Computer- and robot-assisted Surgery

NCT

NATIONALES CENTRUM
FÜR TUMORERKRANKUNGEN
PARTNERSTANDORT DRESDEN
UNIVERSITÄTS KREBSCENTRUM UCC

getragen von:
Deutsches Krebsforschungszentrum
Universitätsklinikum Carl Gustav Carus Dresden
Medizinische Fakultät Carl Gustav Carus, TU Dresden
Helmholtz-Zentrum Dresden-Rossendorf

Lecture 4

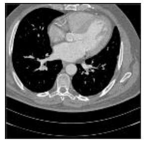
Segmentation 1

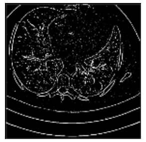
Introduction

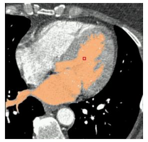
Quelle: RadiologySchools

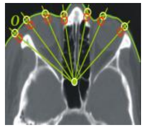
Process chain computer-assisted surgery

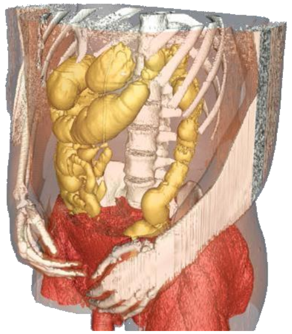
Image acquisition → Pre-processing → Segmentation → Modeling → Registration → Visualization

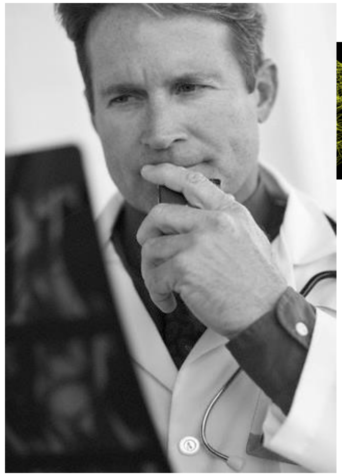


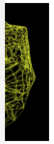












Content

How do you group single pixels that belong to one object?

- What does segmentation mean?

- Methods are based on:
  - Color/Grey value
  - Form/Contour
  - …

Introduction

What does segmentation mean?

- Assigning a label to each pixel, pixels with the same label belong to the same object

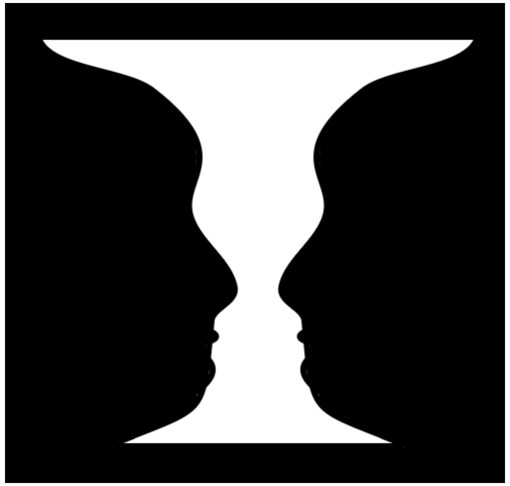
- Partitioning the image in multiple semantic segments
    - Complete: every pixel is assigned to a segment
    - No overlap: a pixel is part of only one segment
    - Connected: every segment is a connected area

Introduction

Goal of segmentation

- Identification and Characterization of Tissue
- Semantic Information
  - Object (Vessels, Organs, Tumors,…)
- Quantitative Analysis
  - Volume
  - Form
- Detection of boundaries
  (e.g.: Tumor yes/no)

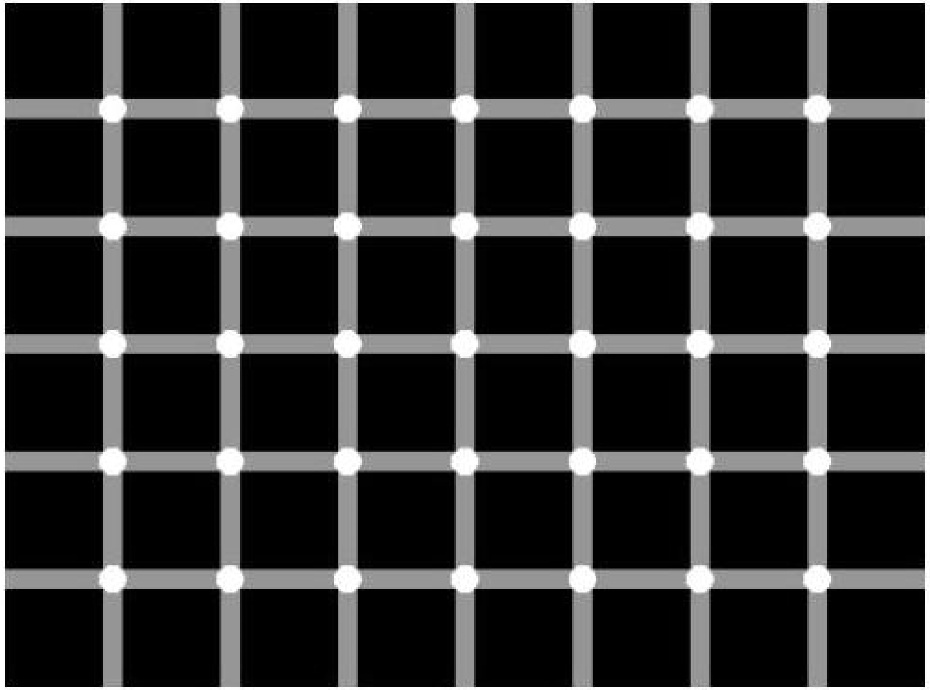
Introduction



Human Perception

⇔

Machine Perception



Important Criteria

- Accuracy
  - Comparison with ground truth

- Performance
  - Real-time, Seconds, Minutes, Hours…

- Level of Interaction
  - User effort

- Robustness
  - Patient movement, Reconstruction artefacts…

Classification

- Point-based
  - Operations only on grey values
  - No global considerations
  - e.g.: Threshold methods

- Region-based
  - Every area of a region fulfills a certain homogeneity critera
  - e.g.: region growing

Classification

- Edge- and contour-based
  - Object has a clear edge
  - Goal: extraction und merging of edges
  - e.g.: Active Contours, Snakes

- Knowledge-/ model-based
  - Integration of problem specific a-priori-knowledge
  - Goal: Enhancement of segmentation by only considering „plausible“ results
  - z.B.: Point Distribution Models…

POINT-BASED SEGMENTATION

Threshold operations

- Point-based (no global features)
- Generation of a binary image

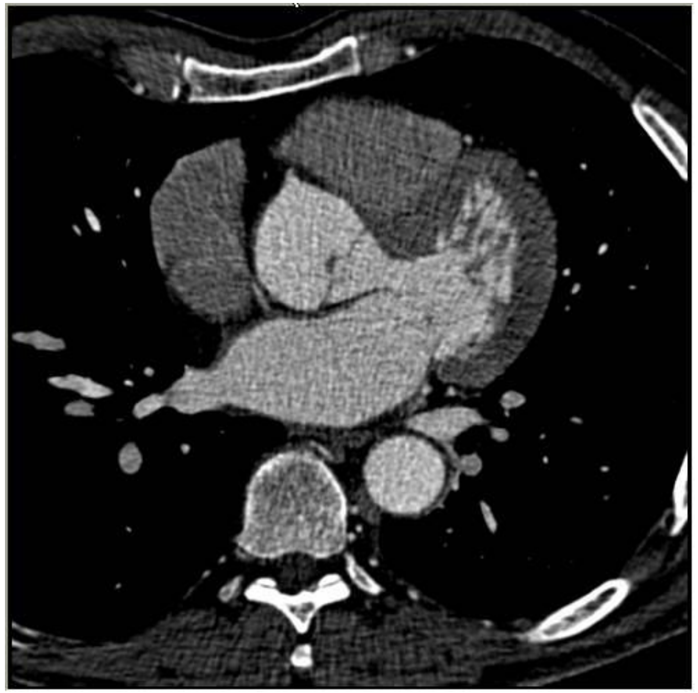
- Operator with threshold Θ

$$f \rightarrow h$$

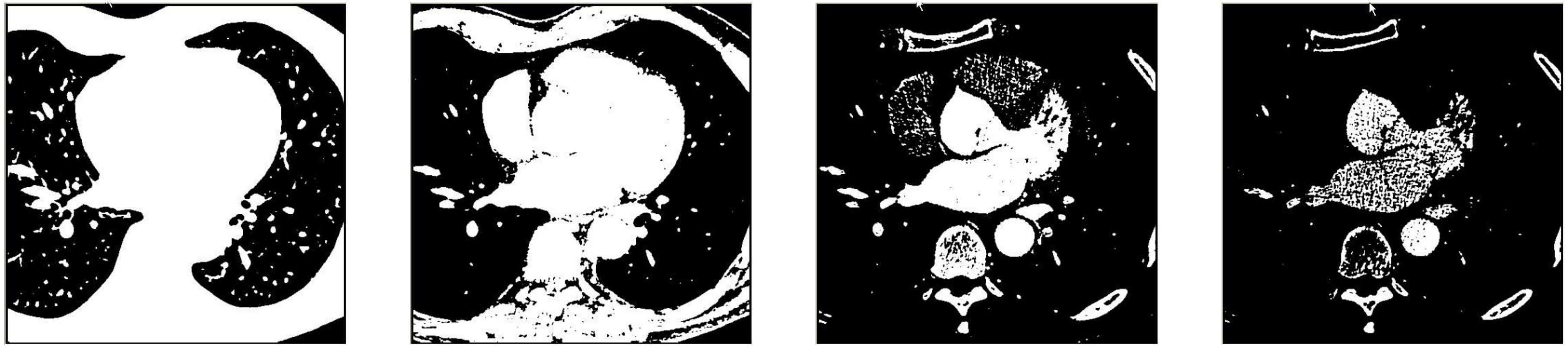
$$h_{jk} = \begin{cases} 1 \ \text{für} \ f_{jk} > \Theta \\ 0 \, \text{sonst} \end{cases}$$

$$h_{jk} = \begin{cases} 1 \ \text{für} \, \Theta_{\min} < f_{jk} < \Theta_{\max} \\ 0 \, \text{sonst} \end{cases}$$

Threshold

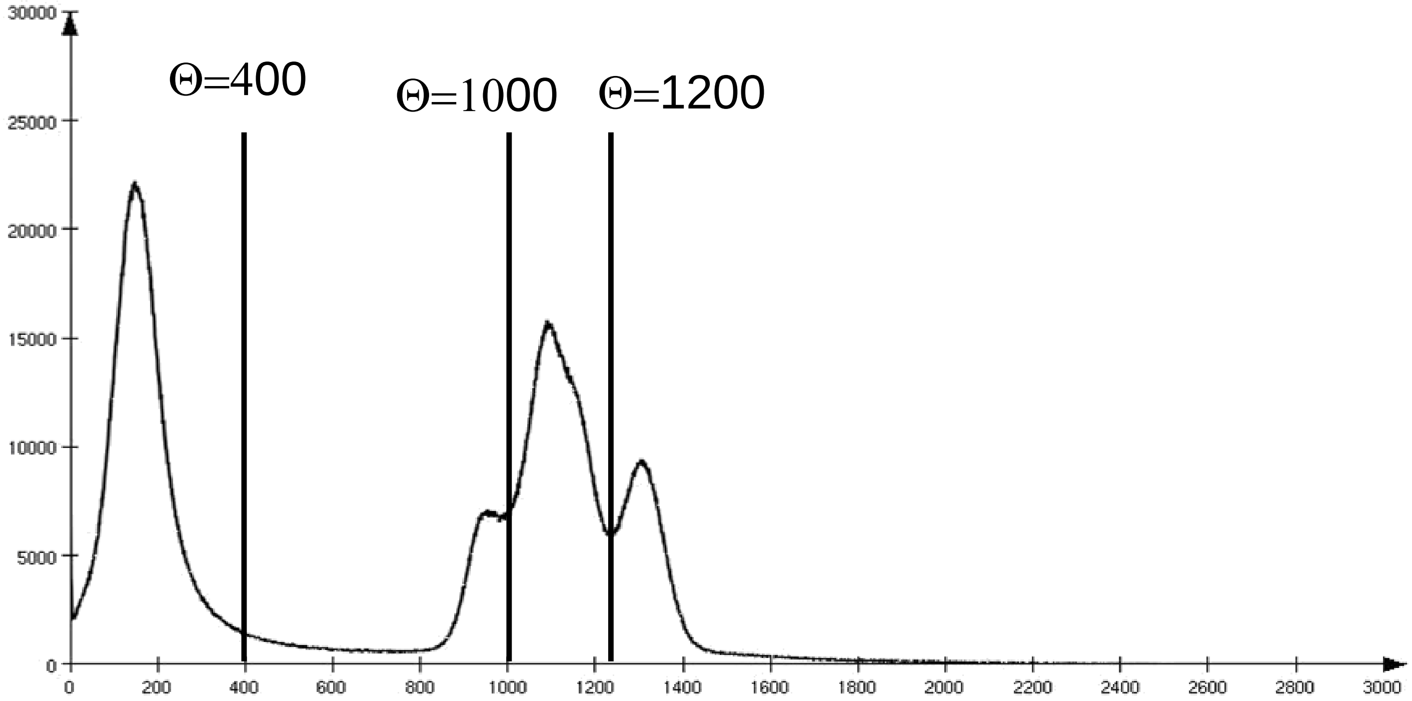


$\Theta=400$ $\Theta=1000$ $\Theta=1200$ $\Theta=1300$



Threshold

- Determination of the threshold
  - Manually, experience-based
  - Consideration of the histogram

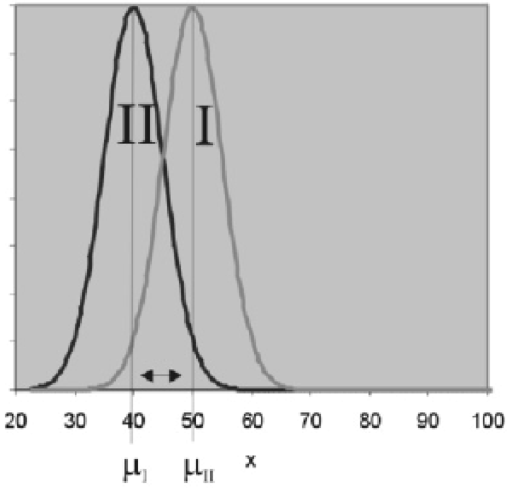


Multilevel Otsu Method

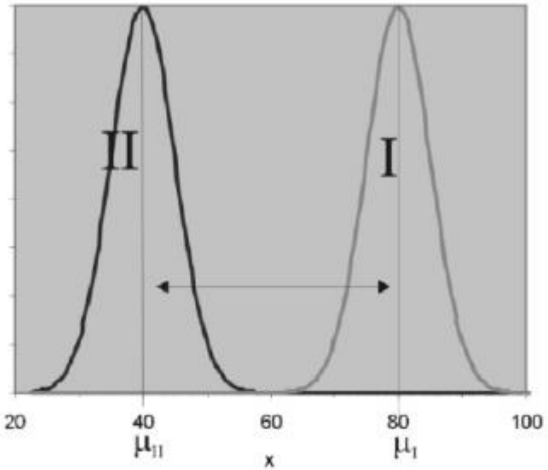
- Automatic multimodal threshold approach
- Optimal thresholds are calculated based on the grey value distribution
- Number of k thresholds means distribution in k+1 classes
- Partition is based on discrimination criteria

Multilevel Otsu Method

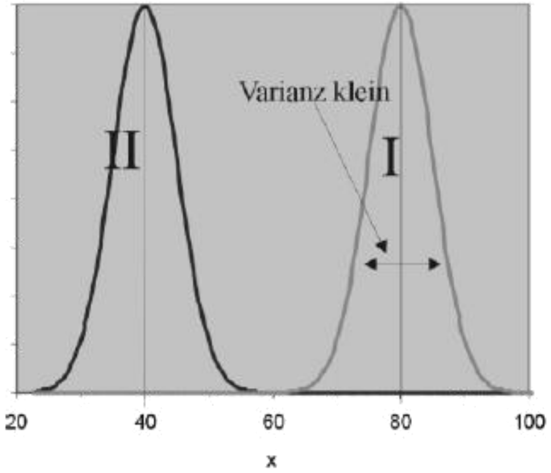
- Inter-class variance σ^2_b : Measure of variance between classes
- Intra-class variance σ^2_w : Measure of variance in class
- Goal: minimizing intra-class variance and maximizing inter-class variance (for partitioning the image)



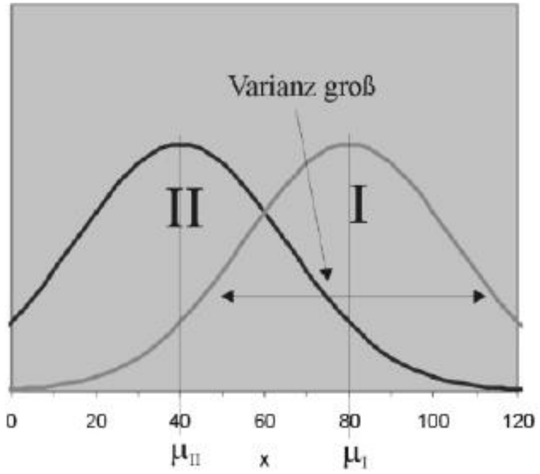
Small Inter-class variance



Big Inter-class variance



Small Intra-class variance



Big Intra-class variance

Quelle: Wikipedia

Quelle: Wikipedia

Multilevel Otsu Method

- Discrimination criteria:

$$\lambda(t_1, \ldots, t_k) = \frac{\sigma_b^2(t_1,\ldots,t_k)}{\sigma_w^2(t_1,\ldots,t_k)}$$

- Norm:

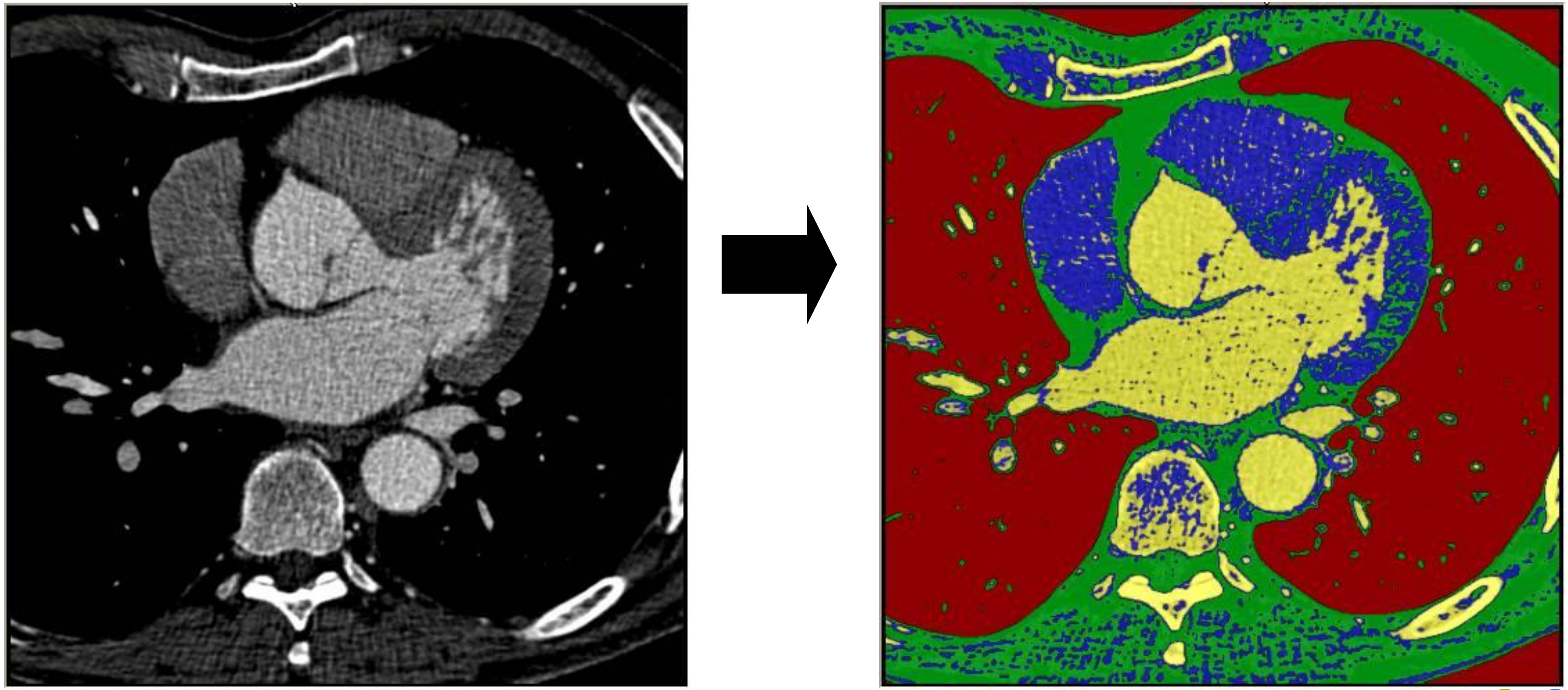
$$\eta(t_1, \ldots, t_k) = \frac{\sigma_b^2(t_1,\ldots,t_k)}{\sigma^2}$$

- Because σ^2 is independent of the threshold maximizing σ^2_b is sufficient

- Multilevel-Otsu Method calculates k optimal thresholds that maximize inter-class variance

Multilevel Otsu Method

- Result (4 classes)



REGION-BASED SEGMENTATION

Regions

- Goal
  - Extraction of connected areas
  - Areas are defined regarding specific homgeneity criteria

- Definition of a region:

Partitioning an image f in regions f_ν

$$\text{mit } f \rightarrow \{ f_\nu \mid \nu = 1, \ldots, N \} \text{ so dass}$$

$$\bigcup_{\nu=1}^{\mathrm{N}} f_\nu = f \text{ mit } f_\nu \cap f_\mu = 0 \qquad \mu \neq \nu$$

Homogeneity criteria

- In general simple grey value criteria $H(f_v)$

- Other criteria:
  - Defined threshold interval
  - Distance to average grey value of a region
  - Grey value distance to neighboring voxel
  - Texture

Regiongrowing

- Connected regions regarding $H(f_\nu)$
- Based on seeds
  - Manually defined from the user
  - Seeds are in a region
  - $H(f_\nu)$ can be defined through the neighbors of the seeds
  - Depending on the criteria the results depends on the positon of the seeds

Regiongrowing

Workflow

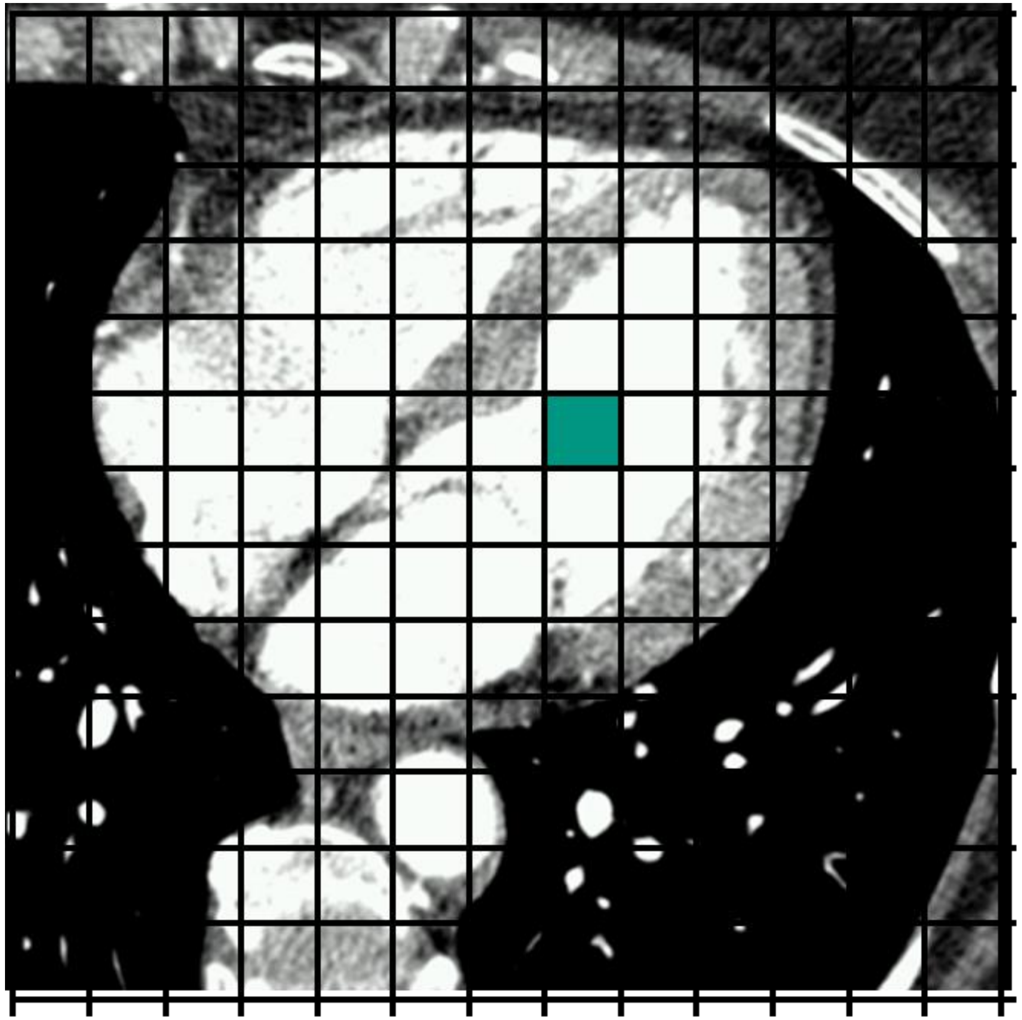
- Define seed as starting value

- Consider neighbors (not seen before)
  - Mark pixel as seen

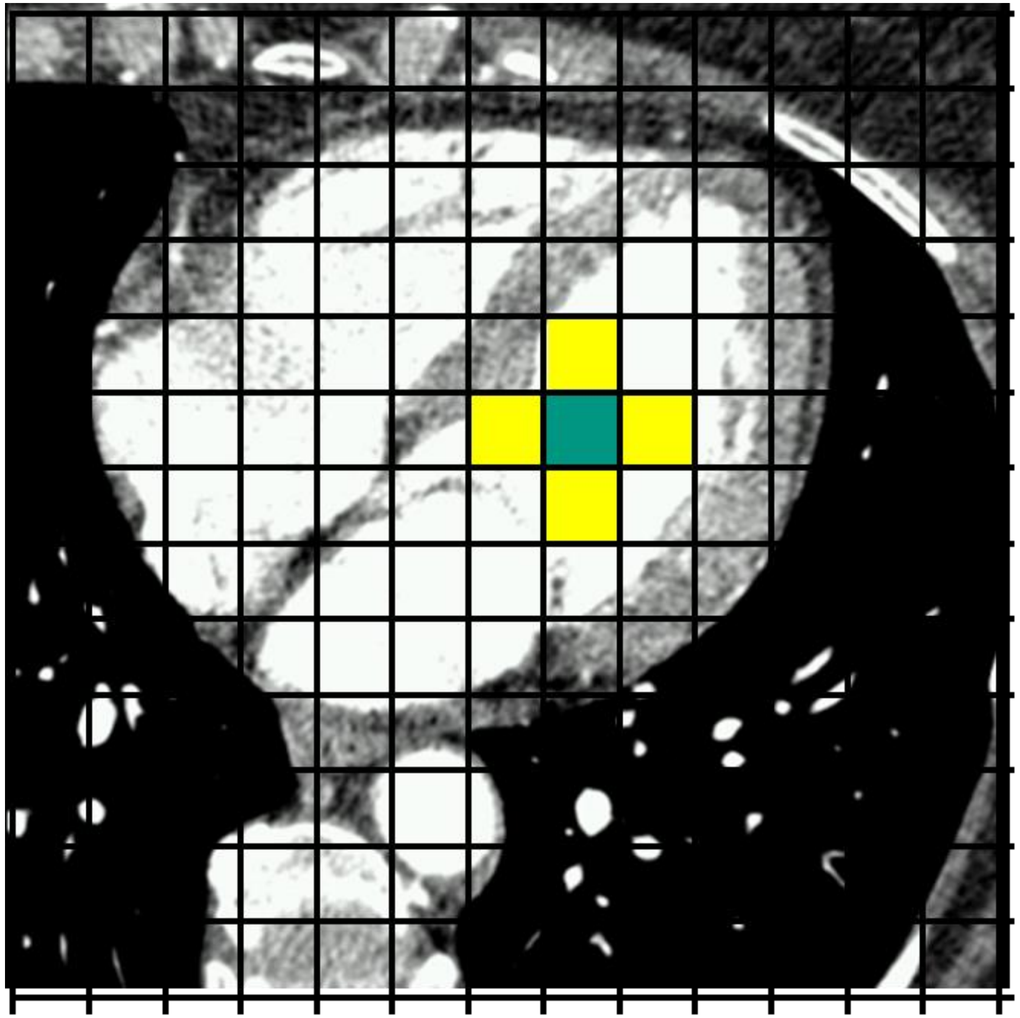
- If pixel fulfills criteria H
  - Yes: Use this pixel as new seed point
  - No: End

NCT

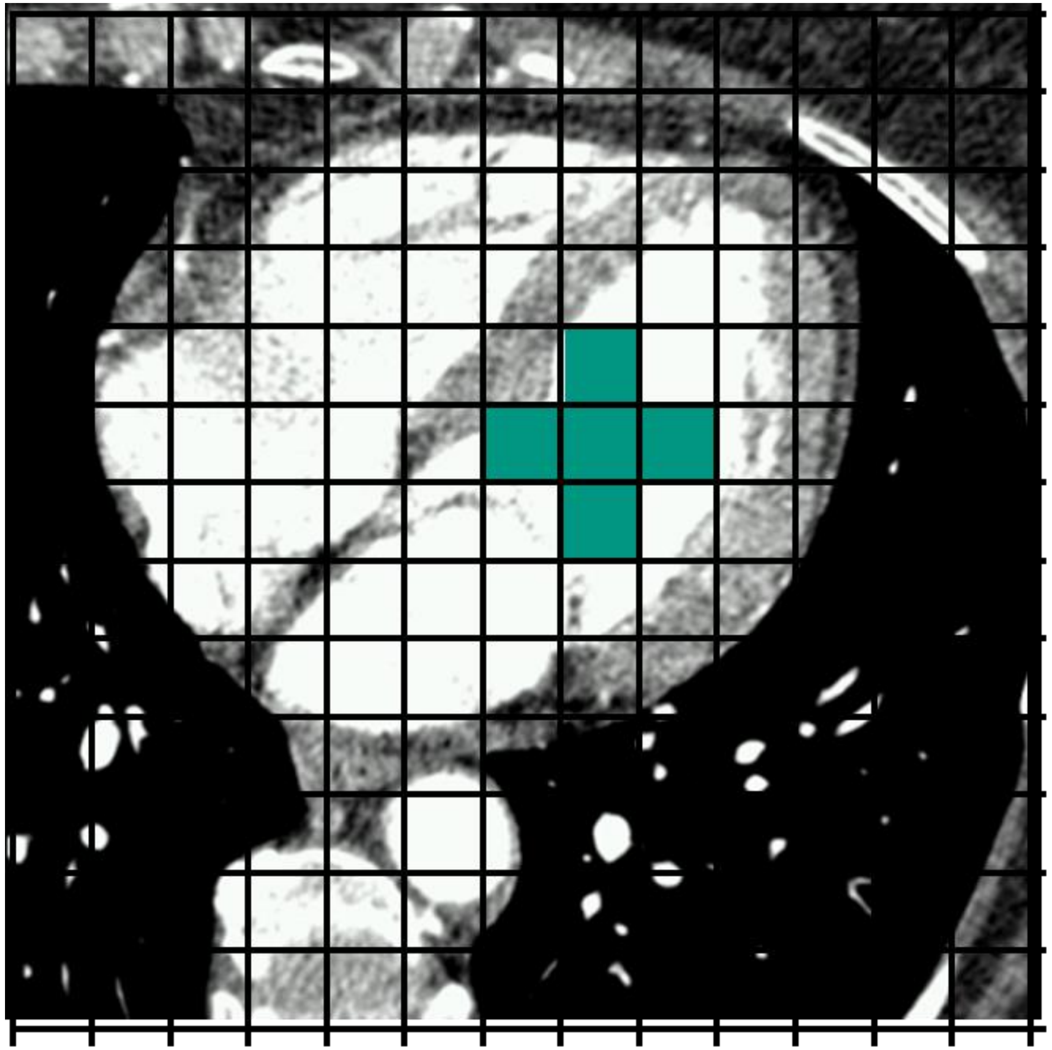
Regiongrowing



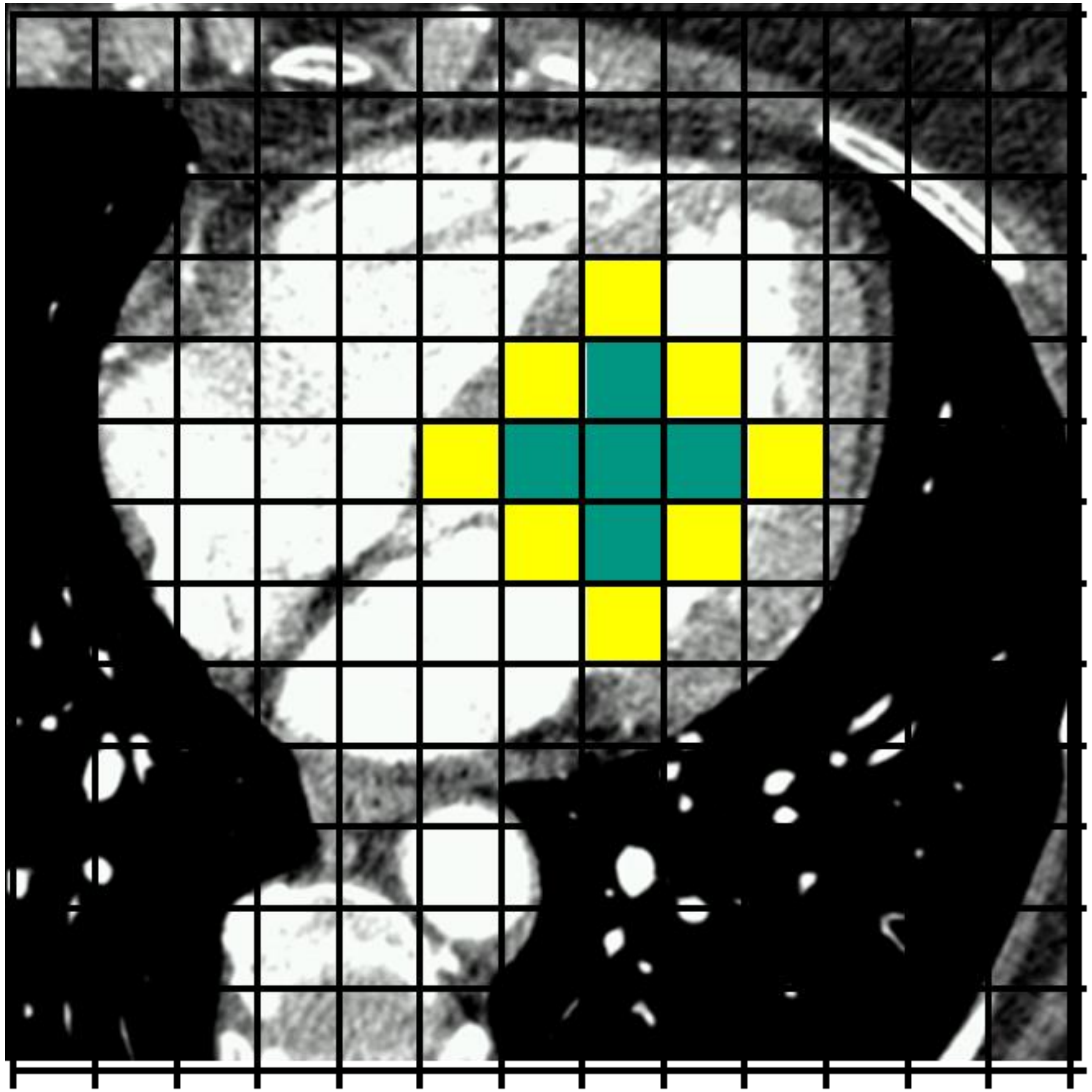
Regiongrowing



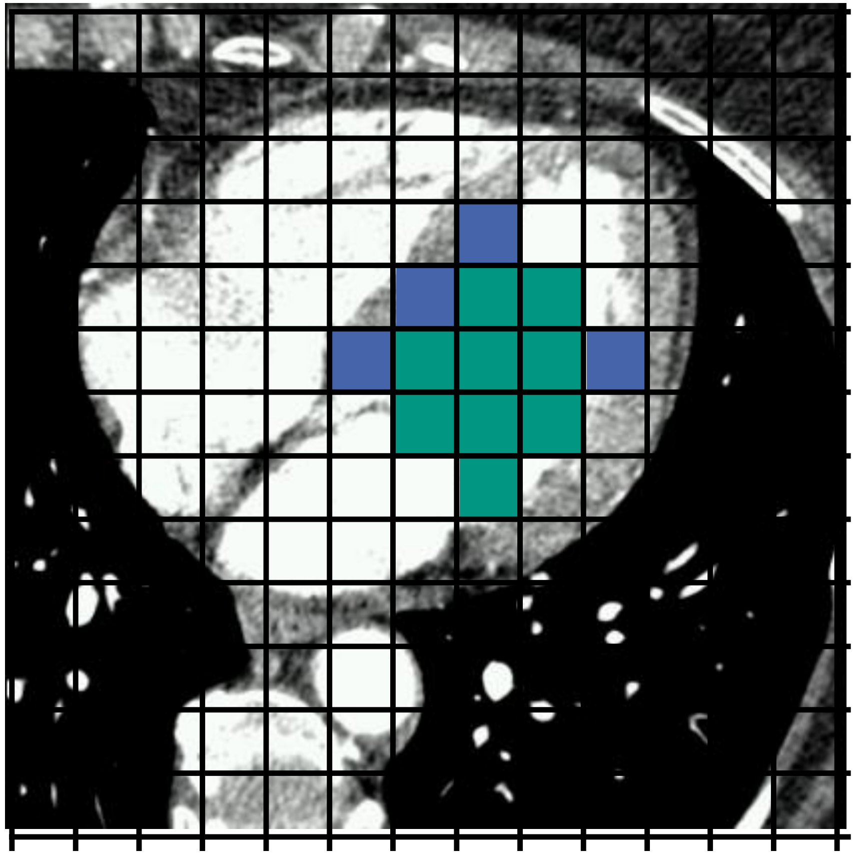
Regiongrowing



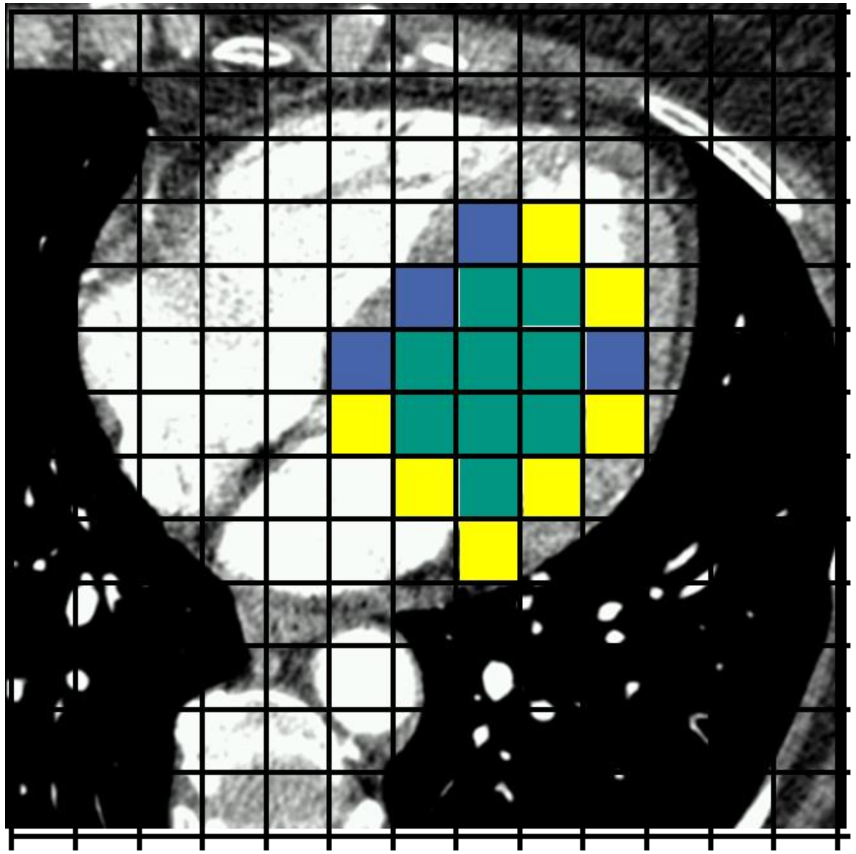
Regiongrowing



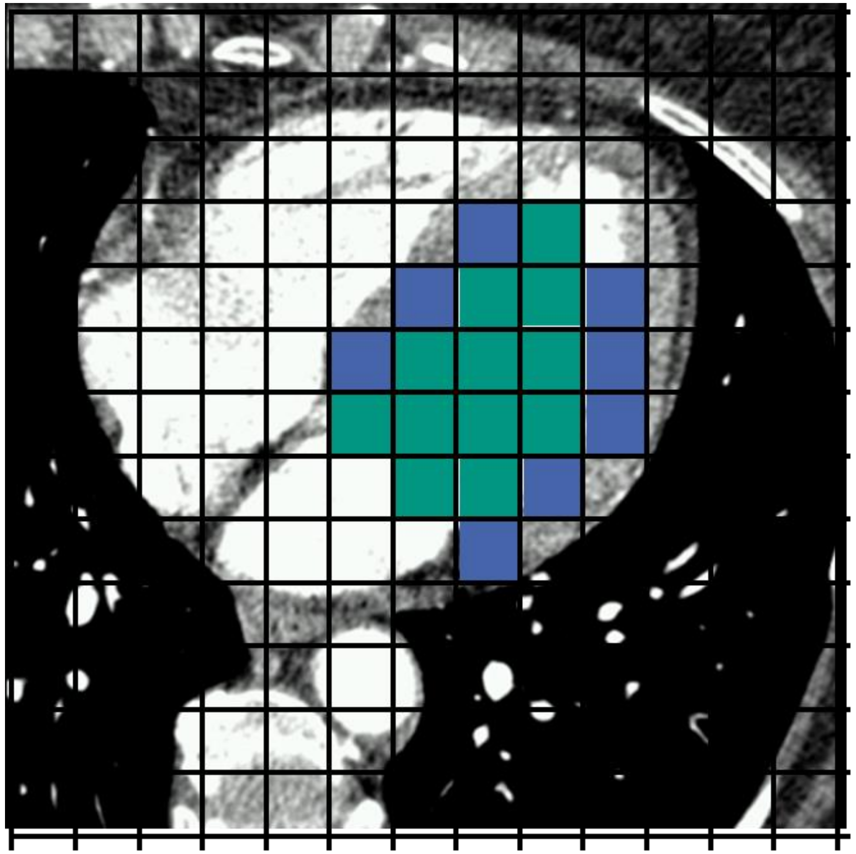
Regiongrowing



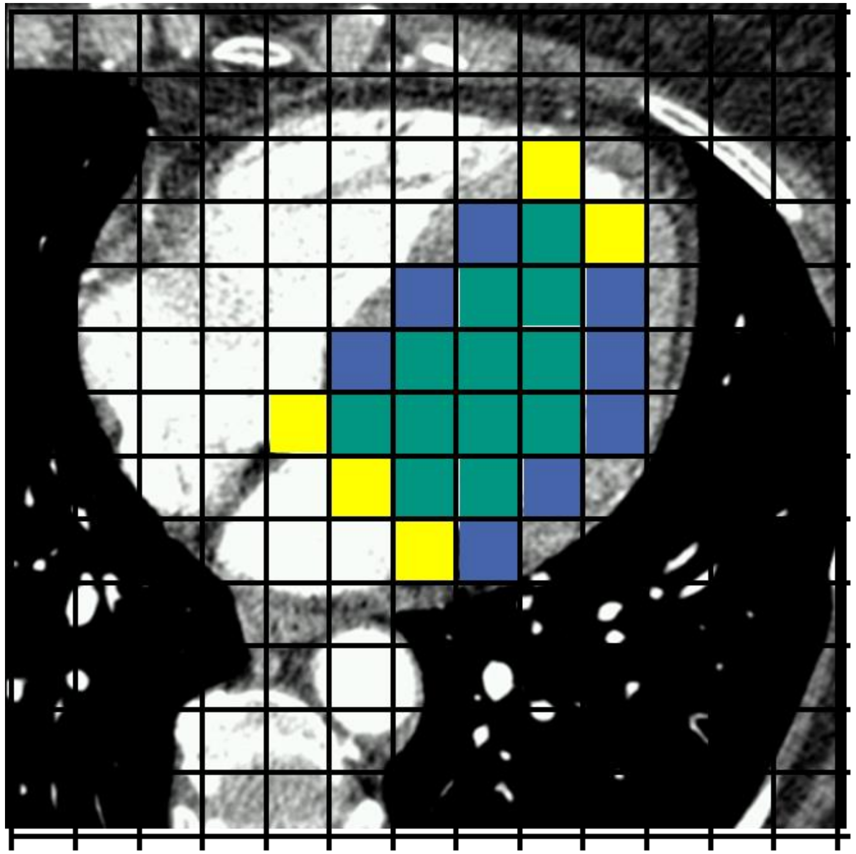
Regiongrowing



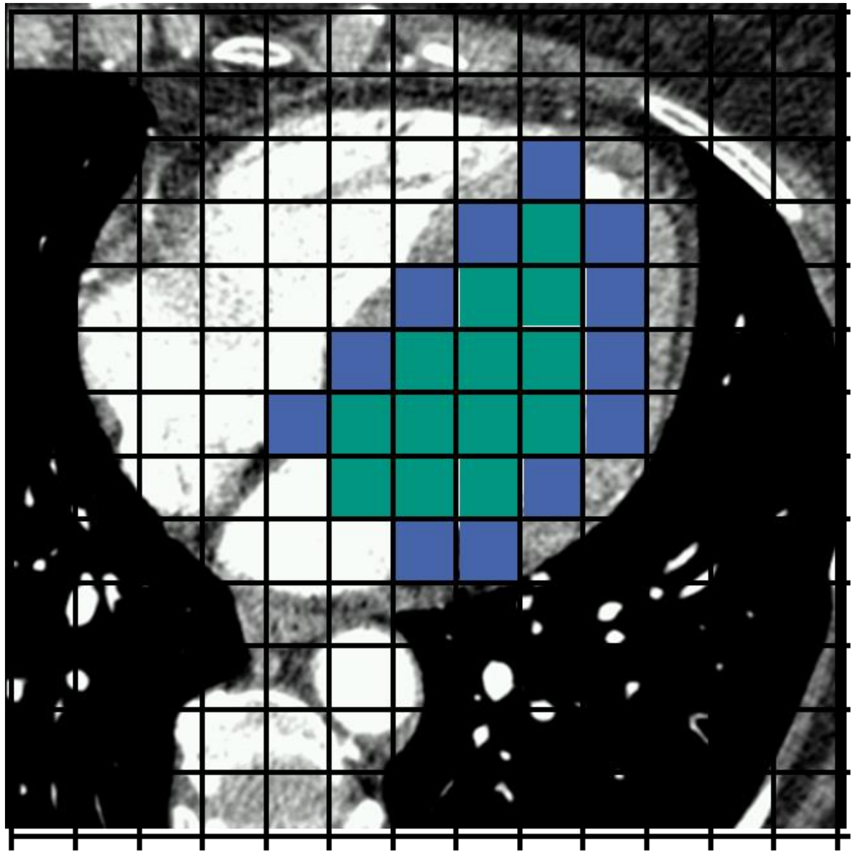
Regiongrowing



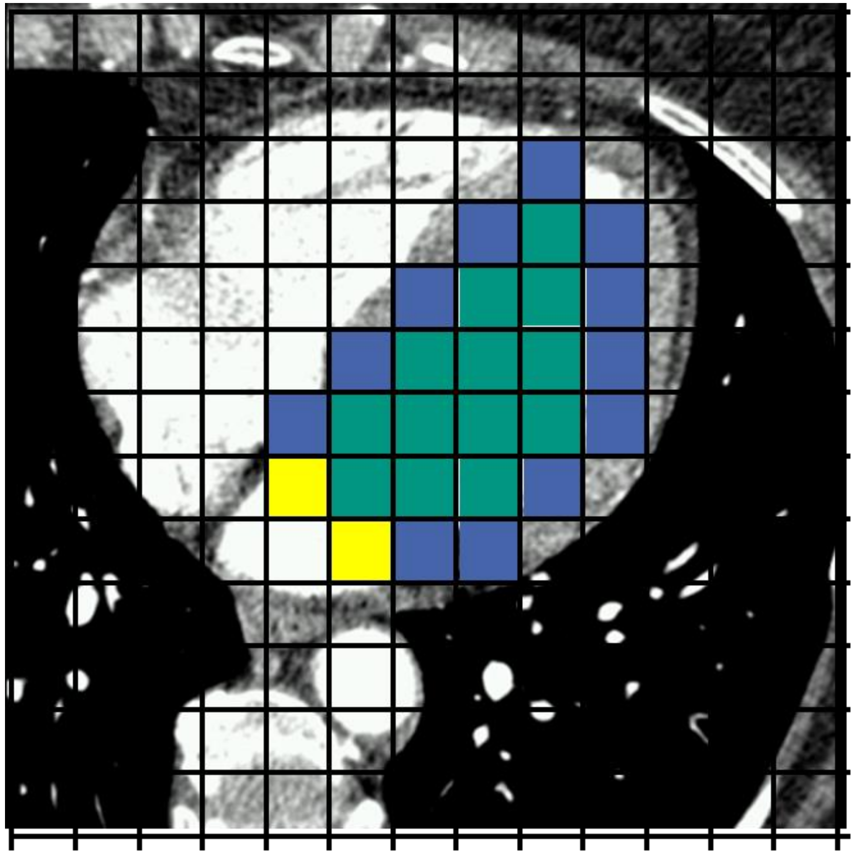
Regiongrowing



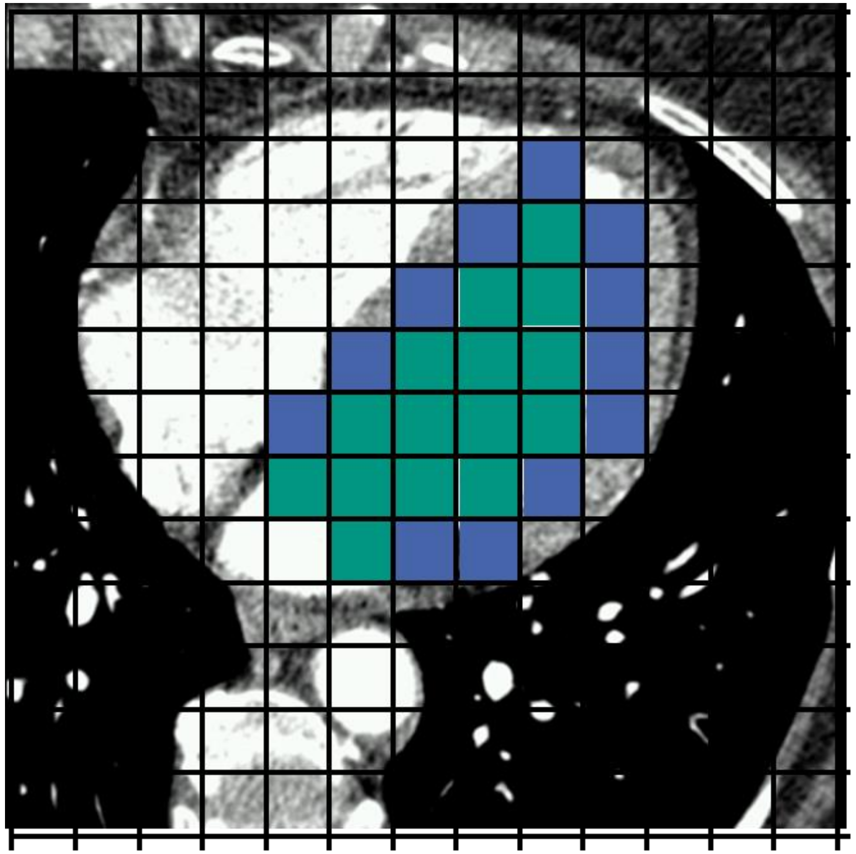
Regiongrowing



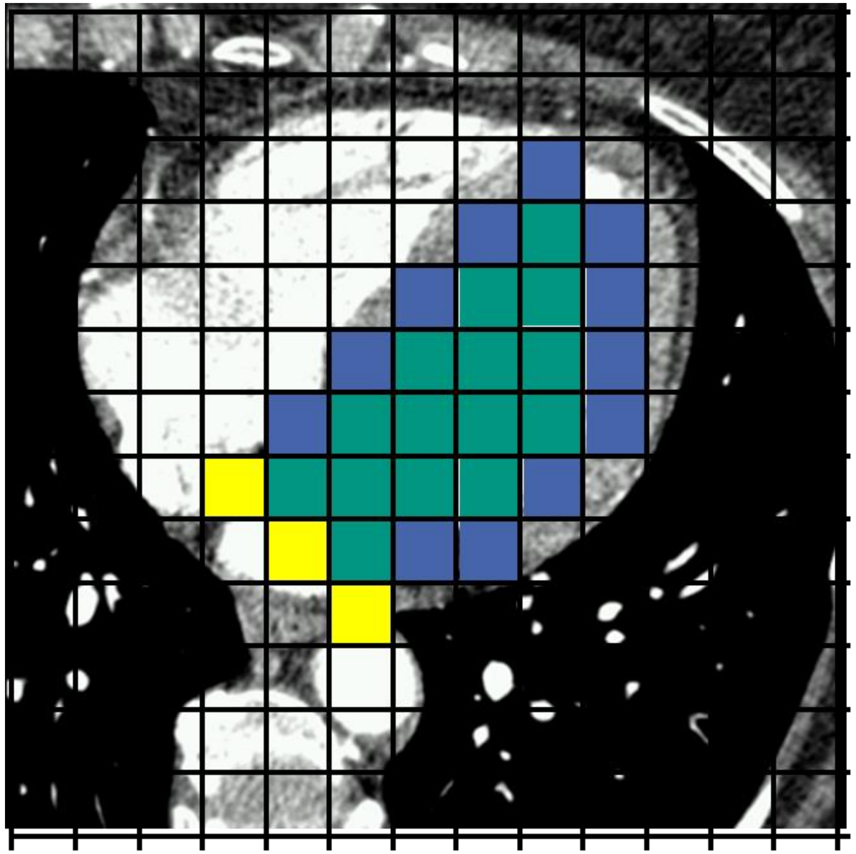
Regiongrowing



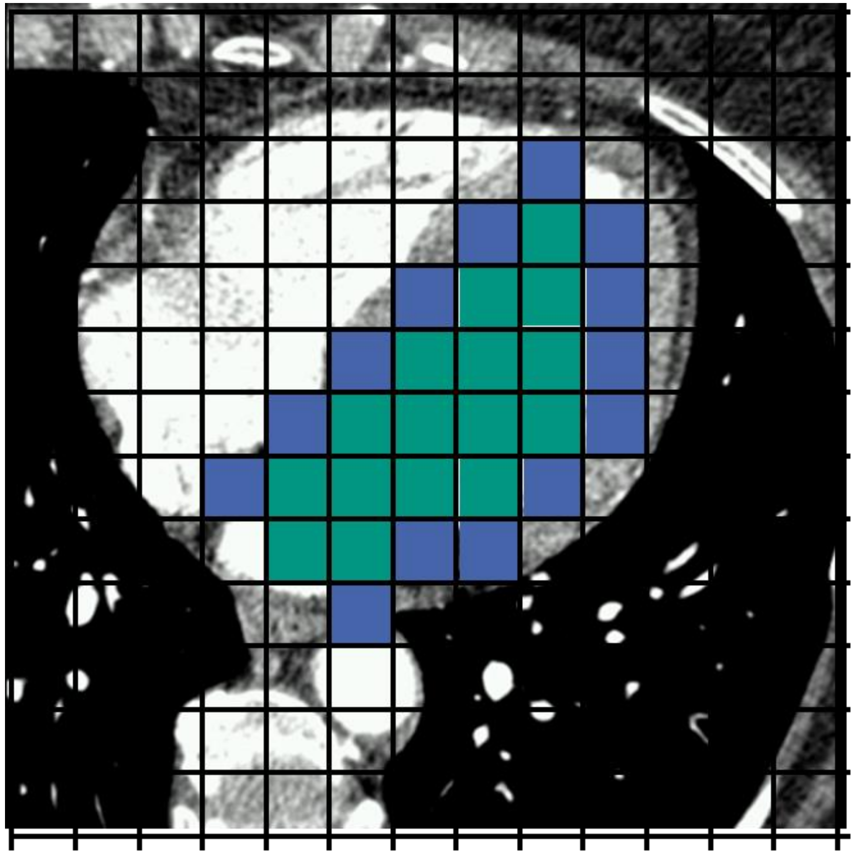
Regiongrowing



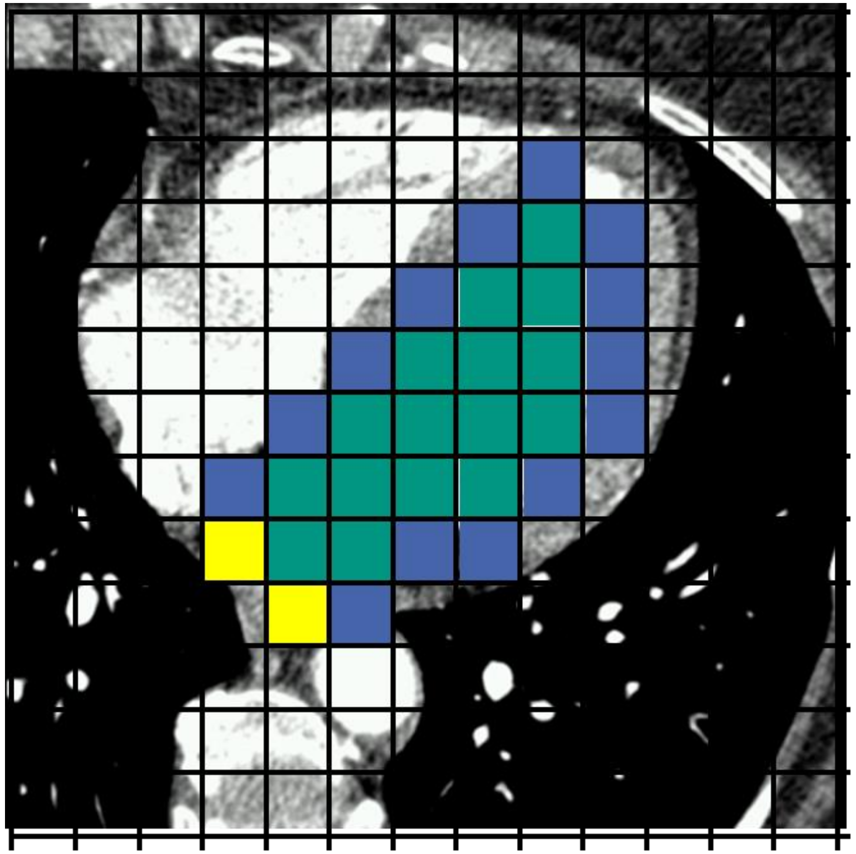
Regiongrowing



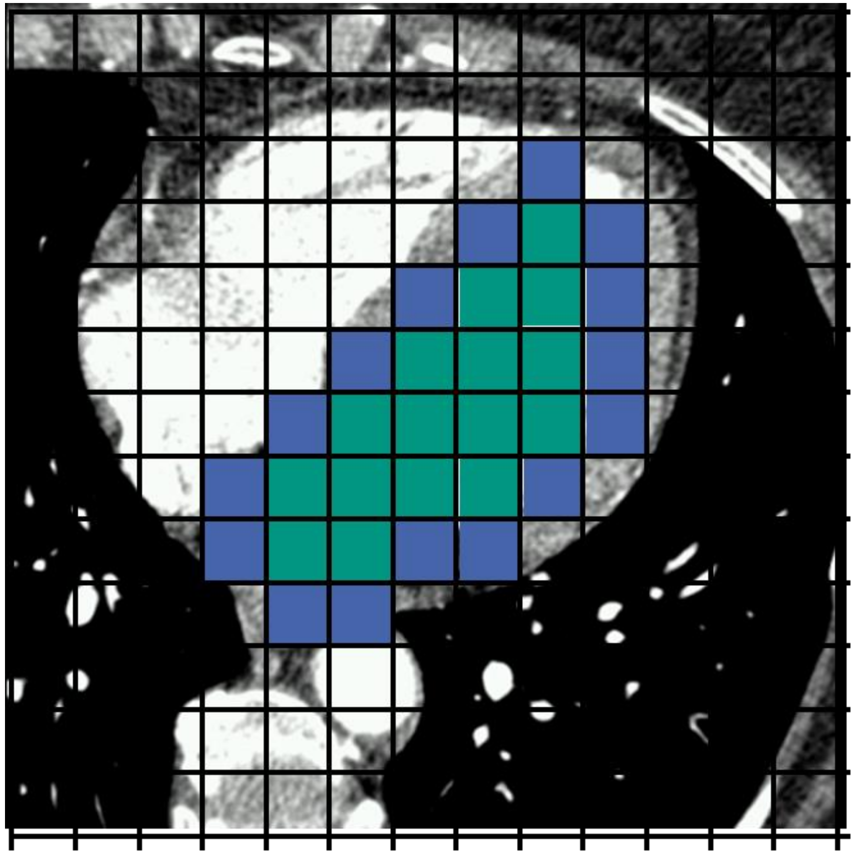
Regiongrowing



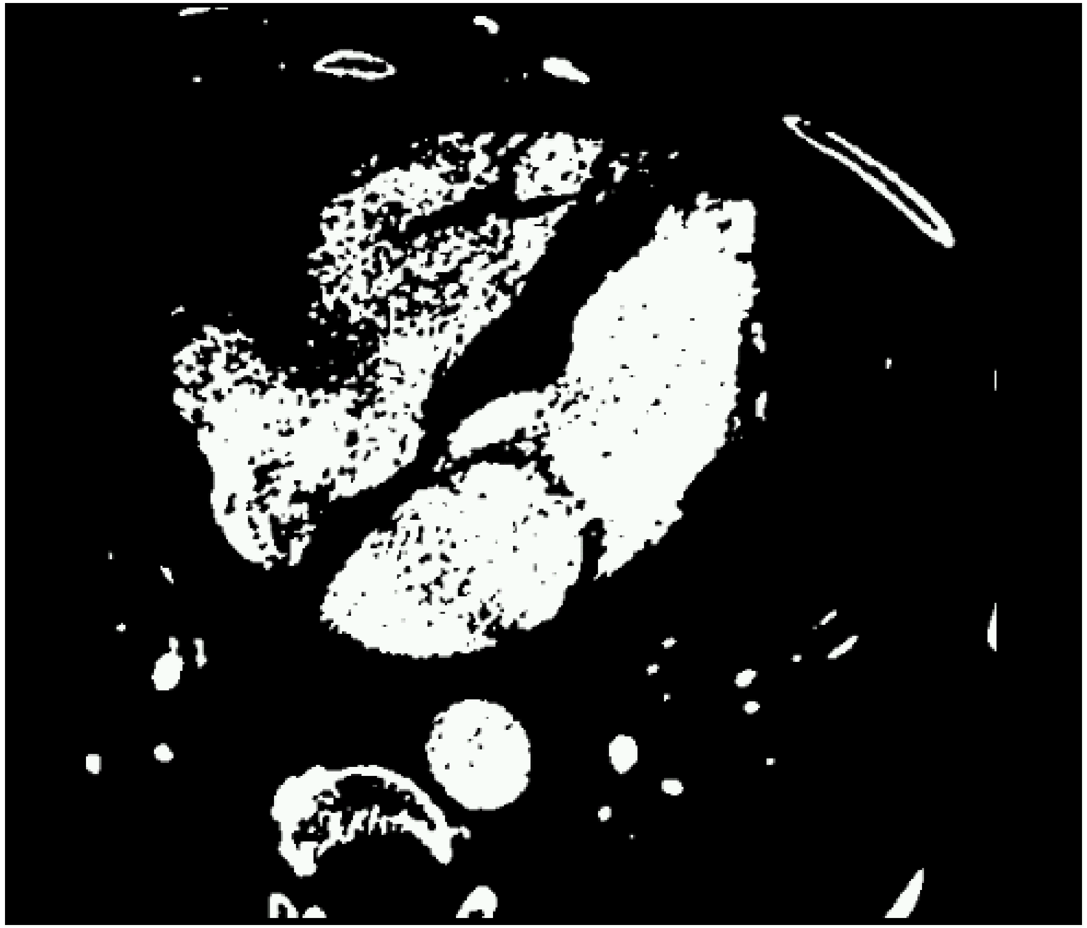
Regiongrowing



Regiongrowing

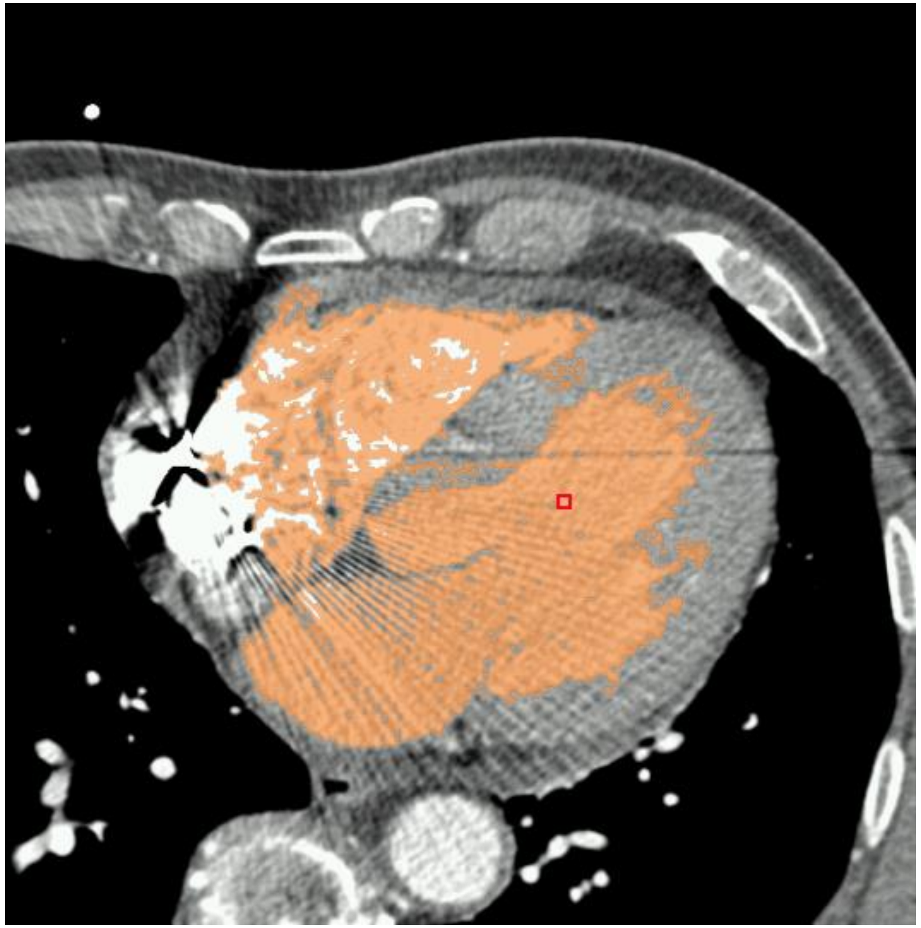


Regiongrowing vs. Threshold

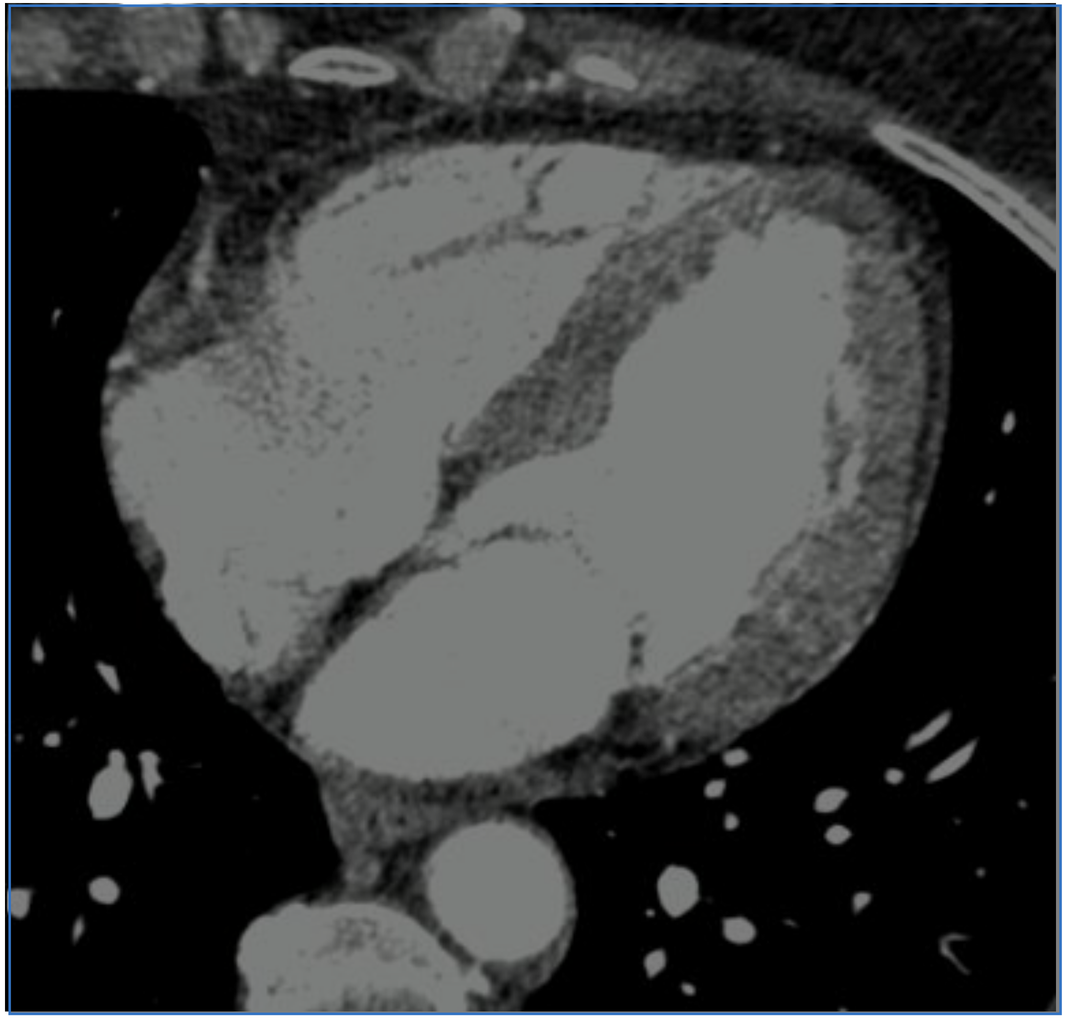


Regiongrowing - Evaluation

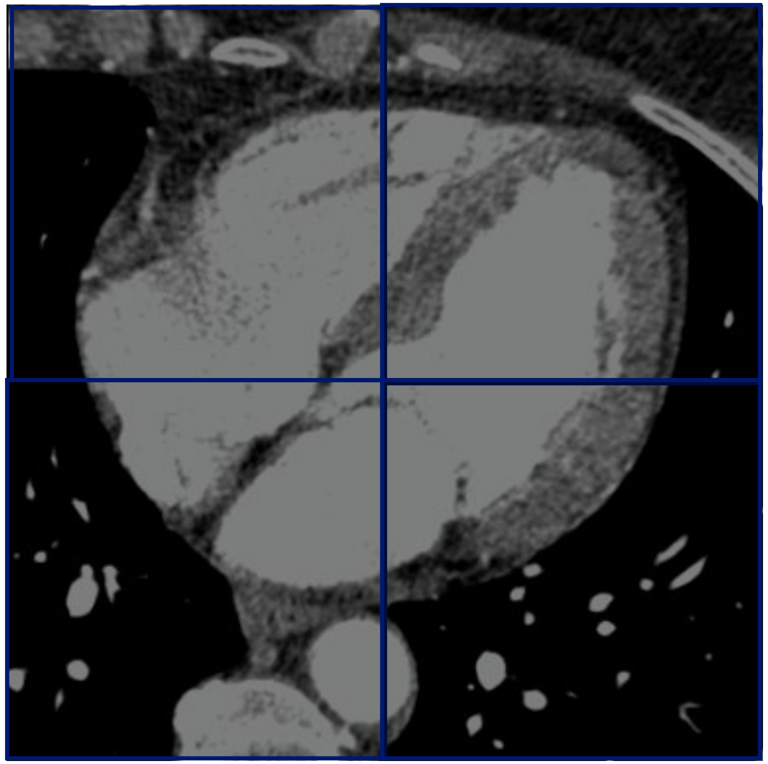
- Easy to implement
- Complexity and runtime depend on criteria H
- Prone to artefacts
- Hierarchical combination
  => Split & Merge



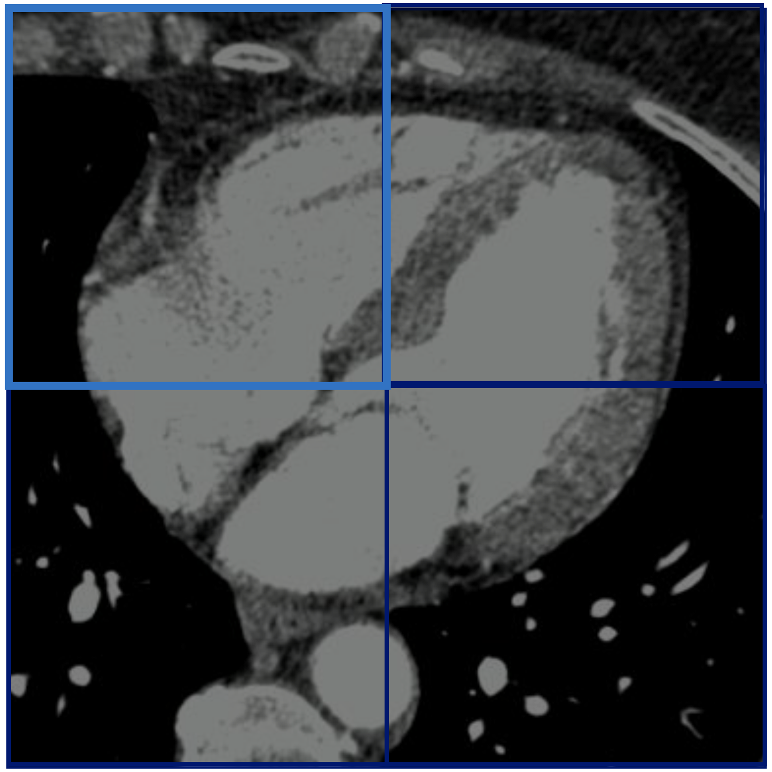
Split & Merge



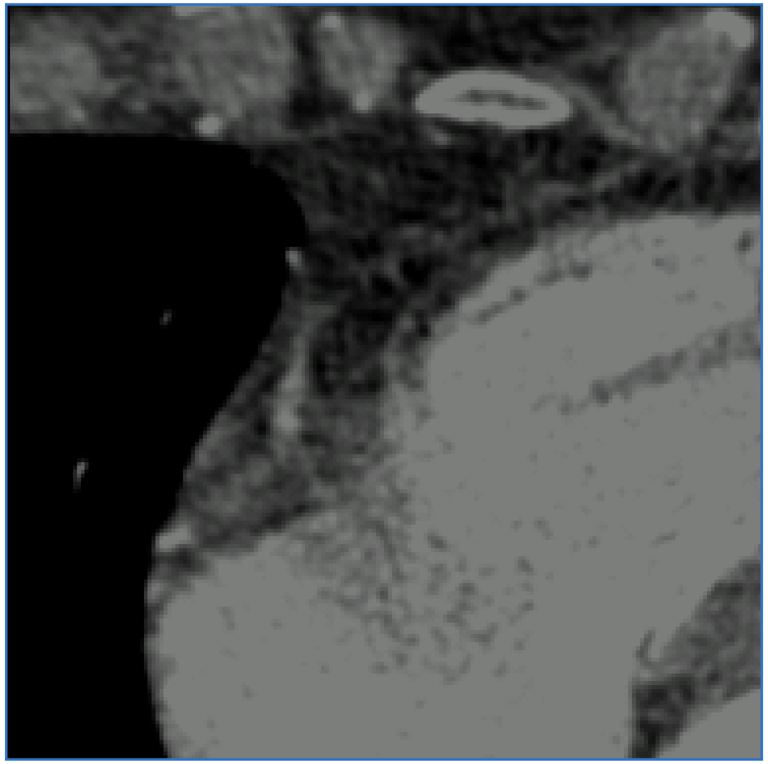
Split & Merge



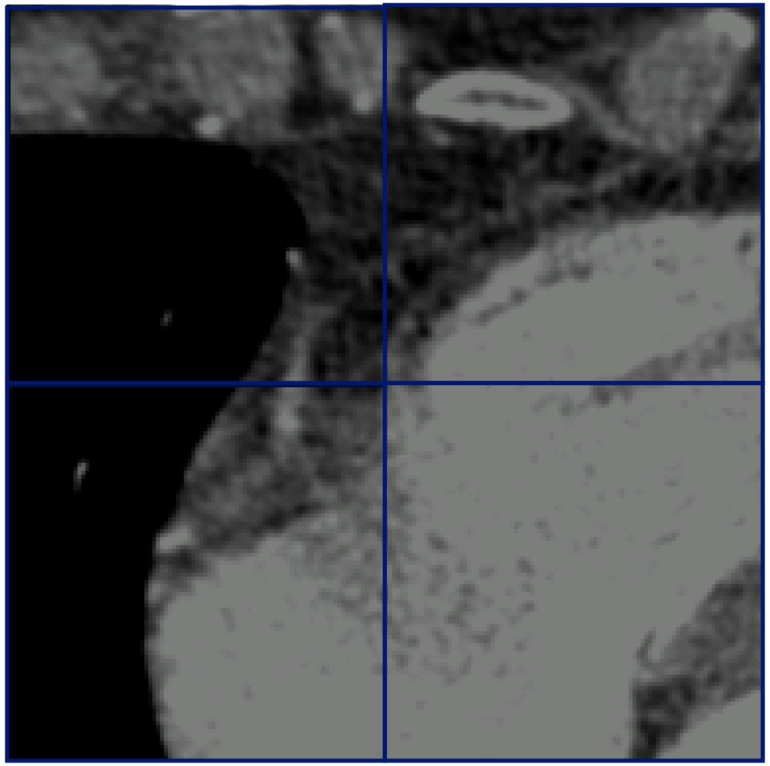
Split & Merge



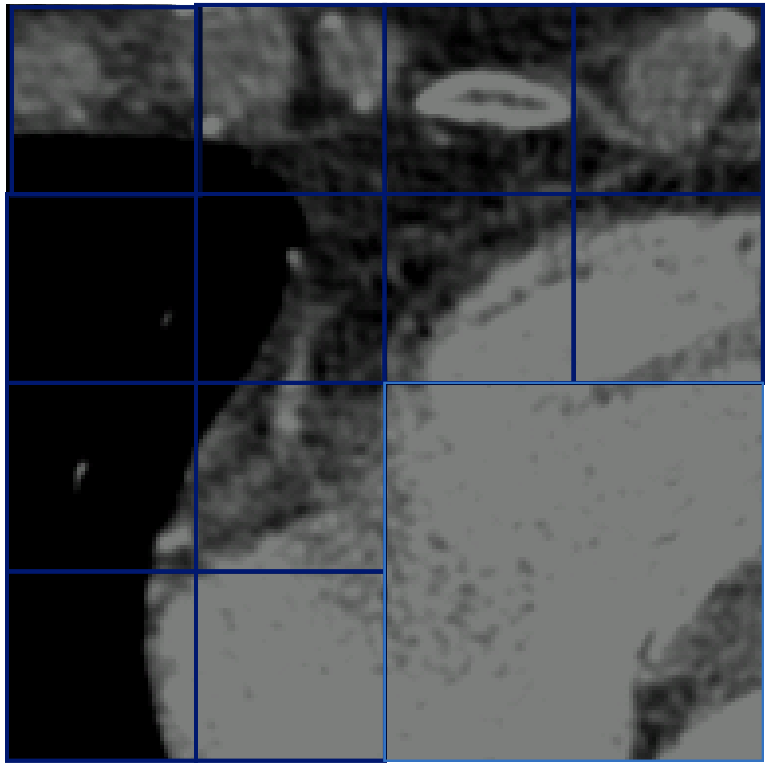
Split & Merge



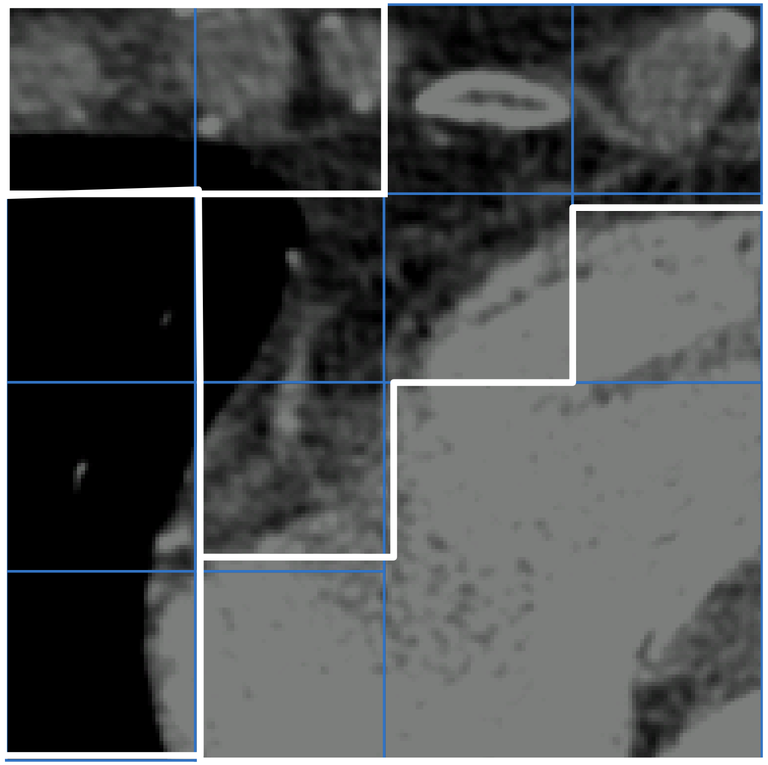
Split & Merge



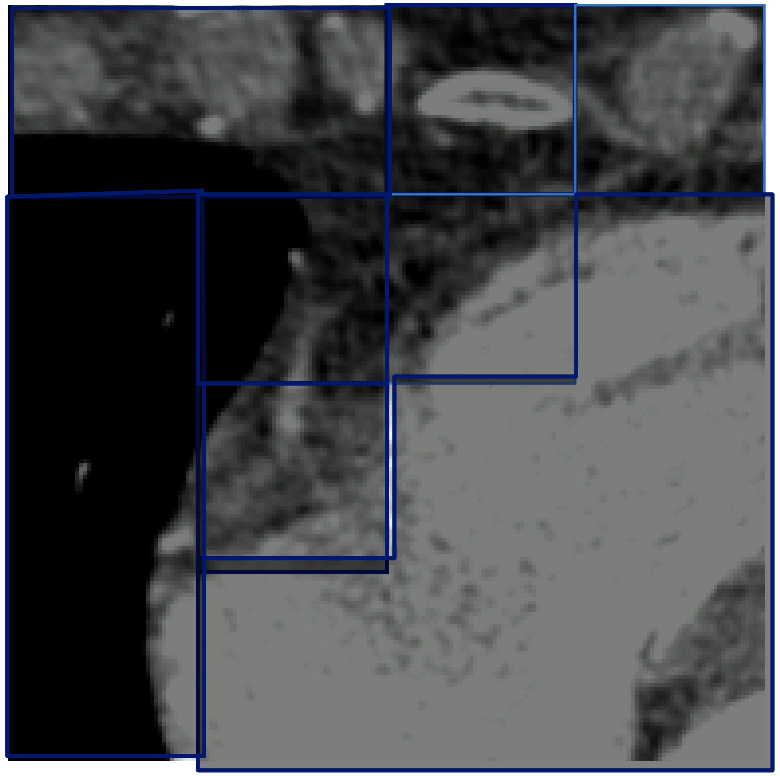
Split & Merge



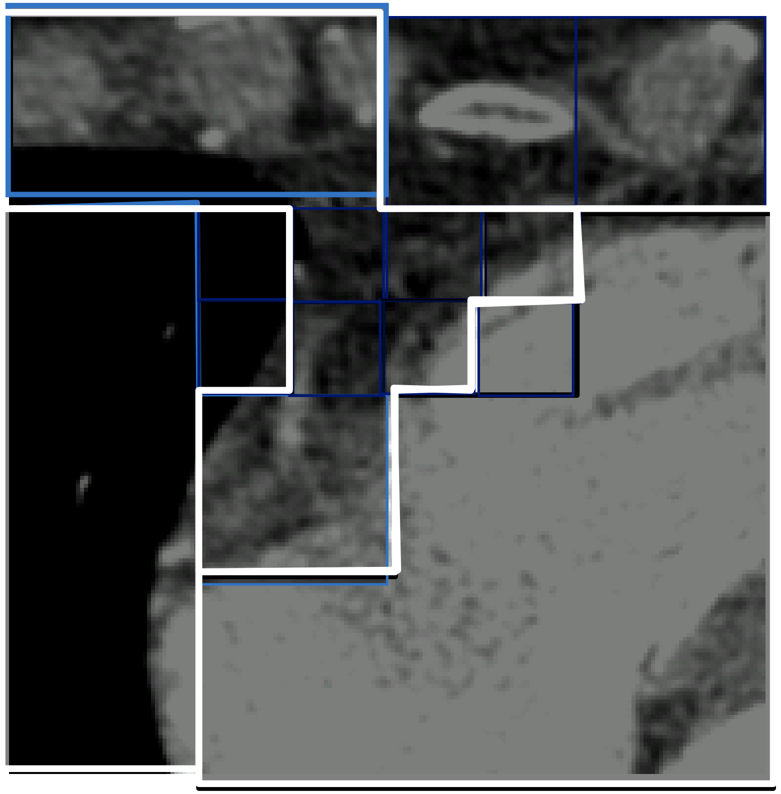
Split & Merge



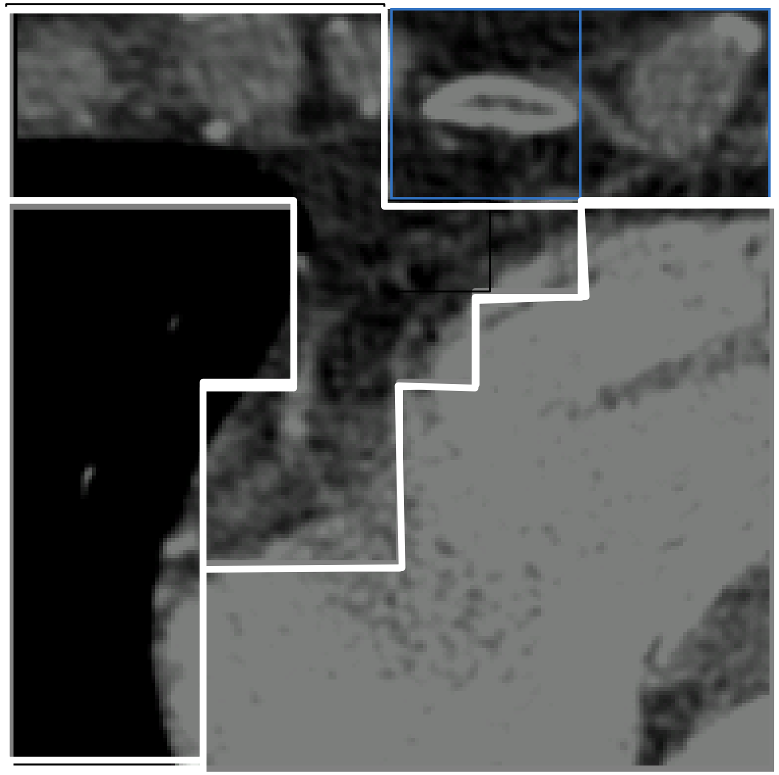
Split & Merge



Split & Merge



Split & Merge



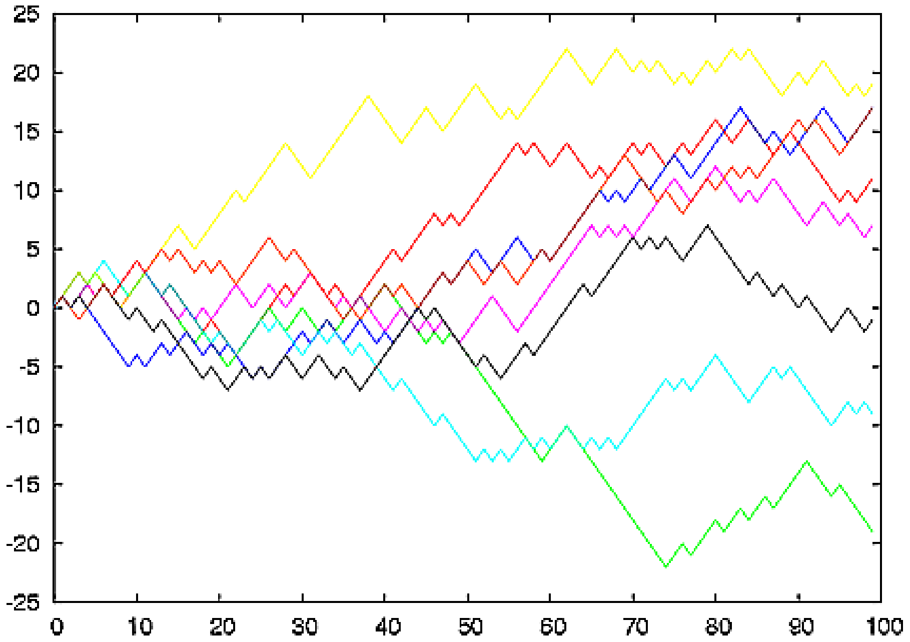
RANDOM WALKER SEGMENTATION

Random Walker Segmentation

- A random walk is a mathematical object, known as a random process, that describes a path that consists of a succession of random steps

- 1D Random walker = Bernoulli Process

- „Random Walker“ is the fictitious person that executes the steps



Simulation of 1D-Random-Walks

„random walk“ on graphs

- Graphically:
  - Random Walker: drunk person in a city
  - Crossroads of streets: nodes
  - Streets: edges
  - Probability of crossing: weight of the edge

- Question:
  - Let S be the starting node, with which probability reaches a „Random Walker“ a specific node?

Random Walker and Segmentation

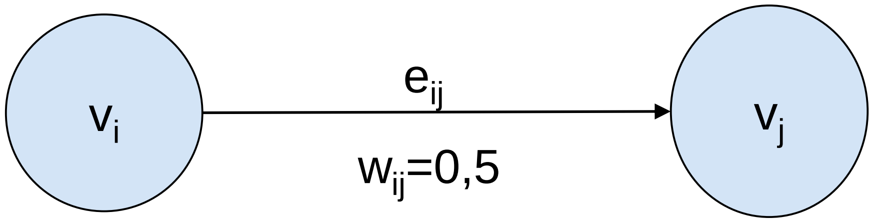
Random Walker

- Input
  - Image: Pixels/Voxels with a specific grey value and position
  - Several seeds are defined by the user

- Question:
  - With which probability reaches a random walker which starts at pixel a a defined seed first?
    - Pixel a is assigned to the seed where the probility is the highest
    - Path is influenced by the grey values

Random Walker Definitions

- Graph: G(V,E) (V: nodes, E: edges)
- if v_i and v_j are connected the edge e_{ij} exists
- Every edge has a weight $w(e_{ij}) = w_{ij}$
- $w_{ij} > 0$ is assumed



Random Walker: from image to graph

- Every pixel is a node (v_i)
- If v_i and v_j are neighbors insert edge e_{ij}
- Let g_k be the grey value of pixel v_k then the edge weight is:

$$w_{ij} = e^{-b(g_i - g_j)^2}$$

- Goal of the weight: Define probability of a random walker transition:
  - Meaningful: big grey value differences have low edge weight
  - Other weight functions possible

Random Walker: Segmentation

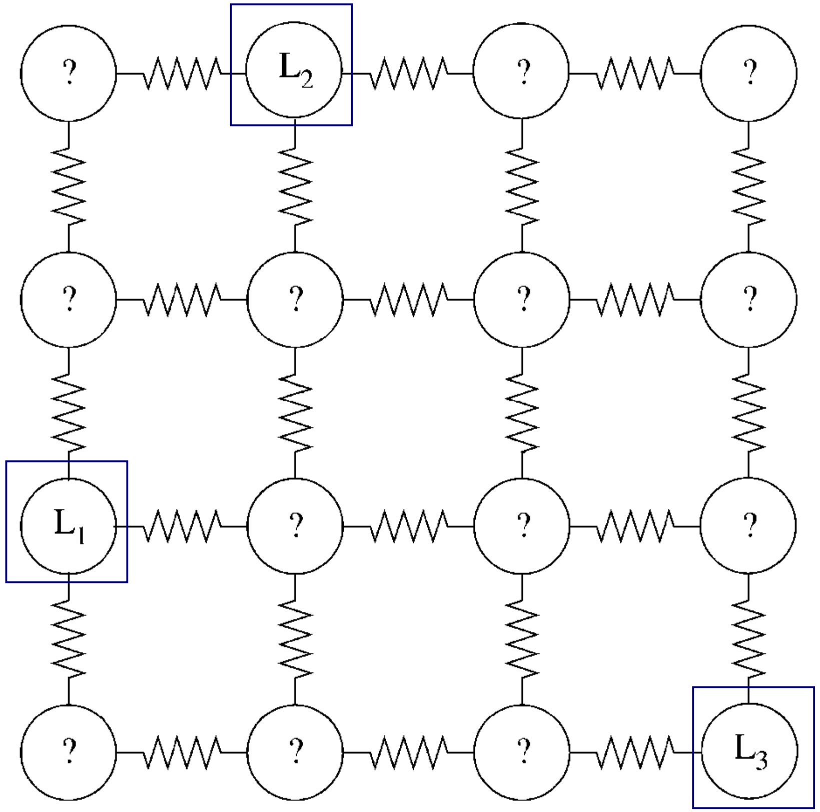
(1) Define Seed s

(2) Determine for every node v_i the probability x_i^s that a random walker reaches a seed s first

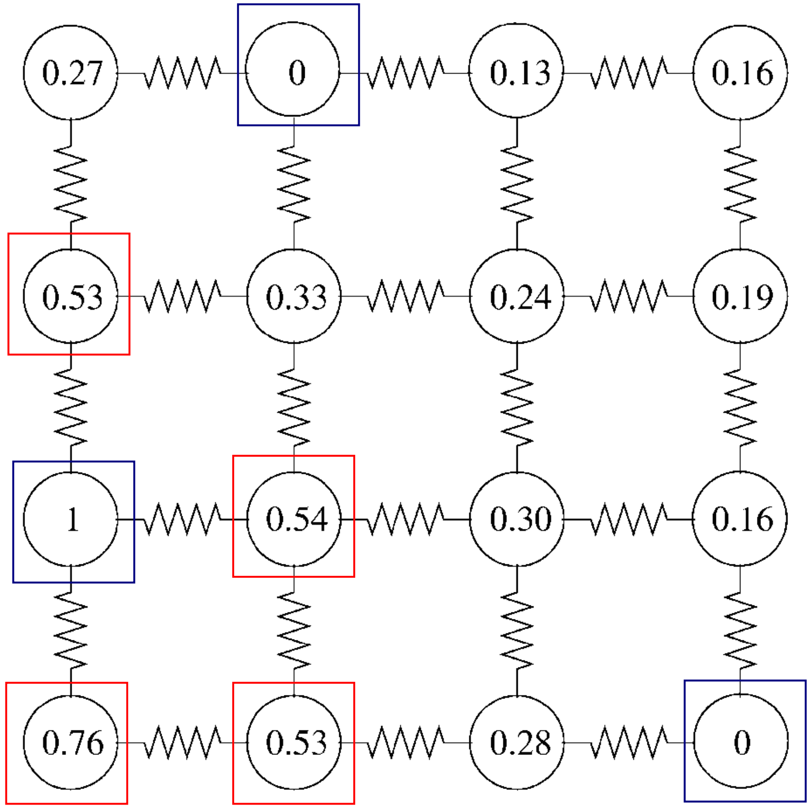
(3) If all nodes are processed choose the next seed

(4) If all seeds are processed
- Assign node to the seed with the highes probability

Segmentation example 1

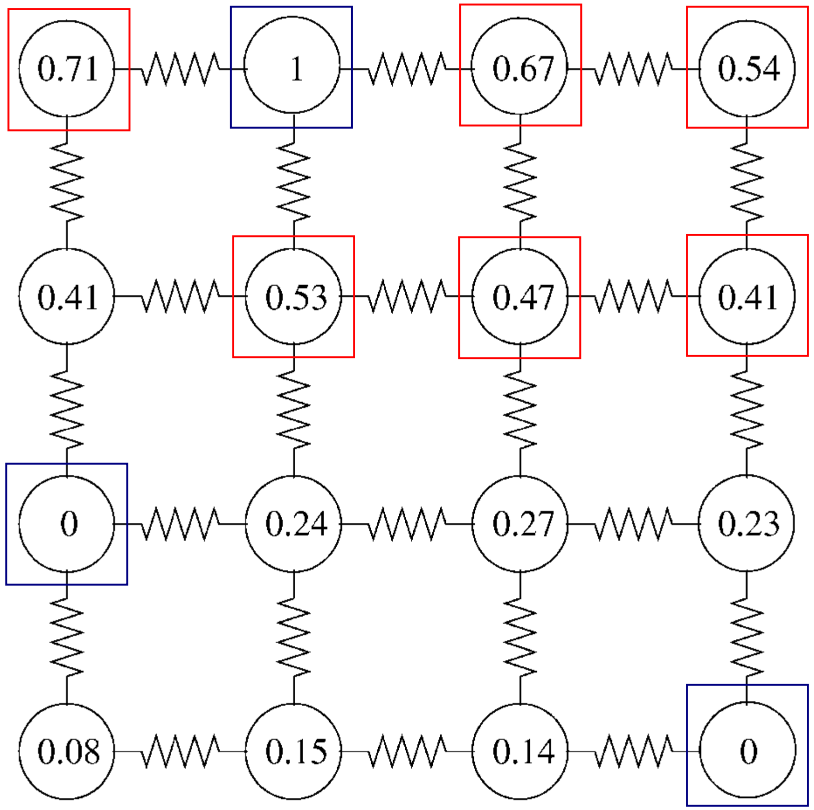


Segmentation example 1



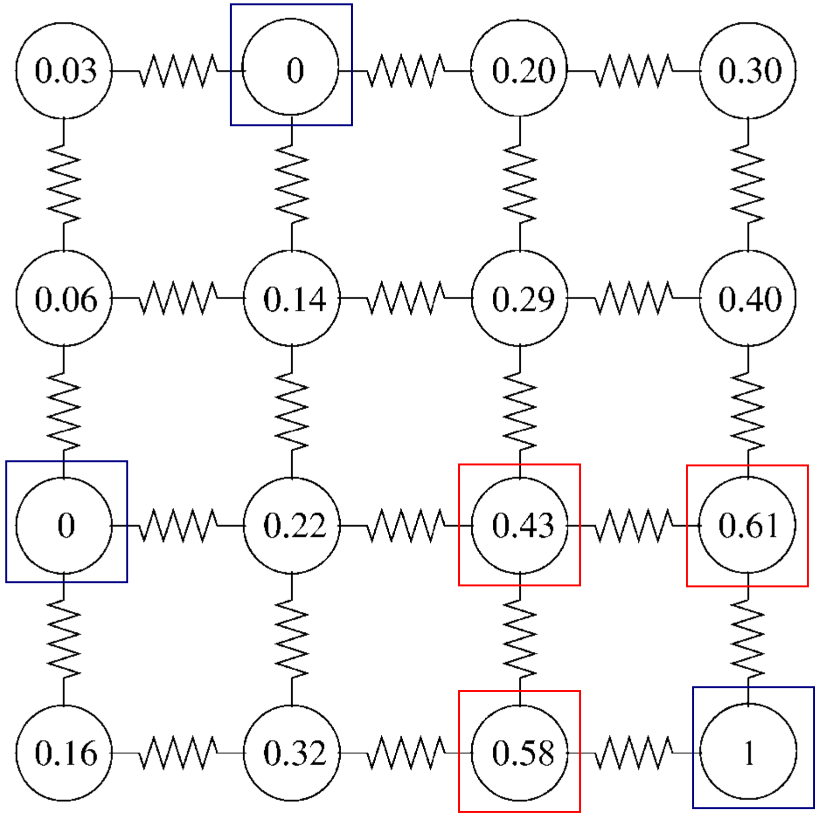
Seed L1 is processed

Segmentation example 1



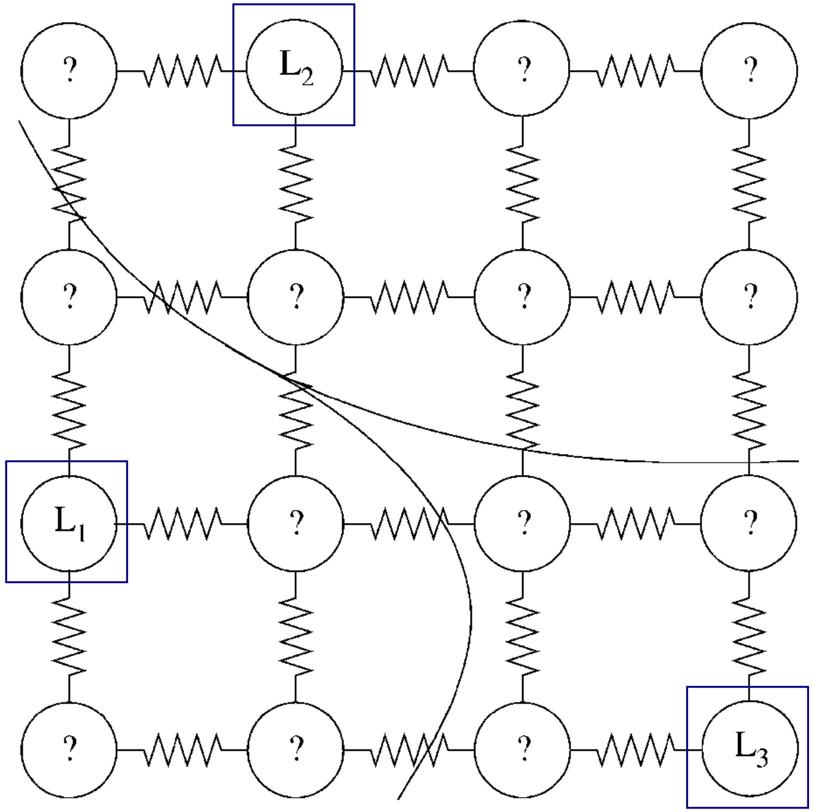
Seed L2 is processed

Segmentation example 1



Seed L3 is processed

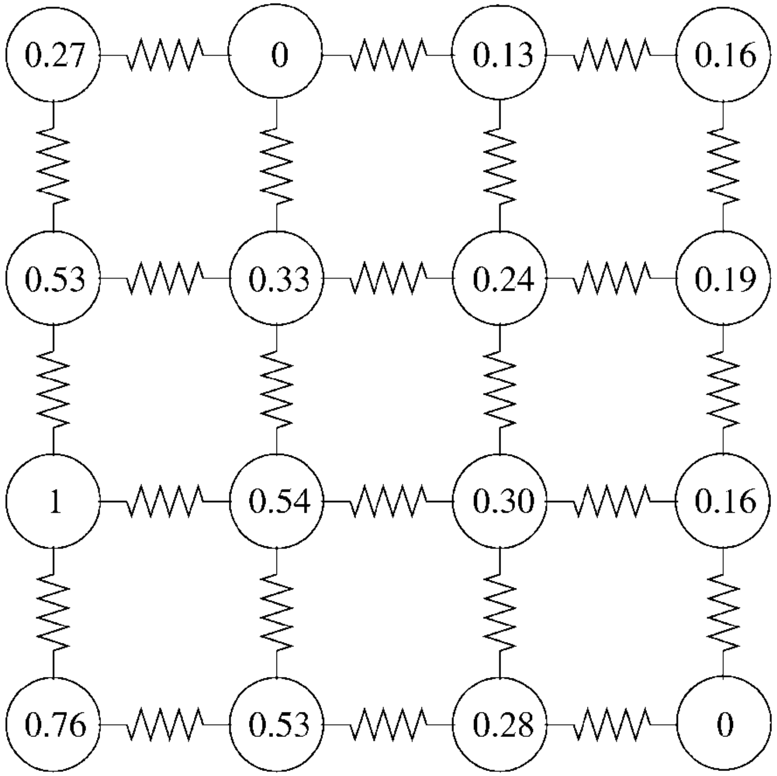
Segmentation example 1



Result

Random Walker – Analogies of electric circuits

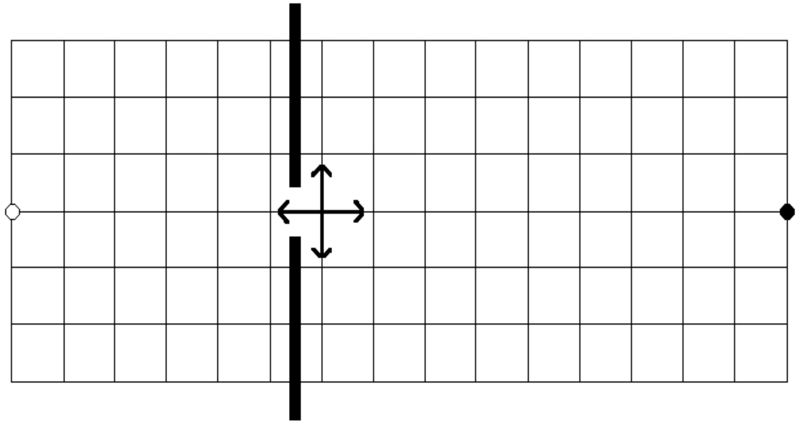
- Choosen Seed = Voltage source
- Other seeds = mass
- Consider Kirchhoff rules and Ohm laws



Random Walk does not have to be simulated, direct solution via equation system

Random Walker: Weak Boundaries

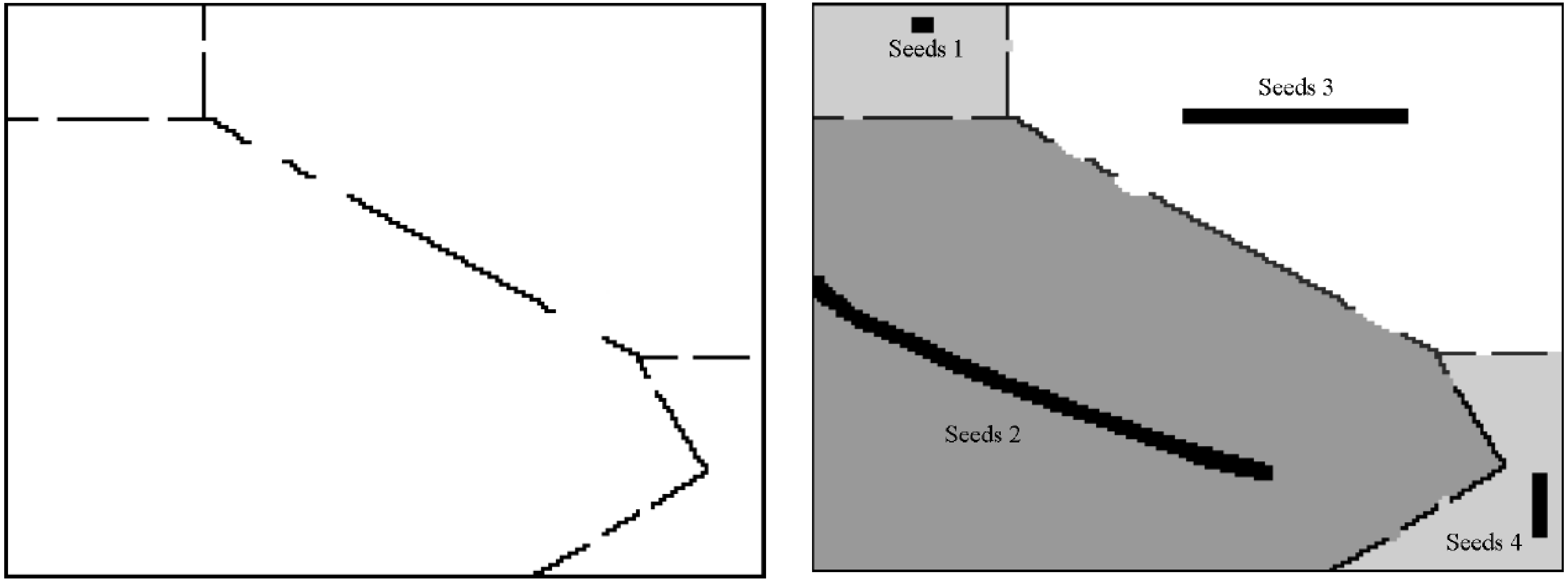
- Weak Boundaries: edges are not clear
- Often in medical images
- Detection of weak boundaries is critical regarding segmentation quality



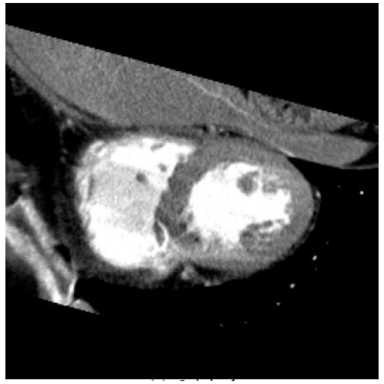
Two seeds and weak boundaries

- Result: Detection of weak boundaries is „build in“

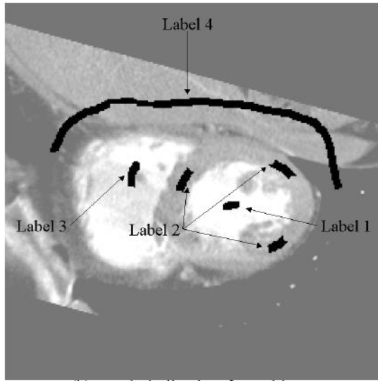
Segmentation example 2



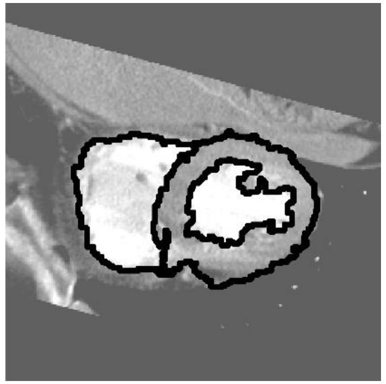
Segmentation example 2



Original

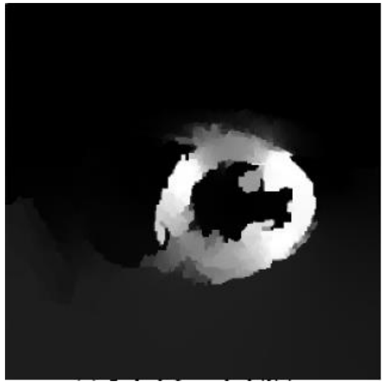


Seeds



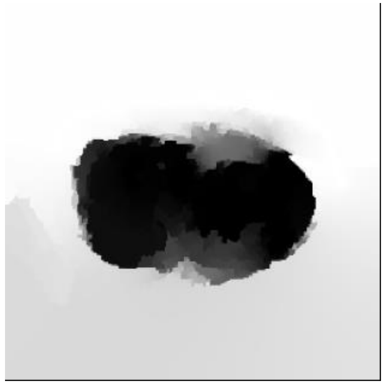
Results

Probability L1



Probability L2

Probability L3



Probability L4

Random Walker: Conclusion

- Application of graph-theoretical approaches for segmentation
- Robust regarding „weak boundaries“
- Semi-automatic method, uses knowledge of the user

- Solution via Dirichlet problem. Seeds Saatpunkte correspond to boundary conditions
- Calculation: sparse linear equation system has to be solved
- No simulation of Random Walker necessary!

CONTOUR BASED SEGMENTATION

Active Contours - Snakes

- Idea:
  - User defines initial segmentation (Contour, close to an edge), computer calculates the results (on the edge)

Active Contours – Example

1. User defines contour
2. Computer calculates solution
3. Solution

Active Contours - Snakes

- What is a snake?

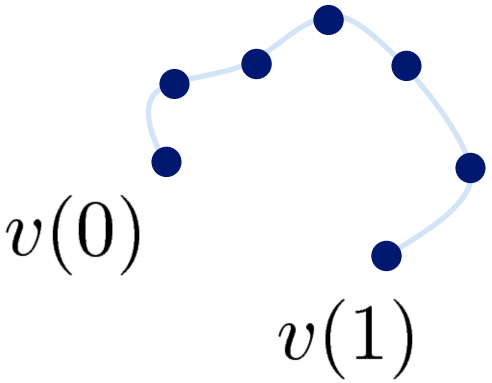
  - Perception of a contour (via a curve) as enery minimzation problem
  - Combination of internal and external energy
  - Internal energy limits the form of the contour (e.g. smoothness of the contour)
  - External energy defines the image features that attract the contour (e.g. gradient)
  - Graphical: Rubber bands that gradually adapts to image content

Active Contours

- Definition:
  - Representation of a curve through n points

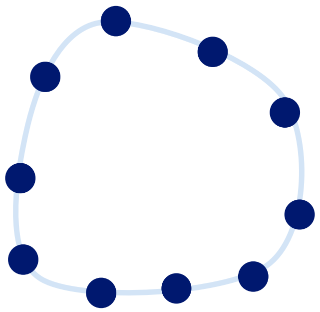
$$v : [0,1] \rightarrow \mathcal{R}^2$$

$$v(s) = \begin{pmatrix} x(s) \\ y(s) \end{pmatrix}$$



- Closed Curve

$$v(0) = v(1)$$



Active Contours

- Energy minimization problem

$$E = \int_0^1 (E_{int}(v(s)) + E_{image}(v(s))ds$$

Internal Energy

Describes the form of a curve

External Energy

Adaption to the edge

Active Contours – Internal Energy

- Internal engery describes the form of the contour
- Weighted sum of 1. and 2. derivation

$$E_{\text{int}}(v(s)) = \frac{1}{2}\left(\alpha(s)\cdot\left|v_s(s)\right|^2 + \beta(s)\cdot\left|v_{ss}(s)\right|^2\right)$$

- α(s) and β(s) are weight factors
  - Often constant
  - Definition of stiffness and strain of the curve

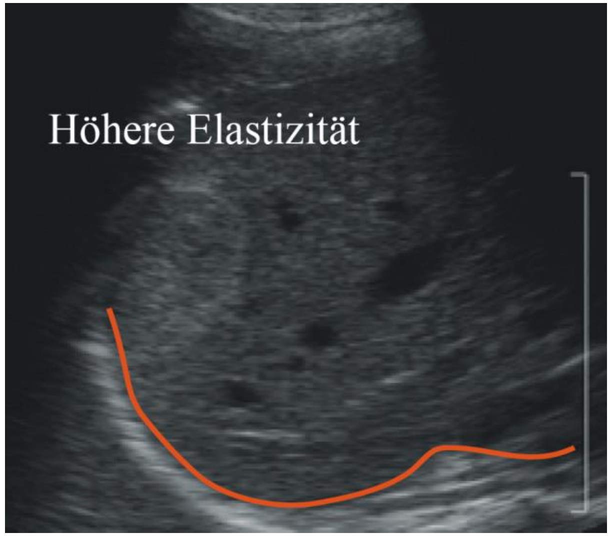
- 1. derivation:

$$v_s(s) = \left(\frac{\partial}{\partial s}x(s), \frac{\partial}{\partial s}y(s)\right)$$

- 2. derivation

$$v_{ss}(s) = \left(\frac{\partial^2}{\partial s^2}x(s), \frac{\partial^2}{\partial s^2}y(s)\right)$$

Active Contours: Energy dependencies



Active Contours – External Energy

- External energy – Image energy
  - The curve has to adapt to the image
  - Possible features:
    - Grey values
    - Edges

    - Definition of the components as function:

$$E_{image} = \omega_{grayvalue} \cdot E_{grayvalue} + \omega_{gradient} \cdot E_{gradient}$$

Active Contours – External Energy

- External Energy – Image energy
  - Grey values:

$$E_{grayvalue}(x, y) = I(x, y)$$

    - But: curve adapts only to bright or dark regions drückt

  - Gradient:

$$E_{gradient}(x, y) = ||\nabla I(x, y)||$$

Active Contours – External Energy

- Gradient is sensitive to noise

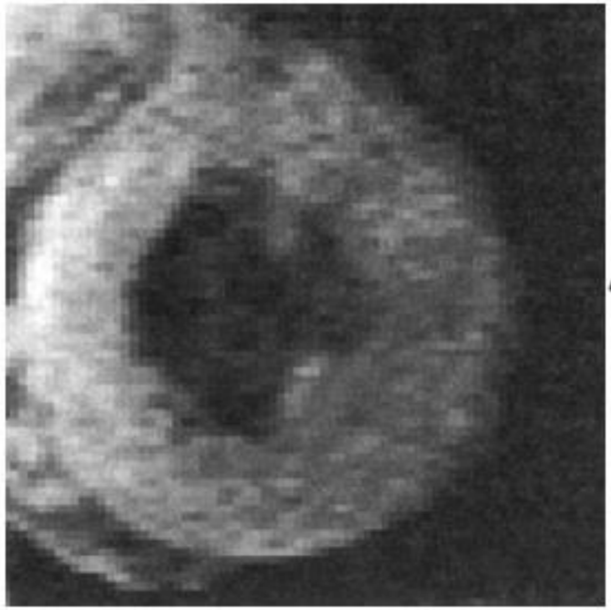
Smoothing of the image

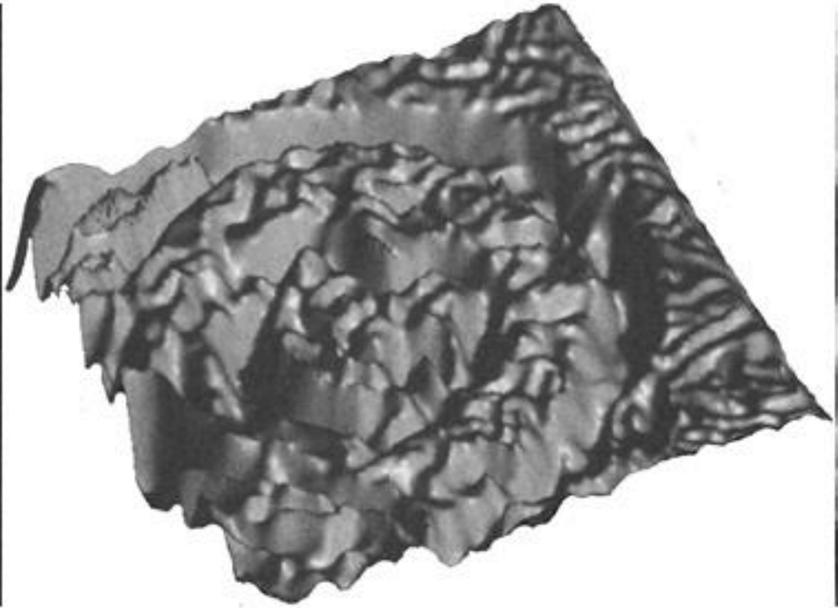
$$E_{ext}(x,y) = -|(\nabla G_\sigma(x,y)) * I(x,y)|$$

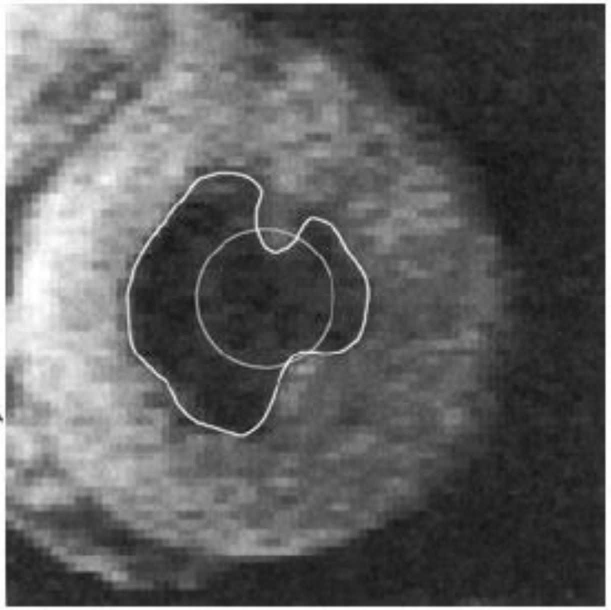
$$g(i,j) = \frac{1}{2\pi\sigma^2} e^{-\frac{i^2+j^2}{2\sigma^2}}$$

Sobel Filter

Active Contours – External Energy







Active Contours - Problems

- Energy functional: local minima

⇨ Minima is not global

Solution

- Control forces
- Gradient Vector Flow

Active Contours – Control Energy

$$E = \int_0^1 (E_{int}(v(s)) + E_{image}(v(s))ds + E_{con}(v(s))ds$$

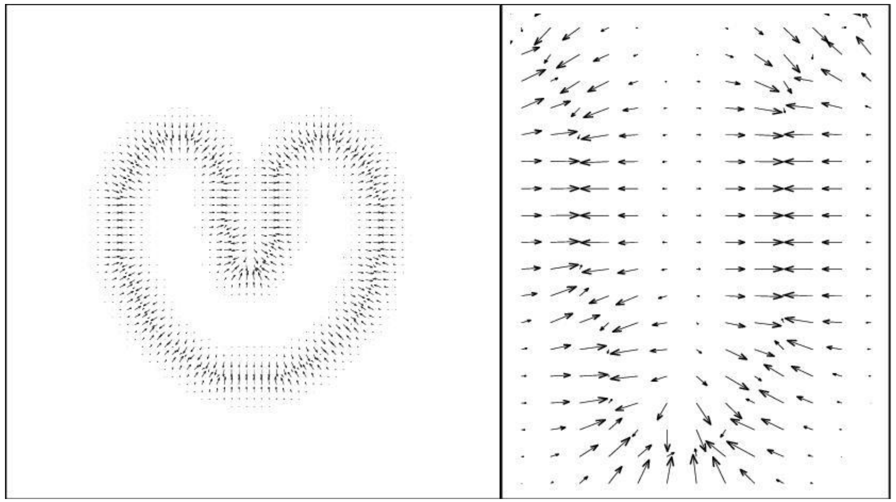
Control Energy

- Optional third energy
  - Defined by the user

  - Volcano forces:
    - High Energy: contour is pusehd

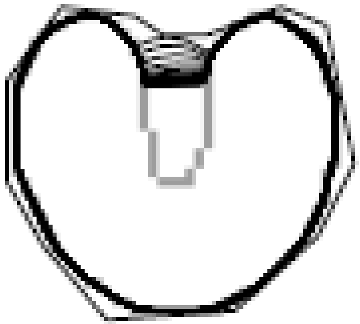
  - Spring forces
    - Low Energy: attract contour

Active Contours – Gradient Vector Flow



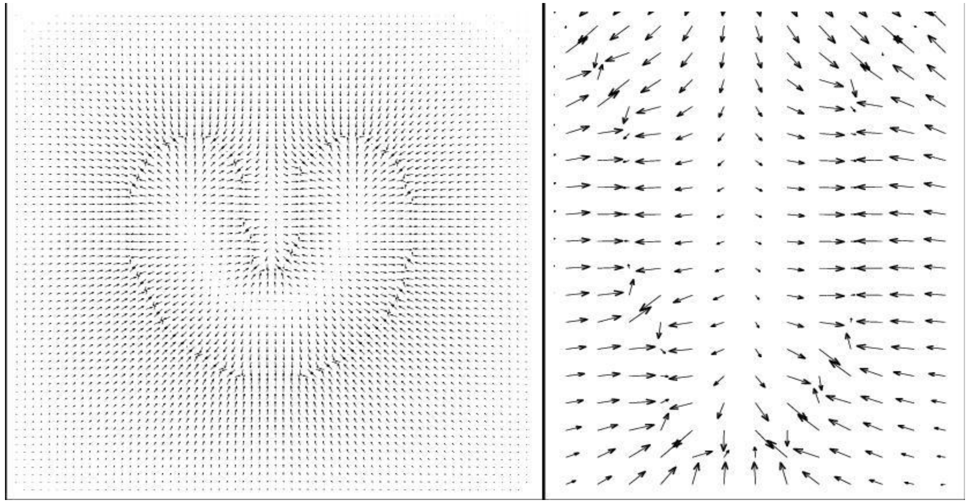
Quelle: Xu et al.

- Problem: gradients are too far away from contour

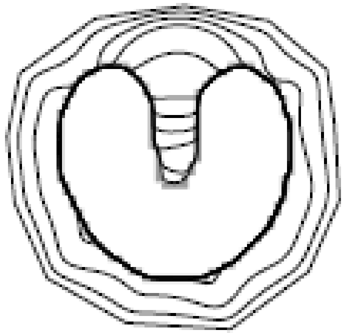


Quelle: Xu et al.

Active Contours – Gradient Vector Flow

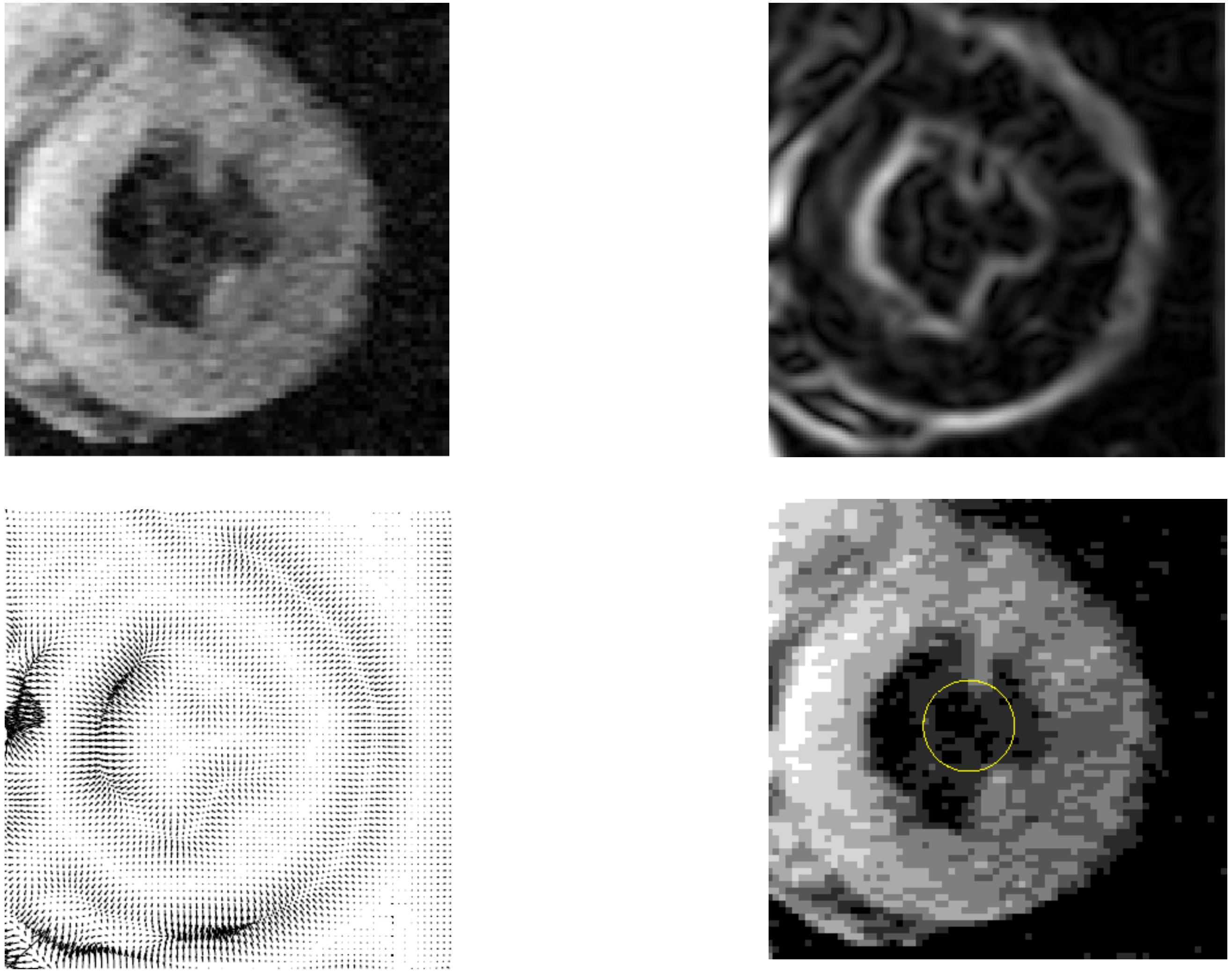


Quelle: Xu et al.



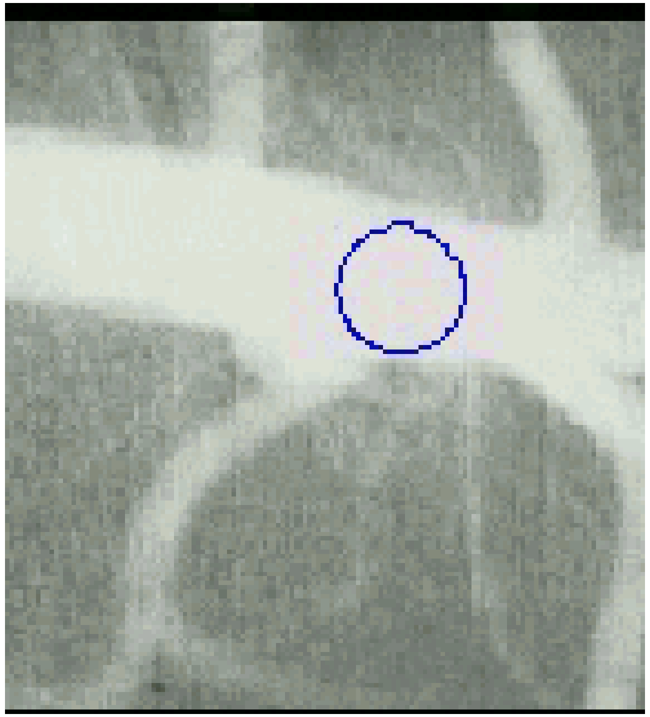
- For every pixel direction, force and distance of the next gradient is calculated

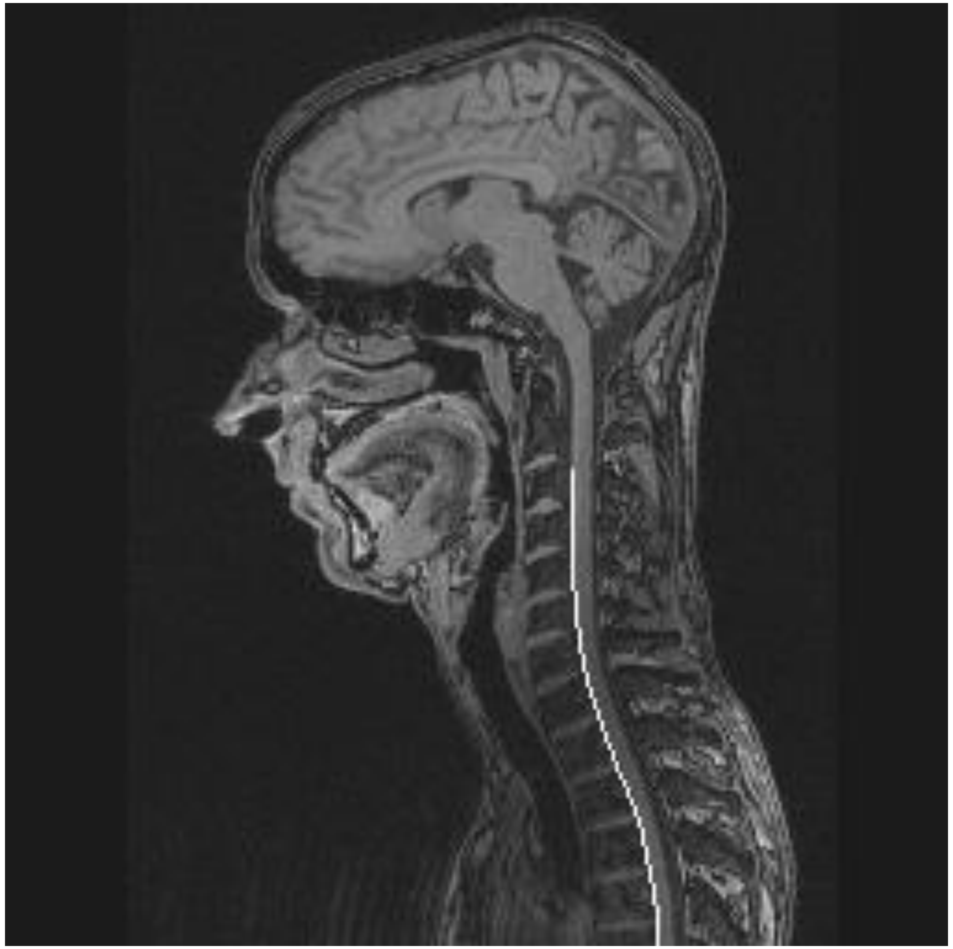
Active Contours – Gradient Vector Flow



Quelle: Xu et al.

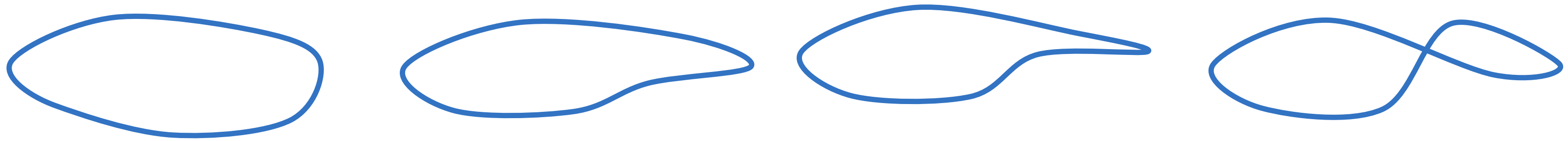
Active Contours example





Active Contours - Conclusion

- Combination of image features and implicit form knowledge
- User has to define weight functions
- Active contours can bridge gaps
- Robust against noise
- Not possible solutions have to checked:



Active Contours - Conclusion

- Definition of energies is very complex

⇨ Every problem has ist own requirements

Advantage:

- Combination of user interaction and computer assistance

- Solution is close to user definition
  Vgl. Region Growing

- Solution also if image information is missing

Literature

- N. Otsu, “A threshold selection method from gray-level histograms,” IEEE Trans. Sys., Man., Cyber., vol. 9, pp. 62–66, 1979

- M. Kass, A. Witkin and D. Terzopoulos, Snakes: Active Contour Models, *First International Conference on Computer Vision*, 1987, pp. 259-268.

- Leo Grady: Random Walks for Image Segmentation, IEEE Transactions on Patern Analysis and Machine Intelligence, Vol 28, 2006