Baby Brain Toolkit

Fbrain ERC project: Computational Anatomy of Fetal Brain

September 13, 2011

Contents

1 Introduction

BTK stands for Baby Brain Toolkit. This toolkit is developed in the context of the Fbrain ERC project: "Computational Anatomy of Fetal Brain"[1]. Studies about brain maturation aim at providing a better understanding of brain development and links between brain changes and cognitive development. Such studies are of great interest for diagnosis help and clinical course of development and treatment of illnesses. Several teams have begun to make 3D maps of developing brain structures from children to young adults. However, working out the development of fetal and neonatal brain remains an open issue. This project aims at jumping over several theoretical and practical barriers and at going beyond the formal description of the brain maturation thanks to the development of a realistic numerical model of brain aging.

1.1 Copyright

[1] http://lsiit-miv.u-strasbg.fr/miv/index.php?contenu=erc

In this respect, the user's attention is drawn to the risks associated with loading, using, modifying and/or developing or reproducing the software by the user in light of its specific status of free software, that may mean that it is complicated to manipulate, and that also therefore means that it is reserved for developers and experienced professionals having in-depth computer knowledge. Users are therefore encouraged to load and test the software's suitability as regards their requirements in conditions enabling the security of their systems and/or data to be ensured and, more generally, to use and operate it in the same conditions as regards security.

1.2 Installation

1.2.1 Dependencies

Baby Brain Toolkit (BTK) needs the following packages[2]:

- CMake (www.cmake.org), Tclap (tclap.sourceforge.net), OpenMP (openmp.org), VTK (www.vtk.org), ANN (www.cs.umd.edu/˜mount/ANN). These libraries can be installed using the following command line:
  - for debian-based distributions: `apt-get install cmake cmake-curses-gui libtclap-dev libgomp1 libvtk5-dev libann-dev`
  - for MacOSX using macports: `port install cmake tclap vtk5 libANN`
- Install Git: this library can be installed using the following command line :
  - for debian-based distribution: `apt-get install git-core`
  - for MacOSX using macports: `port install git-core`
- Insight Toolkit (ITK) version 4[3]:

  ```
  git clone git://itk.org/ITK.git
  git checkout v4.0a08
  mkdir ITK-build
  cd ITK-build
  ccmake ../ITK/
  ```

  This will bring up the CMake configuration screen. Press `[c]` for configure and then use `[t]` to toggle the advanced mode. Make the following changes:

  ```
  BUILD_TESTING = OFF
  CMAKE_BUILD_TYPE = Release
  ITK_USE_REVIEW = ON
  ITK_BUILD_ALL_MODULES = ON
  ```

  Then press `[c]` to configure and `[g]` to generate the make file. Finally, type `make` at the prompt to obtain the final build of ITK.

1.2.2 Download and compile the BTK sources

- Get the BTK sources: `git clone https://github.com/rousseau/fbrain.git`
- Then:

  ```
  mkdir fbrain-build
  cd fbrain-build
  ccmake ../fbrain
  make
  ```

[2] BTK has been tested under Debian 5.0 and MacOSX 10.6.8

[3] As there is still not a stable version for ITK4, there are potential compilation problems when using the HEAD version. We have tested BTK with the version of ITK corresponding to the tag v4.0a08.

Most of the programs of the BTK suite use the OpenMP library for multi-threading purpose. The number of cores used can be tuned using the following command line (in this example, 4 cores will be used): `export OMP_NUM_THREADS=4`

2 The BTK pipeline

BTK allows to implement a pipeline for the processing of fetal images, i.e. the reconstruction of anatomical and diffusion data, and the final tractography, all expressed in the same local coordinate system. This processing can be summarized in the following steps: 1) image conversion, 2) anatomical image reconstruction, 3) recontruction of the diffusion sequence, 4) registration of diffusion to anatomical data and 5) tractography.

2.1 Image conversion

BTK supports and has been tested by using images in Nifti format (http://nifti.nimh.nih.gov/nifti-1). However, images are frequently available in DICOM format and an image conversion is required. This can be performed by using dcm2nii (http://www.cabiatl.com/mricro/mricron/dcm2nii.html), Slicer or other softwares.

Let say that you have 3 (possibly orthogonal) anatomical images:

```
ana01.nii.gz
ana02.nii.gz
ana03.nii.gz
```

and one set of DWI images:

```
dwi.nii.gz
dwi.bvec
dwi.bval
```

Please check that images are not flipped.

NOTE: The set of three files .nii.gz, .bvec, and .bval used to describe a DW sequences are represented in BTK just by the basename ("dwi" in the previous example), and the sequences must be provided in this way to the different applications. This allows to have shorter command lines and a the use of consistent filenames.

2.2 Anatomical image reconstruction

Anatomical image reconstruction can be performed by using **btkImageReconstruction** (Section 3.2)[4], followed by a re-orientation procedure using **btkSetStandardCoorSystem** and **btkReorientImageToStandard** (Section 4)[5]:

```
btkImageReconstruction -i ana01.nii.gz -i ana02.nii.gz -i ana03.nii.gz
                       -m ana01_mask.nii.gz -m ana02_mask.nii.gz -m ana03_mask.nii.gz
                       -o ana3D.nii.gz --mask

btkSetStandardCoorSystem -i ana3D.nii.gz -o ana3D_standard.nii.gz -d 3
btkReorientImageToStandard -i ana3D_standard.nii.gz -o ana3D_oriented.nii.gz -l landmarks.fcsv
```

[4]Best results are obtained by using a mask for each anatomical image. Such image masks can be easily created using ITKSnap for instance.

[5]Note that the landmarks file is obtained using external software (Slicer). Please see Section 4 for details about this step.

2.3 Reconstruction of the diffusion sequence

To reconstruct diffusion data, you want to follow the steps described in Section 3.3. The use of two applications is required here: **btkGroupwiseS2SDistortionCorrection** and **btkRBFInterpolationS2S**.

2.4 Registration of diffusion to anatomical data

This can be performed by using **btkRegisterDiffusionToAnatomicalData** (Section 3.2)

2.5 Tractography

If you have followed the previous steps correctly, at this point you should have the reconstructed anatomical and diffusion data spatially aligned, and ready to perform the tractography. To do this, BTK provides **btkTractography** (Section 3.4)

3 Applications

3.1 Denoising

btkNLMDenoising This program applies a non-local mean filter to a 3D image for denoising purpose. Usage: `-i input_image_filename -o output_image_filename`. The best results are usually obtained by using a mask (or a padding value).

btkNLMDenoising4DImage This program applies a non-local mean filter to each 3D image of a 4D image, for denoising purpose. Usage: `-i input_image_filename -o output_image_filename`. The best results are usually obtained by using a mask (or a padding value).

3.2 Anatomical reconstruction

btkImageReconstruction This program allows to obtain a high-resolution image from a set of low-resolution images, typically axial, coronal, and sagital acquisitions [2].

Minimal usage: `btkImageReconstruction -i image1 ··· -i imageN -o output --box`.

Recommended usage: `btkImageReconstruction -i image1 ··· -i imageN -m mask1 ··· -m maskN -o output --mask`. The use of a mask provide better results since it allows an accuratelly estimation of the initial transform, and constrains the registration to the region of interest.

The full list of optional parameters of the method can be obtained by `btkImageReconstruction --help`

3.3 Reconstruction of DW sequences

The reconstruction of diffusion-weighted (DW) sequences aims at obtaining a sequence corrected for fetal moving and eddy-current distortions. This can be performed in BTK by using the two following applications.

btkGroupwiseS2SDistortionCorrection It performs a groupwise slice-by-slice registration of the image components of the sequence.

Minimal usage: `btkGroupwiseS2SDistortionCorrection -i input -o output -t transform-folder`.

-i input sequence.

-o output sequence.

-t folder to save the transformation files

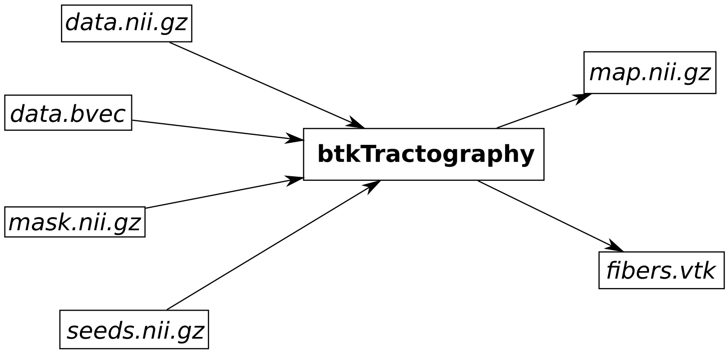


Figure 1: Standard pipeline of the btkTractography program.

The slice-by-slice transform for a given DW image is saved as a set of N transforms in ITK format, with N the number of slices in the image.

NOTE: For slice registration, 20% of the samples are used for computing the image metric. As the region of interest of the slice can be small, this number of samples might be insufficient to compute the metric accurately. However, this percent of samples has been sufficient for the tested sequences.

btkRBFInterpolationS2S It performs an interpolation of scattered data generated from the application of slice-by-slice transforms. To this end, radial basis functions are used. By default, the output has isotropic voxels of size equal to the in-plane resolution of the input.

Minimal usage: `btkRBFInterpolationS2S -i input -m mask.nii -o output -t transformation-folder`.

- -i input sequence
- -m mask for the B0 image
- -o output sequence
- -t folder with the transformations generated by `btkGroupwiseS2SDistortionCorrection`

3.4 Tractography

Standard usage

Suppose you want to perform a tractography on a diffusion weighted MRI dataset. You should have a dwi image, the corresponding gradient vectors' coordinates, a mask of the brain white matter and a label image of the seeds. Assume this data is stored in files named repsectively for instance `data.nii.gz`, `data.bvec`, `mask.nii.gz` and `seeds.nii.gz`. The tractography is accomplished by the command below.

```
btkTractography -d data.nii.gz -m mask.nii.gz -l seeds.nii.gz
```

When the program terminates its task, the probability connection map and the fibers estimation are saved in files respectively named `map.nii.gz` and `fibers.vtk`. The connection map is a volume image of probability intensities (`i.e.` intensities between 0 and 1) with the same origin, orientation and spacing as the diffusion weighted image. The fibers are polygonal data of VTK library in world coordinates. The standard pipeline of the program is shown in Fig. 1.

The file named `seeds.nii.gz` is a label map which is used to generate seeds for the algorithm. The spacing in millimeters between seeds can be adjust with the option `--seed_spacing`.

By default, Right-Anterior-Superior (RAS) coordinate system is used to ensure compatibility with Slicer3D. However, it can be forced to use Left-Posterior-Superior (LPS) coordinate system by using option `--lps` on command line.

If the DWI sequence contains more than one baseline image, the program takes the average of all baseline images. The baseline images can be everywhere in the sequence. It should be recognized by null vector in corresponding gradient's table.

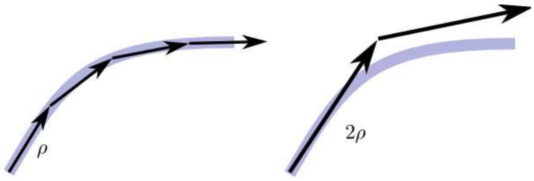


Figure 2: Effect of the step size option on a particle's trajectory. With a large step size (right), the particle may overshoot the trajectory of the ground truth.

Advanced usage

In addition to standard arguments of `btkTractography` program, there are some other parameters that let you to alter algorithm's behaviour. These options can be classified into three groups : model's options, constraints on trajectory and filter's options. The first group options allow you to tweak the model (for more details about it, please refer to [1]). The second group options let you to control the particle's trajectory. These options provide prior informations to the algorithm. The last group options are dedicated to the particle filter control.

Since the default parameters values may work in the most of cases, they are optional. A list is of optional features is avaible by using the command

`btkTractography --help` or `btkTractography -h`

and program's arguments are much more described below.

Model's order

The model's order (i.e. the spherical harmonics' order) can be specified by the option

`--model_order <order>` ,

for `order` $\in \{2, 4, 6, 8\}$. The default value is 4. For more details, please refer to [1].

Model's regularization

A Laplace-Beltrami regularization coefficient is used to assume a better estimation of the model. This coefficient can by manually modified by the option

`--model_regularization <coefficient>` ,

for `coefficient` $\in \mathbb{R}$. The default value is set as 0.006. For more details, please refer to [1].

Displacement step size

The displacement step size of a moving particle can ben adjusted as you want by using the option

`--step_size <length>` ,

where `length` $\in \mathbb{R}_+^*$. Note that this option is expressed in `mm`. The default value is fixed at 0.5 `mm`. By setting a big step size, the particles will move quickly. So the biger is the step, the faster the algorithm will finish, but as shown by Fig. 2, some informations may be missed and the particle's trajectories may overshoot the ground truth, resulting in a bad estimation.

Angular threshold

An angular threshold prevent a particle to return back. This option has to be expressed in radian and can bet set by

`--angular_threshold <angle>` ,

where `angle` $\in]0, 2\pi[$. The default value is set as a $\frac{\pi}{3}$ angle. As illustrated in two dimensions in Fig. 3, an angle threshold is used to define an allowed area for successive sampled directions. This can be seen as a global curvature parameters on trajectories. A small angle defines trajectories with a small curvature. This is a prior information on ground truth trajectory.

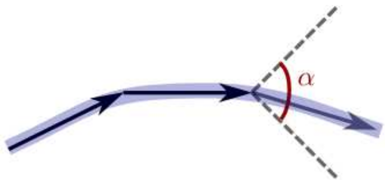


Figure 3: An angle threshold allows the algorithm to sample successive direction only in the cone defined by this angle. This illustration show the principle in two dimensions.

Figure 4: Local effect of rigidity parameter on a particle's trajectory. This parameter helps to "attract" the current displacement vector in the direction of the previous displacement vector of a particle. It correspond to a concentration paramter of a von Mise-Fisher density probability used in the prior density of the system. For instance, a rigidity of 0 leads to an equiprobable distribution, whereas a rigidty tending to infinity leads to a distribution focused on a point.

Rigidity

The rigidity option controls how much you want the particles to have straight trajectory. You can adjust it by

```
--curve_constraint <rigidity> ,
```

where `rigidity` $\in \mathbb{R}_+^*$. The default value is fixed at 30. This value correspond to a concentration parameter of a von Mises-Fisher density probability used in the prior density of the system. As Fig. 4 illustrates locally in two dimensions, a high value leads to a straight trajectory.

Number of particles

The number of particles in the system is set by the option

```
--number_of_particles <number> ,
```

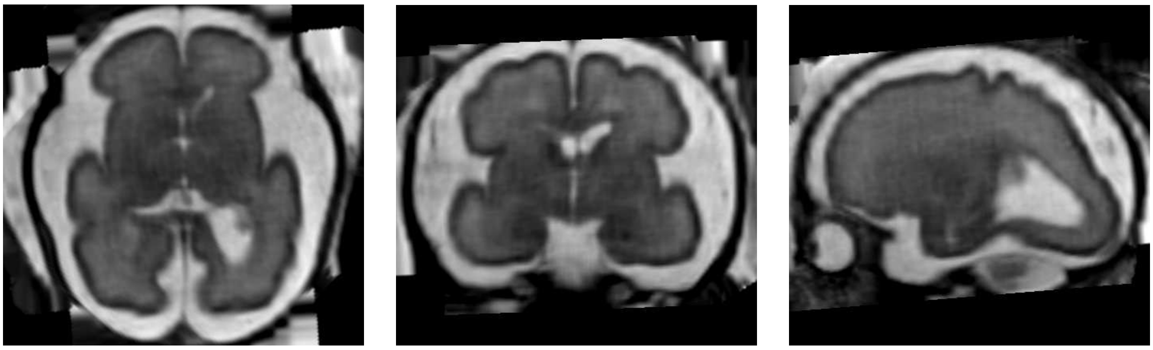
where `number` $\in \mathbb{N}^*$. By default, the algorithm will use 1000 particles. A poor number of particles leads to a short computation time and a poor estimation. A large number of particles leads to a long computation time and a good estimation. In general, the default number of particles is a good compromise between computation time and estimation.

Resampling threshold

This option modify the resampling threshold of the system. When the number of effective particles in the system falls below this resampling threshold, the particles are resampled according a multinomial resampling. It can be adjust by

```
--resampling_threshold <percent> ,
```

where `percent` $\in [0, 1]$ is the percent of minimal effective particles in the system. A low threshold value will result in an inefficient algorithm because the particles with low weight are not are not often eliminated. Conversely, a high threshold value leads to a bad estimation because the search space will not be explored enough.



(a) (b) (c)

Figure 5: Example of an anatomical reconstruction of a fetal brain by using `btkImageReconstruction`. (a) axial, (b) coronal, and (c) sagital view.

4 Utilities

btkModifyImageUsingLookUpTable This program modifies one image using a look up table defined in a ascii file (2 columns, one for the original values, one for the final values). Usage: `-i input_image_filename -t input_table_filename -o output_image_filename`

btkExtractOneImageFromSequence This program extracts one image from a 4D sequence. Usage: `-i input_image_filename -o output_image_filename --image_index index`

btkNrrdToNifti This program convert an image from Nrrd file (*.nhdr and *.nrrd) to a Nifti file (*.nii or *.nii.gz). The conversion of a DWI image is possible by using the option `--dwi`. Usage: `-i input.nhdr -o output.nii.gz`. Usage for DWI sequence: `--dwi -i input.nhdr -o output.nii.gz`.

btkNiftiToNrrd This program convert a diffusion sequence in nifti format[6] to the nrrd format (*.nhdr).

Usage: `-i input -o output.nhdr`

The list of optional parameters can be obtained by `btkNiftiToNrrd --help`

btkReorientImageToStandard Sometimes it is useful to reorient the image to the standard orientation. This is necessary with fetal images since in general the fetus is in a random orientation with respect to the scanner.

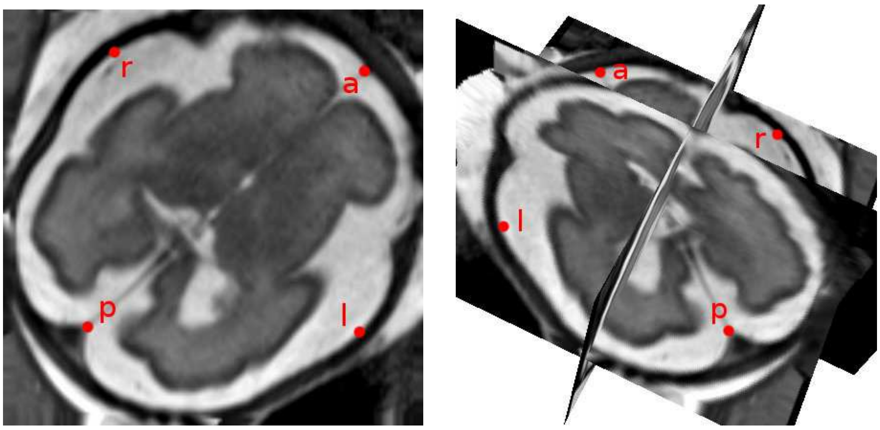
Usage: `btkReorientImageToStandard -i image -o output -l landmarks`. `landmarks` is a text file containing points that define the left-right and the posterior-anterior directions. The points l and r define the left $\rightarrow$ right direction, and the points p and a define the posterior $\rightarrow$ anterior direction. Such file can be easily generated by using Slicer[7] as follows:

1. Open the high-resolution image by using the *Volume* module.
2. Toogle on the visibility of all slices in the 3D view. This allows to identify the left and right sides of the brain in the 2D views.
3. Place the landmarks l, r, p, and a in this order by using `[p]`.
4. Save the file (*.fcsv) by using the menu File $\rightarrow$ Save.

btkReorientDiffusionSequenceToStandard Reorients a DW sequence to the standard orientation. This is necessary with fetal images since the fetus is in a random orientation with respect to the scanner. This is particularly important in DWI because colormaps lack of significance, which makes difficult the identification of specific bundles

[6] Currently there is no nifti standard for DWI, so DW images are saved as a standard nifti sequence (*.nii, *.nii.gz) and two text files containing the b-values (.bval) and the gradient directions (.bvec).

[7] http://www.slicer.org



(a) (b)

Figure 6: Placement of landmarks by using Slicer. (a) axial slice, (b) 3D view.

Usage: `btkReorientDiffusionSequenceToStandard -i image -o output -l landmarks`.

`landmarks` is a landmarks file obtained as explained above.

btkCropImageUsingMask This program crops one (3D or 4D) image using a 3D mask. Usage: `-i input_image_filename -m input_mask_filename -o output_image_filename -d 3`, where '-d' is the dimension of the input image (by default 3).

btkRegisterDiffusionToAnatomicalData This program registers a DW sequence to an anatomical image.

Recommended usage: `btkReorientDiffusionSequenceToStandard -i input -o output -r reference.nii --mask mask.nii`.

- -i input sequence
- -o resampled sequence (by default, linear interpolation is used)
- -r reference image (anatomical image)
- -m image mask for the B0 image

The list of optional parameters can be obtained by `btkNiftiToNrrd --help`

btkImageInjection This program performs the injection of a set of images with an already existing set of transformations. This avoids the need to perform a new image reconstruction (computationally expensive) after modification of the input images (some filtering for example) or an involuntary deletion of the reconstructed image.

Recommended usage: `btkImageInjection -i image1 ··· -i imageN -m mask1 ··· -m maskN -i transform1 ··· -i transformN -o output --mask`.

- -i input image
- -m image mask
- -t image transform
- -o reconstructed image

The list of optional parameters can be obtained by `btkImageInjection --help`

Acknowledgment

The research leading to these results has received funding from the European Research Council under the European Communitys Seventh Framework Programme (FP7/2007-2013 Grant Agreement no. 207667).

References

[1] M. Descoteaux, E. Angelino, S. Fitzgibbons, and R. Deriche. Regularized, fast, and robust analytical q-ball imaging. *Magnetic Resonance in Medicine*, 58(3):497–510, 2007.

[2] Francois Rousseau, Orit A Glenn, Bistra Iordanova, Claudia Rodriguez-Carranza, Daniel B Vigneron, James A Barkovich, and Colin Studholme. Registration-based approach for reconstruction of high-resolution in utero fetal MR brain images. *Acad Radiol*, 13(9):1072–1081, Sep 2006.