CS 8395 - Assignment VTK Part

Sanidhya Mangal

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

1 Introduction

The aim of this project is visualize the white matter tractography as described in [1]. In this module we tried to extend the previous assignment by extending it to visualize the tracts using VTK rather than generating the tractography images. All the implementation is done using ITK and VTK 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, whereas VTK libraries are used to visualize the tracts and FA image.

All the computations for computing the tractography images are kept same as previous part, hence, the new files and code is reorganized to keep for using it in callbacks. However, there is one change which was incorporated in the ITK part which is to create a list of all the locations which are visited while performing segmentation track. These points are later used for visualization operation.

The data for VTK visualization is ingested from ITK only hence to convert the ITK data into VTK we require a filter *itkImageToVtkImage*, this filter is used to convert the ITKImage into the VTK image. Once, the image was prepared for the VTK rendering then an image mapper was created to map the image for this *vtkImageSliceMapper* class is used, in addition to this, a default slice at 55 is set for the image.

Now in order to visualize the tracts of the images we need to create a poly data into the render. To perform this we need to grow all the tracts with the time in stead of simply rendering it. This could be done using a TimerCallback where it calls the function to traverse the image for us but it could be more time consuming and might take a large amount of time to compute and then render it. Hence, instead of performing the tactography and finding next set of voxels we first perform the tactography using the image and then store all the values in a form of vector points which is of type *list<ImageType::IndexType>*. And then use these points to grow over the period of time using TimerCallback. For visualizing the points I tried to create an image using the *vtkPolyData* to map all the pixels into a point mesh and then render it using VTK. Hence, to execute this a global list of all the points is maintained to keep a track of all the points visited, additionally another list is maintained which contains list of only initial pixels, which are then modified using the timer callback. And using this callbacks the new points are rendered again.

Now as per the instructions in part 4 we need to select some new seed voxels and then based on that we need to select the pixels or the seed using Mouse-

Callbacks and then use it to create tracks. Hence, in order to perform this we first need to pick the points, store them in a stack and then use it as an initial points and follow the procedure same as above used in timer callback. The only difference in this is that in this user will specify the seed points and later it would be used for computing the ops.

Besides, this one more component is used i.e., *vtkProgrammableFilter*, it is used in timer callback to update the list of initalpoints its work is to use update the i_{th} point in the initialpoints list with the i_{th} from the global points which helps in changing the point cloud which could be used later to regrow the points using the timer callback.

Fig. 1, 2 and 3 renders the output of the runs

Fig. 1. A figure consisting of FA image along with the point estimates without mouse callbacks

Code Usage

The entire vtk portion was divided into two different files, one was to run the files using without mouse callbacks and another one was using the mouse callback. Which are as follows:

Without Mouse Click

./TactographyVTK InputDiffusionMRIInputCorpusCollosumSegmentedFile

Fig. 2. VTK window with FA image and mouse callbacks, spheres denotes the interactors of the mouse.

With Mouse Click

./Tactography_Click InputDiffusionMRIInputCorpusCollosumSegmentedFile

Sidebar

I tried my best to render the polydata using VTK for the tractography images but failed to do so, upon debuging I was able to find that the all the points are rendered and despite of setting them in a polydata it failed to produce viable results. Hence, the second image in my example runs are empty.

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), [5] 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.

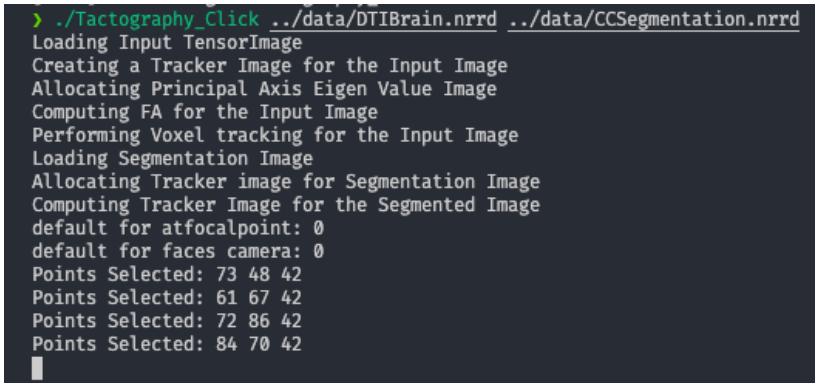

Fig. 3. Console output of point selected using mousecallbacks

2. https://github.com/xj-sun/White-Matter-Tractography
3. https://github.com/VU-CS8395-Fall2021
4. https://itk.org/Doxygen/html/index.html
5. https://vtk.org/doc/nightly/html/