**Introduction**

This documents provides short description of the contributed software that can be included in the next release of the STIR user’s manual or other relevant documentation.

**Motion Correction**

Motion correction has become an important task in PET imaging. Amongst several approaches to motion-correct PET data, two are the most common: reconstruct-transform-average (RTA) [Kle96] and motion-compensated image reconstruction (MCIR) [Li06]. In RTA, separate images are reconstructed for each motion “state” (or “frame”), which are then transformed to one reference frame and averaged to produce a motion-corrected image. In MCIR, the projection data from all frames are reconstructed together by including the motion information into the reconstruction system matrix, so that a motion-corrected image is produced directly. In both cases it is assumed that an accurate description of the motion is available.

In STIR 2.3, the motion compensation can happen either before or during reconstruction. Two research papers have validated the implementation of RTA and MCIR in STIR [Pol12], [Tso13]. As these are based on OSMAPOSL their names are RTA-OSMAPOSL and MCIR-OSMAPOSL. The implementation has been validated only for the additive median-root-prior (MRP), but it is compatible with any either MAP reconstruction within STIR. Furthermore, the implementation can also work for OSSPS but the current version of MCIR-OSSPS needs further debugging.

Example of the Median Root Prior

RTA-OS-MRP-OSL: Equations (1-2) describe the separate reconstruction of each respiratory gate. Equation (3) describes the transformation step that is followed after reconstruction of all gates.

(1)

Where:[[1]](#footnote-1)

(2)

(3)

While the corresponding MCIR-OS-MRP-OSL equation (4) is:

(4)

Notations:

* ***Λv(s)***is the estimated radioactivity at voxel ***v*** and subiteration ***s***
* ***Ybg*** is the number of measured coincident photons of each detector pair (bin) ***b*** that belongs to the ***l****th* subset ***S*** and gate ***g***
* ***S****l* corresponds to the ***l****th* subset of the projection space, which is divided into ***L*** total subsets
* ***s***is the sub-iteration number. A set of ***L***sub-iterations comprises a full iteration
* ***Pbv*** is the system projection matrix
* ***Ŵ***, ***Ŵ–1*** represent the forward / backward warping operations of the image that move the activity from one location (e.g. ***v΄***) to voxel ***v*** using the motion fields and linear interpolation
* ***E*** is the ‘potential’ function
* ***Mv*** corresponds to the median 3×3×3 mask width of neighbourhood voxels centred at voxel ***v***
* ***G***is the total number of gates
* ***β****,* ***βg***are the penalisation factors for MCIR and RTA, respectively. Note that ***β*** = ***G* × *βg***, but for simplicity all cases are displayed with respect to ***βg***
* ***Abg***and ***Bbg*** are the attenuation coefficient and background (e.g. scatter) term for each bin and gate, respectively

Both RTA-OS-MAP-OSL and MCIR-OS-MAP-OSL are using the same routines for warping each respiratory gate to the reference gate. However, MCIR requires two motion fields: the forward (***Ŵ***) motion fields and the backward (***Ŵ–1***) motion fields (i.e. the same as used in RTA). The forward operator practically warps all frames to the reference frame, while the backward operator ‘unwarps’ the reference frame to each frame so that they can be compared with the corresponding projection gates in the numerator. It is important that the forward and backward projectors are consistent with each other, if not the result might not converge to a solution. In future we plan to include an alternative option for the backward motion fields to be the transpose matrix of the forward motion fields.

**Warning**:

The current implementation has been tested for respiratory gating, where each gate has the same number of counts. If this is not the case, or if there is for instance radioactive decay between frames in dynamic PET, the implementation likely needs minor modifications. For instance, the different gate duration could be accounted for by normalising for the time duration [Rah08].

**A. Data Preparation**

Multiple Files (one for each position): Emission sinogram, Multiplicative corrections (attenuation, normalisation), additive corrections (scatter, randoms), motion vectors, and gate definitions filename.

Sinograms for each position:

These have a standard suffix \_g#, e.g. *sinogram\_g1.hs* is the header of the position 1. To read the sinogram you will need also have set the corresponding definition file for the positions, e.g. *sinogram.gdef*

Images:

These have a similar suffix \_g#, e.g. *image\_g1.hv* is the header of the position 1. To read the image you will need also to set the corresponding definition file for the positions, e.g. *image.gdef*

Motion Vectors:

These have a standard suffix \_g#d%, e.g. *motion\_g1d1.hv* is the header of the motion corresponding to the position 1 and the first direction (i.e. axial according to STIR coordinate system); *d2* corresponds to the vertical axis direction and *d3* to the horizontal axis direction. To read the files you will need also to have set the corresponding definition file for the positions, e.g. *motion.gdef*. The image has exactly the same characteristics as the reconstructed PET image. This currently creates a minor burden as the final voxel sizes and the number of voxels have to predefined on the motion vector images. We hope to change this in future releases.

Coordinate system for motion vectors:

Although motion is designed for general motion information such as rigid motion, affine etc, currently only MotionVectors on a Cartesian grid are implemented. Information is stored in millimeter. The origin of the motion vectors system is located in the same place as the coordinate system The coordinate axes for Cartesian grids are chosen as follows.

***mx***: motion on the horizontal axis, pointing left when looking from the bed into the gantry

***my***: motion on the vertical axis, pointing upwards

***mz***: motion on the scanner axis, pointing from the bed towards the gantry

See figure below for an illustration of how the warping operator works in a 3D grid. As it can be noticed, the motion vector arrow is located at the centre of the voxel and the direction of the motion field points opposite than the standard grid. This is also known as ‘pull-interpolator’ warping operation.



**B. Motion Correction**

B.1) RTA

**warp\_and\_accumulate\_gated\_images**

<output filename> <filename prefix> <motion vectors prefix>

filename prefix: The images need to be placed in the same reference position

motion vectors prefix: To read motion vector images

The procedure involves a warping operation by the use of linear interpolation (based on B-Splines). Note that the current release assumes no weighting over the respiratory positions, but this could be manually included if the images are scaled according to duration of each position prior to the correction.

B.2) MCIR

The procedure involves warping operation by the use of linear interpolation (based on B-Splines) in each forward/backward step. Note that the current release assumes no weighting over the respiratory positions but this could be manually included in the multiplicative sinogram.

As the main reconstruction algorithm is exactly the same as for ordinary image reconstruction, we can use the normal OSMAPOSL program, but with a different objective function. An example is given below.

**OSMAPOSL OSMAPOSL\_with\_motion\_correction.par**

With an example for **OSMAPOSL\_with\_motion\_correction.par:**

OSMAPOSLParameters :=

objective function type:= PoissonLogLikelihoodWithLinearModelForMeanAndGatedProjDataWithMotion

PoissonLogLikelihoodWithLinearModelForMeanAndGatedProjDataWithMotion Parameters:=

input filename := INPUT

projector pair type := Matrix

Projector Pair Using Matrix Parameters :=

Matrix type := Ray Tracing

Ray Tracing Matrix Parameters:=

; use a slightly better approximation than simple ray tracing

number of rays in tangential direction to trace for each bin := 10

End Ray Tracing Matrix Parameters:=

End Projector Pair Using Matrix Parameters :=

; if the next parameter is disabled,

; the sensitivity will be computed using the normalisation object

sensitivity filename:= SENSITIVITY

; if next is set to 1, sensitivity will be recomputed

; and written to file (if "sensitivity filename" is set)

recompute sensitivity := 1

use subset sensitivities := 0

; This input is to read the multiplicative factors (normalisation\*attenuation). The suffix of each file is \_g#

normalisation sinograms := ATTENNORMFACTORS

; The input is to read the additive term (randoms + scatter). The suffix of each file is \_g#

additive sinograms := scaled\_attcor\_upsampled\_scatter\_estimation

Gate Definitions filename := MOTION.gdef

; The Motion Vectors are in image file format and there suffix is: \_g#d% where % corresponds to the dimension (1, 2 or 3)

Motion Vectors filename prefix := MOTION

Reverse Motion Vectors filename prefix := INVERTEDMOTION

; here comes the MRP stuff

prior type := FilterRootPrior

FilterRootPrior Parameters :=

penalisation factor := 0

; you can use any image processor here

; the next parameters specify a 3x3x3 median

Filter type := Median

Median Filter Parameters :=

mask radius x := 1

mask radius y := 1

mask radius z := 1

End Median Filter Parameters:=

END FilterRootPrior Parameters :=

end PoissonLogLikelihoodWithLinearModelForMeanAndGatedProjectionDataWithMotion Parameters:=

; Number of subsets should be a divisor of num\_views/4

number of subsets:= 23

number of subiterations:= 460

save estimates at subiteration intervals:= 23

output filename prefix := MOTIONCORRECTEDIMAGE

END:=

**C. Regularisation and Noise**

According to [Tso13], regularisation is generally advised for either RTA or MCIR. Currently, MRP is validated. Further tests on quadratic prior and OSSPS implementation are recommended to the researchers using STIR. Otherwise, two iterations are usually enough to obtain a relatively good image if followed by postfiltering.

**D) Further Extensions for the Future**

**More robust testing**: Currently the tests are performed based on basic tests.

**OSSPS**: Needs further debugging as it seems the current settings do not reconstruct the motion compensated image.

**Scatter Estimation**: Assumed to have it already estimated prior to reconstruction.

**E) Realistic Datasets and other info**

Apart from the corresponding test class and the recon\_test\_pack files, and extensive database of realistic simulated PET data with motion is available: [http://www.isd.kcl.ac.uk/pet-mri/simulated-data](http://www.isd.kcl.ac.uk/pet-mri/simulated-data/).

Motion fields of these data have been estimated using a local hierarchical affine registration algorithm developed by Christian Buerger [Bue11]. This independent library is provided freely available under BSD-2 licence at: <http://www.isd.kcl.ac.uk/internal/hyperimage>. The library is working with GIPL (Guy’s Image Processing Lab) File Format and we provide in STIR two utilities two convert them to/from Interfile format (conv\_gipl\_to\_interfile and conv\_interfile\_to\_gipl). Note that special care need to be taken with respect to the orientation the original files have been stored in gipl format.

**Literature**

[Bue12] Buerger, C., T. Schaeffter, and A. P. King **(2011)**, Hierarchical adaptive local affine registration for fast and robust respiratory motion estimation. Medical Image Analysis, vol. 15, pp. 551-564, <http://dx.doi.org/10.1016/j.media.2011.02.009>

[Kle96] Klein, G. J., B. W. Reutter, and R. H. Huesman **(1996)**, Non-rigid summing of gated PET via optical flow, *1996 IEEE Nuclear Science Symposium Conference Record*, vol. 2, pp. 1339-1342, <http://dx.doi.org/10.1109/NSSMIC.1996.591692>

[Li06] Li, T., B. Thorndyke, E. Schreibmann, *et al* **(2006)**, Model-based image reconstruction for four-dimensional PET. *Med Phys* vol. 2, pp. 1288-1298  [http://dx.doi.org/10.1118/1.2192581](http://link.aip.org/link/doi/10.1118/1.2192581)

[Pol12] Polycarpou, I., C. Tsoumpas and P. K. Marsden **(2012)**, A new algorithm for scaling of PET scatter estimates using all coincidence events. *Med Phys* vol. 39, 3586 – 3590, <http://dx.doi.org/10.1118/1.4754586>.

[Rah08] Rahmim, A., K. Dinelle, *et al* **(2008)**, Accurate event-driven motion compensation in high-resolution PET incorporating scattered and random events. *IEEE Trans Med Imaging*, vol. 27, 1018 – 1033, <http://dx.doi.org/10.1109/TMI.2008.917248>.

[Tso13] Tsoumpas, C., I. Polycarpou, K. Thielemans, *et al* **(2013)**, The effect of regularisation in motion compensated PET image reconstruction: A realistic numerical 4D simulation study. *Phys Med Biol*, vol. 58, 1759-1773, <http://dx.doi.org/10.1088/0031-9155/58/6/1759>.

**Brief description of some additional files:**

./utilities/warp\_image.cxx

This program warps an image to another position using as input given motion vectors

./utilities/zeropad\_planes.cxx

This program zero pads the start & end planes of an image.

./utilities/conv\_GATE\_projdata\_to\_interfile.cxx

This program converts GATE ECAT output (.ima) into STIR interfile format.

./utilities/shift\_image\_origin.cxx

This utility can be used to simply change the origin in the interfile header of an image.

./utilities/shift\_image.cxx

This utility can be used to apply translations to an image. The translations are applied with no interpolation, but the entire image moves for an given integer number of voxels.

./utilities/calculate\_attenuation\_projdata.cxx [based on other STIR files]

This utility is based on the fwdtest and it is used to give more flexibility on estimating the attenuation coefficient factors with a better projector (e.g. include more rays than the –PMRT option for the calculate\_attenuation\_coefficients utility).

./IO/GIPL\_ImageFormat.cxx

./IO/GIPL\_ImageFormat.h

These files contain information to read and write gipl files. It has been available within lreg (2nd release) under BSD-2 licence and it is adapted for compatibility with STIR.

1. [↑](#footnote-ref-1)