Coordinate Transformation Conventions for Spatial Registration

Mark Jenkinson

Abstract

This document describes, in detail, the coordinate transformation conventions used in several spatial registration/manipulation packages — in particular: FLIRT, MEDx, SPM, UMDS and MINC/MNI.

1 Introduction

Coordinates for images can either be in voxel units or mm. In addition, there is also the issue of what order the voxels are written in the analyze file (note that only analyze format storage will be considered here). Consequently, when specifying a spatial transformation from one image to another it is necessary to specify which system of units is being used for both images (source and destination) and also which direction the transformation specifies (e.g. converting coordinates from new to dest or dest to new).
Affine transformations between spatial coordinates are usually specified by a 4×4 matrix which operates on homogeneous coordinates. That is:

$$\begin{bmatrix} X_2 \\ Y_2 \\ Z_2 \\ 1 \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{32} & A_{33} & A_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \\ 1 \end{bmatrix}. \tag{1}$$

This allows translations to be incorporated easily with a single matrix multiplication. The translation parameters are A_{14}, A_{24} and A_{34}. It is straightforward to see that there are a total of 12 degrees of freedom in the affine transformation of which 3 represent translation.

1.1 Data Storage — Analyze Files

The standard convention for data storage is that the image is written as a sequential string of data values, with the x coordinate running fastest, followed by y, followed by z, and if necessary, followed by t. That is, the first byte represents voxel $(0, 0, 0)$ (with all voxel indexing from 0 upward), the second byte represents voxel $(1, 0, 0)$, *etc.*
In general, the voxel (a, b, c) will be stored as byte N where $N = a + N_x(b + N_y.c)$. Here the matrix size (single slice dimensions in voxels) is (N_x, N_y).

1.2 Voxel and World Coordinates

World coordinates are those that use physical units, such as mm. To convert between voxel and world coordinates the dimension of a voxel needs to be known. If a voxel has dimensions (d_x, d_y, d_z) (in mm) then the coordinate transformation from voxel to world coordinates is:

$$\begin{bmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{bmatrix} = \begin{bmatrix} d_x & 0 & 0 & 0 \\ 0 & d_y & 0 & 0 \\ 0 & 0 & d_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_V \\ Y_V \\ Z_V \\ 1 \end{bmatrix} \tag{2}$$

or more concisely as

$$\mathbf{X_W} = D\,\mathbf{X_V} \tag{3}$$

where D is the matrix shown above, and so D^{-1} converts world to voxel coordinates.
Note that this assumes that the *origin* for both systems is at the very first voxel — that is, the one with coordinates $(0, 0, 0)$ which is normally (but not always) displayed as the bottom, left corner of the first slice. If the origins are different, then they need to be incorporated into the matrix D. For instance, if the world

coordinate origin $(0,0,0)$ occurs at a voxel coordinate of (o_x, o_y, o_z) then the appropriate transformation matrix, D, from voxel to wolrd coordinates is:

$$D = \begin{bmatrix} d_x & 0 & 0 & -o_x.d_x \\ 0 & d_y & 0 & -o_y.d_y \\ 0 & 0 & d_z & -o_z.d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4}$$

In analyze files the origin is usually stored in the originator field (as shorts) but in *SPM voxel conventions.* Note that a major complication to this can be the presence of negative voxel dimensions which may be intended to swap the direction voxel-wise, world-wise, both or neither. This is very dependent on the particular images, packages being used and proposed analysis. In this document all voxel dimensions will be assumed to be positive.

2 FLIRT Conventions

2.1 Coordinates

Voxel indexing is from 0 to $N-1$ in each dimension.
Standard relation to data storage (x then y then z).
Origin of all world coordinate systems correspond to voxel $(0,0,0)$.
These voxel coordinate conventions will be called *standard* for the remainder of the document.

2.2 Transformations

All FLIRT transformations are from the source to the destination with both in *world* coordinates. The transformation specifies a physical transformation of space and no information about the destination voxel dimensions or Field Of View (FOV) is included. Consequently, this transformation can be used easily with differently scaled versions (i.e. different voxel dimensions) of a given image.
To convert to a voxel to voxel coordinate transformation the matrix must be pre-multiplied by D_{dest}^{-1} and post-multiplied by D_{source}.

2.3 File Format

The file format used to save the transformation is a simple 4×4 ascii matrix, with 4 space separated values per line.

3 MEDx Conventions

3.1 Coordinates

Voxel indexing is from 0 to $N-1$ in each dimension.
Relation to data storage is *non-standard.* The difference from the standard is that the y voxel index is flipped from the normal standard. So, to convert between MEDx voxel coordinates and standard (FLIRT) voxel coordinates the following can be used:

$$Y_{standard} = N_y - 1 - Y_{medx}. \tag{5}$$

For example, in a $64 \times 64 \times 25$ EPI volume the MEDx coordinate $(30, 26, 12)$ is the same as the standard coordinate $(30, 37, 12)$.
Origin of the world coordinate system can be specified by the image/user but defaults to the $(0,0,0)$ voxel coordinate (MEDx convention).
Note that the display used by MEDx shows the images in radiographers convention and so appears to be differently ordered in the Left/Right direction, although the voxel reporting of coordinates is as specified above.

3.2 Transformations

The transformations saved by MEDx are from the source to the destination with both in *voxel* coordinates. Conversion to a standard world-world (FLIRT) transformation matrix is given by:

$$A_{flirt} = D_{dest} \cdot SY_{dest} \cdot A_{medx} \cdot SY_{source} \cdot D^{-1}_{source}. \tag{6}$$

where SY represents a voxel swapping in the y direction. That is, if there are N_y voxels in the y direction:

$$SY = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & N_y - 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{7}$$

Note that $SY^{-1} = SY$.

3.3 File Format

The main matrix is written as a column of 16 numbers (one per line), being a reshaped version of the A_{medx} matrix with the column index running fastest. That is the first number is A_{11} followed by A_{12} followed by A_{13} *etc.*
Information about the destination voxel world dimensions (outputusermatrix: reshaped 4×4 matrix) and number of voxels in FOV (outputsize: 3 values) are also stored as part of the file format. Other matrices are non-essential and do not need to be written.

4 SPM Conventions

4.1 Coordinates

Voxel indexing is *non-standard*: it is ***from*** 1 ***to*** N in each dimension. Therefore it is necessary to subtract 1 from each SPM voxel coordinates in order to obtain standard (FLIRT) coordinates.
Standard relation to data storage (x then y then z).
Origin of the world coordinate system can be specified by the image/user.

4.2 Transformations

The transformations used by SPM are from the source to the destination with the source using *voxel* coordinates and the destination using *world* coordinates. Conversion to a standard world-world (FLIRT) transformation matrix is given by:

$$A_{flirt} = D_{dest} \cdot VS \cdot SC^{-1}_{dest} \cdot D^{-1}_{dest} \cdot A_{spm} \cdot VS^{-1} \cdot D^{-1}_{source} \tag{8}$$

where VS is the voxel shift matrix (to convert 1 to N into 0 to $N-1$ conventions). That is:

$$VS = \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{9}$$

In addition, SC is a shift to take into account the Centre of Volume origin of the SPM coordinate system. That is:

$$SC = \begin{bmatrix} 1 & 0 & 0 & -(N_x+1)/2 \\ 0 & 1 & 0 & -(N_y+1)/2 \\ 0 & 0 & 1 & -(N_z+1)/2 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{10}$$

Note: these transformations may not work for all combinations of negative voxel dimensions and origins.

4.3 File Format

There are three distinct types of registration that SPM performs and each has a different convention for files.
For *Realign*: The affine matrix is saved in the MATLAB proprietary binary format. If loaded within MATLAB (e.g. using `load vol0001`) then the matrix `M` is the matrix A_{spm}.
For *Coregister*: Several matrices are saved in the MATLAB proprietary binary format in files called something like `vol_sn3d.mat`. Two matrices are required from these files: `MG` and `Affine`. The first specifies the voxel dimensions on the diagonal such that $MG = D_{dest} \cdot SC_{dest}$. This gives: $SC_{dest}^{-1} \cdot D_{dest}^{-1} = MG^{-1}$ and $D_{dest} = \text{diag}(MG)$. The main affine matrix is given by: $A_{spm} = MG \cdot Affine$.
For *Normalize*: Unknown.

5 UMDS Conventions

5.1 Coordinates

Not known — assumedly standard data relation and voxel indexing from 0 to $N-1$.
Origin appears to default to the Centre of Volume (COV) — see below.

5.2 Transformations

The transformations used by UMDS are from the source to the destination with both in *world* coordinates. Conversion to a standard world-world (FLIRT) transformation matrix is given by:

$$A_{flirt} = D_{dest} \cdot M_{dest} \cdot D_{dest}^{-1} \cdot A_{umds} \cdot D_{source} \cdot M_{source}^{-1} \cdot D_{source}^{-1}. \tag{11}$$

where M moves the origin to the Centre of Volume (in voxels). That is:

$$M = \begin{bmatrix} 1 & 0 & 0 & (N_x - 1)/2 \\ 0 & 1 & 0 & (N_y - 1)/2 \\ 0 & 0 & 1 & (N_z - 1)/2 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{12}$$

Warning: This transformation was originally written to convert to MEDx format as: $A_{medx} = M_{dest} \cdot D_{dest}^{-1} \cdot A_{umds} \cdot D_{source} \cdot M_{source}^{-1}$. The absence of SY matrices in the FLIRT version above represents a likely bug-fix to this old form but, although it should be correct, it is currently untested.

5.3 File Format

This is very non-standard, as they store a parameterised version of the affine decomposition. There are 12 parameters in total, and the order of the stored parameters are:

- 1-3 : Translation (t_x, t_y, t_z)
- 4-6 : Rotation angles $(\theta_x, \theta_y, \theta_z)$ (in degrees)
- 7-9 : Scalings (s_x, s_y, s_z)
- 10-12 : Skew angles (ϕ_1, ϕ_2, ϕ_3) (in degrees).

The default values (when not specified) for the scalings are 1.0 and for the skew angles are 0.0.
Creation of the affine matrix A_{umds} from the file (dof) parameters is done by:

$$A_{umds} = T \cdot K \cdot S \cdot R_x \cdot R_y \cdot R_z \tag{13}$$

where T is a translation matrix, K is a skew matrix, S is a scaling matrix and R_x, R_y, R_z are rotation matrices, defined by:

$$T = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \; K = \begin{bmatrix} 1 & \tan\phi_2 & 0 & 0 \\ \tan\phi_1 & 1 & \tan\phi_3 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \; S = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{14}$$

$$R_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x & 0 \\ 0 & \sin\theta_x & \cos\theta_x & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \; R_y = \begin{bmatrix} \cos\theta_y & 0 & \sin\theta_y & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \; R_z = \begin{bmatrix} \cos\theta_z & -\sin\theta_z & 0 & 0 \\ \sin\theta_z & \cos\theta_z & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{15}$$

6 MINC/MNI Conventions

6.1 Coordinates

Not known — assumedly standard data relation and voxel indexing from 0 to $N-1$.
Negative voxel dimensions in standard images are common.

6.2 Transformations

MINC/MNI transformations are from the source to the destination with both in *world* coordinates. Conversion to a standard world-world (FLIRT) transformation matrix is given by:

$$A_{flirt} = D_{dest} \cdot OX_{dest} \cdot SX_{dest} \cdot D_{dest}^{-1} \cdot A_{minc} \cdot D_{source} \cdot OX_{source}^{-1} \cdot SX_{source} \cdot D_{source}^{-1}. \tag{16}$$

where D is purely diagonal (ignores any origins), and SX represents a voxel swapping in the x direction. That is, if there are N_x voxels in the x direction:

$$SX = \begin{bmatrix} -1 & 0 & 0 & N_x - 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{17}$$

and OX represents a voxel origin offset. That is, if the voxel origin is at (o_x, o_y, o_z) then:

$$OX = \begin{bmatrix} 1 & 0 & 0 & o_x \\ 0 & 1 & 0 & o_y \\ 0 & 0 & 1 & o_z \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{18}$$

Warning: These transformations were derived from using the MNI305 standard image as the reference and so may not work for all combinations of negative voxel dimensions and origins.

6.3 File Format

Single line of text header followed by the first three rows of the affine matrix in standard ascii format. That is, it does not bother to store the final 0 0 0 1 row.