Orientation basics, Nifti 1.1 and Analyze 7.5 formats

(Maria del C. Valdés Hernández <M.Valdes-Hernan@ed.ac.uk>, 14.07.2014)

1.- Orientation basics

1.1.- DICOM

DICOM stands for Digital Imaging and Communications in Medicine, and it “is a standard for handling, storing, printing, and transmitting in medical imaging” (http://en.wikipedia.org/wiki/DICOM). Detailed information about this imaging standard can be found in its official website: http://dicom.nema.org/.

Of extreme importance for image processing is to understand the image attributes that DICOMs provide. The Image Plane Module (in page 409 of part 3 of the standard) defines them:

<table>
<thead>
<tr>
<th>Attribute name</th>
<th>Tag</th>
<th>Attribute description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pixel Spacing</td>
<td>(0028,0030)</td>
<td>Physical distance in the patient between the centre of each pixel, specified by a numeric pair: adjacent row spacing (delimiter) adjacent column spacing in mm</td>
</tr>
<tr>
<td>Image Orientation (Patient)</td>
<td>(0020,0037)</td>
<td>The direction cosines of the first row and the first column with respect to the patient</td>
</tr>
<tr>
<td>Image Position (Patient)</td>
<td>(0020,0032)</td>
<td>The x, y and z coordinates of the upper left hand corner (centre of the first voxel transmitted) of the image in mm</td>
</tr>
<tr>
<td>Slice Thickness</td>
<td>(0018,0050)</td>
<td>Slice thickness in mm</td>
</tr>
<tr>
<td>Slice Location</td>
<td>(0020,1041)</td>
<td>Relative position of the image plane expressed in mm</td>
</tr>
</tbody>
</table>

Focusing in the orientation, DICOM defines a term: “Reference Coordinates System” or RCS where the X direction is from the patient’s right hand side to his/her left’s, the Y direction is from the front (i.e. forehead) to the back and the Z direction is from the patient’s feet to the patient’s head:
Figure 1. Reference Coordinate System for the scanner

The X direction increases from **Right to Left** (of the patient), the Y direction is (i.e. increases) from **Anterior to Posterior** (of the patient), and the Z direction is (i.e. increases) from **Inferior to Superior** (of the patient).

As a note of caution:

In [http://freesurfer.net/fswiki/DICOM](http://freesurfer.net/fswiki/DICOM) (accessed on 02.04.2014) it writes: “the DICOM coordinate system is the LPS (Left-Posterior-Superior)”, which refers NOT to the origin, but to the OPPOSITE POINT, i.e. the one towards which the increase is made, according to what is defined in [http://medical.nema.org/dicom/2003/03_03PU.PDF](http://medical.nema.org/dicom/2003/03_03PU.PDF) and is represented in the following screen-captures from the Annex E Explanation of patient orientation (Normative) PS3.3 -2003, page 711-

Note: Figure 2 below contains captures from the DICOM normative where H means Head (i.e. Superior) and F means Feet (i.e. Inferior)
This is clarified in [http://www.slicer.org/slicerWiki/index.php/Coordinate_systems](http://www.slicer.org/slicerWiki/index.php/Coordinate_systems), which adds that the LPS system (Right to Left, Anterior to Posterior, Inferior to Superior) is used in DICOMs and by the ITK toolkit, while the RAS system (Left to Right, Posterior to Anterior, Inferior to Superior) is used by 3D Slicer.

An important consideration is given to the terms: “radiological convention or radiological orientation” and “neurological convention or neurological orientation” in relation to the way images are viewed (i.e. displayed). As Figure 3 shows, the radiologist’s right hand side corresponds to the patients’ left hand side, while it is the opposite for the neurologist.
1.2.- Analyze 7.5 format

The Analyze 7.5 format was developed by the Biomedical Imaging Resource at Mayo Clinic to display and manipulate medical images. One data item consists of two files: one with the actual voxel data in a binary format (with the filename extension .img) and other named the header file with the information about how to manipulate the binary image data (i.e. contains voxel size, number of voxels on each dimension, etc.). The header has 348 bytes, but when it is stored in the computer occupies 1Kbyte. Detailed information can be found in [http://eeg.sourceforge.net/ANALYZE75.pdf](http://eeg.sourceforge.net/ANALYZE75.pdf).

1.2.1.- MATLAB resources to manipulate Analyze7.5 format

The Image Processing Toolbox of MATLAB 12.0 and higher versions has incorporated 2 functions to manipulate Analyze 7.5 formatted files: **analyze75info** and **analyze75read**.

- **analyze75info**: returns the header information on a ‘struct’
- **analyze75read**: returns the image data in neurological orientation: RAS (Left to Right, Posterior to Anterior, Inferior to Superior), which is NOT the default orientation of Analyze 7.5 format. Note that this MATLAB function reads 3D and 4D data represented as (y,x,z,t). Therefore, for working with a data array in the default Analyze 7.5 orientation, it is necessary to flip left and right and perform an in-plane rotation 90 degrees counterclockwise.
A useful library for manipulating both nifti 1.1 (explained below) and Analyze 7.5 files is NIfTI, by Jimmy Shen, that can be downloaded from [http://www.mathworks.co.uk/matlabcentral/fileexchange/8797-tools-for-nifti-and-analyze-image](http://www.mathworks.co.uk/matlabcentral/fileexchange/8797-tools-for-nifti-and-analyze-image) (website accessed on 02.04.2014).

1.3.- Nifti-1.1 format

Nifti-1.1 is a data format that resulted from an adaptation of the Analyze 7.5 format. Detailed information can be found at: [http://brainder.org/2012/09/23/the-nifti-file-format/](http://brainder.org/2012/09/23/the-nifti-file-format/). Among its advantages over the original Analyze 7.5 format, the following can be mentioned:

- Affine coordinate definitions relating voxel index (i,j,k) to spatial location (x,y,z)
- Codes to indicate spatio-temporal slice ordering for fMRI
- Complete set of 8-128 bit + 256 data types
- Header and image files integrated in only one with extension .nii

The following table shows a comparison between Analyze 7.5 and nifti 1.1 headers and refers to hdrinfo as the field in the struct “header information”. Descriptions follow MATLAB programming convention style.

<table>
<thead>
<tr>
<th>Type</th>
<th>Field size</th>
<th>Offset Size (ana/nifti)</th>
<th>Field name Analyze 7.5</th>
<th>Field name nifti 1.1</th>
<th>Description Analyze 7.5</th>
<th>Description nifti 1.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>int32</td>
<td>4B</td>
<td>0B</td>
<td>sizeof_hdr</td>
<td></td>
<td>Size of the header. Must be 348 (bytes)</td>
<td>not used</td>
</tr>
<tr>
<td>char</td>
<td>10B</td>
<td>4B</td>
<td>data_type[10]</td>
<td>['dsr' '']</td>
<td>not used</td>
<td>not used</td>
</tr>
<tr>
<td>char</td>
<td>18B</td>
<td>14B</td>
<td>db_name[18]</td>
<td>[filename(1:17) 0]</td>
<td>not used</td>
<td>not used</td>
</tr>
<tr>
<td>int32</td>
<td>4B</td>
<td>32B</td>
<td>extents</td>
<td>16384</td>
<td>not used</td>
<td>not used</td>
</tr>
<tr>
<td>int16</td>
<td>2B</td>
<td>36B</td>
<td>session_error</td>
<td>hdrinfo.SessionError</td>
<td>not used</td>
<td>not used</td>
</tr>
<tr>
<td>char</td>
<td>1B</td>
<td>38B</td>
<td>regular</td>
<td>'r'</td>
<td>hdrinfo.Regular must be 'r' to indicate all images and volumes are the same size</td>
<td>not used</td>
</tr>
<tr>
<td>char</td>
<td>1B</td>
<td>39B</td>
<td>dim_info</td>
<td>'0'</td>
<td>not used</td>
<td>encodes directions (phase,frequency, slice). Possible values are 1, 2 or 3 for each. Spiral sequences have frequency and phase both encoded as 0.</td>
</tr>
<tr>
<td>int16</td>
<td>16B</td>
<td>40B</td>
<td>dim[8]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>dim = [4 x y z t 1 1 1];</td>
<td>or dim = [3 x y z 1 1 1 1];</td>
<td>NIFTI intent code (complemented by the 3 fields below)</td>
<td></td>
</tr>
<tr>
<td>char</td>
<td>2B</td>
<td>56B/68B</td>
<td>voxel_units</td>
<td>intent_code</td>
<td>'mm'</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>hdrinfo.VoxelUnits</td>
<td></td>
<td></td>
</tr>
<tr>
<td>float</td>
<td>4B</td>
<td>58B/56B</td>
<td>intent_p[1]</td>
<td>fwrite(fid,zeros(1,12), 'char')</td>
<td>The intent fields are codes to indicate the nature of the data, distribution, etc.</td>
<td></td>
</tr>
<tr>
<td>float</td>
<td>4B</td>
<td>62B/60B</td>
<td>intent_p[2]</td>
<td>not used</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Variable Type</td>
<td>Size</td>
<td>Description</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------</td>
<td>------</td>
<td>-------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>float</td>
<td>4B</td>
<td>intent_p/3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>int16</td>
<td>2B</td>
<td>TYPE datatype</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>int16</td>
<td>2B</td>
<td>bitpix</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>float</td>
<td>32B</td>
<td>pixdim[8]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>float</td>
<td>4B</td>
<td>voxel_offset</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>float</td>
<td>4B</td>
<td>scl_slope</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>int16</td>
<td>2B</td>
<td>slice_end</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>char</td>
<td>1B</td>
<td>slice_code</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>char</td>
<td>1B</td>
<td>xyzt_units</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>float</td>
<td>4B</td>
<td>cal_max</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>float</td>
<td>4B</td>
<td>cal_min</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>int32/float</td>
<td>4B</td>
<td>compresed_slice_duration</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>int32/float</td>
<td>4B</td>
<td>verified/orange</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>int32</td>
<td>4B</td>
<td>glmax</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>int32</td>
<td>4B</td>
<td>glmin</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>


Posted on 23. September. 2012 by A. M. Winkler

```c
switch hdrinfo.ImgDataType
  case 'DT_NONE'
    TYPE = 0;
  case 'DT_UNKNOWN'
    TYPE = 0;
  case 'DT_BINARY'
    TYPE = 1;
  case 'DT_UNSIGNED_CHAR'
    TYPE = 2;
  case 'DT_SIGNED_SHORT'
    TYPE = 4;
  case 'DT_SIGNED_INT'
    TYPE = 8;
  case 'DT_FLOAT'
    TYPE = 16;
  case 'DT_COMPLEX'
    TYPE = 32;
  case 'DT_DOUBLE'
    TYPE = 64;
  case 'DT_RGB'
    TYPE = 128;
  case 'DT_ALL'
    TYPE = 255;
end
```

UNKNOWN = 0
BINARY (1 bit) = 1
UNSIGNED_CHAR (1 byte) = 2
SIGNED_SHORT (2 bytes) = 4
SIGNED_INT (4 bytes) = 8
FLOAT (4 bytes) = 16
COMPLEX (8 bytes) = 32
DOUBLE (8 bytes) = 64
RGB (3 bytes) = 128
ALL = 255
signed char (1 byte) = 256
unsigned short (2 bytes) = 512
unsigned int (4 bytes) = 768
long long (8 bytes) = 1024
unsigned long long (8 bytes) = 1280
long double (16 bytes) = 1536
double pair (16 bytes) = 1792
long double pair (62 bytes) = 2048
RGBA (4 bytes) = 2304

```c
int16 2B 72B bitpix
int16 2B 74B slice_start 0 (first slice index. Normally it is zero)
float 32B 76B pixdim[8] v1 = hdrinfo.PixelDimensions(1);
v2 = hdrinfo.PixelDimensions(2);
v3 = hdrinfo.PixelDimensions(3);
v4 = hdrinfo.PixelDimensions(4);
pixdim = [v1 v2 v3 v4 0 0 0 0]; % Field for grid spacing (unit per dimension).
float 4B 108B vox_offset
```
```plaintext
struct data_history -> fseek(fid, 148, 'bof');

| char | 80B | 148B | descr[80] | hdrinfo.Descriptor | % any text up to 80 characters, for example: 'libBRIC file'
|------|-----|------|-----------|---------------------|
| char/short | 18/2B | 252B | orient | qform | switch hdrinfo.Orientation
| uint16/short | 10B/2B | 253B/254B | originator | sform | originator = [0 0 0 0 0];
| char/float | 85B/4B | 263B/256B | quatern_b | fwrite (fid,zeros(1,81),'char'); % These are the fields for patient and scan details, which we fill with 0 for anonymising the scan
| /float | /4B | 260B | quatern_c | quatern_c parameter (see qform)
| /float | /4B | 264B | quatern_d | quatern_d parameter (see qform)
| /float | /4B | 268B | qoffset_x | Quaternion x shift
| /float | /4B | 272B | qoffset_y | Quaternion y shift
| /float | /4B | 276B | qoffset_z | Quaternion z shift
| /float | /16B | 280B | srow_x[4] | 1st row affine transform
| /float | /16B | 296B | srow_y[4] | 2nd row affine transform
| /float | /16B | 312B | srow_z[4] | 3rd row affine transform
| /char | /16B | 328B | intent_name[16] | Name or meaning of the data (string of 16 characters)
| char | 4B | 344B | magic[4] | [0 0 0 0] or ['ni1' 0] or [6EH 69H 31H 00H] or ['n+1' 0] or [6EH 2BH 31H 00H]
```