1. Brain Mapping by Positron Emission Tomography

Peter Toft, Peter Alshede Philipsen, Lars Kai Hansen, Ulrik Kjems

Department of Mathematical Modelling, Technical University of Denmark, Building 305, DK-2800 Lyngby, Denmark, Email pto,pap,lkh,uk@imm.dtu.dk.

Abstract

In this paper the basis of PET (Positron Emission Tomography) is reviewed, and it is shown that the measured signals can be modelled as the Radon transform of the desired spatial distribution of, e.g., the brain activity.

Next, two of the direct reconstruction methods are presented. Both are derived from inversion of the Radon transform. It is shown that the reconstruction can be based on filtering and integration techniques.

Another major class of reconstruction techniques is presented, namely the linear algebra based methods, which often are formed as iterative methods. A very fast way of implementing a set of iterative reconstruction techniques is shown along with a set of examples.

1.1 Introduction to PET

The PET scanner is based on radioactive tracers. A small dosage of a β^+ emitter, such as O-15 or C-11, is injected into the patient (or the object to be scanned). The β^+ emitter will be distributed in the tissue due to the blood circulation. If for instance the brain is to be scanned then the β^+ emitter will decay in correspondence to the activity of the brain. In the regions with high brain activity the blood flow will be high, and many of the radioactive nucleons will decay under emission of a positron in these regions.

Within a few mm the positron will interact with an electron and annihilate under emission of two 511 keV photons. The two photons will travel in opposite directions outward to the PET scanner where the photons are detected at sensors placed in a ring, as illustrated on Fig. 1.1.

The PET hardware will sort the arrival times of the photons, so only two photons that arrived at the same time at the sensor ring are taken into account (The photons travel with the speed of light, and here the small difference in arrival time is neglected). A two dimensional matrix of the possible line parameters is created, and here denoted by $E(\rho, \theta)$. If the two photons form a line with parameters $(\rho, \theta) = (\rho_0, \theta_0)$, the array is incremented with the value one at that position in the array. This is because the only obtainable

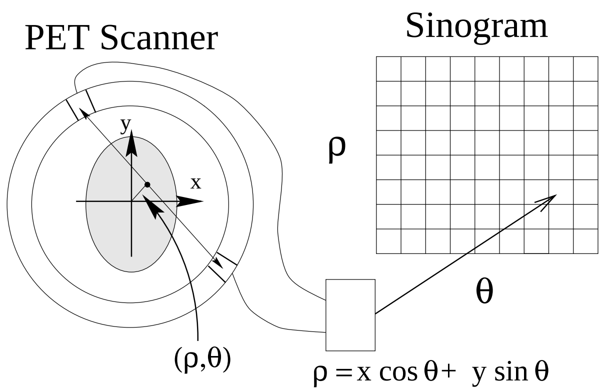

Figure 1.1: Emission of two photons from the place of decay.

information from the two photons is that the photon emission took place somewhere along that line. Depending on the radioactive dosage given to the patient many decays take place per second in each volume element. After the recording is terminated, an array of emissions has been recorded. For $(\rho, \theta) = (\rho_0, \theta_0)$ the array is developed by several β^+ emissions which took place along the line. The individual emission recordings are no longer stored. The obtained array of emissions (called the *sinogram*) are in each position approximately proportional to the emission intensity along that particular line (times the total recording time T). The obtained matrix $E(\rho, \theta)$ divided by T is in mathematical terms an integral of the emission intensity $g(x, y)$ over that particular line.

$$\frac{E(\rho, \theta)}{T} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x, y) \ \delta(\rho - x\cos\theta - y\sin\theta) \ dx \ dy \ T = \breve{g}(\rho, \theta) \tag{1.1}$$

It can be recognized that the recorded array $E(\rho, \theta)$ divided by T is the Radon transform of the emission intensity $g(x, y)$, here denoted by $\breve{g}(\rho, \theta)$ [1],[2]. Note that this intensity will not be a constant in time as it varies with the activity of the brain. This will limit the allowable recording time.

Previously absorption of the photons in the tissue has been neglected, thus the inversion process must be modified to take this into account. In principle, this is easy when using double photon emissions. It can be shown, e.g. [2], that it only requires that the sinogram of emissions must be multiplied in each sample (pixel) with a sinogram corresponding to the transmission. This implies that reconstruction of PET images can be implemented as an inverse Radon transform of a sinogram.

1.2 Direct Reconstruction Methods

1.2.1 The Fourier Slice Theorem

Inversion of the Radon transform can be done in several ways. One standard algorithm is based on the Fourier Slice Theorem also known as the Central Slice Theorem [3],[4]. The Radon transform $\breve{g}(\rho, \theta)$ is to be inverted into $g(x, y)$, where the inversion is based on the Fourier transform.

First the 2D Fourier Transform of $g(x, y)$ is needed.

$$G(k_x, k_y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x, y)\ e^{-j2\pi(k_x x + k_y y)}\ dx\ dy \tag{1.2}$$

Introducing polar frequency parameters

$$\begin{pmatrix} k_x \\ k_y \end{pmatrix} = \nu \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix} \tag{1.3}$$

and inserting in Eq. (1.3) into Eq. (1.2) gives

$$G(\nu\cos\theta, \nu\sin\theta) = \int_{-\infty}^{\infty} \breve{g}(\rho, \theta)\ e^{-j2\pi\rho\nu}\ d\rho \tag{1.4}$$

Thus, a 1D Fourier Transform of the Radon transform gives the 2D spectrum of $g(x, y)$, hence the Fourier Slice Theorem makes it possible to invert the Radon transform by use of 1D Fourier transforms and 2D Fourier transforms. Note, that the theorem can be used in reverse order to calculate the Radon transform, if the signal $g(x, y)$ is given.

1.2.2 Filtered Backprojection

Another very famous inversion scheme is the Filtered Backprojection [5],[6] or as it more correctly should have been named Backprojection of Filtered Projections method [1]. It is derived from the inverse of Eq. (1.2), by introducing polar coordinates

$$\begin{aligned} g(x, y) &= \int_0^{2\pi} \int_0^{\infty} \nu\ G(\nu\cos\theta, \nu\sin\theta)\ e^{j2\pi\nu(x\cos\theta + y\sin\theta)}\ d\nu\ d\theta \\ &= \int_0^{\pi} \int_{-\infty}^{\infty} |\nu|\ G(\nu\cos\theta, \nu\sin\theta)\ e^{j2\pi\nu(x\cos\theta + y\sin\theta)}\ d\nu\ d\theta \end{aligned} \tag{1.5}$$

and inserting Eq. (1.4) and splitting the formula into two parts, then Filtered Backprojection is written in two parts: A filtering part and an integration part.

$$\bar{\breve{g}}(\rho, \theta) = \int_{-\infty}^{\infty} |\nu| \left(\int_{-\infty}^{\infty} \breve{g}(\tilde{\rho}, \theta)\ e^{-j2\pi\tilde{\rho}\nu}\ d\tilde{\rho} \right)\ e^{j2\pi\rho\nu}\ d\nu \tag{1.6}$$

$$g(x, y) = \int_0^{\pi} \bar{\breve{g}}(x\cos\theta + y\sin\theta, \theta)\ d\theta = \int_0^{\pi} \int_{-\infty}^{\infty} \bar{\breve{g}}(\rho, \theta)\ \delta(\rho - x\cos\theta - y\sin\theta)\ d\rho\ d\theta \tag{1.7}$$

Note, the similarity of Eq. (1.7) to the Radon transform in Eq. (1.1). The operation shown in Eq. (1.7) is named backprojection and it can be shown (page 120 of [1]) that the backprojection operator is closely linked to the adjoint Radon transform.

The filtering part in Eq. (1.6) is a forward Fourier transform for each of the angles θ. In the Fourier domain the signal is high pass filtered with the filter $|\nu|$. Subsequently an inverse 1D-Fourier transform is used to get the filtered Radon domain $\bar{\breve{g}}(\rho, \theta)$.

The remaining backprojection part is merely an integration along a sine-curve in the filtered Radon domain. Eqs. (1.6) and (1.7) form the Filtered Backprojection inversion scheme. A movie showing the Filtered Backprojection method can be found at the URL `http://hendrix.imm.dtu.dk` under the movies entry.

1.3 Reconstruction Algorithms Based on Linear Algebra

Inversion of the Radon transform need not be based on the direct inverse formulas as shown in Section 1.2. A different class of reconstruction schemes is based on linear algebra [7],[1],[8].

The (discrete) Radon transform is a linear transform, with respect to the function $g(x, y)$, hence instead of considering the integral version of the Radon transform a matrix representation can used

$$\begin{array}{ccccc} \breve{g}(\rho,\theta) & = & \mathcal{R} & g(x,y) \\ \downarrow & & \downarrow & \downarrow \\ \boldsymbol{b} & = & \boldsymbol{A} & \boldsymbol{x} \end{array} \tag{1.8}$$

In the last 15 years the iterative reconstruction methods have gained much attention in the literature [7],[8]. Several methods have been very prominent, such as EM (Expectation Maximization) [5],[9],[10],[11], ART (Algebraic Reconstruction Technique) [12],[7],[13], and LSCG (Least Squares Conjugate Gradient) [14],[15].

These methods formulate the reconstruction problem as a linear set of equations

$$\boldsymbol{b} = \boldsymbol{A}\boldsymbol{x} \;\Leftrightarrow\; b_i = \sum_{j=1}^{J-1} a_{i,j} x_j, \quad i = 1, 2, \ldots, I \tag{1.9}$$

where $\boldsymbol{b}$ is an I-dimensional vector containing the known sinogram values wrapped into a vector, and $\boldsymbol{x}$ is a J-dimensional vector containing the unknown image to be reconstructed. Here $\boldsymbol{A}$ is the system matrix, which contains the weight factors between each of the image pixels and each of the values in the sinogram, corresponding to line orientations. The use of linear algebra has several advantages, such as easier incorporation of irregular geometries. The system matrix can model several real-world properties, such as finite, i.e., non-zero detector size and varying detector sensitivity. Furthermore, regularization can easily be incorporated [16],[14] in order to affect the often ill-conditioned reconstruction problem.

One problem is the huge size of the system matrix. A 2D sinogram from, e.g., a GE Advance PET scanner contains $I = 281 * 336$ values, and reconstructed into a $J = 301 * 301$ grid, i.e., the system matrix has approximately 8.5 billion elements, requiring over 34 GBytes of memory when using 4 bytes per matrix element. This is a large amount of memory, even looking some years into the future. Besides this aspect it would not be wise to store all that data, due to the fact that approximately 98% of the matrix entries will be zeros. This knowledge should be incorporated into the reconstruction schemes.

Assuming that memory is not available for storing the full system matrix, one possibility is to compute the individual matrix elements in each iteration when needed. This can be done by using the Radon transform, e.g., [2] or other modelling schemes for the scanner. This approach is rather easily implemented and is viable and storage requirements are reduced to a minimum, only requiring memory for the sinogram ($\boldsymbol{b}$) and the current solution ($\boldsymbol{x}$), and perhaps some additional temporary variables of the same size or smaller, but no system matrix is stored in memory. It will be demonstrated that this implementation has a major drawback in speed, since the system matrix will be computed many times during an iterative reconstruction. Each time at the same high computational cost.

1.4 Accelerated 2D Iterative Reconstruction

In this section a hybrid solution is investigated for accelerating the iterative reconstruction algorithms [17],[11],[18], but requiring as much memory as modern workstations are currently equipped with (or will be soon). The idea is to store the non-zero elements of the system matrix in the main memory using sparse matrix techniques. In this way the core of the reconstruction algorithms, highly based on matrix vector multiplications, can be accelerated significantly, and thereby solve one of the major drawbacks of the iterative methods.

It is proposed that the system matrix $\boldsymbol{A}$ is calculated one time only using all the modifications found for the actual scanner setup. If no specific scanner model is provided the system matrix can be modelled and generated using the Radon transform or other simpler schemes. From the system matrix the very small values in the matrix are truncated to zero. Let

$$\tilde{a}_{i,j} = \begin{cases} a_{i,j} & \text{if } a_{i,j} > \gamma \\ 0 & \text{Otherwise} \end{cases} \tag{1.10}$$

where the threshold γ can be chosen to a certain fraction of the maximum matrix value, e.g., $\gamma = 0.05 \max_{i,j}\{a_{i,j}\}$. If γ is chosen sufficiently low, a good compromise between resolution and the sparseness of the matrix can be reached, and normally this does not affect the behaviour of the algorithms. Current work concerns the quantification of the truncation error.

The sparse structure of $\boldsymbol{A}$ can be exploited by storing only non-zero values in the fast memory. For a certain row, number i, all of the matrix elements are calculated, stored, and truncated using Eq. (1.10). Hereby the number of non-zero elements in the row, denoted by Z_i, will be much smaller than the image size $J = M^2$. The values of Z_i are stored in a simple, one dimensional vector. Two vectors of length Z_i, indexed by an integer z, can then be allocated and stored containing the non-zero matrix value a_z and the corresponding column index j_z. The procedure is repeated for all rows.

Assuming a nearest neighbour approximation with one pixel for each point along the integration lines and using 4 bytes for storing each of the vector elements, the total storage requirement is then reduced to approximately $8\sum_{i=1}^{I} z_i \approx 8IM$ bytes. In the example shown above approximately 100 MBytes memory is required. Assuming this amount of memory is present most iterative algorithms can be implemented very fast from three basic operations: Matrix vector multiplication (forward projection) $\boldsymbol{A}\tilde{\boldsymbol{x}}$, scalar product between the i'th row of the system matrix and a vector $\boldsymbol{a}_i^T\tilde{\boldsymbol{x}}$, and finally multiplication with the transpose of the matrix (backprojection) $\boldsymbol{A}^T\tilde{\boldsymbol{b}}$.

A software package has been written in C including the proper structures for manipulating sparse matrices and vectors, along with an optimized code for computing matrix vector products, well suited for iterative reconstruction algorithms. In the package ART, EM, and LSCG are implemented both in a fast version using sparse matrix storage of the system matrix and in a slow version where the system matrix is not stored and needed matrix entries are computed as they are needed in each step of the iterative algorithms. The software package is available for free at `http://eivind.imm.dtu.dk/staff/ptoft`, but protected by the GNU General Public License.

Algebraic Reconstruction Technique (ART)

For a certain row i of the matrix (i depends on the iteration number k), the general iteration step incrementing the current solution $\boldsymbol{x}^{(k)}$ can be found in, e.g., [7]

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \frac{b_i - \boldsymbol{a}_i^T \boldsymbol{x}^{(k)}}{\boldsymbol{a}_i^T \boldsymbol{a}_i} \boldsymbol{a}_i \tag{1.11}$$

Expectation Maximization (EM)

The general iteration step of EM [10],[19] requires a forward projection, a backprojection, and two fast updates in each iteration.

$$x_j^{(k)} = \frac{x_j^{(k-1)}}{\sum_{i'=1}^{I} a_{i',j}} \sum_{i=1}^{I} \frac{a_{i,j} b_i}{\sum_{j'=1}^{J} a_{i,j'} x_j^{(k-1)}} \tag{1.12}$$

Least Squares Conjugate Gradient (LSCG)

The Least Squares Conjugate Gradient method requires some initialization [15]

$$\boldsymbol{s}^{(0)} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^{(0)} \tag{1.13}$$

$$\boldsymbol{r}^{(0)} = \boldsymbol{p}^{(0)} = \boldsymbol{A}^T \boldsymbol{s}^{(0)} \tag{1.14}$$

$$\boldsymbol{q}^{(0)} = \boldsymbol{A}\boldsymbol{p}^{(0)} \tag{1.15}$$

Then for each iteration the LSCG algorithm on the normal equations becomes

$$\alpha = \frac{\left(\boldsymbol{r}^{(k-1)}\right)^T \boldsymbol{r}^{(k-1)}}{\left(\boldsymbol{q}^{(k-1)}\right)^T \boldsymbol{q}^{(k-1)}} \tag{1.16}$$

$$\boldsymbol{x}^{(k)} = \boldsymbol{x}^{(k-1)} + \alpha \boldsymbol{p}^{(k-1)} \tag{1.17}$$

$$\boldsymbol{r}^{(k)} = \boldsymbol{A}^T \boldsymbol{s}^{(k-1)} \tag{1.18}$$

$$\boldsymbol{r}^{(k)} = \boldsymbol{r}^{(k-1)} - \alpha \boldsymbol{q}^{(k-1)} \tag{1.19}$$

$$\beta = \frac{\left(\boldsymbol{r}^{(k)}\right)^T \boldsymbol{r}^{(k)}}{\left(\boldsymbol{r}^{(k-1)}\right)^T \boldsymbol{r}^{(k-1)}} \tag{1.20}$$

$$\boldsymbol{p}^{(k)} = \boldsymbol{r}^{(k)} + \beta \boldsymbol{p}^{(k-1)} \tag{1.21}$$

$$\boldsymbol{q}^{(k)} = \boldsymbol{A}\boldsymbol{p}^{(k)} \tag{1.22}$$

For all three methods an initial value of the solution, i.e., $\boldsymbol{x}^{(0)}$ is needed. In the package an image found by, e.g., a fast direct method, can be supplied and used. If not provided, all of the initial values of the vector are initialized to a properly chosen constant.

1.5 Examples

The program has been used on two types of machines. A Linux machine with a 120 MHz Pentium processor and an Onyx from SGI equipped with four 200 MHz R4400 processors, where the program was running on one processor.

1.5.1 Reconstruction of a Small Sinogram

In the first example the (synthetic) sinogram has $125 * 101$ samples and the reconstructed image has $101 * 101$ samples. In Table 1.1 the reconstruction times on both machines are shown for the fast and the slow method as well as the ratio between the execution times (slow/fast).

Times are measured for ART, EM, and the LSCG-method, when EM and LSCG were running (arbitrarily) 20 iterations, and ART 20 full iterations, i.e., 20 times the number of rows (chosen randomly), which is 20$*$125$*$101 iterations in Eq. (1.11). Note that all times only correspond to the actual iterations. For the fast versions of the iterative reconstruction algorithms, the time to generate the system matrix once should be added if changing the system matrix, e.g., when changing the sampling parameters of the reconstructed image.

For this example the system matrix was modelled using discrete Radon transformation with linear interpolation, where the threshold γ was chosen to zero, hence the slow and the fast methods give exactly the same results. Note that the large difference in speedup between ART and EM/LSCG is due to the implementation of the forward projection is more efficient than the multiplication with the transpose of the system matrix. The slow methods can be accelerated some by implementing multiplication with the transpose of the system matrix (adjoint operator) as a backprojection integral, but note that this implies that the approximation of the system matrix will be different in the forward and the backprojection part. The sparse system matrix for this transformation geometry required approximately 13 MBytes, and each iteration requires approximately one second.

Machine	Type	ART	EM	LSCG
Pentium	Fast	26 sec	17 sec	17 sec
	Slow	2218 sec	5776 sec	5250 sec
	Ratio	85	340	309
SGI Onyx	Fast	16 sec	16 sec	16 sec
	Slow	1306 sec	2722 sec	2715 sec
	Ratio	82	170	170

Table 1.1: Time usage for 20 iterations of EM and LSCG. For ART the time is for 20 full iterations, i.e., 20$*$125$*$101 of the iterations used in Eq. (1.11). The time measurements are for a sinogram with $125 * 101$ samples reconstructed into a $101 * 101$ samples image.

1.5.2 Reconstruction of a Large Sinogram

In Fig. 1.2 a 2D sinogram with 281∗336 samples is shown, which was measured on a GE Advance PET scanner. The sinogram is reconstructed into a (large) image with 301∗301 samples. The system matrix has 94416∗90601 elements of which 0.23% are non-zero when modelling then system matrix using the Radon transform of a square. On the Onyx it required 1466 seconds to generate the 157 MBytes sparse matrix, and each iteration of EM required 15 seconds in the fast version and 11678 sec in the slow implementation. After 10 iterations the algorithm was stopped. The reconstructed image is shown in Fig. 1.3.

For sake of comparison, the same sinogram reconstructed image using Filtered Back-projection is shown in Fig. 1.4, and this reconstruction used 6 seconds on the Onyx. Next the LSCG method is used to reconstruct the sinogram shown in Fig. 1.2. The reconstructed image after 11 iterations is shown in Fig. 1.5 (Note that this particular image has only $101 * 101$ pixels). It can be seen that some low frequency oscillations are emerging from the edges. These can be controlled using regularization or constraints [2].

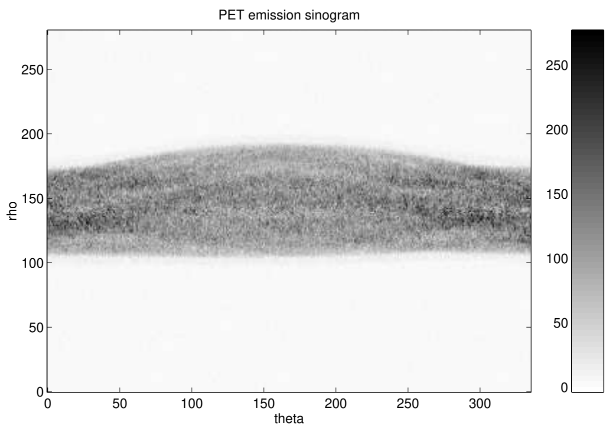

Figure 1.2: A PET sinogram of a human brain.

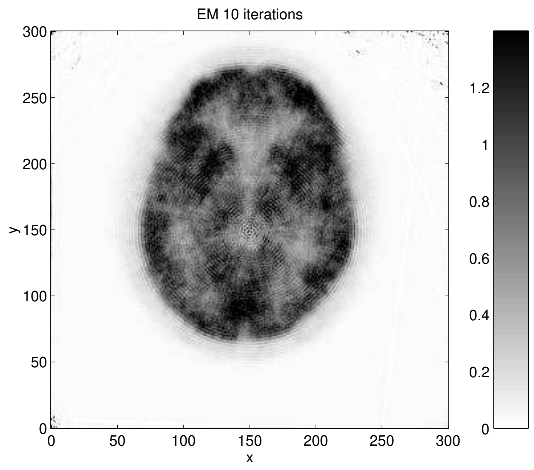

Figure 1.3: The reconstructed image after 10 iterations of EM.

1.6 Conclusion

We have described the basis of PET reconstruction techniques and demonstrated a very fast implementation of iterative reconstruction comparable in speed with direct reconstruction methods. The implementation is based on storing of the system matrix in fast memory using sparse techniques. The approach is mainly applicable to 2D reconstruction, due to the requirements of a sufficient amount of memory, but in principle the method can also be applied to 3D reconstruction.

The cost is that a large amount of memory is required, but for ART we have demonstrated a speedup factor of approximately 80 and for EM and LSCG 170-340 depending on the machine, for a fixed transformation geometry and interpolation level.

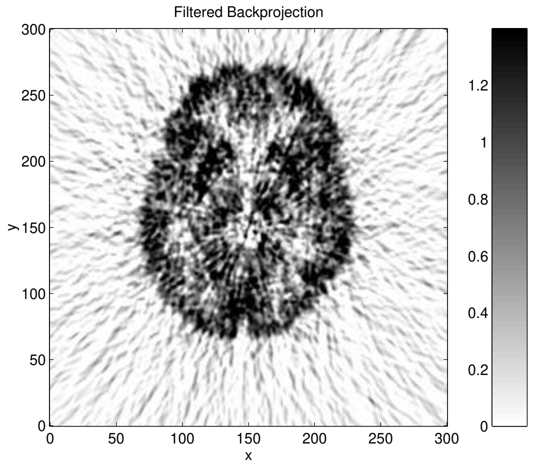

Figure 1.4: The reconstructed image using Filtered Backprojection with a ramp filter.

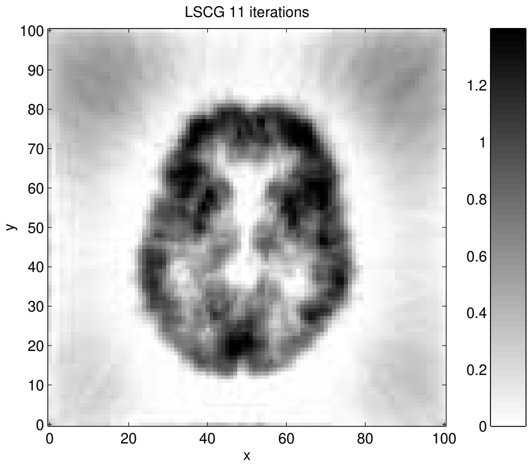

Figure 1.5: The reconstructed image using 11 iterations of the Least Squares Conjugate method. The artifacts around the brain can be removed by proper masking.

Acknowledgment

This research is supported by the Danish Research Councils for the Natural and Technical Sciences through the Danish Computational Neural Network Center.

Bibliography

[1] S. R. Deans. *The Radon Transform and Some of Its Applications.* Krieger Publishing Company, Malabar, Florida, 2 edition, 1993. Previously John Wiley & Sons Inc., 1983.

[2] Peter Toft. *The Radon Transform - Theory and Implementation.* PhD thesis, Department of Mathematical Modelling, Technical University of Denmark, 1996.

[3] R. M. Mersereau. Recovering Multidimensional Signals from their Projections. *Computer Graphics and Image Processing*, 1(4):179–195, Dec. 1973.

[4] Dan E. Dudgeon and Russel M. Mersereau. *Multidimensional Digital Signal Processing.* Prentice-Hall, Englewood Cliffs,N.J, 1984.

[5] L. A. Shepp and J. B. Krustal. Computerized tomography: The new medical x-ray technology. *Am. Math. Monthly*, 85:420–439, April 1978.

[6] A. Rosenfeld and A. C. Kak. *Digital Picture Processing*, volume 1. Academic Press, 2. ed. edition, 1982.

[7] Yair Censor. Finite Series-Expansion Reconstruction Methods. In *Proc. of the IEEE*, volume 71, pages 409–419, March 1983.

[8] Julia K. Older and Paul C. Johns. Matrix formulation of computed tomography reconstruction. *Physics in Medicine & Biology*, 38:1051–1064, 1993.

[9] L. A. Shepp Y. Vardi and L. Kaufman. A Statistical Model for Positron Emission Tomography. *Journal of the American Statistical Association*, 80(389):8–37, March 1985. The pages include comments and discussion by several authors.

[10] R. E. Carson and K. Lange. The EM parametric image reconstruction algorithm. *Journal of the American Statistical Association*, 389(80):20–25, March 1985.

[11] Linda Kaufman. Implementing and Accelerating the EM Algorithms for Positron Emission Tomography. *IEEE Trans. Med. Imag.*, MI-6(1):37–51, march 1987.

[12] Gabor T. Herman and Lorraine B. Meyer. Algebraric Reconstruction Techniques Can Be Made Computationally Efficient. *IEEE Trans. Med. Imag.*, 12(3):600–609, Sep. 1993.

[13] Huaiqun Guan and Richard Gordon. A projection access order for speedy convergence of ART (algebraric reconstruction technique): a multilevel scheme for cumputed tomography. *Phys. Med. Biol.*, 39:2005–2022, 1994.

[14] Linda Kaufman and Arnold Neumaier. Image Reconstruction Through Regularization by Envelope Guided Conjugate Gradients. Technical report, Bell Labs, 1993. Report 11274-94-14.

[15] John A. Scales. Iterative methods for large, sparse, inverse calculations. Samizdat Press. Colorado School of Mines, 76 Olcott Drive, White River Junction, Vermont 05001, April 1993. Lecture notes.

[16] Per Christian Hansen. *Rank-Deficient and Discrete Ill-Posed Problems.* Polyteknisk Forlag, Lyngby, Denmark, 1996. Doctoral Dissertation.

[17] Tom R. Miller and Jerold W. Wallis. Fast maximum-likelihood reconstruction. *Journal of Nuclear Medicine*, 33(9):1710–1711, September 1992.

[18] Peter Toft and Jesper James Jensen. A Very Fast Implementation of 2D Iterative Reconstruction Algorithms. *Accepted for presentation at the 1996 IEEE Medical Imaging Conference*, 1996.

[19] G. T. Herman and D. Odhner. Performance evaluation of an iterative image reconstruction algorithm for positron emission tomography. *IEEE Trans. Med. Imag.*, 10:336–246, 1991.