BIDS Extension Proposal 14 (BEP014): Spaces and mappings

version 0.0.1 (working copy)

Extension moderators/leads: Oscar Esteban <oesteban@stanford.edu>

This document contains a draft of the Brain Imaging Data Structure standard extension. It is a community effort to define standards in data/metadata. This is a working document in draft stage and any comments are welcome.

This specification is an extension of BIDS and BIDS Common Derivatives proposal, and general principles are shared. The specification should work for many different settings and facilitate the integration with other imaging methods.

To see the original BIDS specification, see [this link](), for Common Derivatives, see [this link](). This document inherits all components of the original specification (e.g. how to store imaging data, events, stimuli, and behavioral data), and should be seen as an extension of it, not a replacement.

Modifications to Appendix VIII of the main specification

Definitions

Coordinate System

A coordinate system specifies the how coordinates associated with the data. A coordinate system can be described by its topology (cartesian, spherical, polar, etc.,.), dimensionality, and units of measurement along each dimension. Every device that makes spatial measurements (e.g., an MR scanner, a Polhemus digitizer) or that has sensors described in space (e.g. an MEG or EEG system) has a coordinate system. The data output by most of these devices describe the number of dimensions, the units along each dimension, and in some cases is accompanied by the details of the origin (for example, time relative to the start of recording).

Space, atlas and map

A *space*|*atlas*|*map* is embedded in a coordinate system to ascribe meaning to the coordinates. For instance, *Scanner*, *TT88*, *MNI*, *FS* are different coordinate systems, while the *subject, Talairach atlas*, *MNI152*, and *fsaverage* ascribe meaning to the coordinates in terms of brain anatomy. As an example to distinguish coordinate systems and spaces/atlas/maps, a 4D fMRI time series acquired without motion correction has individual time points all in the same coordinate system, but a voxel maps to different anatomical locations(due to motion) in different time points.

Landmarks

Landmarks are properties of the object being measured, irrespective of the measurement, and can exist in any coordinate system in which the object is placed/measured. They are recognizable points that can be palpated on the object (such as the nasion and pre-auricular points) or are recognizable in the imaged object's structure such as the anterior commissure in a brain. Instead of anatomical landmarks, fiducials can be used. These are objects or devices that in combination with the measurement system generate a localizable signal. Fiducials are placed at the intended position before scanning (i.e. *markers*). Positions of either landmarks or fiducials are identified via manual or automated algorithms that detect these landmarks or fiducials in (e.g., MR) or from (e.g., MEG) the acquired data. It is common, but not required, to place fiducials on the anatomical landmarks to facilitate their detection and to facilitate coregistration between different types of data.

These landmarks can be explicitly represented as coordinates in a file or implicitly as voxel/pixel intensity relations in image files.This information (coordinates or intensity relations) representing landmarks of the same or similar object (e.g., a template brain) in different coordinate systems can be used to transform the object between coordinate systems.

Additions to Metadata:

Any sidecar json file corresponding to a BIDS data object whether created by an instrument (e.g., MR Scanner, MEG/EEG system, iEEG, TMS, DBS), or transformed through some initial processing (e.g., alignment across coordinate systems of MEG, Polhemus and MR data from a participant) should describe the CoordinateSystem(s) within which the shared data can be interpreted. This keyword will follow principles of inheritance.

Key	Required	Description
`CoordinateSystem`	Mandatory if no implicit coordinate system	Key indicates the coordinate system associated with the File. The coordinate system can be implicit to the File, for instance when data are images stored in NIfTI format. Can be a list. See Table below for list of allowed systems.
`ReferenceMap`	Required when coordinate system is Aligned	Key indicates the reference atlas or map that the File is aligned to. See table below for list of common spaces.
`NonstandardReference`	Required when a non standard template or space is used. (e.g., a custom template in	A path to a file that was used as, or can be used as, a reference image for determining the coordinate space of this file. If Space is a list, Space reference must also be a list.

	MNI305 space)	
`ReferenceIndex`	Required when an index into a 4D (ReferenceMap|NonstandardReference) file is used.	Used to index into a 4D spatial-reference file.
`TransformFile`	Optional	A path to the file used to transform the source into this file.

CoordinateSystem reserved values:

Name	Description
`Device`	The coordinate system of the device used to acquire the data.
`Aligned`	The coordinate system is specified by a target space (e.g., Talairach88, MNI305, etc.,.). See the Space keyword for details of targets.
`Custom`	A custom coordinate system that is not in alignment (dimensions, axis orientation, unit) with any device coordinate system.

ReferenceMap reserved values:

`orig`	A (potentially unique) per-image space. Useful for describing the source of transforms from an input image to a target space.
`custom`	This can be used to point to the non-standard space of a file. This should be used only if the reference file is not in any of the other spaces.

Special values for indicating a standard Atlas as ReferenceMap:

`FS305`	FreeSurfer variant of the MNI305 space
`MNI152Lin`	Also known as ICBM (version with linear coregistration) http://www.bic.mni.mcgill.ca/ServicesAtlases/ICBM152Lin
`MNI152NLin6[Sym|Asym]`	Also known as ICBM 6th generation (non-linear coregistration). Used by SPM99 - SPM8 and FSL (MNI152NLin6Sym). http://www.bic.mni.mcgill.ca/ServicesAtlases/ICBM152NLin6
`MNI152NLin2009[a-c][Sym|Asym]`	Also known as ICBM (non-linear coregistration with 40 iterations, released in 2009). It comes in either three different flavours each in symmetric or asymmetric version. http://www.bic.mni.mcgill.ca/ServicesAtlases/ICBM152NLin2009
`MNIColin27`	Average of 27 T1 scans of a single subject

	http://www.bic.mni.mcgill.ca/ServicesAtlases/Colin27Highres. Also has variants
MNI305	Also known as avg305
NIHPD	Pediatric templates generated from the NIHPD sample. Available for different age groups (4.5–18.5 y.o., 4.5–8.5 y.o., 7–11 y.o., 7.5–13.5 y.o., 10–14 y.o., 13–18.5 y.o. This template also comes in either -symmetric or -asymmetric flavor. http://www.bic.mni.mcgill.ca/ServicesAtlases/NIHPD-obj1
TT88	J. Talairach and P. Tournoux, "Co-planar Stereotaxic Atlas of the Human Brain: 3-Dimensional Proportional System - an Approach to Cerebral Imaging", Thieme Medical Publishers, New York, NY, 1988 Mostly not used anymore. But the original Talairach transform was a piece affine transform of 12 boxes.
OASIS30Atropos	https://figshare.com/articles/ANTs_ANTsR_Brain_Templates/915436 http://www.mindboggle.info/data.html
ICBM452AirSpace	Reference space defined by the "average of 452 T1-weighted MRIs of normal young adult brains" with "linear transforms of the subjects into the atlas space using a 12-parameter affine transformation" http://www.loni.usc.edu/ICBM/Downloads/Downloads_452T1.shtml
ICBM452Warp5Space	Reference space defined by the "average of 452 T1-weighted MRIs of normal young adult brains" "based on a 5th order polynomial transformation into the atlas space" http://www.loni.usc.edu/ICBM/Downloads/Downloads_452T1.shtml
IXI549Space	Reference space defined by the average of the "549 [...] subjects from the IXI dataset" linearly transformed to ICBM MNI 452.Used by SPM12. http://www.brain-development.org/
fsnative	Images were sampled to the FreeSurfer surface reconstructed from the subject's T1w image
fsaverage[3|4|5|6|sym]	Images were sampled to the FreeSurfer surface reconstructed from the subject's T1w image, and registered to an fsaverage template
fsLR[164k|32k]	Images were sampled to the 164k or 32k fsaverage_LR surface reconstructed from the T1w image (high resolution space of the HCPPipelines)
UNCInfant[0|1|2]V[21|22|23]	Infant Brain Atlases from Neonates to 1- and 2-year-olds. https://www.nitrc.org/projects/pediatricatlas

Example sidecar files:

For a NIFTI file (Single coordinate system), one could have registered the File on to the standard MNI305 template. The way to write the metadata of such File is:

```
{

"Space": "MNI",

"TransformFile": "/path/to/xfm",
```

```
"ReferenceMap": "MNI305"

}
```

However, it could also be the case that a nonstandard derivative of MNI305 was used as standard space for the File. That can be written as follows:

```
{

"Space": "MNI",

"TransformFile": "/path/to/xfm",

"NonstandardReference": "uri or path to file"

}
```

Some derivatives such as CIFTI Files allow for multiple coordinate systems. Such possibility is enabled by using lists of spaces and references:

```
{

"Space": ["MNI", "fsLR"],

"TransformFile": "/path/to/xfm",

"ReferenceMap": ["MNI305", "fsLR32k"]

}
```

Differing references of the same spaces with respect to the above example can be expressed as follows:

```
{

"Space": ["MNI", "fsLR"],

"TransformFile": "/path/to/xfm",

"ReferenceMap": ["MNI152Lin", "fsLR164k"]

}
```

Mappings (transformation) between spaces

Definitions and introduction

A transformation file encapsulates a mathematical operation on the coordinates of those objects susceptible of being spatially transformed (i.e. images, surfaces, or points). For example, image analysis workflows that operate on a common standardized space for all the subjects will estimate one transform per subject to map the information (e.g. BOLD signal, the pial surface mesh, etc) from the individual to the common space. Although a transform is strictly just a coordinates mapping operation, it is generally understood by developers of registration algorithms that *applying a transform on an image* implicitly includes two operations (see below), which can be confused with *applying a transform on a surface of points* that just finds new coordinates through the transformation, for each vertex of the surface.

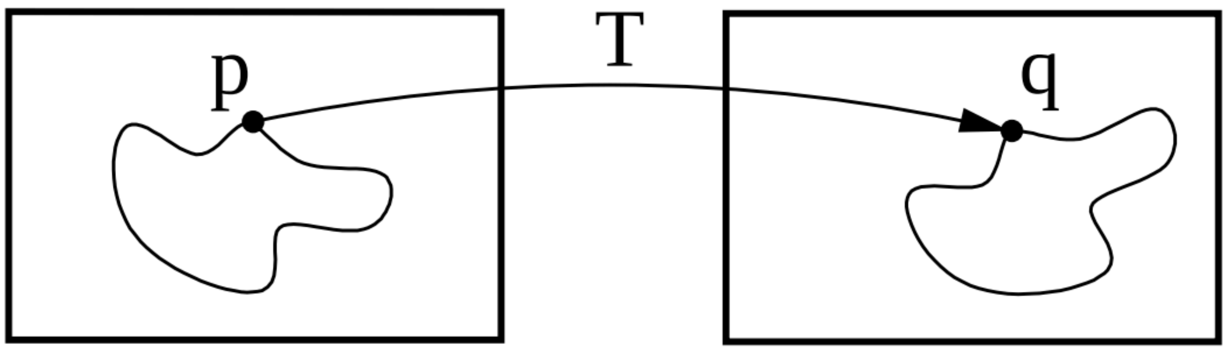

Figure 1. A transformation maps corresponding features or landmarks from one object to another. (Figure reproduced from the ITK Software Guide, Book 2, Figure 3.1).

Applying a transform on an image. Transformations allow to project information *from* one image (also called "moving" image) on *to* another image ("reference" image).As introduced before, this projection involves two operations: 1) the mapping of all points in the reference grid to the coordinates in the moving image; and 2) interpolating data on the mapped moving coordinates (as they are unlikely to fall on points of the moving image grid).

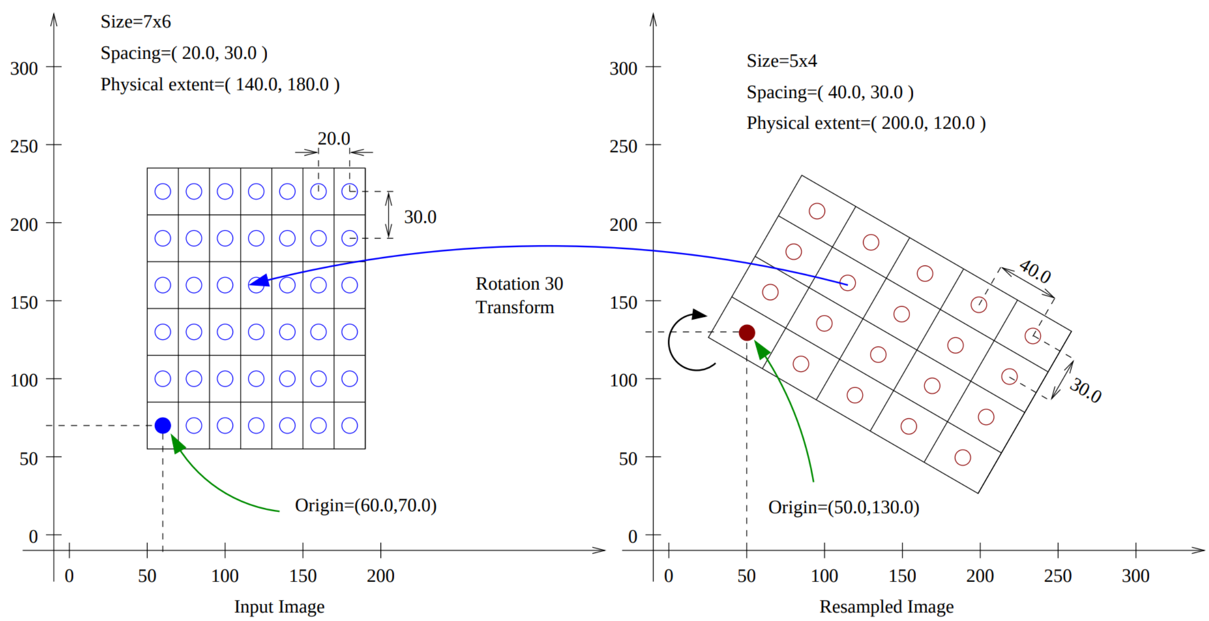

Figure 2. Example of applying a transform on an image, in particular a rotation of 30 degrees clockwise. The image on the right-hand side is the *resampled image* in the reference image's space. To generate such result, all the coordinates of all pixels in the reference (red circles) are iterated and corresponding coordinates in the moving (*input*) image at the left side are computed through the transform (blue arrow). Since the mapped coordinates will not likely fall at the center of a pixels in the input image (blue circles), then a value for the image is interpolated from the neighboring pixels before it can be assigned to the corresponding resampled pixel (red circle at the source of the blue arrow). This figure has been reproduced from the ITK Software Guide, Book 2, Figure 2.51.

Naming transform files - BIDS entities

Transform-files are labeled with the `from-<space>` and `to-<space>` BIDS entities. An additional entity, `mode-<mode_label>`, indicates what kinds of objects the transform operates to, in order to allow accommodating the widely extended use case of *applying a transform on an image* that implicitly involves two operations.

BIDS entities specific of the transform file template:

`from-<space_label>`	(MANDATORY) Source space in the information flow when applying a transform.
`to-<space_label>`	(MANDATORY) Destination space in the information flow when applying a transform.
`mode-<mode_label>`	(MANDATORY) Acceptable `mode_label` values are `image`, `points`, `surface` and `sphere`. Indicates how the `from` and `to` labels should be interpreted: as pure coordinates mapping (`points`, `surface`) or as a coordinates mapping internal to a resampling process (`image`, `sphere`).

Transform file mode

Due to the different ways in which *apply a transform* is understood (described above), the `mode-<mode_label>` defines such convention. Sometimes, the transform needed for one use case can be calculated from another transform, however, sometimes these type of calculations are not appropriate. For example, a `mode-points` transform can sometimes we calculated as the inverse of a `mode-image` transform, but only if the mode-image transform is “Invertible”. To clarify the acceptable use of a particular transform file, the mode key value pair has been introduced.

`<mode_label>`	**description**
`points|surface`	The File maps individual coordinates (i.e. location of implanted electrodes, sets of coordinates for a surface mesh) given in the `from-<space_label>` space into the `to-<space_label>` space.
`image|sphere`	Indicates that this File is intended to apply a transform on discretely sampled data defined on the `from-<space_label>` space (e.g. 2D image, 3D image, sphere, etc.) and resample such those data on a discretely

	sampled to-<space_label> space.

Please note that, given these definitions, a the transform file from-A_to-B_mode-points is completely equivalent to from-B_to-A_mode-image.

Examples of usage of various neuroimaging tools that calculate transforms:

	from	**to**	**mode**	Example command line to moving the same image from “MNI” to “bold” space
FS	mov	target		
FSL (flirt)	in	ref	image	flirt -in MNItemplate.nii.gz -ref <source_bids>_bold.nii.gz -omat <source_bids>_from-MNI_to-BOLD_mode-image_xfm.mat -dof 6
FSL (fnirt)	in	ref	image	fnirt -in=MNItemplate.nii.gz -ref=<source_bids>_bold.nii.gz --cout=<source_bids>_from-MNI_to-BOLD_mode-image_xfm.mat
ANTs	moving	fixed		antsRegistration --dimensionality 3 \ --output <source_bids>_from-MNI_to-BOLD_mode-image_xfm] \ --metric MI[<source_bids>_bold.nii.gz,MNItemplate.nii.gz,...] \ ...

Transform files template

Since there is no standard file-format for transform objects, any transform format will be accepted until a common format for spatial transforms is designed and accepted. In the following, we will use the .h5 extension to refer to the File storing the actual transform, and such extension can be replaced by that generated with any registration software (e.g. .nii[.gz], .mat, .lta, .xfm, .h5). Transforms are a new file-type xfm, and thus they will use such key as file suffix and will be stored under a xfm/ folder.

A folder under the participant level will contain all transforms from/to those subject spaces:

```
<pipeline_name>/
  sub-<participant_label>/
    xfm/
      <source_bids>_from-MNI_to-BOLD_mode-image_xfm.h5
      <source_bids>_from-MNI_to-BOLD_mode-image_xfm.json
      <source_bids>_from-BOLD_to-MNI_mode-points_xfm.h5
      <source_bids>_from-BOLD_to-MNI_mode-points_xfm.json
      <source_bids>_from-MNI_to-BOLD_mode-points_xfm.h5
      <source_bids>_from-MNI_to-BOLD_mode-points_xfm.json
```

where <source_bids>_from-BOLD_to-MNI_mode-points_xfm.h5 is a (soft)link to <source_bids>_from-MNI_to-BOLD_mode-image_xfm.h5, to encode the said equivalence between both.

For example, for a participant with label 001 with both BOLD and T1w images:

```
sub-001/
       xfm/
           sub-001_task-rest_run-01_from-orig_to-index10_mode-points_xfm.h5
           sub-001_from-T1w_to-meanBOLD_mode-points_xfm.h5
           sub-001_from-T1w_to-meanBOLD_mode-points_xfm.json
           sub-001_from-MNI_to-T1w_mode-image_xfm.h5
           sub-001_from-MNI_to-T1w_mode-image_xfm.json
```

Example JSON sidecar for transform files:

```
{
  "Type": ["rigid", "displacementfield"],
  "Multiplexed": [true, false],
  "Software": "antsRegistration",
  "SoftwareVersion": "you wish!",
  "Invertible": true,
  "FromFile": ...,
  "ToFile": ...,
  "FromFileSHA": ...,
  "ToFileSHA": ...,
  "CommandLine" "antsRegistration --collapse-output-transforms 0 --dimensionality 3
--initial-moving-transform [ trans.mat, 0 ] --initialize-transforms-per-stage 0
--interpolation Linear --output [ output_, output_warped_image.nii.gz ] --transform
Affine[ 2.0 ] --metric Mattes[ fixed1.nii, moving1.nii, 1, 32, Random, 0.05 ]
--convergence [ 1500x200, 1e-08, 20 ] --smoothing-sigmas 1.0x0.0vox --shrink-factors
2x1 --use-estimate-learning-rate-once 1 --use-histogram-matching 1 --transform SyN[
0.25, 3.0, 0.0 ] --metric Mattes[ fixed1.nii, moving1.nii, 1, 32 ] --convergence [
100x50x30, 1e-09, 20 ] --smoothing-sigmas 2.0x1.0x0.0vox --shrink-factors 3x2x1
--use-estimate-learning-rate-once 1 --use-histogram-matching 1
--winsorize-image-intensities [ 0.0, 1.0 ]  --write-composite-transform 1"
}
```

`Type`	List of elements that are `translation|rotation|rigid|similarity|affine|displacementfield|parametric|composite`
`Multiplexed`	Whether several transforms are to be applied to the last dimension of the dataset (e.g. a list of affine matrices to correct for head motion each 3D volume)
`Invertible`	Whether the transform is diffeomorphic or not.
`FromFile`	The fixed (reference) file used by `Software` to compute the transform
`FromFileSHA`	
`ToFile`	The moving (target) file used by `Software` to compute the transform

ToFileSHA	
CommandLine	The command line used to create the transform

NOT FOR RC1 STARTING HERE --

Common transform file format

Extension: .x5

Format: h5

1. Store separate image and world metadata for each transform in the header
2. Store transforms in world-to-world representation in RAS convention.
3. Store composite and multiplexable transforms. For example, the affine/rigid matrices of a 4D time series, a mapping to T1, a registration to MNI
4. Support linear, displacement fields, spherical, and parametric transforms

Mappings on spheres / between surfaces