Segmentation of Neuron Bundles from Diffusion MRI

SLT Course Project 2016

Nico Previtali
Marko Pichler Trauber
Jakob Jakob

30. May 2016

Overview

Overview

- Model Extensions
- Optimization
- Implementation
- Results

Model Extensions I

Neighbourhood Definition: No fixed k-nearest neighbourhood $\Longrightarrow$ automatically select *suitable* neighbours:

1. Search for k^*-nearest neighbours of voxel v_i
2. Select the k neighbours ($k^* > k$) with the most similar diffusion profile w.r.t to voxel v_i

We used $k^* = 26$ and $k = 6$ which empirically turned out to be suitable.

Model Extensions II

Similarity Matrix: *Pairwise diffusion profiles similarities* instead of *pairwise voxel distances*

$$D_{ij} = \begin{cases} \|s_i - s_j\| & \text{if } v_i \text{ and } v_j \text{ are mutual neighbours} \\ 0 & \text{otherwise} \end{cases}$$

Use of Inner Products: Further we looked at inner products $\langle s_i, s_j \rangle$ between the diffusion profiles $\Longrightarrow$ allows to extend the model by applying a kernel.

$$D_{ij} = \begin{cases} \langle s_i, s_j \rangle & \text{if } v_i \text{ and } v_j \text{ are mutual neighbours} \\ 0 & \text{otherwise} \end{cases}$$

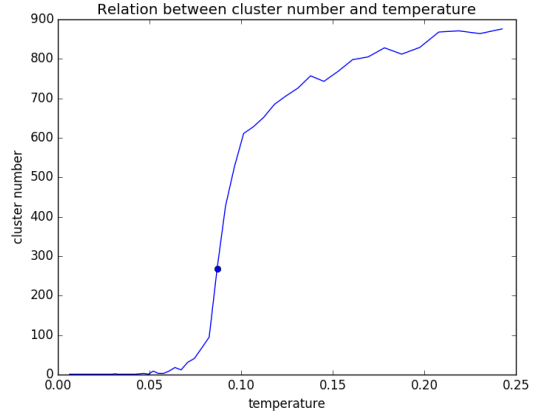
Optimization

Example: If you used Metropolis, mention it and maybe comment on your proposal distribution, but do not explain how it works in general. The same for deterministic annealing, etc. For any optimization strategy not discussed during the course, elaborate more.

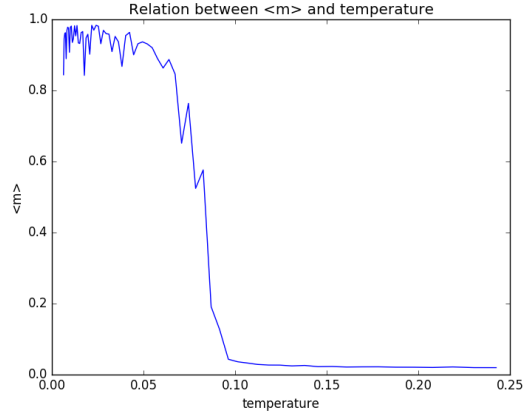
Implementation

State the software that you used. Did you use CPU, GPU, cluster, etc? Did you try to write efficient code? What performance differences did you observe? What were the most useful tricks that you applied? What else did you do to decrease runtime and memory usage?

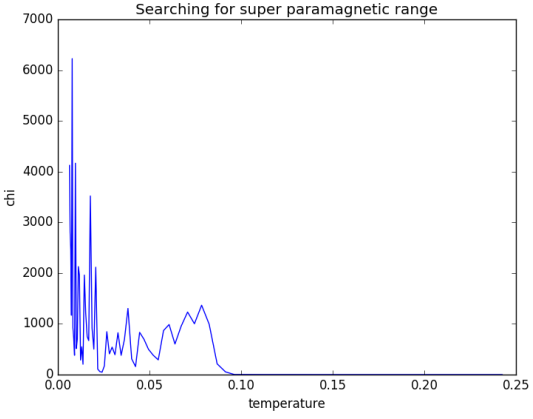
Implementation



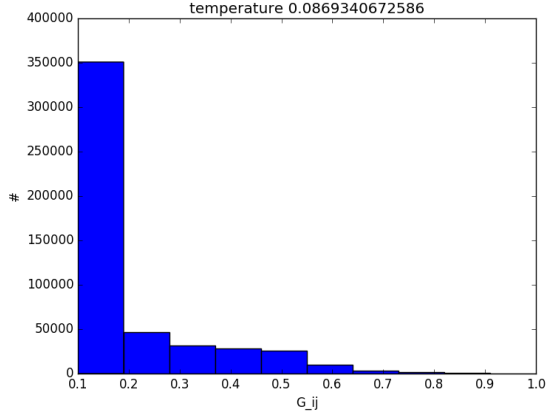
(a)



(b)

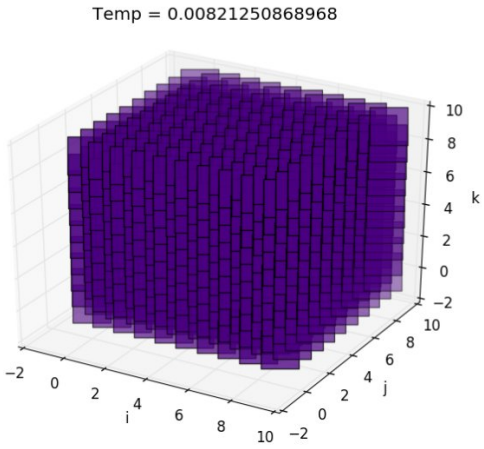


(c)

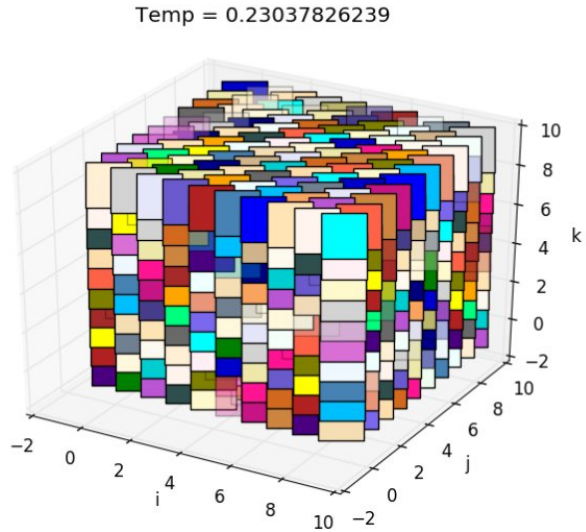


(d)

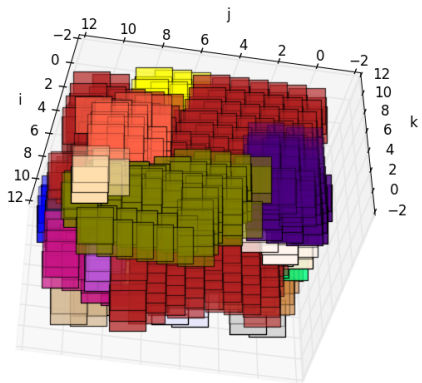
Results



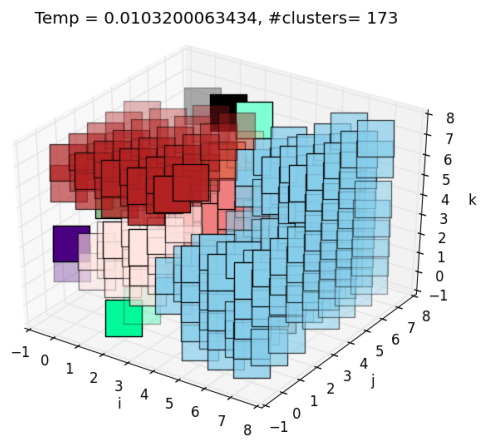
(a)



(b)



(c)



(d)

References