Contents

1 Installation **2**
- 1.1 Requirements . . . 2
  - 1.1.1 to run the different codes: . . . 2
  - 1.1.2 the install libraries are necessary: . . . 2
  - 1.1.3 the following python libraries are necessary: . . . 3
  - 1.1.4 the following libraries are necessary: . . . 3
- 1.2 Installation . . . 4
  - 1.2.1 Previously . . . 4
  - 1.2.2 From version v2.0 . . . 4
- 1.3 Troubleshooting . . . 4

2 Running Astec **5**
- 2.1 The `parameters-example.py` file . . . 6
- 2.2 List of command line interfaces . . . 14
  - 2.2.1 `1-fuse.py` . . . 14
  - 2.2.2 `1.5-intraregistration-GC.py` . . . 17
  - 2.2.3 `1.5-intraregistration.py` . . . 17
  - 2.2.4 `2-mars.py` . . . 20
  - 2.2.5 `3-manualcorrection.py` . . . 20
  - 2.2.6 `4-astec.py` . . . 20
  - 2.2.7 `5-postcorrection.py` . . . 20
  - 2.2.8 `6-named.py` . . . 20
  - 2.2.9 `7-virtualembryo.py` . . . 20

3 Tutorial **21**
- 3.1 Starting with a toy embryo . . . 21
- 3.2 Fusing data . . . 22
- 3.3 Segmenting the first time point . . . 23
- 3.4 Correcting the first time point segmentation . . . 24
- 3.5 Segmentation propagation . . . 25

Chapter 1

Installation

ASTEC can run only on Linux system and was tested on:

- Ubuntu 14.04 64 bits

1.1 Requirements

In order to be able to compile the different source code you need to install several packages, mostly from the terminal application.

1.1.1 to run the different codes:

- python 2.7 or higher
  Installation:
  - Linux: Should be installed, refer with command line
    `'python -V'`
    to get the version or visit `http://php.net`
- and a C compiler
  Installation
  - Linux: Should be installed, refer with command line
    `'dpkg --list | grep compiler'`
    to get list of available C compiler version or install `gcc` with
    `'sudo apt-get install gcc'`

1.1.2 the install libraries are necessary:

- `pip`, an installer for python (`https://pypi.org/project/pip/`). Installation:
  - Linux: run command line
    `'sudo apt-get install python-pip python-dev build-essential'`
  - Others: see *Installation* from `https://pypi.org/project/pip/`
- `cmake`, a software to build source code (`http://www.cmake.org`)
  - Linux: run command line
    `'sudo apt-get install cmake'`
  - Others: see *Download* from `https://cmake.org/`

1.1.3 the following python libraries are necessary:

- numpy, scipy, matplotlib: different scientific packages for python (`http://www.numpy.org`, `http://www.scipy.org`, `http://matplotlib.org`).

  Installation

  - Linux: run command line

    `'sudo apt-get install python-numpy python-scipy python-matplotlib ipython ipython-notebook python-pandas python-sympy python-nose'`

- jupyter notebook: a open-source framework to share scientific code (`http://jupyter.org`)

  Installation:

  - Linux: run command line

    `'sudo pip install jupyter'`

- libhdf5-dev,cython,h5py a library to read/write hdf5 images (`https://www.hdfgroup.org/HDF5/`)

  Installation:

  - Linux: run the command lines for a global installation

    `'sudo apt-get install libhdf5-dev'`

    `'sudo pip install cython'`

    `'sudo pip install h5py'`

    or

    `'pip install --user cython'`

    `'pip install --user h5py'`

- pylibtiff a library to read/write tiff/tif images (`https://pypi.python.org/pypi/libtiff/`).

  Installation:

  - You can install the pylibtiff distributed with ASTEC

    `'cd path/to/Package/ASTEC/CommunFunctions/libtiff-0.4.0/; sudo python setup.py install'`

  - Alternatively:

    `'sudo pip install libtiff'`

    or

    `'pip install --user libtiff'`

1.1.4 the following libraries are necessary:

- `zlib`, a compression library (`http://www.zlib.net`)

  Installation (if not already installed)

  - Linux: run command line

    `'sudo apt-get install zlib1g-dev'`

- `libtiff`, `http://libtiff.org`

1.2 Installation

You need to clone the morpheme-privat project in your computer: run

‘cd <morpheme-privat_parent_directory>‘

‘git clone git+ssh://<username>@scm.gforge.inria.fr/gitroot/morpheme-privat/morpheme-privat.git‘

Please note that for this step, you may need to ask for an account on the inria gforge and for the access to the morpheme-privat project (for further information, contact gregoire.malandain@inria.fr)

Then you need to compile the dependencies LEMON, NIFTICLIB, TIFF in <morpheme-privat_parent_directory>/e by creating in each subrepository a build folder, going in it, and run:

‘ccmake ..‘

then configure, generate and quit (for NIFTICLIB, please take note that you should set the CMAKE_C_FLAGS to value '-fPIC' in the advanced mode ; you need therefore to press 't' to toogle advanced mode)

run ‘make -j<nb proc>‘ where <nb proc> is the number of processor require to compile the code Then go to <morpheme-privat_parent_directory>/vt directory and follow these instructions:

‘mkdir build‘

‘cd build‘

‘ccmake ..‘

Press 'c' to configure. At this step, set the links to the build folders of NIFTI, LEMON and TIFF. Press 'c' to re-configure. Press 'g' to generate and 'q' to quit.

Run ‘make -j<nb proc>‘ where <nb proc> is the number of processor require to compile the code

1.2.1 Previously

Since the vt library is compiled, you need to create the symbolic link of the folder <morpheme-privat_parent_directory>/ to ASTEC/CommunFunctions/cpp. in a terminal:

‘cd path/to/Package/ASTEC/CommunFunctions‘

ln -s <complete_path_of_vt_folder> cpp

1.2.2 From version v2.0

Since the vt library is compiled, you need to create the symbolic link of the folder <morpheme-privat_parent_directory>/ to ASTEC/CommunFunctions/cpp. in a terminal:

‘cd path/to/Package/ASTEC/CommunFunctions‘

ln -s <complete_path_of_bin_folder> cpp

1.3 Troubleshooting

Chapter 2

Running Astec

From version 2.0, the data architecture should be as follows

```
<EMBRYO>
├── RAWDATA
│   ├── LC
│   │   ├── Stack0000
│   │   └── Stack0001
│   └── RC
│       ├── Stack0000
│       └── Stack0001
├── FUSE
│   ├── FUSE_<EXP_1>
│   ├── FUSE_<EXP_2>
│   ├── ...
│   └── FUSE_RELEASE
├── SEG
│   ├── SEG_<EXP>
│   ├── ...
│   └── SEG_RELEASE
└── POST
    ├── POST_<EXP>
    ├── ...
    └── POST_RELEASE
```

`<SOMETHING>_RELEASE` sub-directories should content the "last version" validated by an expert. `<SOMETHING>_<EXP_x>` are folders containing the different experiments conducted at one given step (with different parameters/methods). After inspection of the results, one of them is supposed to be renamed as `<SOMETHING>_RELEASE` and the others can be deleted.

Particular files:

`nomenclature.py`: file fixing the set of naming rules in working directories

- this file should not be modified.

`parameters.py`: file defining the set of parameters useful for all the process steps (parameters are all prefixed with respect to the step they are used in).

- it is a "template" file which should be duplicated and whose copy can be modified by the user to its convenience (only the parameter values should be modified, not their name...).

The scripts of steps `1-fuse.py`, `2-mars.py`, `3-manualcorrection.py`, `4-astec.py`, `5-postcorrection.py` are executables, so that each astec step calling can be made as described here: For example, in order to launch the fusion step on an embryo called `"171107-Karine-St8"` one should:

1. Duplicate the file `<astec-package>/parameters.py` and rename it as a new file `<arbitrary-path>/parameters_ka`
2. Edit this new file `<arbitrary-path>/parameters_karine.py`to specify the desired value of each parameter related to the fusion step
3. In a terminal,

   ```
   $ cd <astec-package> # in order to be in the astec directory
   (/media/DATA/Codes/astec-package on Hermione)
   $./1-fuse.py --parameters <arbitrary-path>/parameters_karine.py
   --embryo-rep /media/DATA/171107-Karine-St8/
   ```

   (or equivalently,a shorter format)

   ```
   $./1-fuse.py -p <arbitrary-path>/parameters_karine.py -e
   /media/DATA/171107-Karine-St8/
   ```

At each Astec step execution, a copy of the parameters file as well as a log file are automatically generated in the target working directory.

For each Astec step, it is possible to display the help relative to the corresponding script by launching the script with the option `'--help'`. For example, for the fusion step:

In a terminal, launch the command line:

```
$ <astec-package>/1-fuse.py --help
```

The terminal displays the following message:

```
Usage:  1-fuse.py [options]
Options:
-h, --help show this help message and exit
-p FILE, --parameters=FILE
python file containing parameters definition
-e PATH, --embryo-rep=PATH
path to the embryo data
-q, --quiet don't print status messages to stdout
```

2.1 The `parameters-example.py` file

[**Pay attention**] The file included below may be not up-to-date with respect to the software package. Please refer to the file `parameters-example.py` of the ASTEC distribution.

```
1
2
3 # File for parameters and nomenclature settings
4
5 PATH_EMBRYO=''
6                 # Must be the path to the embryo data
7 # eg: '/media/DATA/171107-Karine-St8'
8 # Can also be set by providing option '-e' in the command line
9 EN=''
```

```
10                 # Embryo Name
11                 # CRBM naming format is YYMMDD-SaintOfTheDays-Stage
12 # eg: '171107-Karine-St8'
13 # (automatically extracted from PATH_EMBRYO if not provided)
14
15
16 EXP_FUSE=''
17             # Workspace for the considered step; this workspace
18 # is included in the repository corresponding to the
19 # ASTEC step ([FUSE|SEG|POST]) and prefixed by this
20 # repository name (eg. FUSE_EXP_NO_CROP or SEG_RELEASE)
21 # ----> use replaceDIR_EXP_FUSE(path_fuse_exp[...], EXP_FUSE)
22 # method from <astec-package>/nomenclature.py to get the right
23 # path for the given fusion experience
24 EXP_REG=''
25             #
26 EXP_MARS=''
27             #
28 EXP_SEG=''
29             #
30 EXP_POST=''
31             #
32
33
34 #TIME=''# Time-point of an embryo snapshot
35 #TIMEREF=''# For registration, time-point of reference snapshot
36 #TIMEFLO=''# For registration, time-point of floating snapshot
37
38 ############################
39 ### GENERAL PARAMETERS ###
40 ############################
41
42 begin=1
43                           # First time point
44 end=5
45                           # Last time point
46 delta=1
47                           # Delta between two time points (if one does not want
48  # to deal with every single time point) (default = 1)
49 target_resolution = .3
50                           # Isotropic resolution of the final fused and
51  # segmented images (default = 0.3)
52
53
54 ############################
55 ### RAWDATA DEFINITION ###
56 ############################
57
58 raw_ori = 'left'
59                                  # if im2 angle - im1 angle < 0 => right
60 raw_resolution = (.17, .17, 1.)
```

```
                                # Resolution of the raw images (here are the
# known values for 140317-Patrick-St8)
#raw_resolution = (.21, .21, 1.)
                                # Resolution of the raw images for Karine
raw_delay = 0
                                # If the time stamps in the folder are not the
# actual time stamps in the global movie
raw_mirrors = False
                                # Depends on the acquisition protocol, value
# can be set to True or False
#  - the standard value of this parameter is
#    False
#  - in case of axial symmetry between left
#    and right cameras, then set to True



##########################
### FUSION PARAMETERS ###
##########################

fusion_margin_x_0 = 40
                        # margin_x_0 [default=40]: parameter for margin of the
   # bounding box computed for the cropping of the
   # resampled image in 'left' x direction
fusion_margin_x_1 = 40
                        # margin_x_1 [default=40]: parameter for margin of the
       # bounding box computed for the cropping of the
       # resampled image in 'right' x direction
fusion_margin_y_0 = 40
                        # margin_y_0 [default=40]: parameter for margin of the
   # bounding box computed for the cropping of the
   # resampled image in 'top' y direction
fusion_margin_y_1 = 40
                        # margin_y_1 [default=40]: parameter for margin of the
   # bounding box computed for the cropping of the
   # resampled image in 'bottom' y direction
fusion_crop = True
                        # crop [default=True]:
                        # if False, then the resampled image is not cropped ;
   # if True, then image is cropped



#########################
### MARS PARAMETERS ###
#########################

# Modules choice
```

```
112 mars_method=1
113                         # 1 for 'Classic' method
114    # 2 for 'Gace' method
115 # General parameters for MARS segmentation
116 mars_sigma1 = 0.6
117                         # sigma 1 (0.6um) in real coordinates
118 mars_sigma2 = 0.15
119                         # sigma 2 (0.15um) in real coordinates
120 mars_h_min = 4
121                         # H min initialisation to ease correction
122 # Gace Parameters (if mars_method is set to 2):
123 # membrane_renforcement
124 mars_sigma_membrane=0.9
125                         # membrane enhancement parameter (in real units, a
126 # priori 0.9 um is a good choice for data like
127 # Patrick/Ralph/Aquila)
128 # anisotropicHist /!\ critical step
129 mars_sensitivity=0.99
130                         # membrane binarization parameter, /!\ if failure,
131 # one should enter in "manual" mode of the function
132 # anisotropicHist via activation of 'manual' option
133
134 mars_manual=False
135                         # By default, this parameter is set to False. If
136 # failure, (meaning that thresholds are very bad,
137 # meaning that the binarized image is very bad),
138   # set this parameter to True and relaunch the
139   # computation on the test image. If the method fails
140   # again, "play" with the value of manual_sigma...
141   # and good luck.
142 mars_manual_sigma=15
143                         # Axial histograms fitting initialization parameter
144 # for the computation of membrane image binarization
145 # axial thresholds (this parameter is used iif
146 # manual = True).
147 # One may need to test different values of
148 # manual_sigma. We suggest to test values between 5 and
149 # 25 in case of initial failure. Good luck.
150
151 mars_hard_thresholding=False
152                               # If the previous membrane threshold method
153   # failed, one can force the thresholding with a
154   # "hard" threshold applied on the whole image.
155   # To do so, this option must be set to True.
156 mars_hard_threshold=1.0
157                               # If hard_thresholding = True, the enhanced
158   # membranes image is thresholded using this
159   # parameter (value 1 seems to be ok for
160   # time-point t001 of Aquila embryo for example).
161
162 # Tensor voting framework
```

```
163 mars_sigma_TV=3.6
164                       # parameter which defines the voting scale for membrane
165   # structures propagation by tensor voting method (real
166   # coordinates).
167     # This parameter shoud be set between 3 um (little cells)
168     # and 4.5 um(big gaps in the binarized membrane image)
169 mars_sigma_LF=0.9
170                       # Smoothing parameter for reconstructed image (in real
171   # coordinates). It seems that the default value = 0.9 um
172   # is ok for classic use.
173 mars_sample=0.2
174                       # Parameter for tensor voting computation speed
175   # optimisation (do not touch if not bewared)
176
177
178
179 ####################################
180 ### MANUAL CORRECTION PARAMETERS ###
181 ####################################
182
183 mancor_seg_file=''
184                        # segmentation file to be manually corrected
185    # If not provided, then looking for the output of MARS
186 mancor_mapping_file=''
187                        # path to mapping file for manual correction of the
188    # mars segmentation. See above the syntax of this file.
189    # - 1 line per label association
190    # - background label has value 1
191    # - the character '#' denotes commented lines
192    # See file "mapping.txt" in the astec project to get an
193    # example
194 '''
195 # EXAMPLE OF mancor_mapping_file CONTENT:
196 # here the input label 8 will be mapped with new value 7, etc...
197 8 7
198 9 2
199 4 64
200 29 23
201 # ... etc ...
202 # background labels
203 30 1
204 89 1
205 '''
206
207
208
209 ############################################
210 ### SEGMENTATION PROPAGATION PARAMETERS ###
211 ############################################
212
213
```

```
214 # Modules choice
215
216 astec_membrane_reconstruction_method=0
217 # Membrane reconstruction module choice
218 # 0 for 'Classic' method
219 # 1 for 'Glace' method
220 # 2 for 'Gace' method
221 # If not set or set to 0, the input fused image is not processed for
222 # membrane structures enhancement.
223 # If set to 1, the GLACE reconstruction method is going to be called
224 # If set to 2, the GACE reconstruction method is going to be called
225
226 astec_fusion_u8_method=0
227 # Selection of the method which converts the fused image into a 8 bits
228 # images for the segmentation propagation.
229 # If set to 0 (default), calling the historical "to_u8" method
230 # If set to 1, calling the mc-adhocFuse program which enhances the fused
231 # image while converting it to u8 knowing the semgnetation propagation from
232 # previous time point
233
234 astec_flag_hybridation=False
235 # If set to True and if the membrane_reconstruction_method parameter is
236 # provided and not equal to 0, then the reconstructed gray level image
237 # used for the segmentation propagation framework is goind to be an hybridation
238 # between the original fused image and the result of image reconstruction by
239 # the specified method.
240
241 astec_keep_reconstruct_files=False
242 # Set it to True in order to keep a copy
243 # Flag enabling to keep a copy of graylevel files provided to the watershed
244
245
246
247 # General parameters for segmentation propagation
248 astec_sigma1 = 0.6
249                             # sigma 1 (0.6um) in real coordinates
250 astec_sigma2 = 0.15
251                             # sigma 2 (0.15um) in real coordinates
252 astec_h_min_min = 2
253                             # H min initialisation to ease correction
254 astec_h_min_max = 18
255                             # H min initialisation to ease correction
256
257 # Glace Parameters (if astec_membrane_reconstruction_method is set to 1 or 2):
258 # membrane_renforcement
259 astec_sigma_membrane=0.9
260                         # membrane enhancement parameter (in real units, a
261 # priori 0.9 um is a good choice for data like
262 # Patrick/Ralph/Aquila)
263 # anisotropicHist /!\ critical step
264 astec_sensitivity=0.99
```

```
                          # membrane binarization parameter, /!\ if failure,
# one should enter in "manual" mode of the function
# anisotropicHist via activation of 'manual' option

astec_manual=False
                          # By default, this parameter is set to False. If
# failure, (meaning that thresholds are very bad,
# meaning that the binarized image is very bad),
  # set this parameter to True and relaunch the
  # computation on the test image. If the method fails
  # again, "play" with the value of manual_sigma...
  # and good luck.
astec_manual_sigma=15
                          # Axial histograms fitting initialization parameter
# for the computation of membrane image binarization
# axial thresholds (this parameter is used iif
# manual = True).
# One may need to test different values of
# manual_sigma. We suggest to test values between 5 and
# 25 in case of initial failure. Good luck.

astec_hard_thresholding=False
                               # If the previous membrane threshold method
  # failed, one can force the thresholding with a
  # "hard" threshold applied on the whole image.
  # To do so, this option must be set to True.
astec_hard_threshold=1.0
                               # If hard_thresholding = True, the enhanced
  # membranes image is thresholded using this
  # parameter (value 1 seems to be ok for
  # time-point t001 of Aquila embryo for example).

# Tensor voting framework
astec_sigma_TV=3.6
                        # parameter which defines the voting scale for membrane
  # structures propagation by tensor voting method (real
  # coordinates).
    # This parameter shoud be set between 3 um (little cells)
    # and 4.5 um(big gaps in the binarized membrane image)
astec_sigma_LF=0.9
                        # Smoothing parameter for reconstructed image (in real
  # coordinates). It seems that the default value = 0.9 um
  # is ok for classic use.
astec_sample=0.2
                        # Parameter for tensor voting computation speed
  # optimisation (do not touch if not bewared)
astec_rayon_dil=3.6
                        # dilatation ray for propagated ROI from time t to t+1
  # (default: 3.6, in real coordinates)

# Fused image conversion parameters (if astec_fusion_u8_method is set to 1)
```

```
316 astec_min_percentile=0.01
317                       # mc-adhocFuse parameter of type %f (default: 0.01)
318 astec_max_percentile=0.99
319                       # mc-adhocFuse parameter of type %f (default: 0.99)
320 astec_min_method='cellinterior'
321                       # mc-adhocFuse param. (default: 'cellinterior')
322       # taken in global|cell|cellborder|cellinterior|voxel
323 astec_max_method='cellborder'
324                       # mc-adhocFuse parameter (default: 'cellborder')
325   # taken in global|cell|cellborder|cellinterior|voxel
326 astec_sigma_hybridation=5.0
327                       # mc-adhocFuse parameter of type %f (default: 5.0)
328
329 # Default parameters (for classical use, default values should not be changed)
330 astec_RadiusOpening=20
331                       # (using the approximation of a sphere of radius 20
332   # voxels as a structuring element)
333 astec_Thau= 25
334                       # s(c)=h2+(c).N2(c) >t identical
335 astec_MinVolume=1000
336                       # Miss Suppressing cells with to small volumes (not
337   # yet in supdata)
338 astec_VolumeRatioBigger=0.5
339                       # If a cell in St+1 is at least 50% bigger than its
340   # progeny in St+1,
341 astec_VolumeRatioSmaller=0.1
342                       # Cells in St+1 that are 10% or smaller than their
343   # equivalent in St+1 are tagged for correction
344 astec_MorphosnakeIterations=10
345                       # Then, an active contour algorithm is applied
346       # using the dilated shape of c, obtained by
347   # iterating 10 times
348 astec_NIterations=200
349                       # The algorithm is applied up to stability
350   # (at th voxels) or after n iterations
351   # (in our case th = 103 and n = 200).
352 astec_DeltaVoxels=10**3
353                       # y (at th voxels)
354 astec_Volum_Min_No_Seed=100
355                       # Then, if the volume of c is greater than 100
356   # voxels (2.7 um3)
357 astec_nb_proc=10
358                       # Number of processor ...
359 astec_nb_proc_ace=7
360                       # number of processors for ACE (7 is recommanded)
361
362
363
364 ###################################
365 ### POST-CORRECTION PARAMETERS ###
366 ###################################
```

```
367
368 postcor_Volume_Threshold=10000
369                       # volume low threshold for final cells
370   # (in voxels)
371 postcor_Soon=True
372                       # True if the cell life span has to be
373   # taken into account
374 postcor_ShortLifespan=25
375                       # (length < SL time points, in our case
376   # SL = 10)
377 postcor_PearsonThreshold=0.9;
378                       # If they are anticorrelated (Pearson
379   # correlation under -0.9)
380
381
382
```

2.2 List of command line interfaces

2.2.1 `1-fuse.py`

The fusion is made of the following steps.

1. Optionally, a slit line correction. Some Y lines may appear brighter in the acquisition and causes artifacts in the reconstructed (i.e. fused) image. By default, it is not done.
2. a change of resolution in the X and Y directions only (Z remains unchanged). It allows to decrease the data volume (and then the computational cost) if the new pixel size (set by `target_resolution`) is larger than the acquisition one.
3. Optionally, a crop of the resampled acquisitions. It allows to decrease the volume of data, hence the computational cost. The crop is based on the analysis of a MIP view (in the Z direction) of the volume, and thus is sensitive to hyper-intensities if any. By default, it is done.
4. Optionally, a mirroring of the 'right' image. It depends on the value of `raw_mirrors` variable.
5. Linear registration of the 3 last images on the first one (considered as the reference). The reference image is resampled again, to get an isotropic voxel (whose size is given by `target_resolution`), i.e. the voxel size is the same along the 3 directions: X, Y, Z.
6. Linear combination of images, weighted by an ad-hoc function.
7. Optionally, a crop of the fused image, still based on the analysis of a MIP view (in the Z direction). By default, it is done.

Please refer to the file `parameters-example.py` which describes all useful variables to control these steps.

2.2.1.1 Basic use

Assume that the raw data are organized as follows

```
180927-Vincent-Nodal-St8
└── RAWDATA
    └── LC
        └── Stack0000
```

```
        |
        |   |
        |   |-- Cam_Left_00000.zip
        |   |-- Cam_Left_00001.zip
        |   |-- ...
        |   `-- Stack0001
        |       |-- Cam_Left_00000.zip
        |       |-- Cam_Left_00001.zip
        |       |-- ...
        `-- RC
            |-- Stack0000
            |   |-- Cam_Right_00000.zip
            |   |-- Cam_Right_00001.zip
            |   |-- ...
            |   `-- Stack0001
            |       |-- Cam_Right_00000.zip
            |       |-- Cam_Right_00001.zip
            |       |-- ...
```

Then a simple parameter file (see `astec-parameters-examples/parameters-fusion-vincent-1.py`) allows to perform the fusion

```
# path to the embryo data

PATH_EMBRYO = '/Users/greg/COLLABORATIONS/DIGEM/DATA/TEST/180927-Vincent-Nodal-St8'

# Embryo Name

EN = '180927-Vincent-Nodal-St8'

# time points to be processed
# begin: first time point
# end: last time point
# delta: Delta between two time points (if one does not want
#        to deal with every single time point) (default = 1)

begin = 0
end = 0
delta = 1

# output image format
#
# default_image_suffix is for all images (including auxiliary ones)

default_image_suffix = 'mha'

# raw_ori: image orientation, value is in {'right','left'}
#   if im2 angle - im1 angle < 0 => raw_ori = 'right'
# raw_mirrors: value is in {True, False}
#  - the standard value of this parameter is False
#  - in case of axial symmetry between left
#    and right cameras, then set to True
# raw_resolution: acquisition voxel size
# raw_delay: increment to to be added to the time values
#   (values in range [begin,end]) when generating the
```

```
#   fused image names (default is 0)

raw_ori = 'left'
raw_mirrors = False
raw_resolution = (.195, .195, 1.)
raw_delay = 0

# Isotropic resolution of the final fused image

target_resolution = .3
```

2.2.1.2 Other organization of data

In case of the data organization does not follow the one described in section 2.2.1.1, the variables `DIR_RAWDATA`, `DIR_LEFTCAM_STACKZERO`, `DIR_RIGHTCAM_STACKZERO`, `DIR_LEFTCAM_STACKONE`, and `DIR_RIGHTCAM_STACKONE` allow a finer control of the directory names, while the variables `acquisition_leftcam_image_prefix` and `acquisition_rightcam_image_prefix` allow a finer control of the file names.

For instance, assume that the data ar organized as below:

```
180927-Vincent-Nodal-St8
└── RAWDATA
    ├── stack_0_channel_0
    │   ├── Cam_Left_00000.zip
    │   ├── Cam_Left_00001.zip
    │   ├── ...
    │   ├── Cam_Right_00000.zip
    │   ├── Cam_Right_00001.zip
    │   └── ...
    └── stack_1_channel_0
        ├── Cam_Left_00000.zip
        ├── Cam_Left_00001.zip
        ├── ...
        ├── Cam_Right_00000.zip
        ├── Cam_Right_00001.zip
        └── ...
```

Then, fusion can be achieved by adding the following lines to the parameter file of section 2.2.1.1 (see `astec-parameters-examples/parameters-fusion-vincent-2.py`).

```
DIR_LEFTCAM_STACKZERO = 'stack_0_channel_0'
DIR_RIGHTCAM_STACKZERO = 'stack_0_channel_0'
DIR_LEFTCAM_STACKONE = 'stack_1_channel_0'
DIR_RIGHTCAM_STACKONE = 'stack_1_channel_0'
```

2.2.1.3 Linear registration

To decrease the computational cost, images are normalized and cast on one byte before registration. While it generally does not degrade the registration quality, it may induce troubles when hyper-intensities areas are present in the image. In such a case, the useful information may then be summarized in only a few intensity values.

Intensity normalization in registration can be deactivated.

```
fusion_registration_normalization = False
```

Hyper-intensities areas may also the threshold calculation used for the automatic crop (step 3 of fusion). In such cases, the iterative registration method may find a local minimum that is not the desired one, because

the relative positions of the two images to be co-registered are too far apart. To circumvent such a behavior, a hierarchical registration can be done. It consists on a first pre-registration with a transformation with fewer degrees of freedom.

Intensity normalization in registration can be deactivated by

```
fusion_preregistration_compute_registration = True
# fusion_preregistration_transformation_type = 'translation'
fusion_preregistration_normalization = False

# fusion_registration_compute_registration = True
# fusion_registration_transformation_type = 'affine'
fusion_registration_normalization = False
```

It may be also preferable to deactivate the image normalization for both registration steps.

2.2.2 1.5-intraregistration-GC.py

2.2.3 1.5-intraregistration.py

The intra-registration procedure can be done either after the fusion step, or after the segmentation step. It aims at

- resampling the fusion and/or the segmentation images into a common frame/geometry, so they can better be compared, and
- building 2D+t images made of 2D sections from either the fusion and/or the segmentation images, so that the quality of the fusion and/of the tracking step can be visually assessed.

Its results will be stored in the `INTRAREG/INTRAREG_<EXP>` subdirectory where `<EXP>` is set by `EXP_INTRAREG` from the parameter file. `POST` and `SEG` directories are created only if respectively segmentation and post-segmentation sequences are required to be resampled/reconstructed in the common frame.

```
<EMBRYO>
├── ...
├── INTRAREG
│   └── INTRAREG_<EXP>
│       ├── CO-TRSFS
│       ├── FUSE
│       ├── LOGS
│       ├── MOVIES
│       ├── POST
│       ├── SEG
│       └── TRSFS_t<F>-<L>
└── ...
```

The intra-registration procedure is made of the following steps:

1. Co-registration of pairs of successive fused images (section 2.2.3.1). This yields the transformations $T_{t+1 \leftarrow t}$. Fused images are located in `<EMBRYO>/FUSE/FUSE_<EXP_FUSE>`: the parameter `EXP_FUSE` is either set in the parameter file or is set at `RELEASE`. This step may be long.
2. Composition of transformations issued from the co-registration step. This step computes the transformations $T_{ref \leftarrow t}$. towards a reference image `ref` given by the parameter `intra_registration_reference_index`.
3. Computation of the *template* image (section 2.2.3.2). This *template* image dimension are computed so that the useful information of all resampled images fits into it. Useful information can be issued from either the fused sequence, the segmentation sequence or the post-segmentation sequence. It is indicated

by the `intra_registration_template_type` which value can be either `'FUSION'`, `'SEGMENTATION'`, or `'POST-SEGMENTATION'`. This step may be long.

4. Resampling of either the fused or the segmentation images (section 2.2.3.3). Note that changing the parameters for this step will not require to re-compute the first steps.

5. Extraction of 2D+t images from the resampled sequences (section 2.2.3.4). Note that changing the parameters for this step (i.e. requiring extra movies) will not require to re-compute the first steps, with an eventual exception for the resampling step.

2.2.3.1 Co-registration parameters

Default registration parameters for the co-registration are set by:

```
# intra_registration_compute_registration = True
# intra_registration_transformation_type = 'rigid'
# intra_registration_transformation_estimation_type = 'wlts'
# intra_registration_lts_fraction = 0.55
# intra_registration_pyramid_highest_level = 6
# intra_registration_pyramid_lowest_level = 3
# intra_registration_normalization = True
```

Computed transformations are stored in `INTRAREG/INTRAREG_<EXP>/CO-TRSFS`. It may be advised to set the pyramid lowest level value to some higher value to speed up the co-registrations (recall that all pairs of successive images will be co-registered, i.e.

```
intra_registration_pyramid_lowest_level = 4
```

2.2.3.2 Template building parameters

```
# intra_registration_reference_index = None
# intra_registration_template_type = 'FUSION'
# intra_registration_template_threshold = None
# intra_registration_resolution = 0.6
# intra_registration_margin = None
```

The `intra_registration_reference_index` allows to choose the reference image (the one which remains still, i.e. is only displaced by a translation), by default it is the first image image of the series (associated to `begin`).

Depending on `intra_registration_template_type` (`'FUSION'`, `'SEGMENTATION'` or `'POST-SEGMENTATION'`, the two latter assume obviously that the segmentation has been done), the *template* image can be built either after the fusion or the segmentation images. If no threshold is given by `intra_registration_template_threshold`, the built template will be large enough to include all the transformed fields of view (in this case, the template is the same whatever `intra_registration_template_type` is).

If a threshold is given, the built template will be large enough to include all the transformed points above the threshold. E.g., the background is labeled with either '1' or '0' in segmentation images, then a threshold of '2' ensures that all the embryo cells will not be cut by the resampling stage. In this case, adding an additional margin to the template could be a good idea for visualization purpose. Last but not least, using a larger resolution than the `intra_registration_resolution` allows to decrease the resampled images volume (the default resolution i.e. voxel size, given by `target_resolution`, for the fusion step (see section 2.2.1.1), is set to 0.3, while it is set to 0.6 here).

Thus, building a *template* image after the segmentation images can be done with

```
# intra_registration_reference_index = None
intra_registration_template_type = "SEGMENTATION"
```

```
intra_registration_template_threshold = 2
# intra_registration_resolution = 0.6
intra_registration_margin = 10
```

Computed transformations from the *template* image as well as the *template* image itself are stored in `INTRAREG/INTRAREG<EXP>/TRSFS_t<F>-<L>` where `<F>` and `L` are the first and the last index of the series (specified by `begin` and `end` from the parameter file).

2.2.3.3 Resampling fusion/segmentation images

The resampled fusion and segmentation images will be stored respectively in `INTRAREG/INTRAREG_<EXP>/FUSE`, `INTRAREG/INTRAREG_<EXP>/SEG` and `INTRAREG/INTRAREG_<EXP>/POST`. Resampling is done either if the following parameters are set to `True` or if movies are requested to be computed.

```
# intra_registration_resample_fusion_images = True
# intra_registration_resample_segmentation_images = False
# intra_registration_resample_post_segmentation_images = False
```

2.2.3.4 2D+t movies

For either visual assessment or illustration purposes, 2D+t (i.e. 3D) images can be built from 2D sections extracted from the resampled temporal series. This is controlled by the following parameters:

```
# intra_registration_movie_fusion_images = True
# intra_registration_movie_segmentation_images = False

# intra_registration_xy_movie_fusion_images = [];
# intra_registration_xz_movie_fusion_images = [];
# intra_registration_yz_movie_fusion_images = [];

# intra_registration_xy_movie_segmentation_images = [];
# intra_registration_xz_movie_segmentation_images = [];
# intra_registration_yz_movie_segmentation_images = [];

# intra_registration_xy_movie_post_segmentation_images = [];
# intra_registration_xz_movie_post_segmentation_images = [];
# intra_registration_yz_movie_post_segmentation_images = [];
```

If `intra_registration_movie_fusion_images` is set to `True`, a movie is made with the XY-section located at the middle of each resampled fusion image (recall that, after resampling, all images have the same geometry). Additional XY-movies can be done by specifying the wanted Z values in `intra_registration_xy_movie_fusion`
E.g.

```
intra_registration_xy_movie_fusion_images = [100, 200];
```

will build two movies with XY-sections located respectively at Z values of 100 and 200. The same stands for the other orientation and for the resampled segmentation images.

2.2.4 2-mars.py

2.2.5 3-manualcorrection.py

2.2.6 4-astec.py

2.2.7 5-postcorrection.py

2.2.8 6-named.py

2.2.9 7-virtualembryo.py

Chapter 3

Tutorial

Before starting

It is advised to add to your `PATH` environment variable the path to the astec package. So, Astec commands can be launched without specifying the complete path to the command.

It can be done in a terminal (and will be valid only for this terminal)

```
$ export PATH=$PATH:/path/to/astec-package
```

or by modifying a setup file (e.g. `bashrc`, `.profile`, ...).

3.1 Starting with a toy embryo

Say we have an embryo named `160708-Tutorial` with four time points (these are the four first time points of the embryo `160708-Aquila-St8`). Thus at the beginning, we only have the raw data in the `<EMBRYO>=160708-Tutorial` file tree structure.

```
<EMBRYO>=160708-Tutorial
└── RAWDATA
    ├── LC
    │   ├── Stack0000
    │   │   ├── Cam_Left_00001.zip
    │   │   ├── Cam_Left_00002.zip
    │   │   ├── Cam_Left_00003.zip
    │   │   └── Cam_Left_00004.zip
    │   └── Stack0001
    │       ├── Cam_Left_00001.zip
    │       ├── Cam_Left_00002.zip
    │       ├── Cam_Left_00003.zip
    │       └── Cam_Left_00004.zip
    └── RC
        ├── Stack0000
        │   ├── Cam_Right_00001.zip
        │   ├── Cam_Right_00002.zip
        │   ├── Cam_Right_00003.zip
        │   └── Cam_Right_00004.zip
        └── Stack0001
            ├── Cam_Right_00001.zip
            └── Cam_Right_00002.zip
```

```
├── Cam_Right_00003.zip
└── Cam_Right_00004.zip
```

3.2 Fusing data

Let start by fusing data. I advised to create a `PARAMETERS` repository under the root `<EMBRYO>=160708-Tutorial` to store all the parameters file we will use. Then we copy there the `parameters.py` file and rename it as `parameters-fuse1.py`. I added `-fuse` to easily recognize it as a parameter file for fusing, and also add the numbering `1` to handle the case of several parameters files for fusing (if fusion parameters have to be changed for a few time points).

```
$ cd /path/to/160708-Tutorial
$ mkdir PARAMETERS
$ cp /path/to/astec-package/parameters.py PARAMETERS/parameters-fuse1.py
```

Then, this file `parameters-fuse1.py` is edited to be adapted to our experiment at hand. Here, the modification are as follows. Note that we only have the following changes with respect to the template version of the parameter file (see section 2.1).

```
...
5 PATH_EMBRYO='/Users/greg/COLLABORATIONS/DIGEM/TUTO-ASTEC/160708-Tutorial'
...
9 EN='160708-Tutorial'
...
16 EXP_FUSE='EXP1'
...
44 end=4
...
58 raw_ori = 'right'
...
60 raw_resolution = (.21, .21, 1.)
...
68 raw_mirrors = True
...
```

We have then the following file tree.

```
<EMBRYO>=160708-Tutorial
├── RAWDATA
│   └── ...
└── PARAMETERS
    └── parameters-fuse1.py
```

Now we can run the fusion

```
$ cd /path/to/160708-Tutorial
$ 1-fuse.py -p PARAMETERS/parameters-fuse1.py
```

After completion of the fusing stage, the tree structure is

```
<EMBRYO>=160708-Tutorial
├── RAWDATA
│   └── ...
└── PARAMETERS
    └── ...
```

```
└── FUSE
    ├── FUSE_EXP1
    │   ├── 1-fuse.log
    │   ├── 160708-Tutorial_fuse_t001.inr
    │   ├── 160708-Tutorial_fuse_t002.inr
    │   ├── 160708-Tutorial_fuse_t003.inr
    │   ├── 160708-Tutorial_fuse_t004.inr
    │   └── parameters-fuse1.py
```

The folder `FUSE_EXP1` not only contains the fused images, but also a copy of the parameter file (hence the importance that all parameter files have different names) as well as a `log` file (`1-fuse.log`) that keeps trace of all the Astec launched commands. However, it won't keep trace of the other commands (e.g. the user erases a file), thus it is still important to keep all this elsewhere.

3.3 Segmenting the first time point

The segmentation of the first time point consists in labeling each and every cell of the embryo. By convention, the background has the 1 label (0 is used when label images are resampled). Thus cell labels are always larger or equal than 2.

Although any method can be used to obtain this segmentation, the script `2-mars.py` provide a means to use the `MARS` method [**?**].

Again the parameter file `parameters.py` is copied and renamed to eventually edit it.

```
$ cd /path/to/160708-Tutorial
$ cp /path/to/astec-package/parameters.py PARAMETERS/parameters-seg-first1.py
```

Then, this file `parameters-seg-first1.py` is edited. The changes with respect to the template version of the parameter file (see section 2.1) are

```
...
 5 PATH_EMBRYO='/Users/greg/COLLABORATIONS/DIGEM/TUTO-ASTEC/160708-Tutorial'
...
 9 EN='160708-Tutorial'
...
16 EXP_FUSE='EXP1'
...
26 EXP_MARS='EXPMARS1'
...
```

Note that the line #16 indicates where to find the fused images. The segmentation is done through

```
$ cd /path/to/160708-Tutorial
$ 2-mars.py -p PARAMETERS/parameters-seg-first1.py
```

After completion of the segmentation stage, the tree structure is

```
<EMBRYO>=160708-Tutorial
├── RAWDATA
│   └── ...
├── PARAMETERS
│   └── ...
└── FUSE
    └── FUSE_EXP1
        └── ...
```

```
└── SEG
    ├── SEG_EXPMARS1
    │   ├── 160708-Tutorial_mars_t001.inr
    │   ├── 2-mars.log
    │   └── parameters-seg-first1.py
```

Note the name of the result image with the `_mars` suffix. Such a naming convention is mandatory and has to be followed in case of segmentation by other means.

3.4 Correcting the first time point segmentation

As the sequence segmentation proceeds by propagation, the first segmentation must be cured. Under-segmentations may be corrected either by changing segmentation parameters and to relaunch the `2-mars.py` script or by manual edition (then be careful not to use an already used label). Over-segmentations can be automatically corrected (after identification). They have to be listed into a file whom lines contains two numbers, the labels to be merged (more precisely, only the second label will be kept) . E.g.

```
43 54
38 20
```

means that cells of labels 43 and 54 will be merged together (forming a cell of label 54), as well as cells of labels 38 and 20 (forming a cell of label 20). The file `/path/to/astec-package/mapping.txt` is a template for this merging.

Again, copy the parameter template file, as well as the merging template file (cells to be merged have to identified beforehand).

```
$ cd /path/to/160708-Tutorial
$ cp /path/to/astec-package/parameters.py PARAMETERS/parameters-seg-correction1.py
$ cp /path/to/astec-package/mapping.txt PARAMETERS/mancor_mapping_file.txt
```

and edit the two files. The second file contains only the two lines indicated above. The changes with respect to the template version of the parameter file are

```
  ...
  5 PATH_EMBRYO='/Users/greg/COLLABORATIONS/DIGEM/TUTO-ASTEC/160708-Tutorial'
  ...
  9 EN='160708-Tutorial'
  ...
 16 EXP_FUSE='EXP1'
  ...
 26 EXP_MARS='EXPMARS1'
  ...
 28 EXP_SEG='EXPSEG1'
  ...
186 mancor_mapping_file='PARAMETERS/mancor_mapping_file.txt'
  ...
```

Note that the line #186 gives the (here relative) name of the file containing the cells to be merged. The `MARS` result is supposed to be found in the `SEG/EXPMARS1` folder (given by the variable `EXP_MARS`) while the correction result will be placed into the `SEG/EXPSEG1` folder (given by the variable `EXP_SEG`)

```
$ cd /path/to/160708-Tutorial
$ 3-manualcorrection.py -p PARAMETERS/parameters-seg-correction1.py
```

After completion of this stage, we have:

```
<EMBRYO>=160708-Tutorial
├── RAWDATA
│   └── ...
├── PARAMETERS
│   └── ...
├── FUSE
│   └── FUSE_EXP1
│       └── ...
└── SEG
    ├── SEG_EXPMARS1
    │   └── ...
    └── SEG_EXPSEG1
        ├── 160708-Tutorial_seg_t001.inr
        ├── 3-manualcorrection.log
        ├── mancor_mapping_file.txt
        └── parameters-seg-correction1.py
```

Note the name of the result image with the _seg suffix.

3.5 Segmentation propagation

```
$ cd /path/to/160708-Tutorial
$ cp /path/to/astec-package/parameters.py PARAMETERS/parameters-astec.py
```

```
...
5 PATH_EMBRYO='/Users/greg/COLLABORATIONS/DIGEM/TUTO-ASTEC/160708-Tutorial'
...
9 EN='160708-Tutorial'
...
16 EXP_FUSE='EXP1'
...
28 EXP_SEG='EXPSEG1'
...
44 end=4
...
```

```
$ cd /path/to/160708-Tutorial
$ 4-astec.py -p PARAMETERS/parameters-astec.py
```