

STIR: Software for Tomographic Image Reconstruction

Release 2

Kris Thielemans^{1,2}, Charalampos Tsoumpas^{1,3,4,5,10}, Sanida Mustafovic^{1,3}, Tobias Beisel⁶, Pablo Aguiar^{7,8}, Nikolaos Dikaos^{4,11} and Matthew W Jacobson⁹

¹Hammersmith Imanet Ltd, part of GE Healthcare, London, United Kingdom

²Algorithms and Software Consulting Ltd, London, United Kingdom

³MRC Clinical Sciences Centre, Imperial College London, United Kingdom

⁴School of Electrical and Computer Engineering, National & Technical University of Athens, Greece

⁵Division of Imaging Sciences & Biomedical Engineering, School of Medicine, King's College London, United Kingdom

⁶Paderborn Center for Parallel Computing, University of Paderborn, Germany

⁷Unitat de Biofísica, Facultat de Medicina, University of Barcelona, Spain

⁸Molecular Imaging Group, Fundació IDICHUS/IDIS, Santiago de Compostela, Spain

⁹Xoran Technologies, Ann Arbor, MI, USA

Email: Kris.Thielemans@ieee.org

Keywords: image reconstruction; Positron Emission Tomography; open source software

Abstract. We present a new version of STIR, an Open Source object-oriented library implemented in C++ for 3D PET reconstruction. This library has been designed such that it can be used for many algorithms and scanner geometries, while being portable to various computing platforms. This second release enhances its flexibility and modular design and includes additional features such as Compton scatter simulation, an additional iterative reconstruction algorithm and parametric image reconstruction (both indirect and direct). We discuss the new features in this release and present example results. STIR can be downloaded from <http://stir.sourceforge.net>.

1 Introduction

Positron Emission Tomography (PET) is extensively used in modern internal medicine for diagnosis, therapy follow-up, drug discovery and clinical research. In recent years iterative image reconstruction techniques have gained importance and are the *de facto* standard for diagnostic imaging. However, obtaining optimal images for a particular task is an active area of research. Therefore a flexible set of tools is necessary to allow researchers to concentrate on the development, implementation and testing of new algorithms, without having to ‘reinvent the wheel’. Similarly, researchers on the designs of new scanner systems can use these tools to reconstruct either simulated or acquired data with existing algorithms such that they can concentrate on the most desirable design, without investing a considerable amount of time in the implementation of reconstruction software. Finally, tools for fast (analytical) simulation of acquisitions are useful in many cases where full Monte Carlo simulation is impractical or not necessary (Tsoumpas *et al*, 2011).

STIR (*Software for Tomographic Image Reconstruction*) is an **open source C++ library** for reconstruction and processing of PET images that attempts to fulfil these needs. Its aim is to provide a Multi-Platform Object-Oriented framework for research into all data (e.g. images and sinograms) manipulations related to tomographic image reconstruction. Currently, the emphasis is on (iterative) image reconstruction in PET. STIR is not approved for diagnostic applications.

STIR originated in the PARAPET project (1997-1999), which was a European Union funded collaboration on 3D PET reconstruction algorithms. The collaborators of the PARAPET project designed a library of classes and functions for 3D PET image reconstruction (Labbé *et al*, 1999a; Labbé *et al*, 1999b). This library was released as Open Source in June 2000. Following the completion of the project, the software was renamed to STIR and maintained occasionally. The first release of STIR remained restricted to reconstruction, simulation and manipulation / processing of

¹⁰ Contact email address: charalampos.tsoumpas@kcl.ac.uk

¹¹ Dr N Dikaos is currently with the department of Medical Physics, University College London, email: n.dikaos@ucl.ac.uk

static PET data in sinogram format. STIR has been widely used in literature especially as a tool for evaluation purposes and is already used by a relatively large number of scientists. For example, about 125 new users have registered from January until November 2011. The project has three mailing lists (in November 2011 about 200 members for the announcements list, 240 members for the users list and 75 members for the developers list, with many people probably subscribed to more than one list).

In the second major release we provide a major extension and restructuring of the STIR library.¹² The updates offer a more flexible design such that many components can be reused for new reconstruction algorithms. These algorithms can be also applied to different data domains, as for instance list-mode data or dynamic histogram data. Moreover, they can be used to solve other problems as well, such as direct parametric image reconstruction. Additional components have been included as well; such as scatter estimation and correction and a parallel implementation of the gradient computation for the Poisson log-likelihood (utilisable by the different reconstruction algorithms as discussed below). These updates contribute towards the long-term goal of STIR to provide a complete and flexible toolkit for exploring data processing techniques related to image reconstruction in tomography.

STIR is portable to all computer systems supporting the GNU C++ compiler, Intel C++, Lang C++ or Microsoft Visual C++, but we expect no major problems with other ANSI C++ compliant compilers. Most of STIR is distributed under the Lesser GNU Public License; some utilities are made available under the more restrictive/protective GNU Public License. Both GPL and LGPL allow commercial use and redistribution under some conditions related to providing source code. STIR is available for download at the website: <http://stir.sourceforge.net>.

There are a number of other software packages available for image reconstruction. Fessler and his students have implemented an Image Reconstruction Toolbox¹³ as a set of open source matlab¹⁴ routines for image reconstruction and processing. In addition, Fessler has also developed ASPIRE (Fessler, 2004). However, this is distributed via executable files and its license allows non-commercial usage only. Another software is TIRIUS¹⁵, which provides an implementation of OSEM for PET image reconstruction using a well-designed GUI. It can also use reconstruction algorithms available in STIR 1.4. TIRIUS is distributed under the GNU Lesser General Public License. The new package Presto (Scheins *et al*, 2011) provides reconstruction software for PET based on efficiently stored matrices, but it is currently not publically available. NiftyRec¹⁶ (Pedemonte *et al*, 2010) is a recently released package (BSD-style license), which provides many different reconstruction algorithms, where part of the operations can occur on GPU hardware. QSPECT (Loudos *et al*, 2010) is a recently released package for SPECT reconstruction using MLEM or OSEM. OSCaR (Rezvani *et al*, 2007) provides a matlab-based implementation of the FDK algorithm for Cone-beam CT¹⁷ and it is freely available to AAPM members, but it is not clear if non-members can also use it.

All these packages concentrate on image reconstruction for static imaging. STIR 2 is unique in providing a flexible open source framework including scatter estimation and routines for dynamic imaging.

In this paper, part of which has been published previously by Thielemans *et al*. (2006), we describe the general structure of the library, discuss the major changes in this release and provide example results. In the text, we refer to previous papers describing and validating the various components of STIR.

¹² The features described in this paper correspond to release 2.1, dated 30th of June 2011.

¹³ The *Image reconstruction toolbox* can be downloaded from <http://www.eecs.umich.edu/~fessler/code/>

¹⁴ Matlab User's Guide, The MathWorks Inc., Natick, MA, 1998

¹⁵ TIRIUS (Tomographic Image Reconstruction Interface of the Université de Sherbrooke) can be downloaded from <http://www.pages.usherbrooke.ca/jdleroux/Tirius/TiriusHome.html>

¹⁶ NiftyRec can be downloaded from <http://sourceforge.net/projects/niftyrec>

¹⁷ OSCaR can be downloaded from <http://www.cs.toronto.edu/~nrezvani/OSCaR.html>

This paper is organised as follows: in the next section we give an overview of the software aspects, then we present some results to illustrate the capabilities of this release of STIR, ending with a discussion of the potential of the library.

2 Description of the software

2.1 Software architecture

STIR is organised into three parts: the core library, several utilities providing command-line functionality and a framework for automatic testing. Many users of STIR only use the command-line utilities to estimate correction factors for the data, reconstruct the image, manipulate data or images etc. Most utilities use a parameter file that is a simple text file with Interfile-like syntax. Via these parameter files, STIR allows the user to select different components such as projectors at run-time. However, the implementation of new functionality such as additional projectors or reconstruction algorithms requires new C++ coding extending the library appropriately.

To facilitate enhancing STIR, the software makes extensive use of the object-oriented features of C++. A class is defined to describe a certain kind of objects. This class encodes both data and methods to perform a certain action. As each object not only contains data but also defines all possible interactions with the data, object-oriented programming leads to a modular and flexible design. A hierarchy of these classes can be defined using inheritance, providing for each class the attributes and behaviour of its ancestors. This hierarchy is an inverted ‘tree’ structure, which schematically corresponds to the ‘root’ being at the top and the ‘leaves’ at the bottom. Inheritance encourages code re-use by avoiding recoding operations that are common to different classes, but also by allowing software to work on all classes that are derived from a class at a particular level in the hierarchy. As an example, STIR has class hierarchies for the forward and back projection operations common in the field of PET image reconstruction. A reconstruction algorithm can therefore be written in terms of generic projectors, without needing further knowledge of how the projection operations are completed.

Although class hierarchies offer numerous advantages, there are certain limitations regarding their run-time performance and extendibility. In these situations, STIR incorporates C++ templates, a powerful feature of C++ that allows writing generic code for different data-types suffering from run-time performance penalties. As an example, STIR 2 contains a template-class for *objective functions*, i.e. a function of certain parameters, such as the log likelihood in image reconstruction, see §2.3.1. This class is expressed in the form of the template according to the type used to store the parameters, allowing STIR to have a single implementation of reconstruction algorithms e.g. OSEM (Hudson and Larkin, 1994) that works for both 3D and 4D images without sacrificing performance.

An important feature of STIR is that the user is able to select certain functionality such as a projector or output file format at run-time. STIR keeps registries of all implemented components. A mechanism is provided for a developer to add a new component (such as a projector or a new output file format) to the relevant registry. After re-linking a utility or reconstruction program, the new component will be available for the user.

2.2 Overview of components in STIR

STIR has been designed so that it can be used for many different algorithms and scanner geometries. The library classes can be divided into three sets:

“Supporting” building blocks

- Template multi-dimensional arrays (any dimension and data-type) and numerical operations, including n-dimensional Fast Fourier Transform (FFT).
- Various filter transfer functions (3D, 2D and 1D).

- Information classes about the data such as characteristics of the PET scanner (currently restricted to single-layer cylindrical geometry), sinogram formats, images etc. STIR 2 now also includes classes for dynamic data.
- Classes describing geometric shapes that can be used for image generation or region of interest (ROI) calculations.
- Classes for parsing parameter files allowing the user to make at run-time choices of filters, data formats, algorithms etc.
- Classes for reading and writing (I/O) data in various formats, either natively or via conversion. Furthermore, this release contains an extendable and modular way to add new IO routines using ‘factories’ (Gamma *et al*, 1995). In addition to its native file format, i.e. Interfile (Todd-Pokropek *et al*, 1992), STIR comes with interfaces to external libraries such as the ECAT[®] Matrix library¹⁸, and Analyze[™] AVW[™] (Robb, 2001; Robb *et al*, 1989). Classes for reading list mode data are also provided in this release. Currently only list mode data from the Siemens ECAT[®] EXACT[™] HR+ and Siemens ECAT[®] EXACT[™] 3D are supported, but the classes are extendable to other scanners.

Scan modelling

- Classes for projection operations, such as:
 - Probability matrix operations: ray-tracing method using a variation of Siddon's algorithm (Siddon, 1985), optionally using multiple rays per detector-pair; interpolation in sinogram space and matrices stored on disk. Symmetries are used to reduce the matrix size such that it can be stored in memory for most systems.
 - On-the-fly forward projection operators: ray-tracing method using a variation of Siddon's algorithm that takes symmetries into account.
 - On-the-fly back-projection operators: incremental, beam-wise interpolating back-projection using Cho's algorithm (Cho *et al*, 1990; Egger *et al*, 1998) taking symmetries into account, as well.
- Classes for other data manipulations in PET such as normalization, attenuation correction, geometric correction and scatter estimation (see §2.3.3).
- Classes related to parametric images and kinetic modelling (Patlak *et al*, 1983).

Optimisation

- Classes for ‘generalized’ objective functions such as the Poisson emission log-likelihood, optionally with added penalties such as the ordinary quadratic prior or generalisations of the Median Root prior (§2.3.1).
- Classes for analytic and iterative reconstruction algorithms (§2.3.2).

Special care has been taken regarding the extendibility and the most possible reutilization of the existing code. For example, projectors, objective functions and optimisation algorithms are independent components, which can be mixed at run-time (subject to certain limitations such as the fact that OSEM needs a Poisson objective function). Therefore, if a new forward projector is implemented, only its specific features and characteristics need to be defined and then all available reconstruction algorithms can make use of it.

As mentioned above, STIR contains also the following components:

Command-line utilities

- Algebraic manipulations on images and projection data.
- Utilities that use the reconstruction, image generation, filtering, ROI and other classes.

¹⁸ The ECAT[®] 7.0 compatible C library (by M Sibomana and C Michel) can be obtained from the link ftp://dormeur.topo.ucl.ac.be/pub/ecat/z_matrix_70 or via the OpenGATE web site (available to registered GATE users only).

- Conversion of the output sinograms produced by the Monte Carlo package SimSET (Lewellen *et al*, 1998) into a suitable format which can be used for STIR reconstructions. It is also possible to convert STIR images into the indexed format suitable for SimSET and vice versa.

Testing framework

- In order to reassure the proper functionality of many building block classes, STIR contains corresponding test classes that can be performed as part of the installation process.
- Scripts are provided for automatic testing of forward projection, attenuation and subsequent reconstruction. Tests are validated against the expected ROI values.
- An additional ‘test pack’ is also available for download. This contains example data, corresponding reconstructed images and accompanying scripts for performing reconstructions and voxel-wise comparisons with results obtained by the developers.

Finally, extensive documentation is available including documentation automatically generated from comments in the C++ source code using the *doxygen tool* (<http://www.doxygen.org>). STIR also has a wiki page (<http://sourceforge.net/apps/mediawiki/stir/>) including for instance a list of Frequently Asked Questions.

2.3 Description of the main new features in STIR 2

The most important change of this release is the redesign of the reconstruction classes. A new hierarchy has been created for generalized objective functions which are used in the iterative optimization process. From a practical perspective, an important addition is the scatter estimation code for 3D PET. We describe these features in more detail.

2.3.1 Generalised objective functions.

Most iterative reconstruction algorithms are derived from an optimization criterion, such as least-squares minimization, maximum likelihood (ML) or maximum a posteriori (MAP). Each algorithm attempts to optimize the corresponding objective function. However, in practice various modifications are made to these algorithms such that a general objective function no longer exists. Examples are the One Step Late Median Root Prior algorithm (Alenius and Ruotsalainen, 1997; Bettinardi *et al*, 2002) which defines a ‘gradient’ for a prior, although no function corresponds to this ‘gradient’; or other cases where the forward and back projectors are not transposed operators.

In STIR, the generalized objective function hierarchy (Figure 1) provides the functionality to compute a ‘subset-gradient’ (i.e. ‘gradient’ of the objective function over a subset of data). This allows generic algorithms such as (preconditioned) gradient-descent to work on any objective function implemented in STIR. Many algorithms in PET reconstruction crucially involve subset-gradient computations, but they can be expressed in terms of a subset-gradient operation without detailed knowledge of the data.

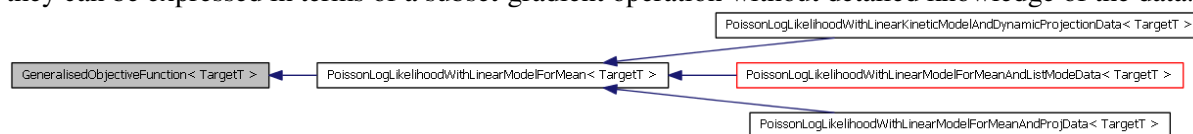


Figure 1 Class hierarchy for objective functions. All classes in this diagram are templated in the type of the parameters that are being estimated (indicated here as `TargetT`). Arrows indicate parent-class relationship, i.e. the derived class inherits and/or implements all features of the parent class. The red box indicates that not all classes in the hierarchy are shown.

A crucial feature is that the generalized objective functions (and reconstruction algorithms) are implemented as ‘templated’ in the type of the parameters (indicated as **TargetT** in Figure 1) over which the optimization is performed. This provides the opportunity that the unknown variables are not necessarily voxel-values in 3D images, but they can also be other data-types such as parametric maps, as discussed below.

2.3.1.1 Objective function for parametric image estimation from dynamic data

In dynamic PET studies, the changing activity of the injected radiotracer is measured through multiple consecutive time frames. The distribution of the physiological parameter of interest is estimated by subsequent application of an appropriate pharmacokinetic model to the time-activity curve of the radioactivity in each image element. The whole process generates images of the distribution of the kinetic parameters, which are known as parametric images. Usually, these parametric images are estimated from reconstructed images. However, it is possible to combine image reconstruction and kinetic model estimation into one optimisation problem, a process sometimes known as ‘direct parametric image reconstruction’. To facilitate this, STIR 2 uses an objective function for dynamic data that includes a kinetic model in the model of the measured data. This is currently implemented by letting the ‘parametric’ objective function store an array of ‘static’ objective functions of type **PoissonLogLikelihoodWithLinearModelForMeanAndProjData**, one for each time frame. Calls for computing the gradient of the parametric objective function at a given parametric image are handled by letting the kinetic model compute the corresponding dynamic image, computing the gradients for each time frame using the ‘static’ objective functions, and finally ‘back-projecting’ these gradients from the dynamic domain to the parametric domain using the gradient of the kinetic model:

$$\frac{\partial \Theta}{\partial \theta_v} = \sum_{f'} \frac{\partial L_f}{\partial x_{f'}} \frac{\partial \tilde{x}_{f'}}{\partial \theta_v}$$

where Θ is the parametric objective function, θ_v a kinetic parameter for a given voxel v , L_f is the ‘static’ objective function for time frame f , $x_{f'}$ the value of the voxel in the dynamic image and $\tilde{x}_{f'}$ the kinetic model for that voxel. Although this approach is not optimal for linear kinetic models, it allowed for maximum code reutilisation and will also allow easy generalisation towards non-linear models in the future.

2.3.1.2 Parallel computations of the Poisson log-likelihood for emission sinogram data.

Part of the aim of the PARAPET project was the study of parallel implementations of iterative algorithms (Labbé *et al.*, 1999a). Jacobson *et al.* (2000) describe the parallel implementation of OSEM (with inter-update filtering) that was implemented by the PARAPET partners. These ideas have now been ported to STIR and extended by adding further classes and modifications to be able to compute the Poisson log-likelihood and its gradient on distributed computing platforms using the Message Passing Interface (MPI). This allows the parallel implementation to run on many different hardware platforms, from a multi-computer cluster to a single multi-core workstation system.

The parallelisation was not implemented for the projection operations but instead at the level of the gradient and objective function computation. When using distributed systems, this strategy has the advantage that much less communication is needed resulting in higher performance. The calls to the objective function classes are the most commonly used (and CPU intensive) calculations for image reconstruction.

The software uses a ‘master-slave’ paradigm, similar to Johnson and Sofer (1999). ‘Slaves’ execute ‘workpackages’ based on a subdivision of the projection data. Following each sub-iteration step, the image-updates are completed at the master node and a new image estimate is broadcasted to all slaves. A special workpackage allocation algorithm has been designed that incorporates load balancing and improves the reutilisation of cached data. This is based on a ‘workpool’ approach where idle slave nodes request the work that is supplied by the master node from a pool of the remaining workpackages. The following two steps reduce the amount of communicating data: Firstly, the slaves cache the data for successive iterations; and then the workpackages that have been already cached by the requesting slaves are prioritised.

The parallel code is integrated in the **PoissonLogLikelihoodWithLinearModelForMeanAndProjData** class, therefore benefitting all reconstruction algorithms using this objective function, and it can be activated

or excluded by compiler flags. Future extensions may include a parallel version for list-mode data as well as the support of Graphic Processing Units as an alternative fast architecture. The parallel implementation is described in section 3.2 and in more detail in Beisel *et al.* (2008).

2.3.2 Image reconstruction algorithms.

In the previous release, STIR included analytic algorithms (2D FBP and 3D FBP using the re-projection algorithm (Kinahan and Rogers, 1989) validated by Dinelle *et al.* (2004) and one iterative algorithm (called OSMAPOSL) which is an ordered subsets version of the One Step Late algorithm (Green, 1990a, b) with optional inter-update (Jacobson *et al.*, 2000) and l or inter-iteration filtering (Silverman *et al.*, 1990). By selecting appropriate options, OSMAPOSL reduces to MLEM (Shepp and Vardi, 1982) and OSEM (Hudson and Larkin, 1994).

The reconstruction class hierarchy for STIR 2 is shown in Figure 2. OSMAPOSL has been rewritten in terms of the objective function hierarchy described in the previous section. This makes it applicable to any problem where the measured data follows (independent) Poisson statistics and the measurement mathematical model is linear. For example, STIR 2 provides an objective function for list mode data, allowing the use of the Maximum Likelihood algorithm from Parra and Barrett (1998).

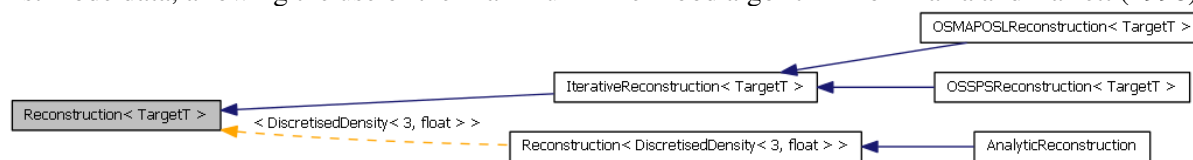


Figure 2 Class hierarchy for reconstruction algorithms (FBP2D and FBP3DRP are not shown for brevity). See previous figure for an explanation of the concept of the diagram. The dotted arrow indicates a parent-class relation where the template-type (i.e. TargetT) is specified. This occurs here because the analytic reconstruction methods in STIR are restricted to 3D volumes.

A fundamental drawback of OSEM is the fact that it does not converge to a single solution, except in special circumstances (i.e. exact subset-balancing and existence of a consistent solution (Hudson and Larkin, 1994)) which are rarely observed in practice. In addition, even without subsets, the One Step Late algorithm does not converge when the initial estimate is far from the MAP solution. Many algorithms have been proposed to overcome these problems (Qi and Leahy, 2006). This release of STIR includes an implementation (Mustafovic and Thielemans, 2004) of the Ordered Subsets Separable Paraboloidal Surrogate (OS-SPS) algorithm with relaxation (Erdoğan and Fessler, 1999) using the objective function hierarchy. It can therefore use the parallel implementation discussed in §2.3.1.2 and is also directly applicable to list mode data as well. OS-SPS could be applied to transmission data reconstruction by implementing a suitable objective function.

In §2.3.1.1, an objective function for dynamic data was described. This gives the opportunity to extend all current reconstruction algorithms in STIR to be able to directly reconstruct parametric images from the raw PET data (if the kinetic model is linear in its parameters)¹⁹. For instance, this release includes an implementation (Tsoumpas *et al.*, 2008) of a subset version of the Parametric Image Reconstruction (PIR) algorithm (Matthews *et al.*, 1997), reusing all of the previously existing OSMAPOSL code. Similarly, due to the flexible platform which has decoupled the objective function from the reconstruction algorithm, the OS-SPS implementation can be also used to reconstruct parametric images directly from sinogram data (Tsoumpas *et al.*, 2007). However, it has been already shown that these parametric algorithms are very slow even with high number of subsets (Angelis *et al.*, 2011) and further extensions are necessary to make them practical, as for example recently suggested by Wang and Qi (Wang and Qi, 2010).

2.3.3 3D Scatter estimation.

¹⁹ This release also includes an implementation of the Patlak plot applied to reconstructed dynamic images.



Figure 3 Transverse slices through reconstructed images at 720 subiterations. *Left: OS-OSL; Right: OS-SPS.*
Grey scale is identical in both images.

Coincidence events where the photons may have been scattered form a major contribution to the data in 3D PET. This scatter background needs to be estimated and taken into account during reconstruction. We have implemented and evaluated (Tsoumpas *et al.*, 2004) a version of the Single Scatter Simulation (SSS) algorithm (Poenisch *et al.*, 2003; Watson, 2000; Watson *et al.*, 1996; Werling *et al.*, 2002). The setup of the algorithm is:

- (i) Estimate the scatter sinogram given the necessary settings (e.g. energy thresholds and resolution, down-sampled attenuation image, down-sampled scanner);
- (ii) Interpolate scatter sinogram to match the data sinogram using B-splines interpolation;
- (iii) If necessary scale the interpolated scatter estimate to the measured data outside the object to account for multiple and / or events originating out of the field of view.

Special care has been taken such that the single scatter estimate has appropriate scale compared to using the STIR forward projectors to compute the unscattered sinogram. This makes it easier to compare to Monte Carlo data, but also to consider higher order or outside the field of view scattering by proper scaling (Thielemans *et al.*, 2007).

Execution time depends on the scatter simulation settings. For instance, for the estimation of the coarse sinogram, computation time is almost proportional to the number of scatter points and the number of detectors used for the coarse sinogram. As an example, for a GE[®] Discovery[™] STE simulating reasonably accurate coarse 2D sinograms (41 radial positions, 35 views, 8 axial positions) takes about 40s on a recent laptop (Intel[®] iCore[™] i7-2620M CPU @ 2.70GHz using GCC 4.5.3) with another 30s for scaling and upsampling to fully 3D.

3 Materials and Results

In this section we demonstrate results that were obtained using STIR as an illustration of its capabilities. For validation of the implementation and algorithms we refer to previous publications.

3.1 Comparison of reconstruction algorithms

Here we compare results obtained with two reconstruction algorithms (OS-SPS and OS-OSL) on a late time frame of a dynamic ¹⁸F-DOPA brain study. Data were acquired with a Siemens EXACT 3D scanner (48 rings, 576 detectors per ring). The total number of prompt counts was about 30 million. Details about the acquisition can be found in Angelis *et al.* (2011). All ‘corrections’ were incorporated into the acquisition modelling (i.e. no pre-corrections were performed). Images were reconstructed on a $128 \times 128 \times 95$ grid with voxel size $2.25 \times 2.25 \times 2.425$ mm³. Approximate striatum and cerebellum 3D ROIs were obtained and kept fixed for all computations. The radioactivity in cerebellum is expected to be relatively uniform for ¹⁸F-DOPA, thus the standard deviation in this ROI is used as a measure of image noise.

First, an image after initialisation with a uniform image and a full OSEM iteration with 36 subsets was obtained. This image was used as initializer for both OS-OSL and OS-SPS. A moderate uniform

quadratic penalty was used in order to maintain good quantification at high iteration number whereas noise remains at reasonable level. OS-SPS used the relaxation scheme proposed by (Erdoğan and Fessler, 1999) starting with step size 1 and decreasing after every full iteration. Example transverse slices are shown in Figure 3 at 720 total subiterations.²⁰ Reconstructions are visually similar, with OS-OSL appearing slightly noisier at this iteration number.

Figure 4 shows a quantitative comparison: Up to 2000 subiterations the two algorithms seem to produce different ROI values, whilst OS-OSL stabilized much earlier. However, OS-SPS obtains a higher value of the (penalized) objective function. This behaviour is consistent with the literature. OSL is known to maximise the objective function - if it does not diverge - only when no subsets are used, while OS-SPS uses relaxation to converge to the true maximum independent of the number of subsets.

In order to investigate this further, we show results in Figure 5 where OS-OSL after 720 subiterations is followed by OSL (i.e. setting the number of subsets equal to 1), and similarly OS-SPS followed by SPS²¹. The results in this figure show that OS-OSL follows limit-cycle behaviour where different values are obtained for each of the 36 subsets. OS-SPS shows similar oscillations but the relaxation allows the gradual reduction of their amplitude significantly, as expected as this version of the algorithm is theoretically convergent (Ahn and Fessler, 2003). Recommencement of the algorithms without subsets is beneficial at early iterations as the mean ROI values coincide for both algorithms and the objective function increases. However, the convergence rate decreases rapidly and consequently both algorithms have not converged after 80 further iterations.

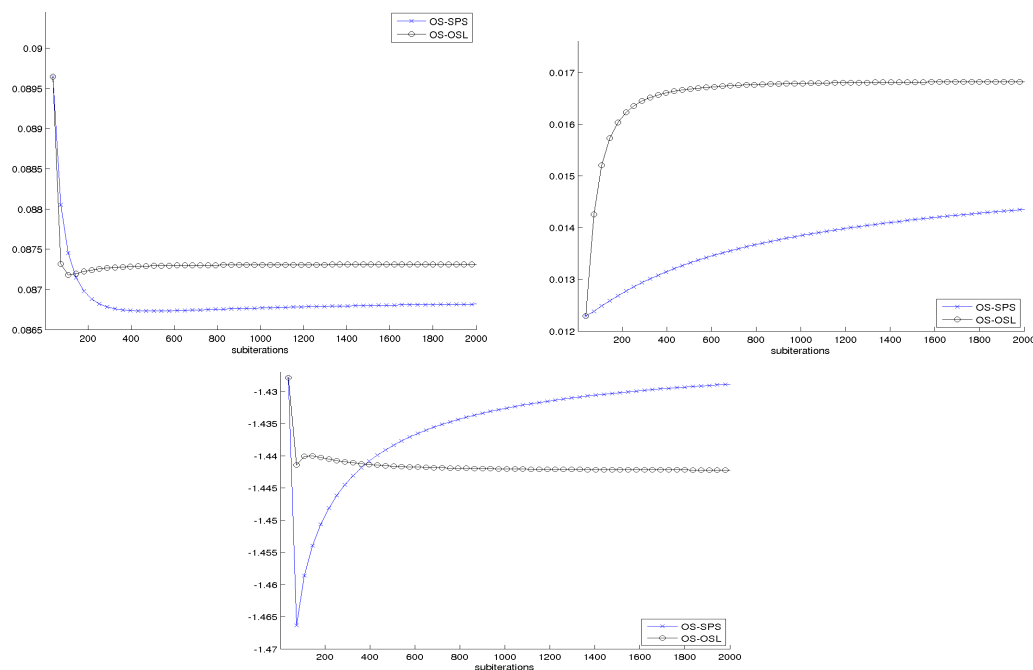


Figure 4 Plots comparing OS-OSL (black circles) with OS-SPS (blue crosses) over subiterations. *Top left:* striatum ROI mean; *Top right:* cerebellum ROI standard deviation; *Bottom:* Objective function. First data point is the OSEM image (plotted with the log likelihood only).

²⁰ As all cases are initialized with the image obtained after 36 OSEM subiterations, we have numbered subiterations continuing from 36, e.g. the first OS-OSL full iteration is obtained after 72 subiterations.

²¹ For SPS a fixed step-size of $1/3$ was used, as this is the value obtained after 720 subiterations with the relaxation scheme for OS-SPS.

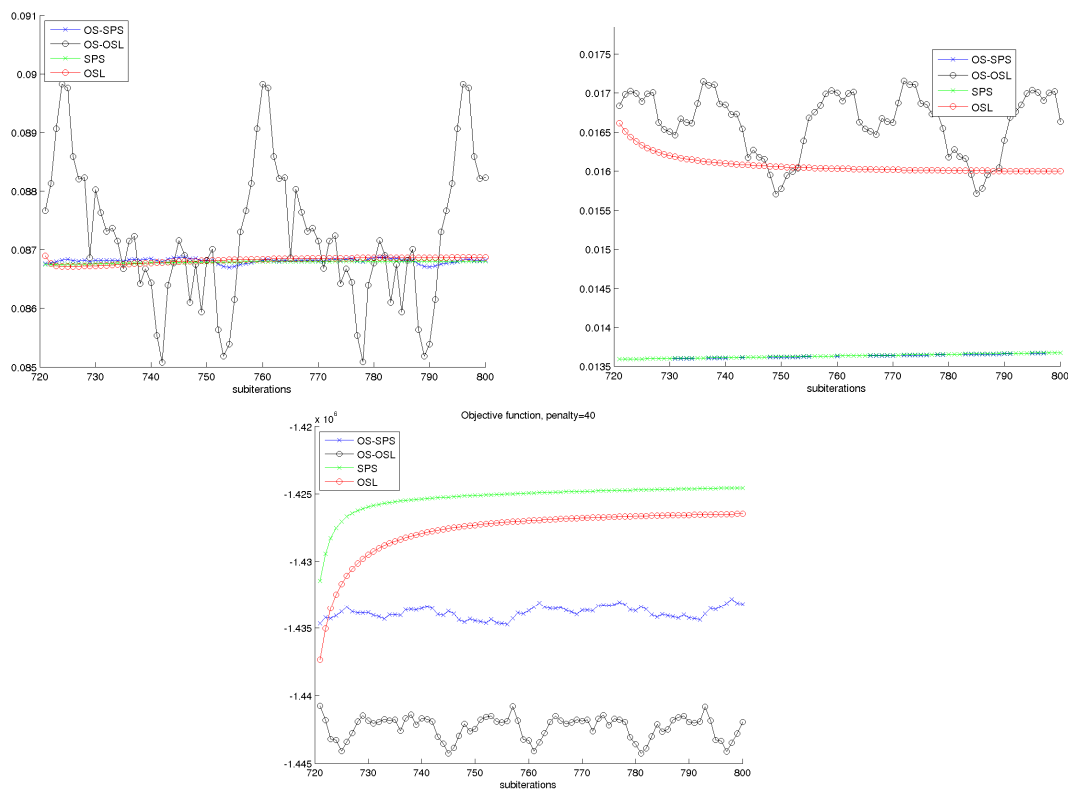


Figure 5 Plots comparing OS-OSL with 36 subsets (black circles), OS-SPS with 36 subsets (blue crosses), OSL (red circles) and SPS (green crosses) over subiterations. OSL was initialized with the OS-OSL image after 720 subiterations, and SPS was initialized with the OS-SPS image after 720 subiterations. *Top left:* striatum ROI mean; *Top right:* cerebellum ROI standard deviation; *Bottom:* Objective function.

3.2 Parallel implementation of image reconstruction for sinogram data

To investigate the efficiency of the parallelisation of the gradient operations discussed in §2.3.1.2, we present example timings using the OSEM algorithm (36 subiterations) for different number of subsets, see Beisel *et al.* (2008) for details. The data used was measured on an ECAT[®] EXACT[™] HR+ tomograph using the default settings for 3D PET (144 views, 288 tangential positions, 5 ‘segments’ of oblique sinograms). The reconstructed images had dimensions of $265 \times 265 \times 63$ with corresponding voxel sizes $2.25 \times 2.25 \times 2.425$ in mm.

The results shown in Figure 6 were produced on an 8-core Intel Xeon Clovertown system. Each of the 8 cores has its own 8 KB L1-cache and every 2 cores share 4 MB of L2-cache. Figure 7 shows results for the same datasets running the parallel code on a cluster system at the University of Paderborn, providing up to 400 Infiniband-connected Dual Intel Xeon 3.2 GHz EM64T processors with 2 MB L2-cache and 4 GB main memory each.

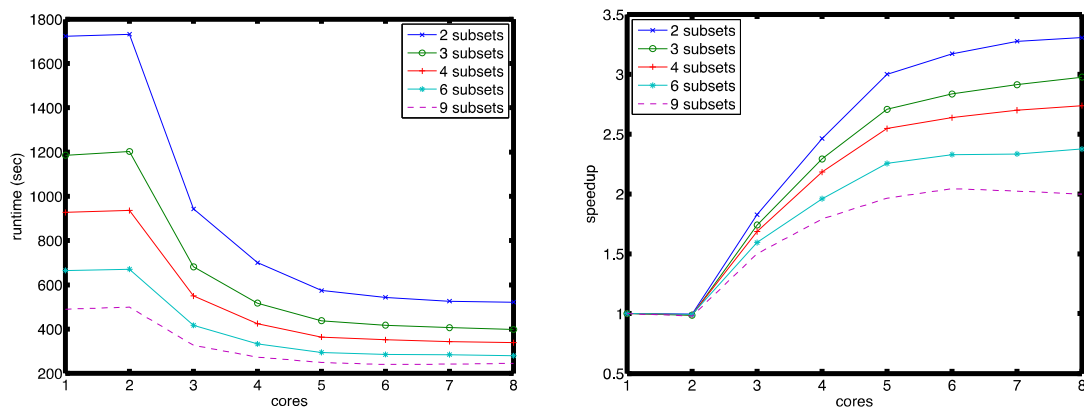


Figure 6 Runtimes and Speedups on the Intel Clovertown system. Figure from Beisel *et al.* (2008), reused with permission.

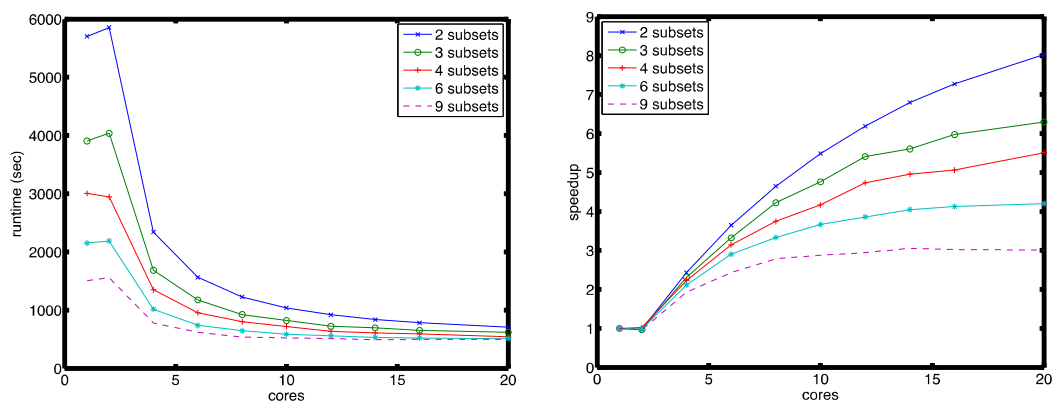


Figure 7 Runtimes and Speedups on the cluster system. Figure from Beisel *et al.* (2008), reused with permission.

The overall runtime is appreciably less on the shared memory system, but this is partly due to the different processor types. The speedup factor is expected to be larger for data acquired by modern PET scanners which consist of more detectors and on computing systems that make use of the latest developments in hardware.

Currently, only the calculation of the gradient and value of the Poisson log-likelihood is parallelized (see §2.3.1.2). Computations of the sensitivity image and penalty terms in MAP are still performed on the master node which reduces the parallelization efficiency. Similar efficiency results are obtained when using OSMAPOSL or OSSPS for the same objective function as the underlying code base is the same.

3.3 Accuracy of Scatter Simulation

Several thorough investigations of STIR scatter simulation have been already presented (Aguilar *et al.*, 2006; Dikaios *et al.*, 2006; Polycarpou *et al.*, 2011; Tsoumpas *et al.*, 2004), using Monte Carlo data provided by SimSET, phantom measurements, and clinical / preclinical data.

Example profiles through normalized sinograms acquired on an ECAT[®] EXACT[™] HR+ are shown in Figure 8 comparing measured data and scatter simulations for the NEMA cylindrical phantom and a line source placed at ~0mm and ~80mm. After scaling with a global factor determined in the tails of the sinograms, the simulated scatter distribution shows a good fit especially when the lower limit of discrimination (LLD) was chosen as suggested by (Watson *et al.*, 1997).

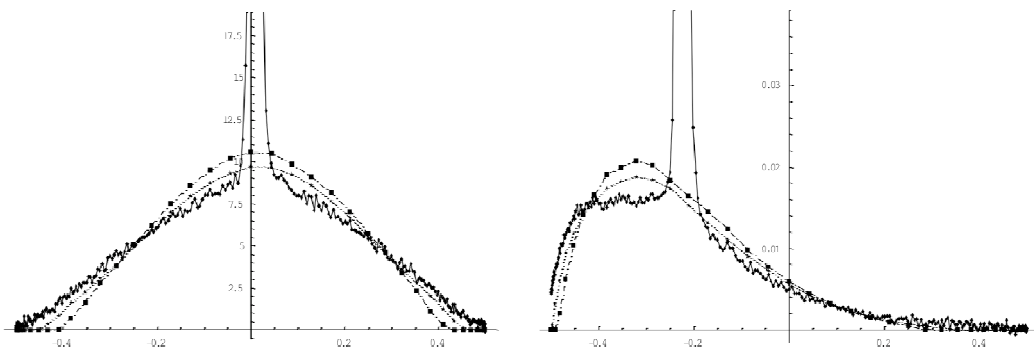


Figure 8. Profiles along radial direction for vertical view of the sinograms, line source at ~ 0 mm (left-hand side) and ~ 80 mm (right-hand side). Normalized measured data (solid lines between points), STIR scatter simulation with LLD at 350keV as during acquisition (dash-dot lines) and 320keV (Watson *et al*, 1997) (dotted lines). Figure from Tsoumpas *et al* (2004), reused with permission.

The effect of the SSS-based scatter correction on the quantification of relative changes in a small animal PET scanner was evaluated (Aguiar *et al*, 2006) where the SimSET Monte Carlo package was used to simulate the FDG uptake of a brain mouse with cerebral ischemia. The results showed that the SSS-based scatter corrections are sufficient without the need of scaling the single scatter estimate.

3.4 Parametric images

A realistic synthetic simulation setup for an FDG brain study was devised within STIR as described by (Tsoumpas *et al*, 2008). Dynamic projection data were calculated for six dynamic frames which were used to reconstruct and recover parametric information. Three main different approaches have been investigated: FBP-3DRP, OSEM-3D (indirect), and POSEM-3D (direct). The latter two have originally higher resolution but their post-filtered version matched the resolution of FBP-3DRP. For OSEM and FBP-3DRP a least squares plot (i.e. Patlak) was applied using STIR and two parametric images that correspond to the FDG uptake and free FDG tissue fraction are shown in the following figures. It is clear that the indirect results are much noisier than POSEM, even at convergence. However, the latter demands high computational power (at least for the Patlak case) preventing its practical use at the moment.

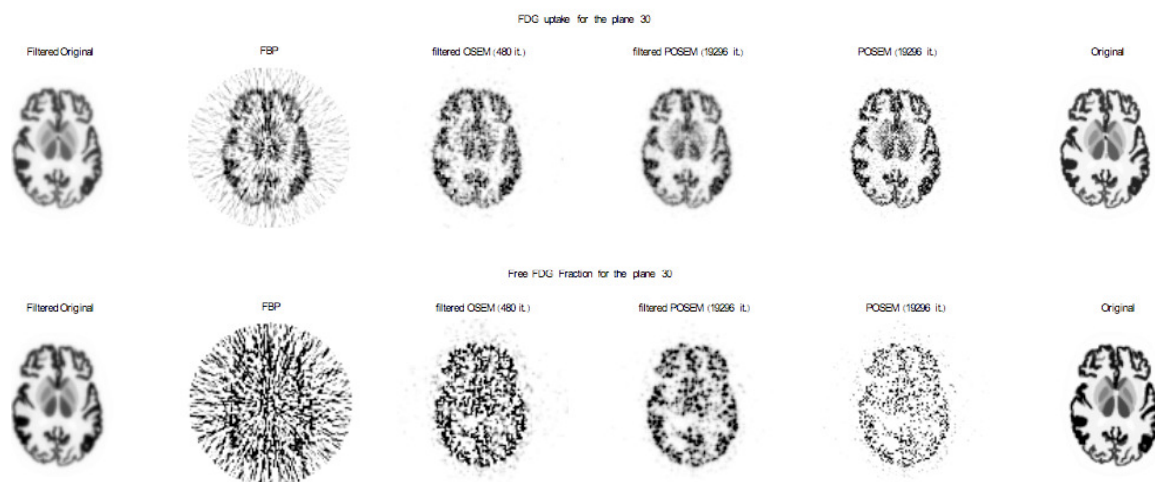


Figure 9 A transaxial plane of the phantom images for one noise realization for the FDG uptake (upper row) and FDG free fraction (lower row) as reconstructed with the different methods. Inverse grey scale was used ranging from 0 to the maximum of the original parametric planes. The last 96 POSEM iterations were performed with PMLEM.

4 Conclusion

In this paper, we have described a major update to STIR, a versatile library which can be used to study existing and / or new reconstruction algorithms for a wide variety of cylindrical scanners. The

use of object-oriented software has been advantageous for: (a) Comparison of analytic and iterative methods; (b) Development of new algorithms; (c) Adaptation and application of the developed reconstruction algorithms to data from various PET systems. The programming paradigm of STIR 2 allows the incremental refinements to the building blocks described in this paper with maximum reutilisation of code. This is achieved with a carefully designed framework for effective definition of standards that use the inheritance mechanism. This approach streamlines development and improves reliability.

STIR offers the prospect to compare different PET scanners using identical correction and reconstruction software. Unfortunately, present-day PET scanners use data formats which are proprietary and cannot be read using open source software. To solve this problem, STIR can be extended to use a closed-source external library from the manufacturer.

, Various developments by the research community have not been incorporated in STIR yet. Topics such as positron range modelling, optimised support for multi-processor systems, GPU reconstruction, are currently under consideration by STIR developers.

At present, STIR is restricted to PET data. However, many of its components would be directly applicable to other modalities with the most obvious candidate being SPECT (Bruyant, 2002). As the community of STIR developers grows, there is the hope that additional components for other modalities will be included expanding STIR towards the direction of multimodality imaging. In the mean time, this second major release provides the PET research community with additional tools and flexibility.

Acknowledgements

STIR is based on work performed by the PARAPET project, which was supported by the European Esprit LTR project PARAPET (EP23493) and the Swiss Federal Office for Education and Science under grant 96.193. PARAPET groups and members are listed on the STIR website.

Part of Dr Tsoumpas' work was funded by MRC (grant number G78/8306). Part of Dr Aguiar's work was funded by the following grants: : FPU (AP2003-1170), ISCIII (CD09/00291 and PS09/01206) and XUNTA DE GALICIA (10CSA918001PR). The authors wish to thank all the numerous people that have assisted throughout the years to make STIR freely available. We are particularly thankful to General Electric Healthcare (Dr T Spinks, A Foster, P Jasani), Medical Research Council UK (L Green), Imperial College London (Dr F Turkheimer), National and Technical University of Athens (Prof. K Nikita), University of Patras (Prof. N Palikarakis), CIBER-BNN, University of Barcelona (Prof. D Ros), CIBER-BNN, Hospital Clínic, Barcelona (Dr J Pavía), University of Paderborn (Dr S Lietsch). Dr Tsoumpas wishes to express his gratitude to Dr Loudos (Technological Educational Institute of Athens) for longstanding inspiring discussions.

We are also very grateful to C R Schmidlein and A Kirov (Memorial Sloan Kettering Cancer Center) for contributing their code to allow creation of wedge and box shapes in STIR 2.

References

- Aguiar P, Tsoumpas C, Pavia J, Falcon C, Cot A, Thielemans K and Ros D 2006 Assessment of scattered photons on the quantification of small animal PET studies *European Journal of Nuclear Medicine and Molecular Imaging* **33** S315
- Ahn S and Fessler J A 2003 Globally Convergent Image Reconstruction for Emission Tomography Using Relaxed Ordered Subsets Algorithms *IEEE Transactions on Medical Imaging* **22** 613-26
- Alenius S and Ruotsalainen U 1997 Bayesian image reconstruction for emission tomography based on median root prior *European Journal of Nuclear Medicine* **24** 258-65
- Angelis G I, Thielemans K, Tziortzi A C, Turkheimer F E and Tsoumpas C 2011 Convergence optimization of parametric MLEM reconstruction for estimation of Patlak plot parameters *Computerized Medical Imaging and Graphics* **35** 407-16
- Beisel T, Lietsch S and Thielemans K 2008 A method for OSEM PET reconstruction on parallel architectures using STIR *2008 IEEE Nuclear Science Symposium Conference Record* (Dresden, Germany: IEEE) pp 4161-8

- Bettinardi V, Pagani E, Gilardi M C, Alenius S, Thielemans K, Teras M and Fazio F 2002 Implementation and evaluation of a 3D One Step Late reconstruction algorithm for 3D Positron Emission Tomography studies using Median Root Prior geometries *IEEE Transactions on Medical Imaging* **9** 207-17
- Bryuant P P 2002 Analytic and iterative reconstruction algorithms in SPECT *Journal of Nuclear Medicine* **43** 1343-58
- Cho Z H, Chen C M and Lee S Y 1990 Incremental algorithm – A new fast backprojection scheme for parallel beam geometries *IEEE Transactions on Medical Imaging* **9** 207-17
- Dikaïos N, Spinks T J, Nikita K S and Thielemans K 2006 Evaluation of the scatter simulation algorithm implemented in STIR. In: *5th European Symposium in Biomedical Engineering*, Patras, Greece
- Dinelle K, Thielemans K, Tsoumpas C and Spinks T J 2004 An evaluation of various analytic reconstruction algorithms and implementations for 2D and 3D PET. In: *2004 IEEE Nuclear Science Symposium and Medical Imaging Conference*, ed. Seibert J A pp 4043-7
- Egger M, Joseph C and Morel C 1998 Incremental beamwise backprojection using geometrical symmetries for 3D PET reconstruction in a cylindrical scanner geometry *Physics in Medicine and Biology* **43** 3009-24
- Erdoğan H and Fessler J A 1999 Ordered subsets algorithms for transmission tomography *Physics in Medicine and Biology* **44** 2835-51
- Fessler J A 2004 Users guide for aspire 3D image reconstruction software (Ann Arbor: University of Michigan)
- Gamma E, Helm R, Johnson R and Vlissides J 1995 *Design patterns: elements of reusable object-oriented software*. Addison-Wesley
- Green P J 1990a Bayesian reconstructions from emission tomography data using a modified EM algorithm *IEEE Transactions on Medical Imaging* **9** 84-93
- Green P J 1990b On use of the EM algorithm for penalized likelihood estimation *Royal Statistical Society* **52** 443-52
- Hudson H and Larkin R 1994 Accelerated image reconstruction using ordered subsets of projection data. *IEEE Transactions on Medical Imaging* **13** 601-9
- Jacobson M, et al 2000 Enhanced 3D PET OSEM reconstruction using inter-update Metz filtering *Physics in Medicine and Biology* **45** 2417-39
- Johnson C A and Sofer A 1999 A data-parallel algorithm for iterative tomographic image reconstruction *The Seventh Symposium on the Frontiers of Massively Parallel Computation* (Annapolis: IEEE) pp 126-36
- Kinahan P E and Rogers J G 1989 Analytic 3D image reconstruction using all detected events *IEEE Transactions on Nuclear Science* **36** 964-8
- Labbé C, et al 1999a An object-oriented library for 3D PET reconstruction using parallel computing. In: *Proc. of the Bildverarbeitung für die Medizin*
- Labbé C, Thielemans K, Zaidi H and Morel C 1999b An object-oriented library incorporating efficient projector / back projection operators for volume reconstruction in 3D PET. In: *Proc. 1999 Int. Meeting on Fully Three-Dimensional Image Reconstruction in Radiology and Nuclear Medicine*, pp 137-40
- Lewellen T K, Harrison R L and Vannoy S 1998 "The SimSET program" in: Monte Carlo calculations in nuclear medicine *Applications in diagnostic imaging*, ed. M Ljungberg, Strand S-E, King M A (Bristol: Institute of Physics Publishing) pp 77-92
- Loudos G K, Papadimitroulas P, Zotos P, Tsougos I and Georgoulas P 2010 Development and evaluation of QSPECT open-source software for the iterative reconstruction of SPECT images *Nuclear Medicine Communications* **31** 558-66
- Matthews J, Bailey D L, Price P and Cunningham V J 1997 The direct calculation of parametric images from dynamic PET data using maximum-likelihood iterative reconstruction *Physics in Medicine and Biology* **42** 1155-73
- Mustafovic S and Thielemans K 2004 Object dependency of resolution in reconstruction algorithms with interiteration filtering applied to PET data *IEEE Transactions on Medical Imaging* **23** 433-46
- Parra L and Barrett H H 1998 List-mode likelihood: EM algorithm and image quality estimation demonstrated on 2-D PET *IEEE Transactions on Medical Imaging* **17** 228-35
- Patlak C S, Blasberg R G and Fenstermacher J D 1983 Graphical evaluation of blood-to-brain transfer constant from multiple-time uptake data *Journal of Cerebral Blood Flow & Metabolism* **3** 1-7
- Pedemonte S, Bousse A, Erlandsson K, Modat M, Arridge S, Hutton B F, and Ourselin S 2010 GPU accelerated rotation-based emission tomography reconstruction. In: *2010 IEEE Nuclear Science Symposium and Medical Imaging Conference* pp. 2657-61
- Poenisch F, Enghardt W and Lauckner K 2003 Attenuation and scatter correction for in-beam positron emission tomography monitoring of tumour irradiations with heavy ions. *Physics in Medicine and Biology* **48** 2419-36
- Polycarpou I, Thielemans K, Manjeshwar R, Aguiar P, Marsden P K and Tsoumpas C 2011 Comparative evaluation of scatter correction in 3D PET using different scatter-level approximations *Annals of Nuclear Medicine* **25** 643-9.
- Qi J Y and Leahy R M 2006 Iterative reconstruction techniques in emission computed tomography *Physics in Medicine and Biology* **51** R541-R78
- Rezvani N, Aruliah D, Jackson K, Moseley D and Siewerdsen J 2007 SU-FF-i-16: OSCaR: An Open-Source Cone-Beam CT reconstruction tool for imaging research. In: *Medical Physics: American Association of Physicists in Medicine*, p 2341
- Robb R 2001 The Biomedical Imaging Resource at Mayo Clinic. *IEEE Transactions on Medical Imaging* **20** 854-67
- Robb R, Hanson D, Karwoski R, Larson A, Workman E and Stacy M 1989 ANALYZE: A comprehensive operator-interactive software package for multidimensional medical image display and analysis *Computerized Medical Imaging and Graphics* **13** 433-54
- Scheins J J, Herzog H and Shah N J 2011 Fully-3D PET Image Reconstruction Using Scanner-Independent, Adaptive Projection Data and Highly Rotation-Symmetric Voxel Assemblies *IEEE Transactions on Medical Imaging* **30** 879-92
- Shepp L A and Vardi Y 1982 Maximum likelihood reconstruction for emission tomography *IEEE Transactions on Medical Imaging* **1** 113-22
- Siddon R L 1985 Fast calculation of the exact radiological path for a three-dimensional CT array *Medical Physics* **12** 252-5

- Silverman B W, Jones M C, Wilson J D and Nychka D W 1990 A smoothed EM approach to indirect estimation problems, with particular reference to stereology and emission tomography *Journal of the Royal Statistical Society. Series B (Methodological)* **52** 271-324
- Thielemans K, Mustafovic S and Tsoumpas C 2006 STIR: Software for tomographic image reconstruction release 2. In: *2006 IEEE Nuclear Science Symposium and Medical Imaging Conference*, San Diego, CA pp 2174-6
- Thielemans K, Manjeshwar R M, Tsoumpas C and Jansen F P 2007 A new algorithm for scaling of PET scatter estimates using all coincidence events. In: *2007 IEEE Nuclear Science Symposium and Medical Imaging Conference*, pp 3586-90
- Todd-Pokropek A, Craddock T D and Deconinck F 1992 A file format for the exchange of nuclear medicine image data *Nuclear Medicine Communications* **13** 673-99
- Tsoumpas C, Aguiar P, Nikita K S, Ros D and Thielemans K 2004 Evaluation of the single scatter simulation algorithm implemented in the STIR library. In: *2004 IEEE Nuclear Science Symposium and Medical Imaging Conference*, ed. Seibert J A (New York: IEEE) pp 3361-5
- Tsoumpas C, Turkheimer F E and Thielemans K 2007 Convergence properties of direct parametric estimation of linear models in dynamic PET. In: *2007 IEEE Nuclear Science Symposium and Medical Imaging Conference*, ed. Yu B (New York: IEEE) pp 3034-7
- Tsoumpas C, Turkheimer F E and Thielemans K 2008 Study of direct and indirect parametric estimation methods of linear models in dynamic positron emission tomography *Medical Physics* **35** 1299-309
- Tsoumpas C, Buerger C, King A P, Mollet P, Keereman V, Vandenberghe S, Schulz V, Schleyer P J, Schaeffter T and Marsden P K 2011 Fast generation of 4D PET-MR data from real dynamic MR acquisitions. *Physics in Medicine and Biology* **56** 6597-6613
- Wang G B and Qi J Y 2010 Acceleration of the direct reconstruction of linear parametric images using nested algorithms *Physics in Medicine and Biology* **55** 1505-17
- Watson C C 2000 New, faster, image-based scatter correction for 3D PET *IEEE Transactions on Nuclear Science* **47** 1587-94
- Watson C C, Newport D and Casey M E 1996 A single-scatter simulation technique for scatter correction in 3D PET *Fully Three-Dimensional Image Reconstruction in Radiology and Nuclear Medicine*, ed Grangeat P and Amans J (Amsterdam, Netherlands: Kluwer Academic) pp 255-68
- Watson C C, Newport D, Casey M E, Beanlands R S, Schmand M and deKemp R A 1997 A technique for measuring the energy response of a PET tomograph using a compact scattering source *IEEE Transactions on Nuclear Science* **44** 2500-8
- Werling A, Bublitz O, Doll J, Adam L E and Brix G 2002 Fast implementation of the single scatter simulation algorithm and its use in iterative image reconstruction of PET data. *Physics in Medicine and Biology* **47** 2947-60