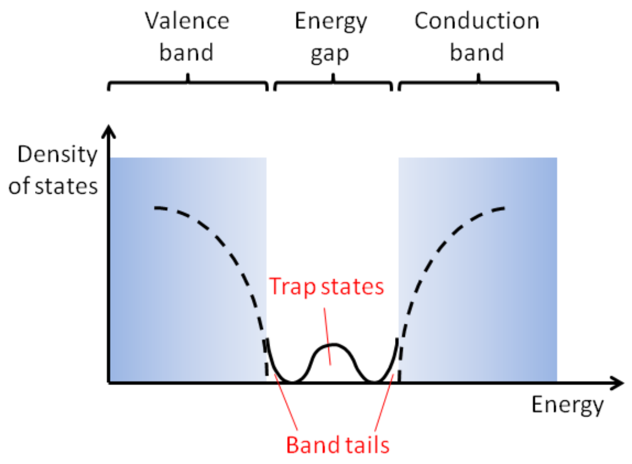


Figure 3.3: Energy states distribution of the electrons in a real crystalline silicon. The energy gap band is characterized by a trap states distribution

Several works have studied and found different origins for the lag effect [17] [13]; anyway, the trapping and detrapping of electrons during the detector operations was found to be the principal cause of lag effect. In particular the quantity of available states and the rate of charge deposition and release produces the typical ghost images. Those characteristics are defined by several parameters and affected by different conditions of the sensor, as well as the history of the panel. This unwanted effect produces a quality image corruption and a loss of temporal resolution of the CT device. A typical condition where lag effect appears is during a constant irradiation of the panel. In this situation, looking at the output image evolution, we can see an overall increasing trend of the number of photons detected. This growing signal effect is caused by the trap states releasing electrons during successive frames, those that follow the one in which charge was caught. That growth reaches a steady state only after a certain time period, when the number of electrons trapped at each frame equals the quantity of charge released in the same frame. That rise of the detector gain over time is known as *rising step-response function* (RSRF). Similarly, the lag effect can be clearly observed even after the constant irradiation has ended. Indeed, if a number of frames continues to be read after the X-rays have been shut down, the detector will acquire and show some delayed photons released by trap states a bit at a time. That exponential decreasing trend is then called *falling step-response function* (FSRF).

Lag effect software correction

All the following discussion and results are widely based on the analysis that Starman [12] made about this effect, with a different detection technology but similar acquisition conditions. Our goal is now to define a possible algorithm to compensate for the lag effect that is found on the projections of our CT device. With this aim, it is necessary to look for a model that describes how the acquired detector signal decays throughout a sequence of frames [18]. In this case the falling trend was supposed to be the sum of a few exponential functions [12]. In other words, the signal acquired in a single projection produces an effect on all the following images, which always decreases with rate proportional to the actual value. Therefore, from another point of view, every image is affected by all the previous frames acquired, with increasing importance as it gets closer to the current image. That means every single projection can be considered as a weighted sum of all the past images, plus the actual true one [19], [20]. Then we might try to correct each projection subtracting the estimated amount of signal that does not belong to the current frame. To realize this idea, first of all the *discrete-time* decay function has to be defined as follows:

$$I_k = \sum_{n=1}^{N} b_n I_0 e^{-a_n k} \qquad \text{for} \qquad k = 1, 2, ... \tag{3.1}$$

where I_0 (that stands for I at time $t = 0$) is the current true lag-free image, I_k are the lag images following the first one, k is the index of the discrete time sequence, N is the number of exponential components that form the decay function, b_n are the lag proportional coefficients (referred to the n-th exponential) and a_n are the n-th lag exponential rates. If we consider a single exponential, the equation gets simplified:

$$I_k = bI_0e^{-ak} \qquad \text{for} \qquad k = 1, 2, ... \tag{3.2}$$

Now, considering for the moment this simplified version as a matter of convenience, we show an example of how to find the relationship between a corrected image X_k (that is $X(t{=}k)$) and all the true frames Y_j for $j = 1, 2, ..., k$. We refer to the situation shown in Figure 3.4, where the first 4 frames of a hypothetical CT are illustrated. The phantom here considered is a slice of high contrast material with a hole in it that changes its position from the upper left to the bottom right corner as frames pass. The first row shows the real acquired projections, while in the second row we find the corrected lag-free frames.

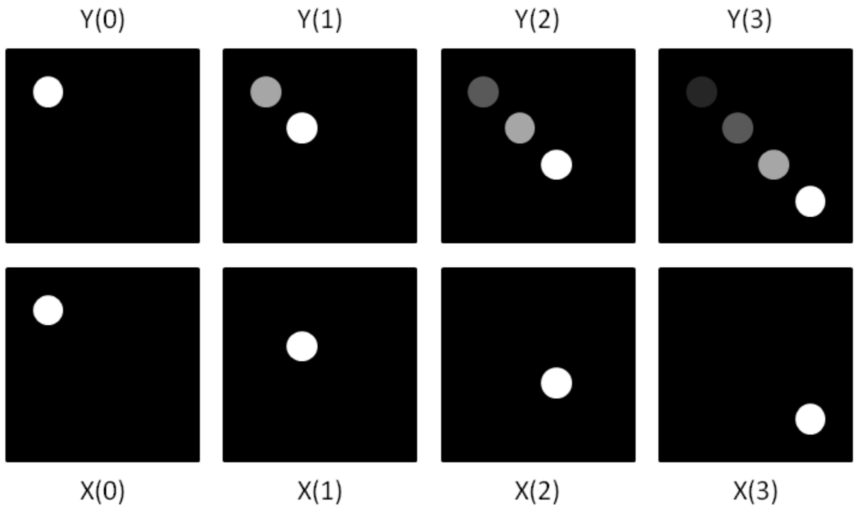


Figure 3.4: Example diagram of 4 consecutive frames.
First row (Y): acquired images (with lag effect).
Second row (X): lag-free images (after correction).

At the first step (t=0, left column of Figure 3.4), we can easily deduce the following relationship:

$$X_0 = Y_0$$

Then, at t=1 (second column of the example), the correction needed is:

$$\begin{aligned} X_1 &= Y_1 - bX_0e^{-a} \\ &= Y_1 - bY_0e^{-a} \end{aligned}$$

With regard to the third column of our hypothetical situation (t=2):

$$\begin{aligned}X_2 &= Y_2 - bX_1e^{-a} - bX_0e^{-2a}\\ &= Y_2 - b\Big[Y_1 - bY_0e^{-a}\Big]e^{-a} - bY_0e^{-2a}\\ &= Y_2 - bY_1e^{-a} - (1-b)bY_0e^{-2a}\end{aligned}$$

At t=3, after all the substitutions needed, the relationship between X_3 and Y_0, Y_1, Y_2 and Y_3 is:

$$\begin{aligned}X_3 &= Y_3 - bX_2e^{-a} - bX_1e^{-2a} - bX_0e^{-3a}\\ &= Y_3 - b\Big[Y_2 - bY_1e^{-a} - (1-b)bY_0e^{-2a}\Big]e^{-a}\\ &\quad - b\Big[Y_1 - bY_0e^{-a}\Big]e^{-2a} - bY_0e^{-3a}\\ &= Y_3 - bY_2e^{-a} - (1-b)bY_1e^{-2a} - (1-b)^2bY_0e^{-3a}\end{aligned}$$

From those examples it is possible to infer the general form of the equation:

$$X_k = Y_k - \sum_{j=1}^{k}(1-b)^{(j-1)}bY_{(k-j)}e^{-ja} \tag{3.3}$$

That in the multi-exponential version is

$$X_k = Y_k - \sum_{j=1}^{k}\sum_{n=1}^{N}(1-b_n)^{(j-1)}b_nY_{(k-j)}e^{-ja_n} \tag{3.4}$$

Anyway, this correction algorithm won't be used in this form, since it is too computationally demanding. Indeed, to calculate every single true image X_k it is necessary to store all the frames Y_j and to always exploit all of them. Starting from the situation already considered, it is possible to attain a *state model*, introduced by J. Hsieh [12,21],that carries out the same results as the previous algorithm with much less computational complexity. That model is composed by two equations: the first one calculates the current true image X_k relying on the actual frame Y_k and the state variable S_k; the second one is the state updating equation, that finds the new version of the state S_{k+1} exploiting only the current true image X_k and the previous state variable S_k. The equations system, together with a brief example about where the state model comes from, is reported in the following.

At t=0, the corrected frame and the state variable are:

$$X_0 = Y_0$$

$$S_0 = 0$$

After one frame, at t=1:

$$\begin{aligned} X_1 &= Y_1 - bX_0 e^{-a} \\ &= Y_1 - bS_1 e^{-a} \end{aligned}$$

$$\begin{aligned} S_1 &= X_0 \qquad \text{that is} \\ S_1 &= X_0 + S_0 e^{-a} \end{aligned}$$

At t=2, the equations of the model are:

$$\begin{aligned} X_2 &= Y_2 - bX_1 e^{-a} - bX_0 e^{-2a} \\ &= Y_2 - b\Big[X_1 + X_0 e^{-a}\Big] e^{-a} \\ &= Y_2 - b\Big[X_1 + S_1 e^{-a}\Big] e^{-a} \\ &= Y_2 - bS_2 e^{-a} \end{aligned}$$

$$S_2 = X_1 + S_1 e^{-a}$$

After three steps, the relationships can be expressed as:

$$\begin{aligned} X_3 &= Y_3 - bX_2 e^{-a} - bX_1 e^{-2a} - bX_0 e^{-3a} \\ &= Y_3 - b[X_2 + X_1 e^{-a} + X_0 e^{-2a}\Big] e^{-a} \\ &= Y_3 - b\Bigg[X_2 + \Big[X_1 + X_0 e^{-a}\Big] e^{-a}\Bigg] e^{-a} \\ &= Y_3 - b\Big[X_2 + S_2 e^{-a}\Big] e^{-a} \\ &= Y_3 - bS_3 e^{-a} \end{aligned}$$

$$S_3 = X_2 + S_2 e^{-a}$$

As in the previous case, from these few examples it is possible to infer the behaviour of the two equations of the correction protocol, when expressed in the form of a state model:

$$\begin{cases} X_k = Y_k - bS_k e^{-a} \\ S_{k+1} = X_k + S_k e^{-a} \end{cases} \tag{3.5}$$

The system will be now presented in the general form, that considers more than a single exponential function for the decay model. In this case, the model needs a different state variable for each exponential:

$$\begin{cases} X_k = Y_k - \sum_{n=1}^{N} b_n S_{n,k} e^{-a_n} \\ S_{n,k+1} = X_k + S_{n,k} e^{-a_n} \end{cases} \tag{3.6}$$

3.2 Method

3.2.1 Lag effect visualization

In order to study the behaviour of the lag effect and understand its influence on the projected images, the first task to accomplish is to verify its presence in a typical acquisition test. This will help us to know what is the lag contribution, in terms of digital counts, we have to expect from an image acquired in normal conditions. A standard framework was chosen because our interest is focused on the investigation of the effect we might find in a real acquisition situation, that is the one we will have to compensate with a correction algorithm.

The main parameters that has to be set while we want to define an acquisition protocol are the following [2,16]:

- *tube voltage* [kV]: it's the voltage set between the anode and the cathode to accelerate the electrons released by the heated filament. This parameters modifies the energy distribution of the X-ray beam.

- *tube current* [μA]: it's the current that flows in the X-ray tube, from the cathode towards the anode. It is composed by the electrons released by the cathode, and it is proportional to the quantity of photons we find in the X-ray beam.

- *frame time* [ms]: it's the length of the period needed to accomplish one entire exposure-readout cycle.

- *pulse width* [ms]: as long as the X-rays working mode can be described by a time-limited step function, the pulse width describes the length of that period. That is obviously related to the panel exposure time.

- *binning*: it describes how the pixels can be grouped to increase the dimensions of a single acquisition element. The flat panel employed allows the 1x1, 2x2 and 4x4 binning modes.

- *triggering mode*: it can be *internal* or *external*, and it specifies where the synchronization signal comes from.

- *before/after/number of images*: they represent the quantity of projections collected before, after and during the X-rays emission period respectively.

The phantom chosen for this test was a 5 mm thick lead plate. This material was chosen for its high X-ray absorption capacity. The plate was positioned in front of the panel, attached to its plastic box, covering the central-right portion of the sensitive rectangle. That means half of the expected projections were light field images, where the X-rays could reach directly the detector, without being filtered by any object. On the contrary, the second half of the collected image would be shadowed by the presence of the lead. This arrangement was chosen since it should be an optimal condition to verify the lag effect presence on the detector images. Indeed, the comparison between the two halves of the panel is a borderline case: the left portion receives an unfiltered X-ray beam, while the right part should be hit by just a few photons. As a consequence, a significant lag effect should appear in the left portion, while a simple dark current signal should be observed in the right one.

Since at this stage we were just interested in showing the temporal evolution of the lag effect (and not the reconstruction artifacts it produces), a static acquisition of several projections was performed. In particular, 50 images were collected with the X-rays turned on and, right after these (that is without any delay between the two acquisitions), 50 *after* projections were acquired with the same parameters, but without any exposure. In particular, the protocol was characterized by 45 kV of tube voltage and 800 μA of tube current, while the frame time was set to 1000 ms, with half of it occupied by the pulse width.

When the acquisition was completed, the images were examined: after the application of a correction for the dark current, two *regions of interest* were extracted from each projection. These submatrices were chosen one in the left and the other in the right side of the image to inspect and compare the two different situations described. Furthermore, those ROIs were chosen as close to the center of the image as possible, since it should correspond to the most flat area of the beam profile. In the end, the two regions were compared, first with a simple comparative observation, and then with an averaged signal analysis.

3.2.2 Lag effect signal processing

The experiment we are now going to introduce was specifically created to get a quantitative evaluation of the lag effect behaviour. Since we advanced the hypothesis that the lag decay develops like a sum of exponential components, we are then interested in estimating a function that describes its evolution. In particular, the framework we should consider is the *impulse response function* (IRF), that first has to be adapted to our discrete time system. In this case, the impulse is represented by a single frame exposition to a X-rays light field; then, the decreasing response of the panel to the lag effect, originated by the first projection, is described by the collection of the following dark images. Indeed these frames will contain not only the dark current signal, but also the falling component of the lag, that is the one we are interested in. However, after a few attempts, it came out that trying to reproduce a discrete impulse response function is not the optimal way to characterize the lag effect. Indeed, as it will be later examined in the *Discussion* section, the lag decay trend is way better defined when the panel is hit by a sequence of several X-rays shots. This new framework necessarily modifies the expected lag behaviour, changing its description from an IRF to a *falling step response function.* This model is more suitable for the new protocol, since the sequence of shots play the role of a rectangular step function signal. Anyway, as it will be later discussed, the identified curves will fully describe the lag effect in its general form [22].

Going back to the experiment, it should be suitable to perform a model identification of the equation 3.1. That would allow us to estimate the model parameters, that are the b_n, the a_n and the number of exponential functions N. Once we get them, we might try to check if we can predict the lag effect behaviour, and then implement the state model correction algorithm.

Since we want to examine the lag effect evolution, an important observation we must consider concerns the selection of the acquisition protocol. As we can presume from the lag physical origin, described in the *Background* section, each protocol specification might imply important consequences for the decay development characteristics. That means: changing the tube voltage or current, the frame time or the pulse width, would obviously modify the X-rays exposure and the energetic content of the beam, and then this would almost surely transform the expression of the lag. Anyway, since in this particular case we can already guess the protocol that will be used to acquire and reconstruct the tomographic images, we can then circumscribe and limit the degrees of freedom in the parameters choice. Then, what we are really interested in is verifying if different levels of X-rays exposure modify the effect of the lag. With this aim, the same experiment has been repeated changing only the tube current, from a minimum of 200 μA to a maximum of 800 μA, since it has been demonstrated it is almost directly proportional to the number of photons produced. All the other parameters were set (and maintained constant) to: 45 kV for the tube voltage, 1000 ms of frame time, 500 ms of pulse width, and a 1x1 pixel binning.

To execute this experiment it was chosen to perform a sequence of X-rays shots (*number of images* equal to 50) and to record 50 *after* frames. This protocol should roughly approximate the response to a rectangular step function signal. Furthermore, the acquisition framework has also to take into account a few other important aspects. Firstly, it is necessary to wait for a certain period between two consecutive acquisitions, since it was found out that the lag effect lasts at least 15 minutes [17,23]. Which means that, to be sure the phenomenon is vanished and the detector is back to its initial condition, waiting for a minimum of 30 minutes is advisable in this case. Secondly, all the acquired images need to be corrected, since their true signal is modified by two different effects: the dark current and the beam profile [23].
In particular, the dark current plays an important role in the lag analysis framework; that is because it represents a noisy pixel-dependent measure, which oscillates around a constant value. This signal is always present and it is composed by a significant amount of digital counts that are added to the true image. Moreover, since the lag effect contribution consists of a modest quantity of digital counts, the dark current becomes a non-negligible component, that might hide the effect we are trying to study. In addition, the dark current signal might vary of several digital counts from one acquisition test to the other, since it is influenced by many environmental conditions, as the room temperature and the warm-up process. These reasons might suggest to acquire a dark current image right before each single test, so that it would be always possible to subtract an accurate noise estimate. Anyway, according to the importance of this component in the lag effect treatment, these aspects will be further investigated and discussed.
At the same time, another correction has to be applied to the images: indeed, all the acquisitions are affected by a unequal X-rays distribution of the beam [23]. This means that the area around the center of the detector is always more exposed than those close to the borders. To compensate this effect it is necessary to divide each projection by a light field image of the beam profile. In this way we normalize the acquired projection and then we get a quite uniform exposure on the whole detector surface.

The complete correction is represented by the following equation:

$$I_C = \frac{I - I_{DC}}{I_{LF} - I_{DC}} \tag{3.7}$$

where I is the acquired raw image, I_{DC} is the dark current image, I_{LF} is the light field image and I_C is the corrected one [4].

Dark current characterization

According to the fundamental role that the dark current component plays in the lag effect study, an in-depth characterization appears to be absolutely necessary. Indeed, to be able to apply an aware correction and to perform a lag effect acquisition with the best conditions available, many aspects of this noisy component should be taken into account. For these reasons a few tests will be conducted to analyze the characteristics that influence our acquisitions the most. Anyway, this study is not meant to do a complete characterization of this phenomenon, since that would require many other experiments than what we are going to discuss.

The first issue that was evaluated is the variability of the values assumed by a single pixel in a dark current condition. The aim of this analysis was to find out the pixel-by-pixel characteristics of this noisy component. To this end, a set of many consecutive frames was acquired and the same cluster composed by a few pixels was extracted from each matrix. Then, those pixels were individually examined throughout all the observation time, and a distribution analysis was performed.

Secondly, the attention was concentrated over the whole panel behaviour. It is well-known from the available detector's information that the pixels' characteristics are not uniform over the whole matrix [14,23]. For this reason, the same set of frames already acquired was exploited to study the spatial distribution of the dark current values over the image.

Thirdly, changes through time of the mean dark current level were investigated. A gradual increase of the noisy component over the hours interested by the detector's warm-up are well known [23]. For this reason, the same acquisition was repeated a few times over the same day, to better understand the evolution and the variation range of the signal associated to the dark current.

In the end, it was necessary to determine a reliable version of the dark current that might be effectively employed to correct a wide range of acquired images. To this end, a long acquisition was performed to collect a large number of dark current realizations. The main goal was to determine the minimum quantity of frames it is necessary to average in order to obtain a consistent dark current estimate. For this reason, a number always increasing of images was averaged to find out after how many steps the matrix values reach a steady state condition. This particular test was performed many hours after the device was powered on, so that the detector's warm-up should have reached a stable condition.

3.2.3 Lag effect model identification

The next step of the lag effect experimental analysis is the model identification through the previously acquired images. This study has several different goals: first of all, we want to verify if the selected function, the sum of exponential components, is suitable to effectively describe the temporal evolution of the lag effect. Secondly, we want to find out if this phenomenon has a consistent behaviour over the different exposure levels considered. This should help to better understand and validate the physical framework that explains the lag origin. Thirdly, the model identification will supply the parameters we can use to apply the correction algorithm illustrated at the end of the *Background* section [12].

The identification algorithm was implemented with *Matlab*, and the entire code can be found in the Appendix A.5. The aim of the program is to estimate the parameters of the function that best fits the experimental data. The chosen function is the sum of a few decreasing exponential components, defined over a discrete time axis:

$$y(k) = \sum_{n=1}^{N} b_n e^{-a_n k} \qquad \text{for} \qquad k = 0, 1, ...K \tag{3.8}$$

where N, in this case, can be either 1, 2, 3 or 4, while K is the total number of dark projection considered. Since the model is non-linear in respect of the parameters a_n, the identification was performed with a *Non Linear Least Squares* (NLLS) algorithm. We now consider the following definitions:

- p is the vector containing all the parameters we want to estimate;
- z is the vector containing the experimental data;
- y contains the estimated curve fitting the experimental data;
- G represents the mathematical function described in equation 3.8, that is how parameters and data are related;
- v is the vector containing the noisy component related to each experimental data.

With these elements we can rewrite the model in the form:

$$z = y + v = G(p) + v \tag{3.9}$$

and then define the *residuals* as the difference between the data and the fitting curve:

$$r = z - G(p) \tag{3.10}$$

What the algorithm does to find out the best parameters to fit the data, is finding the minimum distance between the raw data and the estimated curve. This goal is achieved minimizing the residuals, or to be more precise, the residual sum of squares (RSS):

$$p_{est} = \underset{p}{\operatorname{argmin}} \|r\|^2 = \underset{p}{\operatorname{argmin}} \|z - G(p)\|^2 \tag{3.11}$$

We will now form some additional hypothesis about the noise component v. We assume this measurement error is uncorrelated, with zero mean and with standard deviation σ_k. With these conditions the covariance matrix of the measurement error becomes:

$$\Sigma_v = diag\{\sigma_0^2, \sigma_1^2, ..., \sigma_K^2\} \tag{3.12}$$

Furthermore, considering the measurement error not as a constant but depending on the acquired value they refer to, it is possible to modify the covariance matrix as:

$$\Sigma_v = diag\{(CV \cdot z_0)^2, (CV \cdot z_1)^2, ..., (CV \cdot z_K)^2\} \tag{3.13}$$

where each standard deviation σ_k is replaced by the product between the measured data and a constant value CV called *coefficient of variation.* This condition proves to be much more appropriate to describe the experiment reality. Moreover, this coefficient doesn't need to be known a priori, but can be estimated together with the other parameters. In particular, rewriting the covariance matrix Σ_v as $CV^2 \cdot B$ and the *weighted* residual sum of squares as

$$WRSS(p) = [z - G(p)]' B^{-1} [z - G(p)] \tag{3.14}$$

it is possible to calculate an estimate of the CV, after the parameters identification, as follows:

$$CV^2 = \frac{WRRS(p_{est})}{K + 1 - M} \tag{3.15}$$

In this equation, $K + 1$ represents the total number of data (or frames) considered, while M is the total amount of parameters the model contains. These variations transform the identification into a *Weighted* Non Linear Least Squares (WNLLS) algorithm, since each element has a different importance in the parameters estimate, proportional to the value considered.

Since we are dealing with a non-linear model, an *analytic* (or closed form) solution cannot be found. Instead, it is necessary to adopt a *numerical* approach, that in this case consists in a *successive linearization* method. In accordance with the code in Appendix E, the iterative search for an approximated solution was faced with a Matlab Optimization Toolbox, in particular the *lsqnonlin* function. The *cost function* that this method tries to minimize is the equation 3.14. To start the iteration it is necessary to provide a initial estimate of the unknown parameters. These values are defined in the *LE_modelIdentif.m* function, together with their lower and upper bounds.

The method output consists not only of the model parameters, but also an estimate of the precision of each value, in the form of coefficients of variation. These indices are quite useful to understand if the estimated parameters are reliable or not. Moreover, together with the curve that fits the experimental data, the corresponding *weighted residuals* are displayed:

$$wr_k = \frac{r_k}{\sigma_k} = \frac{y_k - G(p)}{\sigma_k} \tag{3.16}$$

Since, with our hypothesis, the calculated residuals should be a realization of an additive white gaussian noise process ($r \approx v$), checking the weighted residuals characteristics is a good strategy to determine the quality of the model identification. Moreover, since we are testing some different models, which differ in the number of exponential functions they contain, a method to compare them is necessary. To this end, we employ an index called *Akaike Information Criterion* (AIC):

$$AIC = WRSS(p_{est}) + 2M \tag{3.17}$$

To find the best trade-off condition between the quality of fit (that is the weighted residual sum of squares) and the complexity of the model (that is the number of parameters employed), we have to minimize this index.

The last issue concerns how the model identification is applied to the experimental data. Two different approaches are tested: the first one uses a single value for each frame, which is calculated as an average over the whole matrix considered. In this case the identification is implemented only once, and then a single set of parameters is obtained. The second approach is slightly more complex: each frame is divided into a grid, resulting in a group of several square-shaped areas that all together recreate the original matrix. Then, each region of each frame is averaged to obtain a single value. In the end, we will get as many different sets of data as the number of squares the image is divided into. At this point each of them can be separately identified. By doing so, we get many copies of the same parameters, that we can average to obtain a single set. With this second method we might also find

out the distribution of the lag behaviour over the detector, to check if there are substantial differences changing with position.
Both these methods were repeated twice with a small but important difference: in the first case, the considered matrix was the entire detector, while the second trials were computed over a subarea extracted form the central portion of the panel. In other words, the second identification was implemented only where the phantom is expected to be. Moreover, it also corresponds to the most flat part of the beam profile, and then we might guess the pixels behaviour should be rather uniform [23]. For these reasons, that is the portion of the image we are interested the most, and then that is the place where we want the lag correction to be the most effective.

3.2.4 Lag effect correction tests

The last step of the experimental study is represented by the attempt to employ the information about the lag phenomenon to compensate and eventually delete its effect from the acquired images. But, before reaching this goal, it is necessary to verify if the model employed and the parameters estimated were effective or not. To this purpose, a new test was performed, with the same protocol specifications as those used to identify the lag decay, but this time with some objects in front of the X-ray source. A couple of particular phantoms were used, composed by parts and layers of different materials and thicknesses, each of them with a specific filtering capacity. This choice was made to have a wider variety of pixel levels available on the projected image, and then to prove the correction effectiveness over a range of exposure conditions.

In order to prove the exponential model ability to describe the lag effect evolution, a *predictive* test was employed. In particular, the aim of the experiment was to guess, using only the last projection exposed to the beam, the image the lag creates over the 50 following frames. To this end, the exponential model of equation 3.1 was applied to the selected frame, and then all the predicted images were compared with the acquired ones. If the model identification was effective, there shouldn't be much difference between the two sets of images. Then, the subtraction of a predicted image from a real one should produce a distribution of the pixel levels not far from zero. Moreover, before this procedure was applied, it was necessary to consider the dark current signal contained into every image. 50 dark current frames were acquired just before the test and averaged; the resulting one was then subtracted from all the frames of the experiment. In this condition we were then ready to correctly apply the lag effect model.

The second experiment we should have carried out, that aimed at deleting the effects of the lag from the reconstructed images, was impossible to perform at the moment this report was written. This was caused by a matter of acquisition software, as it will be briefly discussed in the following.

The protocol and all the specifications that should be employed are the ones already used for the *Reconstruction* at the end of the *Misalignment correction* chapter. These conditions were, at the moment the tests were conducted, the best available. In particular, an important issue concerns the amount of time the rotation of the whole gantry takes up. It came up that the completion of a single step normally requires a few seconds, used to cover a small angle of the full rotation. For a matter of synchronization, the time period that exists between two consecutive readouts has to be a multiple of the chosen *frame time*. Indeed, every image is not separated by a single *frame time* as it was supposed in the version of the state model previously discussed: between two consecutive images, some frames are acquired and then discarded. It is important to note that none of them receives a radiation exposure, but nevertheless it is essential to consider them while we correct the acquired frames we are interested in. Unfortunately, the temporary version of the acquisition software used prevented us from collecting and employing the complete set of detector images, and consequently, the lag effect evolution was not fully available. A simple software update should be sufficient to solve the problem, and would allow to properly apply the correction protocol.

3.3 Results

Lag effect visualization

The decisions regarding the setup of this first experiment were already discussed in the first *Method* paragraph. Then, the specifications of the acquisition protocol are the following: the voltage was set to the value of 45 kV, the current to 800 μA, and the frame time and pulse with to 1000 ms and 500 ms respectively. The 100 images collected (50 with and 50 without the X-ray turned on) were all corrected by subtraction of a dark current image, which was obtained averaging 50 frames collected a few minutes before the experiment began.

In Figure 3.5 the 50*th* projected image is shown, which is the last one that was hit by the X-rays. In this case, the dynamic range goes from 0 digital counts, corresponding to a black pixel (no photons detected) to 2700, that corresponds to white and saturates all the values above until 4096.

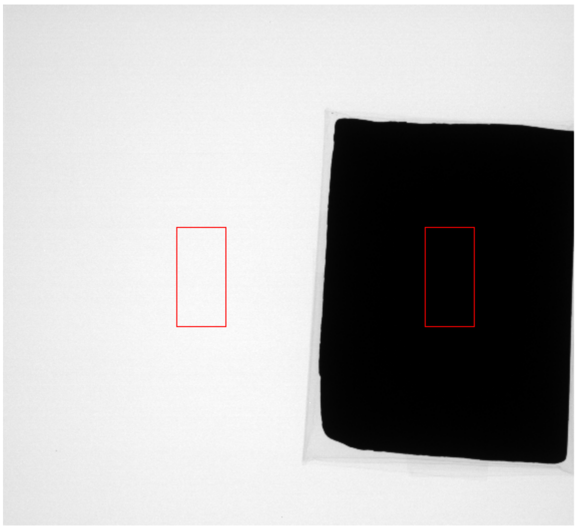


Figure 3.5: Acquired image of the detector' panel covered by a lead plate. The two red rectangular frames select the regions of interest (50th frame, 45 kV, 800 μA, 500 ms p.w., 1000 ms f.t.)

On this image two red rectangles are drawn: they show the two *regions of interest* extracted from the matrix. The first ROI is completely exposed to the beam, while the second one is entirely shadowed by the rectangular plate. Both of them cover a 200-by-400 pixels area.

The following figures are obtained by patching together the two ROIs extracted from an image. They will show some of the successive frames collected during the second half of the experiment. The one in Figure 3.6 refers to the first image that was acquired right after the X-ray tube has been shut down. In this case, the gray levels range from 0 to 80, and reveal an evident difference in the behaviour of the two halves.

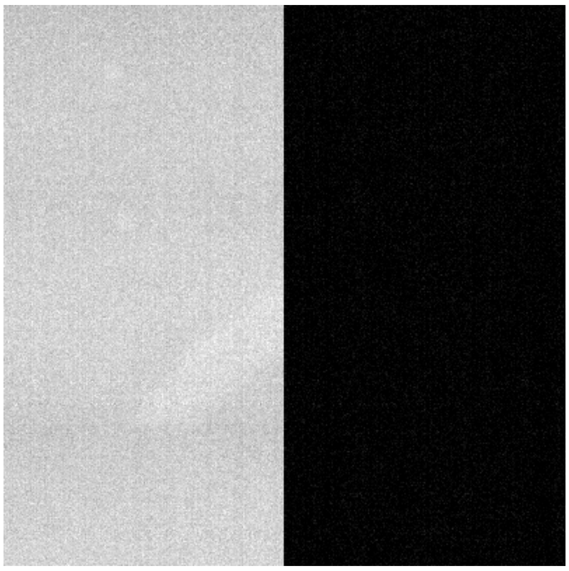


Figure 3.6: ROIs comparison of the first *dark* frame. (dynamic range: 0 - 80 digital counts. Matrix dimensions: 400x400 pixels)

The next figures are all related to successive images. In Figure 3.7, three moments are shown, that are respectively 2, 3 and 4 *after* frames. Similarly, in Figure 3.8 the 10th, 20th, 30th and 50th frames are illustrated. These 7 images are all optimized with the same dynamic range, that goes from 0 to 10 digital counts. This uniform condition implies that all frames are here comparable to each other. Then, it can be clearly observed a slow decay of the left parts from a brighter to a darker average gray level, while the right part seems to roughly maintain the same black background level.

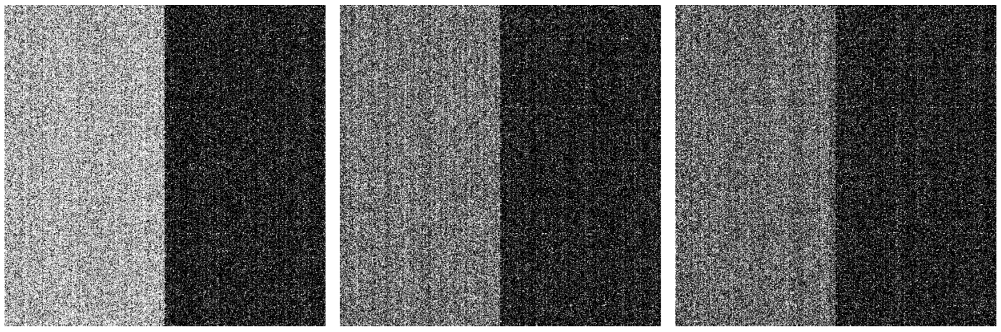


Figure 3.7: from left to right: 2nd, 3rd and 4th ROIs comparison frames (dynamic range: 0 - 10 digital counts)

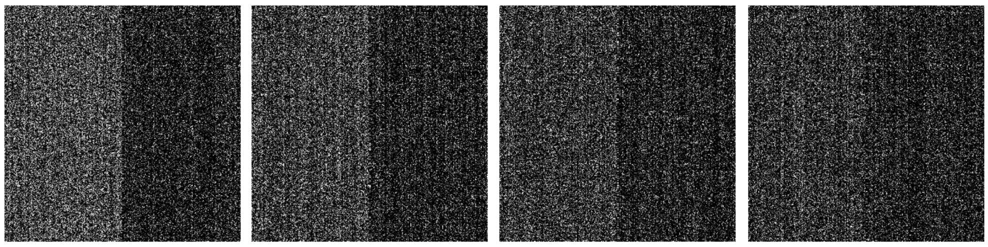


Figure 3.8: from left to right: 10th, 20th, 30th and 50th ROIs comparison frames (dynamic range: 0 - 10 digital counts)

The following two plots will resume this phenomenon and illustrate its behaviour. Each ROI was individually averaged to obtain a single value that describes the mean grey level of that area. This operation was repeated on all the frames for both regions, obtaining two 100 elements sets. These sequences were then plotted in two different colors, blue for the left ROI and red for the right one. Furthermore, the standard deviation over the pixels of each region was computed and then drawn as two dashed lines that create the confidence interval of the averaged values.

The results are divided into two plots to better follow the two different phases of the experiment. In Figure 3.9 the first 50 frames are illustrated; only the left ROI data are shown, since the red line is much lower, almost constantly equal to zero.

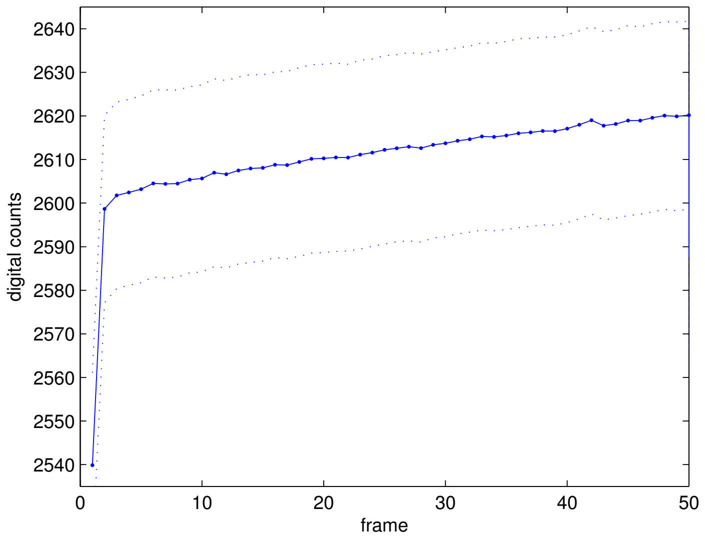


Figure 3.9: left ROI - exposed to X-rays - digital count mean value with ±SD (dashed lines). Frames from 1 to 50.

In Figure 3.10 the last 50 frames are illustrated. In this case, both curves are simultaneously visible within the same plot, since they have comparable values. The first frame was intentionally not included in the plot since it is around 70 digital counts, much higher than all the other following ones that quickly decrease reaching close to zero values.

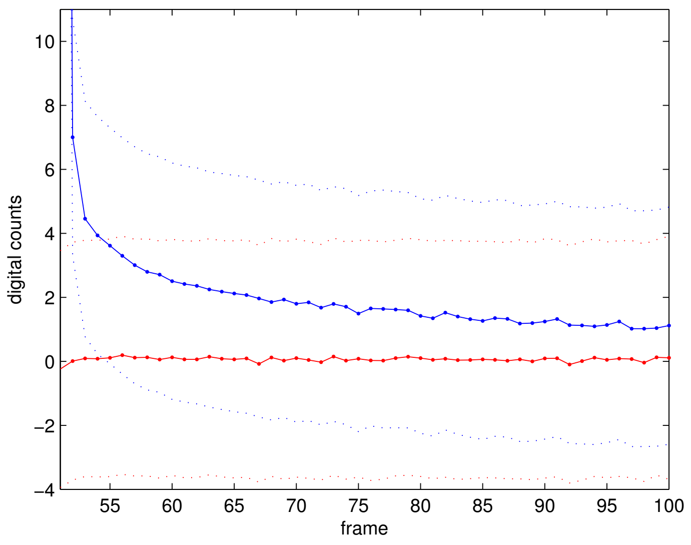


Figure 3.10: left ROI (blue) VS right ROI (red) mean value trends. Dashed lines: confidence interval of ±1 SD. Frames from 52 to 100.

Dark current characterization

The following results are meant to illustrate and characterize the dark current signal. All of the experiments discussed have been realized with a 1000 ms frame time, a 500 ms pulse width, a 1x1 binning mode and an external triggering synchronization. Changing those parameters would considerably modify the acquired dark current signal, so it was extremely important to conduct these tests with the same protocol it was used for the lag effect detection.

The three plots in Figure 3.11 represent a first characterization of the dark current signal. They show the oscillation of the value detected in three different pixels randomly extracted from the central area of the matrix. These three plots are then just an example of the noisy behaviour of the dark current signal. The acquisition lasted 260 seconds, and it was performed more than 10 hours after the detector panel was turned on.

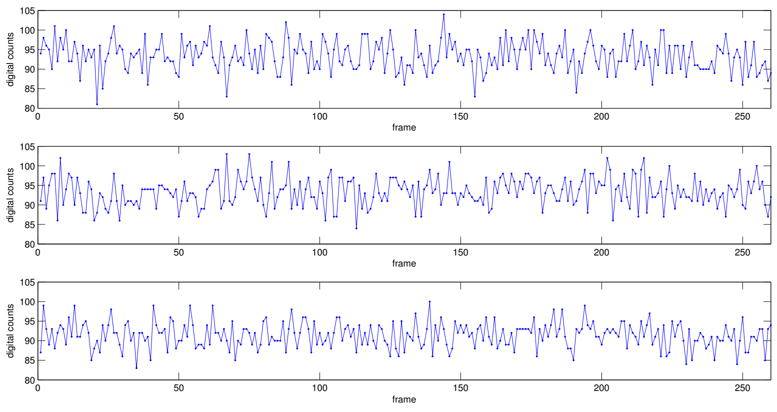


Figure 3.11: 3 pixel's oscillating evolution over a 260 seconds acquisition (x-axis: frame number. y-axis: digital counts). Protocol specifications: 1000 ms f.t., 500 ms p.w.

To verify the variability of the mean level of dark current over the warm-up period, the same acquisition was repeated a few times. Then each collection of frames was divided into groups composed by 10 images. Each subgroup was averaged to obtain a single frame, and then averaged again to get a mean index. These values are plotted in Figure 3.12.

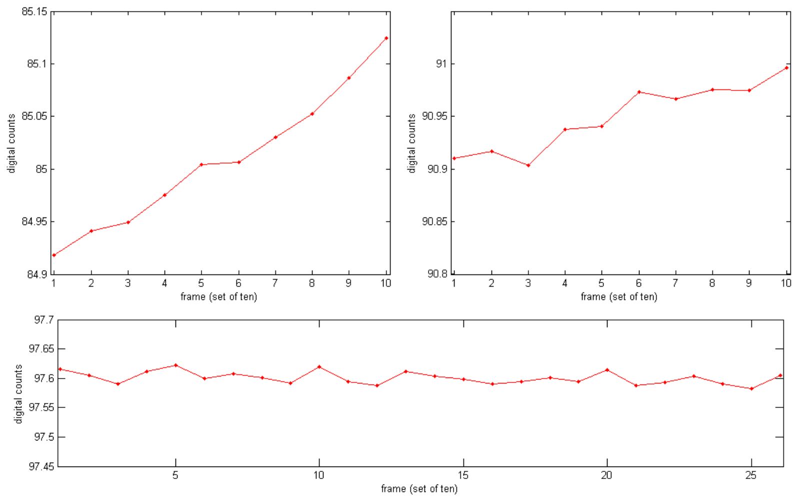


Figure 3.12: Dark current mean value evolution during the detector's warm-up period. Each point corresponds to a 10 frames average. Time passed from panel's switch on: top-left picture - 30 minutes (100 frames). Top-right picture - 2 hours (100 frames). Bottom picture - 10 hours (260 frames).

The upper left subfigure is related to 100 frames acquired approximately half an hour after the device was switched on, while the upper right data were collected two hours later. At last, the lower image regards a longer acquisition operated 10 hours after the start. All the plots are defined by digital counts versus the frames subgroup, and the three of them show a constant range of 0.25 digital counts, so that they all are comparable to each other. As we might guess from these results, the 10 hours employed in the last test seem to be sufficient to reach a steady state of the dark current signal.

The next step consisted in evaluating the minimum quantity of frames it is necessary to average in order to obtain a reliable approximation of the true dark current signal. To this end, the 260 frames acquisition already discussed was employed. 26 different average version of the dark current matrix were calculated: the first one considering the images from the 1st to the 10th, the second one from the 1st to the 20th, until the last one, obtained with a mean of all the 260 frames. Then, the absolute value of the difference between consecutive estimates was calculated: that means the 1st average was subtracted from the 2nd, the 2nd from the 3rd, and so on. By doing so, 25 difference matrices were estimated. At this point, the average and the standard deviation of each of them was calculated and plotted, as it is shown in Figure 3.13.

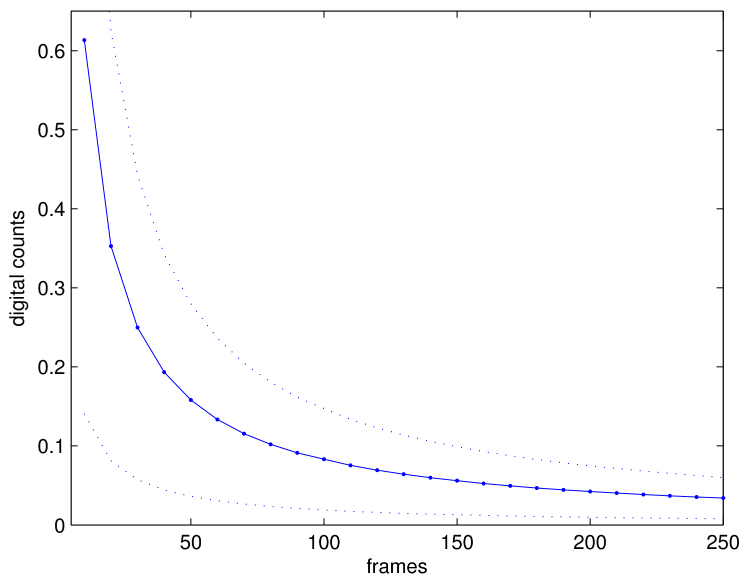


Figure 3.13: Mean value (solid line) and standard deviation range (dashed lines) evolution of the absolute difference between consecutive dark current average estimates.

The desired number of frames should be found when the curve stops decreasing and becomes flat, which means that adding 10 more frames to the average doesn't change the dark current estimate, and then a sufficiently reliable DC correction matrix was reached.

Finally, the best estimate available of the dark current image was ready to be calculated. It was decided to consider the 260 frames acquisition already discussed, and to average all the images. The resulting matrix is shown in Figure 3.14, with a dynamic range specified by the colorbar on its right.

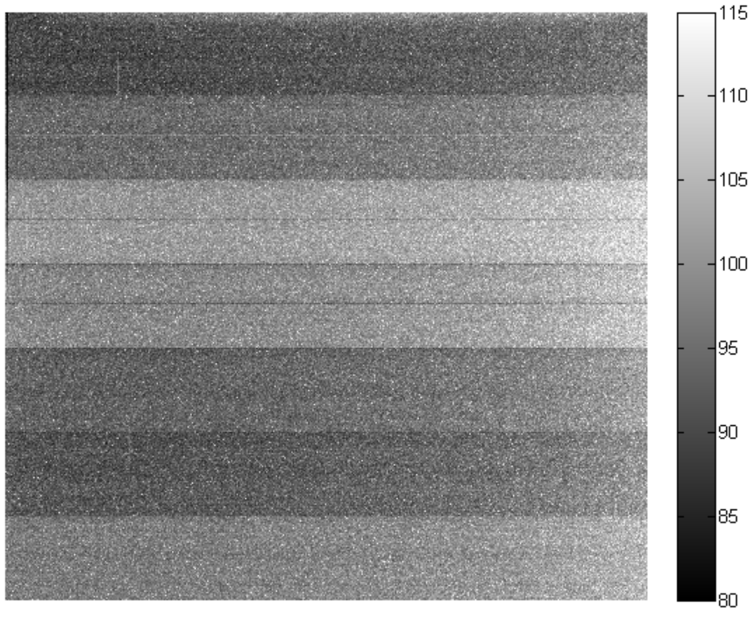


Figure 3.14: Dark current signal obtained averaging 260 frames.

As it was well-known from the detector's manual and other studies [14,23], the distribution of the dark current signal is not uniform over the whole panel. On the contrary, a band pattern can be clearly recognized, which is caused by its manufacturing. Moreover, an increasing trend can also be noticed, that goes from the left to the right side of the matrix.

Lag effect model identification

As discussed in the *Method* section, the identification test was repeated four times, with the same acquisition protocol but different exposure levels. In particular the experiments consisted in collecting 50 images hit by the X-rays shot and 50 *after* dark images. Within this framework, the *tube current* successively ranged from 200 μA to 800 μA. Before being ready for the identification step, all the collected frames needed some image corrections to be applied, according to equation 3.7. In particular, the *light field* image was obtained averaging 40 frames acquired during the test, those from the 11th to the 50th. Indeed, according to the plot in Figure 3.9, these frames are expected to be in a sufficiently uniform condition. On the other hand, to get an effective dark current estimate, 50 frames were collected a few minutes before the real experiment. Then these dark images were averaged to get a single matrix, which should be suitable to apply the correction described in (3.7). The data actually employed in the model identification were just 31 of the total amount of acquired frames: the last image shot by the X-rays and the first 30 ones from the *dark* group. It was decided to leave the last 20 images out since it was observed that they were too close to the dark current level. Which means that those data were particularly noisy and then quite difficult to fit with the model.

The parameters estimated with the model identification are all collected in the following tables. Firstly considering the exponential models with just 2 components, the results are divided in four tables, according to the experiment they belong, that are the four different tube currents adopted. Moreover, each column refers to one of the identification methods previously discussed.

The first two columns list the parameters obtained from the identification performed over a single set of data, the one obtained from the average of each frame. Column A is the case where the pixels of the whole detector were used, while column B refers to values extracted from a 1200x1200 pixels central area. In these two cases, the precision of each parameter's estimate is reported between brackets: these numbers are the coefficient of variation, expressed as a percentage.

The last two columns C and D list the results obtained with the second protocol, the one that performs an individual identification on each submatrix the frame is divided into. In this particular case, the images were cut to obtain a collection of several 50x50 pixels areas. Then, after an identification performed on each single set of data, the parameters were averaged to get a single set, which is the one here reported. Among the full set of identifications, those which failed to find a curve that well fitted the experimental data were excluded from the average. In particular, the discarded curves are those described by a set of estimated parameters that always exceed the confidence bounds, no matter what's their initial value. Moreover, in these cases the residuals appear to be extremely polarized, suggesting the identification failure. As for the first two columns, C and D results are divided according to the matrix area considered: in the left one (C), the whole panel was used, while in the right one (D) a 1200x1200 pixels square was extracted. With this second method, the precision reported between brackets is the standard deviation (normalized to the estimated values and expressed as a percentage) calculated over the parameters that were used to compute the average.

All the discussed parameters are illustrated in the following *Tables* 3.1, 3.2, 3.3 and 3.4 for the 800 μA, 600 μA, 400 μA and 200 μA experiments respectively. All of the b_i values and precisions are normalized for the exposure level, so that results from all the four cases are perfectly comparable to each other.

Table 3.1: Estimated parameters from a model with 2 exponential components. 800 μA test

	A	B	C	D
b_1	0.92 (19%)	0.94 (7%)	0.94 (1%)	0.94 (1%)
a_1	3.49 (6%)	3.50 (2%)	3.53 (2%)	3.50 (1%)
b_2	0.0015 (8%)	0.0015 (3%)	0.0016 (6%)	0.0015 (3%)
a_2	0.040 (11%)	0.039 (4%)	0.040 (8%)	0.040 (5%)

Table 3.2: Estimated parameters from a model with 2 exponential components. 600 μA test

	A	B	C	D
b_1	0.95 (19%)	0.94 (8%)	0.94 (1%)	0.94 (1%)
a_1	3.53 (6%)	3.48 (3%)	3.51 (2%)	3.48 (1%)
b_2	0.0017 (8%)	0.0015 (3%)	0.0016 (6%)	0.0015 (4%)
a_2	0.045 (10 %)	0.039 (5%)	0.042 (10%)	0.041 (7%)

Table 3.3: Estimated parameters from a model with 2 exponential components. 400 μA test

	A	B	C	D
b_1	0.96 (20%)	0.95 (8%)	0.94 (2%)	0.94 (1%)
a_1	3.63 (7%)	3.44 (3%)	3.47 (2%)	3.45 (1%)
b_2	0.0018 (8%)	0.0014 (3%)	0.0016 (13%)	0.0015 (7%)
a_2	0.058 (8%)	0.042 (4%)	0.047 (15%)	0.045 (16%)

Table 3.4: Estimated parameters from a model with 2 exponential components. 200 μA test

	A	B	C	D
b_1	0.93 (29%)	0.92 (10%)	0.93 (5%)	0.94 (5%)
a_1	3.35 (10%)	3.31 (3%)	3.4 (3%)	3.4 (3%)
b_2	0.0016 (12%)	0.0015 (4%)	0.002 (50%)	0.002 (50%)
a_2	0.052 (13%)	0.046 (5%)	0.08 (75%)	0.09 (67%)

To verify the accuracy of the identification outcome, the plots in Figures 3.15 and 3.16 can be quite useful. The upper diagram of each figure shows the identified curve plotted against the raw data. From that, it is possible to check if the estimated curve succeeds in following the expected decreasing trend of the noisy data. These diagrams refer to the single identification protocol applied to the 800 μA experiment. In particular, the plots in Figure 3.15 are obtained from the parameters contained in the column A of Table 3.1, while the plots in Figure 3.16 refers to the column B of the same table. Values in these plots are not normalized, that means the numbers on the ordinate axis are the true digital counts of the lag effect. For a matter of graphical clearness, the values relative to the first two frames have been excluded from the illustration, since they are too large if compared with the lower data shown in figure.

The lower plots in Figure 3.15 and 3.16 show the weighted residuals of the fitting curve. In this case the range of values was limited to ± 2 standard deviations. Checking the residuals distribution around zero, their amplitude and their possible polarisation, these plots can give us important information regarding the quality of the identification outcome. These issues will be further examined in the following *Discussion* section.

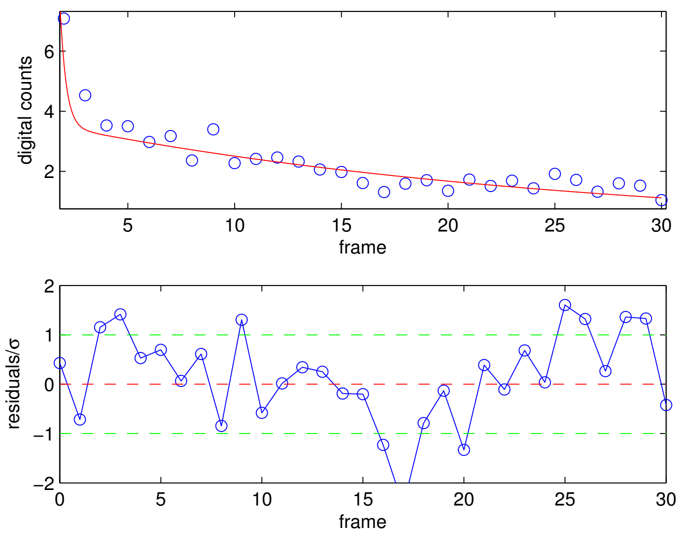


Figure 3.15: Results from the decay trend identification: Method A (average over the whole detector), 2 exponential components model. Top image: fitting curve (red) VS data (blue). Bottom image: model residuals weighted with the standard deviation of the measurement error.

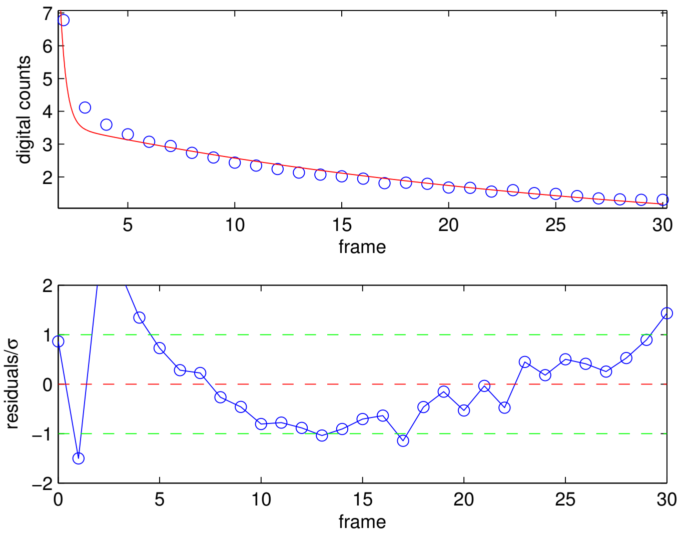


Figure 3.16: Results from the decay trend identification: Method B (average over a central ROI), 2 exponential components model. Top image: fitting curve (red) VS data (blue). Bottom image: model residuals weighted with the standard deviation of the measurement error.

The following Tables 3.5 and 3.6 and Figures 3.17 and 3.18 show the results produced by the same identification framework as the one already discussed, but with the fundamental difference of a more complex model, this time composed by three exponential components. In this case the only experiments exploited were those at 800 μA and 600 μA, since the other ones presented values too noisy to identify up to 6 parameters. That probably happens because the lower exposure implies lower values in the collected data, especially in the slow components of the decreasing signal, which then result too close to the noisy dark current.

Table 3.5: Estimated parameters from a model with 3 exponential components. 800 μA test

	A	B	C	D
b_1	0.98 (15%)	0.99 (2%)	0.99 (1%)	0.99 (1%)
a_1	3.66 (6%)	3.65 (1%)	3.71 (2%)	3.66 (1%)
b_2	0.0017 (23%)	0.0014 (11%)	0.0019 (37%)	0.0017 (35%)
a_2	0.18 (59%)	0.34 (12%)	0.33 (42%)	0.35 (43%)
b_3	0.0007 (61%)	0.0012 (3%)	0.0012 (17%)	0.0012 (17%)
a_3	0.01 (170%)	0.031 (4%)	0.027 (30%)	0.028 (32%)

Table 3.6: Estimated parameters from a model with 3 exponential components. 600 μA test

	A	B	C	D
b_1	1 (18%)	0.99 (4%)	0.99 (1%)	0.99 (1%)
a_1	3.77 (8%)	3.64 (2%)	3.7 (2%)	3.65 (1%)
b_2	0.004 (159%)	0.0015 (23%)	0.0024 (54%)	0.002 (50%)
a_2	0.7 (85%)	0.38 (21%)	0.42 (55%)	0.4 (75%)
b_3	0.0015 (12%)	0.0013 (4%)	0.0012 (25%)	0.0011 (36%)
a_3	0.04 (15%)	0.032 (6%)	0.028 (43%)	0.03 (33%)

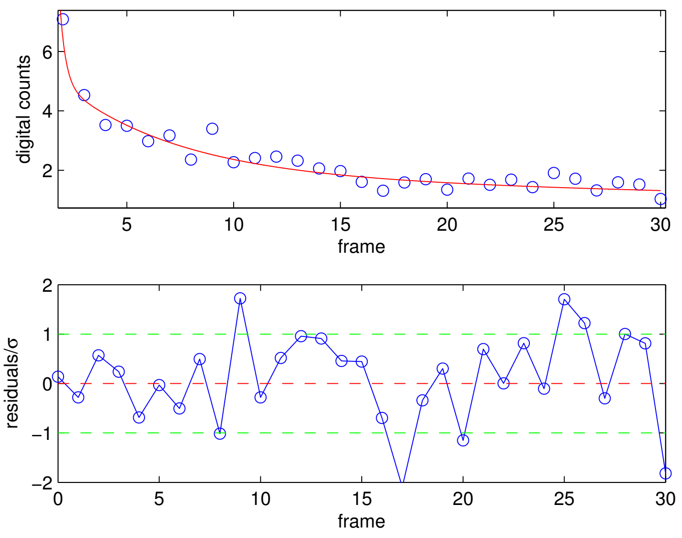


Figure 3.17: Results from the decay trend identification: Method A (average over the whole detector), 3 exponential components model. Top image: fitting curve (red) VS data (blue). Bottom image: model residuals weighted with the standard deviation of the measurement error.

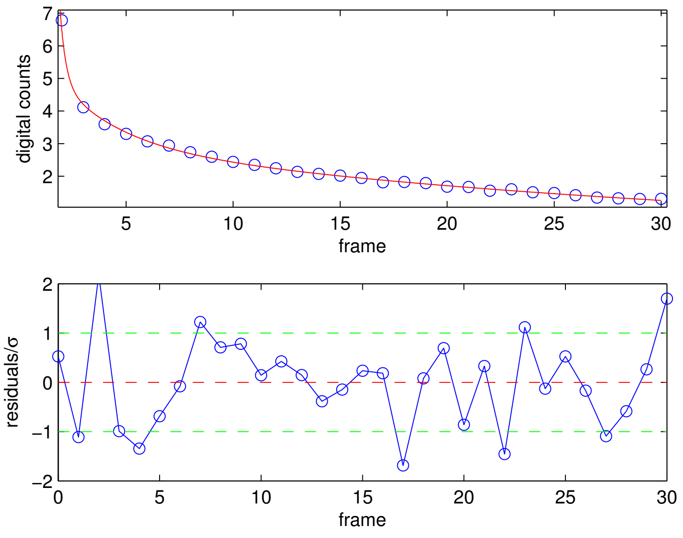
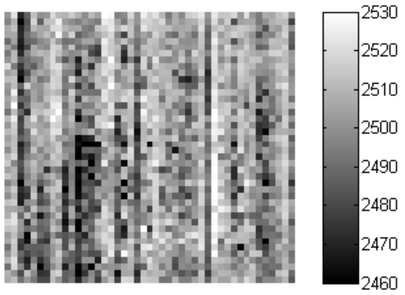


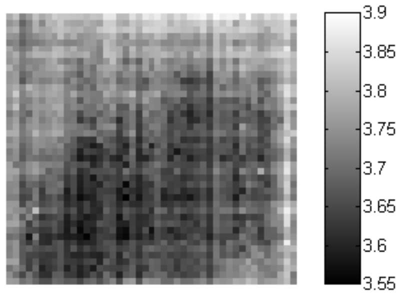
Figure 3.18: Results from the decay trend identification: Method B (average over a central ROI), 3 exponential components model. Top image: fitting curve (red) VS data (blue). Bottom image: model residuals weighted with the standard deviation of the measurement error.

In addition to the standard deviation reported into columns C and D of the previous tables, a different way to verify the uniformity level of the lag effect behaviour over the detector is represented by the *maps of parameters*. These maps show, with different levels of gray, the values assumed by a parameter that was identified over different areas of the panel. Every parameter has its own map, and the map's representation reproduces the detector's geometric organisation. This graphical support allows to immediately understand the lag effect distribution over the panel. In particular it can be verified if there are problematic or excessively noisy regions, or we can check the presence of particular patterns of higher lag intensity.

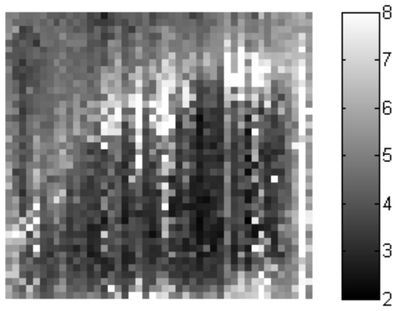
In Figure 3.19 six maps of parameters are shown. They refer to the parameters estimated from the 800 μA experiment, which are listed in column C of Table 3.5. The three frames on the left correspond to the proportional coefficient b_n (expressed in their not normalized form), while the right ones illustrate the distribution of the exponential rate a_n. Each image has its own colorbar, since a specific dynamic range was necessarily chosen for every different parameter, but everywhere a lighter grey corresponds to a higher value. For the right column figures, this means that the areas with a light grey tone are related to exponential functions that decay more rapidly. On the contrary, darker regions are associated with lag components that last longer than the others.



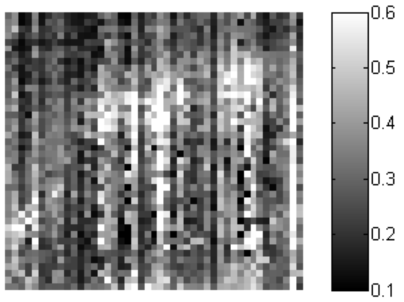
(a) parameter b_1



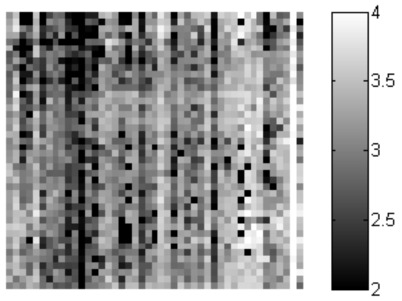
(b) parameter a_1



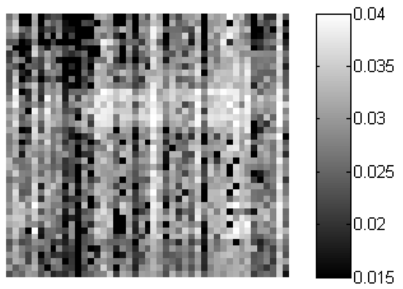
(c) parameter b_2



(d) parameter a_2



(e) parameter b_3



(f) parameter a_3

Figure 3.19: Maps of the 6 parameters estimated with the method C (Different model identification over 1890 ROIs each one with area of 50x50 pixels). Model: 3 exponential components. Experiment: 800 μA.

Lag effect correction tests

Some of the frames resulting from the application of the correction algorithm are illustrated in the following figures. The first acquisition was performed with a phantom composed by a pair of rectangular lead plates. These objects, positioned in front of the detector, absorb a considerable fraction of the photons shot from the X-ray source. As a consequence, the image projected on the panel shows regions with a huge contrast. The last frame exposed to the beam is illustrated in Figure 3.20.

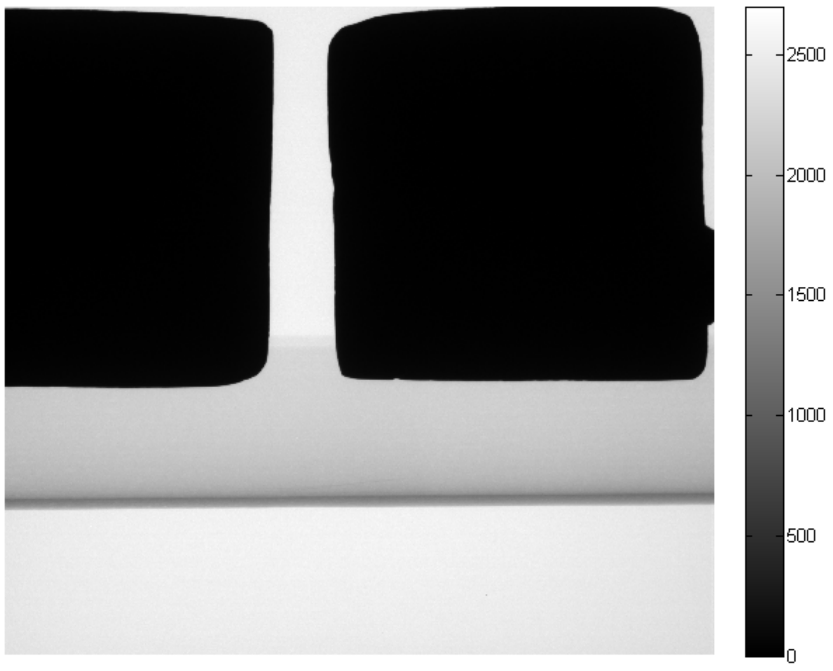


Figure 3.20: Acquired image of the detector's panel with two lead plates in front of it (50th frame, 45 kV, 800 μA, 500 ms p.w., 1000 ms f.t.).

The different exposure levels can be clearly observed in the following *dark* images, where the lag effect maintains a lighter grey tone in the area between the two filters. The following set of figures shows the comparison between the *before correction* images and the *after correction* ones. In particular, those displayed into the left column are the actual acquired frames, affected by the lag signal, while the right column is occupied by the respective images once the effect has been compensated. The dynamic range used to show each pair of images was optimized to enhance the visible effect of the correction, and then it changes from frame to frame, as specified by the colorbar. Anyway, the two images belonging to a pair are always displayed with the same dynamic range, so that the grey levels are fully comparable. Moreover, these images were not normalized, which means that the grey levels always correspond to a digital count number, even after the dark current and lag effect corrections.

The first and second frames collected after the X-rays have been shut down, together with their corrections, are shown in Figure 3.21 and 3.22.

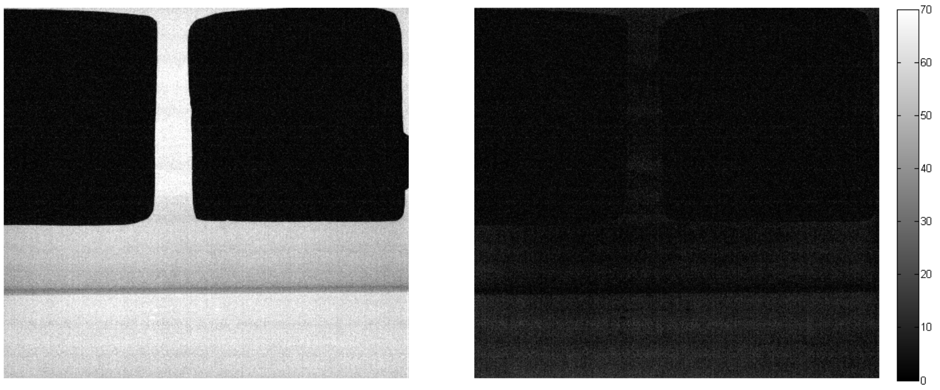


Figure 3.21: First *after* frame acquired (left image) and corrected (right image). Dynamic range: 0 - 70.

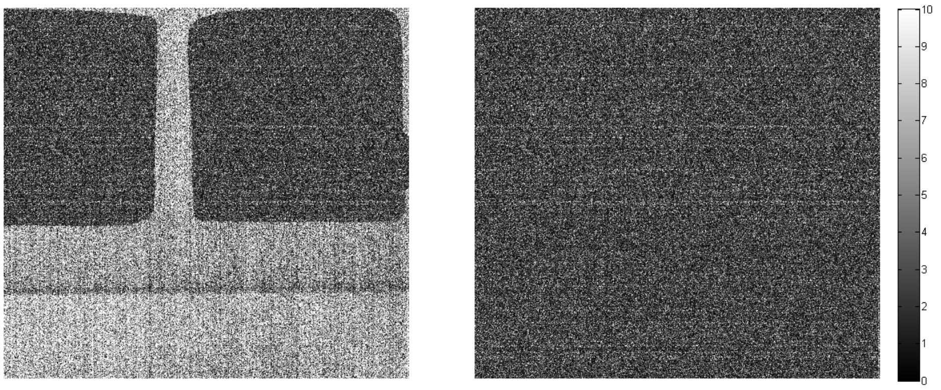


Figure 3.22: Second *after* frame acquired (left image) and corrected (right image). Dynamic range: 0 - 10.

Similarly, the 3rd, 5th and 10th frames are illustrated in Figures 3.23, 3.24 and 3.25 of the next page.

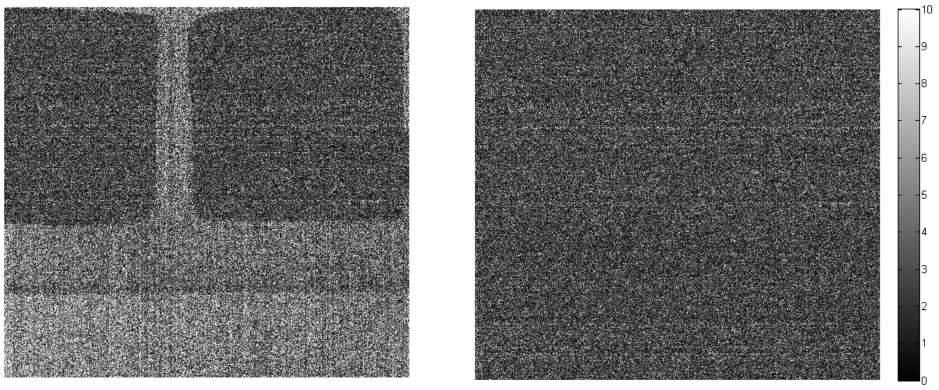


Figure 3.23: 3rd *after* frame acquired (left) and corrected (right)

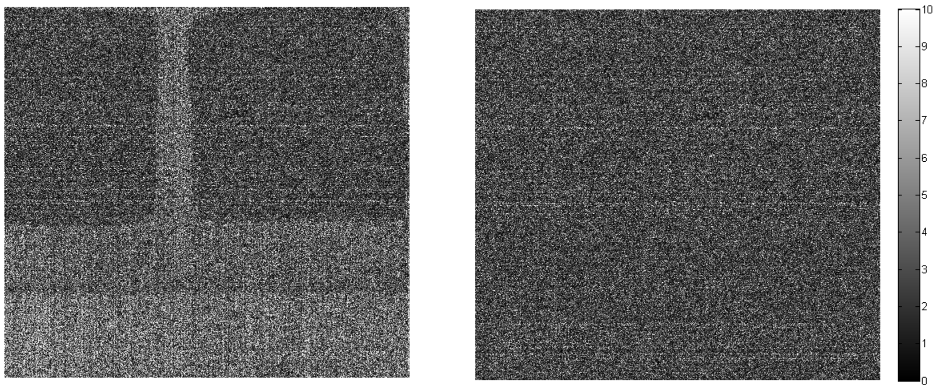


Figure 3.24: 5th *after* frame acquired (left) and corrected (right)

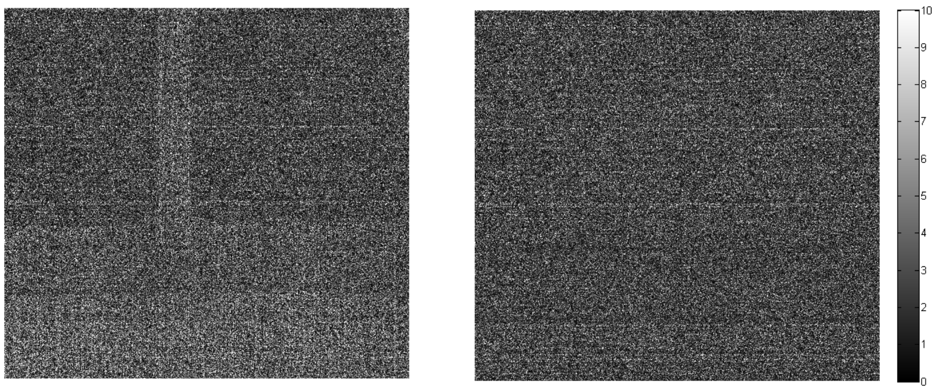


Figure 3.25: 10th *after* frame acquired (left) and corrected (right)

The same acquisition protocol and correction algorithm was then applied to a new experiment, where a different phantom was employed. This new phantom was built putting together a few objects and plates with different thicknesses and materials. The aim was to create a heterogeneous condition of beam filtering, so that different parts of the panel were hit by various exposure levels. The purpose of this test was to verify if the correction system was general enough to equally compensate all of these lag situations.

The frame at time zero, the last one to be exposed to the X-rays, is shown in Figure 3.26.

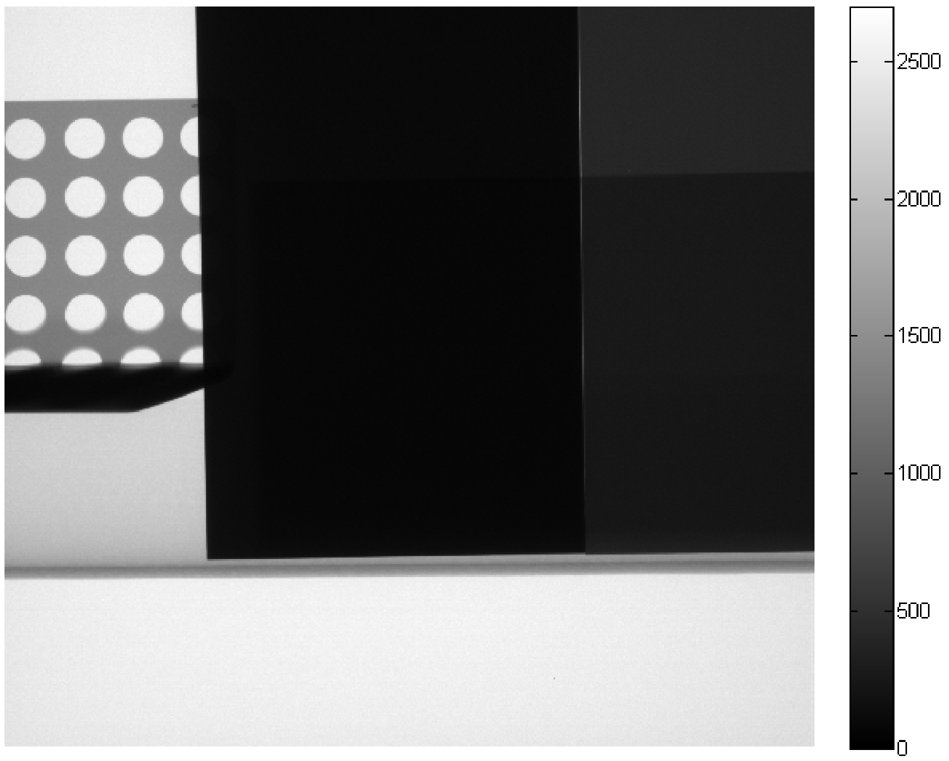


Figure 3.26: Acquired image of the detector's panel with a phantom made of several objects and materials in front of it (50th frame, 45 kV, 800 μA, 500 ms p.w., 1000 ms f.t.).

As for the previous test, the following pictures show the comparison between the acquired frames and the corrected ones, displaying the first, second, third, fifth and tenth image after the X-ray were shut down.

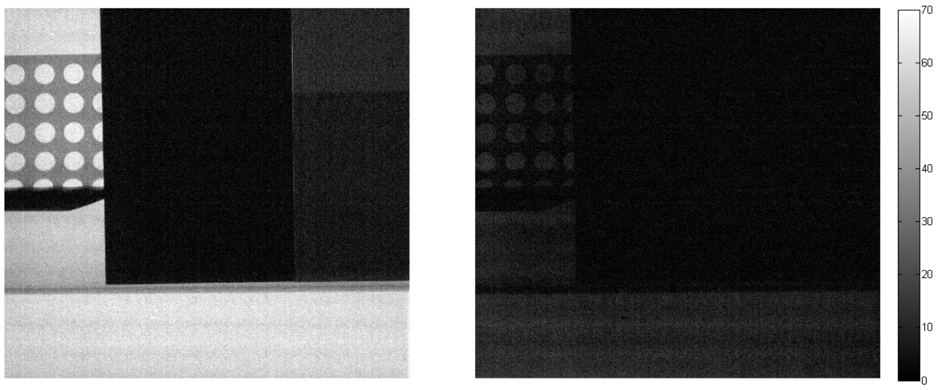


Figure 3.27: First *after* frame acquired (left image) and corrected (right image). Dynamic range: 0 - 70.

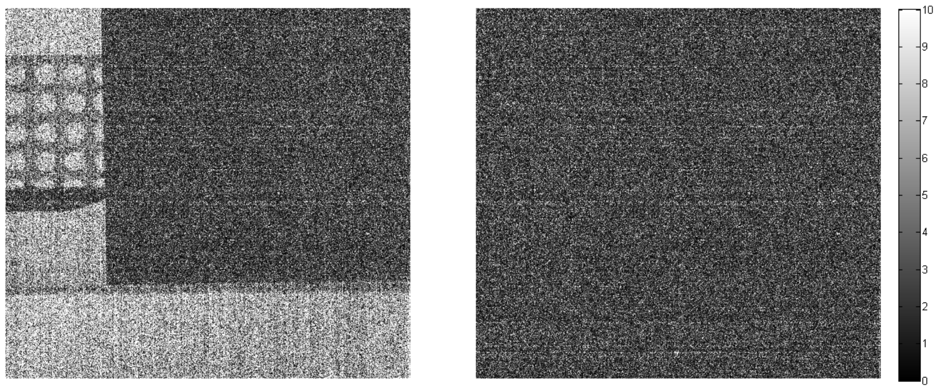


Figure 3.28: Second *after* frame acquired (left image) and corrected (right image). Dynamic range: 0 - 10.

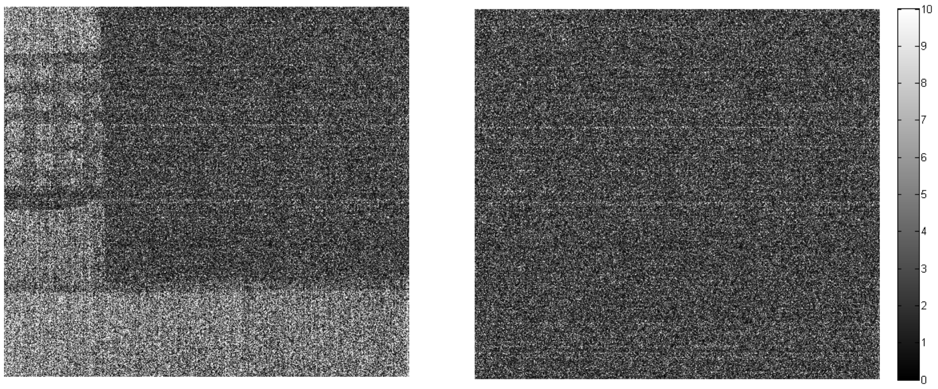


Figure 3.29: 3rd *after* frame acquired (left) and corrected (right)

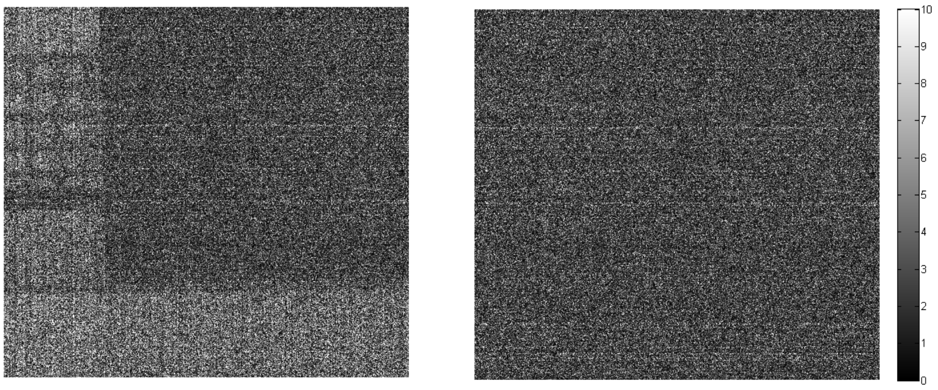


Figure 3.30: 5th *after* frame acquired (left) and corrected (right)

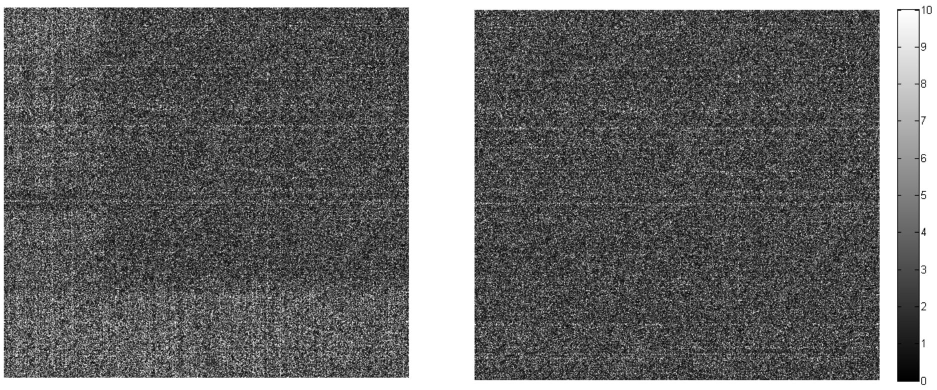


Figure 3.31: 10th *after* frame acquired (left) and corrected (right)

3.4 Discussion

The first issue that deserves a deeper consideration is the particular acquisition protocol that was employed in almost all the tests presented so far. Since they were performed to study and compensate the lag effect, it was of crucial importance to understand how the chosen protocol affects the expression of this phenomenon, and which are the limits of the correction method proposed. In particular, it is essential to remember that even if we tried to describe this effect in the most general possible way, some restrictive assumptions were inevitably made. For example, the *tube voltage*, and with it the energetic content of the X-ray beam, was always maintained to a constant value. Similarly, the *frame time* and *pulse width* duration were never changed, at least in the experiments and tests here presented. These features have important consequences on the lag evolution and effect, and then they would require some further examinations.
For what concerns the acquisition protocol, the number of X-rays shots the panel receives before observing the lag decay represents an important issue. In all the experiments discussed, that quantity was set to 50. This repetition was justified by the necessity of enhancing the successive lag signal. Indeed, it was demonstrated that, with the chosen conditions of acquisition, a single shot would not be able to saturate all the available traps that originate the lag effect. As a consequence, the signal we want to analyze would be too weak and noisy, and then very difficult to fit with any model. For that reason, the close repetitions of several shots gradually fill up all the traps. The resulting increasing trend can be clearly recognized in the plot of Figure 3.9, where the signal, starting from 2540 digital counts, rapidly reaches a maximum value of 2620. Anyway, since the signal is still growing after 50 frames, we can conclude that even this sequence doesn't manage to reach the equilibrium state, where the quantity of electrons released by the traps make even with the number of electrons captured. With this aim, future investigations might try to increase the number of X-rays shots, or maybe prepare the detector with a continuous beam for a few minutes.

Another feature that deserves a brief discussion is the *dark current* characterization. While we tried to find out the best estimate possible averaging a considerable quantity of frames, it was observed that even 260 images weren't enough to reach a steady condition. That is because the plot in Figure 3.13 exhibits a slow but constant decay, and still slightly decreases at the last difference calculated. For this reason, future works should push the observation limit even further, collecting a higher number of dark frames, but at

the same time being sure the warm-up period has already ended.
In our specific case, more than an extremely stable version of the dark current frame, it was of crucial importance to know the mean level of that signal in the exact moment the lag acquisition was performed. The explanation was already supplied, and regards the identification of the low long components of the lag decay. For that reason, a compromise was found, and the dark current was always estimated a few minutes before the corresponding lag acquisition, but with a considerably lower number of frames if compared to the 260 images acquisition.

Getting now to the heart of the matter, the results of the model identification need to be examined more in detail. A first overview on the whole set of parameters reveals quite a uniform estimation in all the conditions analyzed. The only objection concerns a global loss of precision and a slight degradation of the estimates when the current values decrease. That happens in particular in Table 3.3 and 3.4, where the *tube current* was set to 400 and 200 μA respectively. As previously mentioned, this is probably caused by the lower values assumed by the longer components of the signal, that quickly become really close to the dark current level; then they tend to get a quite noisy behaviour, that might affects the identification procedure.

In a second moment, after the identification of the parameters that now are listed in columns A and C of every table, the same estimates were repeated on a central subregion of the detector's matrix. The reasons of interest for this solution were various. First of all the beam profile appears to be more uniform close to the center and slowly decrease towards the edges of the panel. This characteristic modifies the profile of the detector's exposure, and then it may affect the behaviour of the areas close to the borders. Secondly, the central region is obviously more interesting from an acquisition point of view, since the phantoms will typically project their shadows in this area. Then, a proper lag correction appears to be more important in that point of the detector. Thirdly, the areas close to the borders tend to be more noisy, and sometimes totally unusable, than those close to the center. This circumstance is demonstrated by the averaged data behaviour plotted in the upper diagrams of Figure 3.17 and 3.18: those mean values were extracted from the same experiment, and they are plotted with the same digital count range. From these two plots we can clearly see that the decreasing data in the first situation are much more variable (that is the trend is less uniform) if compared with the second plot. Then we can deduce that the border pixels caused that noisy oscillation that hide the real evolution of the lag signal.
Since we are considering the outcome of the curve identification, a further

consideration has to be done. We are interested in comparing the curve fitting and the weighted residuals obtained with the two different models applied, to decide which is the best function between the 2 or 3 exponential components. Considering the noisy data (those averaged over the whole panel), the simpler model seems to better work, since also the weighted residuals exhibit a good behaviour (Figure 3.15 and column of Table 3.1). Conversely the six parameters of the 3 exponential model show poor estimate precisions: that means the model is actually too complex to describe those noisy data (Figure 3.17 and column A of Table 3.5). The situation completely changes while we consider the data from the central region, displayed in Figure 3.16 and 3.18 (columns B of Tables 3.1 and 3.5). In this case, the model with only two components appears unsatisfactory, since the residuals exhibit a clear polarization. That means this model is not complex enough to fit those data. In the end, the most suitable curve we could find is the one illustrated in Figure 3.18, where 3 exponential components well describe the lag decay, with proper residuals and good estimates precision.

For all the aforementioned reasons, the model chosen to implement the lag correction was the one characterized by 3 exponential components identified over the detector's central region, with the selected parameters listed in column D of Table 3.5 and 3.6. But then, the following question was: is that model suitable to successfully describe the lag decay over the whole panel and with every possible exposure condition? The following observations suggest that the answer may be positive.
First of all, the parameters' estimates appear to be quite uniform over the four different methods applied. The only downside of the entire matrix identification (columns A and C) seems to be a loss of precision, but not different mean values.
Secondly, as already discussed, the tests with different *tube current* values exhibit uniform results, suggesting that different exposure levels doesn't change the lag decay behaviour.
At last, a deeper observation of the *maps of parameters* can provide some further information. The six matrices illustrated in Figure 3.19 show that only the parameters associated with the fastest components of the curve (b_1, a_1, b_2 and a_2) exhibit a weak correlation with a pattern that might be similar to the beam profile. On the contrary, the long lasting component (associated with b_3 and a_3) shows only a slight dependence on the typical dark current pattern, with some of the horizontal bands barely visible. Anyway, on the whole, the parameters seem to assume a random distribution, without a clear dependence on the position of the pixel on the panel, meaning that the lag decay has common characteristics almost everywhere. The only difference

that the regions close to the borders exhibit is that they are more likely to fail the model identification, probably due to the higher noise level.
In the end, all these analysis suggest that the lag phenomenon has sufficiently uniform characteristics. This should allow to successfully employ the estimated parameters (those of column C or D of Table 3.5) over the whole detector's matrix and with all exposure conditions.

At last, it is possible to briefly evaluate the correction outcome in the two tested situations. Both experiments, even if they illustrate different conditions, have quite the same results. The only frame where the model doesn't succeed to completely compensate the lag signal is the first one. In this case, a weak shadow of the employed phantom is still visible; anyway, as it can be observed, the residual signal's intensity was substantially reduced. In all the following frames the object projection are no more distinguishable, meaning that the correction algorithm completely deleted the residual signal. These are encouraging results, that will hopefully lead to a successful application of the correction state model, discussed at the end of the *Lag Effect Method* section. Indeed, the first frame (the one where the lag was not perfectly deleted) will probably never be used in the actual tomographic reconstruction, since it is usually acquired during the gantry rotation, and then only used to update the state model.

Chapter 4

Conclusion

The purposes of the study, discussed at the beginning of this work, have been almost completely achieved, and produced some interesting results.

The effects caused by the geometric imperfection in the source-detector alignment have been clearly exhibited. It was found that, even if the mistake committed during the panel placement was certainly small, the consequences on the quality of the reconstructed image were remarkable, as demonstrated by the serious artifact shown. That means a μ-CT device with those characteristics cannot neglect that kind of correction. Then, within a simulation framework the calibration algorithm capabilities were verified, proving the effectiveness of both methods implemented. Moreover, the complete procedure (the one that estimates the full set of misalignment parameters) may be further exploited, to investigate which are the real consequences of the *tilt* and *slant* rotations on a 3D reconstructed image. In the end, the correction protocol was successfully employed to modify the projected frames, and then it might be employed for future calibrations of imaging systems that share similar characteristics with the one here discussed.

The lag effect produced by the detector panel was shown and investigated in some particular acquisition conditions. Even if the detector employed exhibits a much lower lag signal than other sensor technologies commonly used in imaging systems, it was demonstrated that its presence is still undeniable and cannot be ignored. A precise description of its evolution was provided, and the compensation tests have been positively employed with static experiments, proving their capability to completely delete the residual signal. Starting from these results, future works should apply the identification performed and the suggested correction algorithm to check their effectiveness on a reconstructed CT image. Then it will be possible to finally verify the presence of an artifact caused by the lag signal. The same problem should be investigated with different conditions protocols, to reach a deeper comprehension of the phenomenon and to provide a more versatile correction procedure.

Bibliography

[1] F. P. Branca, *Fondamenti di Ingegneria Clinica*, vol. 1. Springer.

[2] J. T. Bushberg, J. A. Seibert, E. M. Leidholdt, and J. M. Boone, *The Essential Physics of Medical Imaging.* Lippincott Williams Wilkins.

[3] P. Dendy and B. Heaton, *Physics for Diagnostic Radiology.* CRC Press.

[4] E. N. Landis and D. T. Keane, "X-ray microtomography," *Materials Characterization*, vol. 61.

[5] A. C. Kak and M. Slaney, *Principles of Computerized Tomographic Imaging.* Society of Industrial and Applied Mathematics.

[6] L. von Smekal, M. Kachelrieß, E. Stepina, and W. A. Kalender, "Geometric misalignment and calibration in cone-beam tomography," *Medical Physics*, vol. 31, pp. 1305–1316, Dec 204.

[7] G. T. Gullberg, B. M. W. Tsui, C. R. Crawford, J. G. Ballard, and J. T. Hagius, "Estimation of geometrical parameters and collimator evaluation for cone beam tomography," *Medical Physics*, vol. 17.

[8] F. Noo, R. Clackdoyle, C. Mennessier, T. A. White, and T. J. Roney, "Analytic method based on identification of ellipse parameters for scanner calibration in cone-beam tomography," *Physics in Medicine and Biology*, vol. 45.

[9] J. Moore, *Adaptive X-Ray Computed Tomography.* PhD thesis, University of Arizona - College of Optical Sciences.

[10] M. Karolczak, S. Schaller, K. Engelke, A. Lutz, U. Taubenreuther, K. Wiesent, , and W. A. Kalender, "Implementation of a cone-beam algorithm for single-circle source orbit with embedded misalignment correction using homogeneous coordinates," *Medical Physics*, vol. 28.

[11] T. Bonham, *Exact Reconstruction of Adaptive Helical Micro-CT Data.* PhD thesis, University of Arizona - College of Optical Sciences.

[12] J. Starman, *Lag Correction in Amorphous Silicon Flat-Panel X-Ray computed Tomography.* PhD thesis, Stanford University - Department of Electrical Engineering.

[13] M. Overdick, T. Soft, and H.-A. Vischmann, "Temporal artefacts in flat dynamic x-ray detectors," *Medical Physics*, vol. 4320.

[14] *X-Ray Flat Panel Sensor - User's Guide.*

[15] H. Mori, R. Kyuushima, K. Fujita, and M. Honda, "High resolution and high sensitivity cmos panel sensors for x-ray," in *Nuclear Science Symposium Conference Record.*

[16] *Application Manual for Flat Panel Sensor.*

[17] J. H. Siewerdsen and D. A. Jaffray, "Spatio-temporal response characteristics of an indirect- detection flat-panel imager," *Medical Physics*, vol. 26, pp. 1624–1641, Sep 1999.

[18] L. Spies and R. Luhta, "Characterization and correction of temporal artifacts in ct," *Medical Physics*, vol. 32, no. 7, pp. 2222–2230, 2005.

[19] N. Mail, D. J. Moseley, H. Siewerdsen, and D. A. Jaffray, "An empirical method for lag correction in cone-beam ct," *Medical Physics*, vol. 35, no. 11, pp. 5187–5196, 2008.

[20] N. Mail, P. O'Brien, , and G. Panga, "Lag correction model and ghosting analysis for an indirect-conversion flat-panel imager," *Journal of applied clinical medical physics.*

[21] J. Hsieh, O. E. Gurmen, and K. F. King, "A recursive correction algorithm for detector decay characteristics in ct," in *Medical Imaging 2000: Physics of Medical Imaging.*

[22] J. Starman, J. Star-Lack, G. Virshup, E. Shapiro, and R. Fahrig, "Investigation into the optimal linear time-invariant lag correction for radar artifact removal," *Medical Physics*, vol. 38, p. 2398–2411, May 2011.

[23] M. Baumann, "Performance of a micro-ct system - characrerisation of hamamatsu x-ray source l10951-04 flat panel c7942ca-22," master of science thesis in medical engineering, KTH Royal Institute of Technology.

Appendix A

A.1 GATE simulation *macros*

The following code is a *macro* file (*CT_test*), suitable for the GATE environment, that describes the structure of the *World* framework, the detector panel and the radiation source.

```
################################
# VISUALIZATION PARAMETERS #
################################

/vis/disable
#/vis/open OGLSX
#/vis/viewer/set/viewpointThetaPhi 90 0
#/vis/viewer/zoom 3
#/vis/viewer/set/projection perspective
#/vis/viewer/set/lightsMove camera
##/vis/viewer/set/edge 1
##/vis/viewer/set/hiddenEdge 1
#/vis/drawVolume
#/tracking/storeTrajectory 1
#/vis/scene/add/trajectories
#/vis/scene/endOfEventAction accumulate
#/vis/scene/add/axes



############################
# SET MATERIAL DATABASE #
############################
```

```
/gate/geometry/setMaterialDatabase/GateMaterials.db



#########
# WORLD #
#########

/gate/world/geometry/setXLength 100. cm
/gate/world/geometry/setYLength 100. cm
/gate/world/geometry/setZLength 100. cm
/gate/world/vis/setVisible 1



############################################
# CT scanner for small animal imaging      #
# 512x512 pixels                           #
# size of pixels : 0.125x0.125x1.0 mm3     #
# pixels are made up of silicon            #
############################################

###############
# CT SCANNER #
###############
/gate/world/daughters/name CTscanner
/gate/world/daughters/insert box
/gate/CTscanner/placement/setTranslation
1.31 0.81 40.5 mm
/gate/CTscanner/geometry/setXLength 90 mm
/gate/CTscanner/geometry/setYLength 90 mm
/gate/CTscanner/geometry/setZLength 25 mm
/gate/CTscanner/placement/setRotationAxis 0 0 1
/gate/CTscanner/placement/setRotationAngle 0.75 deg
/gate/CTscanner/setMaterial Vacuum
/gate/CTscanner/vis/forceWireframe
/gate/CTscanner/vis/setColor white

##############        ############
# CTSCANNER #   ---->  #  MODULE  #
```

```
#############          ############
/gate/CTscanner/daughters/name module
/gate/CTscanner/daughters/insert box
/gate/module/geometry/setXLength 80 mm
/gate/module/geometry/setYLength 80 mm
/gate/module/geometry/setZLength 17 mm
/gate/module/placement/setRotationAxis 1 0 0
/gate/module/placement/setRotationAngle 1.25 deg
/gate/module/setMaterial Vacuum
/gate/module/vis/forceWireframe
/gate/module/vis/setColor white

############           #############
#  MODULE  #   ——>     # CLUSTER_0 #
############           #############
/gate/module/daughters/name cluster
/gate/module/daughters/insert box
/gate/cluster/geometry/setXLength 70 mm
/gate/cluster/geometry/setYLength 70 mm
/gate/cluster/geometry/setZLength 9 mm
/gate/cluster/placement/setRotationAxis 0 1 0
/gate/cluster/placement/setRotationAngle 1.0 deg
/gate/cluster/setMaterial Vacuum
/gate/cluster/vis/forceWireframe
/gate/cluster/vis/setColor white

############       #############       ###########
#  MODULE  # ——> # CLUSTER_0 # ——> # PIXEL_0 #
############       #############       ###########
/gate/cluster/daughters/name pixel
/gate/cluster/daughters/insert box
/gate/pixel/geometry/setXLength 0.125 mm
/gate/pixel/geometry/setYLength 0.125 mm
/gate/pixel/geometry/setZLength 1.0 mm
/gate/pixel/setMaterial Lead
/gate/pixel/vis/setColor red

# REPEAT PIXEl_0
/gate/pixel/repeaters/insert cubicArray
/gate/pixel/cubicArray/setRepeatNumberX 512
/gate/pixel/cubicArray/setRepeatNumberY 512
```

```
/gate/pixel/cubicArray/setRepeatNumberZ    1
/gate/pixel/cubicArray/setRepeatVector
0.125 0.125 0.0 mm
/gate/pixel/cubicArray/autoCenter true

# ATTACH SYSTEM
/gate/systems/CTscanner/module/attach module
/gate/systems/CTscanner/cluster_0/attach cluster
/gate/systems/CTscanner/pixel_0/attach pixel

# ATTACH LAYER
/gate/pixel/attachCrystalSD

# ROTATE CT detector
#/gate/CTscanner/repeaters/insert ring
#/gate/SPECThead/ring/setRepeatNumber 4
#/gate/CTscanner/moves/insert orbiting
#/gate/CTscanner/orbiting/setSpeed 2 deg/s
#/gate/CTscanner/rotation/setAxis 0 1 0
#/gate/CTscanner/orbiting/setPoint1 0 0 0 cm
#/gate/CTscanner/orbiting/setPoint2 0 1 0 cm



###########
# PHANTOM #
###########

/control/execute                    CT_test_phantom.mac



###########
# PHYSICS #
###########

##############
# EM PROCESS #
##############

/gate/physics/addProcess PhotoElectric
```

```
/gate/physics/processes/PhotoElectric/setModel
StandardModel

/gate/physics/addProcess Compton
/gate/physics/processes/Compton/setModel PenelopeModel

/gate/physics/addProcess RayleighScattering
/gate/physics/processes/RayleighScattering/setModel
PenelopeModel

/gate/physics/addProcess ElectronIonisation
/gate/physics/processes/ElectronIonisation/setModel
StandardModel e-

/gate/physics/addProcess Bremsstrahlung
/gate/physics/processes/Bremsstrahlung/setModel
StandardModel e-

/gate/physics/addProcess MultipleScattering e-

/gate/physics/processList Enabled
/gate/physics/processList Initialized



##################
# INITIALIZATION #
##################

/gate/run/initialize



##################
#      MOVE      #
##################

/gate/timing/setTime 0 s                    # 0 degree
/gate/timing/setTime 15 s                   # 30 degrees
/gate/timing/setTime 30 s                   # 60 degrees
/gate/timing/setTime 45 s
```

```
/gate/timing/setTime 60 s
/gate/timing/setTime 75 s
/gate/timing/setTime 90 s
/gate/timing/setTime 105 s
/gate/timing/setTime 120 s
/gate/timing/setTime 135 s
/gate/timing/setTime 150 s
/gate/timing/setTime 165 s
/gate/timing/setTime 180 s              # 360 degrees


#############
# DIGITIZER #
#############

/gate/digitizer/Singles/insert adder
/gate/digitizer/Singles/insert readout
/gate/digitizer/Singles/readout/setDepth 2
/gate/digitizer/Singles/insert thresholder
/gate/digitizer/Singles/thresholder/setThreshold
10 keV
/gate/digitizer/convertor/verbose 0
/gate/digitizer/verbose 0



##############
# SOURCE GPS #
##############

/control/execute         xrayspectrum.mac

#ATTACH SOURCE TO DETECTOR FOR ROTATION
#/gate/source/xraygun/attachTo CTscanner
#/gate/source/xraygun/gps/halfx 0.025 mm
#/gate/source/xraygun/gps/halfy 0.025 mm
/gate/source/xraygun/gps/mintheta 0  deg
/gate/source/xraygun/gps/maxtheta  5.5 deg
/gate/source/xraygun/gps/centre 0.0 0.0 -284 mm
/gate/source/xraygun/gps/angtype iso
```

```
/gate/source/list

#/gate/source/xraygun/visualize 2000 yellow 2

##########
# OUTPUT #
##########

#/gate/output/allowNoOutput

/gate/output/ascii/enable
/gate/output/ascii/setFileName test
/gate/output/ascii/setOutFileHitsFlag 0
/gate/output/ascii/setOutFileSinglesFlag 1
/gate/output/ascii/setSingleMask 1 0 0 0 0 0 1

#/gate/output/sinogram/enable
#/gate/output/sinogram/setFilename
CT_test_20150109_sinogram

#/gate/output/imageCT/enable
#/gate/output/imageCT/verbose 0
#/gate/output/imageCT/setFileName
output_20150114/ball_in_box/



#############
# VERBOSITY #
#############

# NO VERBOSITY



###############
# ACQUISITION #
###############

##################################
# ACQUISITION for 1 projection #
```

```
##################################
#/gate/application/setTimeSlice      1. s #90. s
#/gate/application/setTimeStart      0.  s
#/gate/application/setTimeStop       13. s #180. s

######################################
# ACQUISITION with 180 projections #
######################################
/gate/application/setTimeSlice      1. s
/gate/application/setTimeStart      0.  s
/gate/application/setTimeStop       180. s

/gate/random/setEngineSeed 13029

/gate/application/startDAQ

/vis/viewer/set/viewpointThetaPhi 180 0
```

The following code is a *macro* file (*CT_test_phantom*), suitable for the GATE environment, that describes the shape and movement of the simulated phantom.

```
#####################
# PHANTOM - MEDIUM #
#####################

/gate/world/daughters/name phantomMedium
/gate/world/daughters/insert cylinder
/gate/phantomMedium/geometry/setRmax 23.5 mm
/gate/phantomMedium/geometry/setHeight 59.5 mm
/gate/phantomMedium/setMaterial Air
/gate/phantomMedium/vis/forceWireframe
/gate/phantomMedium/vis/setColor          yellow
#/gate/phantomMedium/placement/setTranslation 0 0 0 mm
/gate/phantomMedium/placement/setRotationAxis 1 0 0
/gate/phantomMedium/placement/setRotationAngle 90. deg
```

```
##############
# LEAD BALLS #
##############

/gate/phantomMedium/daughters/name ball
/gate/phantomMedium/daughters/insert sphere
/gate/ball/geometry/setRmin 0. mm
/gate/ball/geometry/setRmax 0.75 mm
/gate/ball/setMaterial Lead
/gate/ball/vis/forceSolid
/gate/ball/vis/setColor green
/gate/ball/placement/setTranslation 15.75 12.75 0. mm
/gate/ball/repeaters/insert linear
/gate/ball/linear/setRepeatNumber 12
/gate/ball/linear/setRepeatVector 0. 0. 5. mm

##################
# ROTATE PHANTOM #
##################

/gate/phantomMedium/moves/insert rotation
/gate/phantomMedium/rotation/setSpeed 2 deg/s
/gate/phantomMedium/rotation/setAxis 0 0 1 mm
```

The following code is a *macro* file (*xrayspectrum*), suitable for the GATE environment, that describes the photons distribution according to the range of energies available.

```
# X-ray Spectrum simulation (v1)
# Anode material: tungsten
# Peak tube voltage: 40 kV
# Relative voltage ripple: 0.05
# Air kerma: 100 Gy
# Mean energy: ca. 27.34 keV
# Additional filter: Al (1 mm)
```

```
/gate/source/addSource xraygun
/gate/source/verbose 0
/gate/source/xraygun/setActivity 1500000. becquerel
/gate/source/xraygun/gps/verbose 0
/gate/source/xraygun/gps/particle gamma
/gate/source/xraygun/gps/energytype Arb
/gate/source/xraygun/gps/histname arb
/gate/source/xraygun/gps/emin 8.00 keV
/gate/source/xraygun/gps/emax 41.00 keV
/gate/source/xraygun/gps/histpoint   0.008   0
/gate/source/xraygun/gps/histpoint   0.009   1
/gate/source/xraygun/gps/histpoint   0.01   197
/gate/source/xraygun/gps/histpoint   0.011   3595
/gate/source/xraygun/gps/histpoint   0.012   48288
/gate/source/xraygun/gps/histpoint   0.013   341176
/gate/source/xraygun/gps/histpoint   0.014   1667003
/gate/source/xraygun/gps/histpoint   0.015   5619291
/gate/source/xraygun/gps/histpoint   0.016   12777416
/gate/source/xraygun/gps/histpoint   0.017   25000477
/gate/source/xraygun/gps/histpoint   0.018   40821022
/gate/source/xraygun/gps/histpoint   0.019   63537741
/gate/source/xraygun/gps/histpoint   0.02   86649385
/gate/source/xraygun/gps/histpoint   0.021   109765536
/gate/source/xraygun/gps/histpoint   0.022   127535345
/gate/source/xraygun/gps/histpoint   0.023   146619806
/gate/source/xraygun/gps/histpoint   0.024   158073716
/gate/source/xraygun/gps/histpoint   0.025   166160684
/gate/source/xraygun/gps/histpoint   0.026   168010475
/gate/source/xraygun/gps/histpoint   0.027   166762350
/gate/source/xraygun/gps/histpoint   0.028   167681812
/gate/source/xraygun/gps/histpoint   0.029   166711195
/gate/source/xraygun/gps/histpoint   0.03   166408507
/gate/source/xraygun/gps/histpoint   0.031   155979284
/gate/source/xraygun/gps/histpoint   0.032   141113903
/gate/source/xraygun/gps/histpoint   0.033   124735347
/gate/source/xraygun/gps/histpoint   0.034   105537709
/gate/source/xraygun/gps/histpoint   0.035   83935794
/gate/source/xraygun/gps/histpoint   0.036   59972541
/gate/source/xraygun/gps/histpoint   0.037   33885699
/gate/source/xraygun/gps/histpoint   0.038   18615533
```

```
/gate/source/xraygun/gps/histpoint    0.039    6825936
/gate/source/xraygun/gps/histpoint    0.04    1985748
/gate/source/xraygun/gps/histpoint    0.041    0
/gate/source/xraygun/gps/arbint Lin
/gate/source/xraygun/gps/type Plane
/gate/source/xraygun/gps/shape Rectangle
```

A.2 GATE to LabVIEW conversion code

The following MatLab code's purpose is to convert the GATE output information in *raw* files suitable for the LabVIEW environment. Geometric and format transformations are also performed.

```
clear all
close all
clc

test = load('testSingles.dat-fix.dat');
disp('load of test completed')
test1 = load('testSingles_1.dat-fix.dat');
disp('load of test1 completed')
test2 = load('testSingles_2.dat-fix.dat');
disp('load of test2 completed')
test3 = load('testSingles_3.dat-fix.dat');
disp('load of test3 completed')
test4 = load('testSingles_4.dat-fix.dat');
disp('load of test4 completed')

nPix = 512;
image_sum = zeros(nPix,nPix);

for Nproj = 0:179

    indici = find(test(:,1)==Nproj);
    matrice = test(indici,:);
    indici1 = find(test1(:,1)==Nproj);
    matrice1 = test1(indici1,:);
    indici2 = find(test2(:,1)==Nproj);
    matrice2 = test2(indici2,:);
```

```
indici3 = find(test3(:,1)==Nproj);
matrice3 = test3(indici3 ,:);
indici4 = find(test4(:,1)==Nproj);
matrice4 = test4(indici4 ,:);

matrix = [matrice
    matrice1
    matrice2
    matrice3
    matrice4];

image = zeros(nPix,nPix);
l = length(matrix(:,1));
PixVect = matrix(:,2);

for k = 1:l
    val = PixVect(k);
    Xindex = floor(val/nPix) + 1;
    Yindex = mod(val,nPix) + 1;
    image(Xindex,Yindex) = image(Xindex,Yindex)+1;
end

maxHit = max(max(image));
histogram = zeros(maxHit+1,2);
histogram(:,1) = 0:1:maxHit;
for i = 1:nPix
    for j = 1:nPix
        elem = image(i,j);
        histogram(elem+1,2) =
            histogram(elem+1,2) + 1;
    end
end

image_sum = image_sum + image;

image256 = zeros(nPix,nPix);
imageuint8 = zeros(nPix,nPix,'uint8');

% production of an image with 256 levels of gray
istogramma = zeros(256,2);
istogramma(:,1) = 0:1:255;
```

```
    for i = 1:nPix
        for j = 1:nPix
            elem = image(i,j);
            newelem = round((elem/maxHit)*255);
            image256(i,j) = newelem;
            istogramma(newelem+1,2) =
                istogramma(newelem+1,2) + 1;
        end
    end
    % production of an image in uint8
    for p = 1:nPix
        for q = 1:nPix
            element = image256(p,q);
            newelement = uint8(element);
            imageuint8(p,q) = newelement;
        end
    end

    % production of a Black&White image
    imagebw = im2bw(imageuint8,17/255);

    neg_image = imcomplement(image);
    neg_imageuint8 = imcomplement(imageuint8);
%     neg_imageuint8 = neg_imageuint8+254;
    neg_imagebw = imcomplement(imagebw);

%     figure, imshow(neg_image,[])
%     figure, imshow(neg_imageuint8,[])
%     figure, bar(histogram(:,1),histogram(:,2))
%     figure, bar(istogramma(:,1),istogramma(:,2))
%     figure, imshow(imagebw,[])
%     figure, imshow(neg_imagebw,[])
%     pause(0.5)

    vettore = zeros(nPix*nPix,1);
    contatore = 1;
    for v = 1:nPix
        for w = 1:nPix
            elemento = neg_imageuint8(v,w);
            vettore(contatore) = elemento;
            contatore = contatore+1;
        end
    end
    if Nproj>=0 && Nproj<9
        fid=fopen(['simulationFOUR_00'
            num2str(Nproj+1) '.raw'],'w+');
    end
    if Nproj>=9 && Nproj<99
        fid=fopen(['simulationFOUR_0'
            num2str(Nproj+1) '.raw'],'w+');
    end
    if Nproj>=99 && Nproj<=179
        fid=fopen(['simulationFOUR_'
            num2str(Nproj+1) '.raw'],'w+');
    end
    cnt = fwrite(fid,vettore,'uint8');
    fclose(fid);

    disp(['projection ' num2str(Nproj+1)
        '/180 completed'])
end
```

A.3 Acquired data to LabVIEW conversion code

The following MatLab code transforms the data acquired from the real CT device in files suitable for the LabVIEW environment. Geometric and numerical modifications are applied as well.

```
clear all
close all
clc

d = dir('Test3/*.bin');          % name of the DC file
filenames = {d.name};
Nproj = length(filenames);
disp(['Number of projections: ' num2str(Nproj)])

for i = 1:Nproj
    raw = loadBinAndHeader(['Test3/' filenames{i}]);
```

```
nPix = size(raw,1);
%disp(['dim = ' num2str(nPix)])
%figure(1), imshow(raw,[])

raw1 = flipud(raw);
%figure(2), imshow(raw1,[])
raw2 = rot90(raw1);
%figure(3), imshow(raw2,[])

raw3 = raw2 .* 16;
M = max(max(raw3));
raw4 = -(raw3) + M;

imm = zeros(nPix,nPix,'uint16');
for s = 1:nPix
    for t = 1:nPix
        imm(s,t) = raw4(s,t);
    end
end

imm2 = imadjust(imm,[],[],10);
%imm3 = im2bw(imm2,45000/65536);

vettore = zeros(nPix*nPix,1,'uint16');
contatore = 1;
for v = 1:nPix
    for w = 1:nPix
        elemento = imm2(v,w);
        vettore(contatore) = elemento;
        contatore = contatore+1;
    end
end

if i>=1 && i<10
    fid=fopen(['test3v4_00' num2str(i)
         '.raw'],'w+');
end
if i>=10 && i<100
    fid=fopen(['test3v4_0' num2str(i)
         '.raw'],'w+');
end
```

```
    if i>=100 && i<=180
        fid=fopen(['test3v4_' num2str(i)
              '.raw'],'w+');
    end
    cnt = fwrite(fid,vettore,'uint16');
    fclose(fid);

    disp(['projection ' num2str(i) '/180 completed'])
end
```

A.4 Reconstruction misalignment correction code

The following Matlab code's purpose is to apply the estimated misalignment parameters to correct each frame before the reconstruction step.

```
function [corrected]=applyCorrections
(img, lightfieldimage, darkcurrentimage)

% dark image correction and light field correction
warning off;
lfc=lightfieldimage-darkcurrentimage;
lfc(lfc<0)=0;
dc=img-darkcurrentimage;
dc(dc<0)=0;

corrected=-log(dc./lfc);

corrected(isnan(corrected))=0;
corrected(isinf(corrected))=0;
corrected(corrected<0)=0;

%Shifting projections
last_row = 2239;
last_col = 2343;

fascia_oriz = corrected(1:23,1:last_col);
fascia_vert = corrected(1:last_row,1:19);
f_oriz_23x7 = [fascia_oriz; fascia_oriz; fascia_oriz;
fascia_oriz; fascia_oriz; fascia_oriz; fascia_oriz];
f_vert_19x3 = [fascia_vert fascia_vert fascia_vert];
quadrato = f_oriz_23x7(1:161,1:57);
f_oriz = [f_oriz_23x7 quadrato];

corrected(2240:end,1:end) = f_oriz;
corrected(1:2239,2344:end) = f_vert_19x3;

corrected = imrotate(corrected,0.139,'crop');

pUP = 27;
pLEFT = 15;

Mup = corrected(1:pUP,1:end);
Mdown = corrected(pUP+1:end,1:end);
corrected = [Mdown; Mup];

Mleft = corrected(1:end,1:pLEFT);
Mright = corrected(1:end,pLEFT+1:end);
corrected = [Mright Mleft];

warning on;
```

A.5 Model identification

The following code is the main function for the WNLLS model identification employed to fit the lag decay data. This version of the procedure is the one that divides each frame in a collection of subregions, and then evaluates the model parameters and the fitting curve for each of them spearately.

```
clear all
close all
clc

Nrows = 2100;
Ncols = 2300;

%% Dark Current
```

```
d = dir('DC800/*.bin');
filenames = {d.name};
Nproj = length(filenames);

DCtemp = zeros(2400,2400);
newDC = zeros(Nrows,Ncols);
DCsum = zeros(Nrows,Ncols);
m_value = zeros(Nproj,1);
count = 0;
for i = 1:Nproj
    count = count + 1;
    DCtemp(:,:) = loadBinAndHeader(['DC800/'
        filenames{i}]);
    newDC(:,:) = DCtemp(1:Nrows,1:Ncols);
    DCsum = DCsum + newDC;
    m_value(i) = mean(mean(newDC));
    disp(['DC iteration ' num2str(i) ' completed'])
end

meanDC = DCsum./count;

clearvars -except Nrows Ncols meanDC

%% Light Field - Beam Profile

d = dir('LE800/*.bin');
filenames = {d.name};

LFtemp = zeros(2400,2400);
newLF = zeros(Nrows,Ncols);
LFsum = zeros(Nrows,Ncols);
cnt = 0;

for i = 11:50
    cnt = cnt + 1;
    LFtemp(:,:) = loadBinAndHeader(['LE800/'
        filenames{i}]);
    newLF(:,:) = LFtemp(1:Nrows,1:Ncols);
    LFsum = LFsum + newLF;
    disp(['LF iteration ' num2str(i) ' completed'])
end
```

```
meanLF = LFsum./cnt;

clearvars -except Nrows Ncols meanDC meanLF

%% Lag Effect

d = dir('LE800/*.bin');
filenames = {d.name};
Nproj = length(filenames);

LEtemp = zeros(2400,2400);
newLE = zeros(Nrows,Ncols);
LEcorr = zeros(Nrows,Ncols);
M = meanLF - meanDC;
vm = mean(mean(M));
idx = find(M == 0);
M(idx) = 1;

M25 = zeros(100,1890);
ri = 1;
ci = 51;
rf = 2051;
cf = 2251;

for i = 1:Nproj
    LEtemp(:,:) = loadBinAndHeader(['LE800/'
        filenames{i}]);
    newLE(:,:) = LEtemp(1:Nrows,1:Ncols);
    LEcorr = ((newLE - meanDC)./M)*vm;

    cont = 0;
    for p = ri:50:rf
        for q = ci:50:cf
            cont = cont + 1;
            ROI = LEcorr(p:p+49,q:q+49);
            avrg = mean(mean(ROI));
            M25(i,cont) = avrg;
        end
    end
    disp(['LE iteration ' num2str(i) ' completed'])
end

figure, plot(M25,'.-')

data = zeros(31,2);
times = 0:30;
data(:,1) = times;
M25 = M25(50:80,:);

%% Model Identification
Nexp = input('How many exponentials (1, 2, 3 or 4)? ');
Mparam = zeros(6,1890);
for r = 1:1890
    data(:,2) = M25(:,r);
    [p_est,cv_p,Nexp,cv,AIC,y,wres] =
        LE_modelIdentif(data,Nexp);

    newTimes = data(1,1):0.01:data(end,1);

    if Nexp == 1
        newy = p_est(1)*exp(-p_est(2)*newTimes);
    end
    if Nexp == 2
        newy = p_est(1)*exp(-p_est(2)*newTimes)
            + p_est(3)*exp(-p_est(4)*newTimes);
    end
    if Nexp == 3
        newy = p_est(1)*exp(-p_est(2)*newTimes)
            + p_est(3)*exp(-p_est(4)*newTimes)
            + p_est(5)*exp(-p_est(6)*newTimes);
    end
    if Nexp == 4
        newy = p_est(1)*exp(-p_est(2)*newTimes)
            + p_est(3)*exp(-p_est(4)*newTimes)
            + p_est(5)*exp(-p_est(6)*newTimes)
            + p_est(7)*exp(-p_est(8)*newTimes);
    end

    Mparam(:,r) = [p_est(1); p_est(2); p_est(3);
        p_est(4); p_est(5); p_est(6)];
    disp(['identification ' num2str(r) ' completed'])
end

p1 = Mparam(1,:);
p2 = Mparam(2,:);
p3 = Mparam(3,:);
p4 = Mparam(4,:);
p5 = Mparam(5,:);
p6 = Mparam(6,:);
hm1 = reshape(p1,45,42);
hm2 = reshape(p2,45,42);
hm3 = reshape(p3,45,42);
hm4 = reshape(p4,45,42);
hm5 = reshape(p5,45,42);
hm6 = reshape(p6,45,42);

figure, imshow(hm1',[2460 2530])
figure, imshow(hm2',[3.55 3.9])
figure, imshow(hm3',[2 8])
figure, imshow(hm4',[0.1 0.6])
figure, imshow(hm5',[2 4])
figure, imshow(hm6',[0.015 0.04])

Mparam = [Mparam(:,1:1119) Mparam(:,1121:end)];

mean1 = mean(Mparam(1,:));
mean2 = mean(Mparam(2,:));
mean3 = mean(Mparam(3,:));
mean4 = mean(Mparam(4,:));
mean5 = mean(Mparam(5,:));
mean6 = mean(Mparam(6,:));

std1 = std(Mparam(1,:));
std2 = std(Mparam(2,:));
std3 = std(Mparam(3,:));
std4 = std(Mparam(4,:));
std5 = std(Mparam(5,:));
std6 = std(Mparam(6,:));

disp(['b1 = ' num2str(mean1/vm)])
disp(['SD of b1: ' num2str(std1/vm)])
disp(['a1 = ' num2str(mean2)])
disp(['SD of a1: ' num2str(std2)])
disp(['b2 = ' num2str(mean3/vm)])
disp(['SD of b2: ' num2str(std3/vm)])
disp(['a2 = ' num2str(mean4)])
disp(['SD of a2: ' num2str(std4)])
disp(['b3 = ' num2str(mean5/vm)])
disp(['SD of b3: ' num2str(std5/vm)])
disp(['a3 = ' num2str(mean6)])
disp(['SD of a3: ' num2str(std6)])
```