An object-oriented library incorporating efficient projection/backprojection operators for volume reconstruction in 3D PET

C. Labbé[1], K. Thielemans[2], H. Zaidi[1], C. Morel[1]
[1] Division of Nuclear Medicine, Geneva University Hospital, CH-1211 Geneva 4, Switzerland
[2] MRC Cyclotron Unit, Hammersmith Hospital, London, UK

1. Introduction

Iterative reconstruction methods applied to image reconstruction in three-dimensional (3D) positron emission tomography (PET) should result in possibly better images than analytical reconstruction algorithms. However, the long reconstruction time has remained an obstacle to their development and, moreover, their clinical routine use. Together with the constant increase in performances of the computing platforms, recent developments in parallel processing techniques offer practical ways to speed up the calculations and attain clinically viable processing rates [1].

The aim of this paper is to present the development of a library of object-oriented building blocks for the implementation of both analytical and iterative reconstruction algorithms in 3D PET.

2. An Object-oriented library for volume reconstruction in 3D PET

2.1 Reconstruction algorithms

Various analytical (exact and approximate) and iterative algorithms have been proposed for 3D reconstruction in PET. The reconstruction algorithms already implemented or under development using the object-oriented library for volume reconstruction include:

- The reprojection algorithm (PROMIS) [2].
- The Fourier rebinning algorithm (FORE) [3].
- The maximum likelihood by expectation maximisation (ML-EM) algorithm [4].
- The ordered subsets, expectation maximisation (OSEM) algorithm [5].
- The maximum a posteriori, expectation maximisation (MAP-EM) algorithm [6].
- The least squares (LSQ) algorithm [7] and variants of it including the image space reconstruction algorithm (ISRA) and ordered subsets ISRA (OSISRA) [8].
- The ordered subsets, Mirror (OS-MIRROR) algorithm [1].
- The algebraic reconstruction technique (ART) [9].

2.2 Advantages of object-oriented programming

Unlike procedural programming languages which separate data from operations on data defined by procedures and functions, object-oriented programming languages consist of a collection of interacting high-level units, the *objects*, that combine both data and operations on data. This renders objects not much different from ordinary physical objects. This resemblance to real things gives objects much of their power and appeal. They can not only model components of real systems, but equally as well fulfil assigned roles like components in software systems. They interact and communicate with each other using messages.

Variables and methods common to every object of a certain kind define a *class*. Inheritance between objects allows the programmer to define a hierarchy of objects, where each object takes on the attributes and behaviours of its ancestors. This hierarchy is an inverted tree structure (with the root at the top and the leaves at the bottom) where each lower level of the tree defines more specific attributes about a particular class of object.

2.3 Description of the reconstruction library

We have designed a library of classes and functions for 3D PET image reconstruction. Its modular design uses the object-oriented features of C++: self-contained objects hide implementation details from the user, and hierarchies of objects are implemented using inheritance. The library contains classes and functions to run parts of the reconstruction in parallel on distributed memory architectures. This allows to run the library on massively parallel computers, as well as on clusters of workstations.

The library has been designed so that it can be used for many different algorithms and scanner geometries including both cylindrical PET scanners and dual-head rotating coincidence gamma cameras. It is portable on all computer systems supporting the GNU C++ compiler or Microsoft Visual C++.

The building block classes included in this library can be described as follows (Figure 1):

- information about the data to be reconstructed (PET scanner characteristics, PET study, algorithm type, ...);
- memory allocations of multi-dimensional arrays or tensors (1D, 2D, 3D or 4D);
- reading and writing (I/O) data in Interfile format (for which a 3D PET extension is proposed);
- manipulation of multi-dimensional arrays;
- classes for projection data (the complete dataset, segments, sinograms, viewgrams) and images (3D and 2D);
- various filter transfer functions (2D and 1D);
- Fast Fourier Transform (FFT) utilities;

- forward projection operators (ray tracing method using Siddon's algorithm [10]);
- backprojection operators (incremental, beamwise interpolating backprojection using Cho's algorithm [11]);
- trimming, mashing, and zooming utilities on projection and image data.
- classes for both analytic and iterative reconstruction algorithms;
- stream-based classes for message passing between different processes, built on top of PVM (Parallel Virtual Machine), EPX (the native library provided with Parsytec-CC systems), or (in development) MPI (Message Passing Interface).
- classes and functions to use all these modules nearly transparently in a master-slave architecture.

The advantages of having such a library are:

- modularity and flexibility of the reconstruction building blocks to implement new reconstruction algorithms,
- possibility to use the same software implementation of reconstruction building blocks to perform image reconstruction on different scanner architectures (cylindrical PET scanners as well as dual-head coincidence gamma cameras).

Furthermore, operators can work indifferently on data stored in sinograms, viewgrams (set of parallel projections), or coincidence list mode.

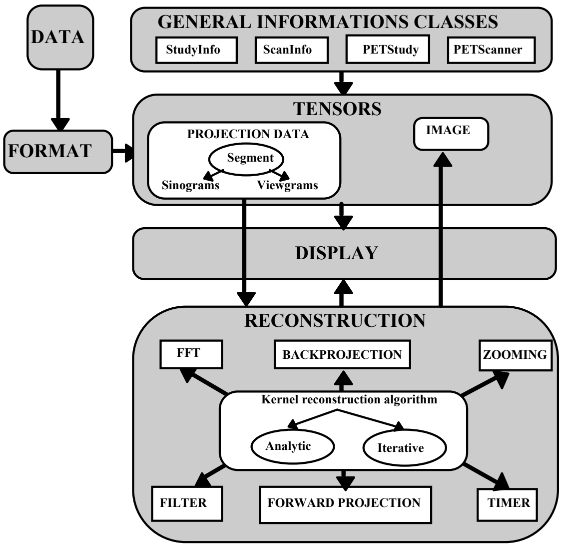

Figure 1. *Chart of the object-oriented library showing its major building blocks .*

2.4 Data structure

The hierarchy of data structures implemented in the library is presented in Figure 2. This is not a hierarchy of class derivation, but of containment. Let a ***PETSegment*** be the 3D data structure by abuse of terminology. A ***PETSegment*** may be expressed in two flavours or derived classes (***PETSiogram and PETViewgram***), depending on the order of the coordinates *View*, *Ring* and *Bin*. Thus, the user has the capacity to decide depending on the scanner geometry what is the best and fastest way to organise the data.

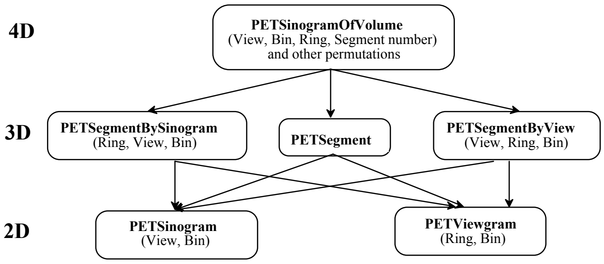

Figure 2. *Hierarchical view of the data structures.*

2.5 The projection/backprojection operators

The *forward projection operator* is based on the implementation of Siddon's algorithm [10], calculating the length of intersection of LORs with image voxels and making use of geometrical symmetries of the image volume.

The *backprojection operator* is based on an extension to 3D cylindrical geometry of Cho’s incremental algorithm [11]: the 3D beamwise, incremental backprojection method [12]. Voxel update values are found from bi-linear interpolation between four projection elements. Sweeping the image volume beam by beam, where a beam is the volume delimited by the central rays of four adjacent projection elements, allows to make use of a number of beam constants. Combining this with the exploitation of the geometrical symmetries of the image volume results in a very fast implementation of the backprojection operator.

3. Software implementation and timing

The above described library was implemented on Sun UltraSparc and Pentium processors, and on the Parsytec-CC system, a distributed memory, message passing parallel computer, globally classified into the multiple instructions operating independently on multiple data (MIMD) category of parallel computers.

The architecture of the Parsytec-CC system is based on the mainstream Motorola MPC 604 processor running at 133 MHz with 512KB L2-cache. The modules are connected together at 1 Gbits/s with high speed (HS) link technology according to the IEEE 1355 standard, allowing data transfer at up to 75 Mbytes/s. The system was integrated to the local area network using standard Ethernet. Currently a CC node has a peak performance of 266 MFlops.

The amount of computer time required for PROMIS reconstruction on a single node of the above described system from data sets obtained from three different PET scanners can be split up into overhead (matrix handling and I/O), filtering, projection, and backprojection times as detailed in Figure 3.

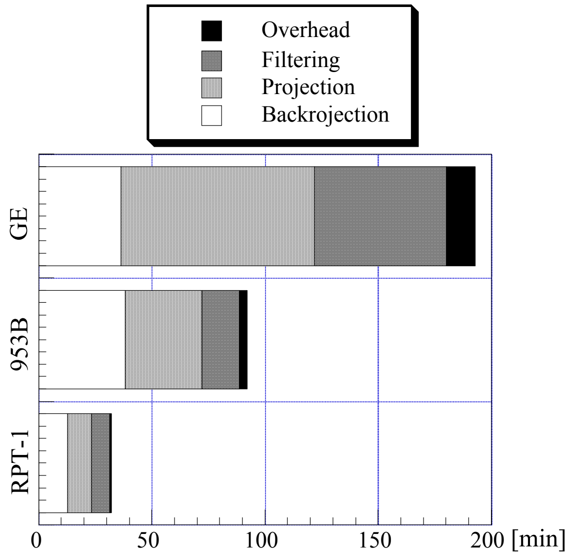

Figure 3. *Compared execution times in minutes of the PROMIS algorithm applied to the data sets of the (a) RPT-1* (16 rings, 256 sinograms, each 96 views by 128 elements)*, (b) ECAT-953,* (16 rings, 256 sinograms, each 192 views by 160 elements) and *(c)* GE-Advance (18 rings, 265 sinograms, each 336 views by 281 elements) *PET tomographs.*

Figure 4 shows representative slices of brain and thorax patient scans reconstructed with PROMIS and OSEM. The later is shown for comparative purposes.

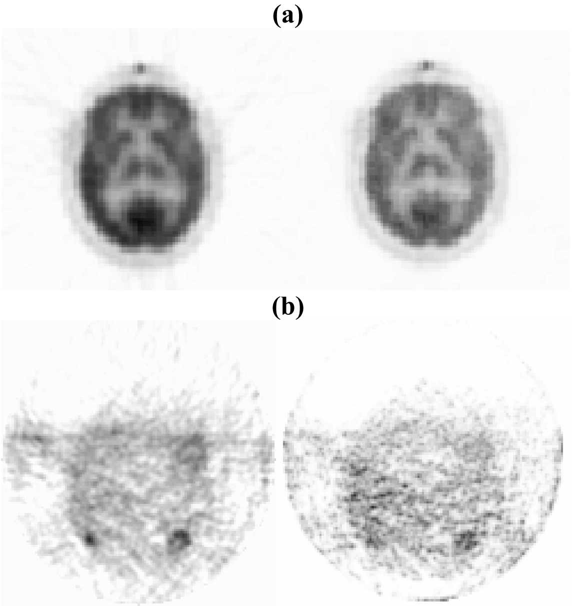

Figure 4. ***(a)*** *Brain image with an external marker and* ***(b)*** *oncologic clinical images as reconstructed with two reconstruction algorithms implemented in the present library: PROMIS (left); and OSEM (right). For the latter, 24 subsets and 5 iterations followed by a Gaussian filter were applied.*

4. Discussion and conclusion

In this paper, we described a flexible library which can be used to study current and implement new reconstruction algorithms for a wide variety of scanner geometries. The use of the object-oriented library described above has helped greatly to (a) compare analytic and iterative methods, (b) develop new iterative algorithms (c) adapt and apply the developed reconstruction algorithms to different designs of tomographs. The powerful constructs promoted by object technology can yield elegant, quality code. This programming paradigm makes it possible to envision incremental refinements to the building blocks described in this paper with maximum code reuse by providing a framework for effectively defining standards using the inheritance mechanism. This approach streamlines development and improves reliability.

Elapsed computational time for reconstruction is an issue in the acceptance of a reconstruction algorithm, hence parallel processing approaches are gaining in importance. Object-oriented programming also results in an undeniable computing time performance penalty associated with the service requests between the objects, because they are implemented as function calls. Moreover, objects are created and destroyed at a huge rate and the dynamic memory allocation/deallocation increases the execution time. The accuracy of iterative algorithms depends strongly on the performance of the projector-backprojector pair used. The influence of different approximations on the projectors, in particular the piece-wise linear interpolation used is discussed elsewhere [13].

Acknowledgements

This work was supported by the European Esprit LTR project PARAPET (EP23493) and the Swiss Federal Office for Education and Science under grant 96.193. The authors thank their partners within the project for fruitful discussions and useful suggestions.

References

[1] Barrett T *et al* **(1998)** “A review of methods and techniques used in 3D PET image reconstruction,” *submitted for publication to* Medical Image Analysis.

[2] Kinahan PE and Rogers JG **(1989)** “Analytic 3D image reconstruction using all detected events,” IEEE Trans. Nucl. Sci. 36: 964-968.

[3] Defrise M, Kinahan PE and Townsend DW **(1995)** “A new rebinning algorithm for 3D PET: principle, implementation and performance” Conf. Rec. 1995 International meeting on fully three-dimensional image reconstruction in radiology and nuclear medicine, Aix-Les-Bains, France, pp 235-239.

[4] Shepp LA and Vardi Y **(1982)** “Maximum likelihood reconstruction for rmission tomography,” *IEEE Trans Med Imag*. 1: 113-122.

[5] Hudson HM and Larkin RS **(1994)** “Accelerated image reconstruction using ordered subsets of projection data,” *IEEE Trans. Med. Imag*. 13: 601-609.

[6] Green PJ **(1990)** “Bayesian reconstructions from emission tomography data using a modified EM algorithm,” *IEEE Trans .Med .Imag*. 9: 84-93.

[7] Kaufman L **(1993)** “Maximum likelihood, least squares and penalized least squares for PET,” *IEEE Trans .Med. Imag*. 12: 200-214.

[8] Depierro AR **(1993)** “On the relation between the ISRA and the EM algorithm for positron emission tomography,” *IEEE Trans. Med. Imag*. 12: 328-333.

[9] Herman GT and Meyer LB **(1993)** “Algebraic reconstruction techniques can be made computationally efficient,” I*EEE Trans. Med. Imag.* 12: 600-609.

[10] Siddon RL **(1985)** “Fast calculation of the exact radiological path for a three-dimensional CT array,” *Med. Phys.* 12: 252-255.

[11] Cho ZH, Chen CM and Lee SY **(1990)** “Incremental algorithm – A new fast backprojection scheme for parallel beam geometries,” *IEEE Trans. Med. Imag*. 9: 207-217.

[12] Egger M, Joseph C, and Morel C. **(1998)** “Incremental beamwise backprojection using geometrical symmetries for 3D PET reconstruction in a cylindrical scanner geometry”,.*Phys. Med. Biol.* 43: 3009-3024.

[13] Thielemans K, Jacobson MW, Belluzzo D **(1999)** “On various approximations for the projectors in iterative reconstruction algorithms for 3D-PET,” *3D’99 Conf. Proc. pp 232-235*