Numerical Methods for Image Segmentation

Philipp Lösel

Engineering Mathematics and Computing Lab
Interdisciplinary Center for Scientific Computing
Heidelberg University

19th of September, 2016

Numerical Methods for Image Segmentation

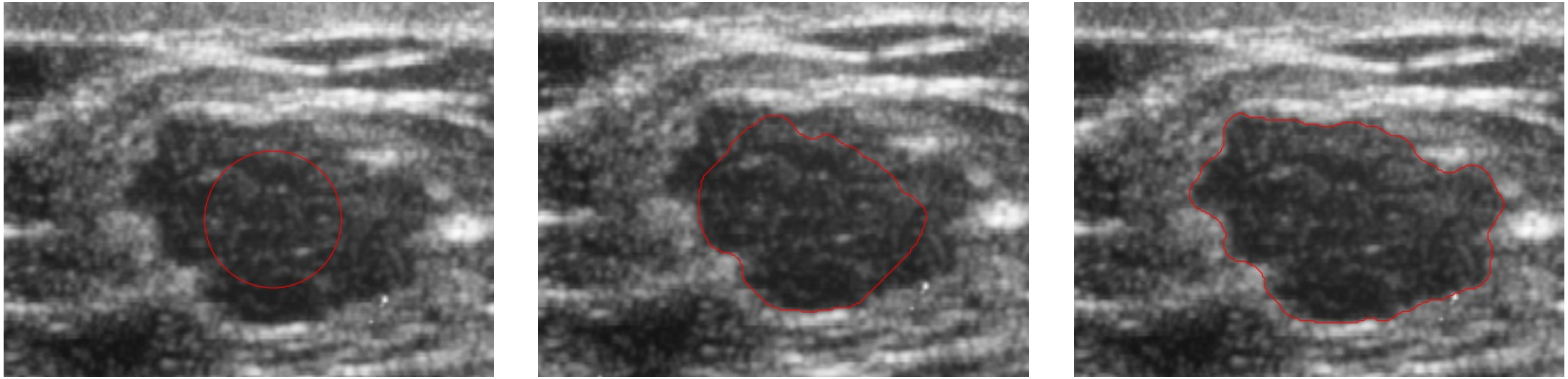
Outline

1. Active Contours Without Edges
2. GPU based diffusion for image segmentation
3. Results
4. Practical Session
5. Uncertainty Quantification
6. Inverse Image Segmentation
7. Summary

Active Contours Without Edges

Let the curve C in Ω be the boundary of an open subset $\omega \subset \Omega$, i.e. $C = \partial\omega$, $\omega = inside(C)$ and $\Omega \backslash \overline{\omega} = outside(C)$. Let $I(x, y)$ be the Image Data, $c_1 = average(inside)$ and $c_2 = average(outside)$, then we determine C so that it minimizes the functional

$$\begin{aligned} F(c_1, c_2, C) = {} & \alpha \cdot \text{length}(C) + \beta \cdot \text{Area}(inside(C)) \\ & + \lambda_1 \int_\omega |I(x, y) - c_1|^2 \, \mathrm{d}x\mathrm{d}y \\ & + \lambda_2 \int_{\Omega \backslash \overline{\omega}} |I(x, y) - c_2|^2 \, \mathrm{d}x\mathrm{d}y. \end{aligned}$$



Active Contours Without Edges

Level Set Method: C is represented by the zero level set of a Lipschitz function $\phi : \Omega \to \mathbb{R}$, such that

$$\begin{aligned} C = \partial\omega &= \{(x, y) \in \Omega : \phi(x, y) = 0\} \\ inside(C) &= \{(x, y) : \phi(x, y) > 0\} \\ outside(C) &= \{(x, y) : \phi(x, y) < 0\}. \end{aligned}$$

A curve that minimizes the Functional has to satisfy the Euler Lagrange equation

$$|\nabla\phi| \left[\alpha \operatorname{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right) - \beta - \lambda_1(u_0 - c_1)^2 + \lambda_2(u_0 - c_2)^2\right] = 0.$$

Introducing an artificial time $t \geq 0$, we have to solve

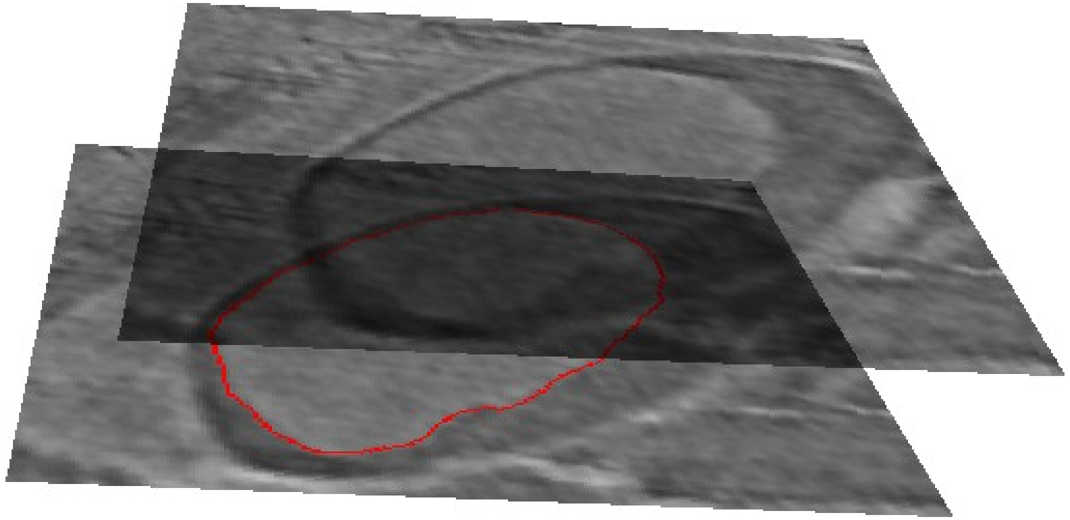
$$\frac{\partial\phi}{\partial t} = |\nabla\phi| \left[\alpha \operatorname{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right) - \beta - \lambda_1(u_0 - c_1)^2 + \lambda_2(u_0 - c_2)^2\right].$$

Active Contours Without Edges

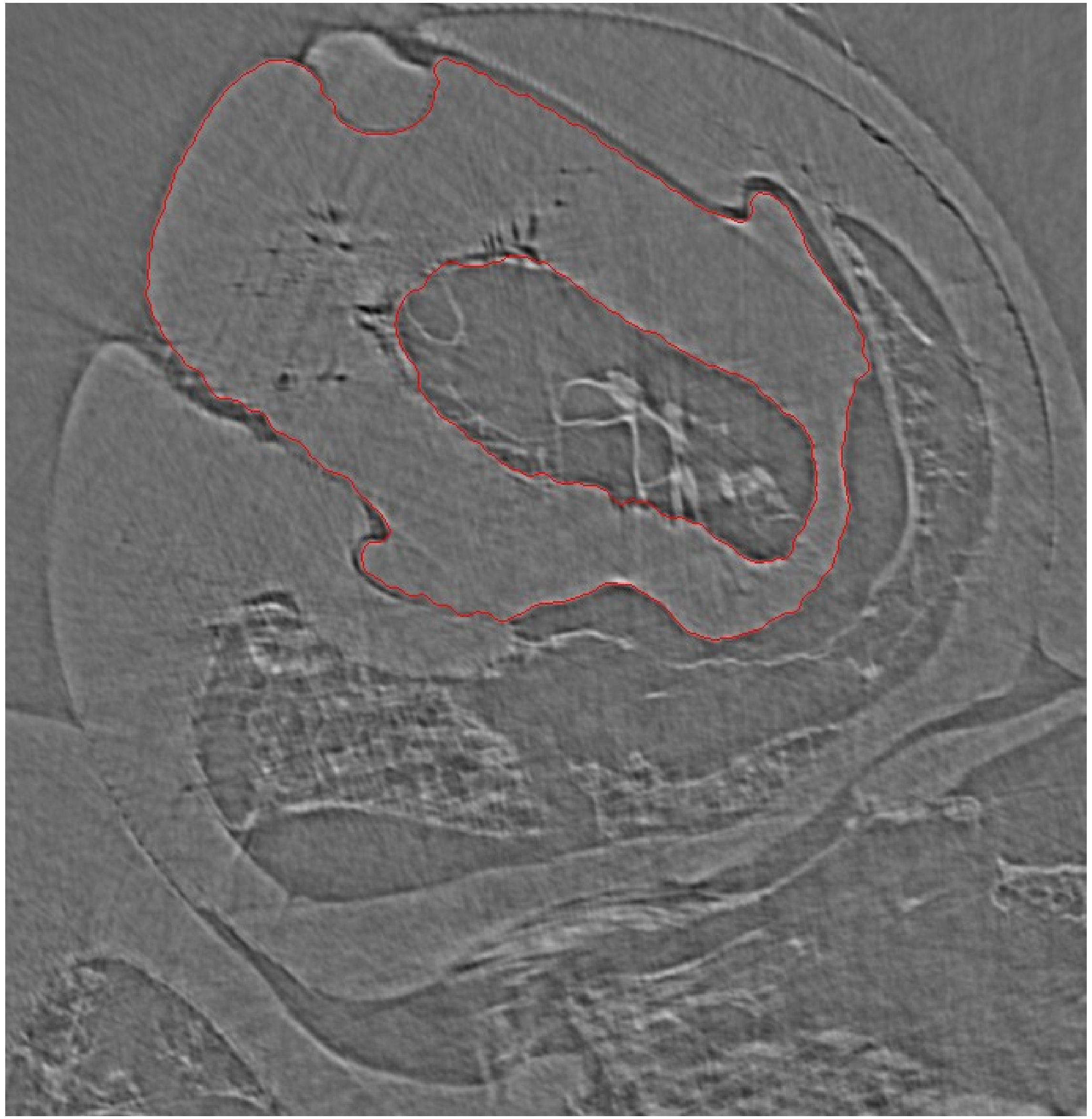
Practical session

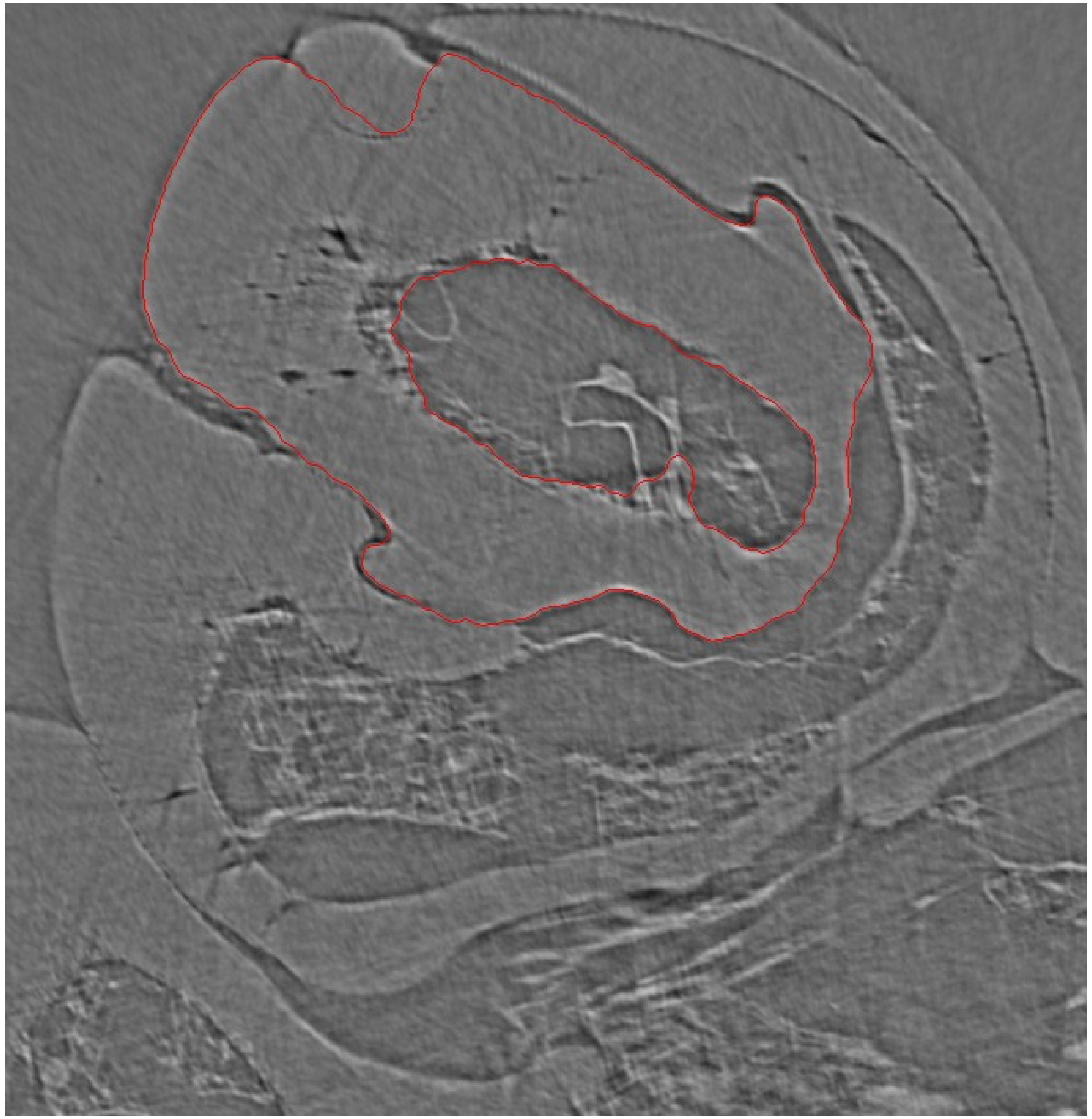
Active Contours Without Edges

- How to segment 3D data?
- Active contours slice by slice
- Use prior segmentation as initialization and move forward in z-direction



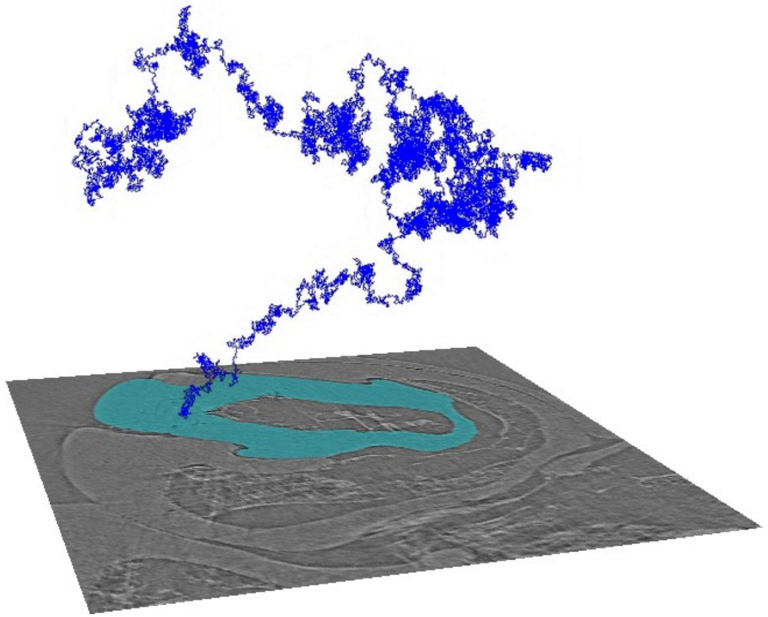
Active Contours Without Edges

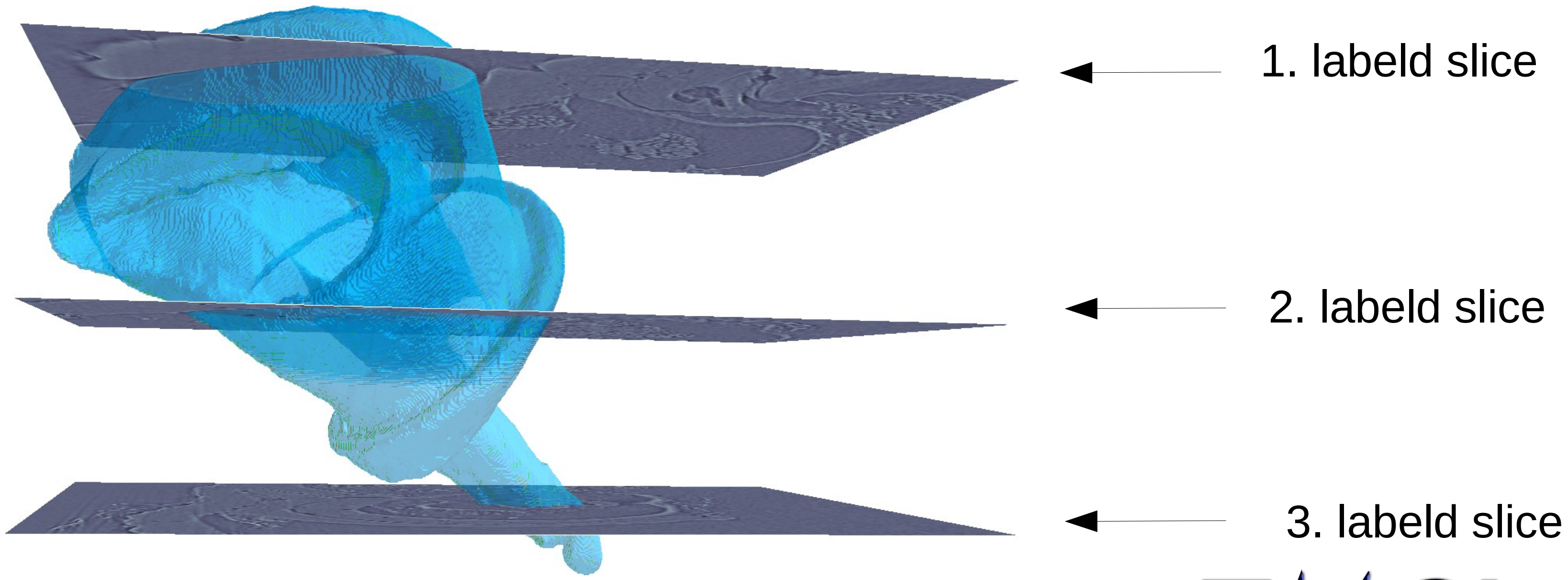




Multi-GPU based, parmeter-free diffusion method

- Segmentation by means of weighted random walks
- Several weighted random walks start in manually labeled slices
- An unlabeled voxel will be hit by random walks over time
- A voxel can be hit by random walks starting in the background or the labels
- Image Segmentation by assigning each voxel to the segment where the most random walks started from





Multi-GPU based, parmeter-free diffusion method

Let V be a set of vertices in $\mathbb{R}^n$. For $x, y \in V$ we write $x \sim y$ if x is adjacent to y. A weighted graph is a couple (V, P) where P is a non-negatove function on $V \times V$. Let $I(x)$ be the image data. Further, let x_0 be the starting point of a random walk and σ_{x_0} the standard deviation from x_0 in a local area, then we set

$$\mu_{x_0}(y) = \frac{1}{\sqrt{2\pi\sigma_{x_0}^2}} \exp\left(-\frac{(I(x_0) - I(y))^2}{2\sigma_{x_0}^2}\right) \text{ for all } y \in V$$

and

$$\mu(x) = \sum_{y \in V, y \sim x} \mu_{x_0}(y) \text{ for all } x \in V.$$

Multi-GPU based, parmeter-free diffusion method

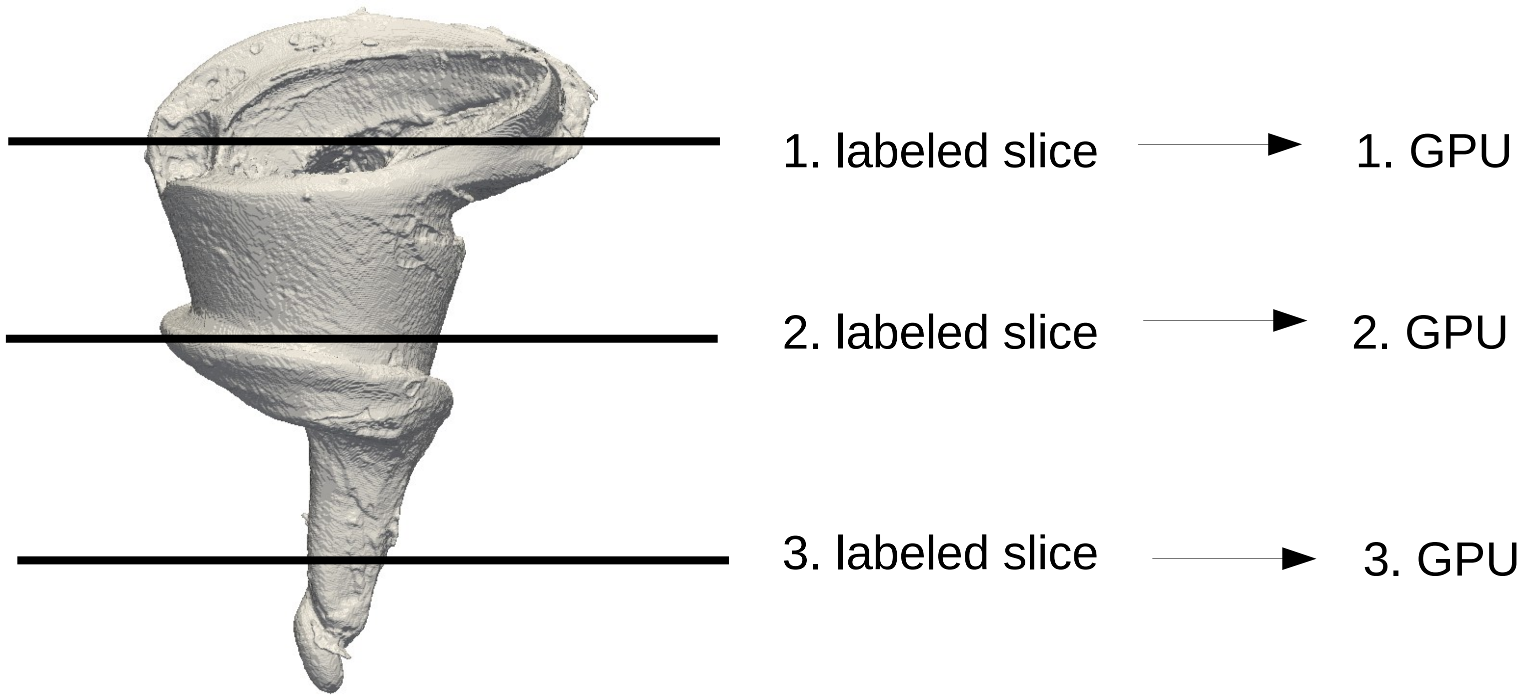
The function μ induces a Markov kernel

$$P_{x_0}(y, x) = \begin{cases} \frac{\mu_{x_0}(y)}{\mu(x)} & \text{if } y \sim x \\ 0 & \text{else,} \end{cases}$$

where $P_{x_0}(y, x)$ is the conditional probability of a random walk moving from x to y given its position x.

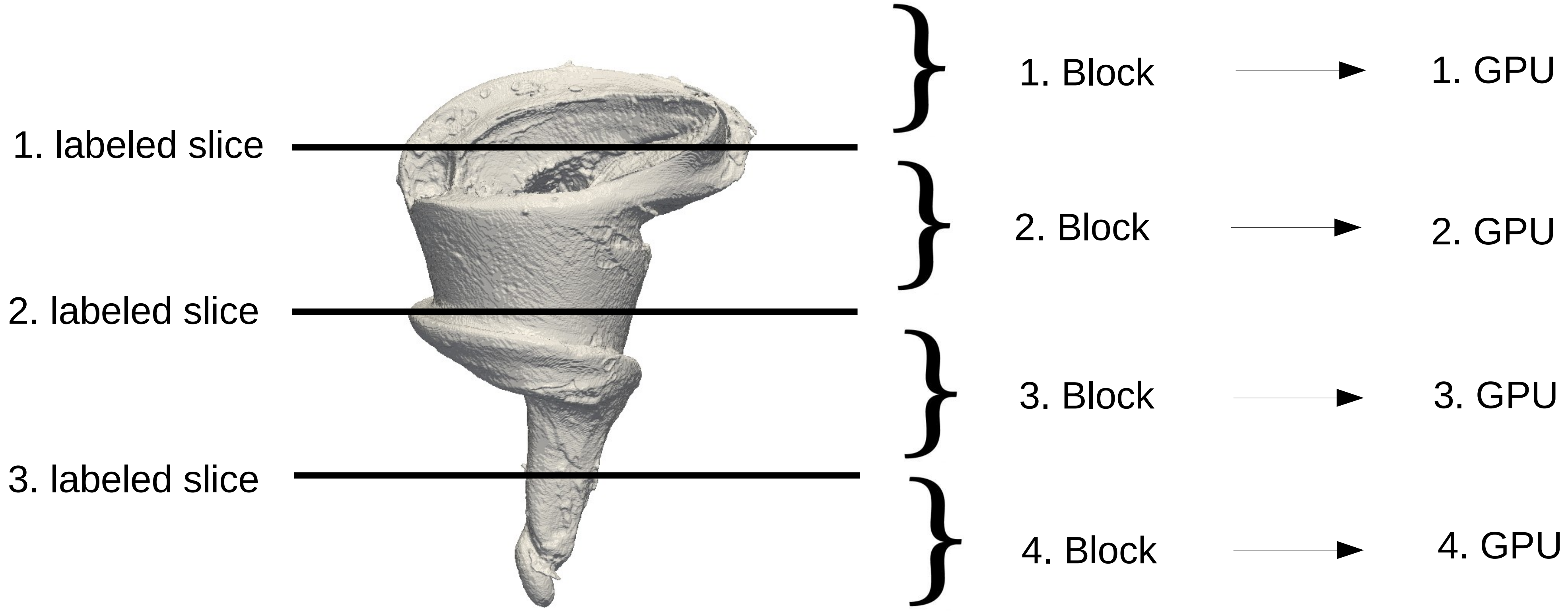
Multi-GPU diffusion (small images)

- Images < 125 MB
- Spread the labeled slices over several GPUs
- 1 GPU per labeled slice
- The GPU runs only the random walks belonging to this slice



Domain decomposition (large images)

- Split the image into blocks
- Each GPU runs all random walks for all labeled slices
- Each GPU considers only the hits in one block
- No storage of all hits necessary
- Argmax on the fly



Biomedical image segmentation app

A free online application for segmentation of tomographic images

Wasp Project – Thomas van de Kamp (ANKA @ KIT)

Practical session

Results

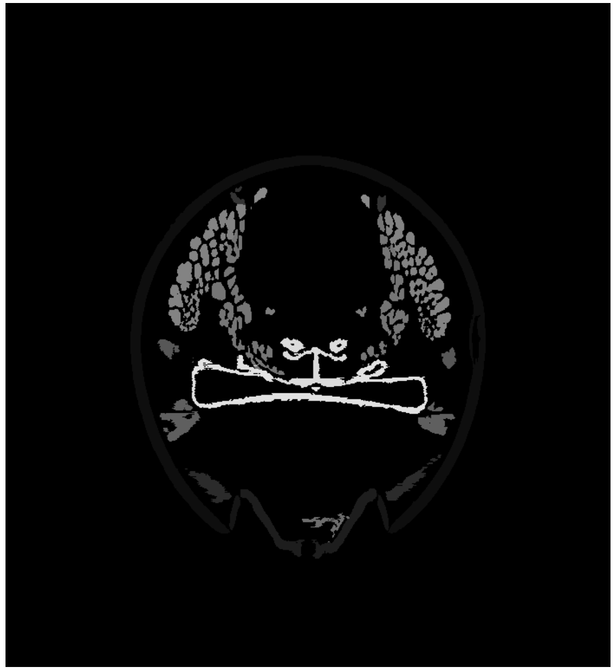
vs.

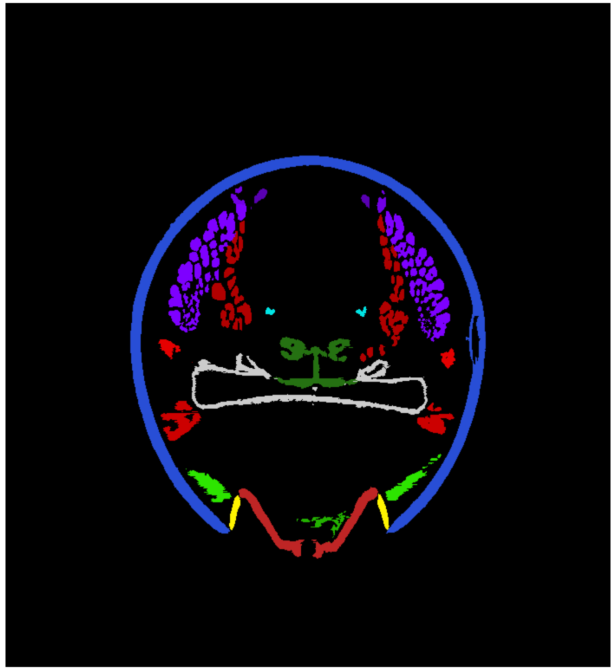
Biomedisa

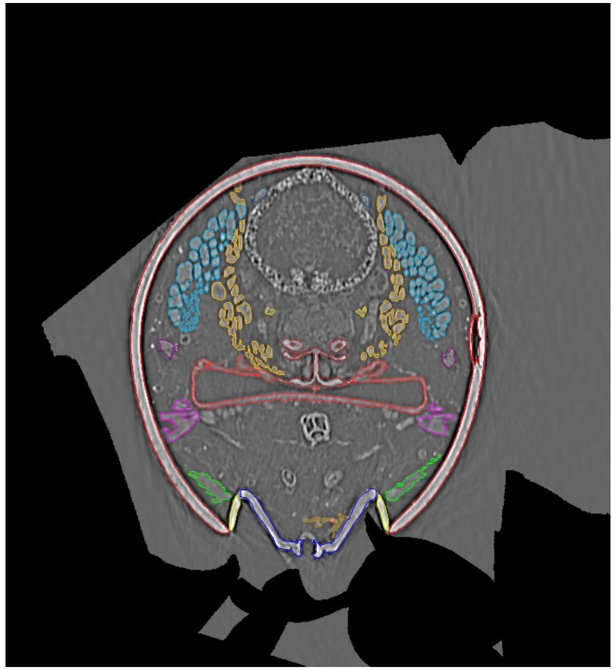
manual segmentation and
interpolation with Amira

Results

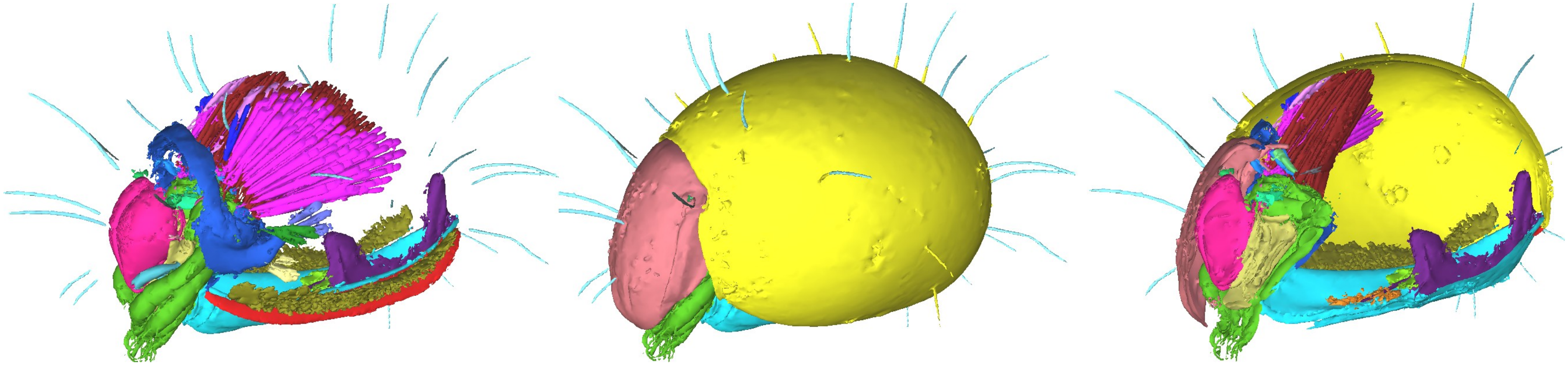
- Any number of labels
- Gray values or colors (ImageJ)







Results



Euphthiracarus reticulatus - Sebastian Schmelzle (TU Darmstadt)

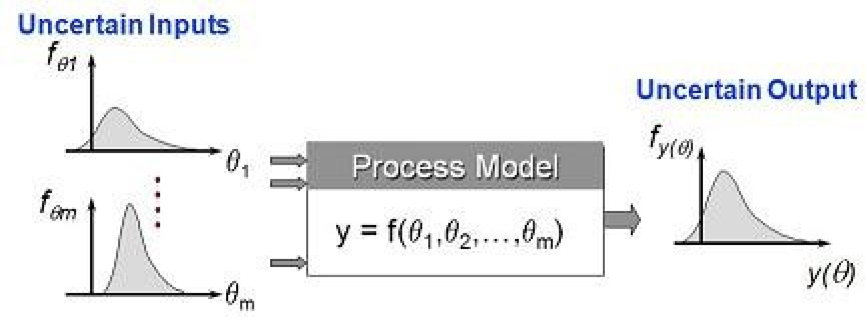
Results (7 GPUs)

Image	Size	No. of labeled slices	No. of segments	time	Dice coefficient
Wasp female	2553 x 992 x 1077	224	56	4 h 21 min	
Wasp male	3311 x 1223 x 1267	299	40	6 h 4 min	
Eucrib	1088 x 766 x 698	109	33	1 h 7 min	0.926
Eucrib	1088 x 766 x 698	55	33	35 min 0 sec	0.924
Eucrib	1088 x 766 x 698	22	33	27 min 55 sec	0.915
Trigonopterus	479 x 482 x 440	21	2	3 min 17 sec	
Hair	348 x 477 x 412	29	2	3 min 51 sec	
Brain Tumor	240 x 240 x 155	1	1	8 sec	

Uncertainty Quantification

„It tries to determine how likely certain outcomes are if some aspects of the system are not exactly known.“ (wiki)

- Uncertainty of parameters, boundary conditions, initialization
- None-intrusive (Monte Carlo) and intrusive methods (Polynomial Chaos Expansion)



Active contours (deterministic)

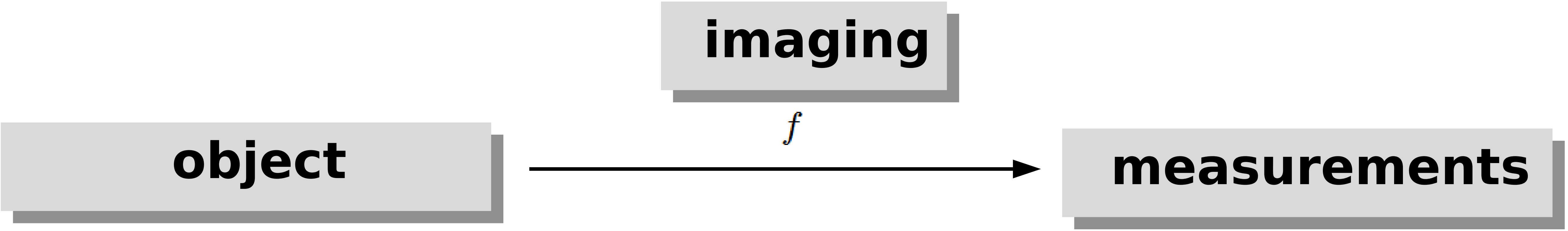
$$\frac{\partial \phi(x)}{\partial t} = |\nabla \phi(x)| \left(-\lambda_1 (u_0 - c_1)^2 + \lambda_2 (u_0 - c_2)^2\right)$$

Active contours (uncertain weights)

$$\frac{\partial \phi(x,\theta)}{\partial t} = |\nabla \phi(x,\theta)| \left(-\lambda_1(\theta) (u_0 - c_1)^2 + \lambda_2(\theta) (u_0 - c_2)^2\right)$$

Inverse Image Segmentation

Forward:



Inverse Image Segmentation

Which segmentation satisfies the measured data?

model

$$\textbf{segmentation} \quad \underset{f^{-1}}{\overset{f}{\rightleftarrows}} \quad \textbf{output}$$

Summary

1. Active Contours for image segmentation
2. Parameter-free, semi-automatic diffusion algorithm
3. Uncertainty Quantification
4. Inverse Segmentation

Thank you for your attention!