TRAKO: Efficient Transmission of Tractography Data for Visualization

Daniel Haehn1, Loraine Franke1, Fan Zhang2, Suheyla Cetin Karayumak2, Steve Pieper3, Lauren O’Donnell2, and Yogesh Rathi2

1 University of Massachusetts Boston
2 Harvard Medical School
3 Isomics, Inc.

Abstract. Fiber tracking produces large tractography datasets that are tens of gigabytes in size consisting of millions of streamlines. Such vast amounts of data require formats that allow for efficient storage, transfer, and visualization. We present TRAKO, a new data format based on the Graphics Layer Transmission Format (gITF) that enables immediate graphical and hardware-accelerated processing. We integrate a state-of-the-art compression technique for vertices, streamlines, and attached scalar and property data. We then compare TRAKO to existing tractography storage methods and provide a detailed evaluation on eight datasets. TRAKO can achieve data reductions of over 28x without loss of statistical significance when used to replicate analysis from previously published studies.

Keywords: compression, diffusion imaging, tractography

1 Introduction

Diffusion-weighted magnetic resonance imaging (MRI) allows estimation of the brain’s white matter properties [2]. Fiber tracking methods [3] then produce clusters of streamlines corresponding to 3D fiber bundles (Fig. 1). Each fiber in these bundles is a line with a collection of $x$, $y$, $z$ coordinates, typically represented using 32-bit floating point numbers. Researchers may attach scalars to these coordinates (per-vertex) to record values such as estimates of local tissue integrity. These values can be of arbitrary dimension, size, and data type. Researchers may also attach many different property values to individual streamlines (per-fiber). Modern tractography studies with scalars and properties can result in datasets that are tens of gigabytes in size per subject [25]. Storing such data can be expensive while transferring and processing the data for visualization can be inefficient. To optimize the costs and minimize overall delays, we need to explore compression techniques and their effect on tractography based neuroanalysis.

Currently existing compression methods are using two approaches by either reducing the number of fiber tracts in a dataset by downsampling [11, 12, 15, 28, 22, 30] or compressing the data of individual fibers [14, 23, 7, 13, 20, 5]. However, none of the existing methods approaches the problem...
Fig. 1: Examples of diffusion tractography fiber tracts. (left) separate fiber clusters, (right) wholebrain tractography. Individual tracts are colored by anatomical orientation.

from the perspective of optimizing storage for graphical processing, nor do they leverage recent developments in data representation and compression standards for spatial computing. In this paper, we present TRAKO, a new tractography data format for efficient transmission and visualization. TRAKO is based on the fully extendable glTF \cite{26} container, which among other things is designed to minimize runtime processing when uploading data to a graphical processing unit (GPU). Furthermore, TRAKO applies state-of-the-art 3D geometry compression techniques which allow to explicitly control the data reduction (lossiness). In addition, TRAKO compresses vertices of each fiber tract and attached scalars and properties, an advantage over existing tractography compression methods.

We compare TRAKO against two compression schemes that are specifically designed for fiber tracts: \textit{zfib} \cite{23} and \textit{qfib} \cite{19}. Zfib, which is now part of the Dipy \cite{9} library, reduces the number of vertices in each fiber tract but does not change the vertices itself (downsampling). Qfib is a recently presented algorithm that compresses individual vertices and allows to choose between a 8 bit and 16 bit precision. Neither zfib nor qfib support the compression of attached per-vertex scalars or per-fiber properties. In contrast, TRAKO encodes vertices and all attached values with the Draco algorithm \cite{4} that combines quantization, prediction schemes, and attribute encoding.

Most tractography compression schemes are configurable to trade-off information loss and data size. Therefore, we explore different settings of TRAKO to encode data points with the goal of sufficiently preserving accuracy for quantitative analysis. We test and evaluate the methods TRAKO, zfib, and qfib on multiple datasets to measure the loss of vertices, scalars, and properties after encoding. TRAKO reduces data sizes by a factor of 10-28x with an average error that is lower than the voxel size of the original diffusion MRI. We further perform a sensitivity analysis and replicate two previously published tractography studies with compressed versions of the original data. We find that compressed fiber tracts are very suitable for real-world processing. Finally, we publicly release all our data, code, experiments, and results.\footnote{https://github.com/google/draco}

\footnote{https://pypi.org/project/trako/}
2 Data Format

2.1 Structure

The TRAKO data format with file extension .tko, is built off the Graphics Library Transmission Format (glTF) \[26\], a JSON-based royalty-free format for efficient transmission and loading of 3D scenes (i.e. to be the ”JPEG of 3D”). glTF containers include mechanisms to store computer graphics scenes but the specification is fully extendable and flexible.

For TRAKO, we define a set of fiber tracts using the glTF mesh data structure (Fig. 2). This structure is defined with arrays of primitives corresponding directly to data required for draw calls of a GPU. Specifically, we use the \texttt{POSITION} attributes (Vec3 floats) to store the vertices of the fiber tracts and then map them to individual streamlines using the \texttt{INDICES} property. Since TRAKO files are valid glTF files as well, we can leverage the whole glTF ecosystem that includes validators, viewers, optimizers, and converters. For examples, we can convert ASCII JSON .tko-files to binary versions with existing converter tools such as the Cesium glTF Pipeline\[6\] or gltf-pack\[7\].

![Diagram](https://example.com/diagram.png)

Fig. 2: The TRAKO data format stores fiber tracts in a standardized glTF \[26\] container. This way, we can use existing mechanisms such as position attributes and indices to store the streamlines as buffers. These buffers are accessible and configurable through accessors and bufferviews and are immediately ready for transmission to the GPU. glTF containers are fully extendable and allow TRAKO to support the storage of per-vertex scalars, per-fiber properties, and metadata in any format.

\[6\] https://github.com/CesiumGS/gltf-pipeline
\[7\] https://github.com/zeux/meshoptimizer
Parameter exploration of TRAKO on the ISMRM 2015 dataset with an original size of 34.1 Megabytes. We test the default parameters of TRAKO (blue, quantization bits (q.bit) 14, compression level (cl) 1), a variation that only compresses the vertices (XYZ, orange), one that compresses XYZ and indices (Ind., green), the same but with compression level 0 for faster speed (red), and finally, TRAKO converted to binary using the glTF Pipeline (purple). The lower left corner indicated low errors and high compression rates. The numbers in the plot indicate the quantization bits.

2.2 Compression

Internally, TRAKO leverages the Draco compression scheme that enables the compression of meshes and point cloud data by combining multiple techniques. For meshes, Draco uses the Edgebreaker algorithm [27]. For point clouds, Draco offers a kd-tree based encoding that re-arranges all points, or a sequential encoding that preserves their order. Preserving the order is important for tractography data since we need to keep track of all vertices and any mapped values along the streamlines. We integrated Draco’s sequential encoding method to TRAKO. This method combines entropy reduction using a configurable quantization rate of 1-31 bits with prediction schemes that compute differences between stored values (similar to delta encoding) [8].

There are two main parameters to control the compression. The quantization rate controls how many bits are used to encode individual values (default: 14). Higher rates allow for greater data precision but yield larger data sizes. We explored quantization rates in the range of 7-14 bits as part of an initial parameter exploration (Fig. 3). The second main parameter of Draco is the compression level from 0-10. This level can be used to trade off encoding speed with better...
compression. Since speed is not of primary importance, we always select the maximum compression level of 10.

2.3 File Formats

Our TRAKO implementation supports conversion and on-the-fly compression of data (trakofy), decompression of data (untrakofy), and comparison of an uncompressed file to the original source file (tkompare). These tools support various widely used tractography data formats including VTK, VTP, TCK, and TRK files. In addition, we provide a reusable Python package to allow integration of TRAKO with other software systems or for extension to support other file formats. The glTF standard itself provides a standard mechanism for embedding domain-specific data within glTF JSON structures, and there exists a wide range of extensions to support features such as advanced graphical rendering, animation, and multiple levels-of-detail. The same approach can be used with TRAKO to embed custom experimental metadata without breaking compatibility with the core standard.

3 Evaluation and Results

3.1 Performance

| Dataset                  | Streamlines | Vertices | Tracking Scalars | Properties | Format | Size   |
|--------------------------|-------------|----------|-----------------|------------|--------|--------|
| qfib-data [19]           | 480,000     | 171,666,931 | iFOD1           | -          | TCK    | 734.21M |
| ISMRM2015 [16]           | 200433      | 19,584,878  | synthetic       | -          | TCK    | 16.55M  |
| HCP (anatomical tracts)  | 480,000     | 171,666,931 | iFOD1           | -          | TCK    | 734.21M |
| ADHD (whole brain tract) | 7,410       | 364,002   | UKF             | -          | TCK    | 0.15M   |
| ADHD (whole brain tract) | 153,537     | 5,650,084  | UKF             | 4          | VTK    | 530.00M |
| ADHD (whole brain tract) | 198,986,754 | 2,971,986,861 | UKF           | 9          | VTP    | 149,678.00M |

Table 1: We evaluate TRAKO on eight different datasets. The top five datasets only contain streamlines and vertices (TCK format). The bottom three datasets include attached per-vertex scalars and per-fiber properties, resulting in large data sizes (VTK and VTP formats). Abbreviations: UKF - unscented Kalman Filter tractography; iFOD1: 1st order integration over fiber orientation distributions tractography; HCP - Human Connectome Project (one example young healthy adult); dHCP - Developing Human Connectome Project (one example neonate); ADHD - Attention deficit hyperactivity disorder dataset (including 30 ADHD patients and 29 healthy control subjects)

[8] https://vtk.org/wp-content/uploads/2015/04/file-formats.pdf
[9] https://mrtrix.readthedocs.io/en/latest/getting_started/image_data.html
[10] http://www.trackvis.org/dtk/?subsect=format
[11] https://github.com/KhronosGroup/gltf/blob/master/extensions/README.md
We consider the TRAKO, zfib, and qfib data formats for efficient tractography storage. We test these formats with eight different datasets and compute the following metrics to measure compression and data loss. Five datasets only include fiber tracts (Table 1, top) while three datasets include mapped per-vertex scalars and per-fiber properties (Table 1, bottom).

Following the qfib paper [19], we use the compression ratio $C_r$. This ratio yields the percentage in reduction of compressed to original size.

$$C_r = 100 \times (1 - \frac{\text{compressed size}}{\text{original size}})$$  (1)

Further, to facilitate comparison with other published results, we compute the compression factor $C_f$ to compare the size of original and compressed data.

$$C_f = \frac{\text{original size}}{\text{compressed size}}$$  (2)

TRAKO and qfib do not change the number of points and we calculate individual data loss by measuring point-wise errors as $L^2$-norm.

$$E = \sum |f_i - g_i|,$$  (3)

for two fiber tracts $f$ and $g$ with the same number of vertices.

We also calculate the endpoint errors by only considering the start and end points of each fiber. This allows to compare with zfib, a method that changes the numbers of fiber points.

Fig. 4: On the five datasets that include only streamlines and vertices, TRAKO produces a comparable compression factor to qfib (and superior to zfib), and in average, a lower mean error (4 out of 5 cases). TRAKO is the only method that supports the three datasets with attached per-vertex scalars and per-fiber properties.
Suprathreshold fiber cluster whole brain tractography statistics. In this experiment, we assessed if group-wise tractography differences can be preserved using restored data after applying TRAKO (compress and restore). To do so, we performed a suprathreshold fiber cluster (STFC) statistical analysis.
the ADHD dataset to identify group differences in the whole brain tractography between the ADHD and healthy population. The STFC method first performs a data-driven tractography parcellation to obtain white matter fiber parcels (a total of 1416 tract parcels). Diffusion measure of interest, i.e., return-to-the-origin probability (RTOP) [21], was extracted from each fiber parcel and tested between the two populations using a student t-test. Then, a non-parametric permutation test was performed to correct for multiple comparisons across all fiber parcels. Overall, the output of the analysis includes STFCs, i.e. a fiber cluster of multiple fiber parcels that are significantly different when comparing the RTOP diffusion measure ($p < 0.05$).

We performed the STFC analysis on the original tractography data, as well as the restored data. Each individual fiber parcel was compressed and decompressed using TRAKO using the default options, yielding the compression factors and error rates as reported in Table 2. In the original data, there were two sets of STFCs (corrected $p$ values 0.015 and 0.035, respectively). In the restored data, the same sets of STFCs were identified (corrected $p$ values 0.009 and 0.028, respectively), suggesting good performance of TRAKO on preserving group-wise tractography differences.

**Bhattacharyya overlap distance.** To ensure TRAKO does not alter the fiber tract points, we have additionally implemented the Bhattacharyya analysis and computed the overlap score ($B$) to quantify the agreement between the original and restored tract points $[24, 6]$:

$$B = \frac{1}{3} \left( \int \sqrt{P_o(x)P_r(x)} \, dx + \int \sqrt{P_o(y)P_r(y)} \, dy + \int \sqrt{P_o(z)P_r(z)} \, dz \right),$$

with the ground truth probability distribution $P_o(.)$ of the original fiber tract, $P_r(.)$ the probability distribution from the restored fiber tract, and the fiber coordinates $x = (x, y, z) \in \mathbb{R}^3$. $B$ becomes 1 for a perfect match between two fiber bundles from original and restored data and 0 for no overlap at all.

We performed the Bhattacharyya overlap distance analysis on the corpus callosum (CC) tract, which was parcellated using [31] for both original and restored fiber tracts. We then computed the overlap score between the original and restored CC in all subjects (0.99±1.6231e-04). The very high overlap between original and restored tract points indicates that TRAKO can successfully preserve this information during compression.

### 4 Conclusions

We have introduced TRAKO, a data format for tractography fiber tracts that allows for high data size reduction with low information loss. Built-off the glTF community standard to allow immediate GPU processing, TRAKO is also the only data format that compresses tractography data with attached per-vertex scalars and per-fiber properties. In the future we plan to use TRAKO to distribute tractography datasets, thus reducing download times for interactive visualization and data transmission costs for large-scale analysis. To encourage community

D. Haehn et al.
adoption, we release TRAKO and our results as free and open research at https://github.com/haehn/trako/.
References

[1] Guy Alexandroni et al. “The fiber-density-coreset for redundancy reduction in huge fiber-sets”. In: NeuroImage 146 (2017), pp. 246–256.

[2] Peter J Basser, James Mattiello, and Denis LeBihan. “MR diffusion tensor spectroscopy and imaging”. In: Biophysical journal 66.1 (1994), pp. 259–267.

[3] Peter J Basser et al. “In vivo fiber tractography using DT-MRI data”. In: Magnetic resonance in medicine 44.4 (2000), pp. 625–632.

[4] Jamieson Brettle and Frank Galligan. Introducing Draco: compression for 3D graphics. 2017.

[5] Cesar F Caiafa and Franco Pestilli. “Multidimensional encoding of brain connectomes”. In: Scientific reports 7.1 (2017), pp. 1–13.

[6] Suheyla Cetin Karayumak, Marek Kubicki, and Yogesh Rathi. “Harmonizing Diffusion MRI Data Across Magnetic Field Strengths: 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part III”. In: Sept. 2018, pp. 116–124. ISBN: 978-3-030-00930-4. DOI: 10.1007/978-3-030-00931-1_14.

[7] Moo K Chung et al. “Efficient parametric encoding scheme for white matter fiber bundles”. In: 2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE. 2009, pp. 6644–6647.

[8] Michael F Deering. Compression of three-dimensional graphics data including quantization, delta-encoding, and variable-length encoding. US Patent 5,867,167. Feb. 1999.

[9] Eleftherios Garyfallidis et al. “Dipy, a library for the analysis of diffusion MRI data”. In: Frontiers in neuroinformatics 8 (2014), p. 8.

[10] Eleftherios Garyfallidis et al. “Quickbundles, a method for tractography simplification”. In: Frontiers in neuroscience 6 (2012), p. 175.

[11] Pietro Gori et al. “Parsimonious approximation of streamline trajectories in white matter fiber bundles”. In: IEEE transactions on medical imaging 35.12 (2016), pp. 2609–2619.

[12] Pamela Guevara et al. “Robust clustering of massive tractography datasets”. In: NeuroImage 54.3 (2011), pp. 1975–1993.

[13] Kuldeep Kumar and Christian Desrosiers. “A sparse coding approach for the efficient representation and segmentation of white matter fibers”. In: 2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI). IEEE. 2016, pp. 915–919.

[14] Peter Lindstrom. “Fixed-rate compressed floating-point arrays”. In: IEEE transactions on visualization and computer graphics 20.12 (2014), pp. 2674–2683.

[15] Meizhu Liu, Baba C Vemuri, and Rachid Deriche. “Unsupervised automatic white matter fiber clustering using a Gaussian mixture model”. In: 2012 9th IEEE International Symposium on Biomedical Imaging (ISBI). IEEE. 2012, pp. 522–525.
[16] Klaus Maier-Hein et al. “Tractography Challenge ISMRM 2015 High-resolution Data”. May 2017. DOI: 10.5281/zenodo.579933 URL: https://doi.org/10.5281/zenodo.579933

[17] Antonios Makropoulos et al. “The developing human connectome project: A minimal processing pipeline for neonatal cortical surface reconstruction”. In: Neuroimage 173 (2018), pp. 88–112.

[18] Corentin Mercier et al. “Progressive and efficient multi-resolution representations for brain tractograms”. In: 2018.

[19] Corentin Mercier et al. “QFib: Fast and Efficient Brain Tractogram Compression”. In: Neuroinformatics (2020).

[20] Gali Zimmerman Moreno et al. “Sparse Representation for White Matter Fiber Compression and Calculation of Inter-Fiber Similarity”. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer. 2016, pp. 133–143.

[21] Lipeng Ning, Carl-Fredrik Westin, and Yogesh Rathi. “Estimating diffusion propagator and its moments using directional radial basis functions”. In: IEEE transactions on medical imaging 34.10 (2015), pp. 2058–2078.

[22] Emanuele Olivetti et al. “Comparison of distances for supervised segmentation of white matter tractography”. In: 2017 International Workshop on Pattern Recognition in Neuroimaging (PRNI). IEEE. 2017, pp. 1–4.

[23] Caroline Presseau et al. “A new compression format for fiber tracking datasets”. In: NeuroImage 109 (2015), pp. 73–83.

[24] Yogesh Rathi et al. “Diffusion Propagator Estimation from Sparse Measurements in a Tractography Framework”. In: vol. 16. Pt 3. MICCAI 2013, pp. 510–7.

[25] Francois Rheault, Jean-Christophe Houde, and Maxime Descoteaux. “Visualization, interaction and tractometry: dealing with millions of streamlines from diffusion MRI tractography”. In: Frontiers in neuroinformatics 11 (2017), p. 42.

[26] Fabrice Robinet et al. “gltf: Designing an open-standard runtime asset format”. In: GPU Pro 5 (2014), pp. 375–392.

[27] Jarek Rossignac. “Edgebreaker: Connectivity compression for triangle meshes”. In: IEEE transactions on visualization and computer graphics 5.1 (1999), pp. 47–61.

[28] Viviana Siless et al. “Anatomicuts: Hierarchical clustering of tractography streamlines based on anatomical similarity”. In: NeuroImage 166 (2018), pp. 32–45.

[29] David C Van Essen et al. “The WU-Minn human connectome project: an overview”. In: Neuroimage 80 (2013), pp. 62–79.

[30] Fan Zhang et al. “An anatomically curated fiber clustering white matter atlas for consistent white matter tract parcellation across the lifespan”. In: NeuroImage 179 (2018), pp. 429–447.

[31] Fan Zhang et al. “Suprathreshold fiber cluster statistics: Leveraging white matter geometry to enhance tractography statistical analysis”. In: NeuroImage 171 (2018), pp. 341–354.