CS 8395 - Assignment

Sanidhya Mangal

Vanderbilt University, sanidhya.v.mangal@vanderbilt.edu

1 Introduction

The aim of this project is to perform white matter tractography as described in [1]. The goal is to simply perform the tractography on Diffusion Tensor MRI image. The tractography is performed on sample DTIBrainMRI present in the Slicer3D. All the implementation is done using ITK package and C++. To test the implementation Vanderbilt's Accre Cluster was used. Moreover, some of the standard filters and libraries present is ITK is leveraged to optimize the code and prevent writing redundant code.

Unlike all the process, we are required to have proper data for performing tactrography. Hence, as mentioned above we are using DTIBrainMRI image from Slicer3D. Therefore, step 0 for this project was to download the image from Slicer and store it as nrrd file format. Once the data was downloaded the project was stubbed from the baseline HelloWorld project and configuration of CMake was carried to aid in build process. The entire script is present in the (Tactography.cxx) file. The project could be broken into two components one is main function which handles most of the logic and other subroutines for recursive tracking of pixels, writing image file and updating index for forward and backward updates. Now that we have some basic overview about the project setup and architecture, lets deep dive into the further process. Using DiffusionTensor3D class we read the diffusion MRI inputfile for performing tactography, after reading the file we compute the largest possible region and create a dummy region named as newRegion for performing iteration and using it for other images. After region creation we will create a 2D tracker image for tracking the voxels visit and tracing the white matter. The tracker image is created by first defining the image and then calling the CreateTrackerImage subroutine which takes trackerimage pointer, base image pointer and region. After creation of tracker image we will create a PrincipalAxisImage (PAImage) which will store all the eigen value vectors for the MRI Image. The allocation of PAImage also requires region, size and origin unlike tracker image but there is a slight difference between two i.e., PAImage is a 3D vector image.

Next, to compute the eigen value and eigen vector and store it in PAImage we need to iterate through each region of the image. In order to achieve this we will require a region iterator. Therefore, we define the region iterator over the input MRI image and PAImage. The computation could be done using ComputeEigenAnalysis() function of itk, it takes a ArrayType and TensorType as an input. After this I computed the TensorFractionalAnisotropy using itkTensorFractionalAnisotropyImageFilter filter class. Once both the major component

was computed for the images we performed the diffusion tractography. To perform this we need to set a seed value which is a IndexType and it was decided after analyzing the image and FA image. In current implementation it was hard-coded in the main function as seed but it could be easily replaced by command line arguments. Once, other hyperparms such as delta is defined we can call the traverseImage recursive function which takes FaImage, PAImage, tracker-Image, current location or the seed, delta and iteration number. For the first time we call this function with zero iteration and seed as current location. In the traverseImage we first check the stopping conditions which are as follows:

- If the current pixel is out of the bound.
- If the current pixel is already visited in the tracker image.
- If the number of iterations exceeds 1000 iterations.
- If the eigen value of pixel is less than threshold FA i.e., 0.2.

If the current function calls fails all the stopping condition then we first set pixel value of the tracker as 1.0 and increment the iterator. After this, the eigen value is extracted from the PA Image for this location and based on that the new set of forward and backward pixels are computed using following formulae $x + dx * delta$ and $x - dx * delta$ respectively. These new location is computed by calling a computeIndex helper method. After computing these new methods we perform the recursive call on the traverseImage function using this new forward and backward pixels. If any of the stopping condition is met we return zero.

Once, we have tested the single voxel implementation of the image of our predefined seeds. The tractography is preformed on the segmented image of corpus callosum. To perform this step first FA image was downloaded and in the ITK-snap manually the segmentation mask was drawn using paint brush tool. After that, we loaded this segmented image, allocated the segmentation tracker image to trace the white matter along this segmentation. Once this is done we iterate over the segmentation iterator which is defined segmented image and newRegion. In each iteration we call the traverseImage subroutine on current index and 0 iteration only if the value of segmentation pixel is 1.0.

After the computation of both the tracker image we output following images using imageWriter helper method:

- EigenValue image file
- FA image file
- Seed tracker file
- Segmented tracker file

Experiment

To carry out the experiments first we decoded the seeds by visualizing the FA image and based on that seed of [73,83,34] was decided.

Once the seed was decided next step was to compute the result and based on that the experiment was ran on the 3 different settings of delta and 3 different settings of FA which are (0.75, 0.8, 0.9) and (0.4, 0.3, 0.25) respectively. The result obtained based on this experiment run can be observed in the fig. 1.

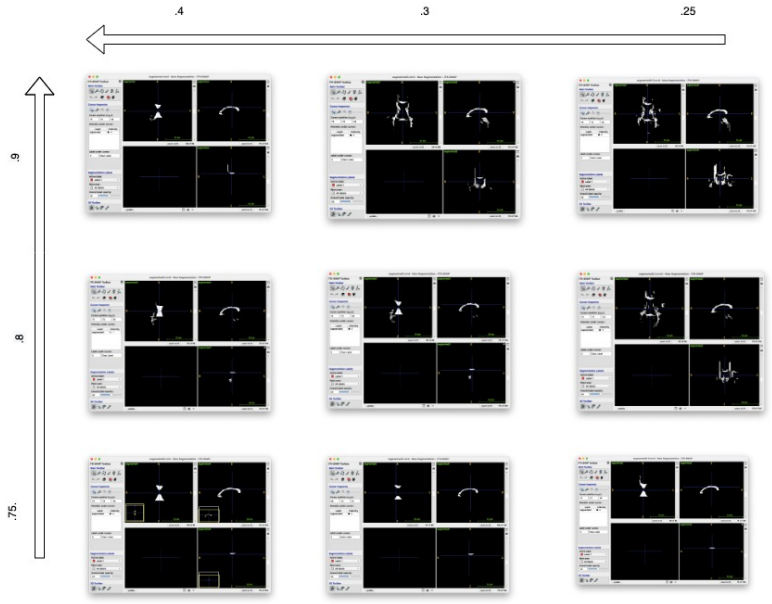

Fig. 1. A figure consisting ITK snap visualization of segmented runs of the algorithm for set of delta and FA

Code Usage

The compiled binary could be run using following command,
./Tactography InputDiffusionMRI.nrrd InputCorpusCollosumSegmentedFile.nrrd OutputPrincipalAxisEigenValueImage.nrrd OutputFAImage.nrrd OutputSeedImage.nrrd OutputSegmentedTrackFileImage.nrrd

Acknowledgement

The most optimized version of the project would not be possible without the help or reference of [2] [3] (referred for optimization and correct setup of the recursive function) and [4].

References

1. Mori, S., Van Zijl, P. C. (2002). Fiber tracking: principles and strategies–a technical review. NMR in Biomedicine: An International Journal Devoted to the Development and Application of Magnetic Resonance In Vivo, 15(7-8), 468-480.
2. https://github.com/xj-sun/White-Matter-Tractography
3. https://github.com/VU-CS8395-Fall2021
4. https://itk.org/Doxygen/html/index.html