Direct Estimation of Dynamic Parameters in SPECT Tomography

E.Hebber (Haber)[†*], D.Oldenburg[†], T.Farnocombe[‡], A.Celler[‡],

* - The University of Calgary, Faculty of Medicine, Calgary, Canada. †UBC - Geophysical Inversion Facility, Department of Earth and Ocean Science, Vancouver, Canada ‡ Vancouver Hospital and Health Sciences Centre, Vancouver, Canada

Abstract

This paper solves the problem of estimating dynamic parameters of SPECT reconstruction directly from the projections. This is done by spanning the dynamic parameters of each pixel with a basis of exponentials. Any time decay system can be modeled in this way. This process leads to a linear Fredholm integral equation of the first kind which is discretized and then solved with conjugate gradient methods. We tested our algorithm by applying it to a numerical phantom and to a simple real phantom.

I. INTRODUCTION

Single Photon Emission Computed Tomography (SPECT) and Positron Emission Tomograph (PET) are two methods of obtaining functional information. This information is recovered by reconstructing the distribution of tracers. Since the tracer is designed to target a specific physiological activity, its distribution contains information about areas which are physiologically active. Standard reconstruction methods do not indicate how the activity changes in time. This is the goal of dynamic studies.

Dynamic SPECT or PET tomography is the study of the kinetics of a tracer, i.e. the tracer distribution in time. Modelling of such behaviour suggests that it is an exponentially decaying process that can be characterized with a few exponentials [1].

The reconstruction of dynamic parameters was discussed in [1][2]. There are two approaches to the reconstruction. A two-stage approach is suggested in [1]. In the first stage image reconstruction at various times is performed, and the dynamic analysis is done by looking at each spatial pixel from images acquired at the different times. The information about the dynamic process contained in each pixel can then be retrieved by a simple parameter estimation which fits an exponential curve to the data for each pixel. Carrying out such reconstructions requires the ability to reconstruct the images at different times. A good quality reconstruction, requires acquisition of a large number of projections, and therefore, can be done only if projections are collected in a relatively short time. Yet, projection data is a Poisson process in time, which means that a relatively long time is needed in order to achieve a good signal-to-noise ratio. Obtaining high signal-to-noise ratio data in short time is therefore difficult. It was demonstrated in [2], that reconstructing an image using data collected at different times could result in a distorted image. One way to overcome this difficulty is to estimate dynamic parameters directly from the projections. The method in [2] assumes that the dynamic process is an exponential model and requires *a priori* information about the number of exponentials which are present in the dynamic model. It then uses a non-linear reconstruction technique in order to obtain its dynamic parameters and the initial amplitude.

There are two disadvantages to this technique. Firstly, one has to know, or assume, how many dynamic parameters are present in the model. Even if the number of dynamic parameters is known *a priori* for one biological cell, every pixel in the model usually represents a sum of biological cells which might have different dynamic parameters, and therefore should be described as a sum of dynamic parameters. Exponential fitting is a poorly determined problem, especially when the data are noisy. This means that results could be wrong depending on the assumptions about the number of dynamic parameters. Secondly, the problem of estimating only a few dynamic parameters is non-linear, and this makes the reconstruction computationally expensive.

In order to avoid the assumptions of a fixed number of exponentials and to make the problem linear, we turn to a different methodology. A similar methodology was suggested in [3] for NMR studies and in [4] for dynamic PET, and we generalize it to our problem.

The paper begins with a description of our methodology. We then discuss the numerical aspects of the reconstruction and give a synthetic example. Finally we carry out a simple experiment to verify our method.

II. BASIC EQUATIONS AND METHODOLOGY

Let $a(x,y,t)$ be the time dependent tracer distribution model under reconstruction. The tomography acquisition is performed and $d_i(t_j)$, the datum in bin i at time j, is collected. This can be represented mathematically by the Radon transform:

$$d_i(t_j) = \int_{c_i} a(x,y,t_j)dl \quad i = 1..n_r, \quad j = 1..n_t \qquad (1)$$

where c_i is the i^{th} ray. We cannot solve equation (1) by regular methods such as back projection because the model a changes in time. To proceed, we must make assumptions about the time change of a. The approach taken in most dynamic SPECT studies [1][2] has been to assume that a has the functional form:

$$a(x,y,t) = m_2(x,y)e^{-\lambda_2(x,y)t} + m_1(x,y)e^{-\lambda_1(x,y)t} \qquad (2)$$

There are a few disadvantages to this approach. The first is that all models obtained by this method are limited to having only two dynamic parameters. However the physiology might

0018-9499/97$10.00 © 1997 IEEE

be quite different and require a different number of dynamic parameters. Even when the number of dynamic parameters is known, it is known only in one biological cell. Since a pixel is a sum of biological cells this assumption is violated. Reconstruction in terms of two exponentials does not likely simulate the real time behaviour. Instead of fixing the number of exponentials, it would be better to let the projection data determine the number of exponentials needed. A second disadvantage is that this technique leads to a large nonlinear system which possesses a "flat" minimum and therefore has many feasible solutions, especially when the data are noisy.

A different approach which increases the size of the problem, but keeps it linear, is as follows. Instead of building the model from only a finite set of exponentials we build it from a basis of exponentials, as in a Laplace transform:

$$a(x, y, t) = \int_0^\infty m(x, y, \lambda) e^{-\lambda t} d\lambda \qquad (3)$$

This approach was suggested in [3] for the reconstruction of NMR decay curves and in [4] for reconstruction of PET curves. Although [3] and [4] implemented this scheme for each pixel separately it could be applied for the whole image.

Notice that if $m(x, y, \lambda) = m_1 \delta(\lambda - \lambda_1) + m_2 \delta(\lambda - \lambda_2)$ this model is identical to (2). We therefore see that our approach is a generalization of the previous work that uses equation (2) and, it allows us to work with any time decaying model.

By substituting (3) into (1) we get:

$$d_i(t_j) = \int_{c_i} \int_0^\infty m(x, y, \lambda) e^{-\lambda t_j} d\lambda \; dl \qquad (4)$$

Equation (4) is a linear integral equation of the first kind for $m(x, y, \lambda)$ (see for example [5]). The nonlinearity of finding a few dynamic parameters λ, is replaced in finding a whole spectrum $m(x, y, \lambda)$. Representing the problem in this way increases the size of our problem, however since the problem is linear and extremely sparse, we do not expect to increase the amount of computations needed for the reconstruction.

The first stage of the solution is to discretize equation (4). We discretize linearly in the spatial domain (x, y), and logarithmically in λ. The integration in (4) is evaluated using the midpoint rule. This leads to a linear algebraic system of equations:

$$G\, m = d + \epsilon \qquad (5)$$

where G is the discretized kernel function, G_{ij} is the area of the i^{th} ray in the j^{th} cell, $m = m(x, y, \lambda)$; $m \in R^M$ is the discretized model, $d \in R^N$ is the collected projection data, and ϵ is the noise. Since the model m, which we seek, is a function, we want to discretize it as finely as possible. The result is that G is of size $N \times M$ with $N < M$, which means that the problem in (5) is typically under-determined and very ill-posed.

Ill-posed under-determined systems have an infinite number of solutions. To handle this difficulty, Tichonov [6] presented the idea of regularization. The main idea is that from all possible solutions we want to choose the one with the smallest norm $||m||_2^2 = \int m(x)^2 dx$, or more generally the smallest weighted norm $||Wm||_2^2 = \int [w(x)m(x)]^2 dx$, where $w(x)$ is a weighting function. Our inverse problem is transformed into a minimization problem:

minimize:

$$||Wm|| \qquad (6)$$

subject to

$$||G\, m - d||^2 \leq \phi_d{}^*$$

The misfit ϕ_d^* relates to the noise level ϵ. If $\epsilon = 0$ then $\phi_d^* = 0$. Unfortunately the data are contaminated with noise and therefore we have to choose a value for ϕ_d^*. We discuss our choice later.

In order to solve the constrained minimization problem (6) we form a global unconstrained minimization problem and minimize:

$$\phi = \beta ||Wm||^2 + ||G\, m - d||^2 \qquad (7)$$

where β plays the role of a penalty parameter. As β increases, the norm $||Wm||$ decreases but the misfit $||G\, m - d||^2$ grows. β is chosen such that the misfit is equal to ϕ_d^*. The solution to the unconstrained minimization problem is:

$$m(\beta) = (G^T G + \beta W^T W)^{-1} G^T d \qquad (8)$$

This can also be represented as a solution to the over-determined system

$$\begin{bmatrix} G \\ \beta W \end{bmatrix} m = \begin{bmatrix} b \\ 0 \end{bmatrix} \qquad (9)$$

Before solving the system we have to decide about a strategy for the choice of the target misfit ϕ_d^*. There are a few possibilities, [7],[8],[9]. Since the L-curve [7] is very simple to implement and gives satisfactory results, we chose our misfit using that technique.

III. Numerical Implementation

A. Choice of Weighting Matrix

Using a Tichonov formulation gives the advantage of choosing a weighting matrix W which incorporates some of our *a priori* information about the problem. Our goal in choosing the weighting matrix is to automatically specify a support region for the source activity (i.e. find the active region) and to impose smoothness in the spatial domain. We therefore choose our weighting by noting first that there are many data which are zero. This means that the rays which generated these data probably did not pass through any active area. We define such a ray as a null ray. Our goal is to locate pixels through which the null rays have passed so we can discriminate against having some activity in these pixels. Let I_z be the index of all null rays. We sum the rows of the null rays together:

$$r_{z(j)} = \sum_{i \in I_z} G_{ij} \qquad (10)$$

where G is the tomography system matrix, and each row of G represents a ray. Each element of the vector r_z is the area of all the null rays in a particular pixel. If this number is large, then many null rays pass through this pixel. However if this number is 0, no null rays pass through this specific pixel and we can therefore assume that the probability of having activity in the pixel is high. We normalize this result by dividing by the total area of rays which pass through the pixel, and hence the final weighting is:

$$w_j = \sum_{i \in I_z} G_{ij} (\sum_{i=1}^{N} G_{ij})^{-1} \qquad (11)$$

The result of this weighting for a numerical experiment which we describe in section four, is seen in Figure (1). The weights are incorporated in the matrix W so that pixels with large w_j are penalized.

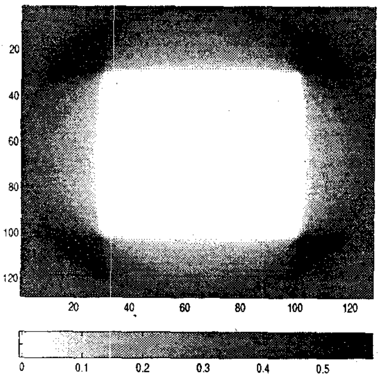

Fig. 1 The weighting used in the numerical experiment. Notice that there are no null rays in the center (where the activity is) and many null rays on the side (no activity)

In order to make the solution smooth, we use a spatial domain (x, y) second order finite difference as an approximation to the Laplacian, effectively minimizing $||Lm(x, y, \lambda_{fix}||_2$ in the spatial domain, where L is the discretized Laplacian, and $(.)_{fix}$ implies that the operator does not apply to the fixed variable. In the spectral (λ) domain we want a small solution and therefore minimize $||m(x_{fix}, y_{fix}, \lambda)||_2$. Thus the matrix W is constructed from:

- weighting the null rays in the spatial domain,
- second order finite difference matrixes which generate smoothness in the spatial domain,
- identity matrix for the spectral domain component which generates a small model in the λ direction.

We expect to generate a spatially smooth model mainly inside the unpenalized area, and at each pixel, the norm in the λ domain, $||m(x_{fix}, y_{fix}, \lambda)||_2$, should be small.

B. Solution of the System

Theoretically the solution is straight forward, but numerically we cannot solve the system (7) directly. The matrix G is constructed by discretizing the model space into a large number (128×128 in our example) of spatial pixels and, as many λ's as possible for each pixel (64 in our case). Thus we have $128 \times 128 \times 64 = 1048576$ unknowns. There are 64 projections and 64 bins in each projection giving 4096 data. The matrix G cannot be stored on a regular workstation and therefore a direct method to invert the matrix cannot be carried out.

We turn our attention to iterative methods. One of the fastest iterative techniques for solving large sparse problems is least square QR (LSQR) which was developed in [10]. It is possible to show theoretically that LSQR is equivalent to the conjugate gradient least squares (CGLS) technique described in [7] which has been used in various reconstruction algorithms. Recent studies show that the LSQR is slightly more stable than CGLS [11]. The method is specifically suited to our problem since it only requires the calculation of the matrix times a vector and the matrix transpose times a vector, and therefore the matrix G need not be formed or stored. It has been shown in [7] that CGLS and LSQR also regularize the reconstruction, by stopping after a few iterations. We therefore use LSQR to solve the system. To do so we must compute the product of G and a vector and G^T and a vector.

Since we do not store the matrix G, we develop a method to calculate the product Gv and $G^T u$, where u and v are arbitrary vectors. The operation of G can be divided into two parts. In the first part the Laplace transform L_λ, is taken and in the second part the Radon transform R_{xy} is performed. This can be written as:

$$Gv = R_{xy} L_\lambda v = R_{xy}[L_\lambda v] \qquad (12)$$

We can calculate the product Gv by performing each of these processes independently. The matrix vector multiplication is then divided into two parts. In the first part we transform each pixel from the Laplace domain to the time domain by applying the discrete Laplace transform. We then operate on the time domain model with the discrete Radon transform. Since the Laplace transform is applied only to the λ domain the size of L_λ is small: it is the number of times that the data were collected (64 in our example) multiplied by the number of λ's (64 in our example). The Radon transform is larger, it is the number of rays (4096 in our example) times the number of pixels $128^2 = 16384$. The Radon transform was calculated directly as suggested in [12]. Although the matrix for the Radon transform is large, the Radon transform is extremely sparse (0.2% fill) and can be easily stored. Notice also that the spatial pixels in our example are half of the bin size. Thus 128 spatial pixels are covered with 64 bins.

In order to calculate the product $G^T u$ we observe that:

$$G^T u = (R_{xy} L_\lambda)^T u = L_\lambda^T R_{xy}^T u \qquad (13)$$

C. Applying Non-negativity Constraints

The least squares solutions does not restrict the sign of the solution, and hence our recovered model could have negative values. This is not physical and we would prefer only non-negative solutions. The method proposed in [3] is to use non-negative least square (NNLS) which is a direct method to overcome this problem. Our problem is too large to apply the NNLS, so we enforce non-negativity by using the conjugate gradient least squares (NNCGLS) with constraints as was suggested in [13]. Since the algorithm is iterative, again it requires the calculation of Gv and $G^T u$. For a full review of the algorithm we refer to [13]. Other possibilities such as the EM-ML [14] are possible but are probably more computationally burdensome since they minimize a nonlinear function.

IV. Numerical Experiments

In order to demonstrate the efficiency of our method we perform a numerical experiment. We choose a model made of two main spatial regions. Each region decays differently in time. The decay is made by combining two exponentials for the first spatial region and two exponentials for the second spatial region. The dynamic parameters of our experiment can be written as:

$$C(x, y)(a_1 e^{-\lambda_1 t} + a_2 e^{-\lambda_2 t})$$

with $a_1 = 1, a_2 = 0.7, \lambda_1 = 1$ minute^{-1} and $\lambda_2 = 5$ minutes^{-1} for the first region and $a_1 = 0.95, a_2 = 0.5, \lambda_1 = 3$ minutes^{-1} and $\lambda_2 = 10$ minutes^{-1} for the second region, and $C(x, y)$ sets amplitudes for each x and y at the grid. Tracer distribution after 2, 20 and 50 minutes is plotted in the top panel of Figure (2), and the time activity of each of the regions is plotted in Figure (3). The size of the model is 40×40 cm and it is discretized into 128×128 spatial pixels. Data were collected from 64 angles from $0 - 180^o$ with 64 bins in each angle. Each bin was 0.625 cm wide. Each angle was collected at a different time. The data collected were equally spaced from 0 to 10 minutes, but using equation (3) we can predict activity levels at time later than 10 minutes.

We preform 2 experiments, with different noise levels. In the first experiment no noise was added to the data. In the second experiment, random Poisson noise whose amplitude is relative to the square root of each simulated datum was added. In the next stage we reconstruct the model by solving (9), with LSQR and with nonnegative conjugate gradient least squares. The L-curve was used as a stopping criteria for both types of solutions.

The results are viewed using two methods. First we transform our results into the time domain and display them as a movie, where every frame is the image at a different fixed time i.e. we can look at $a(x, y, t_{fixed})$ for some fixed t. Three of these frames are shown in Figure 2 (bottom panel). Another way to view the results is to define a region of interest and to look at the activity curves and their spectral content, i.e. $a(x_{fix}, y_{fix}, t)$ and $m(x_{fix}, y_{fix}, \lambda)$. This is done in Figures 3-4. In order to compare the results of the experiments and the true result, we look at the following factors:

- 1. Relative difference between the real and reconstructed models in space at some specific time:

$$||\Delta a(t_j)|| = \frac{||a^{true}(x, y, t_j) - a^{recon}(x, y, t_j)||}{||a^{true}(x, y, t_j)||}$$

- 2. Relative difference between the dynamic parameters of the real and reconstructed models:

$$||\Delta m|| = \frac{||m^{true}(x, y, \lambda) - m^{recon}(x, y, \lambda)||}{||m^{true}(x, y, \lambda)||}$$

- 3. The relative difference between the measured data and predicted data

$$RMS = \frac{||Gm - d||}{||d||}$$

In Tables 1-3 we evaluate these parameters for the two experiments with and without noise. We also tabulate the difference between the LSQR (no positivity) and the NNCGLS (positivity constraint).

Table 1
$||\Delta m||$ for different errors in the data

Data Errors	$\|\|\Delta m\|\|_{LSQR}$	$\|\|\Delta m\|\|_{NNCGLS}$
Exact Data	1.3e-4	7.3e-6
Poisson noise	5.3e-1	1.4e-2

Table 2
$||\Delta a||$ for different errors in the data at different times

Data Errors	Time	$\|\|\Delta a\|\|_{LSQR}$	$\|\|\Delta a\|\|_{NNCGLS}$
Exact Data	2min	3.2e-6	5.1e-6
Exact Data	20min	6.9e-6	8.1e-5
Exact Data	50min	5.5e-5	4.2e-4
Poisson noise	2min	1.1e-2	1.4e-2
Poisson noise	20min	4.3e-2	5.9e-2
Poisson noise	50min	1.6e-1	2.3e-1

Table 3
Results of RMS for different errors in the data

Data Errors	RMS_{LSQR}	RMS_{NNCGLS}
Exact Data	7.9e-9	6.8e-7
Poisson noise	2.8e-1	2.7e-1

Table 4
Floating Points Operation for the Reconstructions

Data Errors	RMS_{LSQR}	RMS_{NNCGLS}
Exact Data	9.5e11	3.5e13
Poisson noise	2.7e9	1.6e11

The results agree well when we work with exact data. The spatial and the spectral content of the model is recovered. The noisy data produces less accurate results, however, the reconstruction still yields acceptable results (which can be seen in the Figures 2-3) and the parameters are reconstructed with reasonable agreement .

Differences in fitting the data are summarized in Table 3. When the data have noise in it we actually do not want to fit it exactly, because a perfect fit means we are fitting to the noise as well.

The floating points operations (flops) for the reconstructions is given in Table 4. Since exact data needs to be fitted more accurately, more iterations are needed to complete the reconstruction and more flops are used. When the data contains errors, we truncate after a few iterations (for this reconstruction 34 iterations are needed). A discussion about the number of iterations needed for the reconstruction is in [7].

V. Experiments with a Simple Phantom

In order to demonstrate the efficiency of the method we tested it on a simple phantom. The phantom is made from a small bottle with fluid in it. The fluid has some initial activity in it and it washes out with a washout time of 2 minutes. The experimental setting and the method of measuring washout time are described in [15]. The goal of our experiment was to recover this washout activity. The data were collected at 64 angles and each angle has 64 bins. The model space is 40cm×40cm and it is discretized into 128×128 pixels.

We used the same technique as in the synthetic example, and the same type of weighting in the spatial-spectral domain. The RMS error for the reconstruction chosen by the L-curve technique was 0.136. The time-curve for the activity is in Figure 5 and the spectral content of that curve is shown in Figure 6. We also plot in Figures 5 and 6 a curve which corresponds to a 2 minute washout parameter. The figures show that our method reconstructs the 2 minute washout quite accurately.

VI. Conclusion

We have found an effective way to reconstruct dynamic SPECT images directly from the projections. Experiments on both synthetic and real data were conducted and show that our method is stable and robust. The linearity of the formulation allows us to obtain the solution in a computationally efficient way, with very few assumptions about the time behaviour.

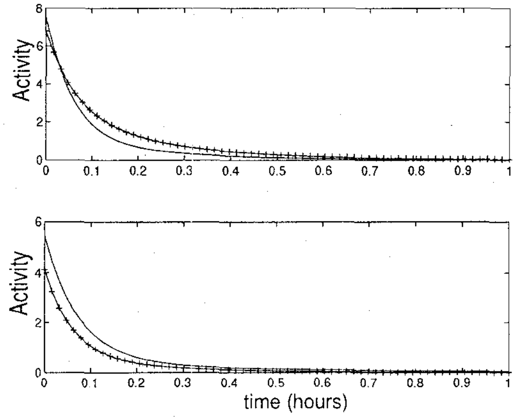

Fig. 3 Time activity for two main areas. + is the reconstructed activity and the continuous line is the true activity. The top curve corresponds to the activity on the top left image (of Figure 2) and the bottom corresponds to the bottom right of Figure 2. The top curve corresponds to: $a(13, 13, t)$ i.e, the pixel which is at location $(x = 24, y = 24)$ and the bottom curve corresponds to $a(30, 30, t)$, i.e. the pixel at $(x = 24, y = 24)$

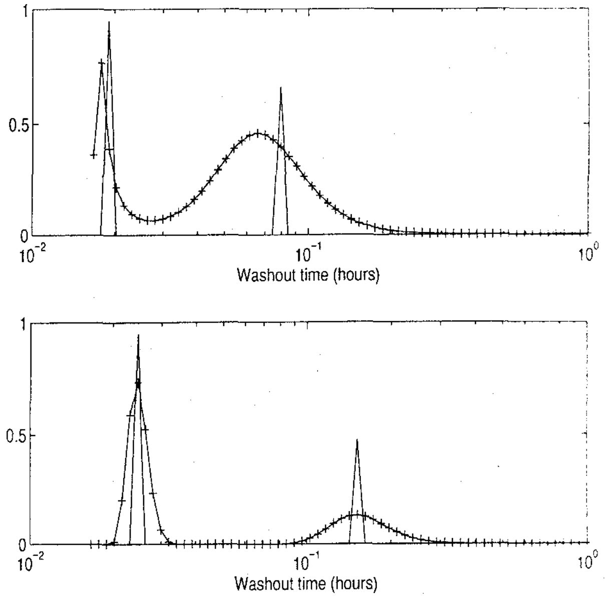

Fig. 4 Reconstructed $m(x_{fix}, y_{fix}, \lambda)$ for the same two pixels that are described in Figure 3. The + line is the reconstructed and the continuous line is the actual $m(x_{fix}, y_{fix}, \lambda)$.

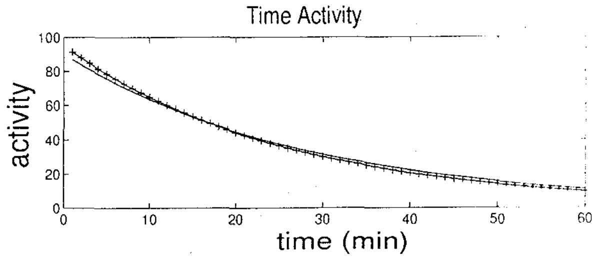

Fig. 5 The calculated washout curve (+) and the expected curve.

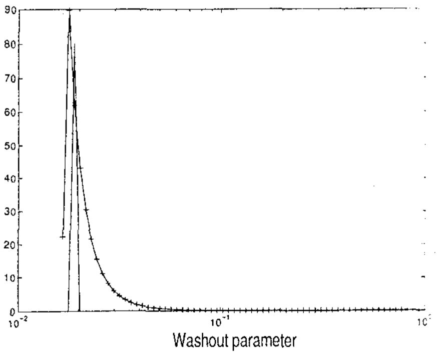

Fig. 6 Reconstructed $m(x_{fix}, y_{fix}, \lambda)$ for the phantom (with +) and a 2 minute decay parameter (spike) which was measured in the experiment.

VII. References

[1] A. M. Smith and G. T. Gullberg, Dynamic Cardiac SPECT

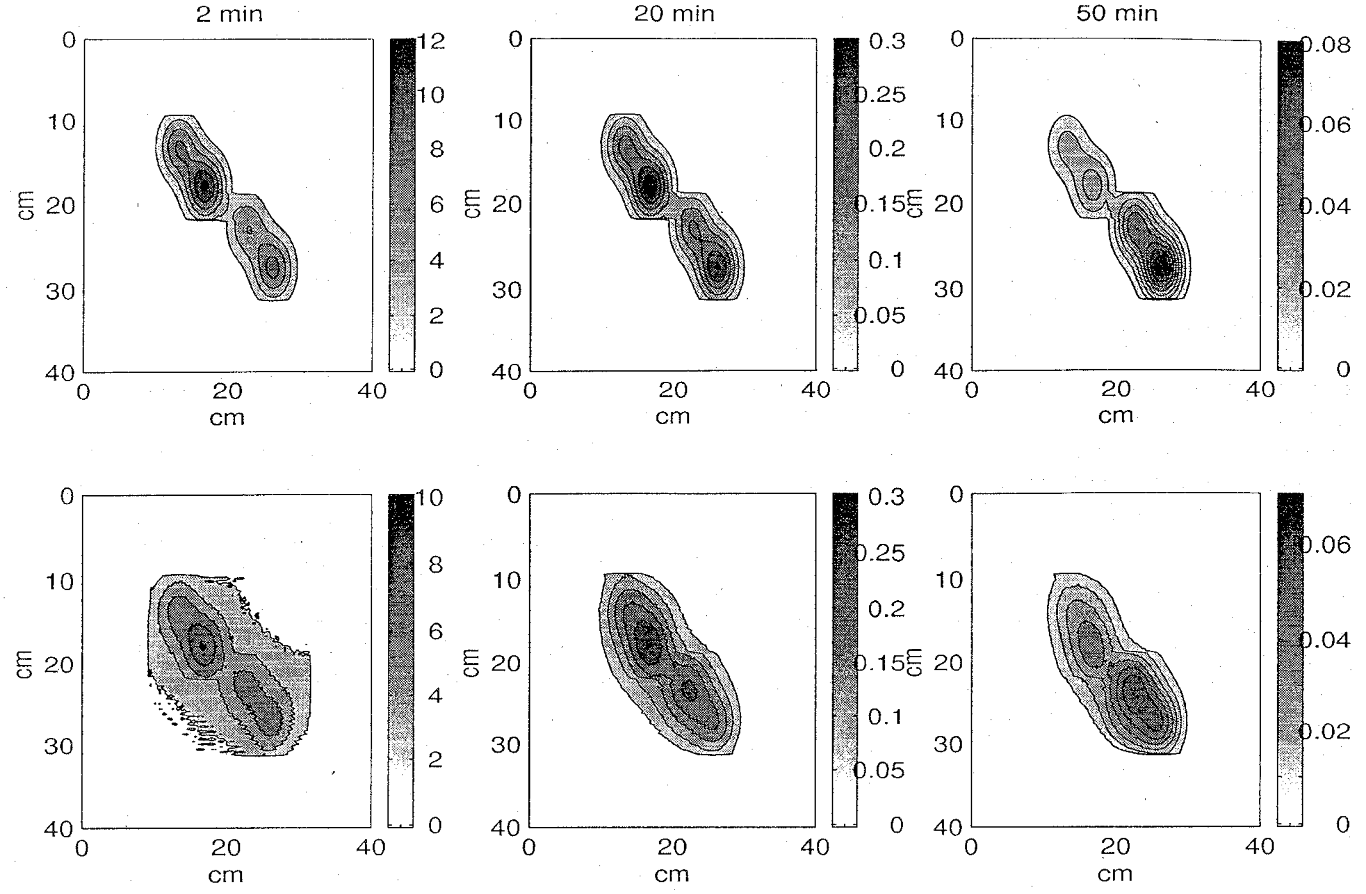

Fig. 2 The real models (top) and reconstructed models (bottom) after 2, 20 and 50 minutes. The reconstruction is done with the NNCGLS, the data were contaminated with Poisson noise. Notice that the color-bar of each image was chosen to view the image in an optimal way.

Computer Simulation for Teboroxime Kinetics *IEEE Trans Nucl Sci, 41, 1994* pp. 1626-1633

[2] M. N. Limber, M. A. Limber, A Celler, J. S. Barney, J. M. Borwein. Direct Reconstruction of Functional Parameters for Dynamic SPECT. *IEEE Trans Nucl Sci, 42, 1995* pp. 1249-1256.

[3] K.P. Whittall, and A.L. MacKay, Quantitative Interpretation of NMR Relaxation Data. *J. Magn. Reson. 84, 1989* pp. 134-152.

[4] V. J. Cunningham, T. Jones, Spectral Analysis of Dynamic PET Studies. *J. Cerebral bloodflow and metabolism, v13, n1, 1993* pp. 15-23

[5] D. Porter, and D. Stirling, "Integral Equations" *Cambridge University Press, 1990*

[6] A. N. Tichonov, Solution of incorrectly formulated problems and the regularization method, *Soviet Math. Dokl. 4, 1963, 1035-1038. English translation of Dokl. Akad. Nauk. USSR 151, 1963* pp. 501-504

[7] P. C. Hansen, Rank Deficient and Discrete Ill-Posed Problems. *Ph.D. thesis, UNIC-95-07 Denmark*

[8] G. Wahba, "Spline Models For Observational Data" *SIAM Press, 1990*

[9] G. Nolet, Solving or resolving inadequate and noisy tomographic system *J. Comp. Phys. 61, 1985* pp. 463-482

[10] C. C. Paige and M. A. Saunders. LSQR: an algorithm for sparse linear equations and least squares, *ACM Trans. Math Software 8, 1982* pp. 195-209

[11] A. Bjork, Stability of Conjugate Gradient Type Methods for Linear Least Square Problems *Linkoping University S-581 83 LiTH-MAT-R-1995-26* pp. 1-35

[12] M. Smith, C. E. Floyd, R. J. Jaszczak, Reconstruction of SPECT Images Using Generalized Matrix Inverses *IEEE Trans on Medical Imaging 1992* pp. 165-175

[13] M. Hestenes, "Conjugate Direction Methods in Optimization" *Springer-Verlag New York Inc, 1980*

[14] K. Lange and R. Carson, Reconstruction algorithms for emission and transmission tomography *J. Comput. Assist. Tomography., 8 1984* pp. 306-316

[15] A. Celler, T. Francombe, R. Harrop, D. Lister Dynamic Heart-in-Thorax Phantom For Functional SPECT. *In: IEEE 1996 Nuclear Science Symposium, Conference Record.*, pp. 1893-1897.