Implementing a patient-based orientation system within STIR

STIR Users' and Developers' Meeting 2018 – MIC18, Sydney

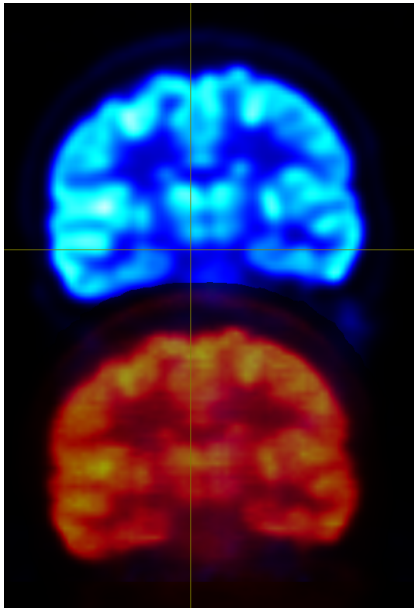
Ashley Gillman, et al. | CSIRO AEHRC and UQ Faculty of Medicine

Richard Brown, David Atkinson, Ben Thomas, Evgueni Ovtchinnikov, Nicholas Dowson, Kris Thielemans

15th November, 2018

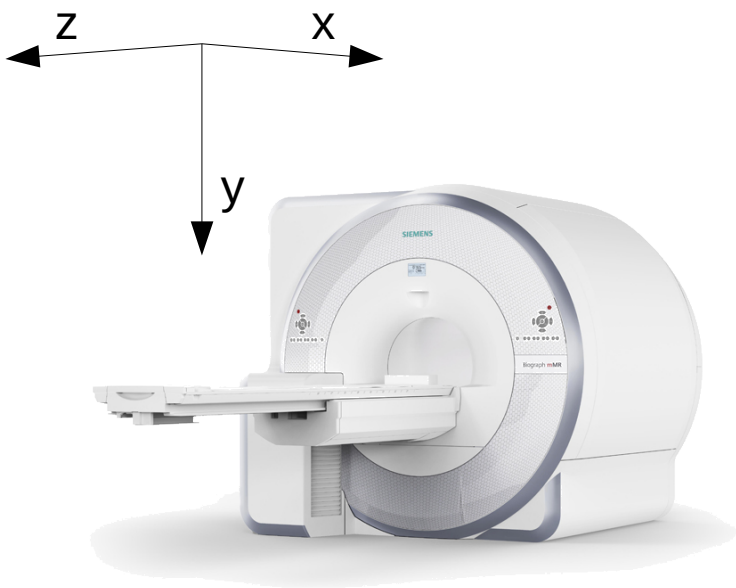
Motivation

- Currently, SIRF's output images aren't aligned with vendor reconstructions.
- This can make it difficult to align vendor images, e.g. for:
  - Attenuation correction using CT or MRAC.
  - Comparing STIR reconstructions with vendor reconstructions.
  - Using anatomical prior images.
  - Potentially simplifies use of STIR in clinical studies.



Coordinate Systems

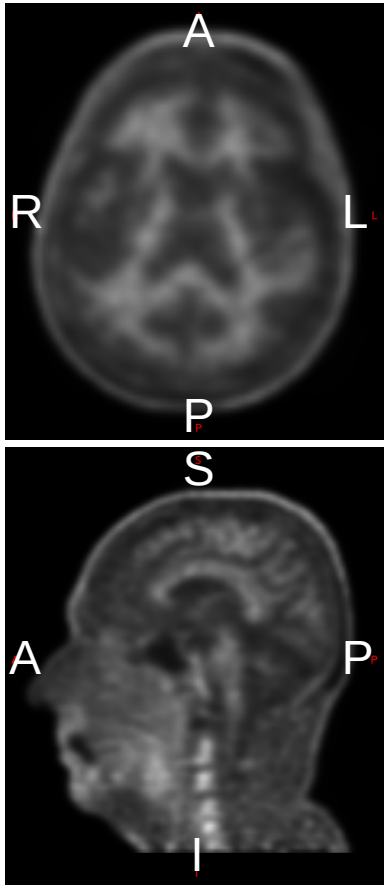
- STIR Uses a gantry-based coordinate system.
- From the STIR Developers' overview:
  - X-axis : horizontal axis, pointing right when looking from the bed into the gantry
  - Y-axis : vertical axis, pointing downwards
  - Z-axis : the scanner axis, pointing from the gantry towards the bed



Demonstration of STIR axes conventions

Coordinate Systems Cont.

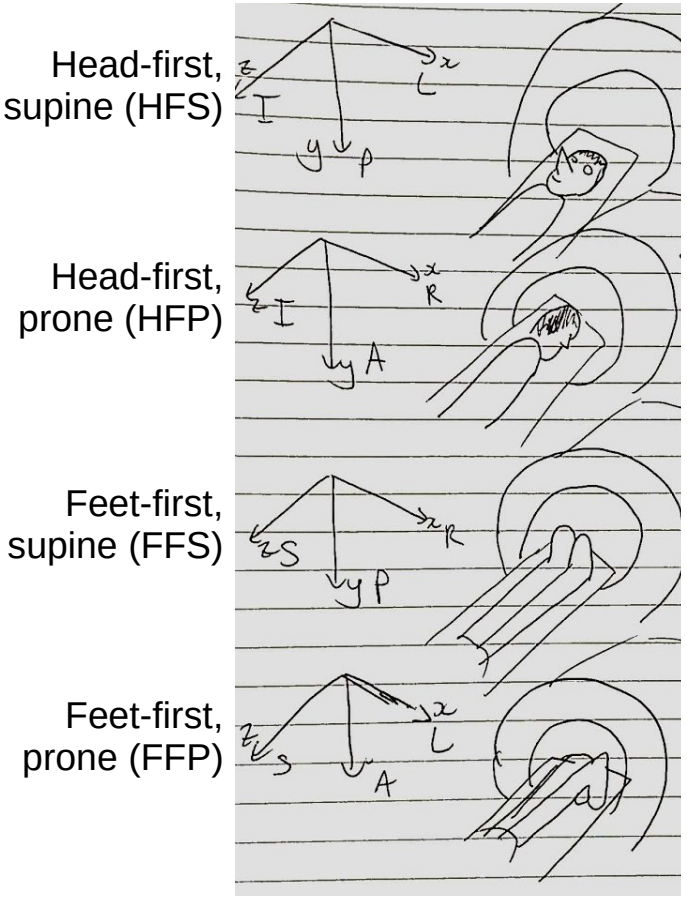
- Many common medical technologies (e.g., DICOM, NifTI) use a patient-based coordinate system, which is define w.r.t. patient anatomy.
  - In DICOM, axes directions are defined with respect to patient's Left, Posterior, Supine (LPS).
  - NIfTI orientation is defined by an affine transform of axes with respect to Right, Anterior, Supine (RAS).
- This system is useful in analysis as it allows rapid approximate alignment of anatomy.



Demonstration of Patient-based axes conventions

Mapping the Coordinate Directions

- In order to convert between these orientation systems, the patient orientation need to be known. Then a mapping is straightforward.



Demonstration of axes mapping for various patient positions

Mapping the Coordinate Directions Cont.

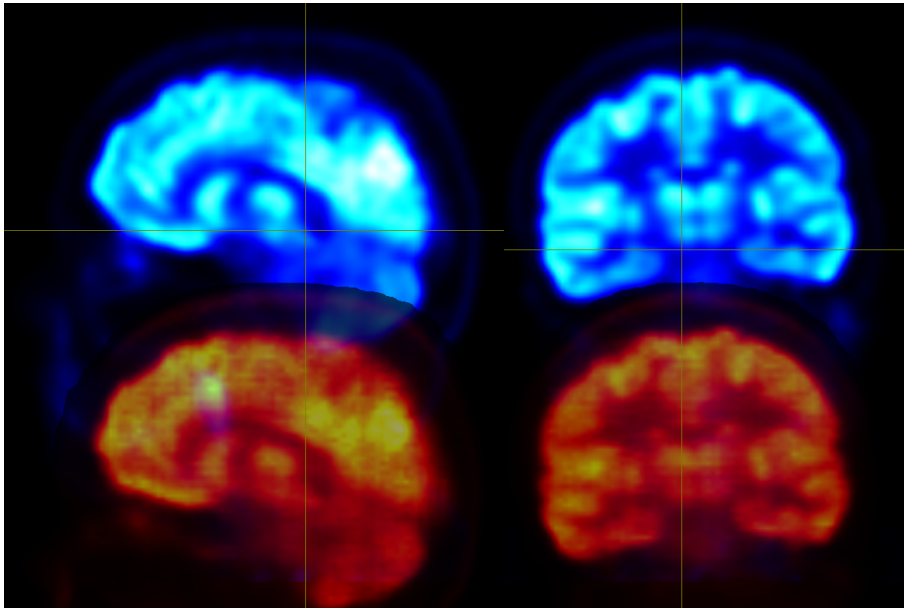
- Implemented in STIR Pull Request #248
- Adds `get_LPS_coordinates_for_indices()` and friends to `DiscretisedDensity`

```
switch (patient_position.get_position())
  {
  case PatientPosition::unknown_position: // If unknown, assume HFS
  case PatientPosition::HFS:              // HFS means currently in patient LPI
    flip_coords.z() *= -1;
    break;

  case PatientPosition::HFP:              // HFP means currently in patient RAI
    flip_coords.x() *= -1;
    flip_coords.y() *= -1;
    flip_coords.z() *= -1;
    break;

  case PatientPosition::FFS:              // FFS means currently in patient RPS
    flip_coords.x() *= -1;
    break;

  case PatientPosition::FFP:              // FFP means currently in patient LAS
    flip_coords.y() *= -1;
    break;
```

```
+   //! Translation from LPS coordinates to continuous indices.
+   inline BasicCoordinate<num_dimensions, float>
+   get_index_coordinates_for_LPS_coordinates(const CartesianCoordinate3D<float>& coords) const;
+
+   //! Translation from physical to LPS coordinates.
+   inline CartesianCoordinate3D<float>
+   get_LPS_coordinates_for_physical_coordinates(const CartesianCoordinate3D<float>& indices) const;
+
+   //! Translation from indices to LPS coordinates.
+   inline CartesianCoordinate3D<float>
+   get_LPS_coordinates_for_indices(const BasicCoordinate<num_dimensions,int>& indices) const;
+
+   //! Translation from continuous indices to LPS coordinates.
+   inline CartesianCoordinate3D<float>
+   get_LPS_coordinates_for_indices(const BasicCoordinate<num_dimensions,float>& indices) const;
+
+   //! Translation from LPS coordinates to physical coordinates.
+   inline CartesianCoordinate3D<float>
+   get_physical_coordinates_for_LPS_coordinates(const CartesianCoordinate3D<float>& coords) const;
+
+   //! Translation from LPS coordinates to indices.
+   inline BasicCoordinate<num_dimensions,int>
+   get_indices_closest_to_LPS_coordinates(const CartesianCoordinate3D<float>& coords) const;
```

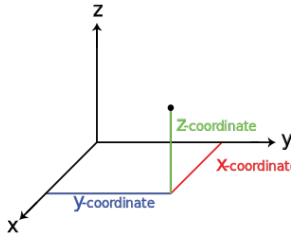
Using new directions with ITK IO

- With Pull Request #257, ITK I/O is LPS compatible.
- When saving, the image will be correctly oriented.
- When reading, only DICOM images store patient orientation and will be correctly oriented.
- NIfTI images will be read as “unknown_orientation”, treated internally as Head-first, supine.



Vendor reconstruction (blue, DICOM) and STIR reconstruction (red, NIfTI), in the same, patient-based orientation system.

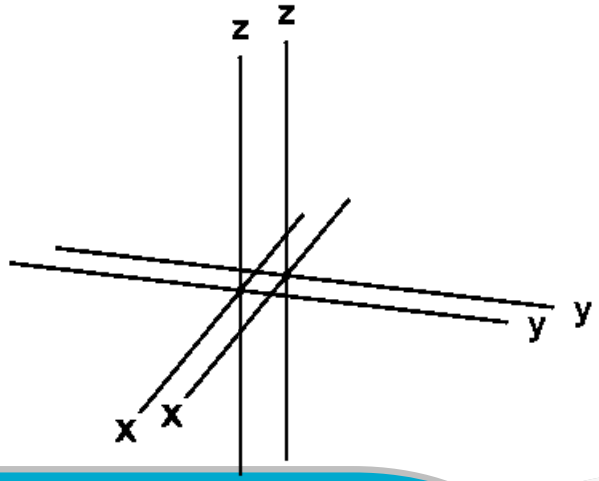
Difference between vendor and STIR FoR



- STIR and vendor images have different Frame-of-References (FoR, [0, 0, 0] points)
- From the STIR Developers' overview:
  - The origin of the X and Y axes are located on the central axis of the PET scanner and the Z origin (z=0) is located in the middle of the first ring (i.e., at the opposite side of the bed).
- DICOM only guarantees that images with the same Frame of Reference Unique Identifier are self-consistent.
  - Since it is patient based, the FoR moves with the bed.

Difference between vendor and STIR FoR

- Testing on the Siemens mMR at HIRF, Brisbane and at UCL, London, indicate the DICOM FoR is, at least approximately, the center of the PET gantry.
  - NB: More likely the MR isocentre, and the PET gantry center happens to be approximately the same.



Experiment – changing the LPS origin

- To test, I created a temporary new origin, called the vendor origin.
  - It is set to:
    STIR-origin + vector-to-centre + bed-position-vector
- LPS uses this, and therefore ITK IO

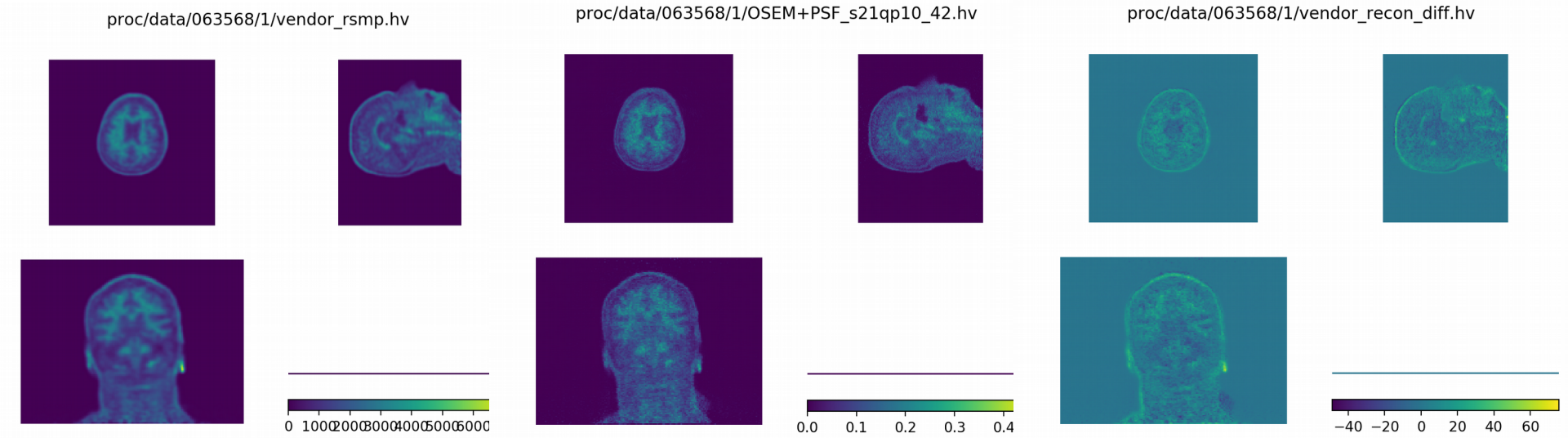
```
const CartesianCoordinate3D<float>
ProjDataInfo::
get_location_of_vendor_frame_of_reference_in_physical_coordinates() const {
  return
    //vendor FoR is from the centre of the gantry, STIR is from first ring
    get_vector_centre_of_first_ring_to_centre_of_gantry()
    //vendor FoR is bed-based
    + get_bed_position();
}
```

```
template<class elemT>
void
VoxelsOnCartesianGrid<elemT>::
init_from_proj_data_info(const ProjDataInfo& proj_data_info,
                         const float zoom,
                         const CartesianCoordinate3D<float>& origin,
                         const CartesianCoordinate3D<int>& sizes)
{
  ... //omitted

  this->set_origin(origin);
  this->set_vendor_origin
    (origin
     - (proj_data_info
        .get_location_of_vendor_frame_of_reference_in_physical_coordinates()));

  ... //omitted
}
```

```
template<int num_dimensions, typename elemT>
CartesianCoordinate3D<float>
DiscretisedDensity<num_dimensions, elemT>::
get_LPS_coordinates_for_indices
(const BasicCoordinate<num_dimensions, float>& indices) const
{
  return swap_axes_based_on_orientation
    (get_relative_coordinates_for_indices(indices) + get_vendor_origin(),
     get_exam_info().patient_position);
}
```

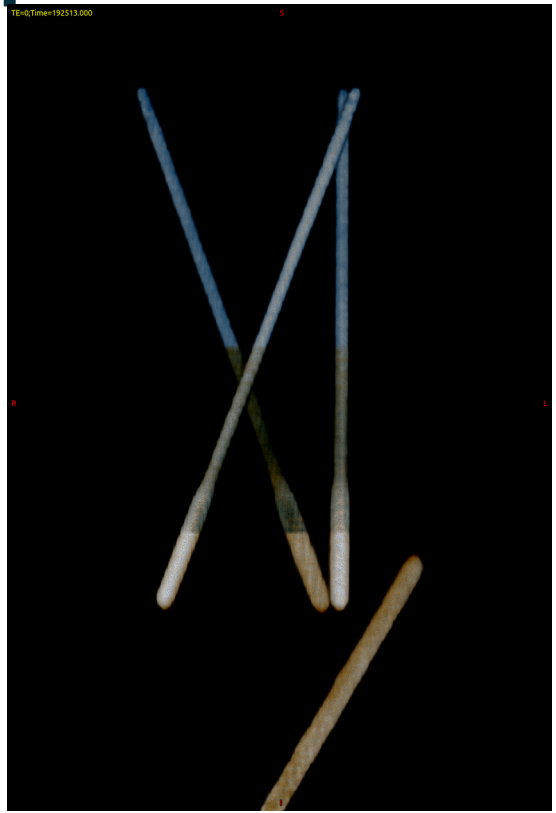
Results – changing the LPS origin

- Here we have the vendor resampled into STIR recon space (but no physical transform), a STIR recon, and the difference.



Results – changing the LPS origin

- Here we have a volume render of a Rod Phantom, acquired on the UCL mMR. 2 Bed positions were acquired with 50% overlap.
- In white, the vendor reconstruction, 2 bed positions composed.
- Overlay in red and blue are STIR reconstructions of each of the bed positions.



TODO: Implementation

- Second origin isn't an elegant solution.
- How to encode the gantry-image offset?
- Image data could encode gantry center position
  - DICOM[1]: Isocenter Position (300A,012C)
    - Isocenter coordinates (x,y,z), in mm. Specifies the location of the machine isocenter in the Patient-Based Coordinate System associated with the Frame of Reference. It allows transformation from the Equipment-Based Coordinate System to the Patient-Based Coordinate System.
  - New Interfile "vendor origin (mm) [X]"?
- ProjDataInfo has Bed Position and Gantry Information

[1] http://dicom.nema.org/MEDICAL/DICOM/current/output/chtml/part03/sect_C.8.9.4.html

TODO: Long Term

- Original plan was to have STIR's origin be consistent with the Vendor FoR.
  - This gets difficult when there is an offset – It means STIR origin is no longer the same every time. Each image would have to be paired with a specific Projection Data to have all the information.
- PET – MR gantry offset.
  - For this scanner, seems to only be in the Raw Data DICOMs – public info?

Conclusion

- ITK Input reading DICOM images will be correctly orientated.
- ITK Input reading NIfTI images will only be correctly orientated if HFS.
- ITK Output NIfTI/DICOM images spatially consistent with vendor DICOMs (soon), including for varying bed position.
- I will be using this facility for attenuation correction, comparisons with vendor reconstructions, using priors in PET reconstruction, and replacing portions of an existing clinical research study.

It takes a village…

Coauthors

Richard Brown
David Atkinson
Ben Thomas
Evgueni Ovtchinnikov
Nicholas Dowson
Kris Thielemans

Supervisors

Nicholas Dowson
Stephen Rose
Jye Smith
Jason Dowling

Advisors

Graham Galloway
Ken Miles
Anders Rodell
Karine Mardon

Support

Jurgen Fripp
Louise Campbell
HIRF Staff

Collaborators

Kris Thielemans
Richard Brown
David Atkinson
Ben Thomas
Evgueni Ovtchinnikov
Paul Thomas
John Woodford
Peter Jenvey
Parnesh Raniga
Pierrick Bourgeat
Thomas Gaass
Christine Guo
Michael Breakspear