Introduction to Image Analysis

Image Segmentation

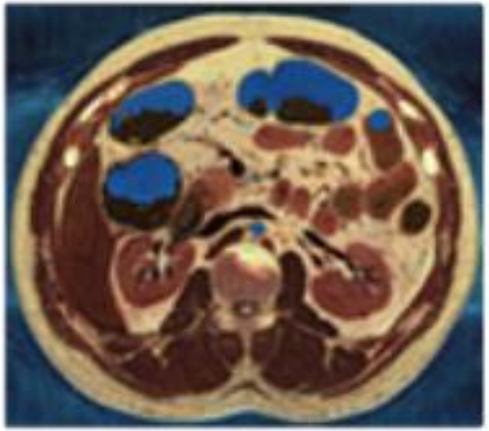
Mauricio Reyes

mauricio.reyes@unibe.ch

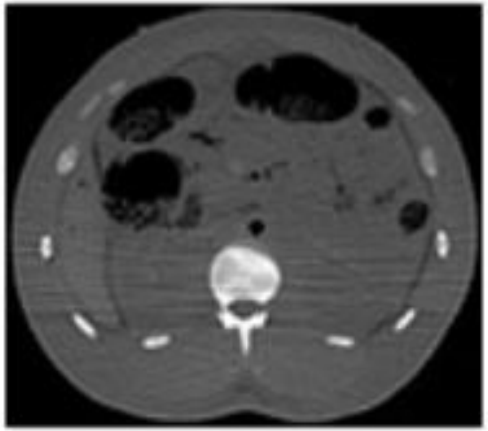
University of Bern
ARTORG Center for Biomedical Engineering Research
Medical Image Analysis Group

BASICS OF IMAGE SEGMENTATION

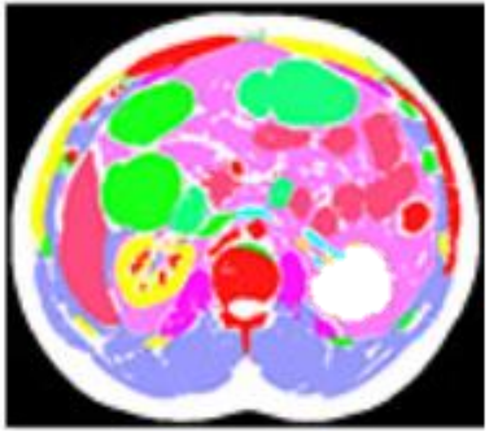
Segmentation – The Goal



Picture



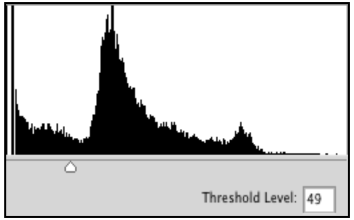
CT Scan

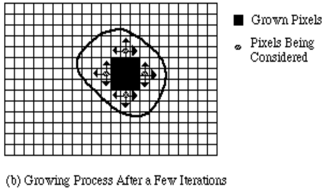


Labels

Visible Human Project
www.nlm.nih.gov/research/visible/visible_human.html

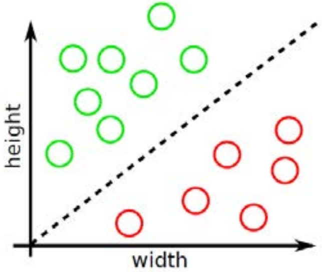
Segmentation Overview – Big Picture





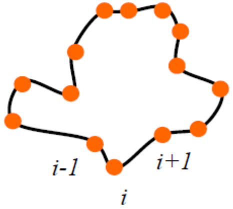
Simple Methods

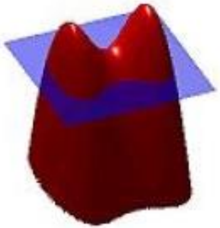
> Thresholding
> Region-Growing
> …



Classification + Clustering

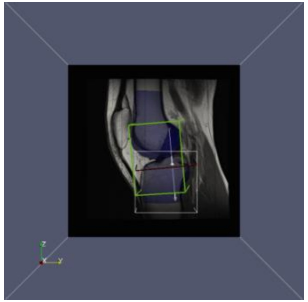
> kNN
> SVM
> …





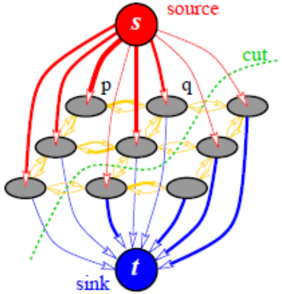
Deformable Models

> Snakes
> Level-Sets
> …



Active Models

> ASM
> AAM

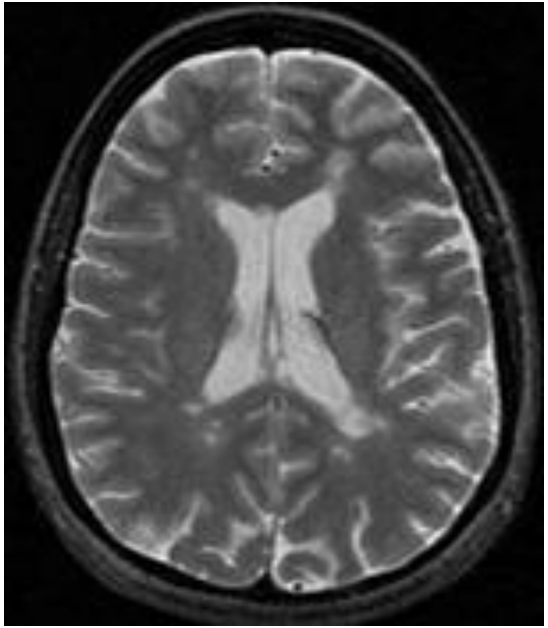


Random Fields & Graph-Cuts

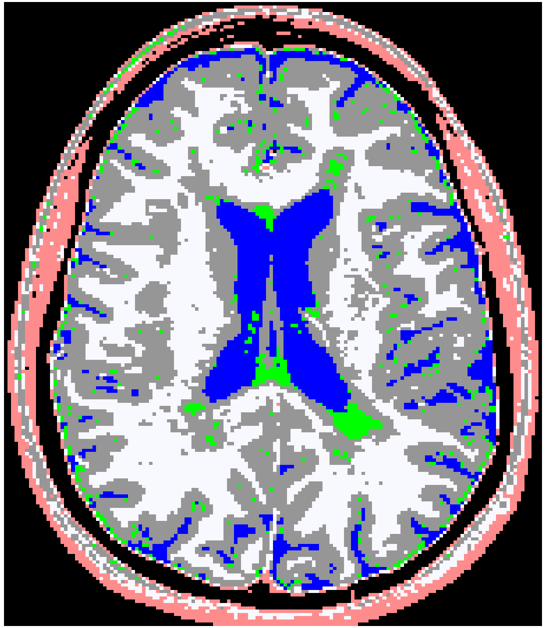
> MRF
> CRF

Classification of Voxels

- Intuitively, we associate different voxels to different structures, depending on their intensity values



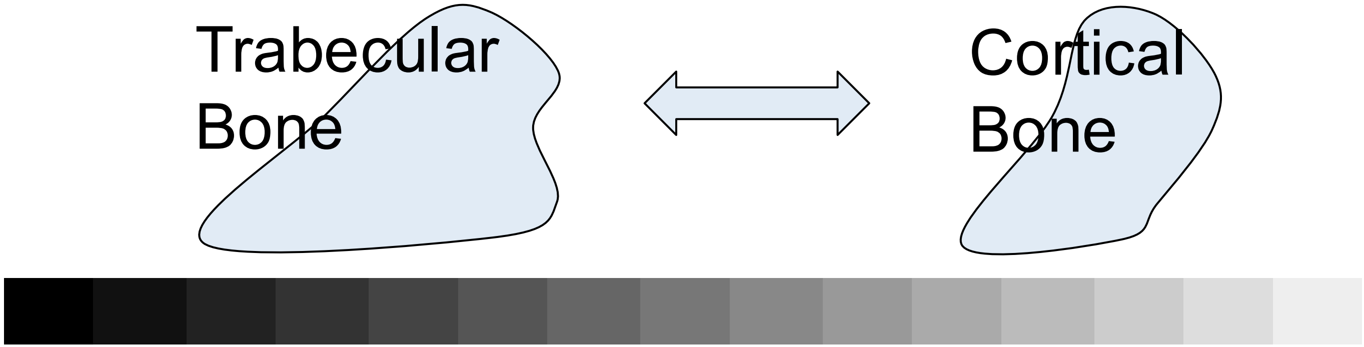
T2w



Segmentation

Classification of voxels

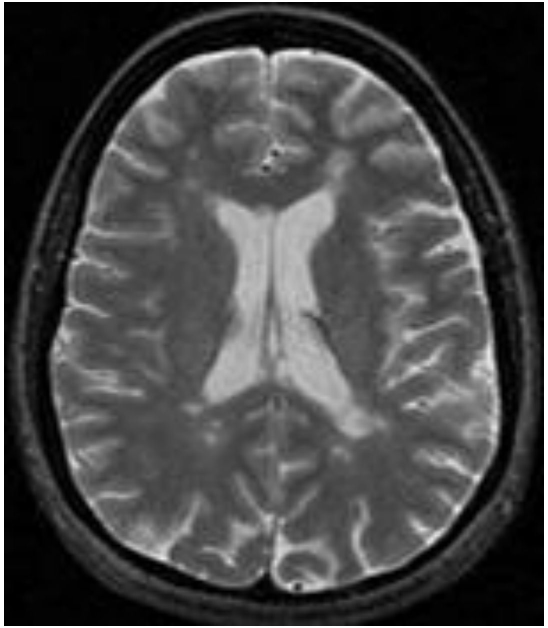
> Less intuitively, we group voxels trying to maximize the "distance" between the groups



Computationally this can modeled through the **Image Histogram**

Image Histogram

> Image histogram: representation of the frequency of $P(i)$ intensity values i in the image.



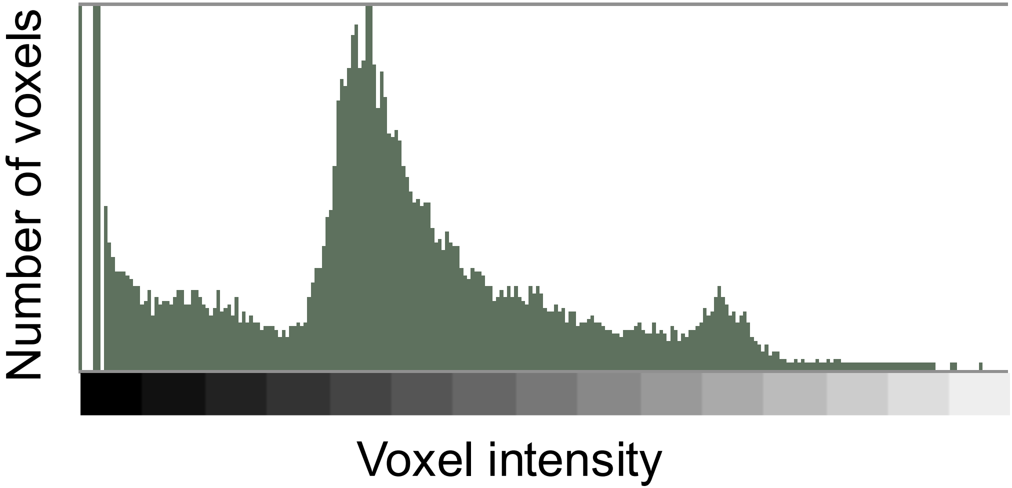
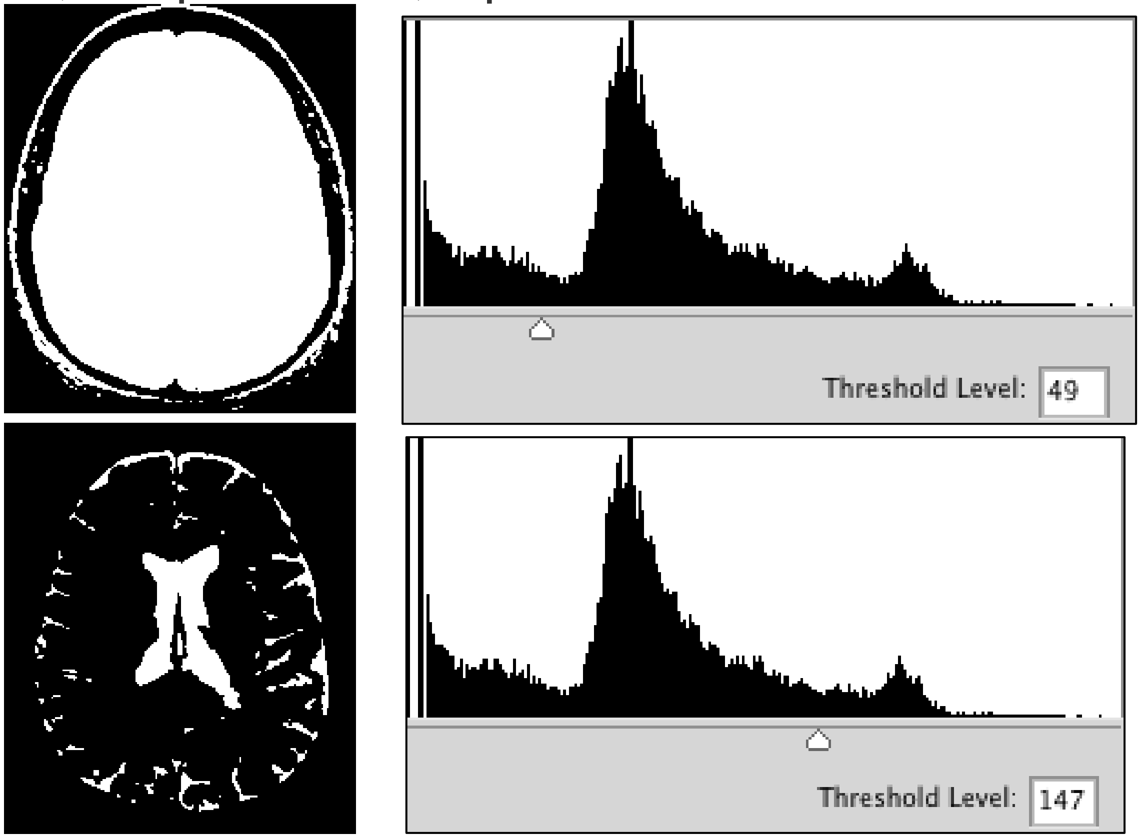


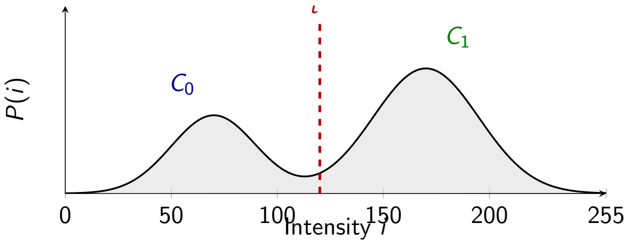
Image Thresholding

- The very basics of image segmentation
  - **Global**, adaptive/local, Optimal



Otsu Segmentation – Motivation

- Given an image histogram, how to **automatically** find the best threshold for binary segmentation?
- Manual threshold selection is subjective and not reproducible
- Otsu (1979): choose t^* that maximizes the **between-class variance** $\sigma_B^2(t)$



Otsu Segmentation – Formulation

- Let the image have L intensity levels $\{0, 1, \ldots, L-1\}$
- Normalized histogram: $p_i = n_i / N$, where $N =$ total pixels
- Choose threshold t to partition pixels into two classes:

$$C_0 = \{0, 1, \ldots, t\} \quad \text{and} \quad C_1 = \{t+1, \ldots, L-1\}$$

Class probabilities:

$$\omega_0(t) = \sum_{i=0}^{t} p_i, \qquad \omega_1(t) = \sum_{i=t+1}^{L-1} p_i = 1 - \omega_0(t)$$

Class means:

$$\mu_0(t) = \frac{1}{\omega_0(t)} \sum_{i=0}^{t} i \cdot p_i, \qquad \mu_1(t) = \frac{1}{\omega_1(t)} \sum_{i=t+1}^{L-1} i \cdot p_i$$

Otsu Segmentation – Variance Decomposition

Key insight: Total variance = within-class + between-class variance

$$\sigma^2_{\text{total}} = \sigma^2_W(t) + \sigma^2_B(t)$$

Within-class variance:

$$\sigma^2_W(t) = \omega_0(t)\,\sigma^2_0(t) + \omega_1(t)\,\sigma^2_1(t)$$

Between-class variance:

$$\boxed{\sigma^2_B(t) = \omega_0(t)\,\omega_1(t)\,\left[\mu_0(t) - \mu_1(t)\right]^2}$$

- Since σ^2_{total} is constant, minimizing $\sigma^2_W \iff$ maximizing σ^2_B
- $\sigma^2_B(t)$ is **much cheaper** to compute (no per-pixel variance needed)

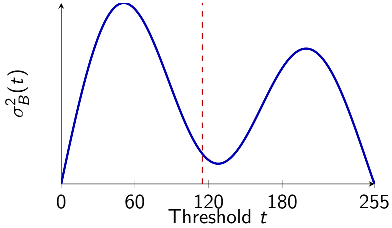
$$t^* = \arg\max_{0<t<L-1} \sigma^2_B(t)$$

Otsu Segmentation – Algorithm

1. Compute the normalized histogram p_i
2. For each candidate threshold t:
   - Compute $\omega_0(t)$, $\omega_1(t)$
   - Compute $\mu_0(t)$, $\mu_1(t)$
   - Compute $\sigma_B^2(t)$
3. Select $t^* = \arg\max_t \ \sigma_B^2(t)$
4. Segment:

$$g(x, y) = \begin{cases} 1 & \text{if } f(x, y) > t^* \\ 0 & \text{otherwise} \end{cases}$$

Complexity: $O(L)$ – single pass using cumulative sums

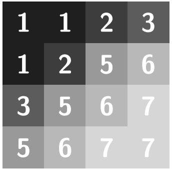


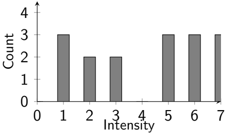
Between-class variance curve

Peak = optimal threshold

Otsu Segmentation – Worked Example

Consider a 4×4 image with $L = 8$ intensity levels:





Evaluating $\sigma_B^2(t)$ for each t:

t	1	2	3	4	5	6
σ_B^2	1.69	3.57	5.14	6.53	4.08	0.95

$$\Longrightarrow \quad t^* = 4 \quad \text{separates low-intensity pixels } \{1,2,3\} \text{ from high } \{5,6,7\}$$

Otsu Segmentation – Strengths & Limitations

Strengths

- Fully automatic – no parameters
- Computationally efficient: $O(L)$
- Optimal for bimodal histograms
- Equivalent to 1D Fisher's discriminant analysis
- Widely implemented (OpenCV, scikit-image, MATLAB)

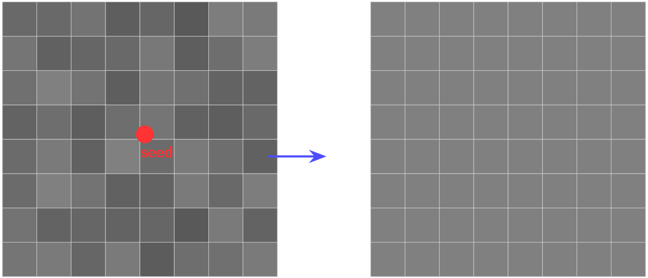
Limitations

- Assumes bimodal histogram
- Poor performance with uneven illumination
- Sensitive to noise
- No spatial information used
- Biased toward classes with larger variance

Variations: multi-level Otsu, 2D Otsu, local/adaptive Otsu, Kittler–Illingworth method

Region Growing – Motivation

- Thresholding uses only **intensity**, ignoring **spatial** relationships
- We want to group spatially connected pixels that share similar properties
- Idea: start from **seed points**, grow regions by appending similar neighbors



Original + seed

Segmented region

Region Growing – Formal Definition

Let R be the entire image region. A segmentation partitions R into n regions $R_1, R_2, \ldots, R_n$ such that:

(a) $\bigcup_{i=1}^{n} R_i = R$ (every pixel belongs to a region)

(b) R_i is a connected region for all i (spatial connectivity)

(c) $R_i \cap R_j = \emptyset$ for $i \neq j$ (no overlaps)

(d) $P(R_i) = \text{TRUE}$ for all i (homogeneity predicate)

(e) $P(R_i \cup R_j) = \text{FALSE}$ for adjacent R_i, R_j (distinct regions)

> $P(\cdot)$ is a **homogeneity predicate**, e.g.:

$$P(R_i) = \text{TRUE} \iff |I(x,y) - \mu_{R_i}| < \tau \ \ \forall (x,y) \in R_i$$

Region Growing – Basic Algorithm

1. **Select seed point(s)** s (manual or automatic)
2. Initialize region: $R = \{s\}$
3. For each unvisited neighbor q of R:
   - If similarity criterion is met: $|I(q) - \mu_R| < \tau$
   - Add q to R, update μ_R
4. Repeat until no more pixels can be added
5. Choose new seed (if unassigned pixels remain)

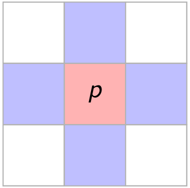
Similarity criteria: intensity difference, gradient magnitude, texture features, color distance, ...

42	48	88
44	47	91
43	**45**	87
41	44	92
40	45	95

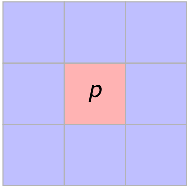
Seed Grown Candidate

Growing from seed (45) with $\tau = 5$. Right side (high intensity) is **not** included.

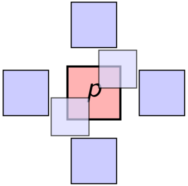
Region Growing – Neighborhood & Connectivity



4-connected



8-connected



6-connected (3D)

- **4-connected** (2D): shares edge → 4 neighbors
- **8-connected** (2D): shares edge or corner → 8 neighbors
- **6/18/26-connected** (3D): for volumetric data
- Choice of connectivity affects the shape of grown regions

Region Growing – Similarity Criteria

The choice of predicate P is **critical** to the quality of segmentation.

Common criteria:

- **Intensity difference from seed:**

$$|I(q) - I(s)| < \tau$$

- **Intensity difference from region mean** (adaptive):

$$|I(q) - \mu_R| < \tau, \qquad \mu_R = \frac{1}{|R|} \sum_{p \in R} I(p)$$

- **Statistical criterion:**

$$|I(q) - \mu_R| < k \cdot \sigma_R$$

  where σ_R is the standard deviation of intensities in R

- **Multi-feature:** combine intensity, gradient, texture

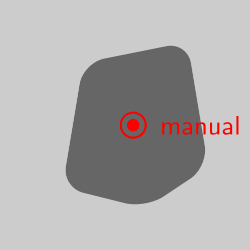
$$d(\mathbf{f}(q), \mathbf{f}_R) < \tau, \qquad \mathbf{f} = (I, |\nabla I|, \text{texture}, \ldots)^\top$$

Region Growing – Seed Selection Strategies

Seed selection significantly influences results.

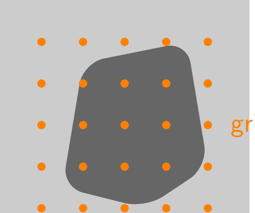
Manual:

- User clicks on region of interest
- Common in medical imaging (e.g., 3D Slicer)



Automatic:

- Grid-based: seeds at regular spacing
- Histogram-based: select pixels near histogram peaks
- Edge-based: seeds placed away from edges
- Clustering: use k-means centroids as seeds



Region Growing – Step-by-Step Illustration

Iteration 1

52	48	45	43	80	85
50	46	44	42	82	88
51	47	**43**	44	79	90
53	49	45	46	83	87
55	51	48	47	81	92
54	52	50	49	84	91

Seed at pixel with intensity 43, $\tau = 10$

Iteration 3

52	48	45	43	80	85
50	46	44	42	82	88
51	47	43	44	79	90
53	49	45	46	83	87
55	51	48	47	81	92
54	52	50	49	84	91

Region grows to similar neighbors

Converged

52	48	45	43	80	85
50	46	44	42	82	88
51	47	43	44	79	90
53	49	45	46	83	87
55	51	48	47	81	92
54	52	50	49	84	91

Growth stops at intensity boundary

Region Growing – Advantages & Disadvantages

Advantages

> Simple concept, easy to implement

> Uses **spatial** information (connected regions)

> Can segment objects with irregular shapes

> Can handle multiple criteria simultaneously

> Results align well with perceived edges

Disadvantages

> Sensitive to **seed selection**

> Sensitive to **noise**

> Order-dependent: first region "dominates"

> Threshold τ must be chosen carefully

> **No global view**: local method

> Can leak through weak boundaries

Variation: Seeded vs. Unseeded Region Growing

Seeded Region Growing (SRG)

Adams & Bischof, 1994

- Seeds provided as input
- Pixels assigned to nearest region based on intensity difference from region mean
- Ordering: pixel with **smallest** $\delta(q) = |I(q) - \mu_{R_i}|$ is assigned first
- Deterministic given seeds

Unseeded Region Growing (URG)

Lin et al., 2000

- No explicit seed selection needed
- Start with single pixel, grow if $S(p, q) > T$
- If a pixel cannot be added, it becomes a new seed for another region
- Avoids seed selection bias
- Less user interaction required

Seeds given → Sorted boundary → Grow nearest → Final regions

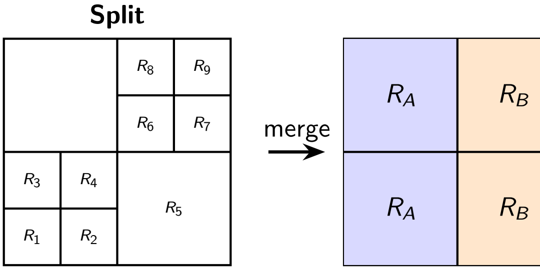
SRG

Variation: Region Splitting and Merging

Opposite approach to region growing:
Start with entire image as one region and **split** non-homogeneous parts.

1. If $P(R_i) = \text{FALSE}$, split R_i into 4 quadrants
2. Repeat recursively until all regions satisfy P
3. **Merge** adjacent regions R_i, R_j if $P(R_i \cup R_j) = \text{TRUE}$
4. Stop when no further splits or merges possible

Uses a **quadtree** data structure.



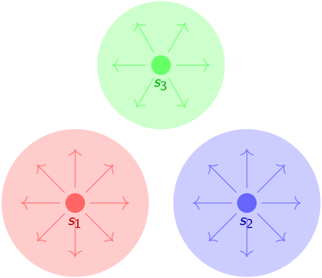
Variation: Simultaneous Region Growing

- **Problem:** Sequential growing biases toward the first region
- **Solution:** Grow all regions **simultaneously**

Approach:

- Multiple seeds placed at once
- Maintain a **priority queue** of boundary pixels sorted by similarity
- At each step, assign the globally most similar pixel to its closest region
- No single region dominates growth
- Efficient on parallel architectures

Resolves ambiguities at region boundaries more fairly than sequential methods.



Competitive growth

Variation: Adaptive Threshold Region Growing

- Fixed threshold τ may be too restrictive or too permissive in different parts of the image
- **Adaptive approach:** update τ as the region grows

Example: Statistical adaptive criterion

$$|I(q) - \mu_R^{(n)}| < k \cdot \sigma_R^{(n)}$$

where $\mu_R^{(n)}$ and $\sigma_R^{(n)}$ are updated after n pixels are added.

Gradient-based adaptation:

- Reduce τ near strong gradients (to avoid leaking)
- Increase τ in smooth homogeneous areas

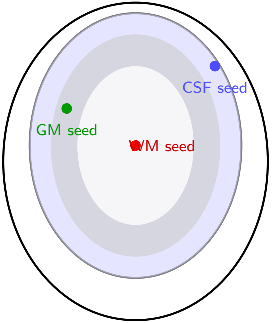
$$\tau(q) = \tau_0 \cdot \exp\left(-\frac{|\nabla I(q)|^2}{2\beta^2}\right)$$

This prevents region leakage through weak boundaries – a common failure mode of basic region growing.

Region Growing – Medical Imaging Applications

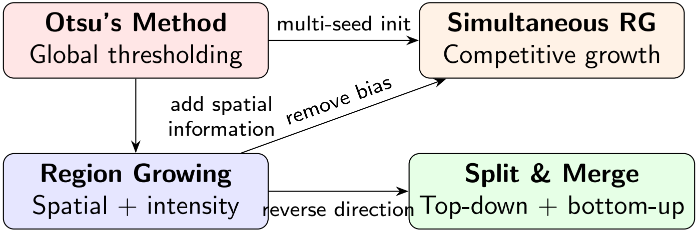
Region growing is widely used in medical image segmentation:

- **Brain MRI:** segment white matter, gray matter, CSF from T1/T2-weighted images
- **Vascular segmentation:** trace vessel trees in CT/MR angiography (branch-based strategies)
- **Tumor segmentation:** delineate lesion boundaries in CT/MRI
- **Bone segmentation:** separate cortical and trabecular bone in CT
- **Lung segmentation:** extract airways/parenchyma from chest CT



Brain tissue segmentation

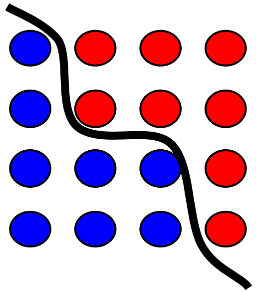
Summary



- **Otsu:** automatic, fast, but no spatial info – good for bimodal images
- **Region Growing:** incorporates spatial connectivity, flexible criteria, but sensitive to seeds and noise
- **Variations** (SRG, URG, split&merge, adaptive) address key limitations
- These methods form the **foundation** for more advanced approaches (deformable models, graph cuts, deep learning)

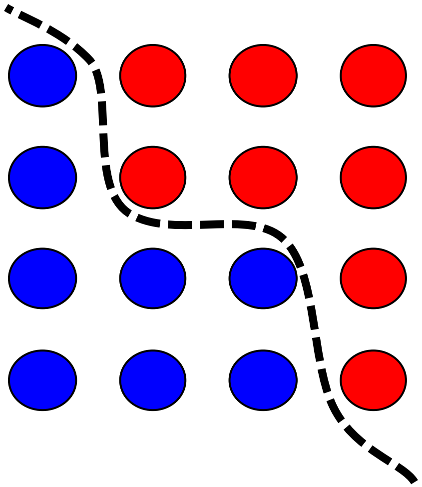
Live-wire segmentation
Barret et al. MedIA 1997

- Motivation: Lack of robustness with automatic methods, time-consuming manual segmentation
- Works on 2D (i.e. finds contours)
- Finds the optimal path between starting and ending point (seen as a graph searching problem)
- Optimal path = **minimal** cost between points = cumulative sum between local segments



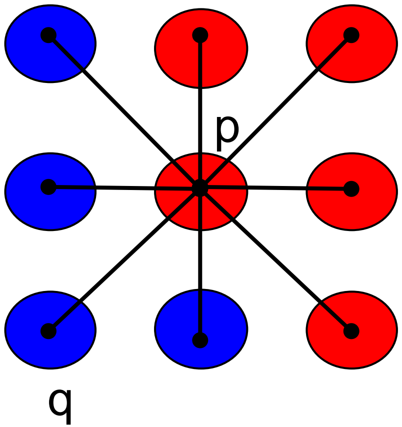
Live-wire segmentation

> Optimal path = **minimal** cost between group of nodes (pixels) = cumulative sum between local (pixel-wise) segments



Global graph

Set={Edges,Nodes}



Situation at one node

Cost between edges (also called links)

Live-wire segmentation

- Optimal path = **minimal** cost between nodes (pixels)
- Global cost = cumulative sum between local segments

Local cost l(p,q)

$$l(p,q) = \omega_G \cdot f_G(q) + \omega_Z \cdot f_Z(q) + \omega_D \cdot f_D(p,q)$$

f_**G** : Gradient information $f_G = 1 - \frac{G}{\max(G)}$.

f_z : Laplacian (L) information (2nd order derivatives)

f_z = 0 if L(I) =0 or neighbor(p) shows change of sign
f_z =1 otherwise

f_d : directional derivative

Live-wire segmentation

- Optimal path = **minimal** cost between nodes (pixels)
- Global cost = cumulative sum between local segments

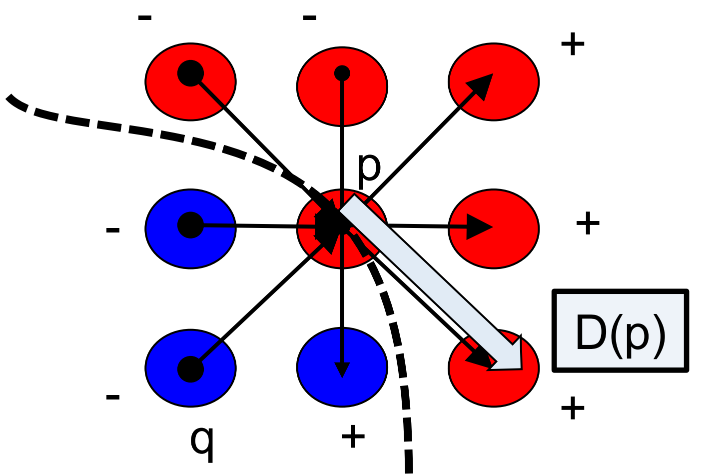
f_d : directional derivative

$$f_D(p,q) = \frac{2}{3\pi}\{\cos[d_p(p,q)]^{-1} + \cos[d_q(p,q)]^{-1}\}$$

$$d_p(p,q) = D(p)\cdot L(p,q)$$
$$d_q(p,q) = L(p,q)\cdot D(q)$$

D(p): normal to gradient direction at point p

$$L(p,q) = \begin{cases} q-p & \text{if} \quad D(p)\cdot(q-p) \geq 0 \\ p-q & \text{if} \quad D(p)\cdot(q-p) < 0 \end{cases}$$

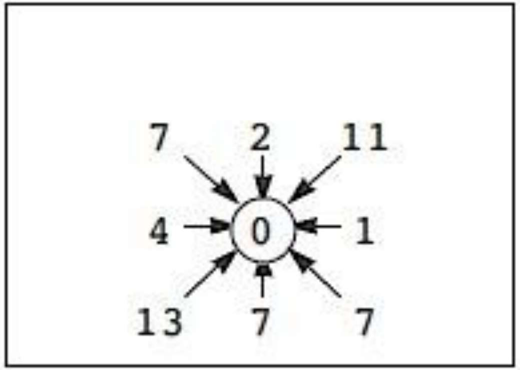


“pq vectors point in the direction of D minimizing the angle between them”

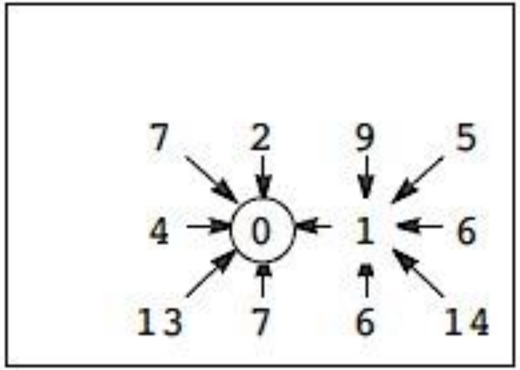
Live-wire segmentation

UNIVERSITÄT
BERN

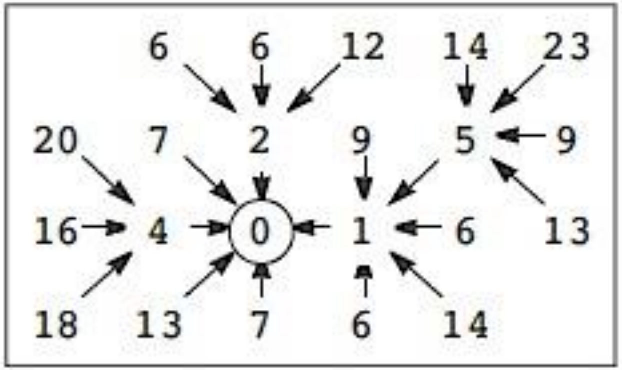
- Graph searching initialized by user “seed” (i.e. mouse click)
- Seed is expanded by computing local costs
- Minimal cost is used as next seed



Initial seed
(one expansion)



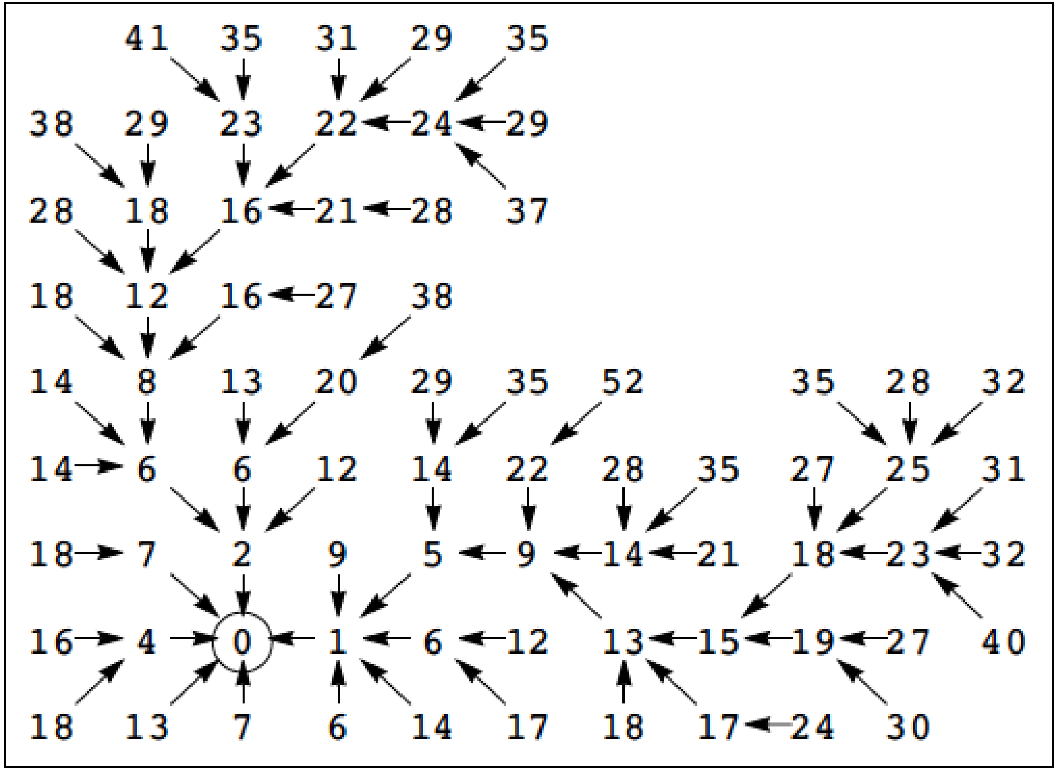
Two expansions



Five expansions

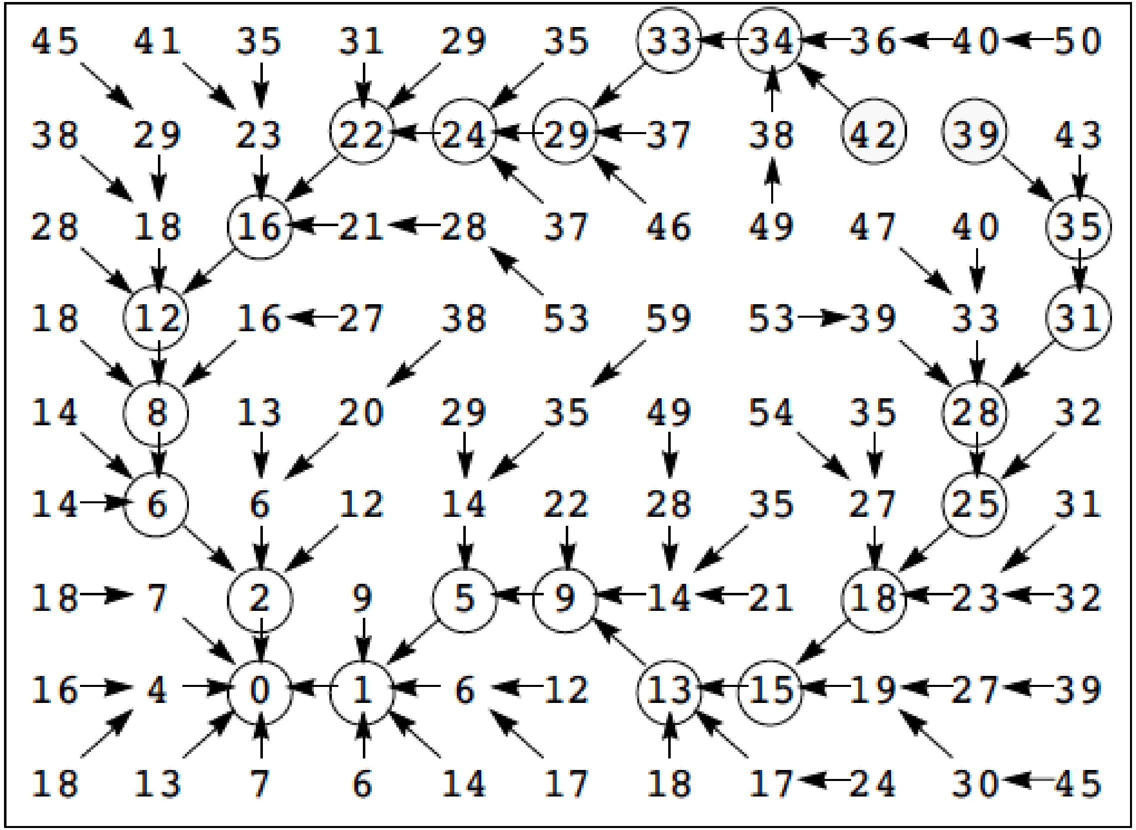
Live-wire segmentation

UNIVERSITÄT
BERN



47 points expanded

Live-wire segmentation



Completed total cost, with total costs of 42 and 39

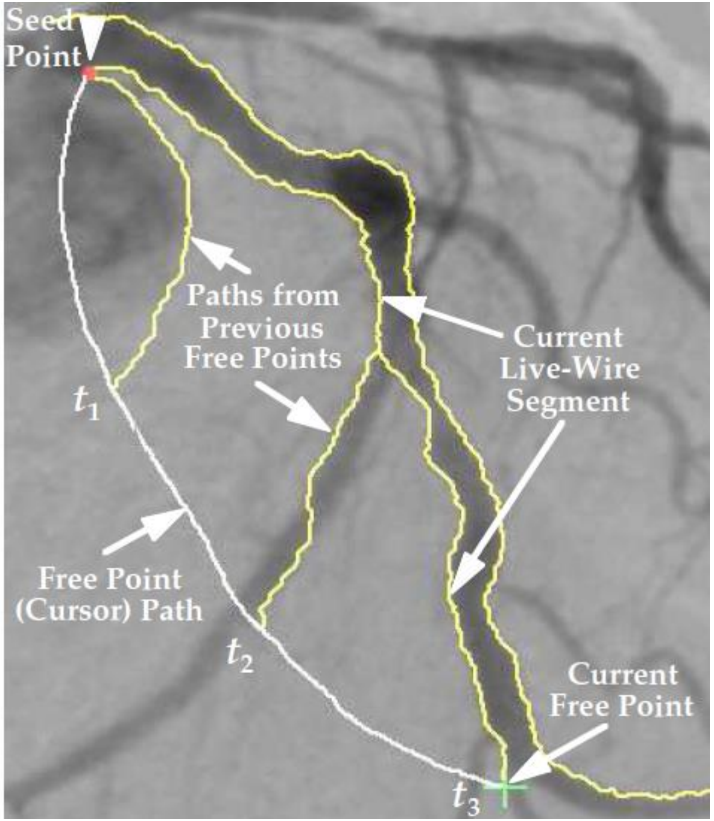
Live-wire segmentation

> Advantages of live-wire segmentation:
  - Monitor segmentation
  - Correct on the fly
  - No fix-path in advance
  - Speed over other path searching approaches. Enables real time

Live-wire segmentation

> Snap and wrap.

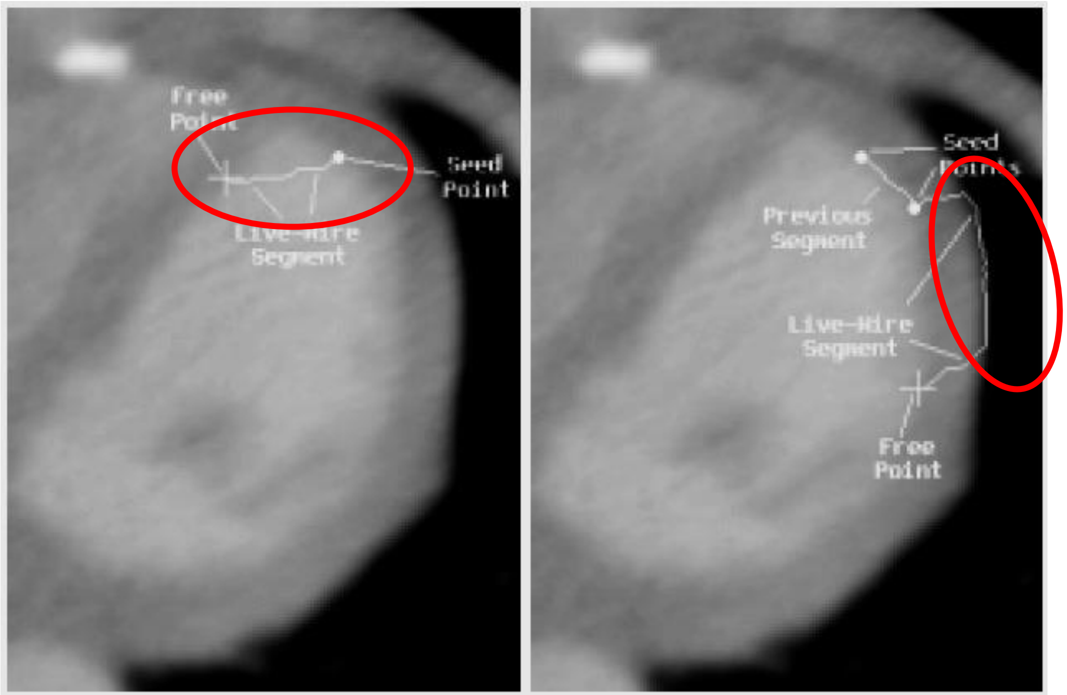
Dynamically changing “free points” elicit a new minimal cost path



Live-wire segmentation

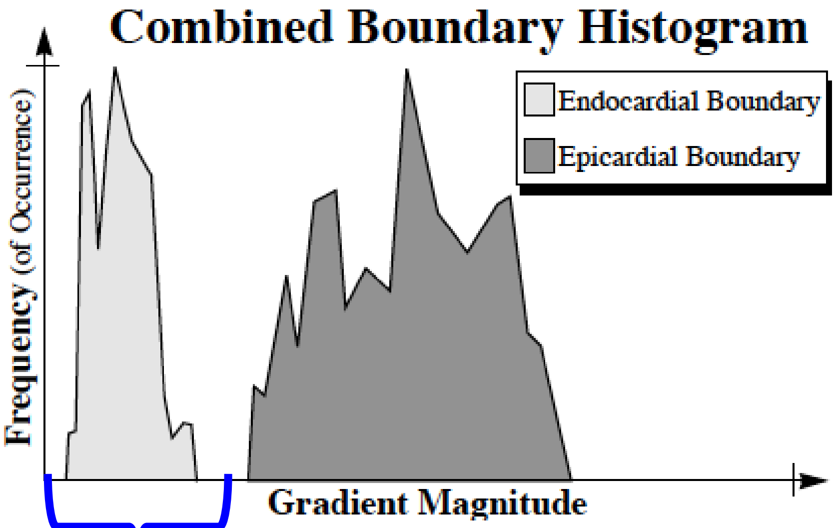
> On the fly training

Goal is to train the cost function to **fit to desired edges** rather than <u>strongest</u> one

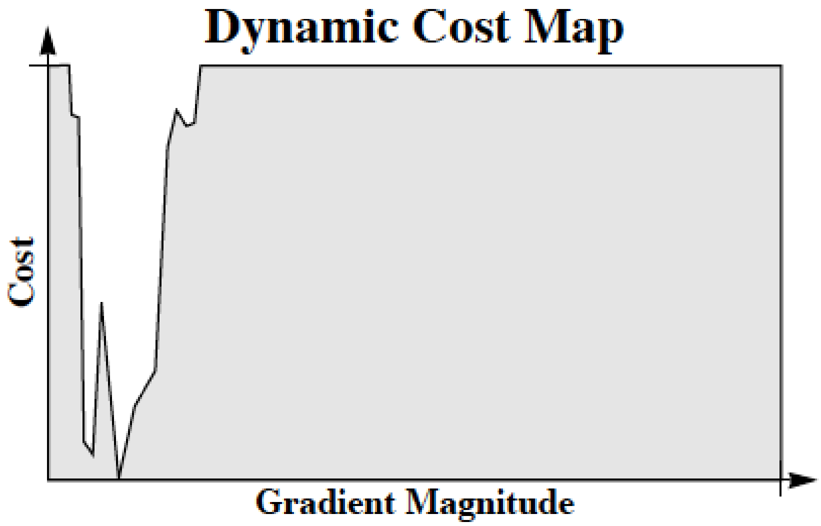


Live-wire segmentation

> On the fly training



Area of interest

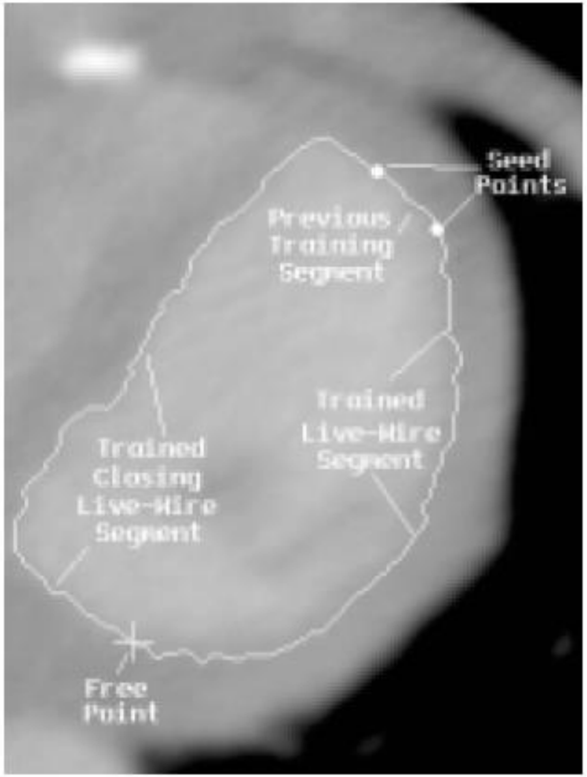


Associated gradient cost
Derived from current segment

Live-wire segmentation

> On the fly training

Resulting segmentation with on the fly training



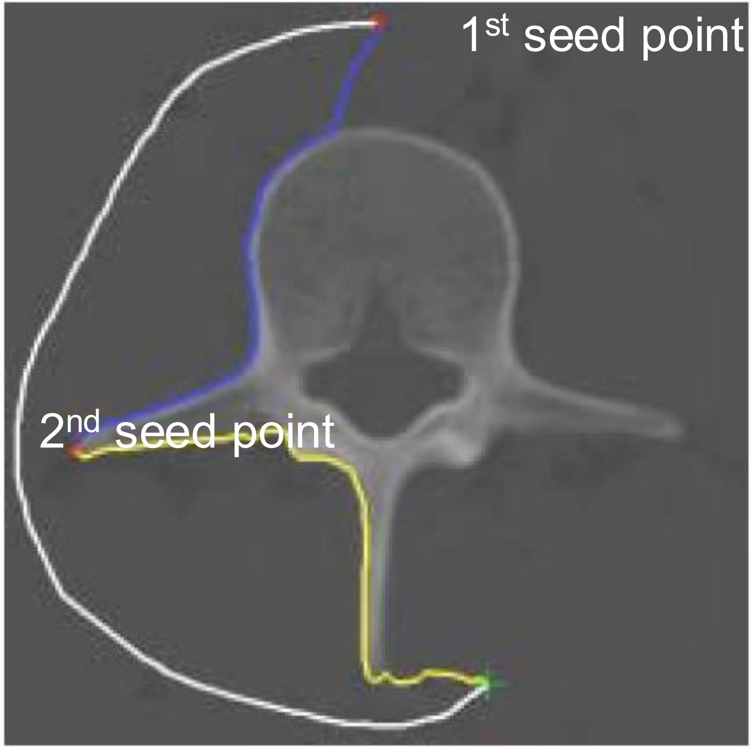
Live-wire segmentation

> Data-driven path cooling

Cooling: path between two fixed points (manual clicks)

Manual = ☹ => **data-driven** cooling!

Data-driven = the **longer** the segment remains “still” the more confidence it has, the cooler it becomes



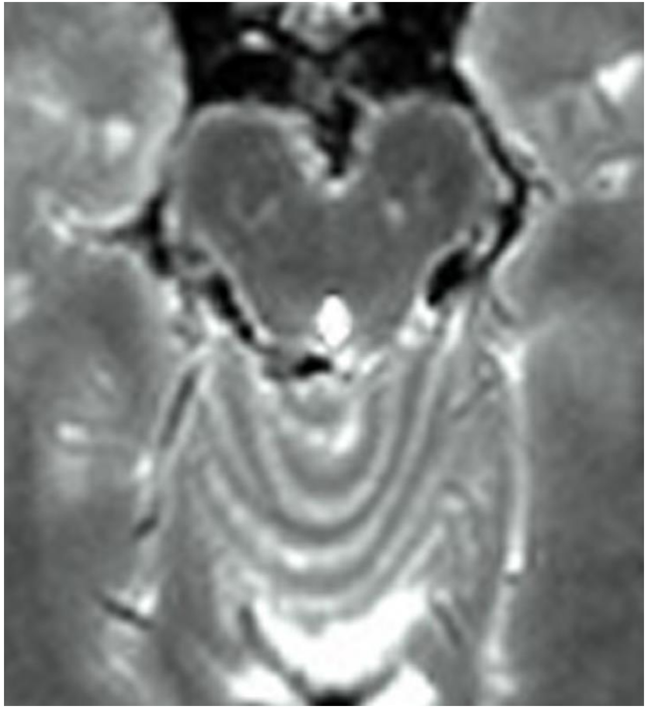
Evaluating segmentation

Validation of Image Segmentation

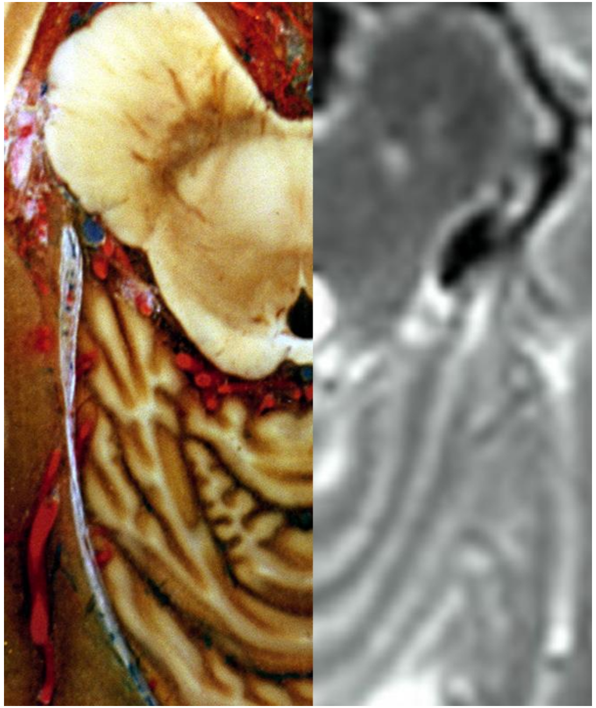
> Digital phantoms.
  - Ground truth known accurately.
  - Not so realistic.
> Acquisitions and careful segmentation.
  - Some uncertainty in ground truth.
  - More realistic.
> Autopsy/histopathology.
  - Addresses pathology directly; resolution
> Clinical data ?
  - Hard to know ground truth.
  - Most realistic model.

Source: S. Warfield

Comparison To Higher Resolution



MRI



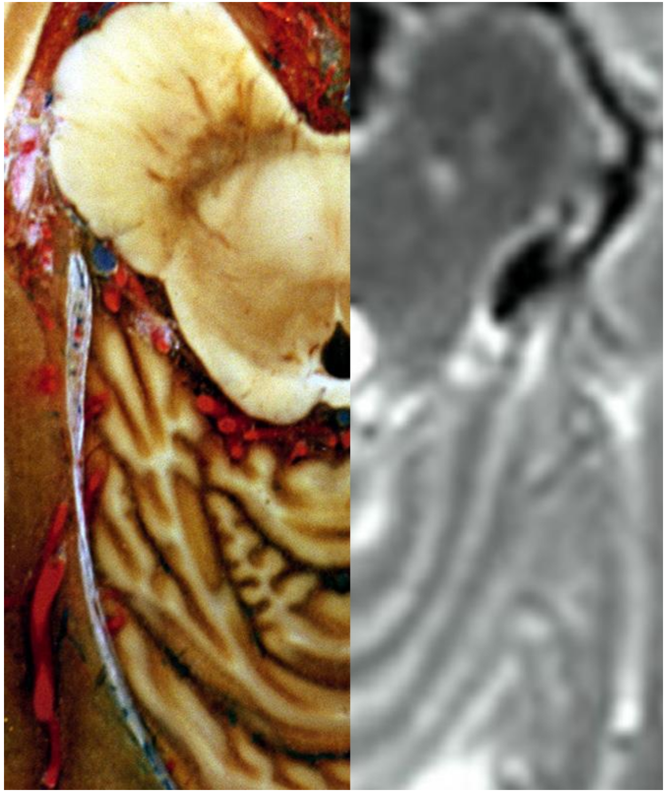
Photograph MRI

Source: S. Warfield

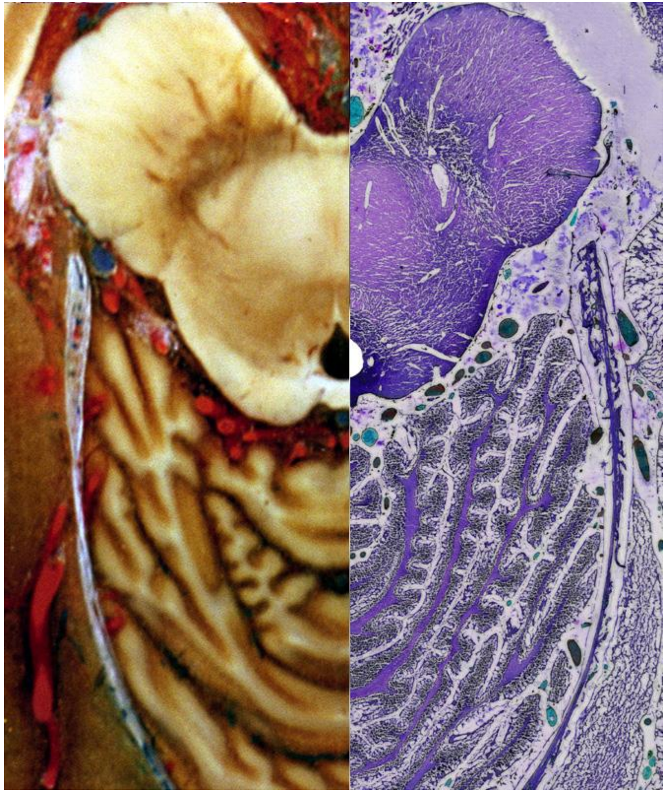
Provided by Peter Ratiu and Florin Talos.

Comparison To Higher Resolution

b UNIVERSITÄT BERN



Photograph MRI



Photograph Microscopy

Source: S. Warfield

Provided by Peter Ratiu and Florin Talos.

Measures of Expert Performance

> Repeated measures of volume
  — Intra-class correlation coefficient
> Spatial overlap
  — Jaccard coefficient: Area of intersection over union.
  — Dice coefficient: increased weight of intersection.
  — Vote counting: majority rule, etc.
> Boundary measures
  — Hausdorff, 95% Hausdorff.
> Bland-Altman methodology:
  — Requires a reference standard.
> Measures of correct classification rate:
  — Sensitivity, specificity ($Pr(D=1|T=1)$, $Pr(D=0|T=0)$)
  — Positive predictive value and negative predictive value (posterior probabilities $Pr(T=1|D=1)$, $Pr(T=0|D=0)$)

Measures of Expert Performance

> Measures of correct classification rate:
  - Sensitivity, specificity (Pr(D=1|T=1), Pr(D=0|T=0))
  - Positive predictive value and negative predictive value (posterior probabilities Pr(T=1|D=1), Pr(T=0|D=0))

$$\text{sensitivity} = \frac{\text{number of true positives}}{\text{number of true positives} + \text{number of false negatives}}$$

true positive (TP)
eqv. with hit
true negative (TN)
eqv. with correct rejection
false positive (FP)
eqv. with false alarm, Type I error
false negative (FN)
eqv. with miss, Type II error

References

- N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," IEEE Trans. Systems, Man, and Cybernetics, 1979.
- R. Adams & L. Bischof, "Seeded Region Growing," IEEE Trans. PAMI, 16(6), 1994.
- A. Mehnert & P. Jackway, "An Improved Seeded Region Growing Algorithm," Pattern Recognition Letters, 18(10), 1997.
- Z. Lin, J. Jin, H. Talbot, "Unseeded Region Growing for 3D Image Segmentation," VIP, 2000.
- R.C. Gonzalez & R.E. Woods, Digital Image Processing, 4th ed., Pearson, 2018.
- C. Revol-Muller et al., "Region Growing: When Simplicity Meets Theory," Springer LNCS, 2013.