Vision, Modeling, and Visualization (2013)
Michael Bronstein, Jean Favre, and Kai Hormann (Eds.)

Parallelized Global Brain Tractography

Stefan Philips[1,2], Mario Hlawitschka[1] and Gerik Scheuermann[1]

[1]Leipzig University, Leipzig, Germany
[2]Max Planck Institute for Human Cognitive and Brain Sciences, Leipzig, Germany

Abstract

Most brain tractography algorithms suffer from lower accuracy, because they use only information in a certain neighborhood and reconstruct the tracts independently. Global brain tractography algorithms compensate the lack of accuracy of those local algorithms in certain areas by optimizing the whole tractogram. The global tractography approach by Reisert et al. showed the best results in the Fiber Cup contest, but the runtime is still a matter for a medical application. In this paper we present the non-trivial parallelization of this global tractography algorithm. The parallelization exploits properties of the algorithm and modifies the algorithm where necessary. We compare the runtimes of the serial and the parallel variant and show that the outcomes of the parallel variant are of the same quality as those of the serial algorithm. The experiments proof also that the parallelization scales well for real world datasets.

Categories and Subject Descriptors (according to ACM CCS): Programming Techniques [D.1.3]: Concurrent Programming—Parallel programming; Computer Graphics [I.3.3]: Picture/Image Generation—Line and curve generation; Numerical Analysis [G.1.6]: Optimization—Optimization; Numerical Analysis [G.1.6]: Optimization—Simulated annealing;

1. Introduction

There are a multitude of algorithms available for performing fiber tractography in the human brain. Most of them are *local* algorithms.

Local algorithms are characterized by considering only the limited information in a voxel and its neighbors to create trajectories. This makes it difficult to obtain the correct tract direction [JJB11]. Another problem is the fact that local algorithms accumulate errors caused by noisy or isotropic diffusion measurements. In addition local algorithms reconstruct the tracts independently.

The class of *global* algorithms compensates these disadvantages by using global information. Global tractography algorithms create the tractography from the whole data at once, instead of computing it tract by tract. This approach is especially helpful for the difficult fiber configurations like crossing or kissing fibers. The lack of information coming from reduced anisotropy in these areas is compensated by using the information of the whole surrounding region.

There exist several global tractography approaches [FPM09, JWAB07, RMK09, RMA*11, ZGHG07, KMK08] that compensate the disadvantages of local algorithms. From the mentioned algorithms, the algorithm by Reisert et al. [RMA*11] stands out because of its performance in the Fiber Cup contest [FDG*11]. A previous version [KMK08] of this global tractography algorithm had a runtime of one month. By algorithmic improvements, the runtime has been decreased to the magnitude of one day or several hours [RMK09, RMA*11]. These runtimes are now much lower than in the beginning, but they are still a hindrance for the practical use. Furthermore, these time values originate from voxel resolutions of 1.7mm, whereas, in the mean time, voxel resolutions up to 1mm are possible for whole brain scans. This is a problem, since higher resolutions increase the runtime of the global tractography algorithm.

In this paper, we present a parallelization of the global tractography algorithm presented in [RMA*11]. Our approach allows the application of this algorithm in areas where the availability of the tractogram within a certain time is critical, e.g., in clinical use. Furthermore, the parallelization benefits from higher resolutions, since higher resolutions result in more simultaneous processable work packages. Although there is a vast amount of papers describing the parallelization of tractography approaches, none of them

DOI: 10.2312/PE.VMV.VMV13.097-104

covers a global approach with its special problems regarding parallelization.

Multiprocessor machines are already very common and one can expect that their distribution and the number of cores will increase in the future. Therefore, a parallelization adapts the global tractography algorithm to the requirements of current and future hardware.

2. Global Tractography

In this section, we give a brief description of the global tractography algorithm from [RMK09] and [RMA*11]. The algorithm works by optimizing an energy function, that consists of internal and external energy. The objective of the function is the optimal configuration of connected line segments that represents a brain tractogram for the given diffusion data. Whereby the internal energy depends on the segment configuration itself, and, the external energy is a function which relates the segment configuration to the measured diffusion-weighted MR image.

To create brain tractograms, tracts are created by connecting oriented line segments to fiber tracts. In the algorithm, each segment has a length, a width, a position, an orientation, and a '−'- and a '+'-end. Each of the two segment ends can be connected to another segment end. Eventually, several connected segments form a tract.

2.1. Internal energy

The internal energy ensures that the tracts are well-behaved by prefering layouts in which connected segments stay close to each other and that the curvature of the reconstructed trajectories is low. Especially, it keeps the orientation of two connected segments similar. This is done by using the following equation for every connection of two segments $X_1 = (\mathbf{x}_1, \mathbf{n}_1)$ and $X_2 = (\mathbf{x}_2, \mathbf{n}_2)$:

$$U_{\text{con}}(X_1^{\alpha_1}, X_2^{\alpha_2}) = \frac{1}{l^2}\left(\|\mathbf{x}_1 + \alpha_1 l\mathbf{n}_1 - \bar{\mathbf{x}}\|^2 + \|\mathbf{x}_2 + \alpha_2 l\mathbf{n}_2 - \bar{\mathbf{x}}\|^2\right) - L \quad (1)$$

Here, the variables denote:

- $\alpha_i \in \{-1, +1\}$ the connected end point of the segment i
- $\mathbf{x}_i \in \mathbb{R}^3$ the position of the segment i
- $\mathbf{n}_i \in S_2$ the orientation of the segment i
- l the half segment length
- $\bar{\mathbf{x}} = \frac{\mathbf{x}_1 + \mathbf{x}_2}{2}$ the midpoint of the two segment positions
- L controls the likelihood for a connection of two segments

Figure 1 shows an example segment.

The value of U_{con} in equation 1 depends on the distance of the connected endpoints $(\mathbf{x}_1 + \alpha_1 l\mathbf{n}_1, \mathbf{x}_2 + \alpha_2 l\mathbf{n}_2)$ to the midpoint $\bar{\mathbf{x}}$. The connection energy guarantees that connected endpoints do not drift apart and it ensures that the angle between two connected segments does not increase too much. This angle constraint ensures that the resulting tracts have a low curvature.

Let $\mathcal{E} = \{(X_i^{\alpha_i}, X_j^{\alpha_j})\}$ be the set of all existing connections then the resulting internal energy formula is

$$E_{\text{int}} = \sum_{(X_1^{\alpha_1}, X_2^{\alpha_2}) \in \mathcal{E}} U_{\text{con}}(X_1^{\alpha_1}, X_2^{\alpha_2}).$$

2.2. External energy

The external energy depends on the difference between the measured diffusion signal and the synthetic signal created from the constructed fiber model. It ensures that the tractogram provides a good representation of the data. The synthetic signal $F_{\mathcal{M}}$ is defined on $\mathbb{R}^3 \times S_2$ and determined by the current set of N segments, more precisely by their position $\mathbf{x} \in \mathbb{R}^3$ and orientation $\mathbf{n} \in S_2$ to each other, and is defined in [RMK09] as:

$$F_{\mathcal{M}}(\mathbf{x}, \mathbf{n}) = w \sum_{i=1}^{N} e^{-c(\mathbf{n}^T \mathbf{n}_i)^2} e^{-\frac{|\mathbf{x}-\mathbf{x}_i|^2}{\sigma^2}}$$

According to [RMK09], $w \in [0, \infty)$ is the segment weight, c is the width of the segment in orientation space and the parameter $\sigma \in \mathbb{R}^+$ determines the influence of a segment in space. All three are equal for all segments.

Let $D(\mathbf{x}, \mathbf{n})$ be the actually measured diffusion signal, also depending on a position $\mathbf{x} \in \mathbb{R}^3$ and an orientation $\mathbf{n} \in S_2$. Then the squared distance of the synthetic and the measured diffusion signal is defined as

$$\|F_{\mathcal{M}} - D\|^2 = \int_{\mathbb{R}^3 \times S_2} |F_{\mathcal{M}}(\mathbf{x}, \mathbf{n}) - D(\mathbf{x}, \mathbf{n})|^2 d^3\mathbf{x} d^2\mathbf{n}$$

2.3. Proposals

During the optimization, five proposals are applied. These are (1) insertion and (2) removal of a segment, (3) a random and (4) an optimal shift of a segment, and (5) a tracking (connection) of segments. Every proposal, except insertion, first chooses a segment randomly.

The general acceptance probability of a proposal is defined as

$$p_{\text{proposal}} = \exp(-\delta E_{\text{int}} - \delta E_{\text{ext}}) p_{\text{proposal specific}}.$$

Thereby δE_{int} and δE_{ext} denote the change of internal and external energy, respectively. The factor $p_{\text{proposal specific}}$ is a probability that is related to the concrete proposal, e.g. the probability for a random shift depending on the movement distance.

Insert proposal. The insertion proposal adds a segment to a random place inside the white matter, randomly orientated.

Remove proposal. The removal proposal deletes a segment if it is not connected.

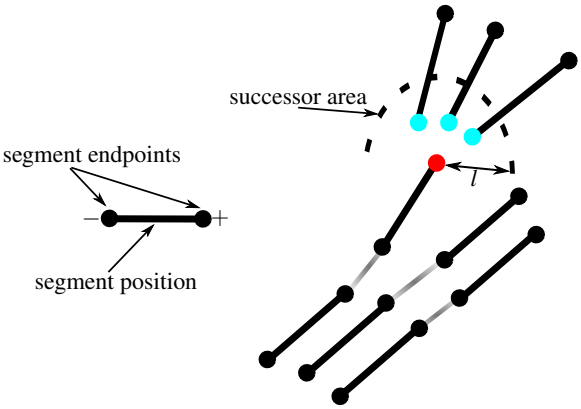

Figure 1: *Example for connected segments and the tracking proposal. The red segment endpoint tries to find another endpoint to connect to. The blue endpoints are candidates, since they are within a distance of the half-segment length l.*

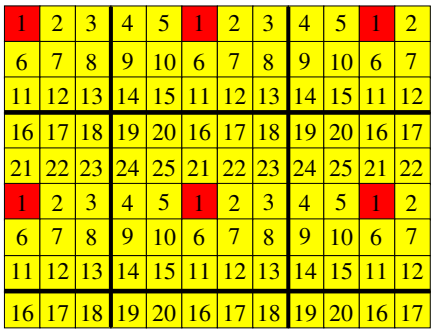

Figure 2: *Parcellation and processing scheme - Example for 2D domain: The red boxes mark the current work areas. Whereas the yellow boxes ensure the safety distance, which is two in this example. Areas with the same number can be processed simultaneously. Furthermore, the numbers stand for the processing order of the areas.*

Random-shift proposal. The random-shift proposal moves the segment's ends by a multi-Gaussian vector and then re-normalizes it to the fixed segment length.

Optimal-shift proposal. The application of the optimal-shift proposal means that the segment is aligned to its connected neighbor segments in terms of minimal internal energy.

Tracking proposal. The tracking proposal connects the segments. It starts with a random segment and a random endpoint of this segment. In Figure 1 this random endpoint is marked in red. From this endpoint, it selects the end of another segment according to a probability function that depends on the internal energy of a connection of both segment ends. The segment endpoints that can be possibly connected are colored blue in Figure 1. The maximum Euclidean distance of both segment ends is restricted to l, which will be relevant for our parallelization approach.

3. Parallelization of Global Tractography

One straightforward way to work with several threads on data is the parcellation of the domain. In theory, both energies depend on the whole domain and every proposal can be applied anywhere in domain space and to any segment. This is a problem for a spatial parallelization, because this means that a single thread writing on the model data blocks all other threads. In the following, we explain how we solve the synchronization problems of the parallelization.

3.1. Parcellation

For a spatial parallelization of the global tractography algorithm, we have to consider two problems. First, the possibility to make an energy calculation that is unaffected by simultaneously working optimization threads. Second, the locality of the proposals. A proposal must not change model properties in a different area.

The algorithm uses several parameters. The most important parameter regarding the parallelization is the segment length $2l$. The whole parcellation depends on that because the smallest assignable cubical area has a edge length of $2l$.

For the serial implementation [NSR*12] the domain is partitioned into buckets, where each bucket is a container for segments. The partitioning of the domain into these cubic buckets with edge length $2l$ is useful. It allows to get the relevant neighbored segments during the tracking and external energy calculation in a fast way.

For the parallelization, we also use these buckets as the smallest assignable work area. Since the values of the safety distances are multiples of the value l, it is also the ideal size for our work areas.

3.2. Independent Buckets

Let B be the set of all buckets. We call two buckets $b_1, b_2 \in B$ independent, if they can be processed simultaneously by two threads. This requirement is fulfilled, if the distance $d(b_1, b_2)$ between both buckets is big enough to allow the application of all types of proposals and the calculation of both kinds of energy simultaneously.

A set of independent buckets is defined as

$$S = \{b_1 \in B\} \text{ with } \bigwedge_{b_1, b_2 \in S \wedge b_1 \neq b_2} d(b_1, b_2) > d_{\text{thresh.}} \quad (2)$$

and can be considered as a work unit for a set of worker threads, whereas the buckets within the set can be considered as the work units for a thread. In Figure 2 all buckets with the same number belong to a set of independent buckets S, e.g. S_1 would consist of all buckets that are marked with a 1.

The work load of a bucket depends on the count of white matter voxels in it. For that reason, we sort the areas by descending white matter voxel count. By processing the areas in this order, a maximal parallelization of the a set of independent buckets is possible.

Let Σ be the set of bucket sets, so that all buckets with white matter are in exactly one of the bucket sets $S \in \Sigma$. The cardinality of Σ depends on the safety distance d_{safety}. For our parcellation approach within the 3D domain yields

$$\|\Sigma\| = (2d_{\text{safety}} + 1)^3. \quad (3)$$

The $+1$ stands for the work area. Since the processing of the bucket sets $S \in \Sigma$ must be serial, it is recommendable to set the safety distance d_{safety} as low as possible.

3.3. External Energy

The calculation of the external energy theoretically needs all segments. But it has been pointed out in [RMK09] already that one can omit segments above a certain distance as the contribution of these segments to the external energy difference is negligible, because of the Gaussian factor $\exp\left(-\frac{|x_i - x_j|^2}{2\sigma^2}\right)$. For the external energy calculation, this means that we do not need exclusive read access for the whole domain but only for the surrounding of our current work area. This makes it possible to work in an area without synchronization of multiple threads.

3.4. Internal Energy

The internal energy change is always restricted to a segment and its connected neighbors. A segment has a length of $2l$ and the distance between two connected endpoints is at most l. Therefore, the necessary diameter of the blocked area to calculate the internal energy change is $4l$ around the segments midpoint.

3.5. Proposals in Parallel

For each type of proposal, we explain why and how the different proposals can be restricted to a limited region.

Insert proposal. The insert proposal is not problematic. We can simply restrict the insertion of a segment to a certain region. Since a new segment is not connected, only the external energy is affected. Therefore, the above described safety distance for the external energy calculation is necessary.

Remove proposal. The remove proposal is not problematic either. The thread chooses one segment from a restricted area. Analogously to the insert proposal, the internal energy is not affected, because the algorithm does not remove connected segments.

Random-shift proposal. The random-shift proposal is theoretically problematic, since the segment could be moved elsewhere. Practically, the case of such big translations due to the Gaussian distribution of the movement vector is highly improbable. The parameter values for the Gaussian distribution $\mathcal{N}(\mu, \sigma)$ are $\mu = 0$ and $\sigma = l/8$, where l is the half segment length. This σ value is the working parameter from the original implementation of Reisert [NSR*12]. In fact, due to the Gaussian distribution with $\sigma = l/8$, in 99.7% of all cases, the movement is smaller than $3\sigma = \frac{3}{8}l$. We omit the proposal when the movement is above the safety distance.

Optimal-shift proposal. The optimal-shift proposal changes the position and orientation of a segment only in a certain distance to one or two other segments with fixed position and orientation. The maximum distance between two connected segments is the half segment length l.

Tracking proposal. The tracking proposal changes only the internal energy, but theoretically iterates through the whole domain. Therefore, we examined how many connections are created for finally accepted proposals. This showed us that 96.7% of all accepted tracking proposals of the serial algorithm consist of less than four connections. For more detail, see the diagrams in Figure 3. These diagrams show the number of segment connections, created by an accepted tracking proposal, on the abscissa. The ordinate shows the observed number of accepted tracking proposals with the respective number of connections. Since these histograms of both datasets are similar, even though they have different resolutions, we can assume that this behavior is independent of the resolution of the dataset.

The fact that the number of tracking steps is below four in almost all cases allows us to restrict the maximum number of tracking steps.

Although the maximal number of tracking steps is quite low, it would require a distance of two times the distance of the maximum number of tracking steps between each two work areas. This increases the necessary safety distance d_{safety} and according to Equation 3 $\|\Sigma\|$. As mentioned before the processing of Σ has to be done serially and $\|\Sigma\|$ should therefore be as low as possible.

Our solution for this problem is the introduction of a bidirectional tracking. In contrast to the usual one-directional tracking, the bidirectional tracking alternates the starting points in each step. Figure 4 shows this difference for the first two tracking steps. The alternating continues between the two "ends" of the tracking. Figure 4 shows a comparison of the two variants. The advantage of the bidirectional tracking is that the maximal distance to the original start of the tracking is equally distributed to both ends. As a consequence the safety distance to the work area of another thread can be smaller.

Figure 5 shows the acceptance rate of the tracking proposal during the optimization. We found that the tracking proposal acceptance rates are 10% at maximum and decreasing as the optimization progresses. As a consequence of this,

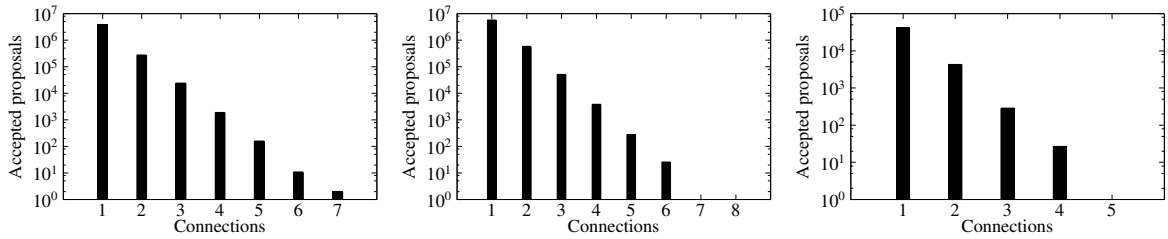

Figure 3: *Accepted tracking proposals depending on the number of created connections for dataset 1 (left), dataset 2 (middle) and dataset 3 (right).*

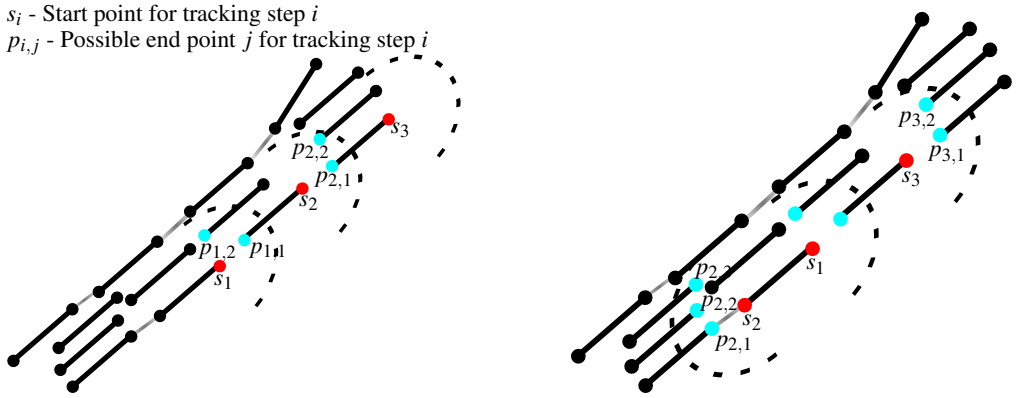

Figure 4: *Comparison of one-directional tracking (left) and bidirectional tracking (right). The two images illustrate the difference between the two tracking variants. The special characteristic of the bidirectional tracking is the alternating start point. One can see that the different start points s_2 for the second tracking step result in a lower distance between the start points s_1 and s_3.*

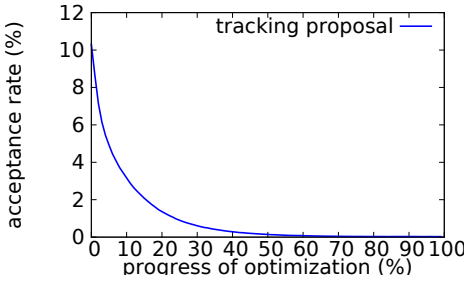

Figure 5: *The tracking proposal acceptance rate during the optimization for dataset 1.*

```
bool successful = proposeTracking();
if successful then
    if acceptTracking() and not areasHasChanged()
    then
        blockAreas();
        applyTracking();
        releaseAreas();
    end
end
```

Algorithm 1: Pseudocode of the tracking proposal for simultaneous read of an area.

a change of the model data is rarely required. According to this, it makes sense to do a simultaneous read with several threads on an area. Algorithm 1 shows the draft of our approach: During the calculation of possible segment connections a read-only access to the different areas is necessary. When the newly calculated connections shall be created, we check if the areas have changed. If not, we block the areas and write the new connections to the model data.

3.6. Processing

This section describes how the processing of the parallel approach works. The main thread iterates over all $S \in \Sigma$, starts for each worker thread, then waits till all threads are finished.

Once started, the worker threads process the buckets. Every worker thread continues until no bucket from the current $S \in \Sigma$ is left. This is drafted in Algorithm 2.

```
global S = getNextBucketSet(Σ);
while S ≠ ∅ do
    Let B ∈ S;
    S = S \ {B};
    i = stepsPerVoxel · getWMVoxels(B);
    while i > 0 do
        applyRandomProposal(B);
        i = i − 1;
    end
end
```

Algorithm 2: Pseudocode of a worker thread

3.7. Scalability Estimation

The maximal speedup is limited by the minimal $n = \|S\|$ with $S \in \Sigma$. Since the buckets $b \in S$ are the smallest distributable work unit for the threads, n should be greater or equal than the number of threads. Practically the value of n should be significantly higher, because one has to consider a synchronization overhead and the fact that the workload of each bucket depends on the number of white matter voxels it contains.

4. Implementation

Our implementation of the presented serial and parallel algorithms is available in the Open Source brain visualization system OpenWalnut [Ope]. For a fast computation of the diffusion signal interpolation, we use the "kind of barycentric interpolation" [RMK09] which is freely available in Diffusion MITK implementation of the global tractography algorithm [NSR*12]. We implemented the parallelization and its synchronization with the thread classes from OpenWalnut and some classes from the Boost library [Boo].

5. Results

To test the algorithm, we used three different datasets (see Table 1). The test data were two diffusion MRI images of a brain (Dataset 1 and 2) and one diffusion MRI image of a phantom. The phantom dataset is the fiber cup dataset from the homonymous contest [FDG*11]. Dataset 2 was created by upsampling dataset 1 with the software FLIRT (FMRIB's Linear Image Registration Tool) [JBBS02].

The purpose of all three datasets is to show that the output quality of the parallelized algorithm is fine for different resolutions. Furthermore, the different resolutions allow us to compare the behavior of the parallelized algorithm depending on varying voxel resolutions.

Table 1: *The properties and the parameters used for the processing of the datasets.*

Dataset	1	2	3
b-value (s/mm^2)	2000	2000	2000
Voxel volume (mm^3)	1.7x1.7x1.7	1x1x1	3x3x3
Voxel count			
x	128	160	64
y	128	200	64
z	72	160	3
Segment width (mm)	0.85	0.5	1.5
Segment length (mm)	2.55	1.5	4.5
Steps per WM-voxel	10000	10000	10000
Steps overall	$120 \cdot 10^7$	$312 \cdot 10^7$	$1 \cdot 10^7$

5.1. Parameters

The global tractography algorithm has many parameters. If not other stated, we used the default parameter configuration from the diffusion MITK implementation [NSR*12]. The only additional parameter of the parallelization is the safety distance. For the before mentioned default parameter a safety distance of one works fine. Different parameter configurations, especially those which create more connections per tracking proposal, might require a bigger safety distance.

5.2. Comparison of Results

The comparison of the results allows us to show that both variants of the algorithm, serial and parallelized, produce output of similar quality.

5.2.1. Energy

The energy function values of the parallel variant during the optimizations have no apparent differences to those of the serial variant (see Figure 6). This shows that the parallel algorithm reaches the same degree of quality in terms of the objective function.

5.2.2. Tractograms

The actual tractograms of the two types of the algorithm are too complex to compare them side-by-side. Therefore, we do not show all of them, but Figure 7 gives an example result of the parallel variant.

Fingerprints of the reconstructions To demonstrate that the parallelized version of the algorithm produces similar results to the serial variant, we use some characteristics of the reconstruction model. These characteristics are the distribution of the angles between connected segments and the number of connections per reconstructed tract. These properties can be seen as a *fingerprint* of the tractogram.

The purpose of the fingerprint is the possibility to compare tractography results, especially those of the serial and

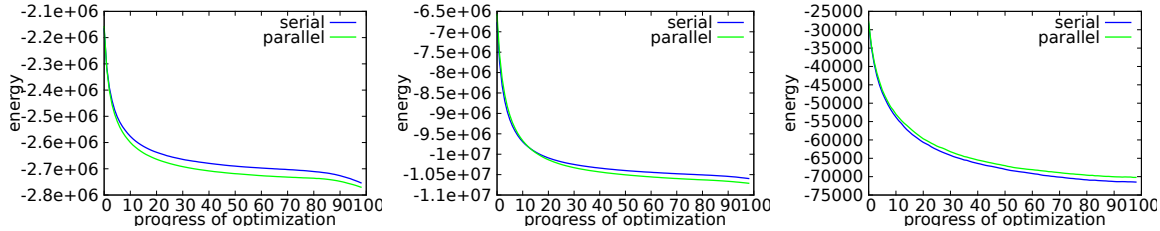

Figure 6: *The energies during the optimization. Left to Right: Dataset 1, 2, 3*

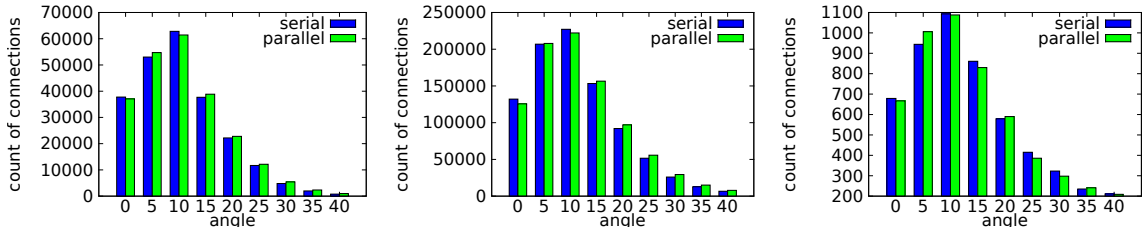

Figure 8: *These histograms show the angles between connected segments on the abscissas and their frequency on the ordinates. Left to right: Dataset 1, 2, 3*

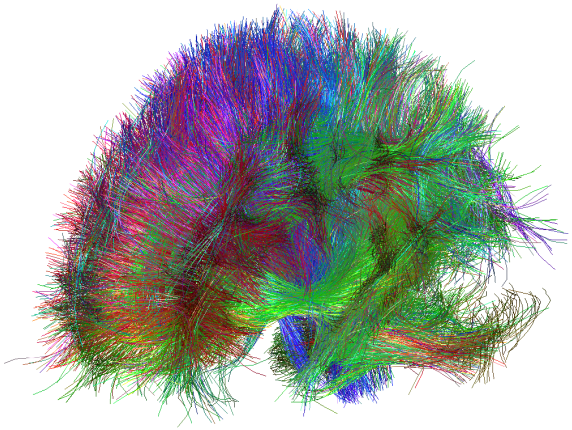

Figure 7: *The tractogram of dataset 1 created by the parallel variant of the algorithm.*

the parallel variant. These fingerprints are chosen, since they are a good characterization of the tractogram and are tightly related to the tractography algorithm. The angles between the connected segments are part of the objective function itself. The tract lengths is a prominent property of a tractogram and are closely related to the tracking proposal. As we restricted the number of connections in each calculation step created by the tracking proposal in the parallel implementation, it is especially useful to compare the tractograms of the serial and parallel variant. To illustrate the similarity of the tractograms which are produced by the serial and the parallel algorithm, we use Figure 8 and 9. The histograms of all three datasets in Figure 8 show the similarity of the tractograms regarding the segment angles. Also the histograms for the number of connections per tract in Figure 9 back up the similarity of the tractograms.

5.3. Speedup

For the test runs, we used the eight cores of Intel(R) Xeon(R) 2.4 GHz processors. Table 2 shows the runtimes and the accomplished speedups for the different datasets. The speedup values for dataset 1 and 2 are satisfying on a 8 core machine. The values also show that the implementation benefits only slightly by the use of hyperthreading.

The parallel processing of dataset 3 results in the worst speedup (factor 1.8), despite the use of eight processor cores. The reason for this is the almost two-dimensional structure of the Fiber Cup dataset. Therefore, the number of independent buckets is too low. The algorithm scales well for the real 3D datasets, which are the common data format for tractography algorithm. However, the Fiber Cup dataset shows the limitations of the scalability.

6. Summary

The global tractography algorithm proved its quality in the Fiber Cup contest. The work by Fillard et al. [FDG*11] gives a detailed and sound comparison of several tractography algorithms, including [DDKA08], [FGP*03], [RMK09], [MCCVZ99] and [JLJ*11]. Therefore, we do not repeat this

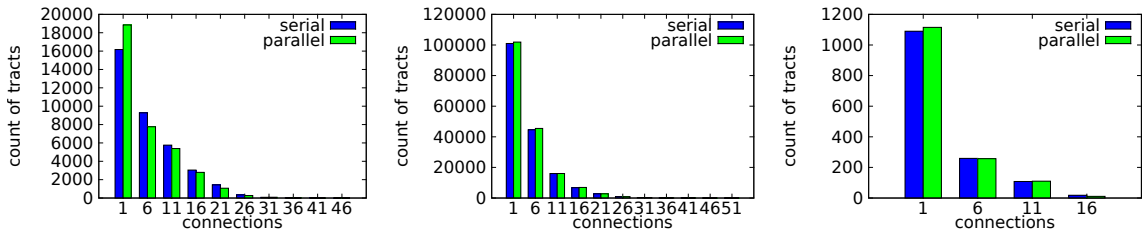

Figure 9: *These histograms show the connection count per tract on the abscissas and the frequency on the ordinates. Left to Right: Dataset 1, 2, 3*

Table 2: *Runtimes and speedups for 1, 4, 8, and 16 threads for the different datasets.*

	Dataset 1	Dataset 2	Dataset 3
1	5h 3m (1.0×)	13h 02m (1.0×)	94s (1.0×)
4	1h 21m (3.7×)	3h 21m (3.9×)	35s (2.7×)
8	41m (7.3×)	1h 44m (7.5×)	51s (1.8×)
16	37m (8.1×)	1h 29m (8.8×)	51s (1.8×)

comparison, but focus on showing that the characteristic of the produced tractograms are equal to that of the serial implementation of the algorithm.

The examination of the energy function during the optimization has shown that the parallelized variant behaves similar regarding the energy optimization. The comparison of the tractograms by their fingerprints proves that the final results have no signifcant difference to these of the serial variant. We showed that it is possible to calculate tractograms in approx. 30 minutes for realistic datasets and about 1.5h for a high-resolution dataset on an eight core machine, allowing to use global tractography in time-critical medical applications. The figures in Table 2 show that the implementation scales for eight processor cores when using real world datasets.

Acknowledgement: We thank Christian Heine and Roxana Bujack for proofreading.

References

[Boo] www.boost.org. visited 11th June 2013. 6

[DDKA08] DESCOTEAUX M., DERICHE RACHID D., KNOESCHE T., ANWANDER A.: Deterministic and probabilistic tractography based on complex fiber orientation distributions. *IEEE Transactions in Medical Imaging* (2008). accepted July 6th 2008, in press. 7

[FDG*11] FILLARD P., DESCOTEAUX M., GOH A., GOUTTARD S., JEURISSEN B., MALCOLM J., RAMIREZ-MANZANARES A., REISERT M., SAKAIE K., TENSAOUTI F., YO T., MANGIN J.-F., POUPON C.: Quantitative evaluation of 10 tractography algorithms on a realistic diffusion MR phantom. *NeuroImage 56*, 1 (2011), 220 – 234. 1, 6, 7

[FGP*03] FILLARD P., GILMORE J., PIVEN J., LIN W., B G. G. A.: Quantitative analysis of white matter fiber properties along geodesic paths. In *Proc. of Medical Image Computing and Computer-Assisted Intervention* (2003), Springer, pp. 16–23. 7

[FPM09] FILLARD P., POUPON C., MANGIN J.-F.: A novel global tractography algorithm based on an adaptive spin glass model. In *MICCAI (1)* (2009), pp. 927–934. 1

[JBBS02] JENKINSON M., BANNISTER P., BRADY M., SMITH S.: Improved optimization for the robust and accurate linear registration and motion correction of brain images. *NeuroImage 17*, 2 (2002), 825–841. 6

[JJB11] JBABDI S., JOHANSEN-BERG H.: Tractography: Where Do We Go from Here? *Brain Connectivity 1*, 3 (2011), 169–183. 1

[JLJ*11] JEURISSEN B., LEEMANS A., JONES D. K., TOURNIER J.-D., SIJBERS J.: Probabilistic fiber tracking using the residual bootstrap with constrained spherical deconvolution. *Human Brain Mapping 32* (2011), 461–479. 7

[JWAB07] JBABDI S., WOOLRICH M. W., ANDERSSON J. L. R., BEHRENS T. E. J.: A bayesian framework for global tractography. *NeuroImage 37*, 1 (2007), 116–129. 1

[KMK08] KREHER B. W., MADER I., KISELEV V. G.: Gibbs tracking: A novel approach for the reconstruction of neuronal pathways. *Magnetic Resonance in Medicine 60*, 4 (2008), 953–963. 1

[MCCVZ99] MORI S., CRAIN B. J., CHACKO V. P., VAN ZIJL P. C. M.: Three-dimensional tracking of axonal projections in the brain by magnetic resonance imaging. *Annals of Neurology 45*, 2 (1999), 265–269. 7

[NSR*12] NEHER P. F., STIELTJES B., REISERT M., REICHT I., MEINZER H.-P., FRITZSCHE K. H.: MITK global tractography. In *Medical Imaging 2012: Image Processing* (2012), Haynor D. R., Ourselin S., (Eds.), SPIE, SPIE. 3, 4, 6

[Ope] www.openwalnut.org. visited 11th June 2013. 6

[RMA*11] REISERT M., MADER I., ANASTASOPOULOS C., WEIGEL M., SCHNELL S., KISELEV V.: Global fiber reconstruction becomes practical. *NeuroImage 54*, 2 (2011), 955 – 962. 1, 2

[RMK09] REISERT M., MADER I., KISELEV V.: Kiselev v. global reconstruction of neuronal fibres. In *Proc. of MICCAI, Diffusion Modelling Workshop. 2009; Notes in Computer Science 3* (2009). 1, 2, 4, 6, 7

[ZGHG07] ZHANG F., GOODLETT C., HANCOCK E. R., GERIG G.: Probabilistic fiber tracking using particle filtering and von Mises-Fisher sampling. In *EMMCVPR* (2007), Yuille A. L., Zhu S. C., Cremers D., Wang Y., (Eds.), vol. 4679 of *Lecture Notes in Computer Science*, Springer, pp. 303–317. 1