Random Walks for Image Segmentation

Stefan Steidl
June 24, 2013
Computer Science Dept. 5 (Pattern Recognition)
Friedrich-Alexander University Erlangen-Nuremberg

FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT

Random Walks for Image Segmentation

- Problem Statement
- Algorithm
- Properties
- Implementation
- Example

Problem Statement

K-way image segmentation

- User-defined seeds
- indicating regions of the image belonging to K objects

Problem Statement

K-way image segmentation

- User-defined seeds
- indicating regions of the image belonging to K objects

Random walk

- Labeling an unseeded pixel by resolving the question:

  What is the probability of a random walker starting at this pixel that it first reaches seed point k?

Problem Statement

K-way image segmentation

- User-defined seeds
- indicating regions of the image belonging to K objects

Random walk

- Labeling an unseeded pixel by resolving the question:

  What is the probability of a random walker starting at this pixel
  that it first reaches seed point k?

- Selecting the label of the most probable seed destination for each pixel

Problem Statement

K-way image segmentation

- User-defined seeds
- indicating regions of the image belonging to K objects

Random walk

- Labeling an unseeded pixel by resolving the question:

  What is the probability of a random walker starting at this pixel
  that it first reaches seed point k?

- Selecting the label of the most probable seed destination for each pixel
- Biasing the random walker to avoid crossing sharp intensity gradients

Problem Statement (cont.)

Image as discrete object

- Graph with a fixed number of vertices and edges
- Each node represents one pixel in the image
- Edges connect neighboring pixels: e. g. 4-connectivity (2D), 6-connectivity (3D), 8-connectivity (2D)
- Real-valued weight assigned to each edge representing the likelihood that a random walker will cross this edge

  weight of zero: the random walker may not move along that edge
- purely combinatorial operators:
  - no discretization
  - no discretization errors or ambiguities

Problem Statement (cont.)

Edge weights for adjacent pixels *i* and *j*

Gaussian weighting function

$$w_{ij} = \exp(-\beta(g_i - g_j)^2)$$

- g_i: image intensity at pixel i
- β: only free parameter!
- useful operation: prior normalization of the square gradients:

$$\forall e_{ij} \in E : \ (g_i - g_j)^2$$

- modification to handle color or general vector-valued data: $||g_i - g_j||^2$

Problem Statement (cont.)

Four mathematically equivalent ways

1. *"If a random walker leaving the pixel is most likely to first reach a seed bearing label s, assign the pixel to label s."*

Problem Statement (cont.)

Four mathematically equivalent ways

1. *“If a random walker leaving the pixel is most likely to first reach a seed bearing label s, assign the pixel to label s.”*
2. *“If the seeds are alternately replaced by grounds/unit voltage sources, assign the pixel to the label for which its seeds being ‘on’ produces the greatest electrical potential.”*

Problem Statement (cont.)

Four mathematically equivalent ways

1. *“If a random walker leaving the pixel is most likely to first reach a seed bearing label s, assign the pixel to label s.”*
2. *“If the seeds are alternately replaced by grounds/unit voltage sources, assign the pixel to the label for which its seeds being ‘on’ produces the greatest electrical potential.”*
3. *“Assign the pixel to the label for which its seeds have the largest effective conductance (i. e., smallest effective resistance) with the pixel.”*

Problem Statement (cont.)

Four mathematically equivalent ways

1. *"If a random walker leaving the pixel is most likely to first reach a seed bearing label s, assign the pixel to label s."*
2. *"If the seeds are alternately replaced by grounds/unit voltage sources, assign the pixel to the label for which its seeds being 'on' produces the greatest electrical potential."*
3. *"Assign the pixel to the label for which its seeds have the largest effective conductance (i. e., smallest effective resistance) with the pixel."*
4. *"If a 2-tree is drawn randomly from the graph (with probability given by the product of weights in the 2-tree), assign the pixel to the label for which the pixel is most likely to remain connected to."*

Algorithm

Combinatorial Laplacian matrix L

$$L_{ij} = \begin{cases} d_i & \text{if } i = j \\ -w_{ij} & \text{if } v_i \text{ and } v_j \text{ are adjacent nodes} \\ 0 & \text{otherwise} \end{cases}$$

where L_{ij} is indexed by vertices v_i and v_j.

$d_i = \sum w(e_{ij})$ for all edges e_{ij} incident on node v_i

Algorithm (cont.)

Example: pixels of an 4×4 image

	1	2	3	4
1	v_1	v_2	v_3	v_4
2	v_5	v_6	v_7	v_8
3	v_9	v_{10}	v_{11}	v_{12}
4	v_{13}	v_{14}	v_{15}	v_{16}

Algorithm (cont.)

Example: combinatorial Laplacian matrix L

$$L = \begin{bmatrix}
d_1 & -w_{1,2} & 0 & 0 & -w_{1,5} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-w_{1,2} & d_2 & -w_{2,3} & 0 & 0 & -w_{2,6} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -w_{2,3} & d_3 & -w_{3,4} & 0 & 0 & -w_{3,7} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -w_{3,4} & d_4 & 0 & 0 & 0 & -w_{4,8} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-w_{1,5} & 0 & 0 & 0 & d_5 & -w_{5,6} & 0 & 0 & -w_{5,9} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -w_{2,6} & 0 & 0 & -w_{5,6} & d_6 & -w_{6,7} & 0 & 0 & -w_{6,10} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -w_{3,7} & 0 & 0 & -w_{6,7} & d_7 & -w_{7,8} & 0 & 0 & -w_{7,11} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -w_{4,8} & 0 & 0 & -w_{7,8} & d_8 & 0 & 0 & 0 & -w_{8,12} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -w_{5,9} & 0 & 0 & 0 & d_9 & -w_{9,10} & 0 & 0 & -w_{9,13} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -w_{6,10} & 0 & 0 & -w_{9,10} & d_{10} & -w_{10,11} & 0 & 0 & -w_{10,14} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -w_{7,11} & 0 & 0 & -w_{10,11} & d_{11} & -w_{11,12} & 0 & 0 & -w_{11,15} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -w_{8,12} & 0 & 0 & -w_{11,12} & d_{12} & 0 & 0 & 0 & -w_{12,16} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -w_{9,13} & 0 & 0 & 0 & d_{13} & -w_{13,14} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -w_{10,14} & 0 & 0 & -w_{13,14} & d_{14} & -w_{14,15} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -w_{11,15} & 0 & 0 & -w_{14,15} & d_{15} & -w_{15,16} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -w_{12,16} & 0 & 0 & -w_{15,16} & d_{16}
\end{bmatrix}$$

Algorithm (cont.)

Combinatorial formulation of the Dirichlet integral

$$D(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T \boldsymbol{L}\boldsymbol{x} = \frac{1}{2}\sum_{e_{ij}\in E} w_{ij}(x_i - x_j)^2$$

Algorithm (cont.)

Combinatorial formulation of the Dirichlet integral

$$D(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T \boldsymbol{L}\boldsymbol{x} = \frac{1}{2}\sum_{e_{ij}\in E} w_{ij}(x_i - x_j)^2$$

Partitioning the vertices into two sets:

- marked/seed nodes V_M
- unseeded nodes V_U

such that $V_M \cup V_U = V$ and $V_M \cap V_U = \varnothing$.

Algorithm (cont.)

Combinatorial formulation of the Dirichlet integral

$$D(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T \boldsymbol{L}\boldsymbol{x} = \frac{1}{2}\sum_{e_{ij}\in E} w_{ij}(x_i - x_j)^2$$

Partitioning the vertices into two sets:

- marked/seed nodes V_M
- unseeded nodes V_U

such that $V_M \cup V_U = V$ and $V_M \cap V_U = \emptyset$.

Without loss of generality:

- the nodes in $\boldsymbol{L}$ and $\boldsymbol{x}$ are ordered: seed nodes are first, unseeded nodes are second.

Algorithm (cont.)

Decomposition:

$$\begin{aligned} D[\boldsymbol{x_U}] &= \frac{1}{2}\begin{bmatrix}\boldsymbol{x}_M^T & \boldsymbol{x}_U^T\end{bmatrix}\begin{bmatrix}\boldsymbol{L}_M & \boldsymbol{B} \\ \boldsymbol{B}^T & \boldsymbol{L}_U\end{bmatrix}\begin{bmatrix}\boldsymbol{x}_M \\ \boldsymbol{x}_U\end{bmatrix} \\ &= \frac{1}{2}\left(\boldsymbol{x}_M^T\boldsymbol{L}_M\boldsymbol{x}_M + 2\boldsymbol{x}_U^T B^T\boldsymbol{x}_M + \boldsymbol{x}_U^T\boldsymbol{L}_U\boldsymbol{x}_U\right)\end{aligned}$$

Algorithm (cont.)

Decomposition:

$$\begin{aligned} D[\boldsymbol{x_U}] &= \frac{1}{2}\begin{bmatrix}\boldsymbol{x}_M^T & \boldsymbol{x}_U^T\end{bmatrix}\begin{bmatrix}\boldsymbol{L}_M & \boldsymbol{B} \\ \boldsymbol{B}^T & \boldsymbol{L}_U\end{bmatrix}\begin{bmatrix}\boldsymbol{x}_M \\ \boldsymbol{x}_U\end{bmatrix} \\ &= \frac{1}{2}\left(\boldsymbol{x}_M^T\boldsymbol{L}_M\boldsymbol{x}_M + 2\boldsymbol{x}_U^T B^T\boldsymbol{x}_M + \boldsymbol{x}_U^T\boldsymbol{L}_U\boldsymbol{x}_U\right) \end{aligned}$$

$\boldsymbol{L}$ is positive semi-definite: the only critical points of $D[\boldsymbol{x}]$ will be minima.

Algorithm (cont.)

Decomposition:

$$
\begin{aligned}
D[\boldsymbol{x_U}] &= \frac{1}{2}\begin{bmatrix}\boldsymbol{x}_M^T & \boldsymbol{x}_U^T\end{bmatrix}\begin{bmatrix}\boldsymbol{L}_M & \boldsymbol{B}\\ \boldsymbol{B}^T & \boldsymbol{L}_U\end{bmatrix}\begin{bmatrix}\boldsymbol{x}_M\\ \boldsymbol{x}_U\end{bmatrix}\\
&= \frac{1}{2}\left(\boldsymbol{x}_M^T\boldsymbol{L}_M\boldsymbol{x}_M + 2\boldsymbol{x}_U^T B^T\boldsymbol{x}_M + \boldsymbol{x}_U^T\boldsymbol{L}_U\boldsymbol{x}_U\right)
\end{aligned}
$$

$\boldsymbol{L}$ is positive semi-definite: the only critical points of $D[\boldsymbol{x}]$ will be minima.

Differentiating w. r. t. $\boldsymbol{x}_U$ and finding the critical points:

$$\boldsymbol{L}_U\boldsymbol{x}_U = -\boldsymbol{B}^T\boldsymbol{x}_M$$

Algorithm (cont.)

$$\boldsymbol{L}_U \boldsymbol{x}_U = -\boldsymbol{B}^T \boldsymbol{x}_M$$

- System of linear equations with $|V_U|$ unknowns
- Equation will be non-singular
  - if the graph is connected, or
  - if every connected component contains a seed

Algorithm (cont.)

Solution to the combinatorial Dirichlet problem for label s:

- x_i^s: probability (potential) assumed at node v_i for label s
- Set of labels: $\forall v_j \in V_M : \; Q(v_j) = s, \quad s \in \mathbb{Z}, 0 < s \leq K$
- $V_M \times 1$ vector $\boldsymbol{m}^s$:

$$m_j^s = \begin{cases} 1 & \text{if } Q(v_j) = s \\ 0 & \text{if } Q(v_j) \neq s \end{cases}$$

Algorithm (cont.)

Solution to the combinatorial Dirichlet problem for label s:

- x_i^s: probability (potential) assumed at node v_i for label s
- Set of labels: $\forall v_j \in V_M : \; Q(v_j) = s, \quad s \in \mathbb{Z}, 0 < s \leq K$
- $V_M \times 1$ vector $\boldsymbol{m}^s$:

$$m_j^s = \begin{cases} 1 & \text{if } Q(v_j) = s \\ 0 & \text{if } Q(v_j) \neq s \end{cases}$$

Solution for one label:

$$\boldsymbol{L}_U \boldsymbol{x}^s = -\boldsymbol{B}^T \boldsymbol{m}^s$$

Algorithm (cont.)

Solution to the combinatorial Dirichlet problem for label s:

- x_i^s: probability (potential) assumed at node v_i for label s
- Set of labels: $\forall v_j \in V_M: \; Q(v_j) = s, \quad s \in \mathbb{Z}, 0 < s \leq K$
- $V_M \times 1$ vector $\boldsymbol{m}^s$:

$$m_j^s = \begin{cases} 1 & \text{if } Q(v_j) = s \\ 0 & \text{if } Q(v_j) \neq s \end{cases}$$

Solution for one label:

$$\boldsymbol{L}_U \boldsymbol{x}^s = -\boldsymbol{B}^T \boldsymbol{m}^s$$

Solution for all labels:

$$\boldsymbol{L}_U \boldsymbol{X} = -\boldsymbol{B}^T \boldsymbol{M}$$

$\boldsymbol{X}$, $\boldsymbol{M}$: matrix with K columns taken by each $\boldsymbol{x}^s$ and $\boldsymbol{m}^s$, respectively

Algorithm (cont.)

Note:

- At any node the probabilities x^s will sum to unity:

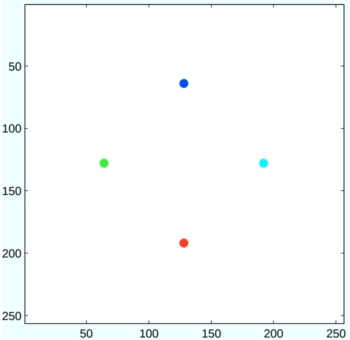
$$\forall v_i \in V : \sum_s x_i^s = 1$$

- Hence, only $K - 1$ sparse linear systems must be solved.

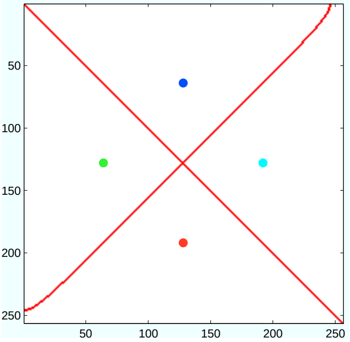
Properties

Neutral segmentation: corresponds roughly to Voronoi cells (1)

Original image with seed points



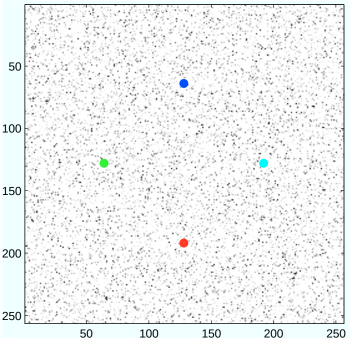
Outlined mask



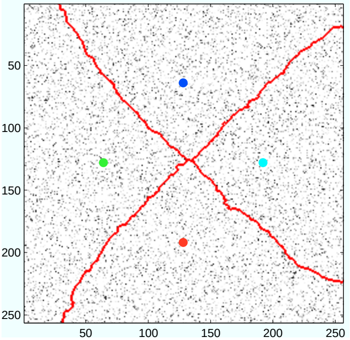
Properties (cont.)

Neutral segmentation: corresponds roughly to Voronoi cells (2)

Original image with seed points



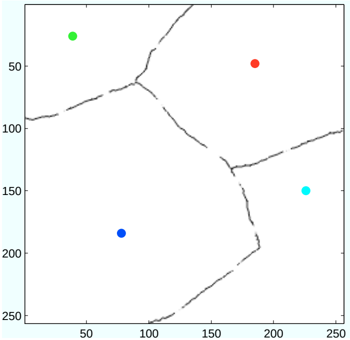
Outlined mask



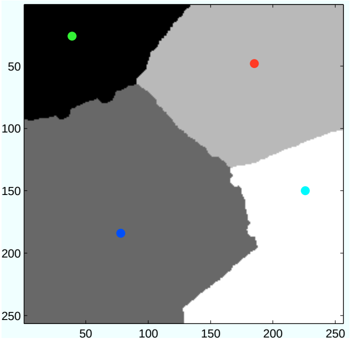
Properties (cont.)

Weak boundaries (1)

Original image with seed points

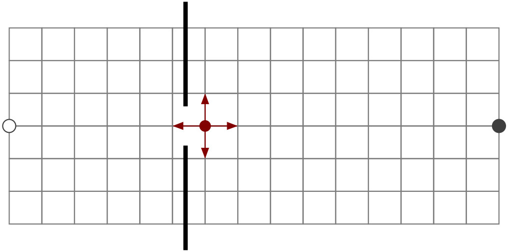


Output mask



Properties (cont.)

Weak boundaries (2)



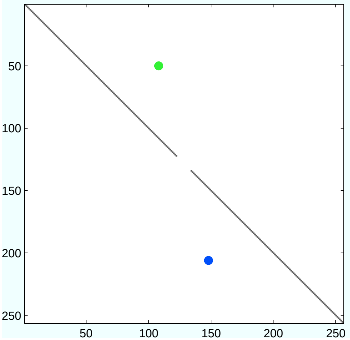
- On its initial step, the current pixel has 3 out of 4 chances to enter into the region that is likely to be labeled as belonging to the black circle.
- On the other side of the weak boundary, the same holds for the white circle.
- Due to the sharp drop in the probabilities, the segmentation will respect the weak boundary.

Properties (cont.)

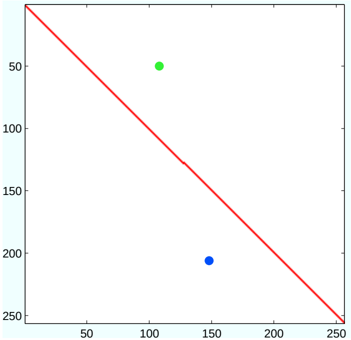
Properties (cont.)

Weak boundaries (3)

Original image with seed points



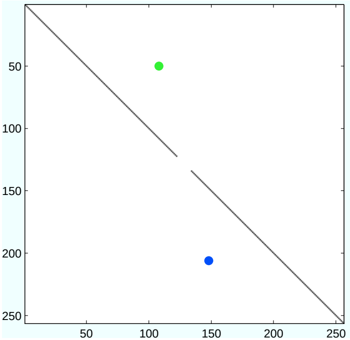
Outlined mask



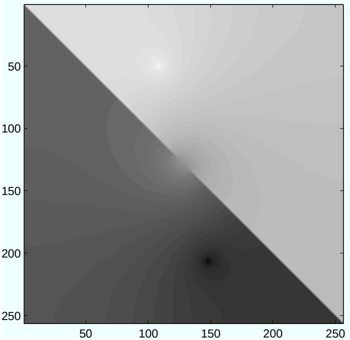
Properties (cont.)

Weak boundaries (4)

Original image with seed points



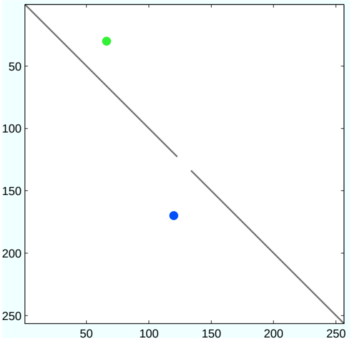
Probabilities for reaching seed 1



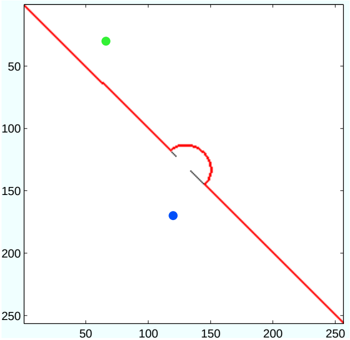
Properties (cont.)

Weak boundaries (5)

Original image with seed points



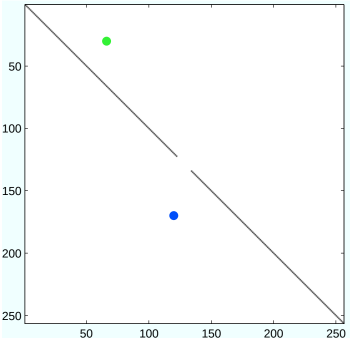
Outlined mask



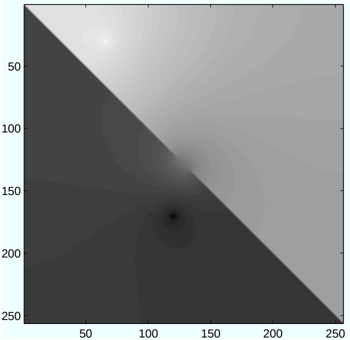
Properties (cont.)

Weak boundaries (6)

Original image with seed points

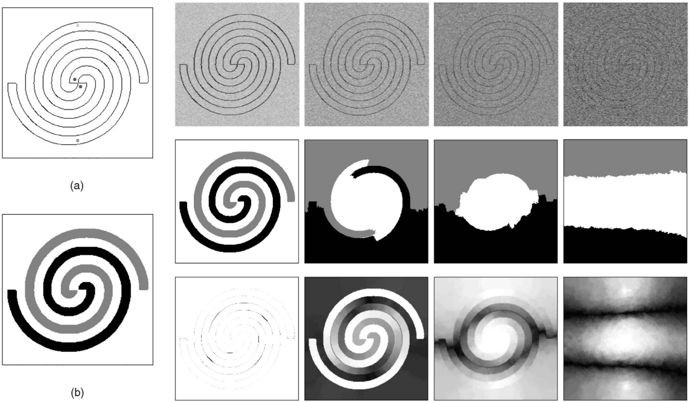


Probabilities for reaching seed 1



Properties (cont.)

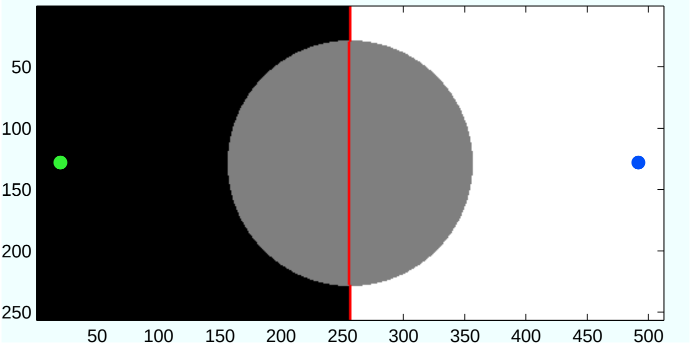
Noise robustness



Properties (cont.)

Ambiguous unseeded regions:

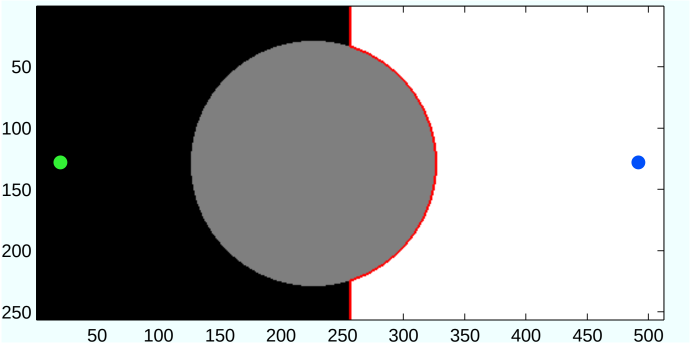
- centered precisely with respect to surface area and intensity



Properties (cont.)

Ambiguous unseeded regions:

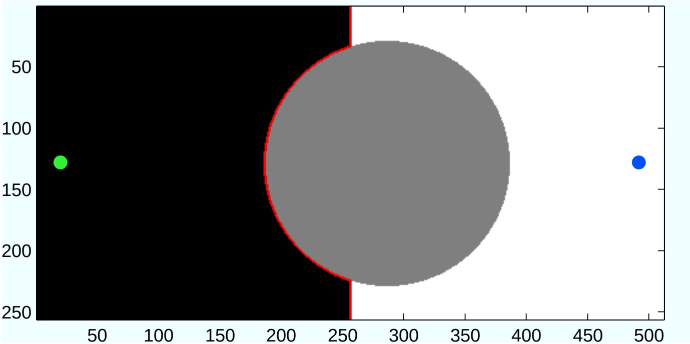
- sharing more surface area with black region



Properties (cont.)

Ambiguous unseeded regions:

- sharing more surface area with white region

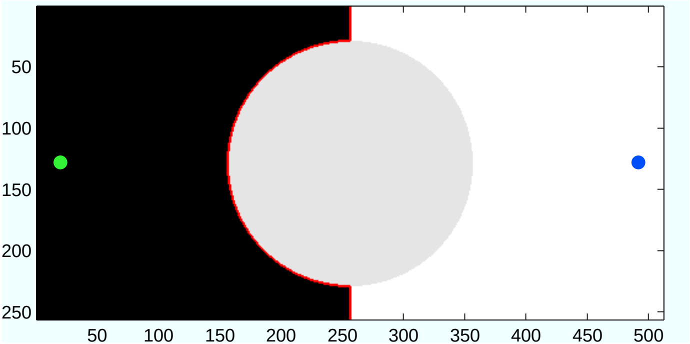


FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT

Properties (cont.)

Ambiguous unseeded regions:

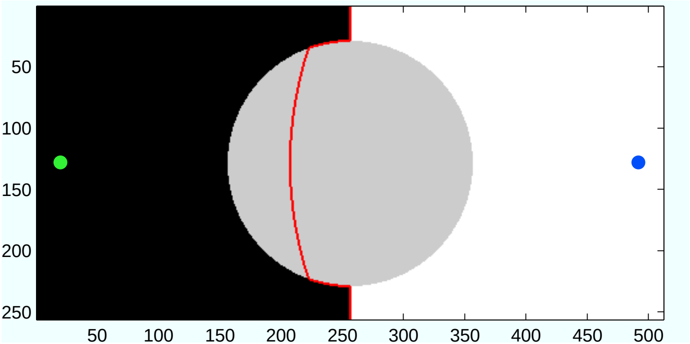
- closer in intensity to the white region (gray value 0.9)



Properties (cont.)

Ambiguous unseeded regions:

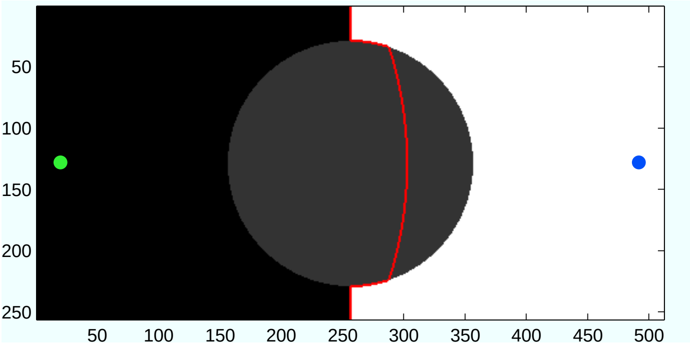
- closer in intensity to the white region (gray value 0.8)



Properties (cont.)

Ambiguous unseeded regions:

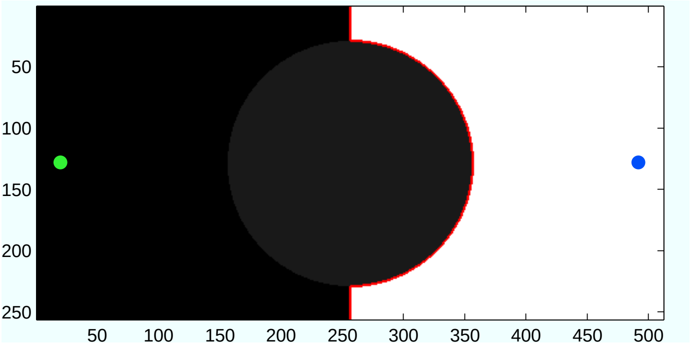
- closer in intensity to the white region (gray value 0.2)



Properties (cont.)

Ambiguous unseeded regions:

- closer in intensity to the white region (gray value 0.1)



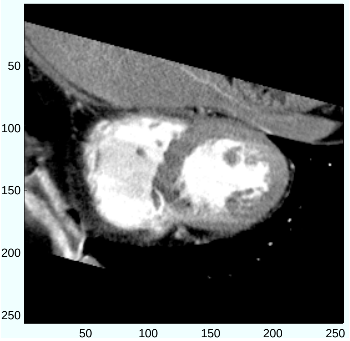
MATLAB Implementation

Implementation

- Graph Analysis Toolbox available for MATLAB to
  - easily build weighted image graphs
  - solve requisite system of linear equations
- Specialty code to perform the random walker segmentation
  - recommended for research purposes
  - sufficient for 512×512 images
  - more industrial use requires C++ implementation of conjugate gradients or multigrid code

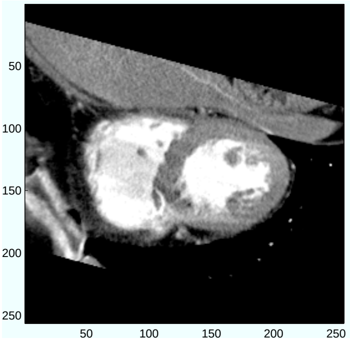
Example: axial CT slice

Original image

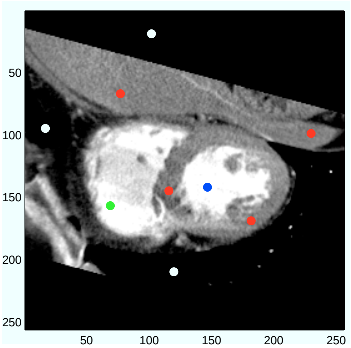


Example: axial CT slice

Original image

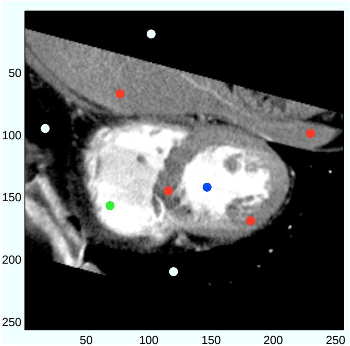


Original image with seed points



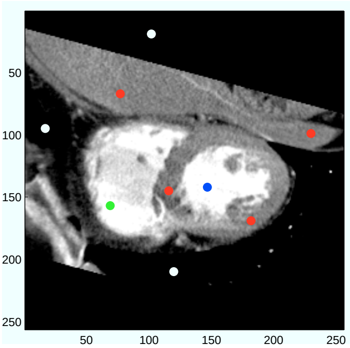
Example: axial CT slice

Original image with seed points

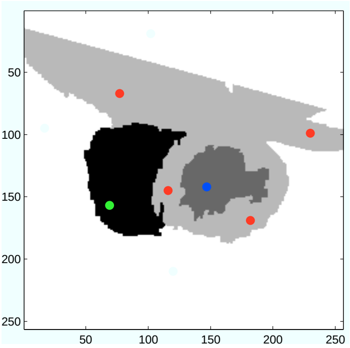


Example: axial CT slice

Original image with seed points

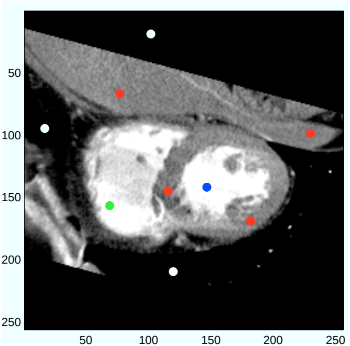


Output mask

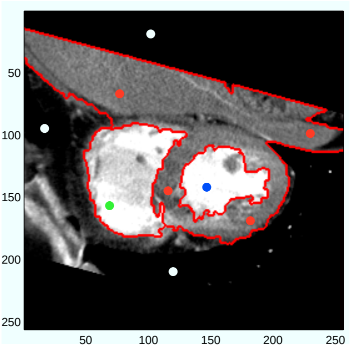


Example: axial CT slice

Original image with seed points

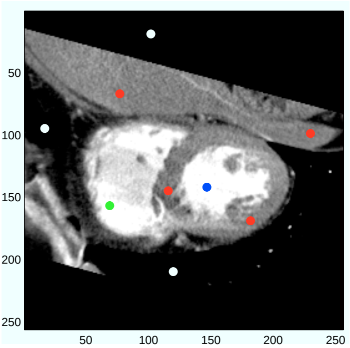


Outlined mask

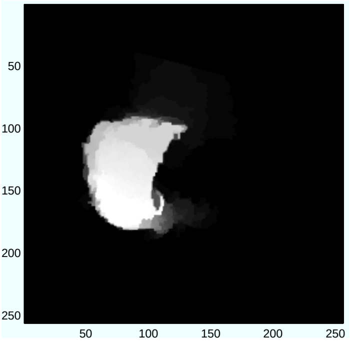


Example: axial CT slice

Original image with seed points

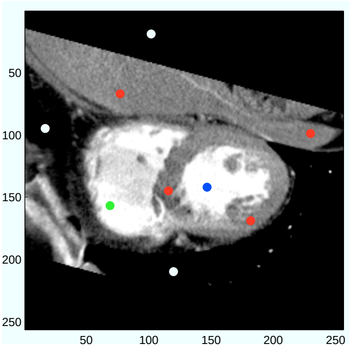


Probabilities for reaching seed 1

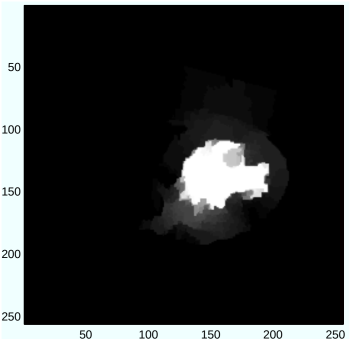


Example: axial CT slice

Original image with seed points

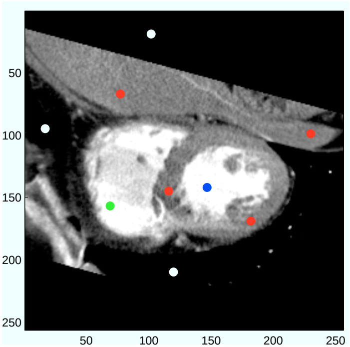


Probabilities for reaching seed 2

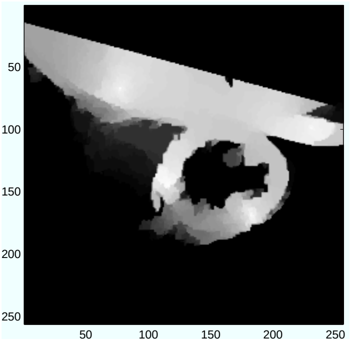


Example: axial CT slice

Original image with seed points

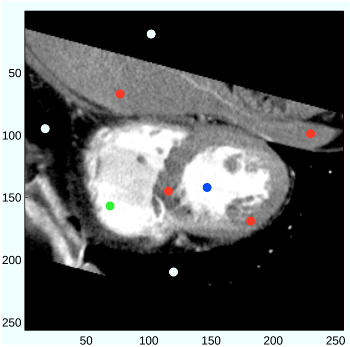


Probabilities for reaching seed 3

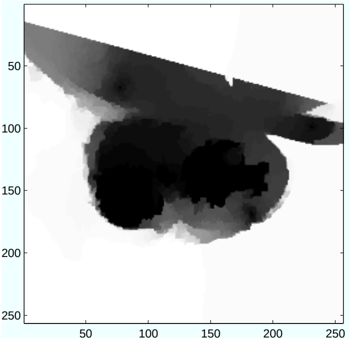


Example: axial CT slice

Original image with seed points



Probabilities for reaching seed 4



Conclusion

Random Walker

- β is the only free parameter
- Solution to a sparse, symmetric, positive-definite system of equations
- Straightforward implementation
- Efficient performance
- Interactive editing: previous solution as an initial solution for an iterative matrix solver
- Segments are guaranteed to be connected
- Noise robustness
- No discretization errors
- No variations in implementation

Literature

These slides are based on the following publication:

- Leo Grady: ☞ Random Walks for Image Segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 28, No. 11, Nov. 2006

MATLAB implementation:

- ☞ Graph Analysis Toolbox
- ☞ Random Walker

KEEP WALKING

Johnnie Walker®