TRAKO: Efficient Transmission of Tractography Data for Visualization

Daniel Haehn[1][0000−0001−9144−3461], Loraine Franke[1][0000−0001−8560−2729], Fan Zhang[3][0000−0002−5032−6039], Suheyla Cetin-Karayumak[3][0000−0002−3315−5821], Steve Pieper[2][0000−0003−4193−9578], Lauren J. O'Donnell[3][0000−0003−0197−7801], and Yogesh Rathi[3][0000−0002−9946−2314]

[1] University of Massachusetts Boston, {haehn,franke}@mpsych.org
[2] Isomics, Inc., pieper@isomics.com
[3] Harvard Medical School,
{fzhang,skarayumak,odonnell,yogesh}@bwh.harvard.edu

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 (glTF) 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 [26]. 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.

Fig. 1: Examples of diffusion tractography fiber tracts. (left) separate fiber clusters, (right) wholebrain tractography. Individual tracts are colored by anatomical orientation.

Currently existing compression methods are using two approaches by either reducing the number of fiber tracts in a dataset by downsampling [12, 1, 11, 13, 19, 16, 30, 23, 32] or compressing the data of individual fibers [15, 24, 7, 14, 21, 5]. However, none of the existing methods approaches the problem 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 [27] 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: *zfib* [24] and *qfib* [20]. Zfib, which is now part of the Dipy [10] 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 [4] that combines quantization, prediction schemes, and attribute encoding.[4]

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

---

[4] `https://github.com/google/draco`

tracts are very suitable for real-world processing. Finally, we publicly release all our data, code, experiments, and results[5].

Scene
|
Mesh
  Attributes
    POSITION ————————————→ Accessor
    PER-VERTEX SCALAR 1 ———— |
                      ⋮   Bufferview
    PER-VERTEX SCALAR N ———— |
  INDICES ————————————————   Buffer
  EXTRAS ———————————————— |
    PROPERTIES ————————————
      PER-FIBER PROPERTY 1 ————
                      ⋮
      PER-FIBER PROPERTY M ————
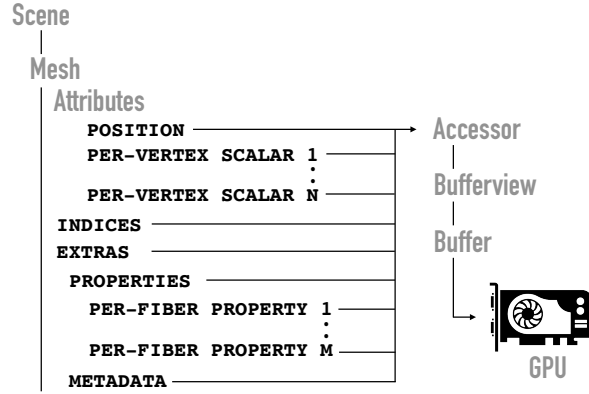   METADATA ————————————————   GPU

Fig. 2: The TRAKO data format stores fiber tracts in a standardized glTF [27] 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.

2   Data Format

2.1   Structure

The TRAKO data format with file extension .tko, is built off the Graphics Library Transmission Format (glTF) [27], 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 POSITION attributes (Vec3 floats) to store the vertices of the fiber tracts and then map them to individual streamlines using the 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

---

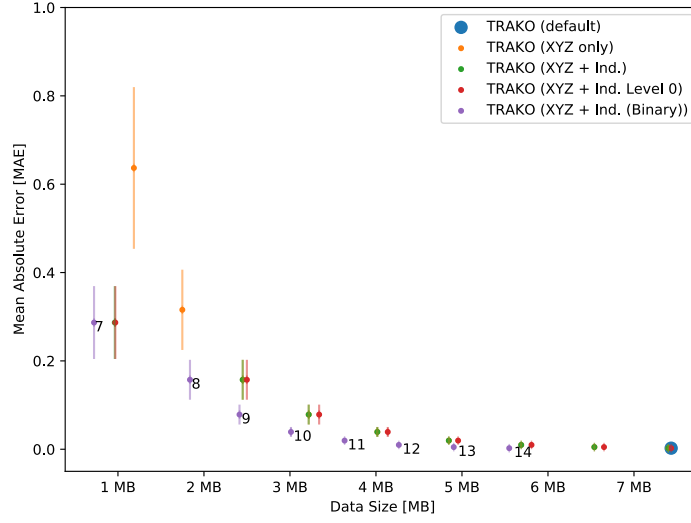[5] `https://github.com/bostongfx/trako`, current version: 0.3.4.dev9

Fig. 3: 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.

convert ASCII JSON .tko-files to binary versions with existing converter tools such as the Cesium glTF Pipeline[6] or gltf-pack[7].

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 [28]. 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, 9, 29].

There are two main parameters to control the compression. The quantization rate controls how many bits are used to encode individual values (default: 14).

---

[6] https://github.com/CesiumGS/gltf-pipeline
[7] https://github.com/zeux/meshoptimizer

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. The compression also works for streamlines with a variable step size even though the resulting file sizes will be slightly larger.

2.3   File Formats

Our TRAKO implementation supports conversion and on-the-fly compression of data (*trackofy*), 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[8], TCK[9], and TRK[10] 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[11]. The same approach can be used with TRAKO to embed custom experimental metadata without breaking compatibility with the core standard. By default, our encoding matches the coordinate system of the underlying image data but it also fully supports adding transformation matrices or image space information.

3   Evaluation and Results

3.1   Performance

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 [20], we use the compression ratio C_r. This ratio yields the percentage in reduction of compressed to original size.

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

---

[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

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)

Dataset	Streamlines	Vertices	Tracking	Scalars	Properties	Format	Size
qfib-data [20]	480,000	171,666,931	iFOD1	-	-	TCK	734.21M
ISMRM2015 [17]	200,433	19,584,878	synthetic	-	-	TCK	16.55M
HCP (anatomical tracts) [31, 32]	7,410	364,002	UKF	-	-	TCK	0.15M
ADHD (whole brain tract) [33]	199,240	30,897,382	UKF	-	-	TCK	1.23M
dHCP (whole brain tract) [18]	153,537	5,650,084	UKF	-	-	TCK	187.08M
HCP [31]	7,410	364,002	UKF	5	5	VTP	33.00M
ADHD [33]	19,898,754	2,971,986,861	UKF	9	5	VTP	149,678.00M
dHCP [18]	153,537	5,650,084	UKF	4	-	VTK	530.00M

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

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

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.

$$E = \sqrt{\sum_i |f_i - g_i|^2}, \qquad (3)$$

for two fiber tracts, f before and g after compression, with the same number of vertices ($i \in N$). 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.

3.2   Sensitivity Analysis

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 [33] on the ADHD dataset to identify group differences in the whole brain tactography 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) [22], was extracted from each fiber parcel and tested between

Table 2: Detailed comparison of qfib (8bit and 16 bit), zfib/Dipy, and TRAKO (JSON and Binary). The first five datasets only contain fiber tracts. TRAKO yields a lower mean error in 4 out of 5 datasets with compression rates of up to 28×. The bottom three datasets include per-vertex scalars and per-fiber properties. Lowest errors are **bold**, and second-to-lowest are *italic*. zfib/Dipy yields the lowest endpoint error but changes the number of fiber points. For 4 out of 5 datasets, TRAKO offers the lowest error and second-to-lowest endpoint error.

	Size	Ratio C_r	Factor C_f	Error min	max	mean	Endpoints Error min	max	mean	Timings [min.] enc.	dec.
qfib-data	734.21M										
qfib (8bit) [20]	22.9M	96.881%	32.064×	0.0	0.758	0.058±0.023	0.0	0.74	0.038±0.038	476.644	65.973
qfib (16bit) [20]	44.24M	93.975%	16.597×	0.0	0.019	**0.002±0.001**	0.0	0.017	*0.001±0.001*	476.738	66.711
zfib/Dipy [24]	118.65M	83.839%	6.188×	-	-	-	0.0	0	**0.0±0.000**	95.14	2997.115
TRAKO	46.18M	93.71%	15.899×	0.0	0.018	0.01±0.003	0.0	0.018	0.01±0.002	273.328	190.095
TRAKO (Binary)	34.63M	95.283%	21.199×	0.0	0.018	0.01±0.003	0.0	0.018	0.01±0.002	272.421	188.598
ISMRM2015	16.55M										
qfib (8bit) [20]	0.98M	94.103%	16.957×	0.0	59.541	11.686±6.327	0.0	59.522	10.501±10.501	269.627	45.37
qfib (16bit) [20]	1.74M	89.465%	9.492×	0.0	59.316	11.61±6.293	0.0	59.296	10.443±10.443	272.044	48.281
zfib/Dipy [24]	8.69M	47.512%	1.905×	-	-	-	0.0	0.0	**0.0±0.000**	46.237	354.191
TRAKO	1.46M	91.2%	11.364×	0.0	0.233	**0.092±0.027**	0.001	0.229	*0.092±0.015*	32.803	48.85
TRAKO (Binary)	1.09M	93.401%	15.154×	0.0	0.233	**0.092±0.027**	0.001	0.229	*0.092±0.015*	16.708	26.481
HCP (tracts only)	0.15M										
qfib (8bit) [20]	0.01M	94.442%	17.992×	0.0	18.687	0.418±0.251	0.0	18.687	0.351±0.351	9.432	2.847
qfib (16bit) [20]	0.01M	91.362%	11.576×	0.0	116.186	0.456±0.321	0.0	116.186	0.451±0.451	9.571	3.137
zfib/Dipy [24]	0.08M	48.524%	1.943×	-	-	-	0.0	0.0	**0.0±0.000**	1.498	0.305
TRAKO	0.01M	91.385%	11.608×	0.001	0.27	**0.097±0.028**	0.005	0.247	*0.097±0.016*	0.923	0.949
TRAKO (Binary)	0.01M	91.731%	12.093×	0.001	0.27	**0.097±0.028**	0.005	0.247	*0.097±0.016*	1.314	1.206
ADHD Single (tracts only)	1.23M										
qfib (8bit) [20]	0.04M	96.38%	27.624×	0.0	72.832	1.762±1.391	0.0	71.284	1.496±1.496	165.298	40.044
qfib (16bit) [20]	0.08M	93.286%	14.895×	0.0	120.936	4.123±3.119	0.0	120.936	3.331±3.331	165.486	40.681
zfib/Dipy [24]	0.25M	80.058%	5.014×	-	-	-	0.0	0.0	**0.0±0.000**	36.811	12.235
TRAKO	0.06M	95.349%	21.501×	0.0	0.276	**0.08±0.023**	0.001	0.264	*0.079±0.013*	61.298	40.806
TRAKO (Binary)	0.04M	96.523%	28.76×	0.0	0.276	**0.08±0.023**	0.001	0.264	*0.079±0.013*	66.261	42.501
dHCP (tracts only)	187.08M										
qfib (8bit) [20]	9.33M	95.01%	20.041×	0.0	53.695	0.452±0.235	0.0	53.695	0.282±0.282	14.954	2.027
qfib (16bit) [20]	14.68M	92.154%	12.746×	0.0	53.381	0.475±0.375	0.0	53.381	0.442±0.442	15.647	2.408
zfib/Dipy [24]	73.68M	60.616%	2.539×	-	-	-	0.0	0.0	**0.0±0.000**	23.993	2532.927
TRAKO	12.7M	93.213%	14.734×	0.001	0.273	**0.152±0.043**	0.005	0.271	*0.152±0.025*	9.575	5.963
TRAKO (Binary)	9.52M	94.91%	19.645×	0.001	0.273	**0.152±0.043**	0.005	0.271	*0.152±0.025*	9.091	5.921

	Mean Error		Mean Error
HCP [31], 13.43M, C_r: 59.162%, C_f: 2.449×			
Scalars		*Properties*	
EstimatedUncertainty (N, range: 0.032-15233.791)	0.135±0.081	EmbeddingCoordinate (N, range: -4.543-3.047)	0.00026±3.72188e-05
tensor1 (N×9, range: -0.00095-0.0024)	1.121e-07±2.27e-08	ClusterNumber (N, range: 8-665)	0.4237±0.4763
tensor2 (N×9, range: -0.00087-0.0021)	8.73e-08±1.78e-08	EmbeddingColor (N, range: 0-180)	0.8776±0.4748
HemisphereLocataion (N, range: 1.0-3.0)	0.0±0.0	TotalFiberSimilarity (N, range: 199220.9-920767.25)	8.0194±4.7547
cluster_idx (N, range: 0-39)	0.246±0.361	MeasuredFiberSimilarity (N, range: 0.00179-0.00266)	7.4e-09±4.5e-09
ADHD [33], 50,462.34M, C_r: 66.286%, C_f: 2.966×			
Scalars		*Properties*	
NormalizedSignalEstimationError (N, range: 0.0-0.05)	0.0±0.0	EmbeddingCoordinate (N×10, range: -3.18-4.93)	0.0±0.0
EstimatedUncertainty (N, range: 0.04-31041.65)	0.3±0.176	ClusterNumber (N, range: 12-768)	0.0±0.0
RTOP1 (N, range: 1.13-23901.94)	0.04±0.023	EmbeddingColor (N×3, range: 2-180)	0.869±0.511
RTOP2 (N, range: 1.32-8651.45)	0.014±0.008	TotalFiberSimilarity (N, range: 149876.58-696306.3)	5.599±3.341
RTAP1 (N, range: -13541.7-7914.96)	0.031±0.018	MeasuredFiberSimilarity (N, range: 0.0-0.0)	0.0±0.0
RTAP2 (N, range: 1.11-6820.54)	0.01±0.006		
RTPP1 (N, range: 0.71-9.88)	0.0±0.0		
RTPP2 (N, range: 0.71-15.96)	0.0±0.0		
SignalMean (N, range: 0.0-0.04)	0.0±0.0		
dHCP [18], 256.31M, C_r: 52.799%, C_f: 2.119×			
Scalars		*Properties*	
FreeWater (N, range: 0.0-1.0)	1.42e-05±9.34e-06	-	
tensor1 (N × 9, range: -0.00132-0.0031)	2.27e-07±4.63e-08		
tensor2 (N × 9, range: -0.00132-0.0043)	2.895e-07±5.9e-08		
EstimatedUncertainty (N, range: 0.0332-196.16)	0.291±0.177		

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.
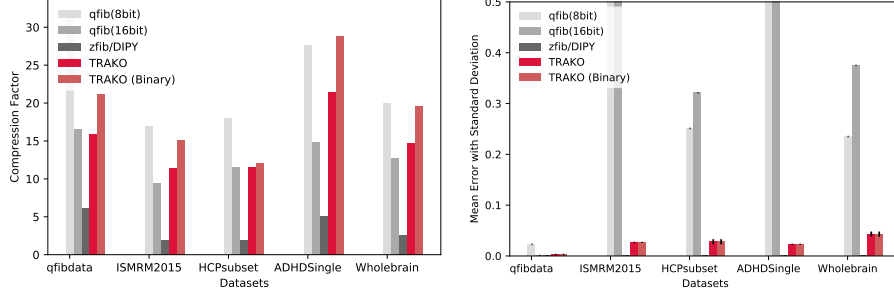
Fig. 4: On the five datasets that include only streamlines and vertices, TRAKO produces a comparable compression factor to qfib (and superior to zfib), and in average, a lower mean error (4 out of 5 cases). TRAKO is the only method that supports the three datasets with attached per-vertex scalars and per-fiber properties.

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 [25, 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 $\mathbf{x} = (x, y, z) \in \mathbb{R}^3$. 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 [33] 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 investigate standardized notation of coordinate systems and other metadata. We will then use TRAKO to distribute tractography datasets, reducing download times for interactive visualization and data transmission costs for large-scale analysis. To encourage community adoption, we release TRAKO and our results as free and open research at `https://github.com/bostongfx/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] Olivier Devillers and Pierre-Marie Gandoin. "Geometric compression for interactive transmission". In: *Proceedings Visualization 2000. VIS 2000 (Cat. No.00CH37145)*. 2000, pp. 319–326.

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

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

[12] 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.

[13] Pamela Guevara et al. "Robust clustering of massive tractography datasets". In: *Neuroimage* 54.3 (2011), pp. 1975–1993.

[14] 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.

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

[16] 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.

[17] Klaus Maier-Hein et al. *Tractography Challenge ISMRM 2015 High-resolution Data*. Zenodo, May 2017. DOI: `10.5281/zenodo.579933`. URL: `https://doi.org/10.5281/zenodo.579933`.

[18] 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.

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

[20] Corentin Mercier et al. "QFib: Fast and Efficient Brain Tractogram Compression". In: *Neuroinformatics* (2020).

[21] 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.

[22] 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.

[23] 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.

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

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

[26] 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.

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

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

[29]   Ruwen Schnabel and Reinhard Klein. "Octree-based Point-Cloud Compression". In: *Symposium on Point-Based Graphics 2006*. Ed. by M. Botsch and B. Chen. Eurographics, July 2006.

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

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

[32]   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.

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