A very fast Implementation of 2D Iterative Reconstruction Algorithms

Peter Toft† and Jesper James Jensen‡

† IMM, Technical University of Denmark, DK-2800 Lyngby, Denmark, Email ptoft@ei.dtu.dk.
‡ ØDS-Holding A/S, Kroghsgade 1, DK-2100 Copenhagen E, Denmark, Email jjj@oedan.dk.

1 Introduction

In the last 15 years the iterative reconstruction methods have gained much attention in the literature [1]. Several methods have been very prominent, such as EM (Expectation Maximization) [2, 3], ART (Algebraic Reconstruction Technique) [4], and LSCG (Least Squares Conjugate Gradient) [5]. Each of the methods formulates the reconstruction problem as a linear set of equations $\boldsymbol{b} = \boldsymbol{A}\boldsymbol{x} \Leftrightarrow b_i = \sum_{j=1}^{J-1} a_{i,j} x_j$, 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. The matrix $\boldsymbol{A}$ is the system matrix, which contains the weight factors between each of the image pixels and each of the sinogram values, corresponding to line orientations.

One problem is the huge size of the system matrix, often impossible to store in memory. Assuming that memory is not available for storing the full system matrix, then one possibility is to compute the individual matrix elements in each iteration when needed. This can be done by using of the discrete Radon transform or other modelling schemes of the scanner. This solution is rather easily implemented and is viable and storage requirements are reduced to a minimum, by 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. This implementation has a major drawback in speed, due to the many times the system matrix has to be computed during an iterative reconstruction. Each time at the same high cost.

2 Accelerated 2D Iterative Reconstruction

Here a hybrid solution is proposed for accelerating the reconstruction speed of the iterative methods and only 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 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 then 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 by using a very small threshold level.

The sparse structure of $\boldsymbol{A}$ can be exploited by only storing the non-zero values in the fast memory. For a certain row, number i, all of the matrix elements are calculated, stored, and truncated. 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 containing both the non-zero matrix value a_z and the corresponding column index j_z. Assuming that 4 bytes are required for storing each of the vector elements then the total storage requirement is reduced to approximately $8 \sum_{i=1}^{I} z_i \approx 8IM$ bytes, (assuming a nearest neighbour approximation with one pixel for each point along the integration lines). Assuming that a sufficient amount of memory is present most iterative algorithms can be implemented from three basic operations: Matrix vector multiplication $\boldsymbol{A}\tilde{\boldsymbol{x}}$, 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 $\boldsymbol{A}^T \tilde{\boldsymbol{b}}$.

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 have all been implemented 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 in each part of the iterative steps. The software package is provided for free, protected by the GNU General Public License.

Several interpolation methods have been implemented for generation of the system matrix: (1): Nearest Neighbour interpolation based on the Radon transform. (2): Linear interpolation based on the Radon transform. (3): Analytical Radon transform of a square, i.e., the length through a quadratic pixel is used. (4): The Radon transform of a sinc expansion in the image domain.

3 Results

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

As an example a sinogram with $125 * 101$ samples is reconstructed into an image with $101 * 101$ pixels. In Table 1 the reconstruction time is shown for the fast and the slow methods on the two machines.

Times are measured for ART, EM, and the LSCG-method, where 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. 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 parameters determining the system matrix, e.g., the sampling parameters of the sinogram or the reconstructed image.

For this example the system matrix was modelled using discrete Radon transformation with linear interpolation, where the threshold was chosen to zero, hence the slow and the fast methods give exactly the same results. In this example the sparse system matrix required approximately 13 MBytes memory.

Note that the large difference in speedup between ART and EM/LSCG is due to a very effective implementation of the forward projection compared with the backprojection. 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.

Machine	Fast ART	Slow ART	Fast EM	Slow EM	Fast LSCG	Slow LSCG
Pentium	26 sec	2218 sec	17 sec	5776 sec	17 sec	5250 sec
Onyx	16 sec	1306 sec	16 sec	2722 sec	16 sec	2715 sec

Table 1 Time usage for 20 iterations of EM and LSCG. For ART the time is for 20 full iterations, i.e., 20*125*101 iterations.

4 Conclusion

We have demonstrated a very fast implementation of iterative reconstruction based on storing the system matrix in fast memory by 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 of the proposed idea is that a large amount of memory is required, but for ART we have demonstrated a speedup factor of approximately 70 and for EM and LSCG a speedup factor larger than 175, between the slow and the fast implementation. In the example it was showed that each iteration of ART, EM, or LSCG requires approximately one second on most modern workstations.

References

[1] Yair Censor. Finite series-expansion reconstruction methods. In *Proc. of the IEEE*, volume 71, pages 409–419, March 1983.

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

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

[4] Gabor T. Herman and Lorraine B. Meyer. Algebraic reconstruction techniques can be made computationally efficient. *IEEE trans. on Med. Imag.*, 12(3):600–609, Sep. 1993.

[5] 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.