Computer Vision, Speech Communication & Signal Processing Group,
National Technical University of Athens, Greece (NTUA)
Robotic Perception and Interaction Unit,
Athena Research and Innovation Center (Athena RIC)

Graph-theoretic Approaches to Segmentation based on Active Contours and Random Walk Schemes on Arbitrary Graphs

Petros Maragos

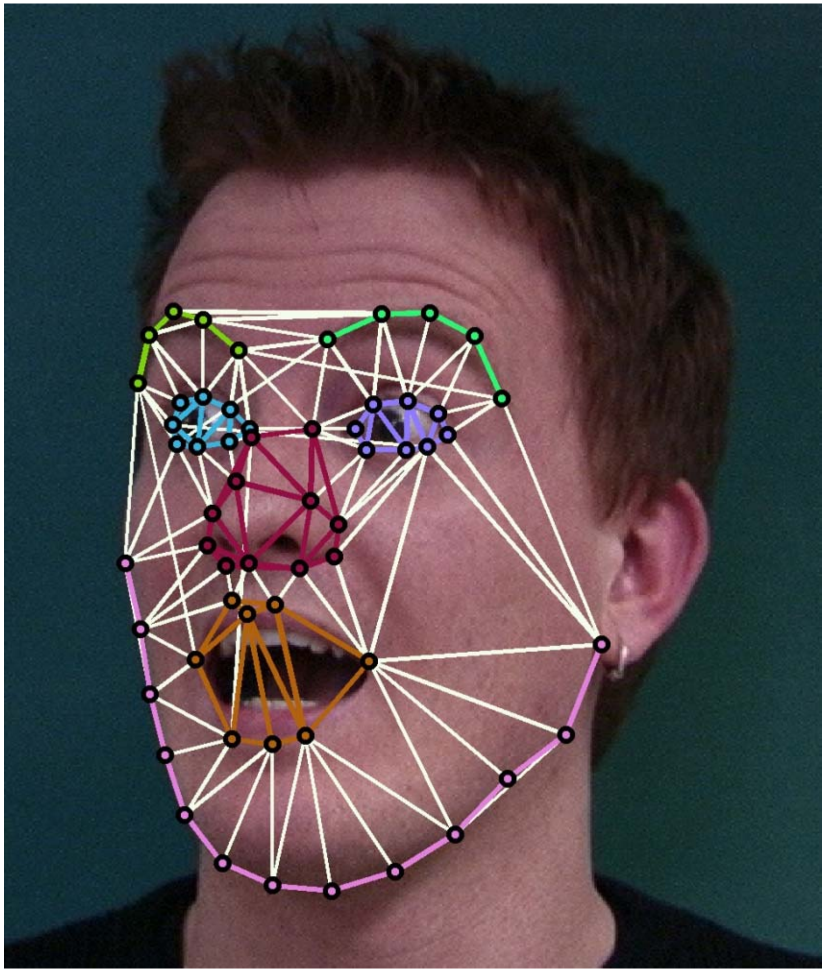
Aristotle University of Thessaloniki - Dept. of Informatics, 31 May 2018,
Distinguished Lecturer Series "Leon the Mathematician".

Motivations

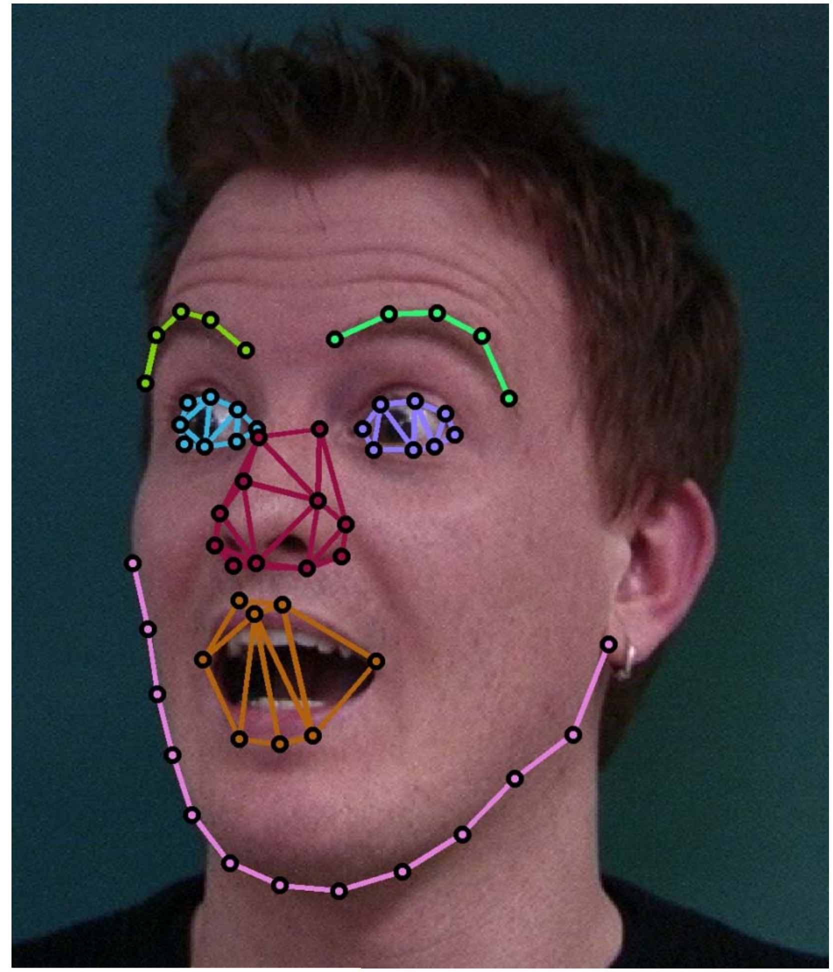
- Graph problems in Vision
- Graph representations of images/videos
- Network problems modeled with Graphs (GIS, communications, social nets, …)
- Theory of graph-based signal/image processing

Graphs in Computer Vision – I: Face

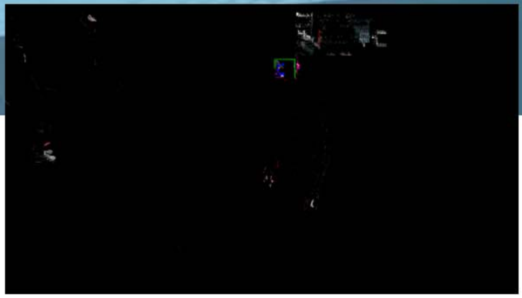
Global Active Appearance Model (AAM)



Local Active Appearance Models



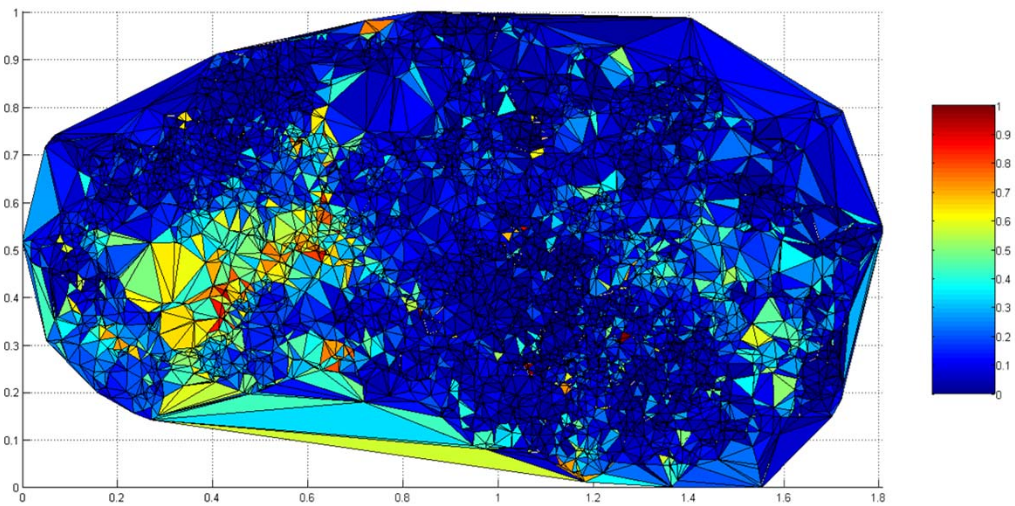
Graphs in Computer Vision – II: Pose and HRI



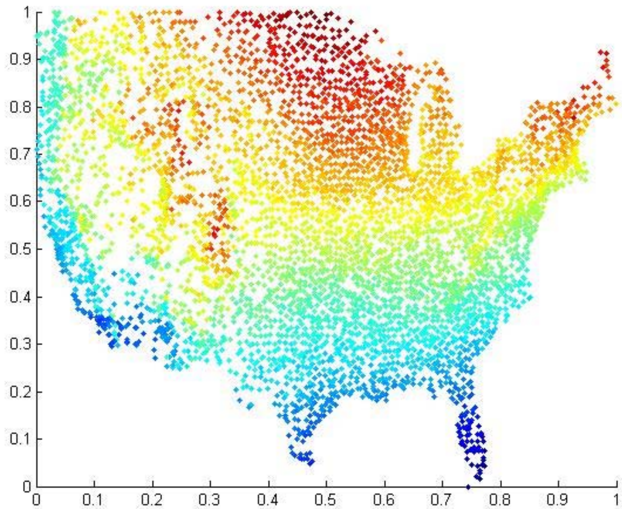
BabyRobot

Graphs in Network Science

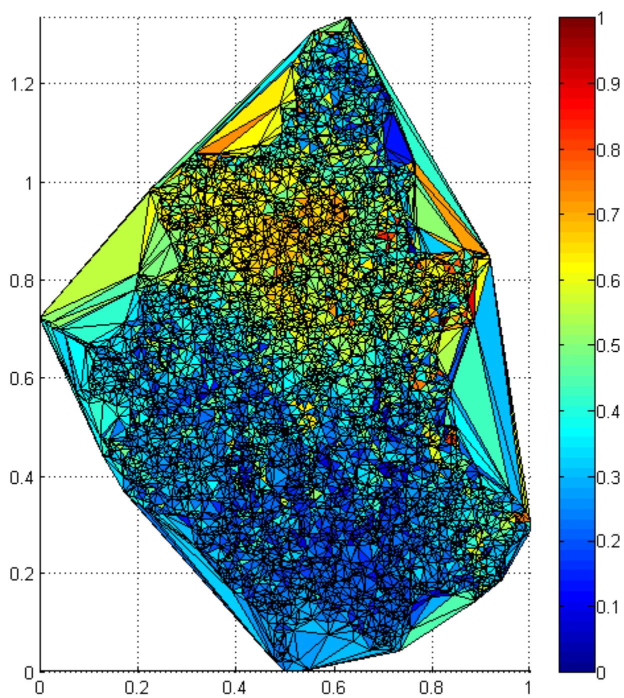
Signal strength of Cellular network

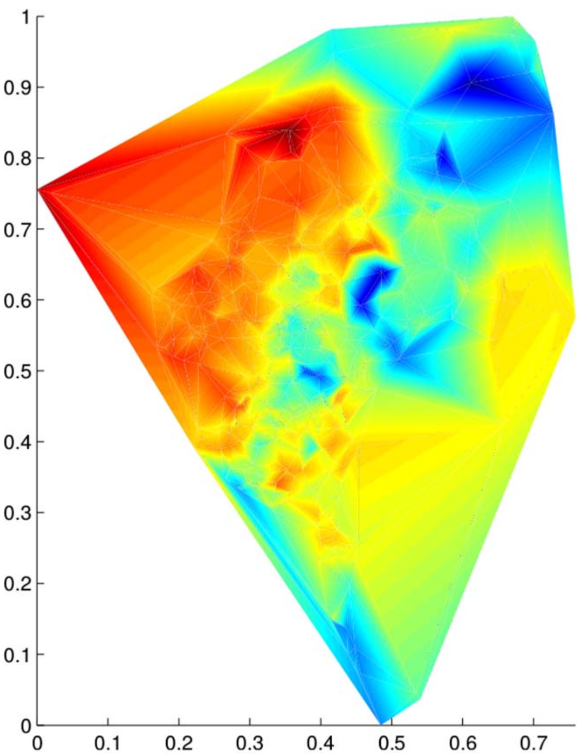


Rainfall Data, 4000 US cities



Average annual Wind speed data



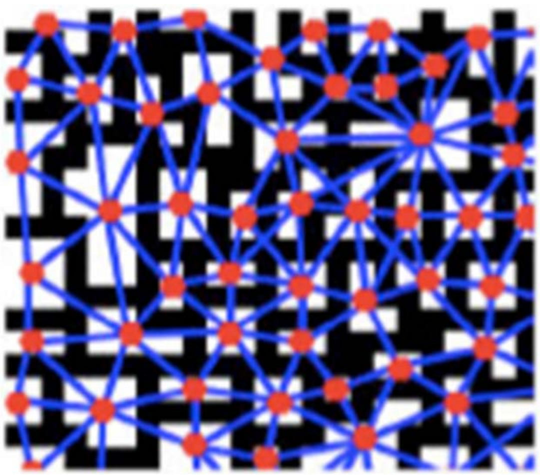


Referendum results at 443 sites over metro Athens area, 2015

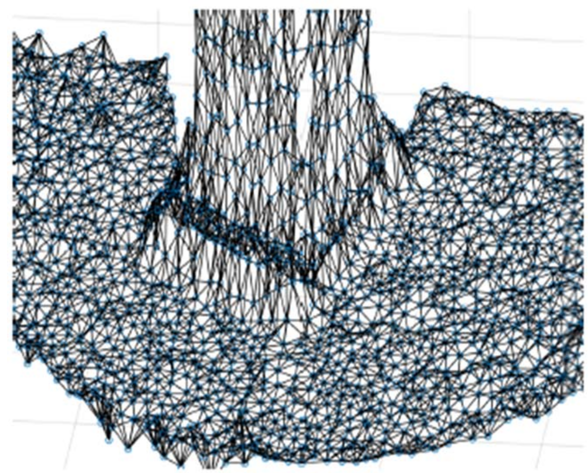
Using Arbitrary Graphs for Image Segmentation

- Visual data can be represented in various ways
- Regular grid representation for images: Simple but does not incorporate local image properties. Also high-dimensional.

2D image graph



point cloud



- Arbitrary graphs can capture intrinsic data structures.
- We can use graph-clustering approaches for image segmentation. It also reduces dimensionality and extends to non-visual data.

Previous Work: Graph-based Segmentation Approaches

- Normalized Cuts:
  - J. Shi and J. Malik, “Normalized Cuts and Image Segmentation”, T-PAMI, 2000.
- Random Walks:
  - L. Grady, "Random Walks for Image Segmentation,” IEEE Trans. PAMI, 2006.
- Power Watershed:
  - C. Couprie, L. Grady, L. Najman, and H. Talbot, “Power watershed: A unifying graph-based optimization framework,” IEEE Trans. PAMI., 2011.
- Supervised Graph Clustering/Machine Learning:
  - D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, “Learning with local and global consistency,” Proc. NIPS, 2004.
  - X. Zhu, Z. Ghahramani, and J. Lafferty, “Semi-supervised learning using Gaussian fields and harmonic functions,” Proc. ICML, 2003.
- Regularization and PDEs on arbitrary graphs:
  - A. Elmoataz, O. Lézoray and S. Bougleux, “Nonlocal discrete regularization on weighted graphs,” IEEE Trans. Imag. Proc., 2008.
  - V.-T. Ta, A. Elmoataz and O. Lezoray, ``Nonlocal PDE-based Morphology on Weighted Graphs for Image and Data Processing", IEEE Trans. Im. Proc., 2011.
- Graph Cuts:
  - Y. Boykov and M.-P. Jolly, “Interactive Graph Cuts”, ICCV 2001;
  - Y. Boykov, O. Veksler & R. Zabih, “Fast Approximate Energy Mimimization”, TPAMI 2001.
  - V. Kolmogorov and R. Zabih, “What Energy Functions can be Minimized”, TPAMI 2004.
  - C. Rother, V. Kolmogorov and A. Blake, “GrabCut”, SIGGRAPH 2004.

Outline of Talk

- I. Active Contours on Arbitrary Graphs
  - Multiscale Morphology on Graphs
  - Active Contours (GAC, ACWE) on Graphs
  - Theoretical results: convergence and error bound
  - Finite Elements
- II. Graph-driven Diffusion and RW Schemes
  - SIR epidemic propagation model and RW
  - Normalized Random Walker (NRW)
- Experimental results on graph segmentation

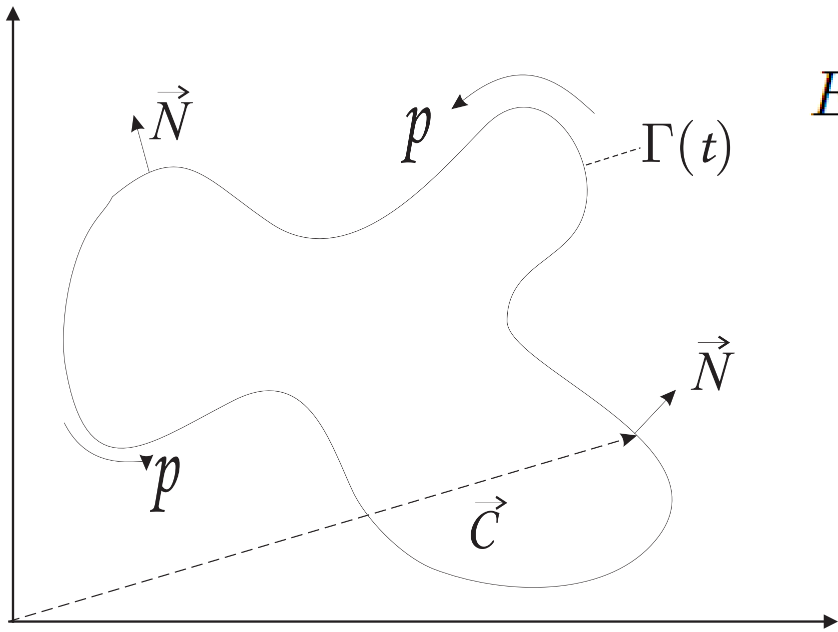
Active Contours on Euclidean plane/space

- Active **Contours**
  - Snakes [Kass et al. 1987]
  - Level set method for Curve Evolution [Osher & Sethian 1988]
  - Deformable Templates [Yuille et al. 1989]
  - Balloons [Cohen, 1991]
  - Geometric Curve Evoln [Caselles et al. 1993, Malladi et al. 1995]
  - Geodesic Active Contours [Caselles, Kimmel & Sapiro 1997]

- Active **Regions**
  - Mumford-Shah Energy Minimization (1989)
  - Region Competition [Zhu & Yuille 1996]
  - Active Contours Without edges [Chan & Vese 2001]
  - Geodesic Active Regions [Paragios & Deriche 2002]

Curve Evolution via "Energy" Minimization

Active Contours: Evolution of curve **Γ** in 2D space under the influence of "energy" minimizing forces until convergence:



$$E = \int_0^1 [E_{int}(\vec{C}) + E_{edge}(\vec{C}) + E_{con}(\vec{C})]dp$$

E_{int}: smoothness; E_{edge}: edges; E_{con}: external constraints

$$\frac{\partial \vec{C}(p,t)}{\partial t} = V \cdot \vec{N}_o(p,t)$$

In the limit curve **Γ** is identified with the boundary(ies) among the objects to be detected.

References:

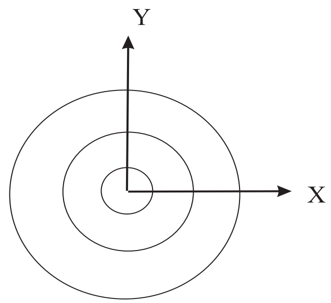
- *Snakes*: Kass, Witkin & Terzopoulos (1987)
- *Deformable Templates*: Yuille et al (1989)
- *Level sets:* Osher & Sethian (1988)
- *Geometric Models for ACs*: Caselles et al (1993), Malladi et al (1995)
- *GACs*: Caselles, Kimmel & Sapiro (1997)
- *ACWE*: Chan & Vese (2001)

Level-set Method for Curve Evolution [Osher & Sethian 1988]

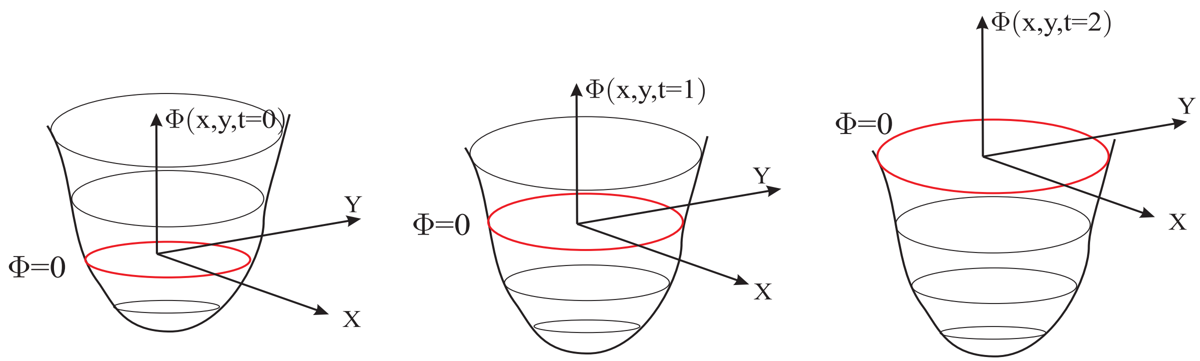
Embed curve Γ as level-set of 3D function Φ at level λ=0.

$$\Gamma(t) = \{(x,y) : \Phi(x,y,t) = \lambda\}$$

$$\Phi_0(x,y) = \begin{cases} \lambda, & (x,y) \in \Gamma(0) \\ \lambda - \bigwedge_{(v,w)\in\Gamma(0)} \|(x-v, y-w)\|, & (x,y) \in \text{interior}(\Gamma(0)) \\ \lambda + \bigwedge_{(v,w)\in\Gamma(0)} \|(x-v, y-w)\|, & (x,y) \in \text{exterior}(\Gamma(0)) \end{cases}$$



$$\Phi(\vec{C}(p,t),t) = \lambda$$



$$\vec{N}_o = \frac{\nabla\Phi}{\|\nabla\Phi\|}$$

$$\kappa = \nabla \cdot \vec{N}_o = \text{div}\left(\frac{\nabla\Phi}{\|\nabla\Phi\|}\right)$$

PDE for 2D curve evolution

$$\frac{\partial\vec{C}(p,t)}{\partial t} = V \cdot \vec{N}_o(p,t)$$

PDE for 3D function evolution

$$\begin{aligned} \partial\Phi/\partial t &= -V\|\nabla\Phi\| \\ \Phi(x,y,0) &= \Phi_0(x,y) \end{aligned}$$

Part I.A

Active Contours and Multi-Scale Morphology on Graphs

Main Refs:

- K. Drakopoulos and P. Maragos, "Active Contours on Graphs: Multiscale Morphology and Graphcuts", *IEEE Journal of Selected Topics in Signal Processing*, Nov. 2012.

- C. Sakaridis, K. Drakopoulos and P. Maragos, "Theoretical Analysis of Active Contours on Graphs", *SIAM J. Imaging Sciences*, vol. 10, no. 3, pp. 1475-1510, 2017.

Geodesic Active Contour (GAC) Model

[Caselles, Kimmel & Sapiro, IJCV 1997]

- Energy Minimization (*C*=curve, *I*=image, *g*=edge-stop fcn)

$$E(\mathcal{C}) = \alpha \int_0^1 |\mathcal{C}'(q)|^2 \, dq + \lambda \int_0^1 g(|\nabla I(\mathcal{C}(q))|)^2 \, dq$$

- PDE for Curve Evolution:

$$\vec{C}_t = g(C) \cdot \kappa \cdot \vec{N} - (\nabla g(C) \cdot \vec{N}) \cdot \vec{N}$$

- Implementing Curve Evolution by propagating Level Sets of Embedding Level Function $u(x,y,t)$ with a PDE:

$$\frac{\partial u}{\partial t} = g(I)|\nabla u|(\kappa + c) + \nabla g(I) \cdot \nabla u, \quad c \geq 0$$

Augmented Geodesic Model

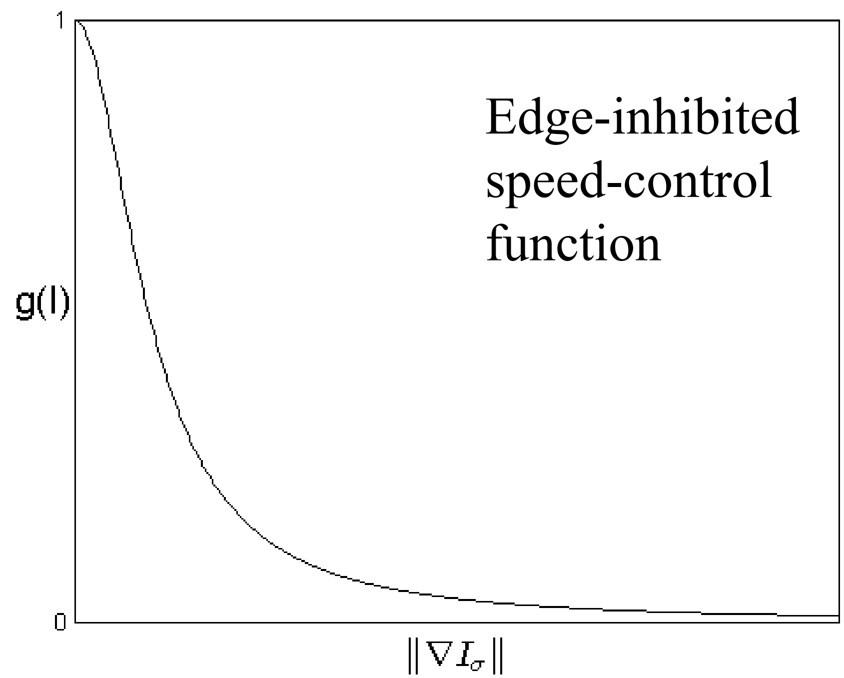
$$\frac{\partial u}{\partial t} = g(I)|\nabla u|(\kappa + c) + \nabla g(I) \cdot \nabla u, \quad c \geq 0$$

$g(I)|\nabla u|\kappa$ — Curvature motion

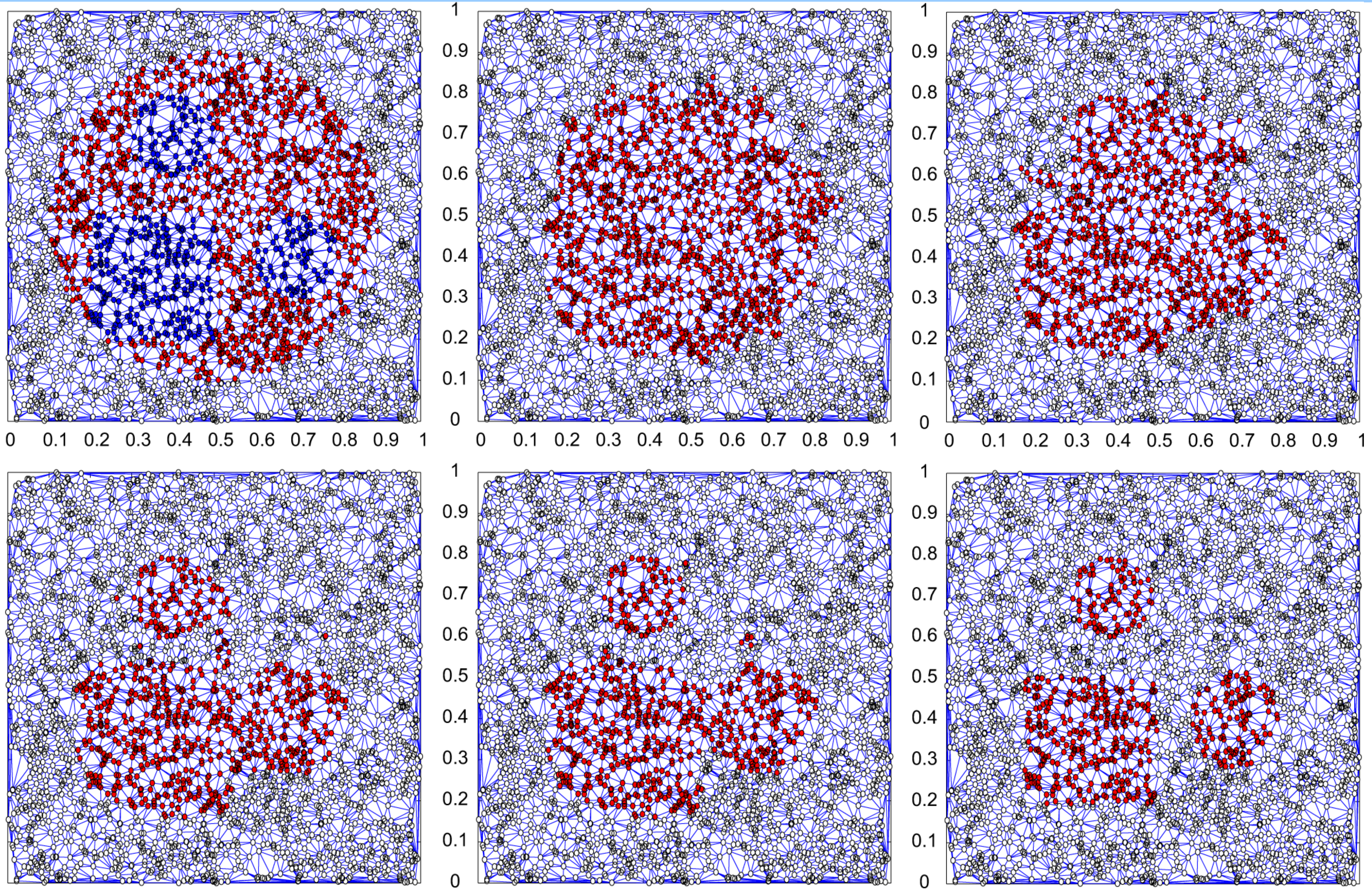
$g(I)|\nabla u|c$ — Balloon force (Dilation/Erosion)

$\nabla g(I) \cdot \nabla u$ — Spring force

$$g(\|\nabla I_\sigma\|) = \frac{1}{1 + \frac{\|\nabla I_\sigma\|^2}{\lambda^2}}$$



Cluster Detection on Graphs w. Active Contours



[Drakopoulos & Maragos 2012]

GACs on Graphs

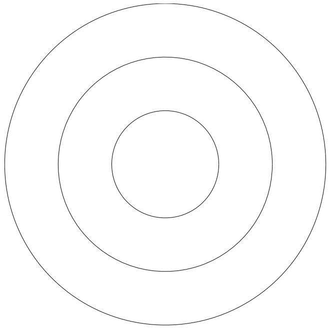
- Motion with Constant velocity (Dilation/Erosion PDE)

$$\frac{\partial u}{\partial t} = c \, || \nabla u ||$$

PDE for 2D Multiscale Flat Dilations

- initial image, multiscale flat convex structur. elems. (disks)

$f(x,y)$



tB

- multiscale dilations by disks : $\delta(x,y,t) = (f \oplus tB)(x,y)$

$t = 3$

$t = 6$

$t = 9$

- PDE:

$$\frac{\partial \delta}{\partial t} = \|\nabla \delta\| = \sqrt{\left(\frac{\partial \delta}{\partial x}\right)^2 + \left(\frac{\partial \delta}{\partial y}\right)^2}$$

[Brockett & Maragos 1992]

[Alvarez et al. 1993]

Morphology on Graphs - I

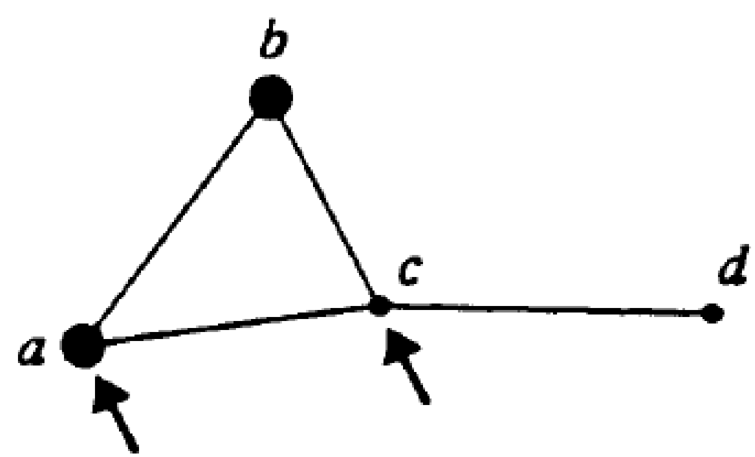
[Vincent 1989, Heijmans et al. 1992]

Structuring Graph ***A*** = graph $G_A(V_A, E_A)$

with two nonempty subsets:

roots: $R_A \subseteq V_A$, buds: $B_A \subseteq V_A$

Example: roots = {a,c},
buds ={a,b}

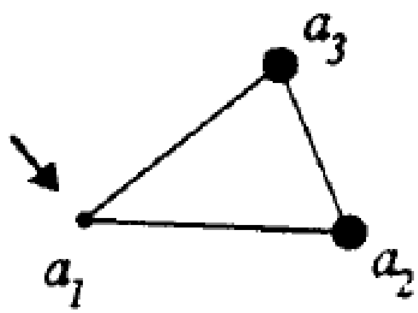


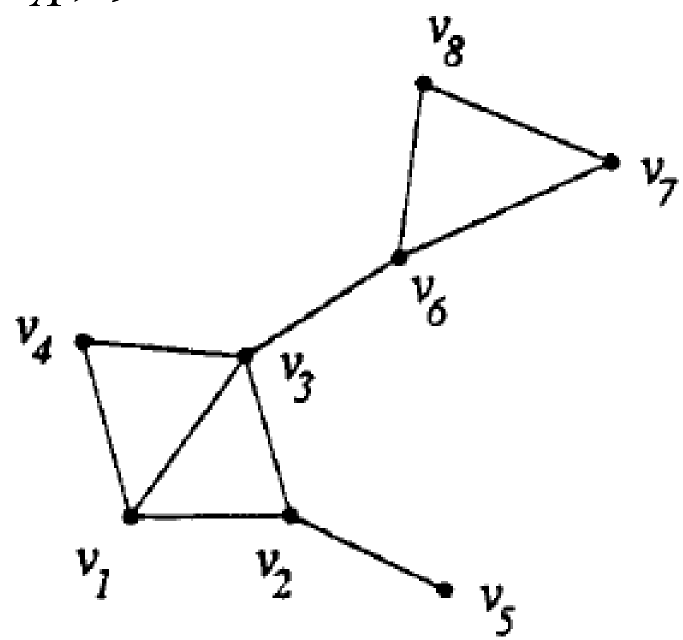
Morphology on Graphs – II

structuring graph $A = G_A(V_A, E_A)$

$A-$**Neighborhood** of node $v \in V$ in graph $G(V,E)$:

$$N_A(v \mid G) = \bigcup \{\theta(B_A) : \ \theta : G_A \to G \text{ is homomorphism and } v \in \theta(R_A)\}$$





Morphology on Graphs – III

- Graph dilation : $$\delta_A(X|G) = \bigcup_{v \in X} N_A(v, G)$$
- Graph erosion : $$\epsilon_A(X|G) = \{v \in V : N_A(v, G) \subseteq X\}$$

Erosion – Dilation on Graphs

- Levelsets on graph:

$$X_i(f) = \{v \in V : f(v) \geq i\}$$

- Function reconstruction:

$$f(v) = \max_{i=1,\dots,n-1} \{i : v \in X_i(f)\}$$

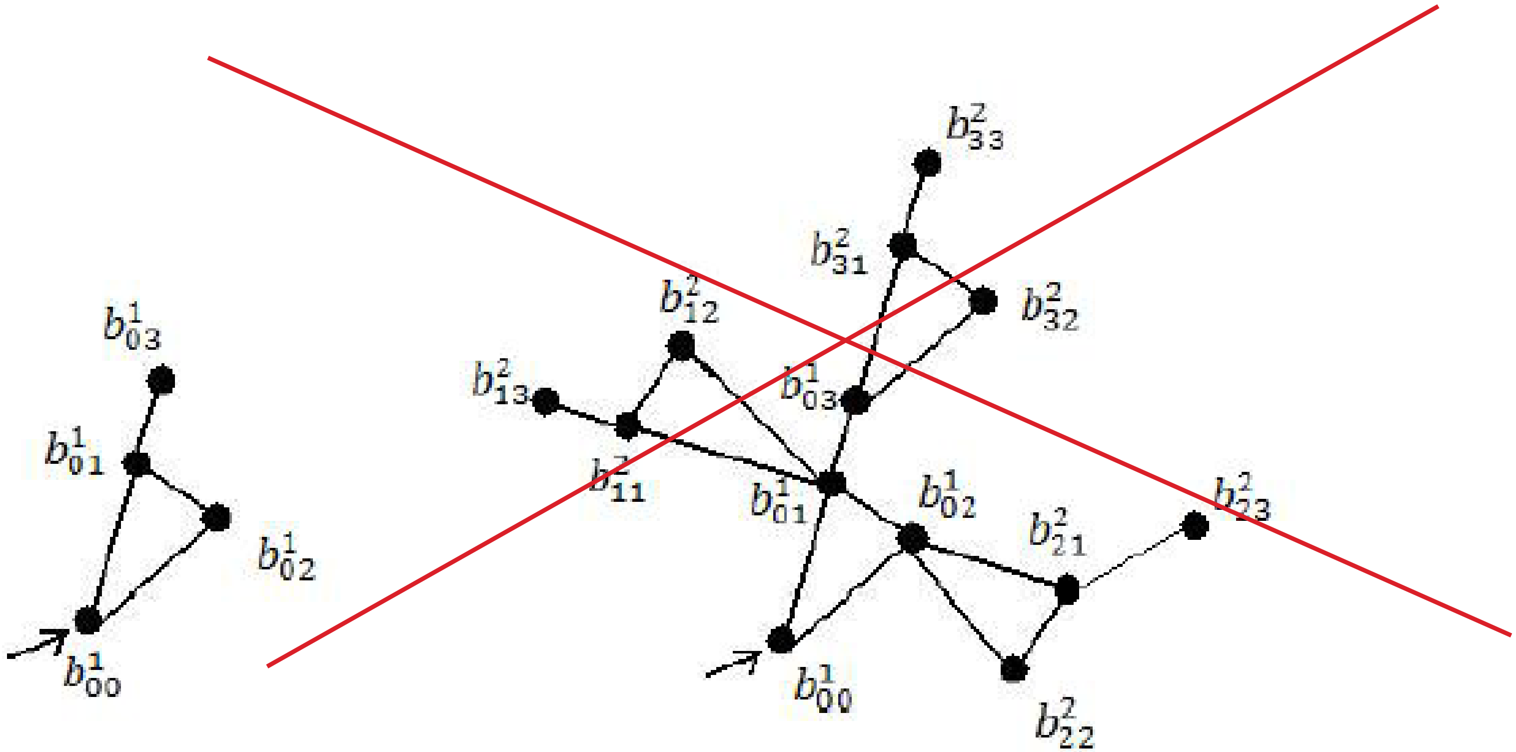
- Flat Graph Dilation :

$$\delta_A(f,G)(v) = \sup\{f(w) : w \in N_{\hat{A}}(v|G)\}$$

$$G_A = G_{\hat{A}}, Buds_A = Roots_{\hat{A}}, Roots_A = Buds_{\hat{A}}.$$

Scale Definition on Graphs – I

- A possible choice for multiscale structuring graphs:



Scale Definition on Graphs – II

- Define **scale *r*** recursively as:

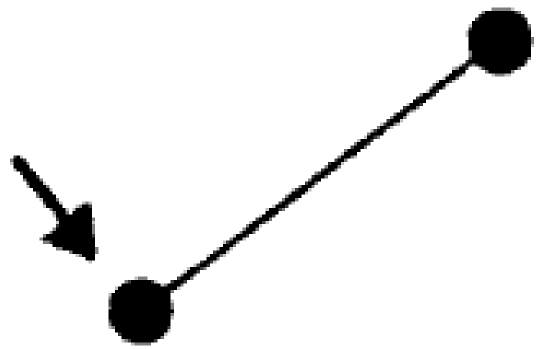
$$\phi_r(v) \triangleq \delta_{\mathscr{A}}(\phi_{r-1}(f \mid G) \mid G)(v)$$

- Difference equation that models above:

$$\phi_{r+1}(v) - \phi_r(v) = \max_{w \in N_{\mathscr{A}}(v|G)} \{\phi_r(w) - \phi_r(v)\}$$

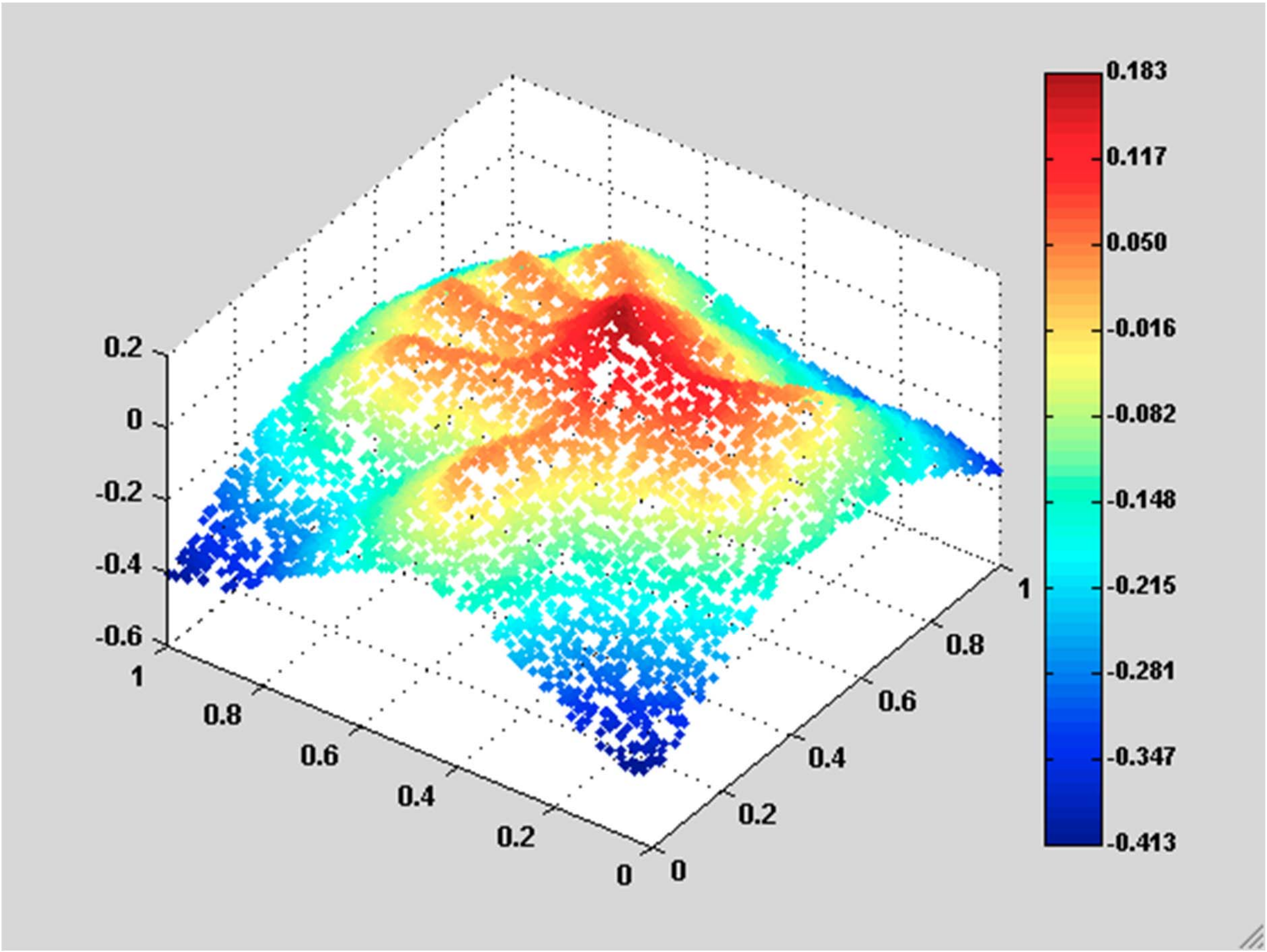
Scale Definition on Graphs – III

- For the experiments in this presentation, choose the structuring graph $\boldsymbol{A}$

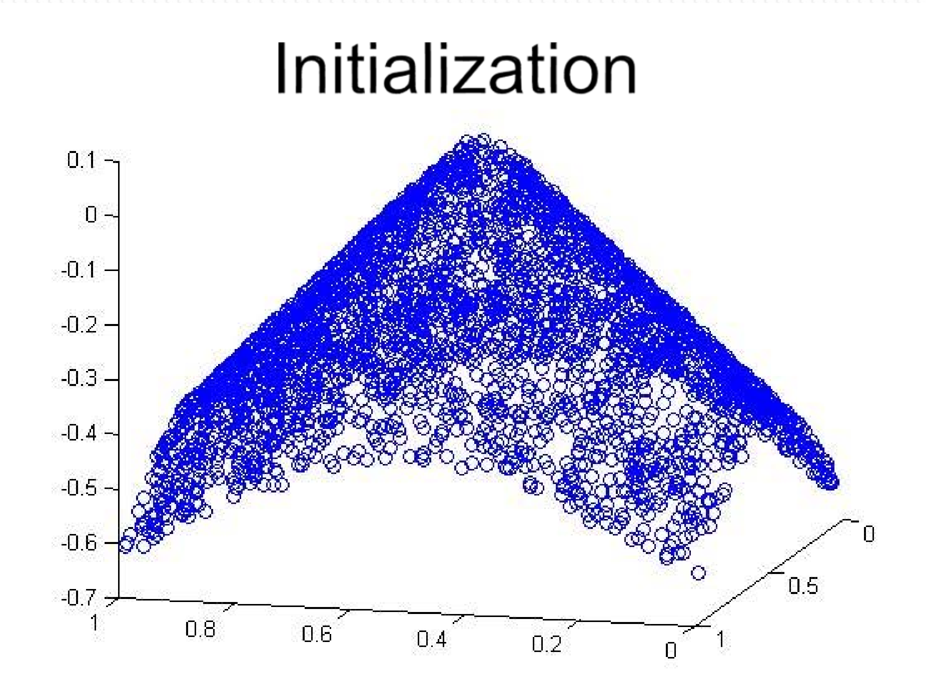


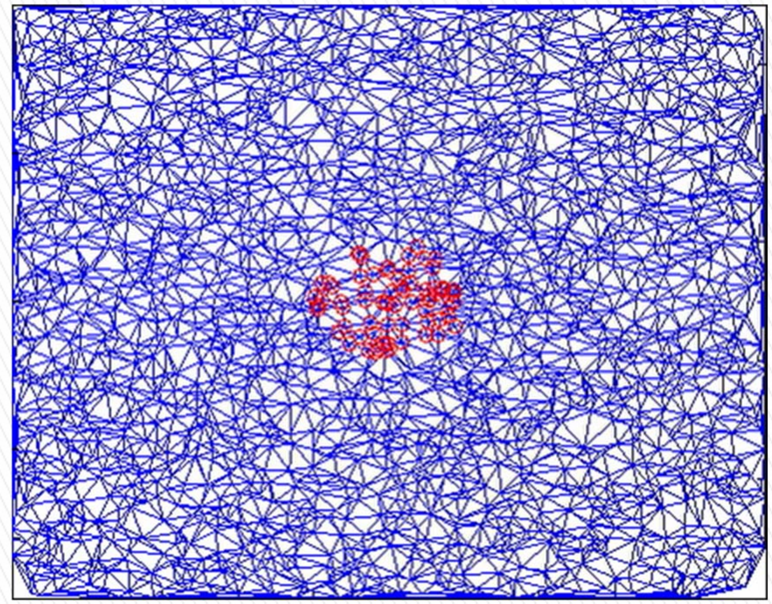
- Corresponding Neighborhood at each node:
  Set of 1-neighbors around each node

Signed distance function ←→ multiscale dilations of a shape on graphs



Implementing Multiscale Dilation via Level Sets on Arbitrary Graphs





Signed Distance Function

Dilate circular initial curve

Domain: a geometric random graph

GACs on Graphs with Level Sets

Approximating the Full Motion

$$\frac{\partial u}{\partial t} = g(I)|\nabla u|(\kappa + c) + \nabla g(I) \cdot \nabla u, \quad c \geq 0$$

Approximating the Gradient

- **Gradient Magnitude**:

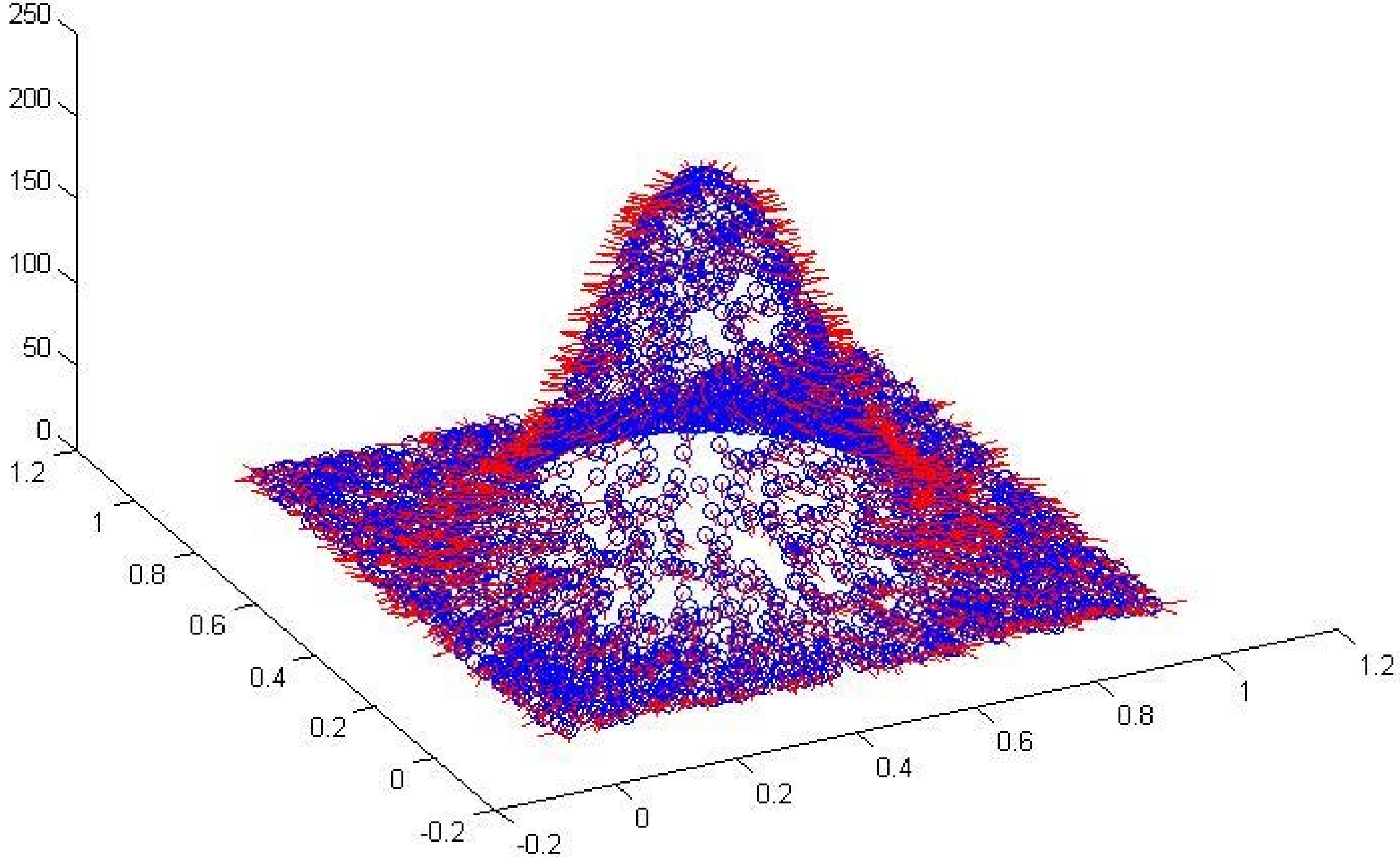
$$|\nabla\phi|(u) = \max_{w\in N_B(u,G)} \{|\phi(w) - \phi(u)|\}$$

- **Gradient orientation: unit Normal**:

$$\vec{n} = \frac{\sum_{w\in N_B(v,G)}(f(w) - f(v))\left(\frac{p(\vec{w})-p(\vec{v})}{|p(\vec{w})-p(\vec{v})|}\right)}{|\sum_{w\in N_B(v,G)}(f(w) - f(v))\left(\frac{p(\vec{w})-p(\vec{v})}{|p(\vec{w})-p(\vec{v})|}\right)|}$$

$p(w)$ = position vector of node w

Approximate Estimate of Normal (of a Gaussian on the graph)



Theorem 1: Let $u : [0,1]^2 \longrightarrow \mathbb{R}$ be a twice differentiable function on the unit square and let $G(n, \rho(n))$ be a geometric random graph embedded in the same domain. Let v be a node of $G(n, \rho(n))$ and let N_v denote the set of nodes adjacent to v. Then, if $\rho(n) = 1/n^\alpha$ where $\alpha < 1/2$,

$$\lim_{n \longrightarrow \infty} \max_{w \in N_v} \left\{ \frac{u(w) - u(v)}{d(w,v)} \right\} = |\nabla u(v)|,$$

and

$$\lim_{n \longrightarrow \infty} \mathbf{e}_{vw(v)} = \frac{\nabla u(v)}{|\nabla u(v)|},$$

in probability, where $w(v) = \operatorname{argmax}_{w \in N_v} \{u(w) - u(v)/d(w,v)\}$.

[Drakopoulos & Maragos, IEEE J-STSP 2012]

Approximating the Curvature

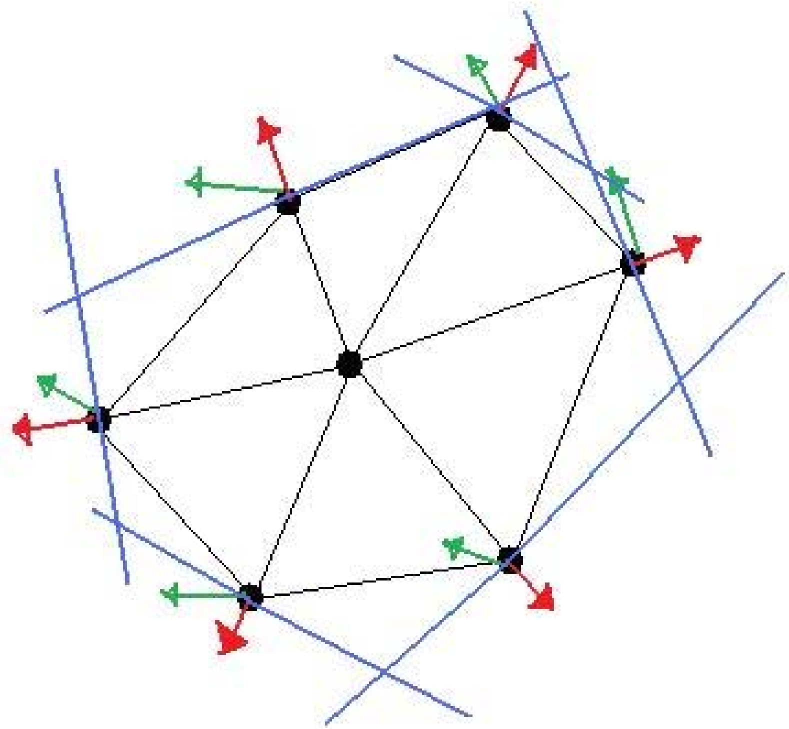
- Based on

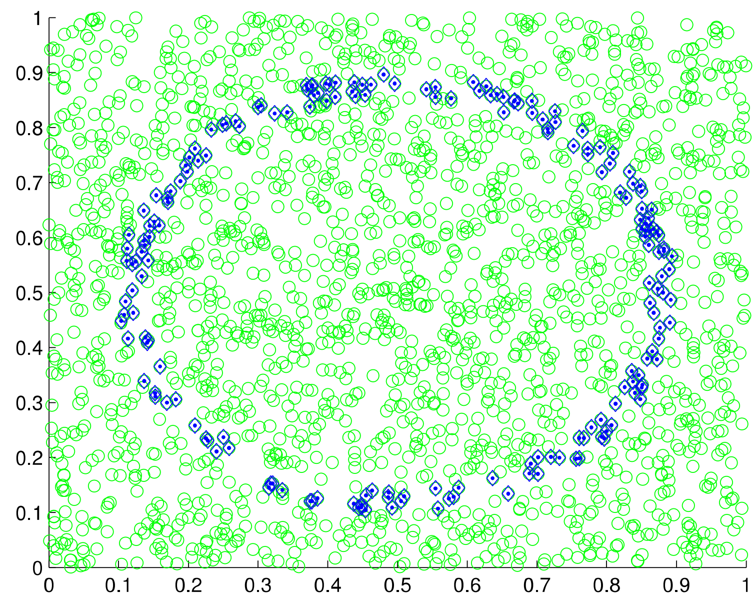
$$\kappa = div\left(\frac{\nabla u}{|\nabla u|}\right)$$

- Divergence of vector field:

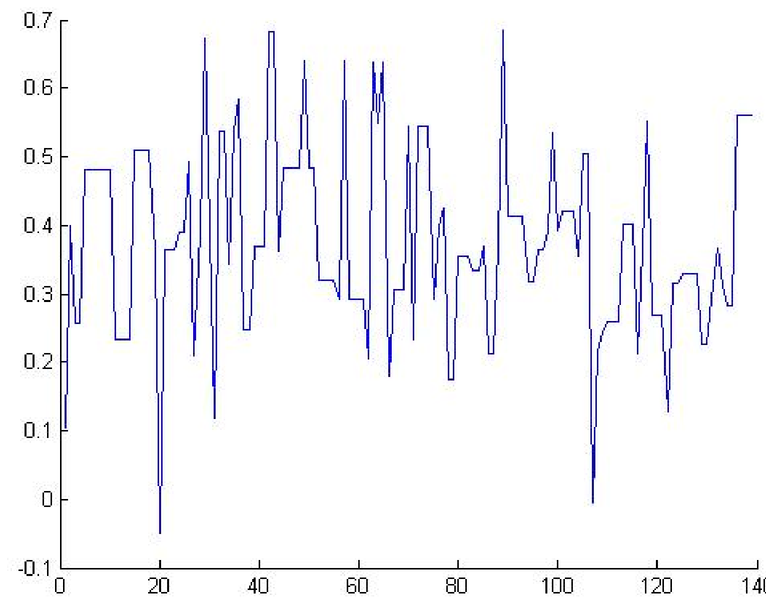
$$div(\vec{F}) = \lim_{S \to 0} \frac{\int_{L(S)} \vec{F}\vec{n}dl}{S}$$

Approximate Estimate of Curvature

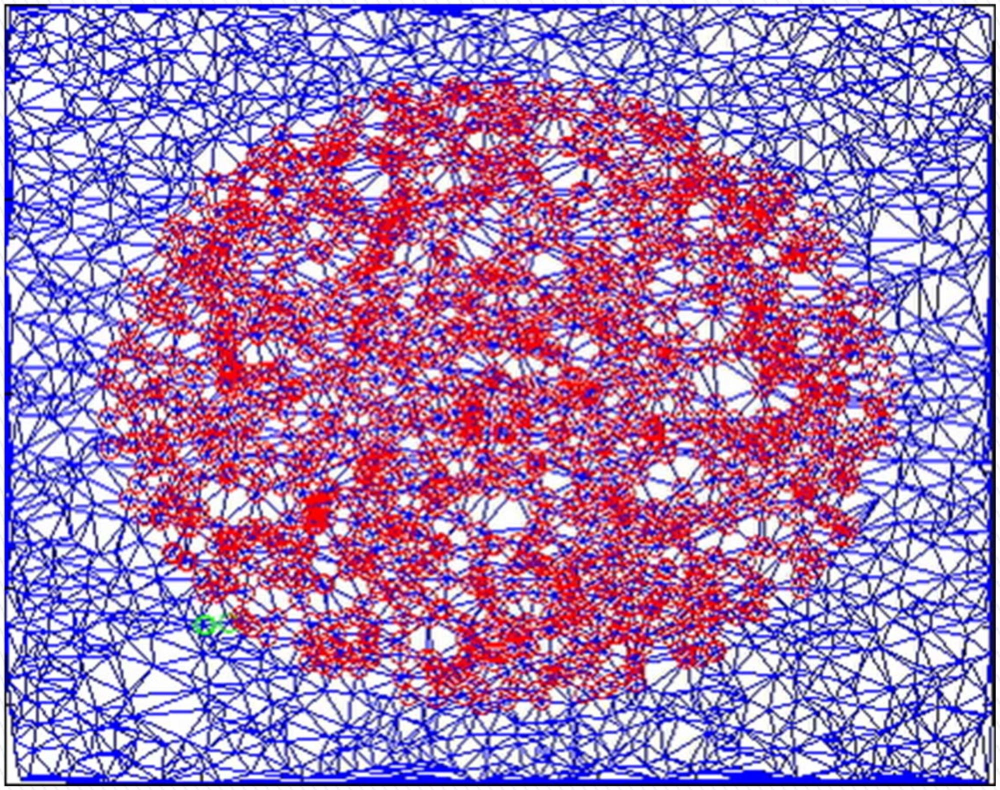


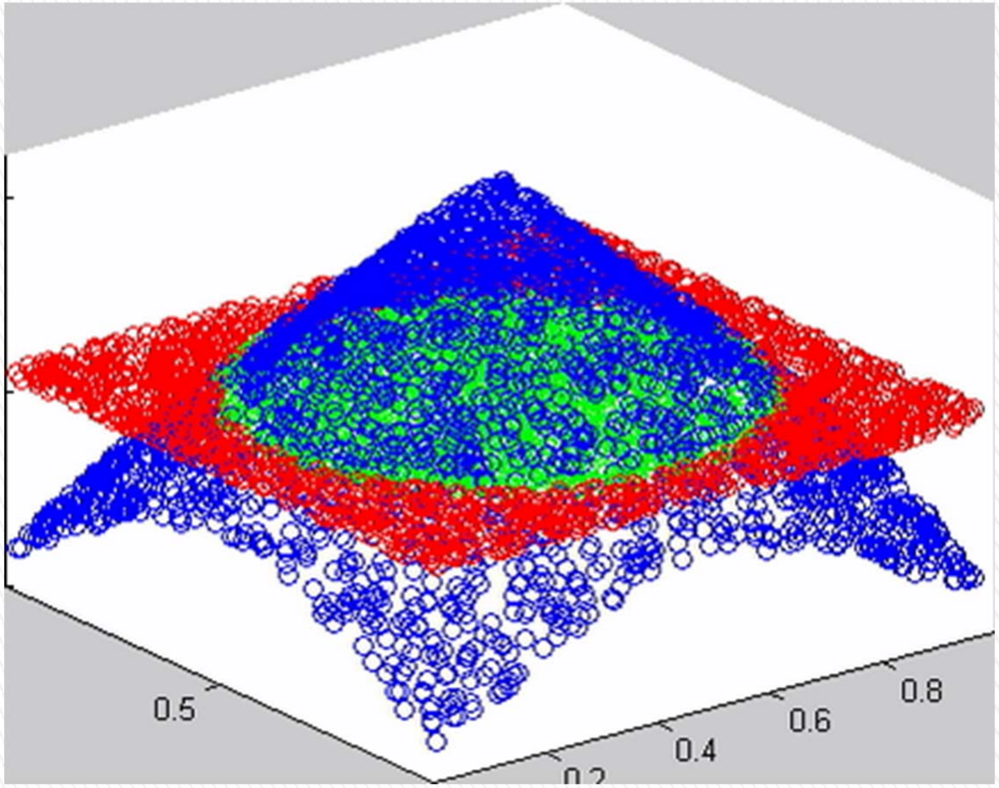


$$div(\vec{f})(v) \propto \sum_{w \in N_B(v,G)} \vec{f_w} \cdot \vec{n_w}$$



Three Regions

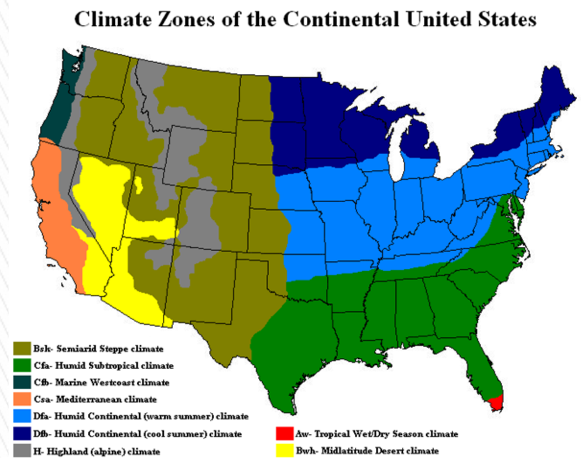


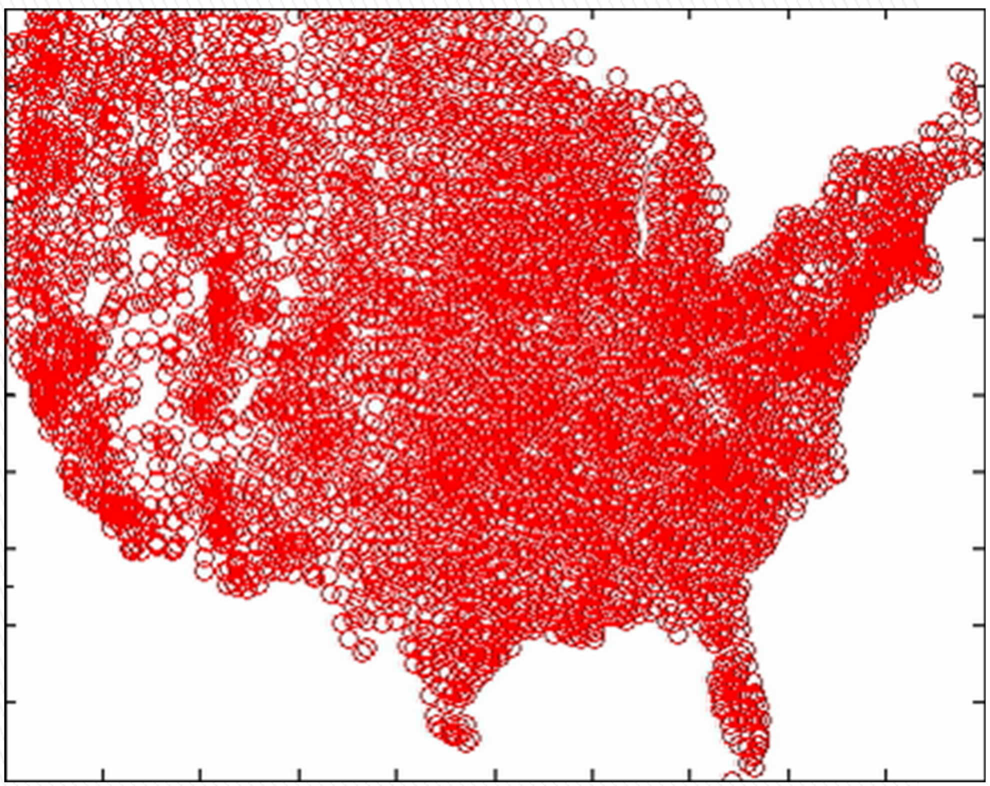


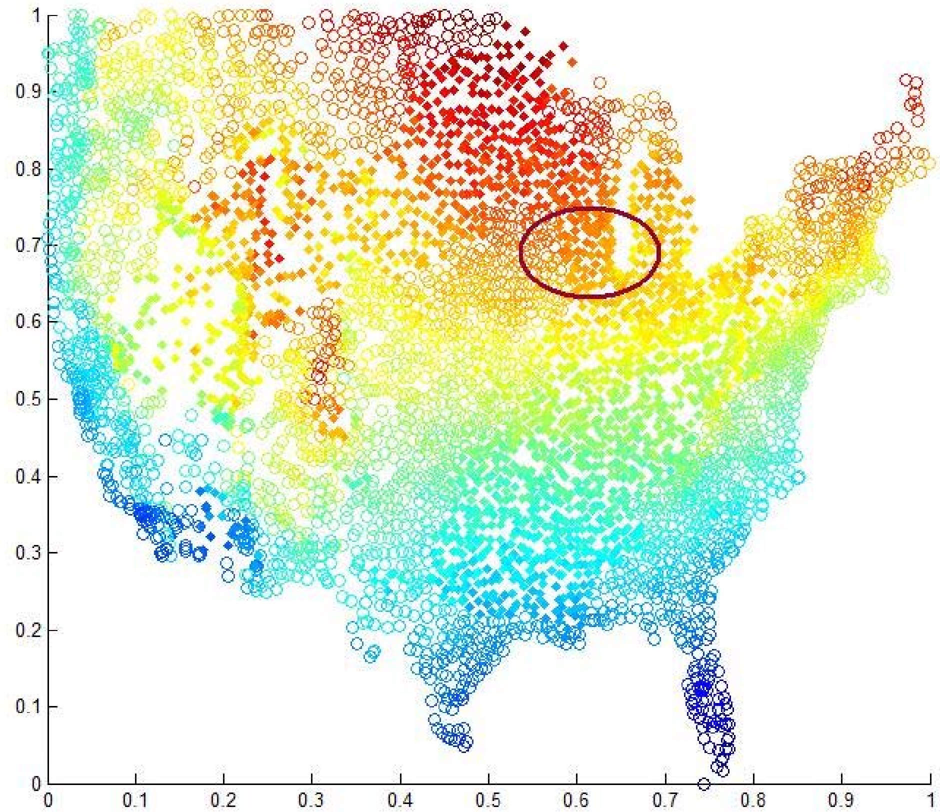
Active Region

Embedding Level Function

Rainfall Data over 4000 USA cities







Run Algorithm

Final Result

Questions

- How do the results scale with # of graph nodes?
- Do graph geometric quantities converge to Euclidean values?
- Is the error bounded?
- Segmentation:
  - Stability ?

Part I.B

Theoretical Analysis of Active Contours on Graphs

Main Reference:

- Christos Sakaridis, Kimon Drakopoulos and P. Maragos, *SIAM J. Imaging Sciences*, vol. 10, no. 3, pp. 1475-1510, 2017.

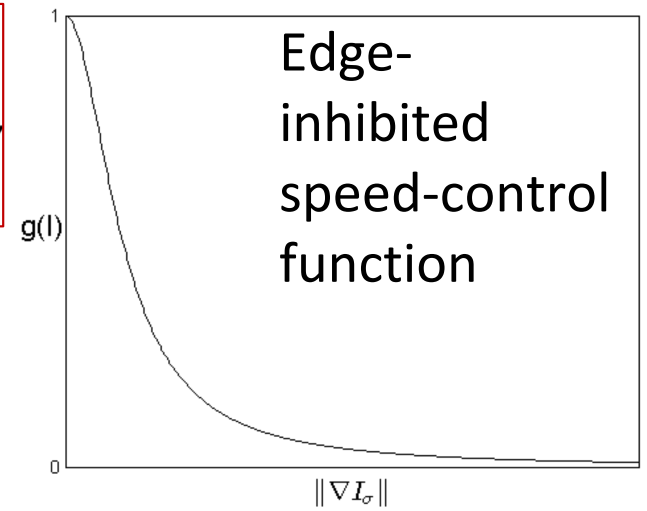
Background, References

- **Active contours based on partial differential equations (PDEs)**
  - Osher and Sethian, J. Comp. Physics 1988: curve evolution with PDEs using level sets, balloon + curvature force, numerical implementation
  - Caselles, Kimmel, and Sapiro, IJCV 1997: geodesic active contours, boundary detection/image segmentation, +spring force, uniqueness and stability of solution
- **PDE-based morphology and active contours on arbitrary graphs**
  - Ta, Elmoataz, and Lezoray, IEEE T-IP 2011: discretized morphological PDEs on graphs, nonlocal image processing
  - Drakopoulos and Maragos, IEEE J-STSP 2012: geodesic active contours on graphs, gradient/curvature approximation, multiscale morphological operators on graphs
- **Graphcut-based optimization for image segmentation:** Couprie, Grady, Najman, and Talbot, IEEE T-PAMI 2011: unification and generalization of graphcuts and watershed

Geodesic active contour (GAC) model

$$\frac{\partial u}{\partial t} = \underbrace{\kappa \|\nabla u\| \, g(I)}_{\text{curvature}} - \underbrace{c \|\nabla u\| \, g(I)}_{\text{balloon}} + \underbrace{\nabla u \cdot \nabla g(I)}_{\text{spring}}$$

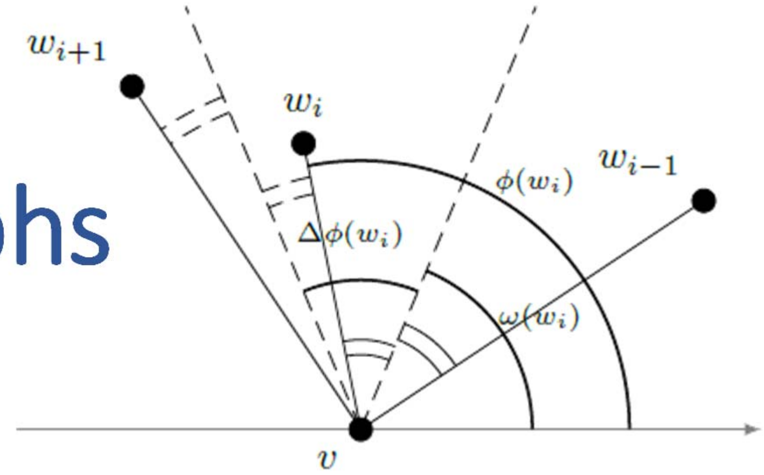
$$\kappa = \text{div}\left(\frac{\nabla u}{\|\nabla u\|}\right) \qquad g(I) = \frac{1}{1 + \frac{\|\nabla I_\sigma\|^2}{\lambda^2}} \qquad c > 0$$



The active contour is **embedded** in the evolving function $u(x,y,t)$ as its zero **level set**.

How can we approximate **gradient** and **curvature** in the PDE on discrete structures like geometric graphs?

Notation for geometric graphs



$$\mathcal{G} = (\mathcal{V}(\mathcal{G}), \mathcal{E}(\mathcal{G})) \qquad \mathcal{V}(\mathcal{G}) \subset \mathbb{R}^2,\ |\mathcal{V}(\mathcal{G})| = n$$

$$\forall v \in \mathcal{V}(\mathcal{G}):\ \mathcal{N}(v) = \{w \in \mathcal{V}(\mathcal{G}) : v \sim w\},\ |\mathcal{N}(v)| = N(v) \equiv N$$

$$d(v,w) = \|\mathbf{w} - \mathbf{v}\| \qquad \mathbf{e}_{vw} = \frac{\mathbf{w} - \mathbf{v}}{d(v,w)} \qquad \forall w \in \mathcal{N}(v):\ \phi(w) = \mathrm{Arg}(\mathbf{e}_{vw}) \in [0, 2\pi)$$

Set of vertex neighbors with edge angles in **ascending** order:

$$\mathcal{N}(v) = \{w_1, w_2, \ldots, w_N\}$$

Neighbor angle

$$\Delta\phi(w_i) = \begin{cases} \dfrac{\phi(w_{i+1}) - (\phi(w_N) - 2\pi)}{2} & \text{if } i = 1, \\ \dfrac{\phi(w_1) + 2\pi - \phi(w_{i-1})}{2} & \text{if } i = N, \\ \dfrac{\phi(w_{i+1}) - \phi(w_{i-1})}{2} & \text{otherwise.} \end{cases}$$

$$\omega(w_i) = \begin{cases} \dfrac{\phi(w_i) + \phi(w_N) - 2\pi}{2} & \text{if } i = 1, \\ \dfrac{\phi(w_i) + \phi(w_{i-1})}{2} & \text{otherwise.} \end{cases}$$

Geometric gradient approximation on graphs

$$\nabla u(v) = \frac{\int_0^{2\pi} D_\phi u(v)\, \mathbf{e}_\phi \, d\phi}{\pi}$$

$$\nabla u(v) \approx \frac{\sum_{i=1}^{N} \frac{u(w_i) - u(v)}{d(v, w_i)} \mathbf{e}_{vw_i} \Delta\phi(w_i)}{\pi}$$

$$D_\phi u(\mathbf{x}) = \lim_{h \to 0} \frac{u(\mathbf{x} + h\mathbf{e}_\phi) - u(\mathbf{x})}{h}$$

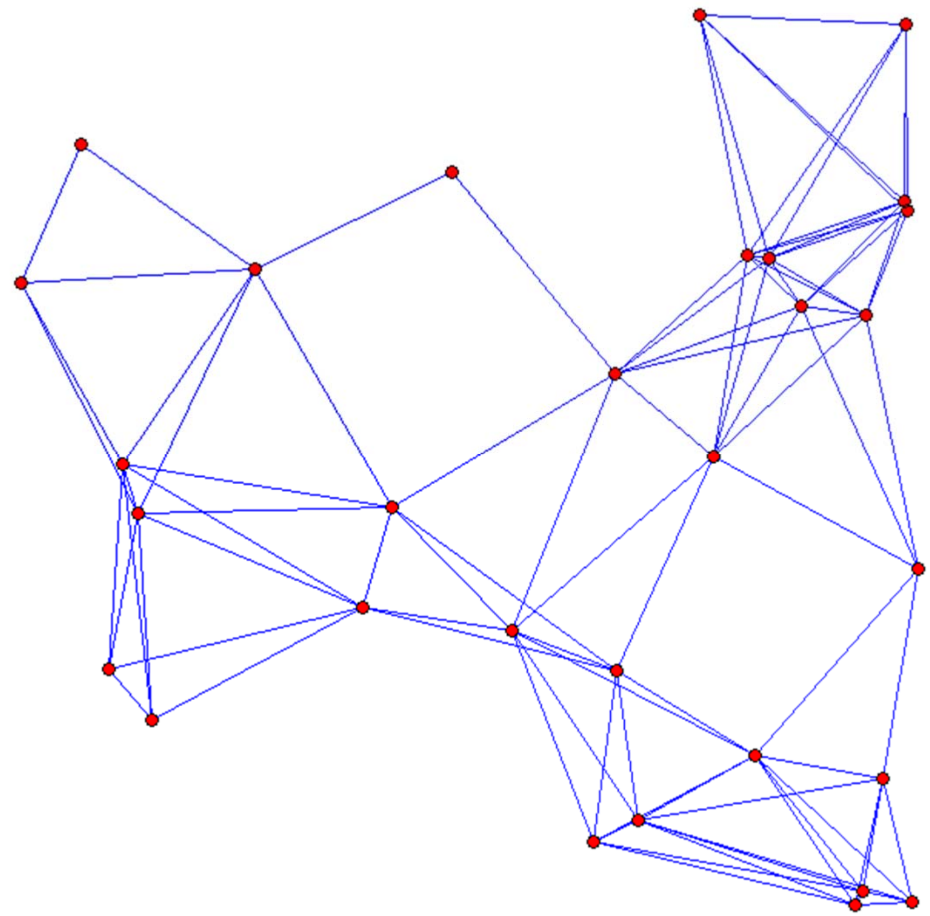
Threefold approximation:

1. Directional derivatives along edges are approximated with difference quotients.
2. The difference quotient along each edge is used as a constant approximation of all directional derivatives in those directions that “fall into” the respective neighbor angle.
3. The unit vector in the direction of each edge is used as a constant representative for all directions that “fall into” the respective neighbor angle.

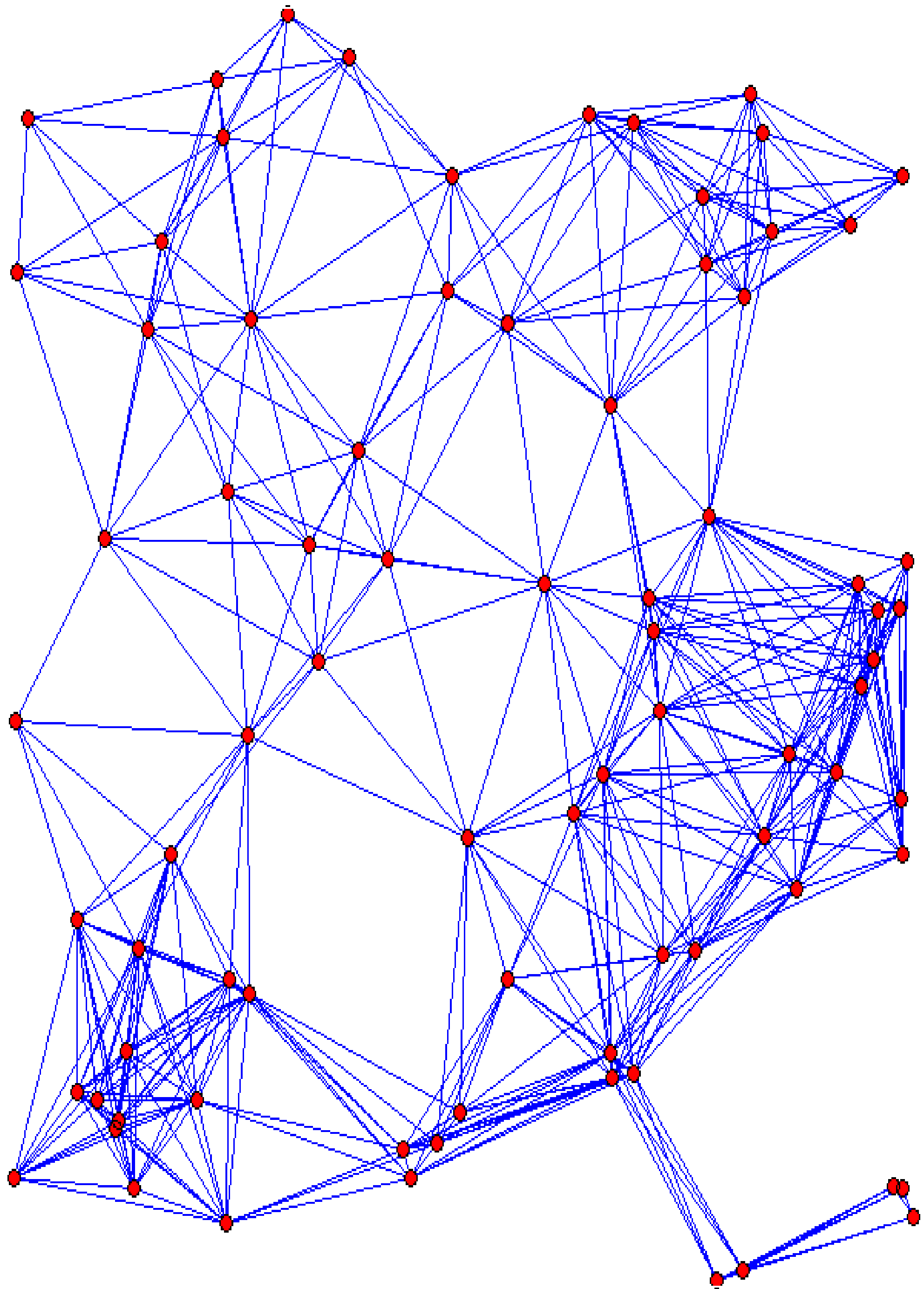
Rationale: use information from closest direction about local change

Random geometric graphs (RGGs)

Definition 1 *A random geometric graph (RGG)* $\mathcal{G}(n, \rho(n))$ *is comprised of a set* $\mathcal{V}(\mathcal{G})$ *of vertices and a set* $\mathcal{E}(\mathcal{G})$ *of edges. The set* $\mathcal{V}(\mathcal{G})$ *consists of* n *points distributed uniformly and independently at random in a bounded region* $D \subset \mathbb{R}^2$*. The set* $\mathcal{E}(\mathcal{G})$ *of edges is defined through the radius* $\rho(n)$ *of the graph: an edge connects two vertices* v *and* w *if and only if their distance is at most* $\rho(n)$*, i.e.* $d(v, w) \leq \rho(n)$*.*



$$n = 80,\ \rho = 0.25,\ D = [0,\ 1]^2$$



Convergence of gradient approximation for RGGs

Limit of large graphs: $|\mathcal{V}(\mathcal{G})| = n \to +\infty$

- **Stochastic** nature of RGGs: only convergence of probabilistic kind can be established, e.g. **convergence in probability**
- Easy to handle RGGs theoretically due to their **simple definition**
- Asymptotic restrictions for radius required

Reminder:

$$f(n) \in O(g(n)) \Leftrightarrow \exists k > 0 \; \exists n_0 \; \forall n \geq n_0 : \; f(n) \leq k g(n)$$
$$f(n) \in \Theta(g(n)) \Leftrightarrow f(n) \in O(g(n)) \wedge g(n) \in O(f(n))$$
$$f(n) \in o(g(n)) \Leftrightarrow \forall k > 0 \; \exists n_0 \; \forall n \geq n_0 : \; f(n) < k g(n)$$
$$f(n) \in \omega(g(n)) \Leftrightarrow \forall k > 0 \; \exists n_0 \; \forall n \geq n_0 : \; f(n) > k g(n)$$

Theorem 1 *Let $u : \mathbb{R}^2 \rightarrow \mathbb{R}$ be a differentiable function and $\mathcal{G}(n, \rho(n))$ an RGG embedded in $[0,1]^2$, with $\rho(n) \in \omega\left(n^{-1/2}\right) \cap o(1)$. For a vertex v of $\mathcal{G}$, the proposed gradient approximation converges in probability to $\nabla u(v)$:*

$$\nabla u_g(v) \equiv \frac{\sum_{i=1}^{N} \frac{u(w_i) - u(v)}{d(v, w_i)} \mathbf{e}_{vw_i} \Delta\phi(w_i)}{\pi} \xrightarrow{\mathrm{P}} \nabla u(v).$$

Required asymptotic behavior for *ρ*(*n*):

$$\rho(n) \in o(1) \Rightarrow \lim_{n \to +\infty} \rho(n) = 0$$

$$\rho(n) \in \omega\left(n^{-1/2}\right) : \ \forall \sqrt{k} > 0, \ \rho(n) > \sqrt{\frac{k}{n}} \Leftrightarrow n\rho^2(n) > k, \ \text{i.e.} \ \lim_{n \to +\infty} n\rho^2(n) = +\infty$$

Asymptotic analysis of error in gradient approximation for RGGs

Threefold approximation leads to threefold error:

1. Directional derivatives along edges are approximated with difference quotients.
2. The difference quotient along each edge is used as a constant approximation of all directional derivatives in those directions that “fall into” the respective neighbor angle.
3. The unit vector in the direction of each edge is used as a constant representative for all directions that “fall into” the respective neighbor angle.

$$\mathcal{S} = \sum_{i=1}^{N} \frac{u(w_i) - u(v)}{d(v, w_i)} \, \mathbf{e}_{vw_i} \, \Delta\phi(w_i)$$

$\mathcal{E}_1$

$$\mathcal{S}_1 = \sum_{i=1}^{N} D_{\phi(w_i)} u(v) \, \mathbf{e}_{vw_i} \, \Delta\phi(w_i)$$

$\mathcal{E}_2$

$$\mathcal{S}_2 = \sum_{i=1}^{N} \int_{\omega(w_i)}^{\omega(w_i)+\Delta\phi(w_i)} D_\phi u(v) \, \mathbf{e}_{vw_i} \, d\phi$$

$\mathcal{E}_3$

$$\mathcal{I} = \int_0^{2\pi} D_\phi u(v) \, \mathbf{e}_\phi \, d\phi$$

$$\mathcal{E} = \mathcal{E}_1 + \mathcal{E}_2 + \mathcal{E}_3$$

Theorem 2 *Let* $u : \mathbb{R}^2 \to \mathbb{R}$ *be a differentiable function and* $\mathcal{G}(n, \rho(n))$ *an RGG embedded in* $[0,1]^2$*, with* $\rho(n) \in \omega\left(n^{-1/2}\right) \cap o(1)$*. For every vertex* v *of* $\mathcal{G}$*, if we define* $\mathcal{E} = \mathcal{S} - \mathcal{I}$*, it holds that*

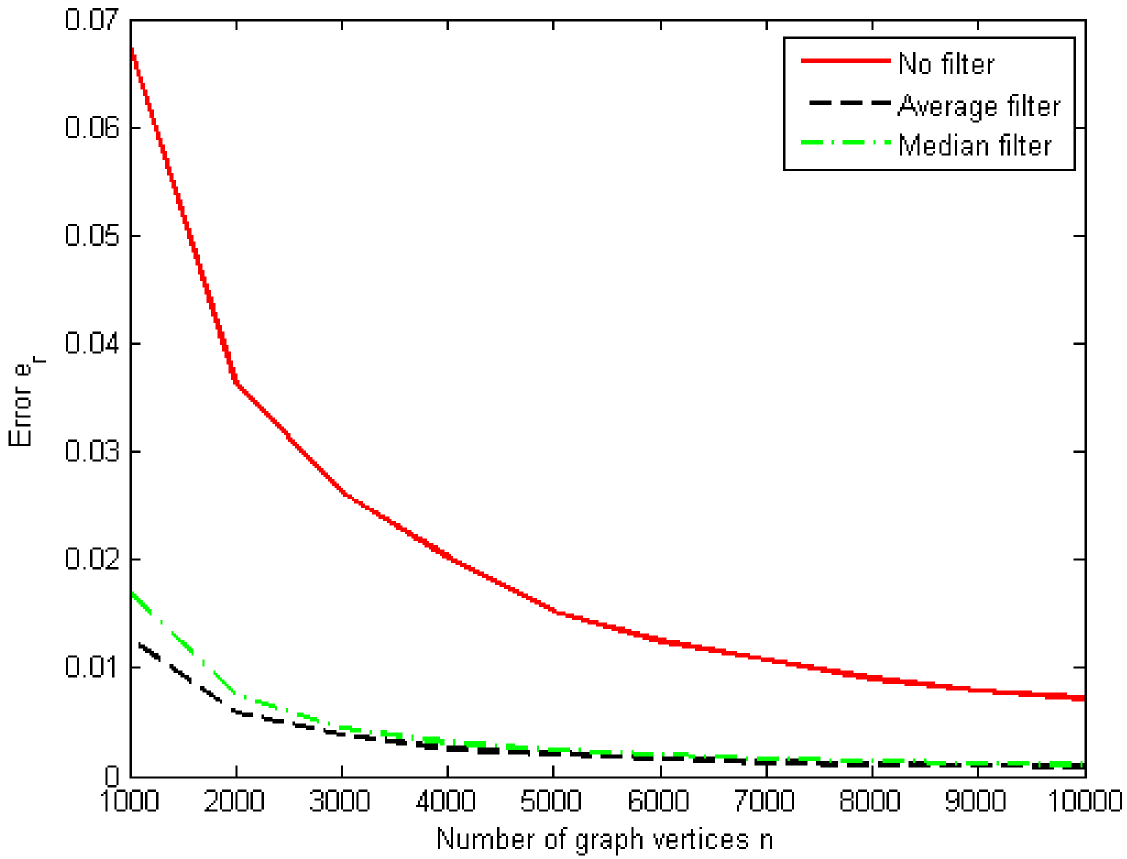
$$\mathrm{E}[\|\mathcal{E}\|] \in O\left(\rho(n) + \frac{1}{n\,\rho^2(n)}\right).$$

$\rho(n) \in \Theta(n^{-a}),\ a \in (0, 1/2)$ ⇒ **Strictest bound**: $O\left(n^{-1/3}\right)$

Smoothing filtering (on the gradient estimate)

- Geometric approximation exhibits **local deviations** from true gradient field
- For **smooth** input functions *u*, **average** or **median filtering** at neighborhood level can ameliorate the approximation.

$$u(x,y) = \exp\left(-\frac{(x-x_0)^2 + (y-y_0)^2}{2\sigma^2}\right)$$



Curvature approximation on graphs

Based on approximate gradient values: a **cascaded** approximation

$$\kappa(v) = \operatorname{div}\mathbf{F}(v) = \underbrace{\lim_{S\to\{\mathbf{v}\}} \frac{\oint_{\Gamma(S)} \mathbf{F}\cdot\mathbf{n}\,d\ell}{|S|}} = \underbrace{\frac{\partial F_1(v)}{\partial x} + \frac{\partial F_2(v)}{\partial y}}, \quad \nabla u(v) \neq \mathbf{0}$$

$$\mathbf{F} = \frac{\nabla u}{\|\nabla u\|}$$

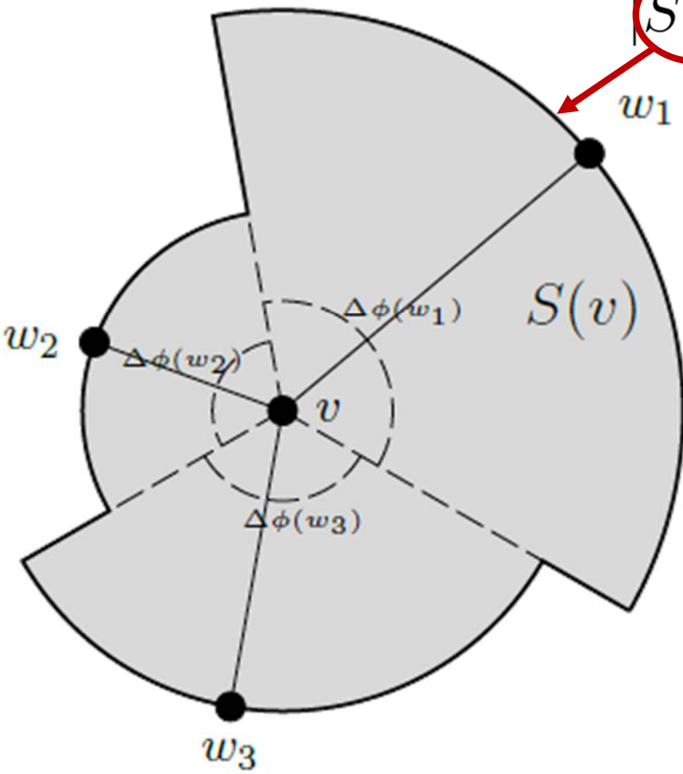
Approach 1: **geometric**

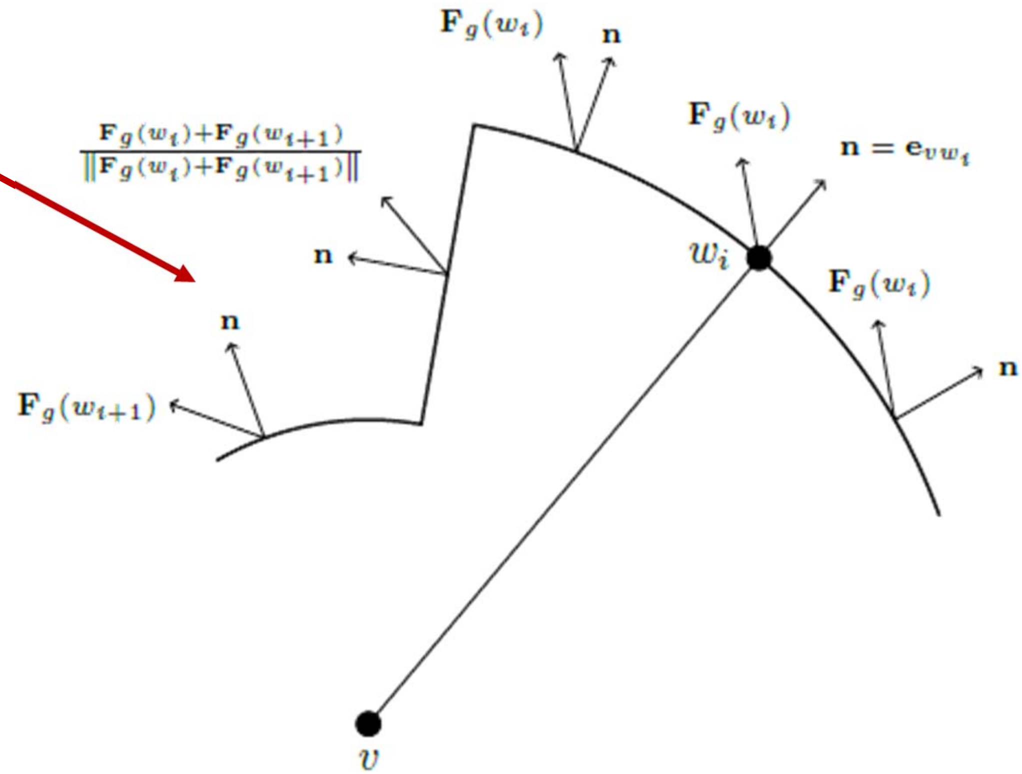
Approach 2: **gradient-based**

Both approaches rely on: $\mathbf{F}_g(v) \equiv \frac{\nabla u_g(v)}{\|\nabla u_g(v)\|}$

Geometric curvature approximation

$$\frac{\oint_{\Gamma(S(v))} \mathbf{F}\cdot\mathbf{n}\, d\ell}{|S(v)|}$$





$$|S(v)| = \sum_{i=1}^{N} \frac{\Delta\phi(w_i)}{2}\, d^2(v, w_i)$$

Constant value of approximate unit gradient field along each arc and line segment – **closed forms** for the respective line integrals:

$I_a(w_i),\ I_l(w_i)$

$$\kappa(v) \approx \frac{\sum_{i=1}^{N} I_a(w_i) + I_l(w_i)}{|S(v)|}$$

Theorem 3 *Let $\mathcal{G}(n, \rho(n))$ be an RGG embedded in $[0,1]^2$, with $\rho(n) \in \omega\left(n^{-1/2}\right) \cap o(1)$ and v a vertex of $\mathcal{G}$. If $u : \mathbb{R}^2 \to \mathbb{R}$ is continuously differentiable and $\nabla u(v) \neq \mathbf{0}$, then*

$$\frac{\sum_{i=1}^{N} I_a(w_i) + I_l(w_i)}{|S(v)|} \xrightarrow{\mathrm{P}} \kappa(v).$$

Gradient-based curvature approximation

$$\kappa(v) = \frac{\partial F_1(v)}{\partial x} + \frac{\partial F_2(v)}{\partial y} \qquad \mathbf{F}_g = (F_{1,g},\, F_{2,g}) \qquad \mathbf{e}_{vw_i} = (\cos(\phi(w_i)),\, \sin(\phi(w_i)))$$

- Partial derivatives are elements of the **gradients** of the unit gradient vector field's components
- Use **geometric gradient approximation** on the approximate values of these components

$$\kappa(v) \approx \frac{\sum_{i=1}^{N} \frac{F_{1,g}(w_i) - F_{1,g}(v)}{d(v, w_i)} \cos(\phi(w_i))\Delta\phi(w_i) + \sum_{i=1}^{N} \frac{F_{2,g}(w_i) - F_{2,g}(v)}{d(v, w_i)} \sin(\phi(w_i))\Delta\phi(w_i)}{\pi}$$

Theorem 4 *Let $\mathcal{G}(n, \rho(n))$ be an RGG embedded in $[0,1]^2$, with $\rho(n) \in \omega\left(n^{-1/2}\right) \cap o(1)$ and v a vertex of $\mathcal{G}$. If $u : \mathbb{R}^2 \to \mathbb{R}$ is twice differentiable and $\nabla u(v) \neq \mathbf{0}$, then*

$$\frac{\sum_{i=1}^{N} \frac{F_{1,g}(w_i) - F_{1,g}(v)}{d(v, w_i)} \cos(\phi(w_i))\Delta\phi(w_i) + \sum_{i=1}^{N} \frac{F_{2,g}(w_i) - F_{2,g}(v)}{d(v, w_i)} \sin(\phi(w_i))\Delta\phi(w_i)}{\pi} \xrightarrow{\text{P}} \kappa(v).$$

Gaussian smoothing

$$G_\sigma(\mathbf{x}) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{\|\mathbf{x}\|^2}{2\sigma^2}\right)$$

Approach 1: **normalized Gaussian filtering**

$$I_\sigma(\mathbf{v}) = \frac{\sum_{\mathbf{w}\in\mathcal{V}} I(\mathbf{w}) G_\sigma(\mathbf{v}-\mathbf{w})}{\sum_{\mathbf{w}\in\mathcal{V}} G_\sigma(\mathbf{v}-\mathbf{w})}$$

- Compute smoothed image explicitly
- Use approximation for its gradient

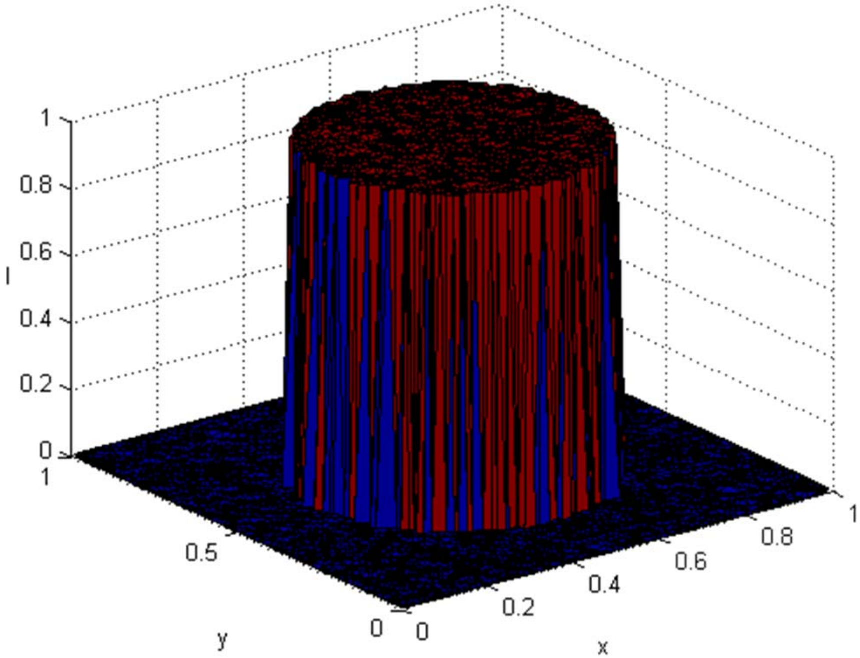
Approach 2: **Gaussian derivative filtering with separate normalization**

$$(I_\sigma)_x(\mathbf{v}) = \frac{\sum_{\substack{\mathbf{w}\in\mathcal{V}:\\ w_1 \geq v_1}} I(\mathbf{w}) \frac{\partial G_\sigma(\mathbf{v}-\mathbf{w})}{\partial x}}{\sum_{\substack{\mathbf{w}\in\mathcal{V}:\\ w_1 \geq v_1}} \frac{\partial G_\sigma(\mathbf{v}-\mathbf{w})}{\partial x}} + \frac{\sum_{\substack{\mathbf{w}\in\mathcal{V}:\\ w_1 < v_1}} I(\mathbf{w}) \frac{\partial G_\sigma(\mathbf{v}-\mathbf{w})}{\partial x}}{-\sum_{\substack{\mathbf{w}\in\mathcal{V}:\\ w_1 < v_1}} \frac{\partial G_\sigma(\mathbf{v}-\mathbf{w})}{\partial x}}$$

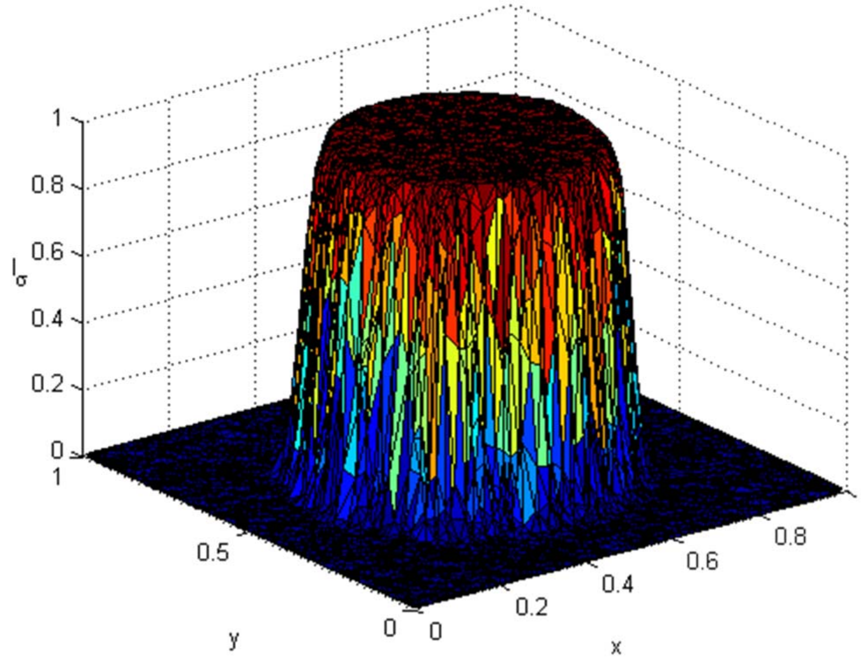
- Similar for second component
- Normalize **separately** for positive and negative part of Gaussian derivative
- Differentiation of Gaussian instead of approximation

*Plain convolution with Gaussian filter does not work due to **non-uniformities** in local vertex density*

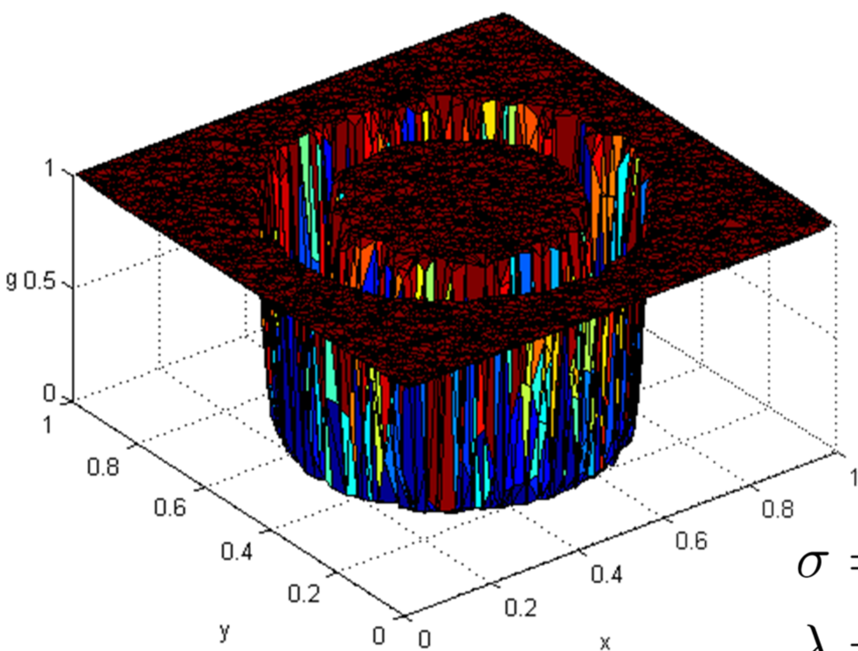
Disk image on graph



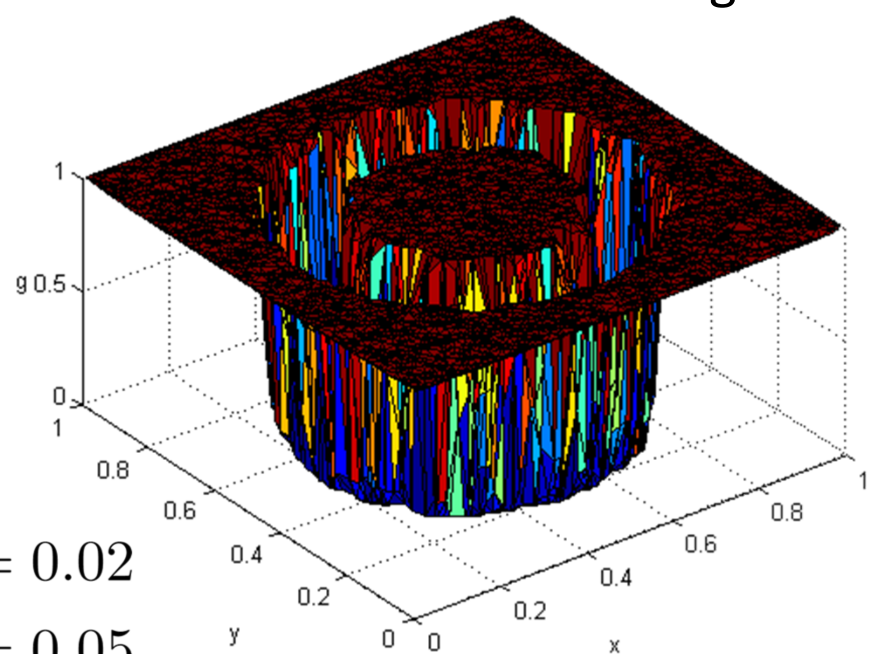
Normalized Gaussian filtering – smoothed image



Gaussian derivative filtering – stopping function



Normalized Gaussian filtering – stopping function



$\sigma = 0.02$

$\lambda = 0.05$

Final algorithm for graph segmentation with geodesic active contours

Input: $\mathcal{G} = (\mathcal{V}(\mathcal{G}), \mathcal{E}(\mathcal{G}))$ $\quad I : \mathcal{V}(\mathcal{G}) \to \mathbb{R}^+$

1. Compute stopping function using normalized Gaussian filtering or Gaussian derivative filtering with separate normalization
2. Initialize the contour's interior to a set of vertices which **contains** the clusters to be segmented and the embedding function to the **signed distance function** of this set u_0
3. Iterate until the embedding function has not changed sign at any vertex for several consecutive iterations:

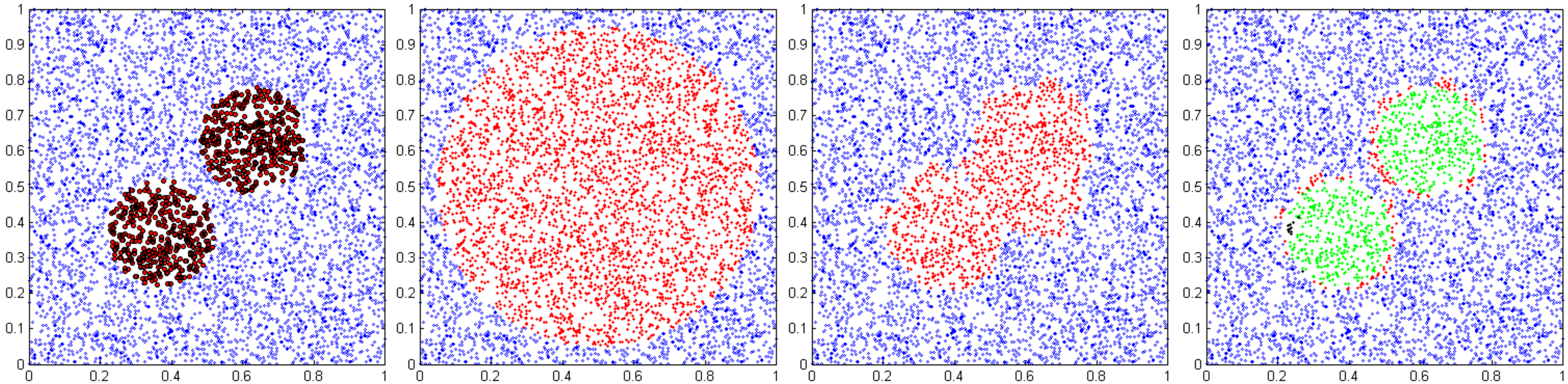
$$u_r = u_{r-1} + \Delta t((\kappa - c) \|\nabla u_{r-1}\| g + \nabla g \cdot \nabla u_{r-1}), \; r \in \mathbb{N}$$

Output: set of graph vertices where the final embedding function is positive

Parameters: $\Delta t, \, c, \, \sigma, \, \lambda$

Separation of two close disks

Disks on RGG | Initial contour | 80 iterations | 120 iterations



$n = 5500$

Legend: **true positive**, **true negative**, **false positive**, **false negative**

Radius of RGG and scale of Gaussian filter have been adjusted to help distinguish between nearby edges at boundaries of disks

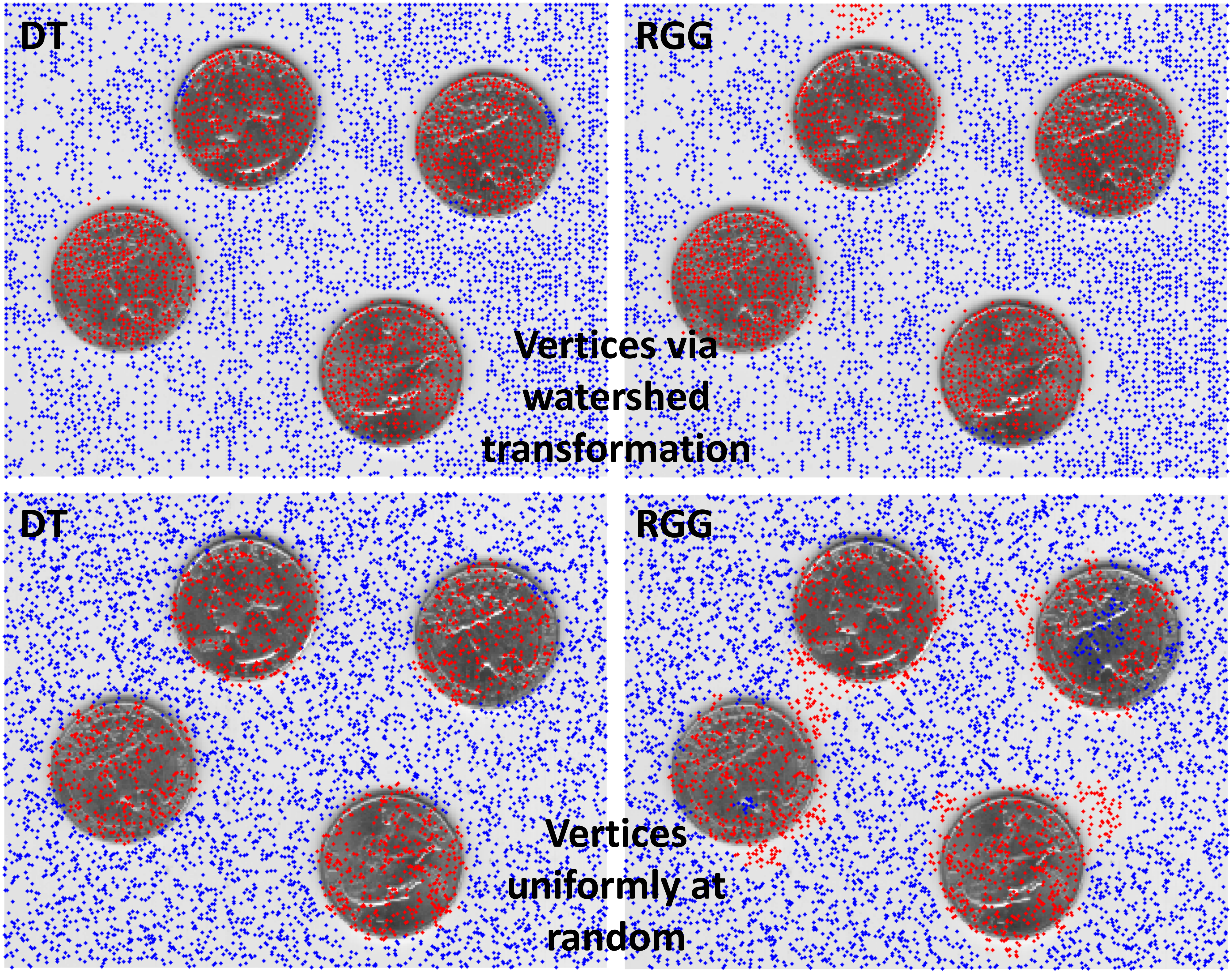
Segmentation of regular images using the graph framework

To place the vertices, **subsample** the image:

1. Uniformly at random (agnostic)
2. Based on image content: **watershed** transformation – one vertex at each segment

Alternatives for defining the set of edges:

1. Vertices closer than a threshold – radius are connected (RGG rule)
2. Delaunay triangulation (DT)


DT
RGG
Vertices via watershed transformation
DT
RGG
Vertices uniformly at random

Source of original images: BSDS500 dataset

F_1 = 95.0% F_1 = 93.2%

Performance is measured at the level of graph vertices

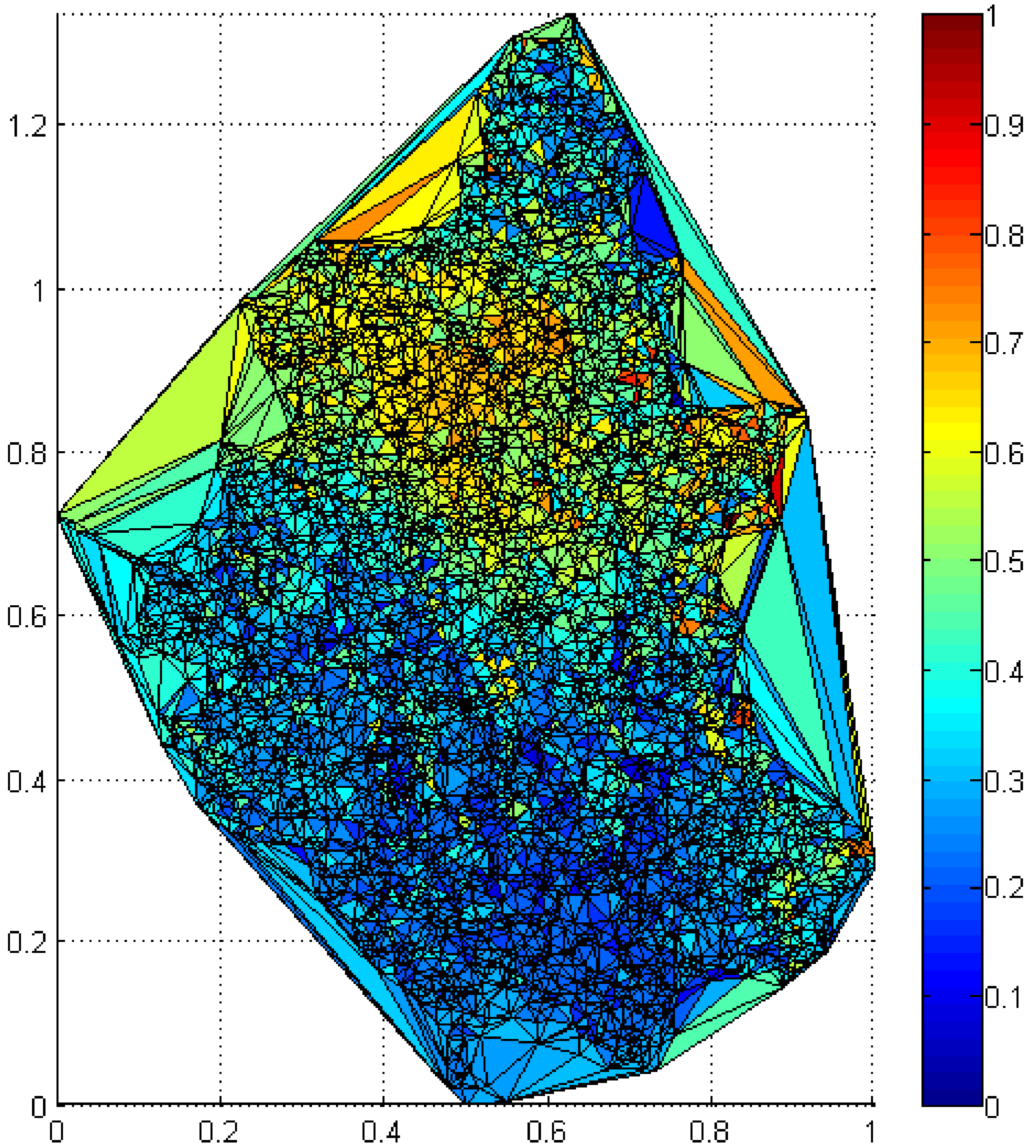
$F_1 = 94.7\%$

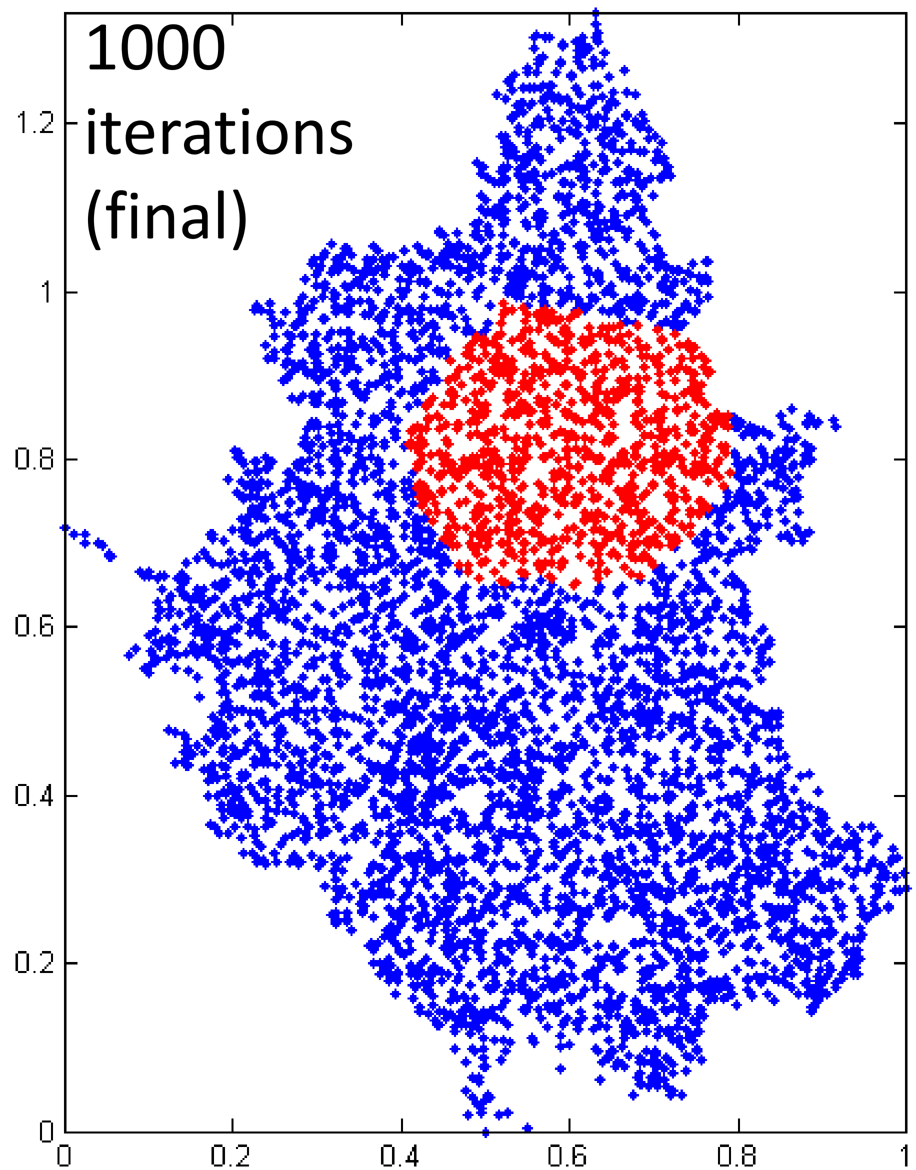
Source of images: BSDS500 dataset

$F_1 = 84.9\%$

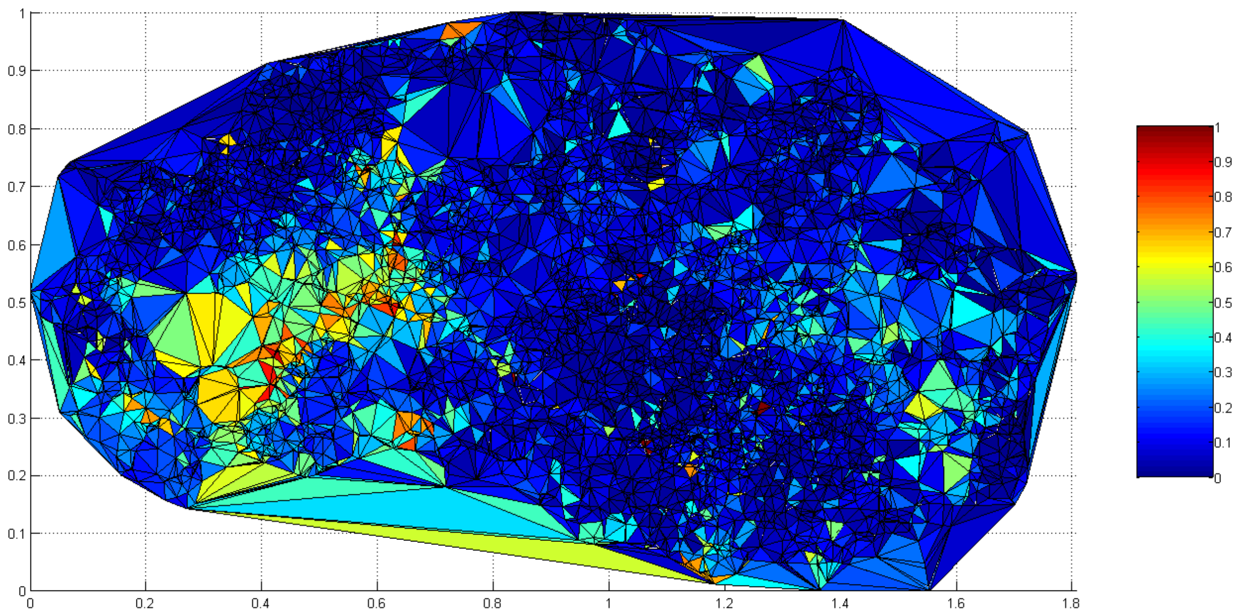
Performance is measured at the level of graph vertices

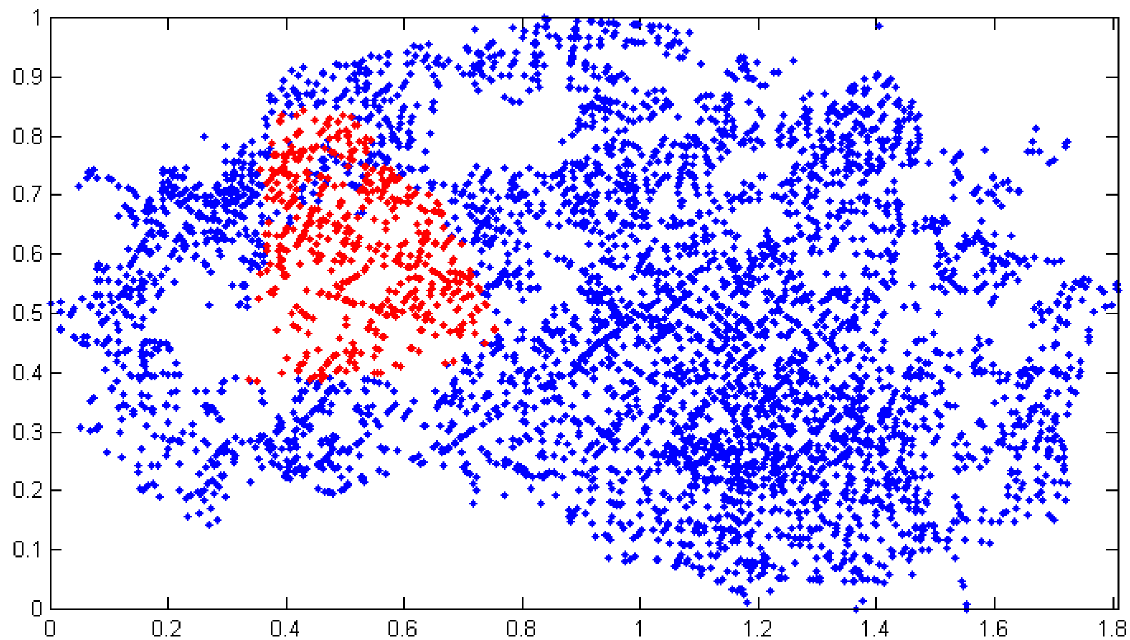
Segmentation of geographical data: average annual wind speed





Segmentation of geographical data: signal power of cellular network





40000 iterations (final)

$$\Delta t = 10^{-4}$$

Generality of approximation framework – active contours without edges (ACWE)

$$\frac{\partial u}{\partial t} = \delta_\epsilon(u)\left(\mu\kappa - \nu - \lambda_1(I - c_1)^2 + \lambda_2(I - c_2)^2\right)$$

$$c_1 = \text{average}\{I(\mathbf{x}) : \ u(\mathbf{x}) \geq 0\} \qquad c_2 = \text{average}\{I(\mathbf{x}) : \ u(\mathbf{x}) < 0\}$$

$\delta_\epsilon(x) = H'_\epsilon(x) = \dfrac{\epsilon}{\pi(x^2 + \epsilon^2)}$: regularized Dirac delta

- Like in GAC, the active contour is **embedded** in the evolving function $u(x,y,t)$ as its zero **level set**.
- Unlike GAC, there is no edge-stopping function. Instead, the image-driven force impels **discrimination of intensity values** between contour interior and exterior.

Segmentation of regular images using the graph framework with ACWE

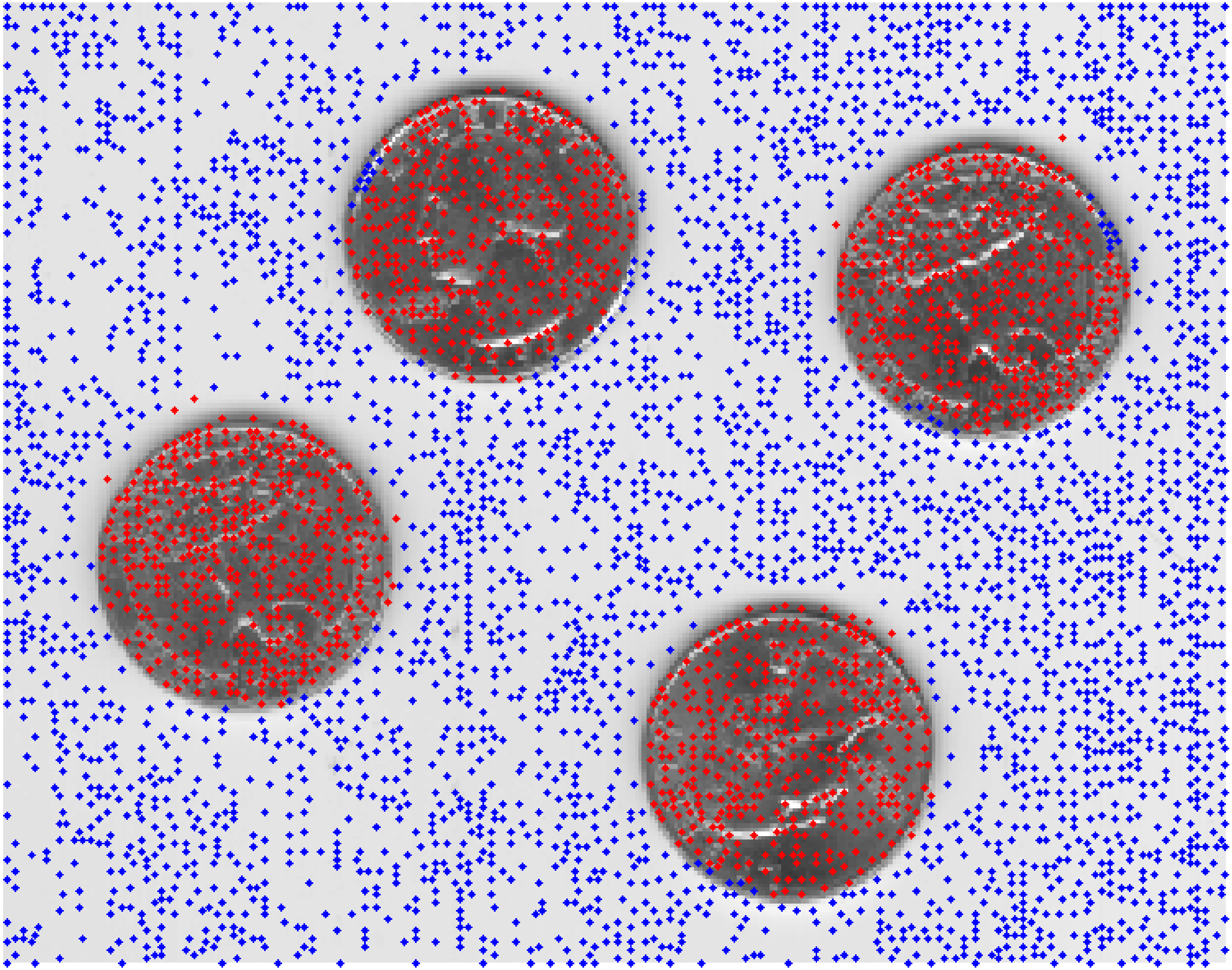
Source of images: BSDS500 dataset

Contributions

- **Novel, neighborhood-based approximations for gradient and curvature on arbitrary graphs**
- **Proofs of convergence in probability of proposed approximations to the true value of the operators for the class of random geometric graphs**
- **Asymptotic upper bound for error of gradient approximation for the class of random geometric graphs**
- **Neighborhood-based average and median filtering on graphs**
- **Two variants of Gaussian smoothing with normalization on graphs that handle non-uniform vertex distributions**
- **Applicability of our active contour framework on graphs for different level set-based models (GAC, ACWE)**

Future work

- **Further theoretical study of curvature approximation on graphs – establishment of asymptotic upper bound for error ideally**
- **Theoretical examination of graph structures that have greater regularity than random geometric graphs, such as Delaunay triangulations**
- **Anisotropic smoothing on graphs for faithful preservation of predominant edges in computation of stopping function**



Part I.C

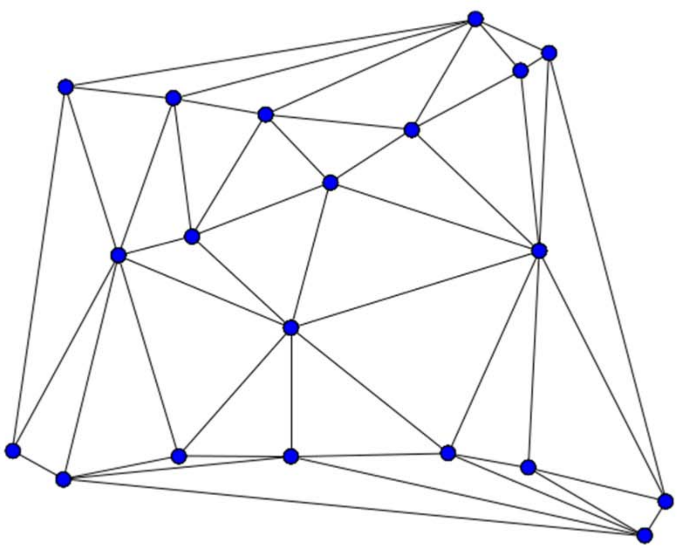
A Finite Element Computational Framework for Active Contours on Graphs

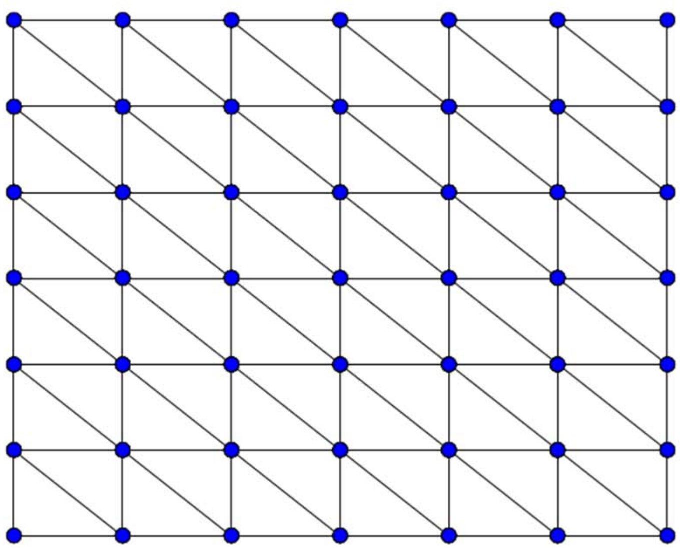
Main Reference:

- Nikos Kolotouros and P. Maragos,
  https://arxiv.org/abs/1710.04346

Delaunay Graphs

- Planar graphs that are constructed from the Delaunay triangulation of a set of points
- The Delaunay triangulation divides the convex hull of the points in triangles in a way that avoids the creation of sharp triangles.
- Good convergence properties for the solution of PDEs





GAC (Revisited)

Express the GAC evolution in a suitable form to use the Finite Element Method

$$\frac{1}{||\nabla u||}\frac{\partial u}{\partial t} = \operatorname{div}\left(g(I)\frac{\nabla u}{||\nabla u||}\right) + g(I)$$

$$\nabla u \cdot \mathbf{n} = 0 \text{ on the boundary}$$

and then transform it to an equivalent integral form

$$\iint_\Omega \frac{1}{||\nabla u||}\frac{\partial u}{\partial t}\phi = -\iint_\Omega g(I)\frac{\nabla u}{||\nabla u||}\cdot\nabla\phi + \iint_\Omega \beta\phi$$

$$\forall \phi \in H^1(\Omega)(\text{Sobolev space})$$

Galerkin approximation

Let S_1 be an n-dimensional subspace of $H^1(\Omega)$ with basic functions $\{\phi_i\}_{i=1}^n$

We approximate u by

$$\bar{u} = \sum_{i=1}^{n} c_i(t)\phi_i\,(x, y)$$

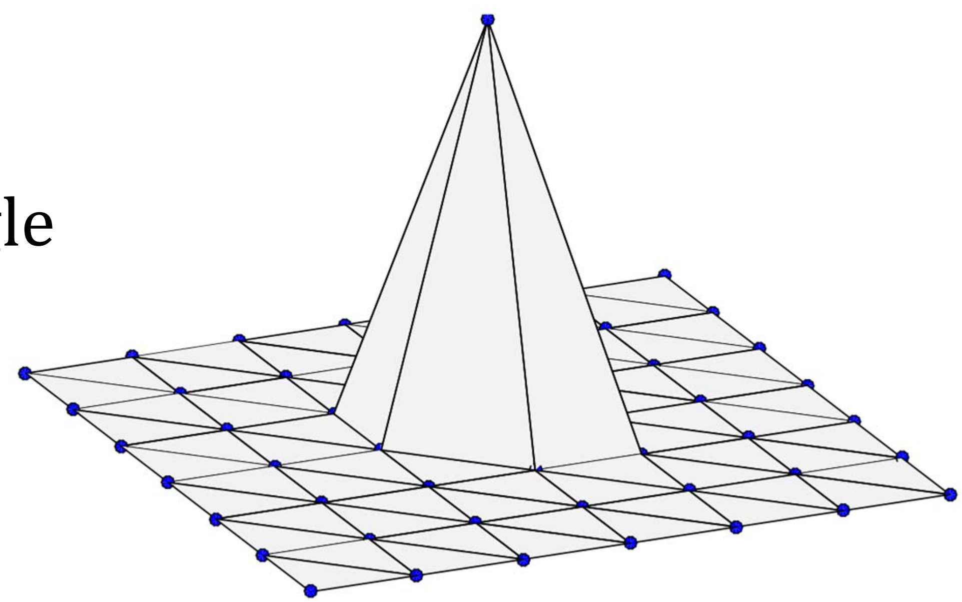
If we substitute $\bar{u}$ in the integral equation and demand that it holds for all ϕ_i in this subspace, we obtain a nonlinear system of ODEs,

$$\mathbf{A}(\mathbf{c})\dot{\mathbf{c}} = \mathbf{b}(\mathbf{c})$$

Choice of subspace and basis

- Approximation in the space of continuous, piecewise linear functions
- The basis functions should have minimal common support to simplify calculations
- We choose the pyramid functions with the following properties

- $\phi_i(v_i) = 1$
- $\phi_i(v_j) = 0, j \neq i$
- ϕ_i is linear in each triangle



Computational complexity

- **A** is a $N \times N$ sparse band matrix with a band length $\sqrt{N}$
- Solution of the system of ODEs

- Solution of the system of ODEs using the explicit Euler method requires $O(N^2)$ operations per time step

- Too expensive for large graphs!

- Later on we will present a method to reduce it to linear complexity

General active contour models

- The proposed can solve more general active contour models of the form

$$F(u)\frac{\partial u}{\partial t} = \mathrm{div}(G(u)\nabla u) + H(u)$$

- We can build on the previous framework to solve locally constrained active contour models of the form

$$F(u)\frac{\partial u}{\partial t} = \delta_\epsilon(u)\big(\mathrm{div}(G(u)\nabla u) + H(u)\big)$$

- Active Contours Without Edges are a special case of the above model

Locally constrained active contours (1)

- Inspired from the Active Contours Without Edges
- At each time step we evolve the levelset function in a small area around the curve
- The vertices of the graph in this area are called **active points**
- Using an appropriate transformation, locally constrained models require $O(N)$ computations on average per time step, because we only need to consider the submatrix of the active points

Locally constrained active contours (2)

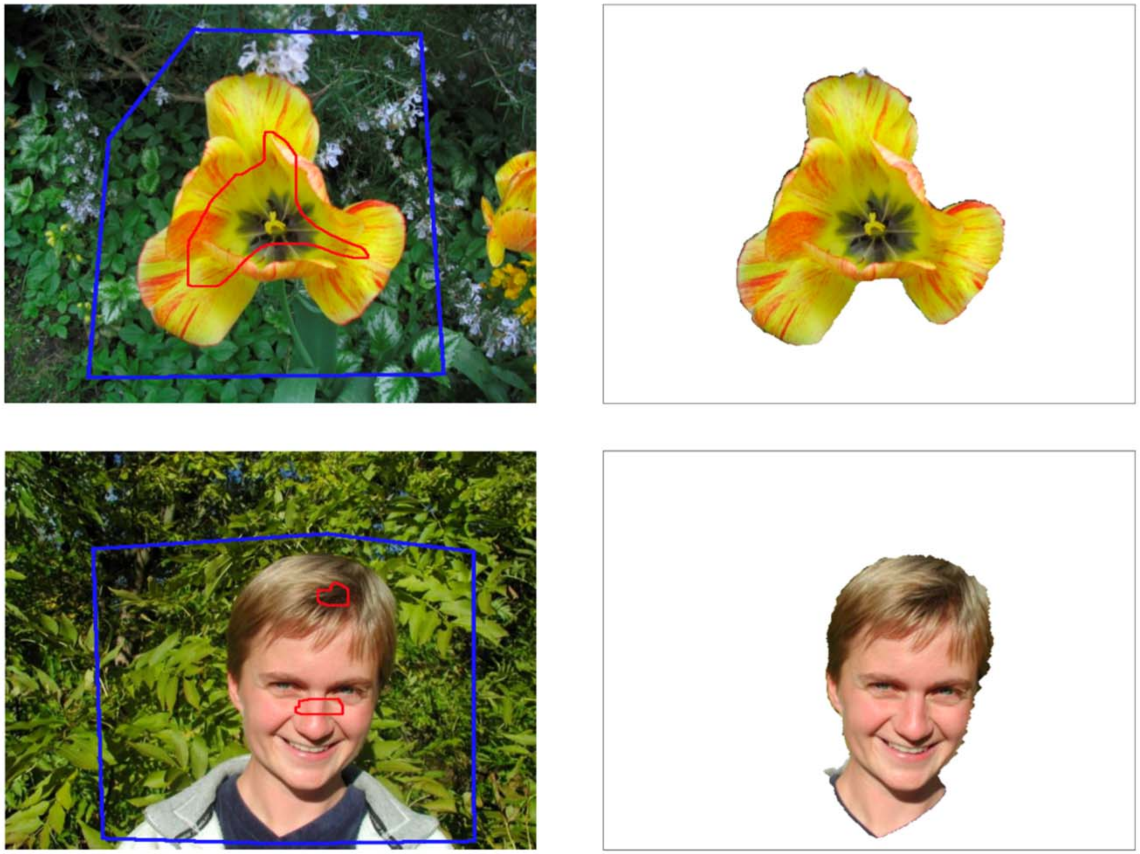
- This modified framework can be used to speed up significantly our initial Active Contours algorithm

➢ **Important observation:** the levelset function evolves in a way such that points outside or inside the curve will not change status until the moment that the active contour reaches them.

- As a result, general Active Contour models can be approximated by their locally constrained counterpart.

- $O(N)$ instead of $O\left(N^2\right)$ computations

Results - Geodesic Active Regions



Example of supervised segmentation.
Area inside the red contour: Foreground seeds.
Area outside the blue contour: Background seeds

Some numbers...

TABLE I
COMPARISON OF THE METHODS ON THE GRABCUT DATASET

Method	RI (↑)	IoU (↑)	VoI (↓)	Error (↓)
GC	0.7861	0.8796	0.5419	5.869 %
LC	0.7763	0.8671	0.5642	6.208 %
PW	0.7171	0.8358	0.6768	7.977 %
RW	0.7200	0.8343	0.6652	7.854 %
CV	0.2899	0.4833	1.2244	24.828 %
Ours	0.7268	0.8519	0.6704	7.793 %

TABLE II
COMPARISON OF THE METHODS ON THE PASCAL DATASET

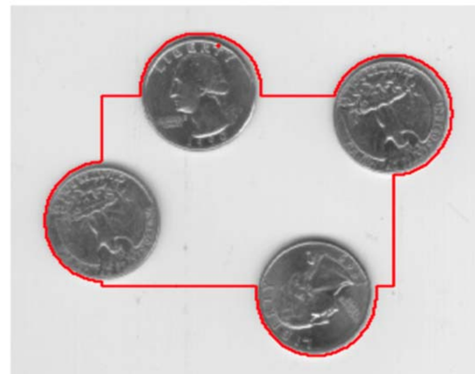
Method	RI (↑)	IoU (↑)	VoI (↓)	Error (↓)
GC	0.6939	0.8321	0.7113	8.945 %
LC	0.5861	0.7566	0.8834	12.421 %
PW	0.5683	0.7639	0.9345	12.926 %
RW	0.3898	0.6872	1.1578	20.329 %
CV	0.2045	0.4142	1.2056	29.744 %
Ours	0.6858	0.8317	0.7266	9.309 %

Results – Comparison with GrabCut

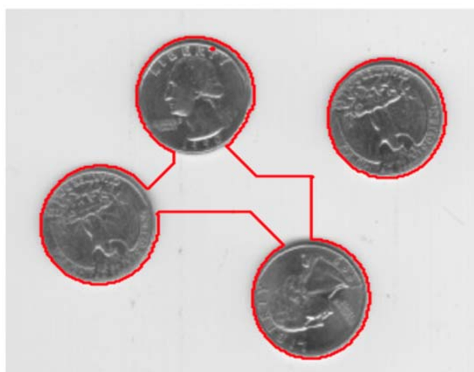
Columns 1-4: GrabCut dataset. Columns 5-8: Pascal dataset
Top row: Ground truth. Middle row: GrabCut
Bottom row: GAR

Results - Active Contours Without Edges

(a) Initial Contour



(b) 30 iterations

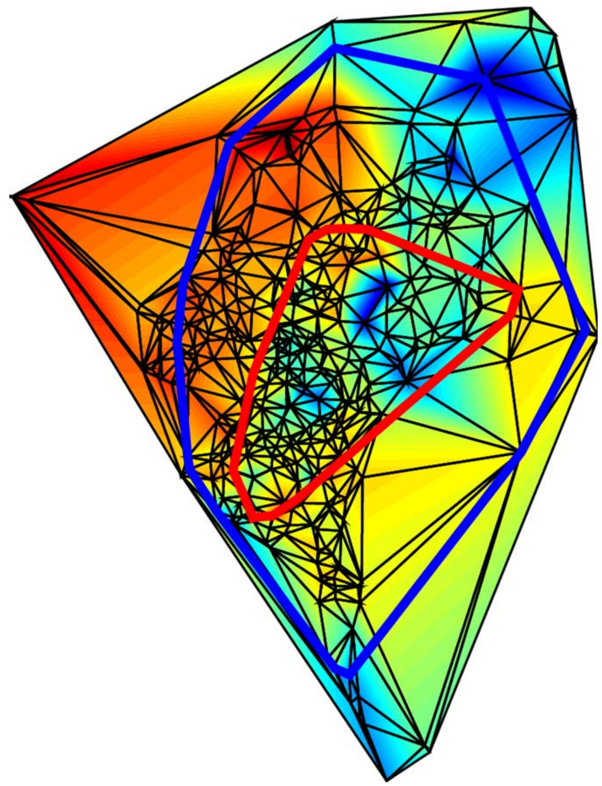


(c) 80 iterations

(d) Converged after 107 iterations

ACWE contour evolution using our framework

Results - GAC on graphs



Example of segmentation in a Delaunay graph.
Blue curve: Initial position of the contour.
Red curve: Position of the contour after convergence.

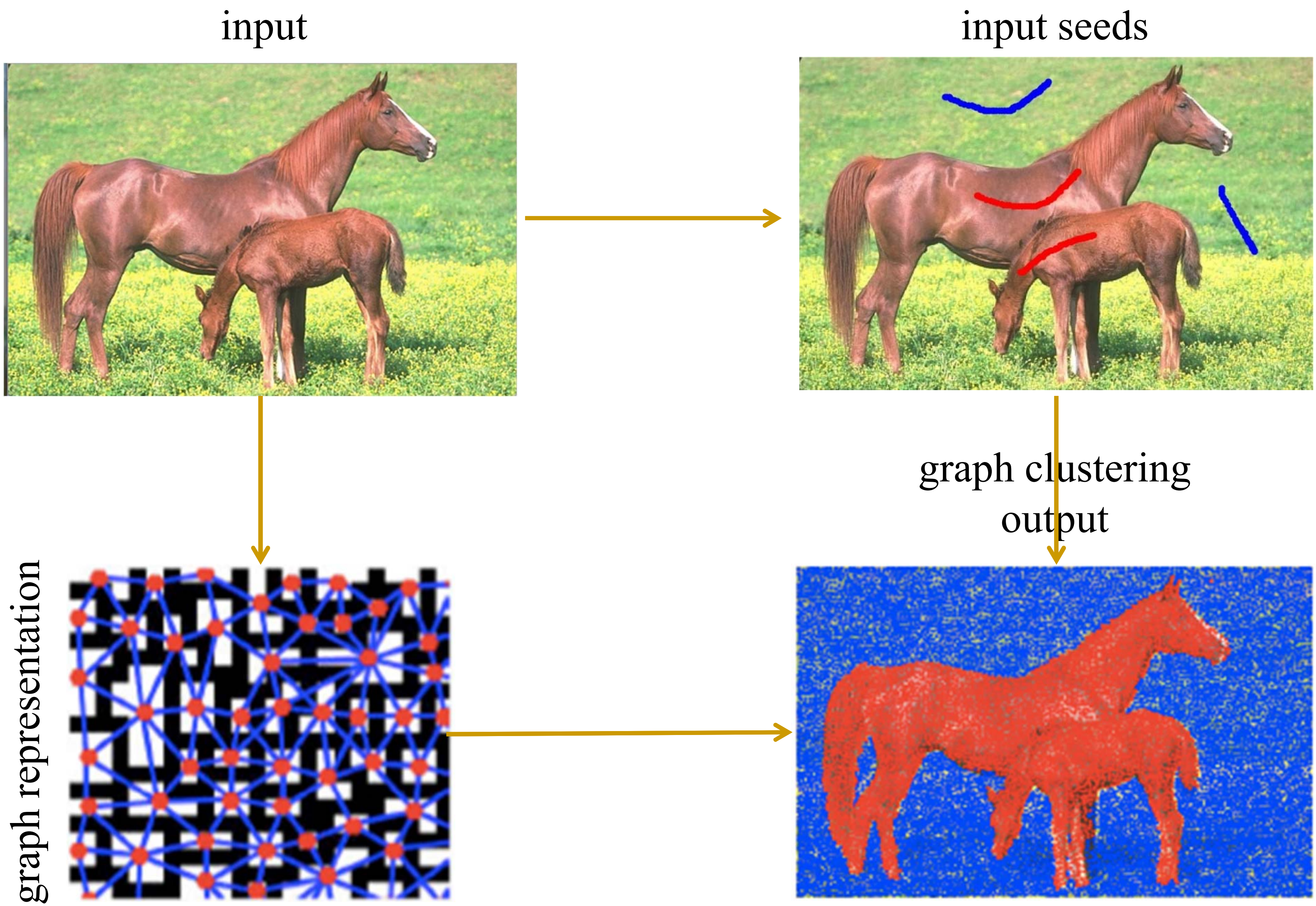
Part II.

Graph-driven Diffusion and Random Walk Schemes for Image Segmentation

Main Reference:

- Christos Bampis, P. Maragos and A.C. Bovik,
  IEEE Transactions Image Processing, vol.26, pp.35-50, Jan. 2017

Supervised Graph Clustering



main references

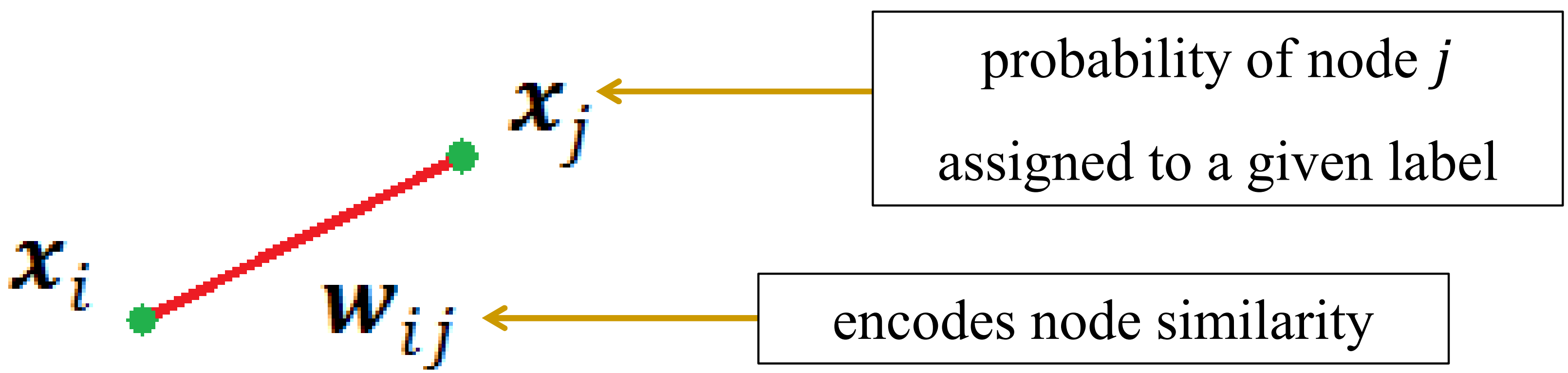
- Random Walks for Image Segmentation
  - L. Grady, "Random Walks for Image Segmentation," IEEE Trans. PAMI, 2006
- Power Watershed
  - C. Couprie, L. Grady, L. Najman, and H. Talbot, "Power watershed: A unifying graph-based optimization framework," IEEE Trans. PAMI., 2011.
- SIR epidemic modeling
  - E. B. Postnikov and I. M. Sokolov, "Continuum description of a contact infection spread in a SIR model," Math. Biosci., 2007.
- Supervised Graph Clustering/Learning
  - D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," in Proc. NIPS, 2004
  - X. Zhu, Z. Ghahramani, and J. Lafferty, "Semi-supervised learning using Gaussian fields and harmonic functions," in Proc. ICML, 2003
- Regularization on arbitrary graphs
  - A. Elmoataz, O. Lézoray, and S. Bougleux, "Nonlocal discrete regularization on weighted graphs: A framework for image and manifold processing," IEEE Trans. Imag. Proc., 2008.

The Random Walker Algorithm

- Considers the trajectories of a random walker starting from un-marked pixels and ending at marked points (seeds). Each unmarked pixel is assigned to the most probable type of seed.
- The original formulation is computationally hard; instead minimize:

$$J(\boldsymbol{x}) = \frac{1}{2}\sum_{i,j=1}^{N} w_{ij}\left(x_i - x_j\right)^2 = \frac{1}{2}\boldsymbol{x}^{\boldsymbol{T}}\boldsymbol{L}\boldsymbol{x}$$

given the seeds, the weights w_{ij}, the number of nodes N and the Graph Laplacian $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}$, where $\boldsymbol{D} = [d_i]$ is the degree matrix, $d_i = \sum_{j\sim i} w_{ij}$, $j\sim i$ denotes that j is adjacent to i and $\boldsymbol{W} = [w_{ij}]$



- The Random Walker (RW) is related to graph-diffusion: it minimizes the smoothness term $J(\boldsymbol{x})$ to derive probabilities $\boldsymbol{x}$ for each seed type.

- $\boldsymbol{x}$ satisfies the following:
  - sums to 1 over all seeds
  - $\in [0, 1]$
  - $x_i = \frac{1}{d_i} \sum_{j \sim i} w_{ij} x_j$

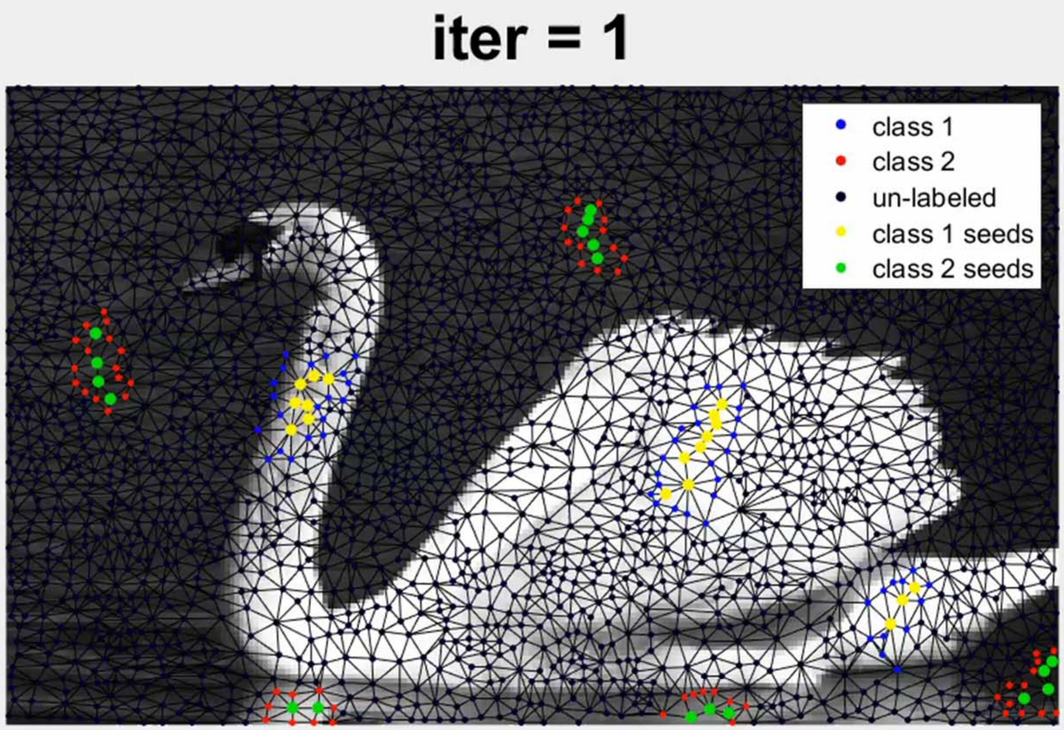
well-defined probability

(combinatorial) harmonic function

- w_{ij} models node similarity: $w_{ij} = exp\left[\frac{\|g_i - g_j\|^2}{\sigma_g^2}\right]$, g_i: color feature vector for i

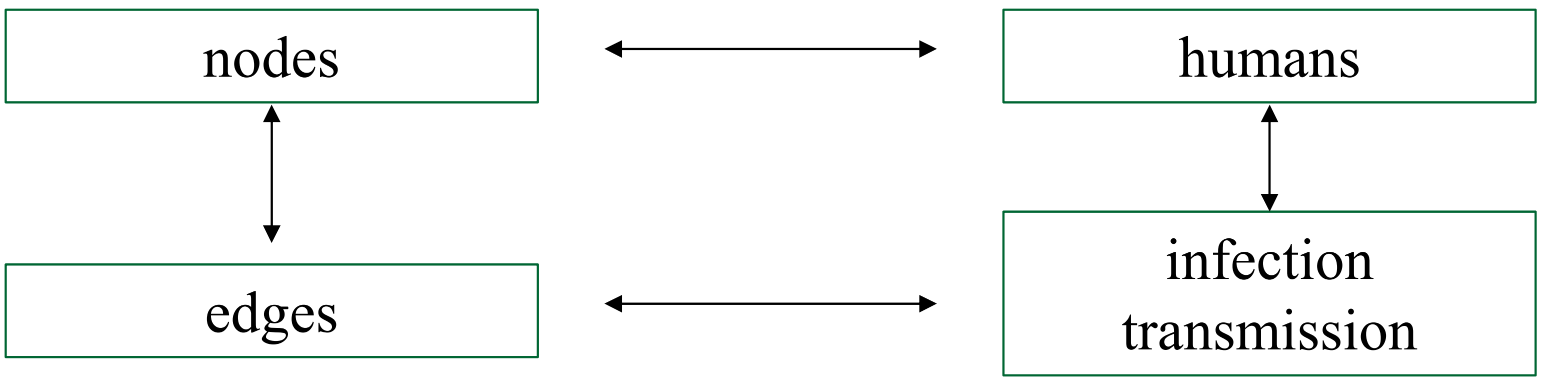
- RW can be extended to arbitrary graphs and use other feature types

RW Demo



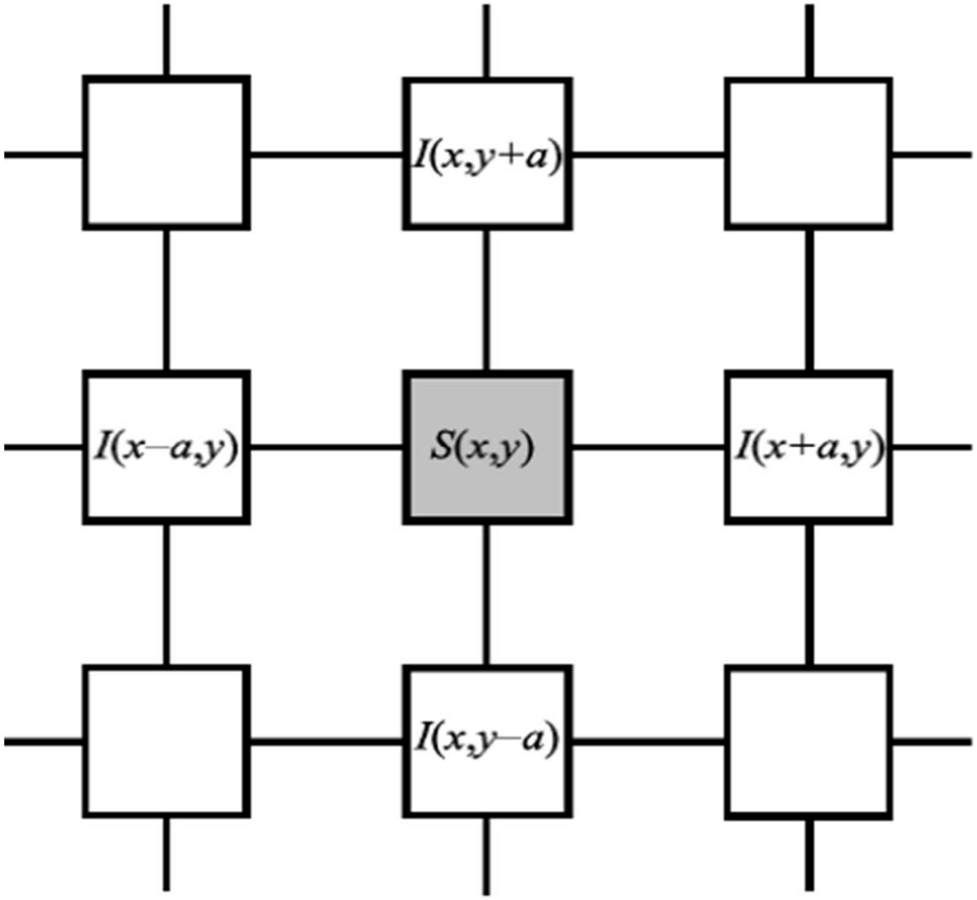
RW and Epidemiological Analogues

- RW considers the interactions between nodes: if w_{ij} is large, x_i and x_j will be similar
- An analogue of node interactions can be made in epidemiological modeling:



- The graph diffusion underlying RW can be formulated as the evolution of infectious wavefronts governed by a contact/local infection mechanism: the more similar or close friends two persons are, the more likely they are to infect each other.

Susceptible-Infected-Recovered (SIR) model and RW



$S(x,y)$: probability of (x,y) to be susceptible

$I(x,y)$: probability of (x,y) to be infected

k is the similarity

SIR epidemic propagation modeling: E. B. Postnikov and I. M. Sokolov, "Continuum description of a contact infection spread in a SIR model," Math. Biosci., 2007.

SIR Diffusion: $$\Delta I(x,y,t) = \frac{k}{4} S(x,y,t)[4I(x,y,t) + \alpha^2 \nabla^2 I(x,y,t)]\Delta t$$

RW Diffusion: $$I_{i,t+1} = I_{i,t} + \sum_{j \sim i} \frac{w_{ij}}{d_i}(I_{j,t} - I_{i,t})$$

- Graph-based RW diffusion:

$$I_{i,t+1} = I_{i,t} - \Delta_{i,t}$$

infection probability of node i at time t

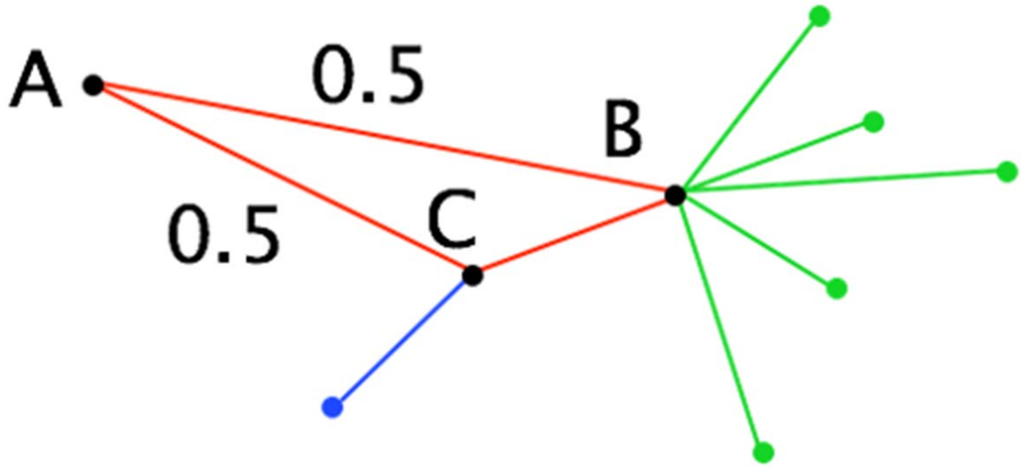
$$\sum_{j\sim i} \frac{w_{ij}}{d_i}\left(I_{i,t} - I_{j,t}\right)$$

- Intuition: $I_{i,t}$ changes until it becomes the weighted average of the neighboring nodes (steady state)
- Resembles a gradient descent approach.
- The RW gives a steady solution to this diffusion by minimizing:

$$x_i \longrightarrow I_i$$

$$J(\boldsymbol{I}) = \frac{1}{2}\boldsymbol{I}^T\boldsymbol{L}\boldsymbol{I}, \boldsymbol{I} = \begin{bmatrix} I_1 \\ \vdots \\ I_N \end{bmatrix}$$

Incorporating the Importance of Nodes



- The RW diffusion assumes that in the neighborhood A-B-C, B and C have an equal contribution to A and its local infection profile.
- But B is more important: it has 7 connections while C has only 3.

<u>for A,   B is a more important neighbor than C</u>

To account for the node importance, consider a different diffusion:

$$I_{i,t+1} = I_{i,t} - \Delta_{i,t}$$

infection probability of node i at time t

$$\sum_{j\sim i} w_{ij} \left(\frac{I_{i,t}}{d_i} - \frac{I_{j,t}}{\sqrt{d_i d_j}} \right)$$

Old steady state

$$I_{i,t_\infty} = \frac{1}{d_i} \sum_{j\sim i} w_{ij} \, I_{j,t_\infty}$$

RW

New steady state

$$I_{i,t_\infty} = \frac{1}{\sqrt{d_i}} \sum_{j\sim i} \frac{w_{ij}}{\sqrt{d_j}} I_{j,t_\infty}$$

Normalized RW (NRW)

t_∞: steady state

New properties:

- $I_{i,t_\infty} \notin [0, 1]$
- I_{i,t_∞} no longer sums to 1 across different seeds
- I_{i,t_∞} is no longer a weighted average of I_{j,t_∞} for $j \sim i$
- In practice, we compute I_{i,t_∞} by minimizing:

$$J_n(\boldsymbol{x}) = \frac{1}{2}\sum_{i\sim j} w_{ij}\left(\frac{x_i}{\sqrt{d_i}} - \frac{x_j}{\sqrt{d_j}}\right)^2 = \frac{1}{2}\boldsymbol{x}^{\boldsymbol{T}}\boldsymbol{L_n}\boldsymbol{x}$$

where $\boldsymbol{L_n} = \boldsymbol{D}^{-\mathbf{0.5}}(\boldsymbol{D} - \boldsymbol{W})\boldsymbol{D}^{-\mathbf{0.5}}$ denotes the Normalized Graph Laplacian

- For each node i pick the label maximizing I_{i,t_∞}.

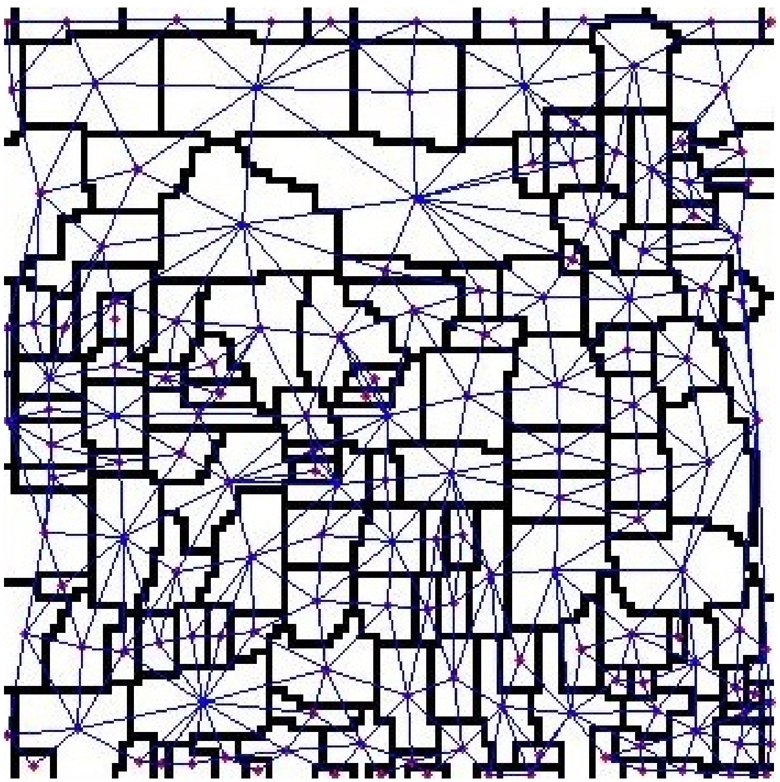
Why this is useful:

- simple and efficient extension of RW which is readily deployable
- using a node degree-aware for arbitrary graph segmentation methods, gives us more information about the local graph structure which is rich in images (images are locally correlated) and highly non-regular
- this degree-aware term can be also incorporated to other minimization schemes
- NRW is designed to adapt to any arbitrary graph structure for performing graph-based segmentation

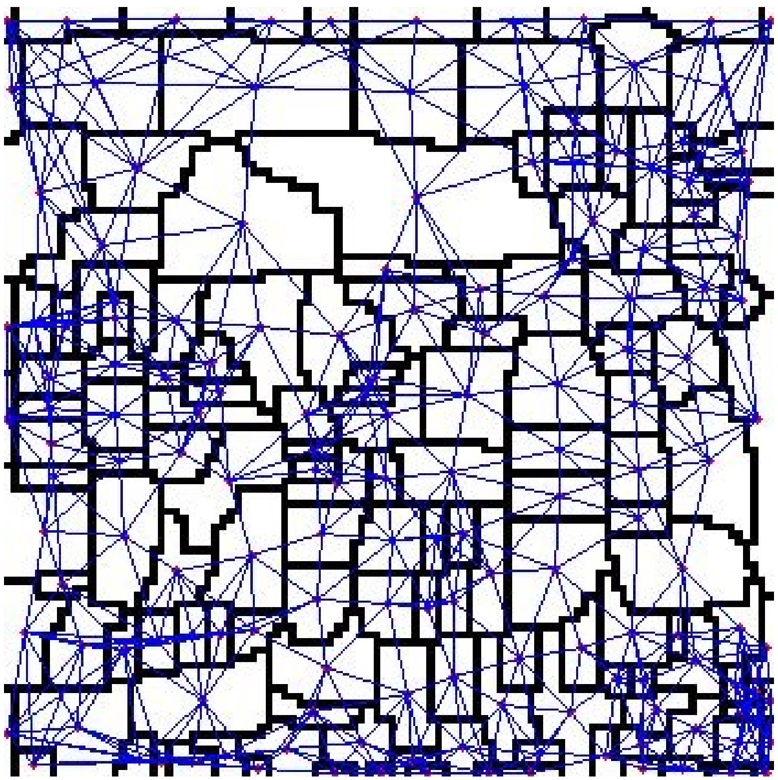
Practical Considerations

- constructing the image graph: oversegment the image using watershed, then pick nodes for each region. Assign color/texture features to each node (e.g. average RGB values in each region)
- creating the edge set (e.g. RAG or k-nn graphs)

RAG



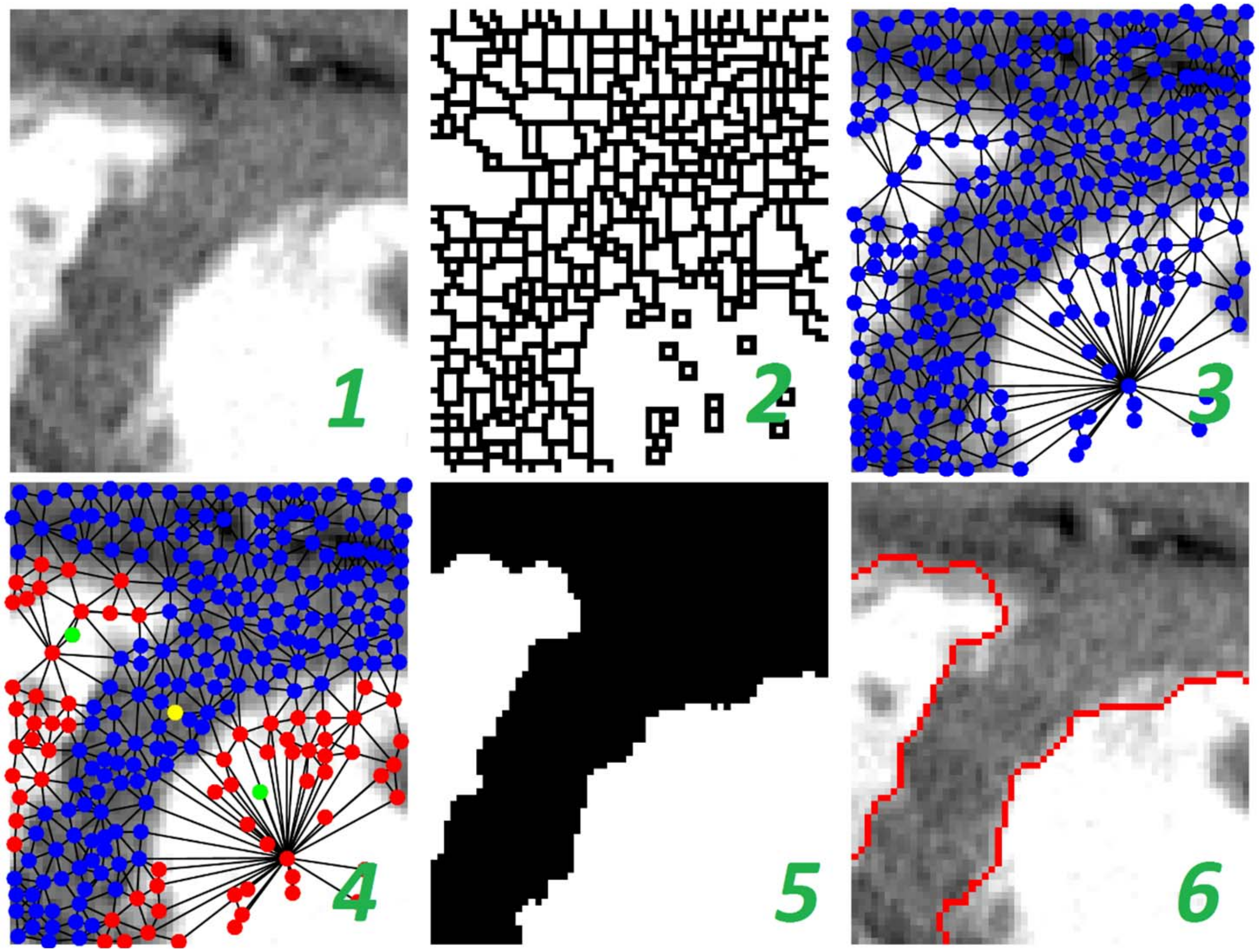
8-nn



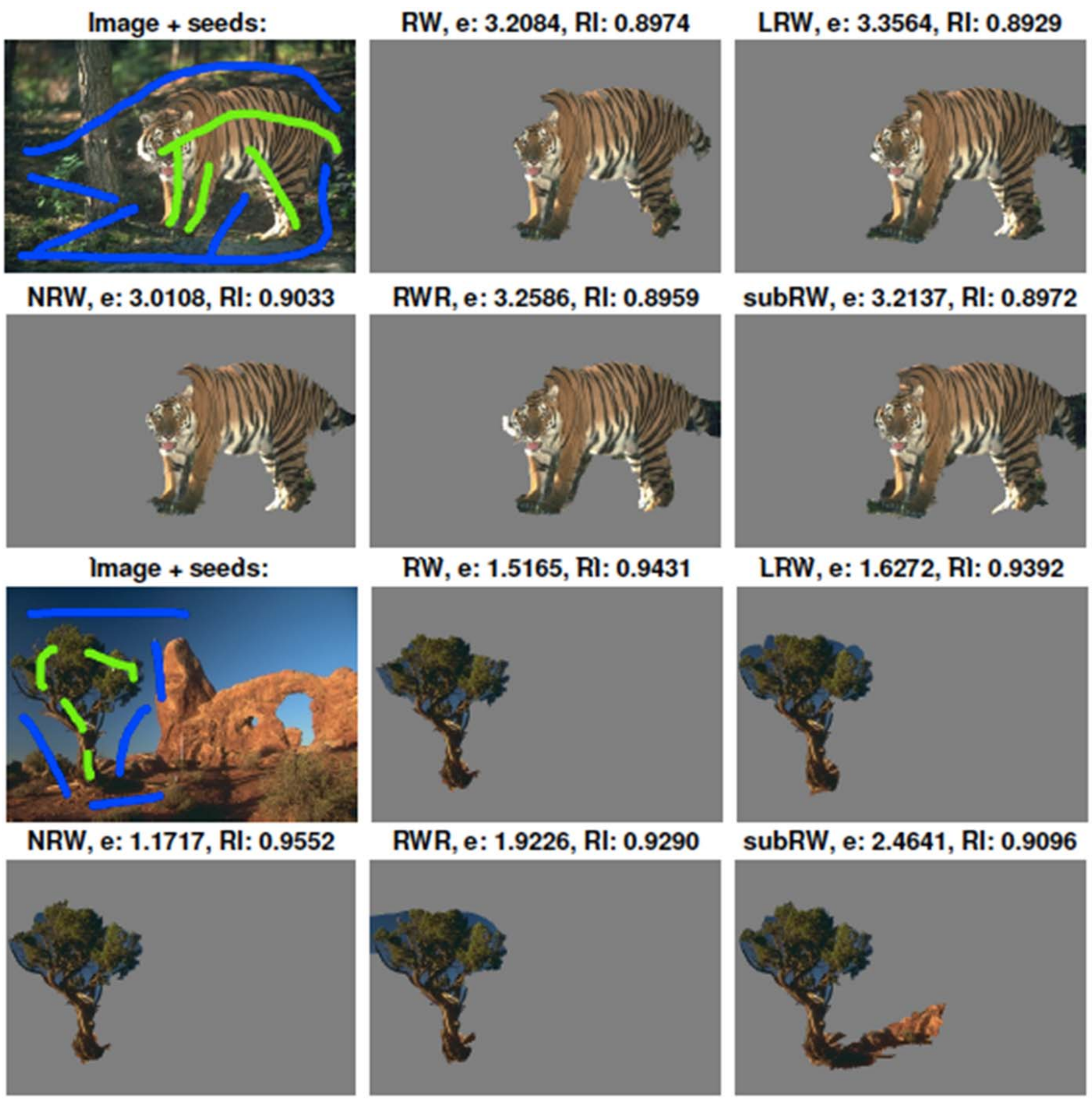
RAG: only the contingent regions are connected

Visualizing the Steps

1. original image
2. watershed transform
3. assign nodes and use RAG
4. NRW graph clustering
5. pixel level segments
6. boundaries



Qualitative Experiments (Pixel)

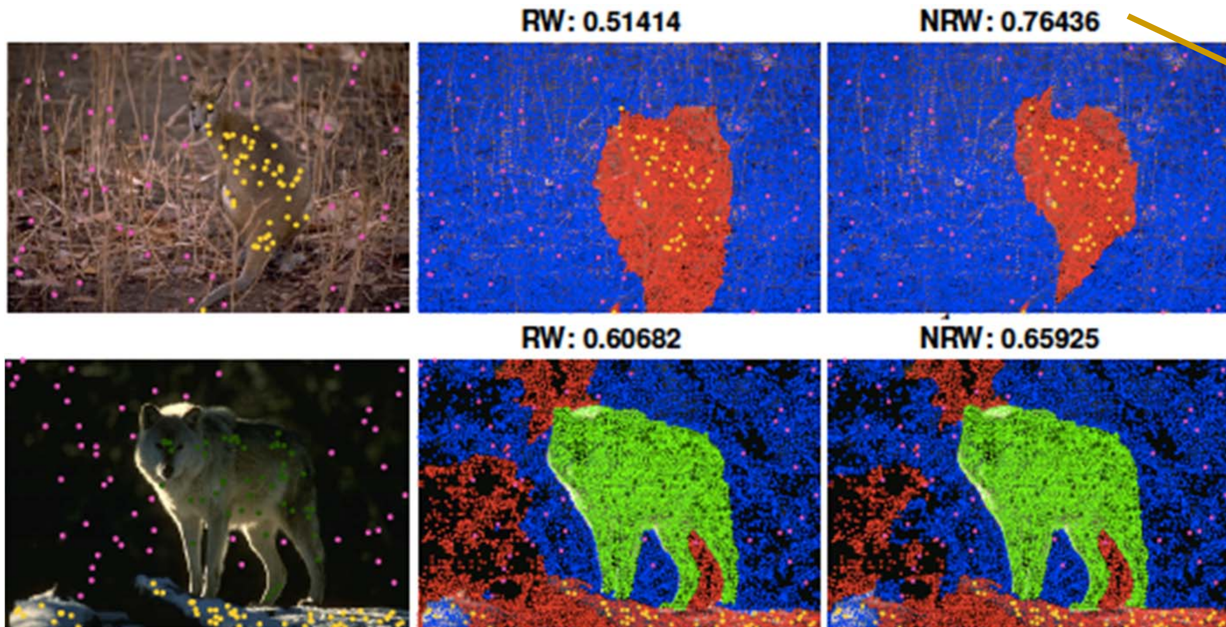


e: segmentation error rate

RI: rand index

even for pixel based methods, NRW is very competitive

Qualitative Experiments (Node)



RI
(Rand Index)

introduced erroneous seeds

both affected by seed quality (L2-norm outlier sensitivity), but NRW performs better

References of compared works

- Random Walker (**RW**)
  - L. Grady, "Random walks for image segmentation," IEEE Trans. PAMI, 2006
- Laplacian Coordinates (**LC**)
  - W. Casaca, L. G. Nonato, and G. Taubin, "Laplacian coordinates for seeded image segmentation," Proc. CVPR, 2014.
- Random Walker with Restart (**RWR**)
  - T. H. Kim, K. M. Lee, and S. U. Lee, "Generative image segmentation using random walks with restart," Proc. ECCV, 2008.
- Lazy Random Walker (**LRW**)
  - J. Shen, Y. Du, W. Wang, and X. Li, "Lazy random walks for superpixel segmentation," IEEE Trans. Imag. Proc., 2014.
- Normalized Random Walker (**NRW**) and Normalized Lazy Random Walker (NLRW)
  - C. G. Bampis, P. Maragos and A. C. Bovik, "Graph-Driven Diffusion and Random Walk Schemes for Image Segmentation," IEEE Trans. Imag. Proc., 2017.

Quantitative Experiments (Node)

(a) RI for all 6 datasets

Dataset	G_1	G_2	W_1	W_2	S	V	E
RW	0.6419	0.8528	0.7938	0.8633	0.8192	0.6518	0.7738
LRW	0.6473	0.7980	0.7954	0.8016	0.7444	0.5285	0.7134
LC	0.6787	0.8260	0.8025	0.8459	0.8111	0.6137	0.7612
RWR	0.6358	0.7809	0.7600	0.7804	0.7169	0.5018	0.6636
NRW	0.6827	0.8698	0.8401	0.8806	0.8502	0.6761	0.8081

(b) Statistical Analysis for RI

Method	RW	LRW	LC	RWR	NRW	NLRW
RW	- - - - -	- 1111	- - -1-	-1111	0-000	0 - -00
LRW	-0000	- - - - -	- -000	- - -11	00000	00000
LC	- - -0-	- -111	- - - - -	-1111	- -000	- -000
RWR	-0000	- - -00	-0000	- - - - -	00000	00000
NRW	1-111	11111	- -111	11111	- - - - -	- - - - -

1: row better than column

0: column better than row

- : statistically indistinguishable

G1 and G2: GrabCut dataset with different input seed sets

W1 and W2: Weizmann Dataset (one and two objects)

S: Semantic Dataset

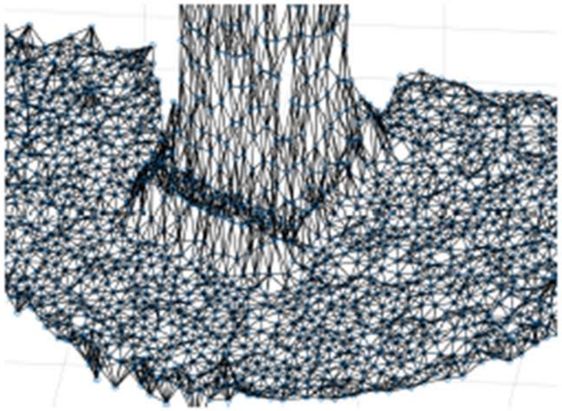
V: Pascal VOC12 Dataset

E: ECSSD Dataset

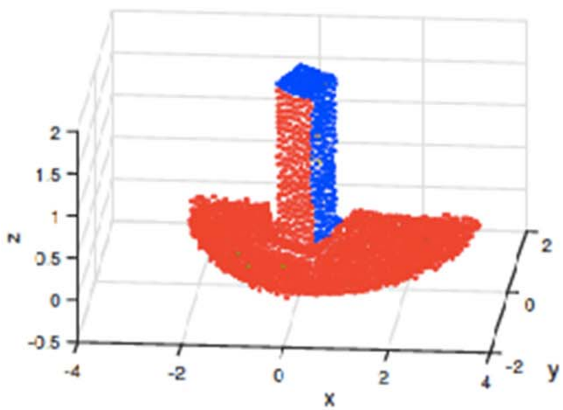
Point Clouds

original Kinect
point cloud

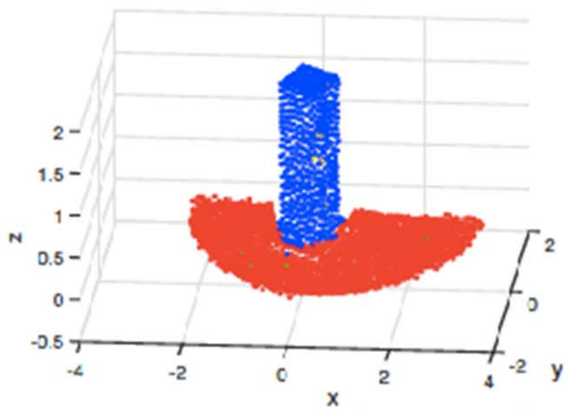
8-nn graph



RW

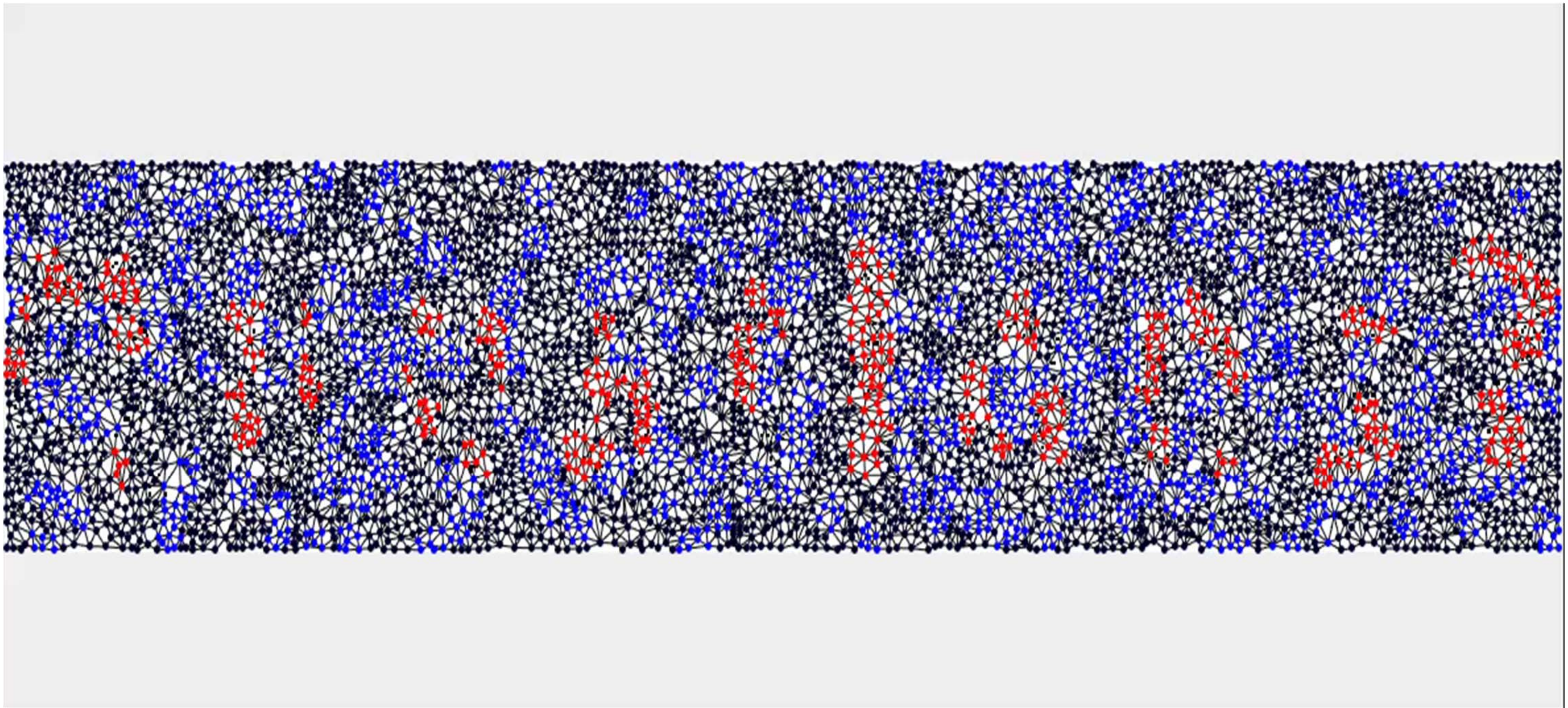


NRW



Conclusions

- I. Active Contours on Graphs: Theory
  - Novel neighborhood-based approximations for gradient and curvature on arbitrary graphs.
  - Proofs of convergence in probability of proposed approximations to true value of operators for RGGs.
  - Asymptotic upper bound for error of gradient approximation for RGGs.
  - Neighborhood-based smoothing on graphs.
  - Efficient computation with Finite elements
- II. Graph-driven Diffusion and RW Schemes
  - Use of arbitrary graphs allows us to consider data-driven representations of visual data with reduced dimensionality.
  - In this context, NRW encodes the local neighborhood statistics and delivers highly performing results.
- Promising Experimental results



For more information, demos, and current results:

http://cvsp.cs.ntua.gr and http://robotics.ntua.gr