# ISISD and IReSISD User Manual

This document explains how to set up the parameters of ISISD's and IReSISD's xml configuration file. For detailed explanation of the algorithms and on the choice of their parameters, please refer to our publications:

Adeline Paiement, Majid Mirmehdi, Xianghua Xie, Mark Hamilton: **Integrated Segmentation and Interpolation of Sparse Data**. *IEEE Transactions on Image Processing,* Vol. 23, Issue 1, pp. 110-125, 2014.

Adeline Paiement, Majid Mirmehdi, Xianghua Xie, Mark Hamilton: **Registration and Modeling From Spaced and Misaligned Image Volumes**. *IEEE Transactions on Image Processing,* in press.

We welcome any comment and suggestion that would allow to improve this guide.

# 1. Data parameters

These parameters contain all the information required by IReSISD to load the data. Compulsory parameters are:

* datatype: There is a choice between "dicom", "dicom\_cardiac", "nifti", "nrrd", and "2D". dicom\_cardiac is a special case of dicom for cine cardiac MRIs, which is better suited for handling multiple timeframes that contain a periodic movement. dicom\_cardiac will also attempt to differentiate short-axis (SA) and long-axis (LA) slices, although it may also be used on cine cardiac MRIs that do not contain LA (axial datasets) or SA (radial datasets) slices.  
  "2D" is for 2D images or temporal sequences of 2D images in format supported by OpenCV, i.e. bmp, pbm, pgm, ppm, sr, ras, jpeg, jpg, jpe, jp2, tiff, tif, and png.
* directory: path (absolute or relative) to the folder containing the data files

The following parameters are optional and take a default value if not provided:

* periodic\_motion: In case of datasets with multiple timeframes, this option specifies if time should be treated as periodic. Default value: false. If datatype=dicom\_cardiac, its value is always true.
* accurate\_volume\_filling: An accurate volume filling produces a nicer display of the nrrd volumes VolumeData.nii and VolumeData\_aligned.nii, at the expense of a longer computing time. Default value: false.
* use\_first\_timeframe\_only: For datasets with multiple time frames, it is possible to use this option to process only one timeframe. This may be useful when selecting optimal parameters, as processing time is faster. Default value: false.
* serie\_ref\_defines\_volume\_size: When loading the data, the user is required to choose a reference serie (see 1.1). If this option is set to *true*, the spatial extent of the reference serie defines the size of the volume, and any other (non-reference) serie that covers a larger area will be cropped. If this option is set to *false*, the volume size is defined by the bounding box of all series.

## 1.1 Selection of reference serie

We call “serie” a set of slices having the same orientation and acquired during one run of the acquisition device. When loading the data, the user is required to choose a reference serie. This serie is used to define the resolution of the volume and its orientation. This selection is always done interactively in the command line prompt. When using cardiac MRIs ('dicom\_cardiac'), this reference serie should be a SA slice or stack of SA slices. If the dataset contains no SA serie, it is possible to select a LA serie instead, then specify that this is a LA serie and indicate which image orientation corresponds to the LA direction. ISISD will then infer the orientation of the SA plane and set the orientation of the volume accordingly.

## 1.2 Selection of an ROI

A limitation of the current implementation of IReSISD is that is cannot process too large volumes due to RAM usage. However, it is possible to select an ROI in order to limit the volume size.

Graphical interface

A basic graphical interface is implemented within IreSISD that allows drawing ROIs on the images of the dataset. To enable this option, set parameter “ROI\_selection” to true.

All slices of the dataset will be displayed in turn, allowing the user to draw a rectangle around the region to select. The rectangle may be drawn several times until the user is happy with his selection. He should then press 'v' (while in the image window) to validate the selection and go to the next image. Not all images require the ROI to be drawn on them. After all images have been processed, the final ROI is the 3D bounding box of all defined 2D ROIs. The size and offset of the ROI region is then displayed in the prompt window and should look like this:

ROI volume limits: x 244.994 397 y 210.694 309 z -49.7398 59.6723

new volume limits: x 0 152.006 y 0 98.3062 z 0 109.412

volume size : 153 99 110 1

Manually provided ROI coordinates

It is possible to enter manually the ROI coordinates into the xml file in order to avoid repeating the drawing process every time we process the dataset. This may be done by using the “volume size” and “ROI volume limits” values obtained at the previous “graphical interface” stage. They should be provided in the xml file using the following parameters:

* <volume\_size>  
   <x>153</x>  
   <y>99</y>  
   <z>110</z>  
  </volume\_size>
* <ROI\_offset>  
   <x>245</x>  
   <y>211</y>  
   <z>-50</z>  
  </ROI\_offset>

“ROI\_offset” takes the lower values of “ROI volume limits” displayed on the prompt.

Note: If an ROI is defined in this way, the option “ROI\_selection” is overridden and has no effect.

## 1.3 Manual registration of the slices

It is possible to provide a registration offset for the slices that make up the dataset. This offset will be applied once to the slices' positions and will remain unchanged for the rest of the data processing. If (active) registration is used (see Section 6), the automatically computed registration will be independent of this offset.

The position offset for the slices may be specified as the following:

* <registration>  
   <serie description="SA">  
   <slice number="5">  
   <translation>  
   <x>0</x> <!-- in pixels --> <y>0</y>  
   <z>0</z>  
   </translation>  
   <rotation>  
   <x>0</x> <!-- in radians -->  
   <y>0</y>  
   <z>0</z>  
   </rotation>  
   </slice>  
   </serie>  
  </registration>

If not provided, this offset is null by default.

# 2. RBF parameters

These parameters are used to define the RBF(s) used in ISISD.

* beta: It is usually good practice to set it to the number of dimensions of the dataset.
* psi: This parameter defines a single RBF. It may be used several times to define more than one RBF when the double or mixed schemes of ISISD are used (see section 5). It contains the following sub-parameters:
* index: This is the index of the individual RBF. It is only useful when several RBFs are defined.
* gamma: This is the main RBF parameter, which defines its “slope”.
* dirac\_width: This sets the width of the approximation function for the Dirac function used in ISISD. Default value: 1.
* borders: Borders (in pixels) may be added around the processed volume as a padding that helps preventing the contour to leek from one volume edge to the opposite one. Indeed, due to the use of a FFT, the volume is periodic and opposite edges correspond to the same position in space (or time). While this is desirable for the time dimension in case of cardiac MRIs, this can be quite inconvenient for the spacial dimensions. Padding helps dealing with this by adding a buffer with no data where the contour has no reason to go. Default value: [20, 20, 20, 20].

# 3. Level set parameters

These parameters define the initial level set function.

* phi: The parameters of an individual level set are defined in the <phi> parameter. Several level set functions may be created by defining several <phi> parameters. In that case, parameter <index> must be specified to identify the different level set functions:  
  <phi>  
   <index>1</index>  
  </phi>  
  <phi>  
   <index>2</index>  
  </phi>
* initialisation: defines the initial shape of the level set surface. Choice from 'circle' (for drawing circles on the images. This creates a sphere in 3D/4D.), 'nifti' (the initial values of the level set are loaded from a nifti file), or 'mask' (the initial shape of the level set contour is loaded from a black&white image file). Default value: circle.

If the 'nitfi' initialisation option is selected, parameter ' nifti\_file' should be provided and contain the path to the nifti file:

<phi>  
 <nifti\_file>path\_to\_nifti\_file</nifti\_file>  
 </phi>

Similarly, if the 'mask' initialisation option is selected, parameter ' mask\_file' should be provided and contain the path to the folder that contains mask files:

<phi>  
 <mask\_file>path\_to\_mask\_file(s)\_folder</mask\_file>  
 </phi>  
There should be one mask file per time frame, and their file names should reflect their ordering. The format should be an image format readable by OpenCV, i.e. bmp, pbm, pgm, ppm, sr, ras, jpeg, jpg, jpe, jp2, tiff, tif, and png.

If the 'circle' initialisation option is selected, the user has a choice between using a graphical interface to draw circles on the images, or specifying the coordinates and radius of circles in the xml file.

Graphical interface  
A basic graphical interface is implemented within IreSISD that allows drawing circles on the images of the dataset. If no <phi> field is provided, the user will first be asked how many level set should be created and initialised. If <phi> fields exist but no circle is defined, the graphical interface will be used.

All slices of the dataset will be displayed in turn, allowing the user to draw circles in the region to segment. The circles may be drawn several times until the user is happy with his selection. He should then press 'v' (while in the image window) to validate the selection. Several circles may be drawn on the same image, so the initial level set may be made up of several circles/spheres. Press 'n' go to the next image. Not all images require circles to be drawn on them.

When defining several level set functions, the prompt will ask the user to draw circles for all regions defined by the level sets in turn. For two level set, there are 4 regions: inside both level sets (region 0), inside level set 1 and outside level set 2 (region 1), inside level set 2 and outside level set 1 (region 2), and outside both level sets (region 3). The current implementation cannot support more than 2 level sets.

After all circles have been drawn, the level set functions will be initialised with spheres of centre's coordinates and radius defined by the drawn circles. Centres and radius are displayed in the prompt window and should look like this:  
 new circle/sphere for phi\_1 centred at (53, 44, 50, 0) with radius = 10.198  
 new circle/sphere for phi\_2 centred at (43, 60, 45, 0) with radius = 13.256

Manually provided circles

It is possible to enter manually the centre's coordinates and radius of circles into the xml file in order to avoid repeating the drawing process every time we process the dataset. This may be done by using the values obtained at the previous “graphical interface” stage. They should be provided in the xml file using the following parameters:

<phi>  
 <circle>  
 <center>  
 <x>150</x> <!-- in pixels --> <y>150</y>  
 <z>150</z>  
 <t>0</t>  
 </center>  
 <radius>10</radius> <!-- in pixels -->  
 </circle>  
 <circle>  
 <center>  
 <x>10</x>  
 <y>150</y>  
 <z>200</z>  
 <t>0</t>  
 </center>  
 <radius>5</radius>  
 </circle>  
 </phi>

# 4. Speed function parameters

Parameter fields 'speed\_function' define the modules that implement segmentation criteria and thus set the speed of the level set curve. There are two types of segmentation criteria modules:

* SF\_geom: geometric criteria, that only depend on the state of the level set curve and not on the data
* SF\_data: segmentation criteria based on image data.

The exact segmentation algorithm is specified by the following parameters:

* name: name of the segmentation algorithm. So far, only the Chan&Vese ('CV') and gradient vector flow ('GVF') algorithms are implemented for 'SF\_data', and a normalisation algorithm that normalises the gradient of the level set function to a norm of 1 ('normalisation') for 'SF\_geom'. More algorithms may be easily added to ISISD and IReSISD by inheriting fro; the SpeedFunction\_Data and SpeedFunction\_Geom classes in SpeedFunction.h/cpp.
* param: parameter of the segmentation algorithm. Not all algorithms need one, while some need several. In this last case, the definition order is used to identify them.

Example of definition of a 'SF\_data' field:

<speed\_function>  
 <SF\_data>  
 <name>GVF</name>  
 <param>1</param> <!-- optional - as many as needed -->

</SF\_data>  
 </speed\_function>

Several segmentation criteria may be used, by declaring any number of 'SF\_geom' and 'SF\_data' fields. In this case, the following parameters should be defined:

* index: to identify the individual segmentation criteria module
* weight:to set and balance the relative weights of all the segmentation criteria modules. If this parameter is provided, it will be used for all data sequences. Otherwise, the user will be asked to interactively choose one weight value per sequence.

Example of definition of several 'SF\_data' and 'SF\_geom' fields:

<speed\_function>

<SF\_data>

<name>CV</name>  
 <index>1</index>  
 <weight>0.5</weight>  
 </SF\_data>

<SF\_geom>

<name>normalisation</name>  
 <index>2</index>  
 <weight>0.5</weight>  
 </SF\_geom>

</speed\_function>

# 5. Scheme parameter

This parameter allows choosing between the 'simple', 'double', and 'mixed' schemes of ISISD.

# 6. Registration parameters

If registration is needed (IReSISD), parameter 'registration' should be defined. If no 'registration' field is provided, no registration is applied and the dataset is processed by ISISD only.

The following parameters define the type of registration to be applied by IReSISD to the dataset:

* global\_variant: if true, the global variant of the registration method of IreSISD is used. If false, the local variant is used. Default value: false.
* selection\_of\_connected\_regions: This parameter is only needed when 'global\_variant' is true. It is ignored otherwise. If true, only speeds of regions connected to the active contour are used for registration, and speeds of other regions are discarded. Default value: false.
* slice\_wise: if true, slice-wise registration is performed, else stack-wise. Default value: true for dicom\_cardiac, false other data types.
* shifts\_in\_one\_plane\_at\_beginning: if '0': registration is always performed in all directions, if '1': registration is performed in the planes of the images only, and if '2': registration is performed in the reference plane only, which is the SA plane for 'dicom\_cardiac' type of data. Default value: 0.
* wait\_before\_full\_translation: if 'shifts\_in\_one\_plane\_at\_beginning' is different to 0, this parameter specifies the number of iterations to wait before performing a full translation in all directions. If not provided or set to -1, the waiting time is infinity. This parameter is ignored if 'shifts\_in\_one\_plane\_at\_beginning' = 0.
* wait\_before\_rotation: number of iterations to wait before aligning by rotation. During this time, alignment by translation is still performed. Default value: 0.

# 7. Display parameter

This parameter allows choosing the amount of display at runtime. 0: no display, 1: minimal display, 2: full display. Default value = 0.

# 8. Stop condition parameters

These parameters define the stopping conditions of IReSISD for each step of ISISD independently.

* step: The parameters of an individual step are defined in the <step> parameter. Two steps are performed when using the double ISISD scheme, thus two <step> parameters must be defined in this case. Parameter <index> identifies the different steps and may be omitted if only one step is used:  
  <step>  
   <index>1</index>  
  </step>  
  <step>  
   <index>2</index>  
  </step>

Two simple types of stopping conditions have been implemented: fixed number of iterations, and fixed number of stable iterations, where an iteration is “stable” when the contour does not change size significantly. The number of iterations is defined by the 'nb\_iterations' and 'nb\_stable\_iterations' parameters respectively:

<step>  
 <index>1</index>  
 <nb\_iterations>100</nb\_iterations>  
 </step>  
 <step>  
 <index>2</index>  
 <nb\_stable\_iterations>20</nb\_stable\_iterations>  
 </step>