6000

How to avoid biased streamlines-based metrics for streamlines with variable step sizes

Jean-Christophe Houde[1], Marc-Alexandre Côté-Harnois[1], and Maxime Descoteaux[1]
[1]*Computer Science department, Université de Sherbrooke, Sherbrooke, Quebec, Canada*

Target audience – Researchers developing tractography analysis tools, especially tools that compute statistics or connectivity analyses on streamlines, as well as end users of those tools, who should be aware of some potential pitfalls when working with compressed streamlines and streamlines having a large and variable step size.

Purpose – This abstract aims to show that current methods used to analyze and derive metrics from streamlines, also known as *Tractometry* methods, need to be updated to work correctly with streamlines with non-equidistant points along them. This update procedure is shown to be straightforward, and the effect of not taking compression / sampling into account is shown to be important.

Problem statement – Basic tractometry techniques compute metrics and statistics along a streamline by finding the voxels containing each point of the streamline and using the value of those voxels in further calculations (for example, to compute the mean FA value along a streamline). For streamlines generated using a small and regular step size, such as those generated by most of the current tractography algorithms, this voxel selection procedure is reasonable because the sampling of voxels should be regular and correctly cover all the streamline. However, if the step size is too large, or if it varies along the streamline, the points-based voxel selection technique can be biased, because some voxels may simply be missed. Some tractography algorithms are beginning to generate streamlines with variable step sizes, because they may decide to locally vary the step size according to some prior information (fODF uncertainty, etc). This kind of streamlines may also be generated when using the lossy streamline compression method presented in [1]. The main step of this algorithm consists in linearizing the streamlines by removing points that are almost collinear, up to a specified error threshold. Using a small error threshold of 0.1mm results in datasets that can be from 80% to 95% smaller than the original files, while being visually indistinguishable from uncompressed streamlines. This allows users to transfer and visualize those files more easily. However, this also implies that the distance between 2 consecutives points may vary along the streamline, introducing the same kind of bias as with a large step size.

Methods – To overcome the voxel selection bias caused by a variable or large step size, we replaced the normal points-based technique by a Bresenham-style line integration technique [2, 3]. Instead of simply finding all voxels containing at least one point of the streamline and using those voxels for further computations, our technique computes the voxels intersected by each segment of the streamlines. All those voxels are then added to the tractometry computations, and the bias is therefore eliminated. To validate this method, we generated 500k streamlines covering the whole brain using Mrtrix [4]. We then extracted 27 of the main WM bundles using the TractQuerier [5] with a custom query, and compressed those bundles using the Fibercompression tool (https://github.com/scilus/FiberCompression) with 7 maximal error values, ranging from 0.001mm to 1mm. For each error threshold ε, as well as for the uncompressed original bundles, we then calculated the mean values of various diffusion metrics (AD, FA, MD, RD) along the 27 bundles, as well as the number of voxels occupied by those bundles. The mean values and voxel counts were computed using the basic points-based technique available in Dipy ([6], dipy.org), as well as with our improved segments-based technique [3]. Then, for each bundle B_i, we computed the difference between the "uncompressed" mean and the mean computed on bundle B_i for each compression level ε. This difference was expressed as a percentage of the mean value of the uncompressed bundle, and will be called $Diff(B_{i,\varepsilon})$.

Results – Table 1 shows the average of $Diff(B_{i,\varepsilon})$ for the FA over each error threshold ε in {0.001, 0.01, 0.1, 1}mm, for the default (points-based) and the adapted (segments-based) techniques. As can be seen, the difference between uncompressed and compressed streamlines quickly rises with the error threshold when using the points-based method. This can be partly explained because more points are kept in curved sections of streamlines, where the FA is typically lower than in linear sections. That implies that voxels in curved sections will have a more substantial contribution to the mean FA value than they should. However, it is clear that using the segments-based technique yields mean values that are nearly identical to those obtained on uncompressed streamlines, at least for reasonable error values. A difference can still be seen on higher error thresholds (1mm) because such an error threshold leaves the possibility that some segments of a streamline may be shifted by up to 1 voxel. In that case, the mean FA value will of course be changed if the new path of that segment goes through a part of the brain with different FA. Table 2 shows the number of voxels considered to be part of the left Inferior Longitudinal Fasciculus (ILF) according to the 2 techniques, over various error thresholds. One can see that a points-based technique greatly underestimates the number of voxels occupied by the bundle, even in the uncompressed case. This discrepancy between the 2 uncompressed counts is partly caused by segments of streamlines crossing over parts of a voxel without having a point in that voxel. This is clearly visible in Figure 1b, where we see that only 5 voxels were selected with the points-based technique because multiple voxels didn't have at least one point inside them. Those cases cannot be detected and accounted for using a points-based technique. On the other hand, Figure 1c shows that the Bresenham-style technique correctly found all voxels. The astute reader will also notice that the volume of the bundle compressed with the 0.01mm and 1mm error rates is actually increasing when measured with the segments-based technique. That is expected, because some streamlines that curve can take a shortcut for the curve, given a large enough error threshold. If no uncompressed streamline was going through those voxels, the count will then be impacted.

	0.001	**0.01**	**0.1**	**1**
Points-based	0,086%	2,206%	6,14%	11,81%
Segments-based	0,00015%	0,00699%	0,13627%	2,23049%

Table 1: Differences between uncompressed and compressed FA, averaged over all bundles

	Uncompressed	**0.01**	**0.1**	**1**
Points-based	8160	7448 (-8.72%)	4704 (-42.35%)	2331 (-71.43%)
Segments-based	8928	8933 (+0.05%)	8922 (-0.06%)	9344 (+4.65%)

Table 2: Number of voxels traversed by a streamline for the 2 techniques, for 3 error thresholds

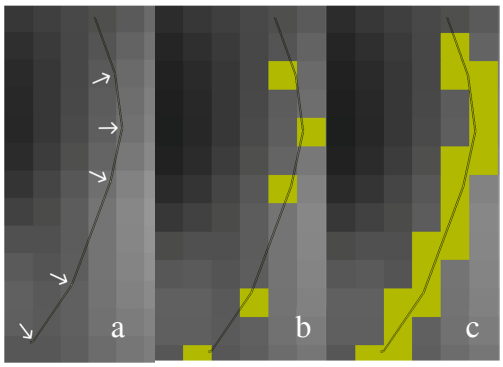

Figure 1: a) compressed streamline, arrows showing where points are; b) voxels found by the points-based method; c) voxels found by the segments-based method

Conclusion – We have shown that is it critically important to adapt *Tractometry* techniques to enable them to correctly handle compressed streamlines and different sampling of points along streamlines. Without this adaptation, all techniques that rely on a points-based analysis of the streamlines will yield biased, if not simply incorrect, results. Since the size of streamlines datasets is expected to continue increasing in the coming years (because of better techniques and higher resolution datasets), we expect the use of compressed streamlines will gain more traction. We also expect that more tractography techniques will be able to generate streamlines with variable step sizes [7], to account for various factors and perform global-tractography-like algorithms based on a lower number of points. In general, major software tools will have to follow and adapt their techniques to ensure that analyses ran on those files will still be valid and accurately reflect the underlying streamline structure.

References – [1] Presseau et al., ISMRM 2014. [2] Bresenham, IBM Systems Journal, 1965. [3] Amanatides and Woo, Eurographics 1987. [4] Tournier et al., IJIST 2012. [5] Wassermann et al., MICCAI 2013. [6] Garyfallidis et al, Front. Neuroinformatics. 2014. [7] Schreiber et al, NeuroImage 2014.