STIR

Software for Tomographic Image Reconstruction

http://stir.sourceforge.net
https://github.com/UCL/STIR

Kris Thielemans
University College London
Algorithms And Software Consulting Ltd

Charalampos Tsoumpas
University of Leeds

STIR objectives

- Research enabler

- Offline image reconstruction and data manipulation

- Portable to any system with a capable C++ compiler
  - GNU C++, MS Visual Studio, Clang, Intel C++
  - Linux, Windows, MacOS, Solaris, …

- Open Source
  (L)GPL now, Apache 2.0 soon

- Use Sustainable Software Development techniques
  - For software quality
  - For training the next generation of researchers

Overview

- Using STIR
- Extending STIR
- MATLAB/Python interface
- Challenges

Overview

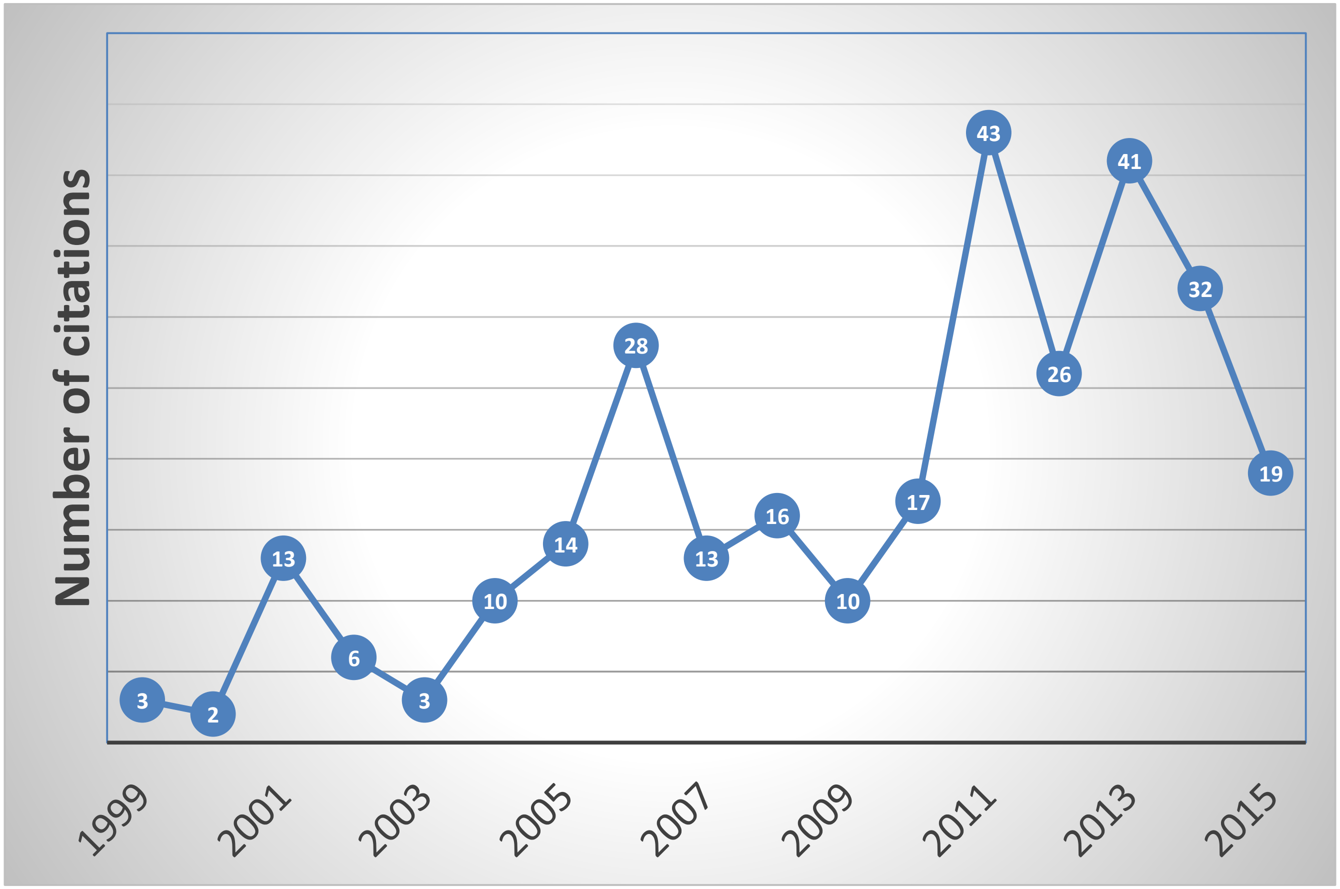
- Using STIR
  - Overview of capabilities
  - Example results
  - User perspective
  - Missing features
- Extending STIR
- MATLAB/Python interface
- Challenges

Capabilities

- PET and SPECT
- Quantitative
  - PET scatter, normalisation and randoms estimation
- Analytic and iterative 3D reconstruction algorithms
  - FBP-3DRP, FORE, OSEM, OS-MAP-OSL, OS-SPS, list-mode EM and SPS
- Pharmacokinetic modelling
  - linear models only
  - indirect and direct parametric reconstruction
- Motion correction
  - post-reconstruction and MCIR for gated data
  - LOR rebinning for rigid motion
  - no motion estimation
- Various utilities
  - data manipulation, ROI values, analytic image generation ...

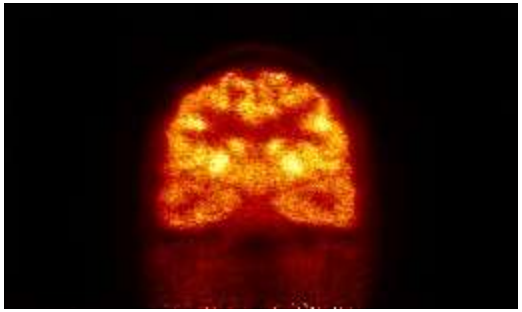
User statistics

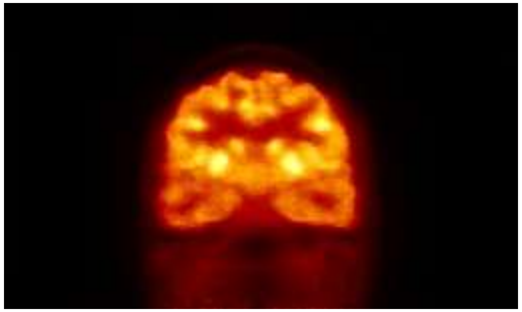
~280 subscribers to stir-users@lists.sf.net



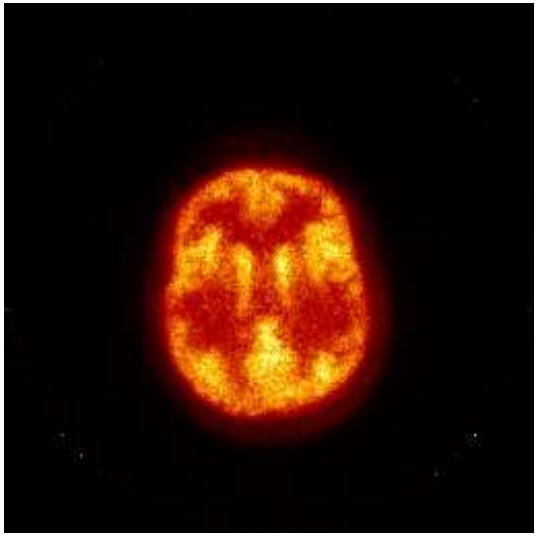
OSEM & OSL-MAP reconstruction for brain PET

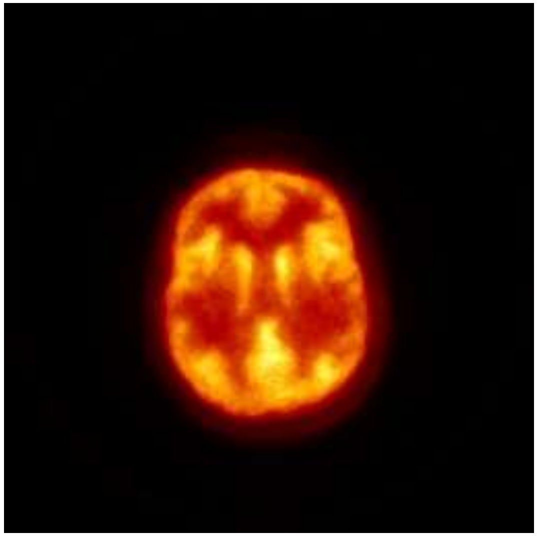
Patient data – Reconstructed Images





Coronal Image



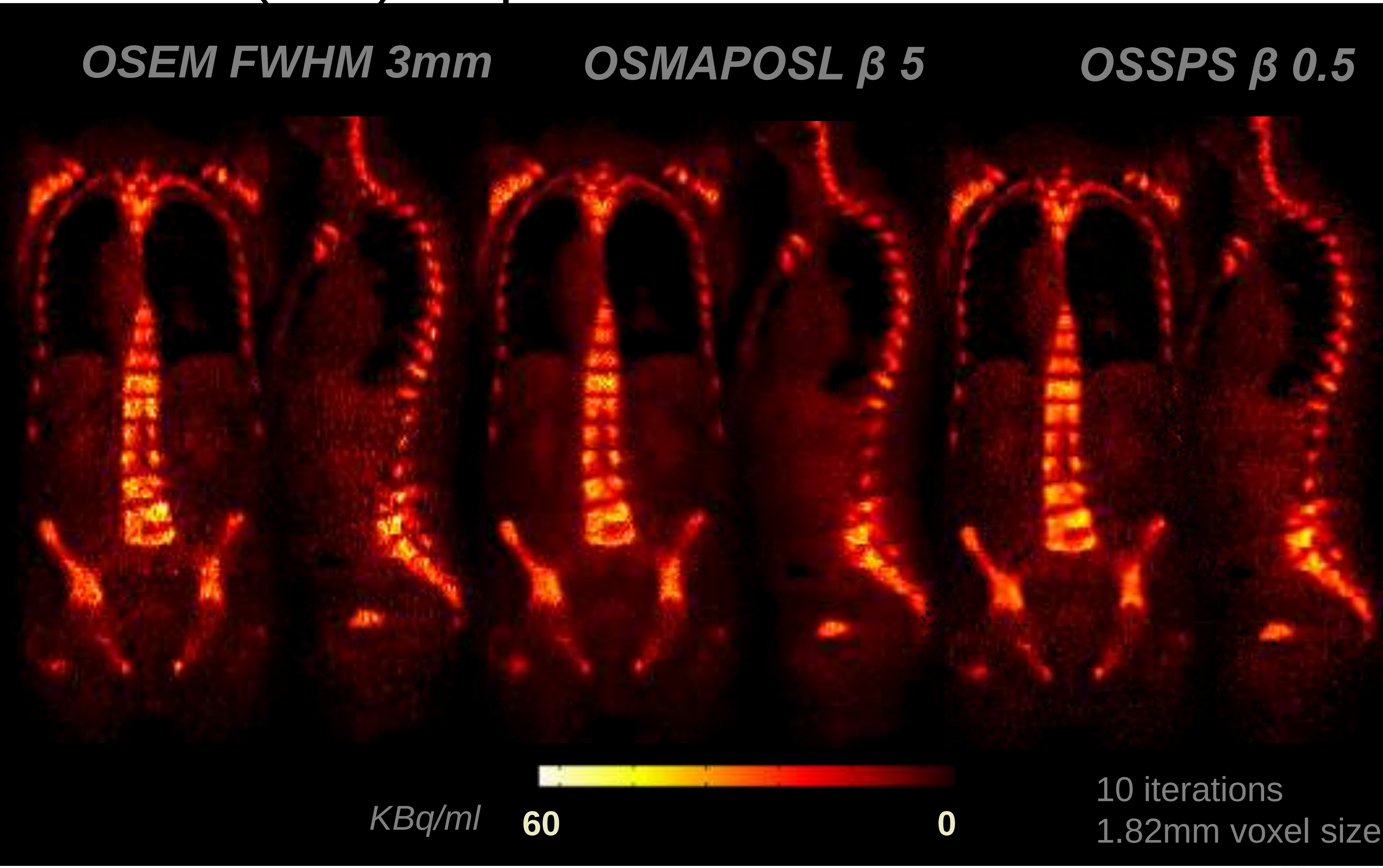


Transverse Image

Coronal Image

Courtesy of Liliana Caldeira

Patient (F18) acquired on GE PET/CT

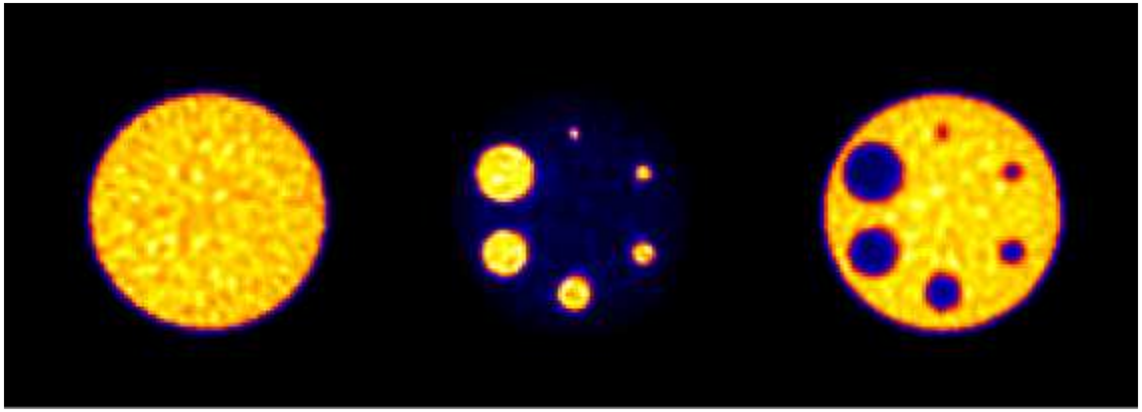


Grecchi et al (2013) IEEE NSS MIC

SPECT reconstruction

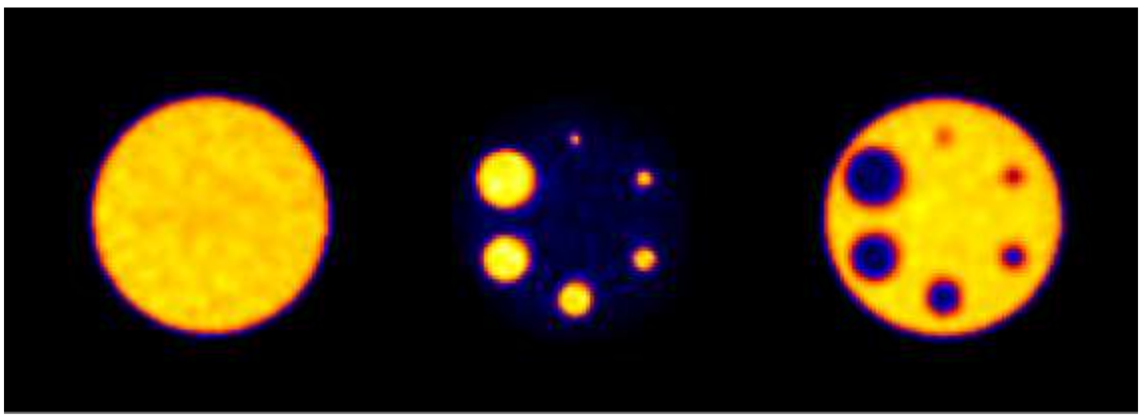
SIMULATED DATA

OSMAPOSL it 80



C_v = 6.8%

OSSPS it 80



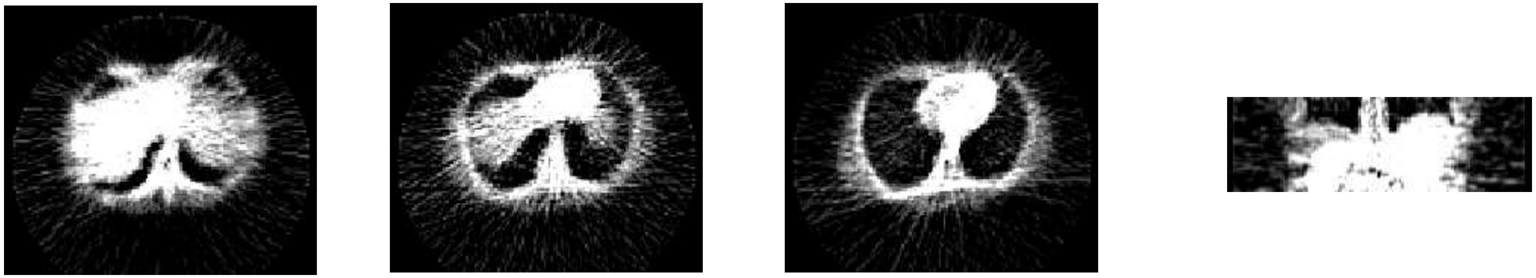
C_v = 2.5%

C_v = Coefficient of variation

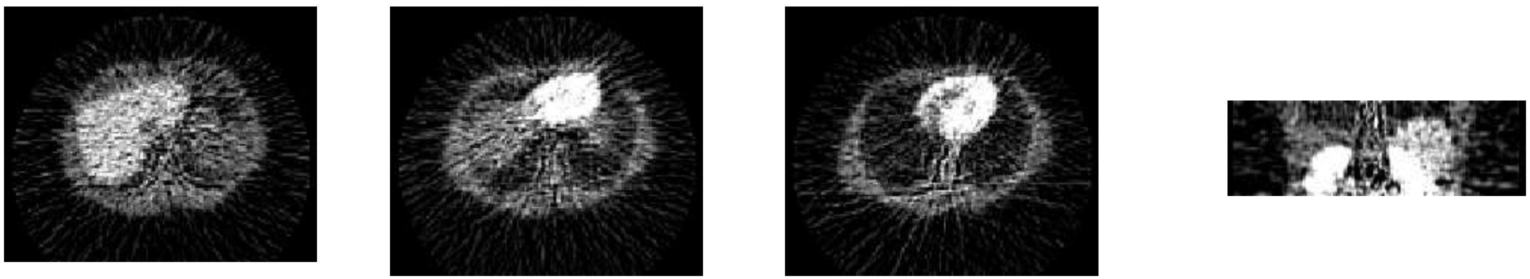
Marti Fuster et al (2013) Med Phys

Scatter Correction Example

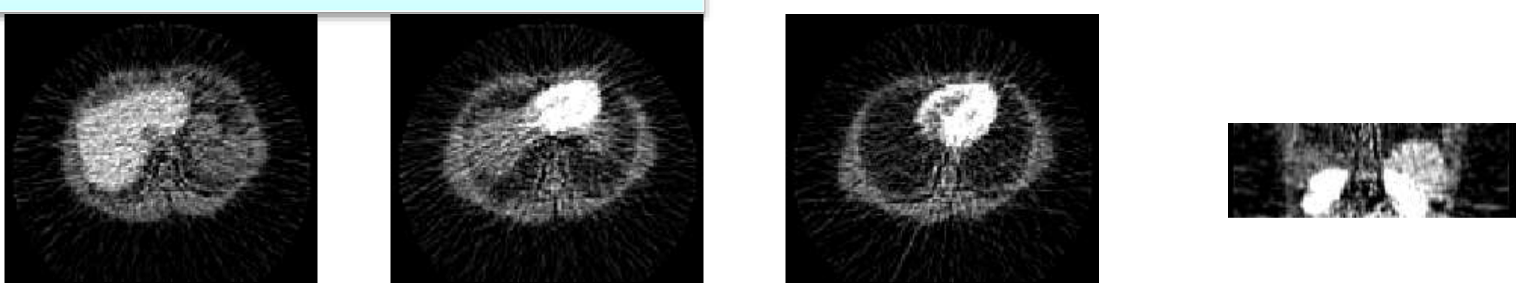
Non corrected



Corrected with SSS

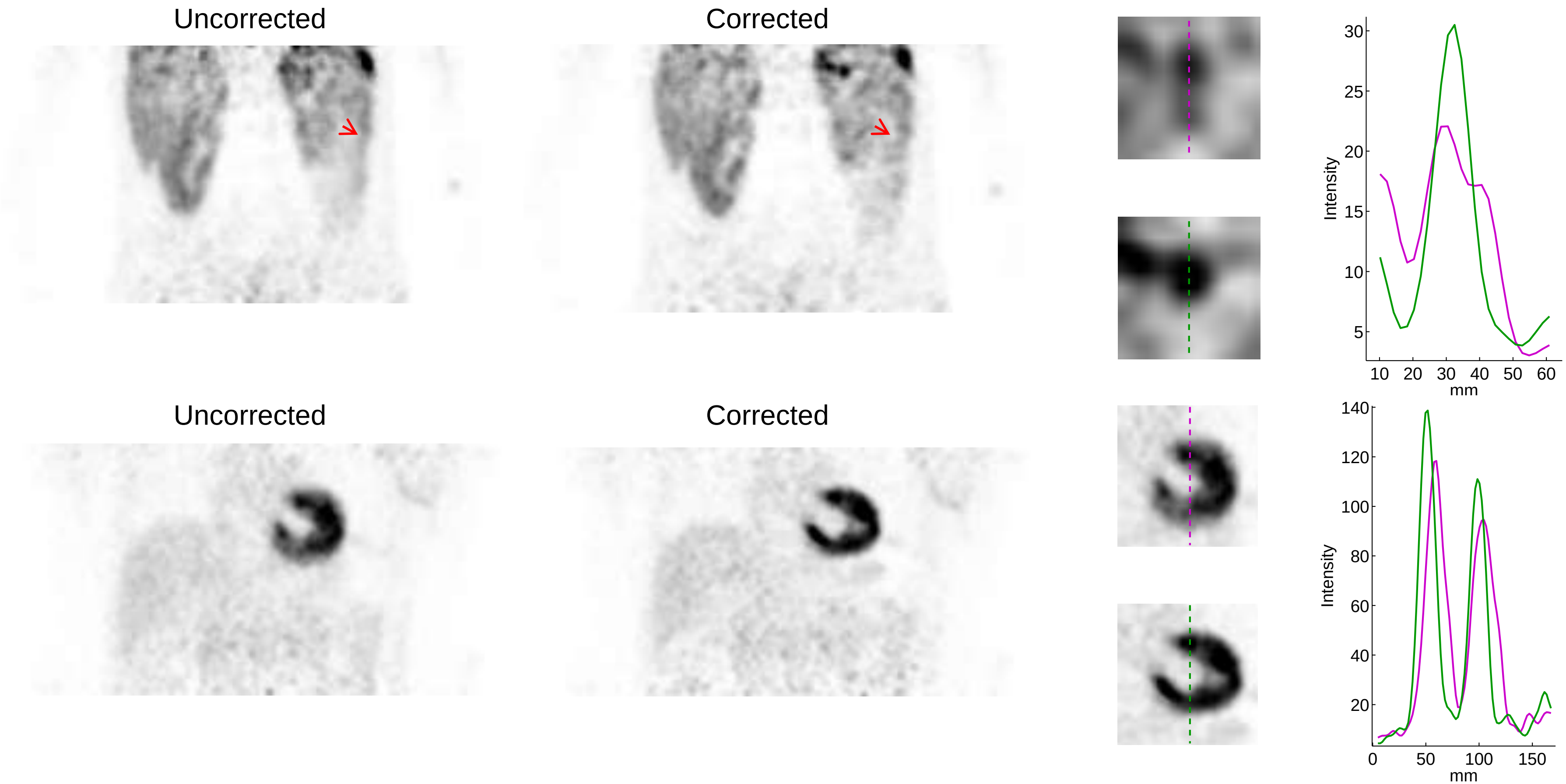


Without Scatter



Polycarpou *et al* (2011) *Ann Nucl Med*

Motion-Compensated Image Reconstruction



Manber et al, JNM 2015, *Practical PET Respiratory Motion Correction in Clinical PET/MR*

Missing features

- PET
  - Reading raw data from GE, Philips (Siemens mostly ok)
  - Non-cylindrical scanners
  - TOF (WIP)
- SPECT
  - Dicom sinogram import
  - Non-parallel hole collimators
  - Scatter
- Extra reconstruction options
  - More optimisation algorithms
  - More priors (WIP)
- Closer connection with SimSET/GATE (WIP)
- GPU

Current user perspective

- Command line utilities

```
OSSPS parameterfile
```

- Documentation
  - PDFs (Overview, detail)
  - Wiki
  - Example parameter files
  - No easy place to start

Run-time parameter selection

```
OSSPSParameters :=
objective function type:= PoissonLogLikelihoodWithLinearModelForMeanAndProjData
PoissonLogLikelihoodWithLinearModelForMeanAndProjData Parameters:=
  input file := test.hs
  projector pair type := Matrix
       Projector Pair Using Matrix Parameters :=
            Matrix type := Ray Tracing
                  Ray tracing matrix parameters :=
                  End Ray tracing matrix parameters :=
       End Projector Pair Using Matrix Parameters :=
  Bin Normalisation type := From ProjData
      Bin Normalisation From ProjData :=
            normalisation projdata filename:= norm.hs
      End Bin Normalisation From ProjData:=
  prior type := quadratic
       Quadratic Prior Parameters:=
              penalisation factor := 1
      End Quadratic Prior Parameters:=
end PoissonLogLikelihoodWithLinearModelForMeanAndProjData Parameters:=
initial estimate:= some_image
output filename prefix := test
number of subsets:= 12
number of subiterations:= 24
relaxation parameter := 1
relaxation gamma:=.1
END :=
```

Overview

- Using STIR
- Extending STIR
  - General developer's perspective
  - Example class hierarchies
- MATLAB/Python interface
- Challenges

Developer's perspective

- Object-oriented (C++) and modular
- Documented (doxygen)
- Test framework
- Extendable
  - Mechanism for extending library such that current STIR applications can use your module (e.g. projector) after recompilation
  - Mechanism for writing new applications using (original or extended) library

Code statistics

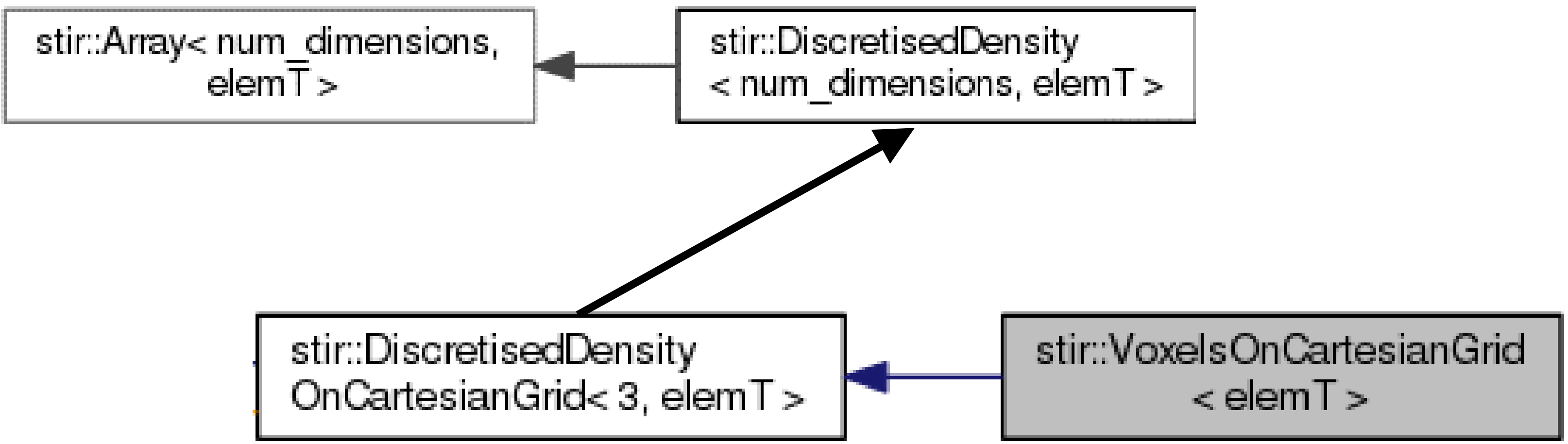
- Physical Source Lines of Code (SLOC)
  = **105,886**
- Total Number of Source Code Files
  = **836**
- Development Effort Estimate
  = **26.74 Person-Years**
  (Basic COCOMO model)

generated using David A. Wheeler's 'SLOCCount'

Images

Discretised representations of a “density”, e.g.

$$f(\hat{x}) = \sum_{ijk} \lambda_{ijk} b_{ijk}(\hat{x})$$

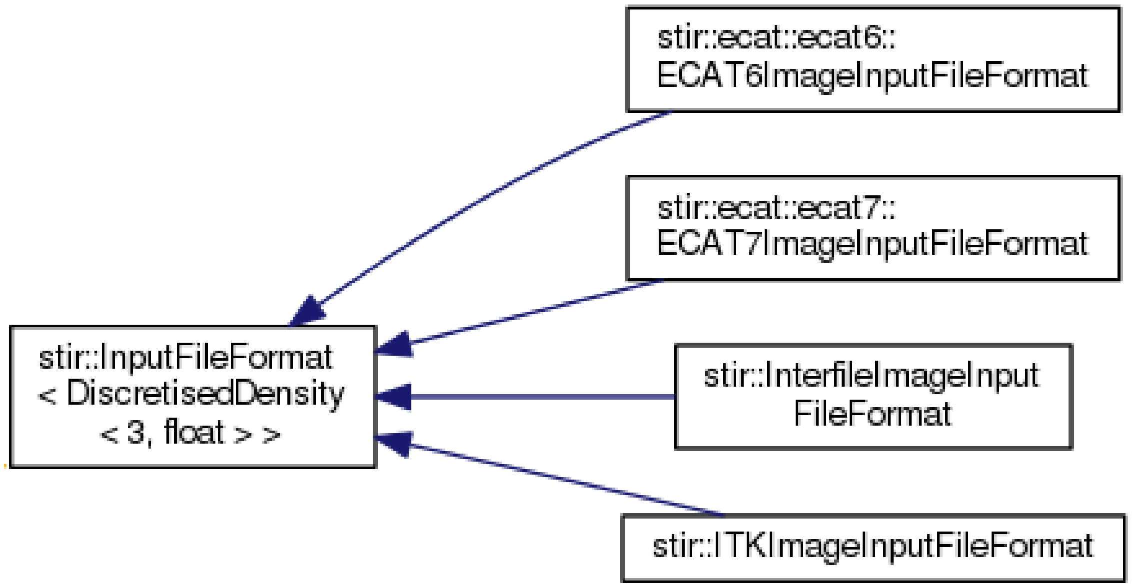



```
float sx = image.get_voxel_size().x();

auto voxel_location =
 image.get_physical_coordinates_for_indices(make_coord(i,j,k));

image[i][j][k] = 4;
float value = image[make_coord(i,j,k)];
```

IO: pluggable factories

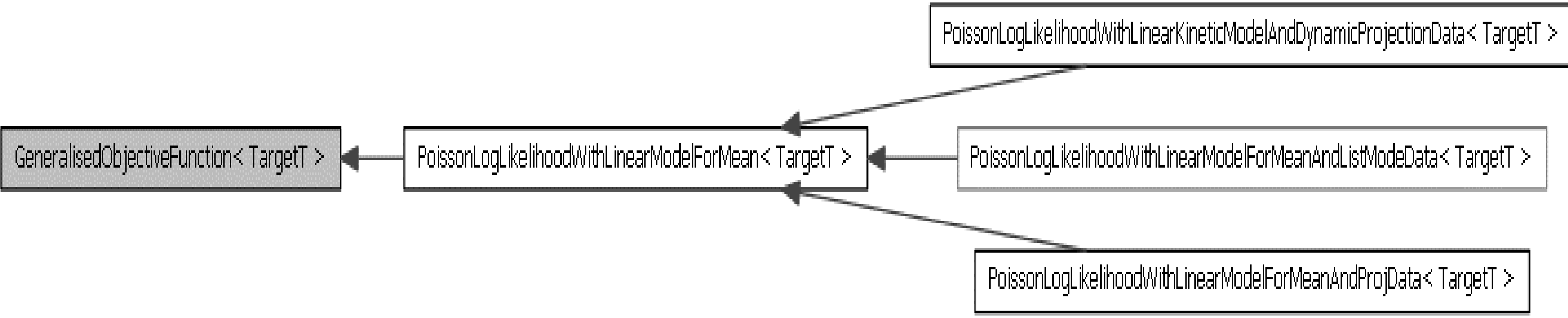



```
typedef DiscretisedDensity<3,float> ImageType;

auto density_sptr(read_from_file<ImageType>(filename));
```

Similar for dynamic data, list mode data, …

Objective functions

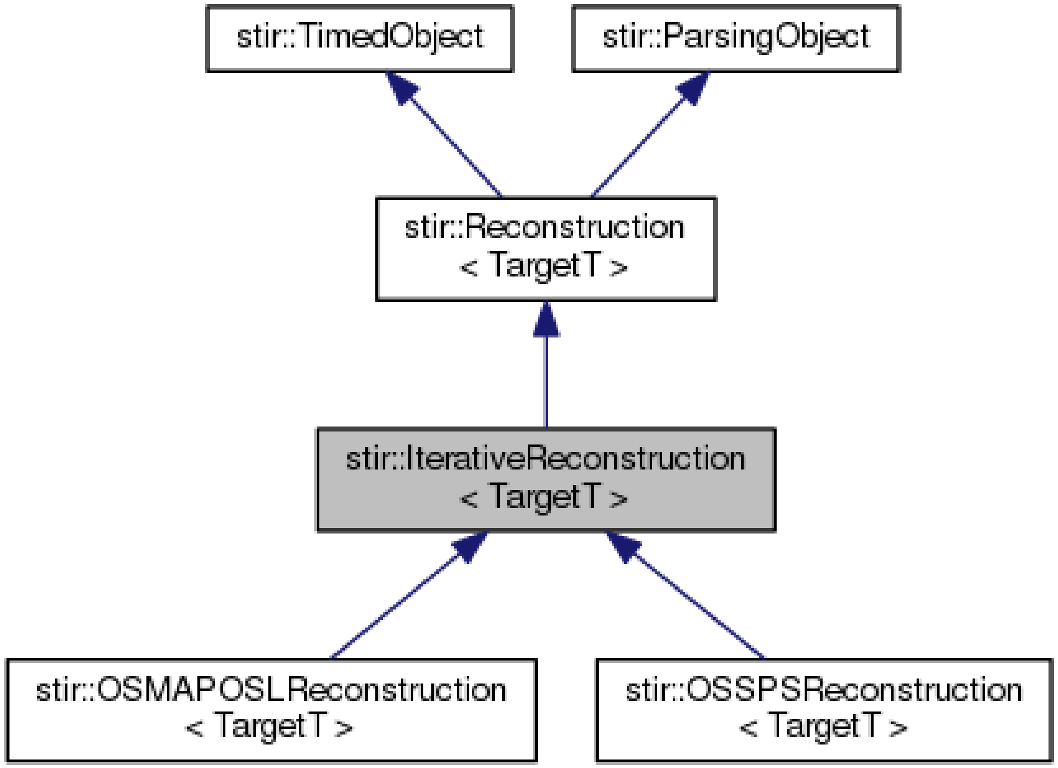



```
double value =
      objf.compute_objective_function (image, subset_num);

objf.compute_sub_gradient (gradient, image, subset_num);
```

STIR

Reconstruction algorithms



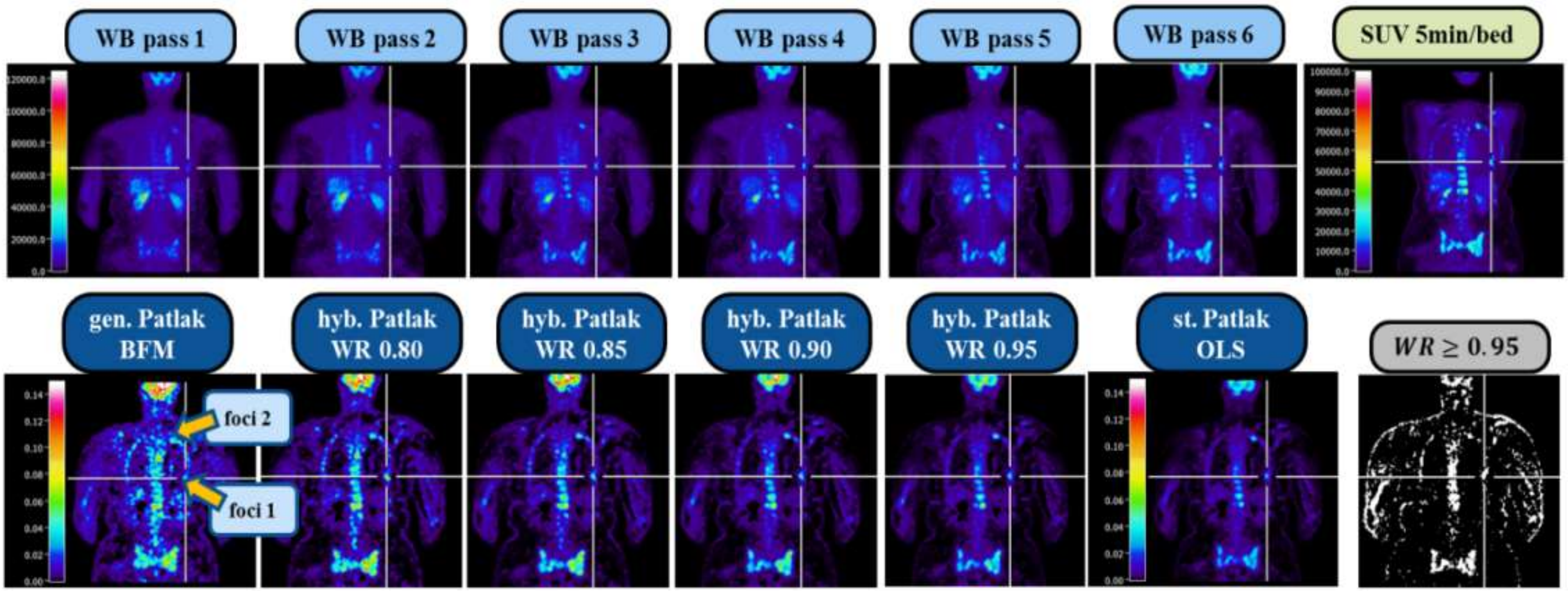

```
OSMAPOSLReconstruction<ImageType> recon(parameter_file);
recon.set_num_subiterations(5);
// reconstruct from initial image
recon.reconstruct(image);
```

STIR

Generalized whole-body Patlak PET

generalized Patlak model equation

$$\frac{C(t_k)}{C_P(t_k)} = K_i \frac{\int_0^{t_k} e^{-k_{loss}(t_k-\tau)} C_P(\tau) d\tau}{C_P(t_k)} + V, \qquad t_k > t^*,$$



Karakatsanis *et al* (2015) Phys Med Biol

Overview

- Using STIR
- Extending STIR
- MATLAB/Python interface
  - How?
  - Examples
- Challenges

STIR and MATLAB/Python

- Interface constructed via SWIG

  SWIG Simplified Wrapper and Interface Generator

  - Parses “interface” text file and C++ headers
  - Generates MATLAB/Python/C++
  - Compile to generate library

- Object-oriented MATLAB/Python
  (close to C++, but no templates, pointers etc)

- Work-in-Progress
  – SWIG-MATLAB is under development.

Python: objective function computation

```
## initialise reconstruction object via a parameter file
recon=stir.OSMAPOSLReconstruction3DFloat('recon_demo_OSEM.par');

## construct image related to the data to reconstruct
projdata=stir.ProjData.read_from_file('input_sinogram.hs');
target=stir.FloatVoxelsOnCartesianGrid(projdata.get_proj_data_info());

## set-up objective function
recon.set_up(target);
% get corresponding objective function
poissonobj=recon.get_objective_function();

## compute gradient of objective function
# put some data in the image
target.fill(1);
# create an image to store the gradient
gradient=target.get_empty_copy();
poissonobj.compute_sub_gradient(gradient,target)

## display
gradientdata=stirextra.to_numpy(gradient);
pylab.figure();
pylab.imshow(gradientdata[10,:,:])
pylab.show()
```

MATLAB: objective function computation

```
%% initialise reconstruction object via a parameter file
recon=stir.OSMAPOSLReconstruction3DFloat('recon_demo_OSEM.par');

%% construct image related to the data to reconstruct
projdata=stir.ProjData.read_from_file('input_sinogram.hs');
target=stir.FloatVoxelsOnCartesianGrid(projdata.get_proj_data_info());

%% set-up objective function
recon.set_up(target);
% get corresponding objective function
poissonobj=recon.get_objective_function();

%% compute gradient of objective function
% put some data in the image
target.fill(1);
% create an image to store the gradient
gradient=target.get_empty_copy();
poissonobj.compute_sub_gradient(gradient,target)

%% display
gradientdata=gradient.to_matlab();
figure;
imshow(gradientdata(:,:,10),[])
```

Overview

- Using STIR
- Extending STIR
- MATLAB/Python interface
- Challenges

Challenges (I)

- Lots of functionality
  - Good software design is crucial
  - Modular/flexible involves overhead

- Large code-base
  - Good software design is crucial
  - Not enough documentation
  - Too much documentation

- Rapid development in software/hardware

Challenges (II)

- Manage user expectations

- Foster user involvement
  - Lots of questions on the mailing list
  - Current group of "developers" is small
  - Hopefully will increase with Python/MATLAB capabilities

- Needs time investment

Main publication:

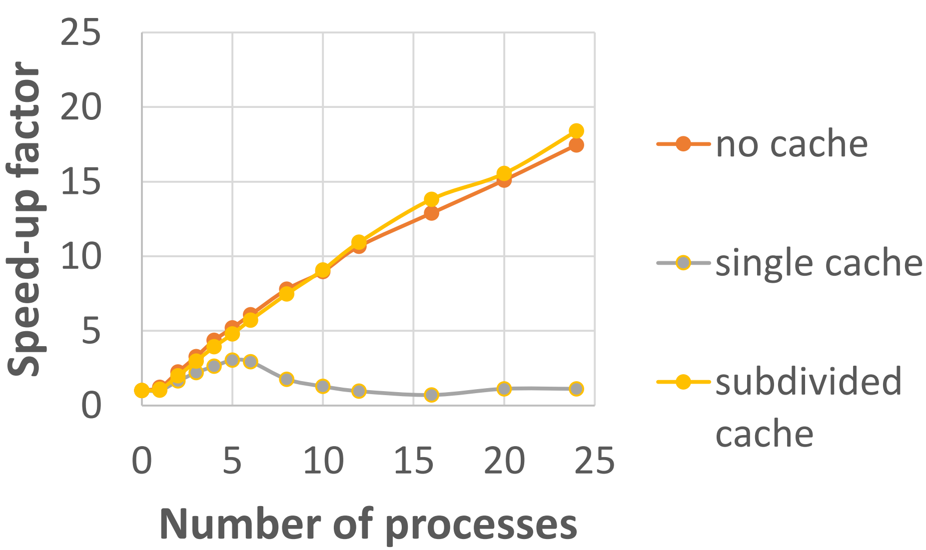
Thielemans, Tsoumpas, *et al* (2012) STIR: Software for Tomographic Image Reconstruction Release 2, *Physics in Medicine and Biology*, 57(4):867-83.

STIR

Parallelisation

- Cluster: MPI
- Multi-threading: OpenMP

Dual-Opteron system



Wall-clock times per MLEM iteration

Siemens mMR data (span 11)

No threading	315s
20 threads	20s

Future contributions

- 4D Generalised Patlak for multi-bed position data
  Nicolas Karakatsanis, Arman Rahmim, Habib Zaidi
- List-mode reconstruction fixes
  Nikos Efthimiou, Charalampos Tsoumpas
- TOF
  Nikos Efthimiou, Charalampos Tsoumpas
- Non-cylindrical scanners => cylindrical
  Jannis Fischer
- Support for GE PET-MR