niidotmask.m: process NIfTI “dotmask” file

Syntax:

```
1) paths = niidotmask('in', in, 'oris', oriS, 'orif', oriF, 'outtype', 'mask', ...
                      'outdir', outDir, 'outname', outName)
2) slice = niidotmask('in', in, 'oris', oriS, 'orif', oriF, 'outtype', 'slice')
3) fics  = niidotmask('in', in, 'oris', oriS, 'orif', oriF, 'outtype', 'fics')
```

Description:

1) creates NIfTI logical files with masks as defined in the NIfTI dotmask file. The NIfTI dotmask is a logical file where the only 1-valued voxels are extreme points that define rectangles around regions of interest. There are several ways to define the dotmask file:

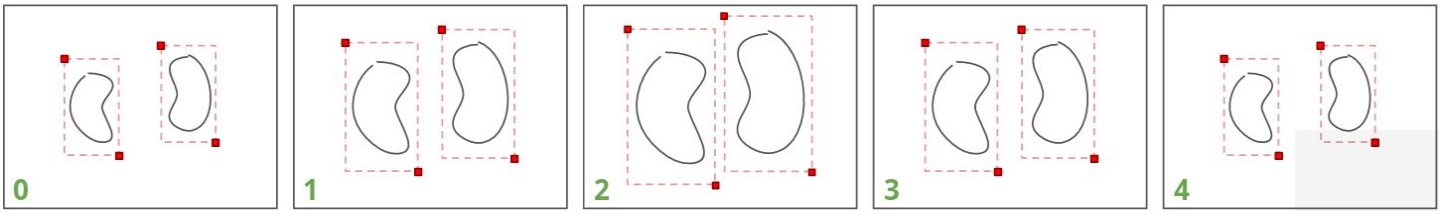

Fig. 1

Fig. 1 depicts the creation of a dotmask file (e.g. in FSLview [1]) for a volume where all slices are included with specific masks for each. The small red squares correspond to 1-valued voxels that are defined in the dotmask file. The dashed lines represent the corresponding boundaries of the mask given by the files created by this function when provided with the dotmask file and when using 'outtype'='mask'. In this case, the binary masks in the output mask files are as shown in Fig. 2.

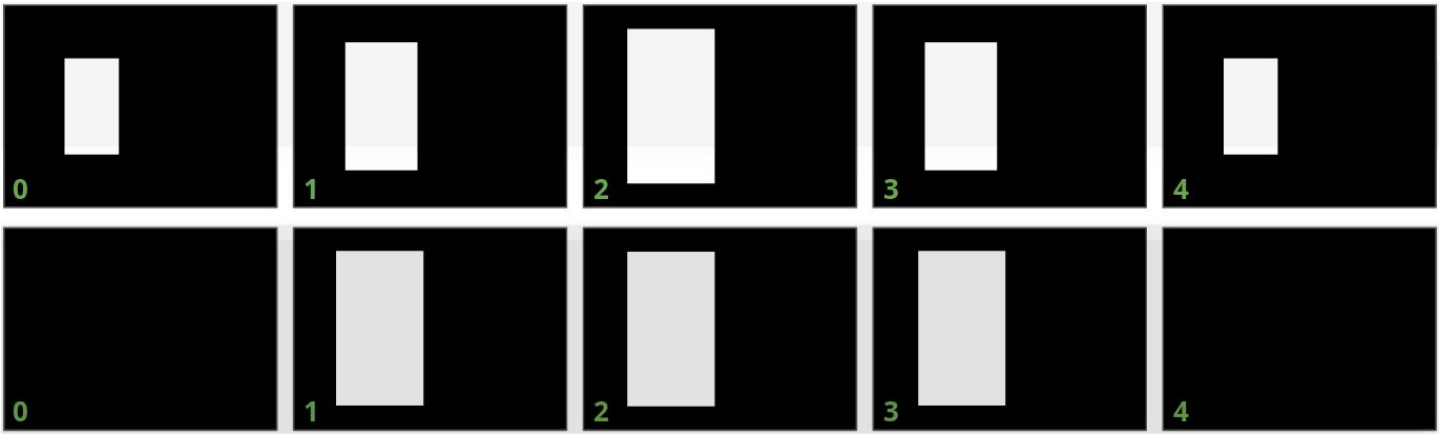

Fig. 2 top: mask file 1 of 2; bottom: mask file 2 of 2

Say that we want to create a dotmask file where only slices 1,2,3 (see Note 1) are included, and all have the same mask as in slice 2 above. Defining this particular dotmask file would look like Fig. 3 and the corresponding mask files like Fig. 4.

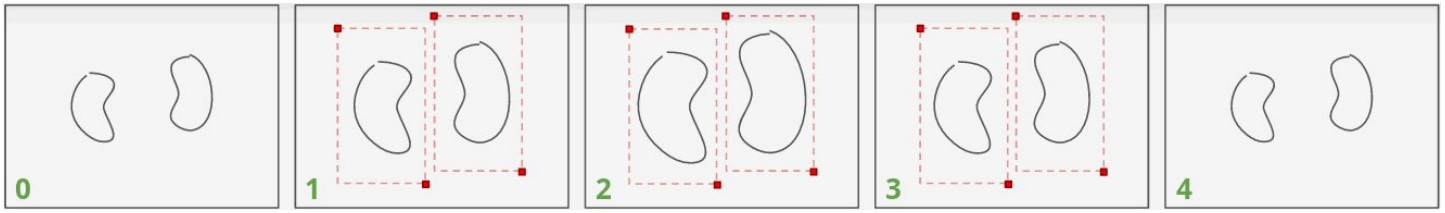

Fig. 3

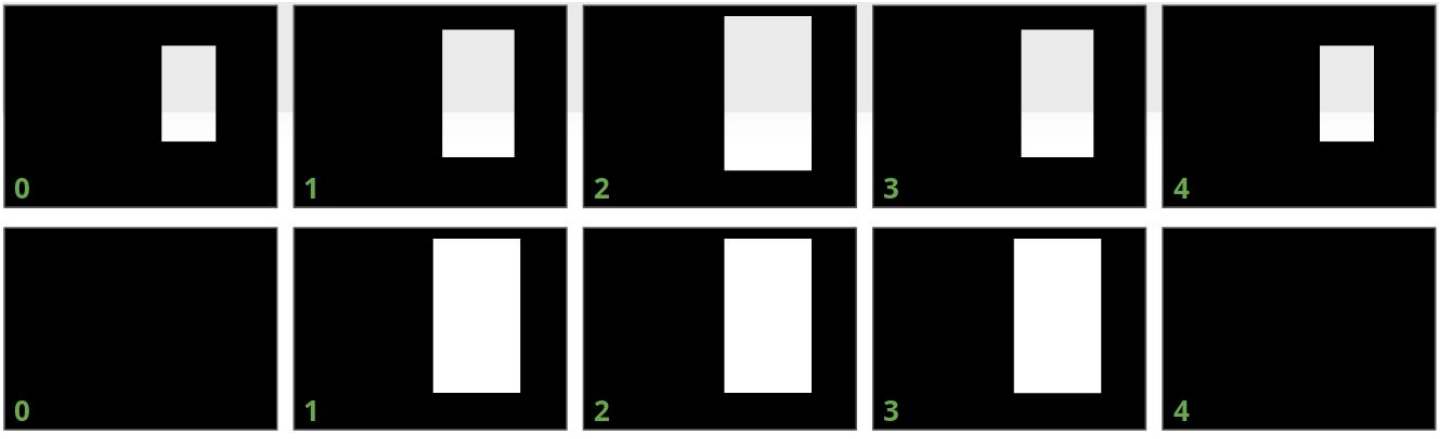

Fig. 4 top: mask file 1 of 2; bottom: mask file 2 of 2

2) allows one slice to be highlighted by adding one 1-valued voxel to the boundary of the desired slice. Say in the example above we wanted to highlight the central slice (slice 2). Defining dotmask include a new voxel in the boundary of the slice of interest (see blue voxel in Fig. 5. No mask files will be created. The only output would be sliceIdx = 2 in this case (See Note 1).

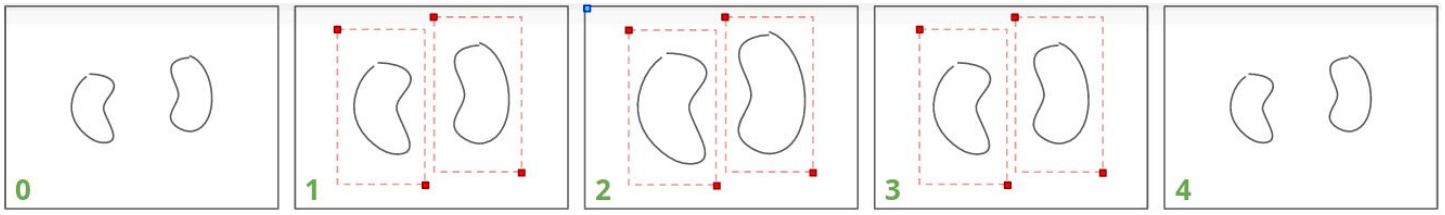

Fig. 5

3) Instead of creating NIfTI mask files, the output will be sets of [1x6] coordinate vectors that can be directly fed to fslroi to crop the NIfTI file used to generate dotmask, as specified in dropmask to crop NIfTI datasets with FSL's fslroi function [2].

Inputs:

```
------------------------------- MANDATORY -------------------------------
<in>       string  :  path to dotmask file
<oris>     cell    :  start orientation descriptor (cell of strings) - see volori.m
<orif>     cell    :  final orientation descriptor (cell of strings) - see volori.m
<outtype>  string  :  string which can be:
                          - mask  : to create NIfTI mask files
                          - slice : to retrieve the highlighted slice
                          - fics  : compute "[f]slroi [i]nput [c]oordinate[s]"
------------------------------- OPTIONAL --------------------------------
<outdir>   string  :  directory where created NIfTI mask files are to be saved
                      (only allowed when <outtype> is 'mask')
<outname>  string  :  base name for to-be-created NIfTI mask files
                      (only allowed when <outtype> is 'mask')
                      Example: say i) <outname> is 'mask' and ii) the dotmask
                      file defines two masks: the absolute paths of the
                      output files will be
                      >> fullfile(outdir, 'mask_01of02')
                      >> fullfile(outdir, 'mask_02of02')
-------------------------------------------------------------------------
```

Outputs:

```
1) Depending on the input argument <outtype>:
   - If <outtype> is 'mask':
          maskPath: full paths to created NIfTI mask files
   - If <outtype> is 'slice'
          highlightIdx: index of slice highlighted in dotmask
   - If <outtype> is 'fics'
          fics: cell of strings with [f]slroi [i]nput [c]oordinate[s]
                each element of the cell is a [1x6] vector (in string
                format to use in the terminal) with format:
                    <xmin> <xsize> <ymin> <ysize> <zmin> <zsize>
                 which defines one cropping operation
```

Notes/Assumptions:

1) Unlike MATLAB, this function follows the same indexing convention as FSL, i.e. 0-indexing (since as of now my plan is to create dotmask files using FSLview).
2) This function assumes that we are moving laterally (left<-->right) in dotmask in the column direction (more info in the subfunction named processmask.m)

References:

[1] https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/FslView
[2] https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/Fslutils
[3] mathworks.com/matlabcentral/fileexchange/8797

Required functions:

1) load_untouch_nii.m (available in [3])
2) volori.m
3) save_untouch_nii.m (available in [3])
4) m2fsl.m

fnery, 20180228: original version