ECCV 2006 tutorial on

Graph Cuts vs. Level Sets

part III

Connecting Graph Cuts and Level Sets

Yuri Boykov — University of Western Ontario

Daniel Cremers — University of Bonn

Vladimir Kolmogorov — University College London

Graph Cuts versus Level Sets

- Part I: Basics of *graph cuts*
- Part II: Basics of *level-sets*
- Part III: Connecting *graph cuts* and *level-sets*
- Part IV: Global vs. local optimization algorithms

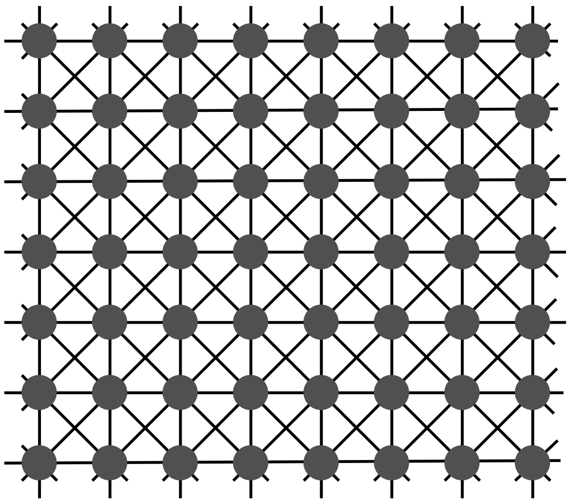
Graph Cuts versus Level Sets

- Part III: Connecting graph cuts and level sets
  - Minimal surfaces, global and local optima (VK)
  - Integral and differential approaches (YB)
  - Metrics on the space of contours, learning and shape prior in graph cuts and level-sets (DC)

Discrete vs. continuous functionals

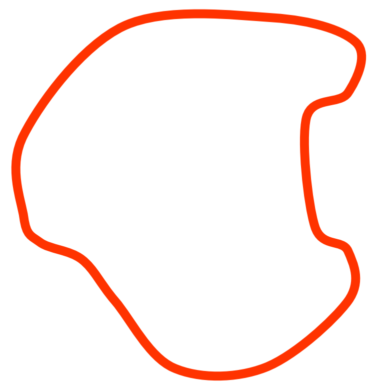
Graph cuts

$$E(\mathbf{x}) = \sum_{p} E_p(x_p) + \sum_{p,q} E_{pq}(x_p, x_q)$$



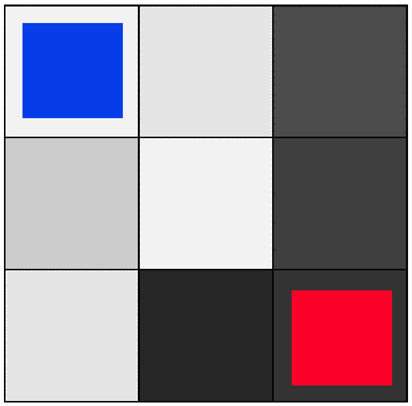
Geodesic active contours

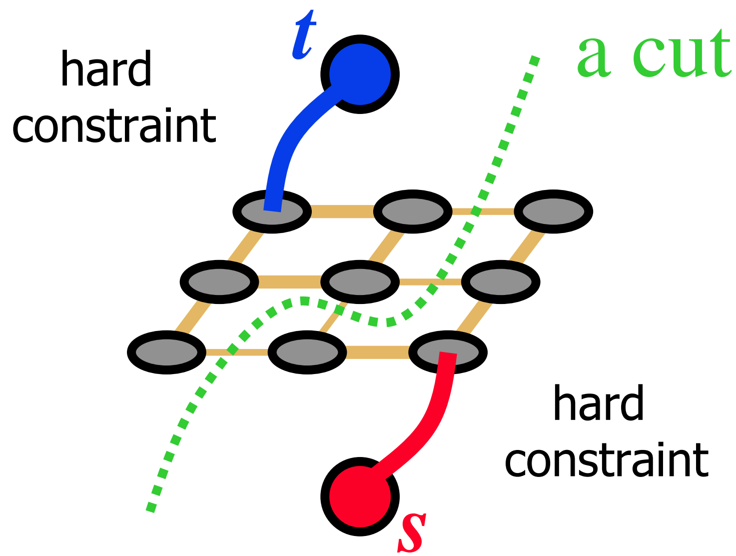
$$E(C) = \int_C g(C(s), \vec{N}) \; ds$$



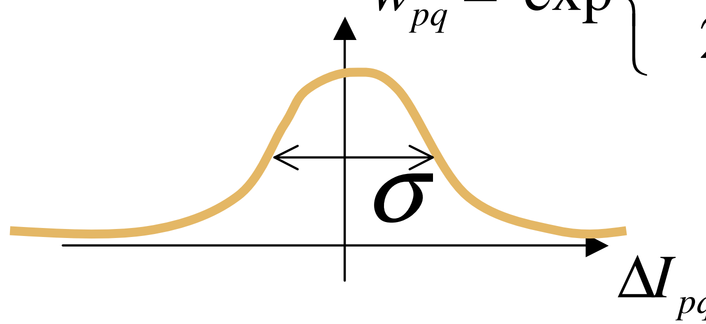
- Both can incorporate basic segmentation cues
  - Image contrast
  - Regional bias
  - Alignment (flux)

Incorporating image contrast: graph cuts [Boykov&Jolly'01]



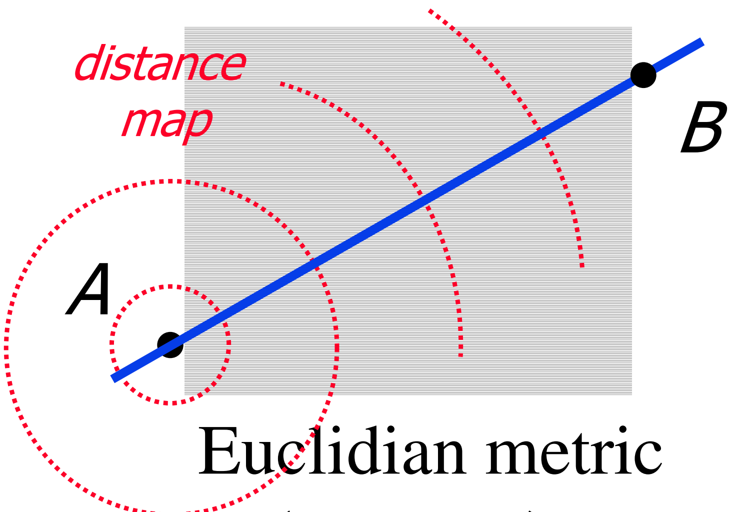


$$w_{pq} = \exp\left\{ -\frac{\Delta I_{pq}}{2\sigma^2} \right\}$$

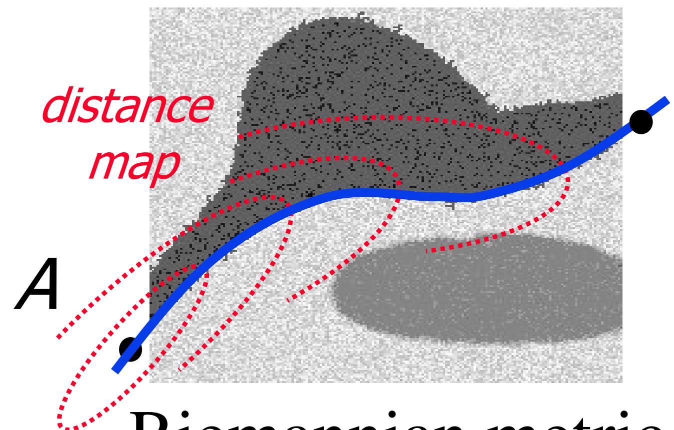


Incorporating image contrast: geodesic active contours [Caselles,Kimmel,Sapiro’97]

- Define *Riemannian metric* from image gradient



Euclidian metric
(constant)

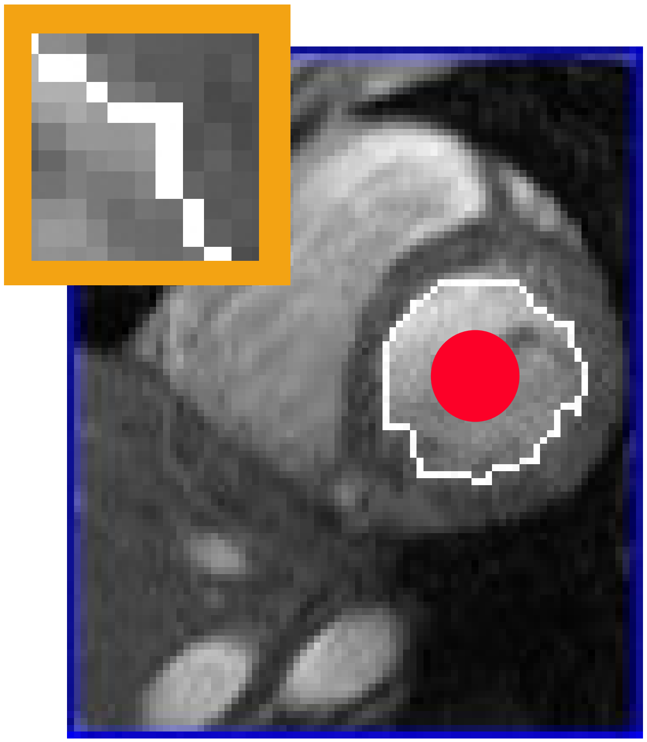


Riemannian metric
(space varying, tensor $D(\cdot)$)

- Compute *geodesics*
  - shortest curve between two points

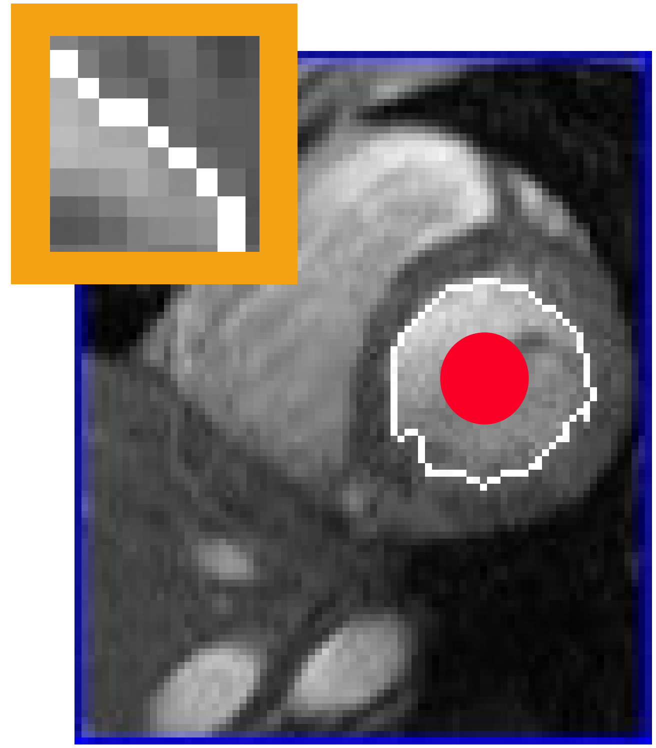
$$E(C) = \int_C g(\cdot)\ ds$$

Metrication errors on graphs



Minimum cost cut
(standard 4-neighborhoods)

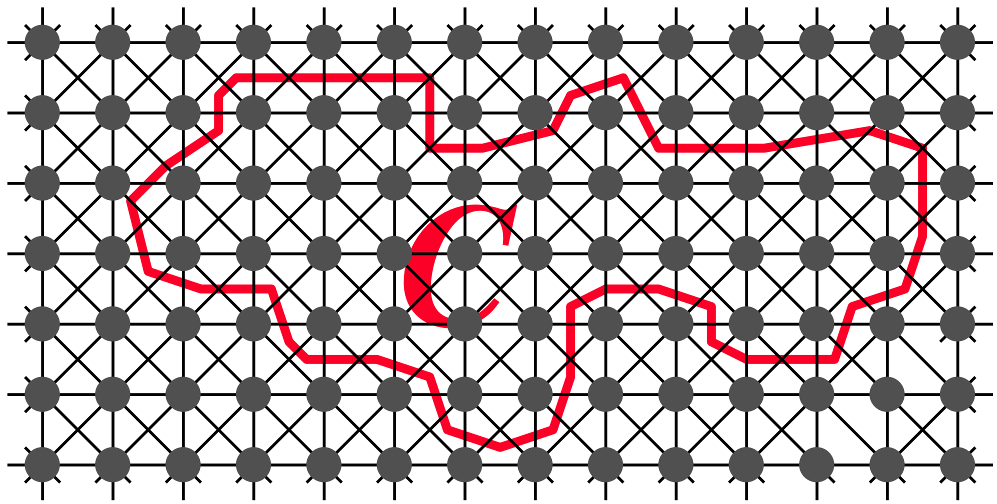
discrete metric ???



Minimum length **geodesic** contour
(image-based Riemannian metric)

Continuous metric space
(no geometric artifacts!)

Cut Metrics : cuts impose *metric* properties on graphs

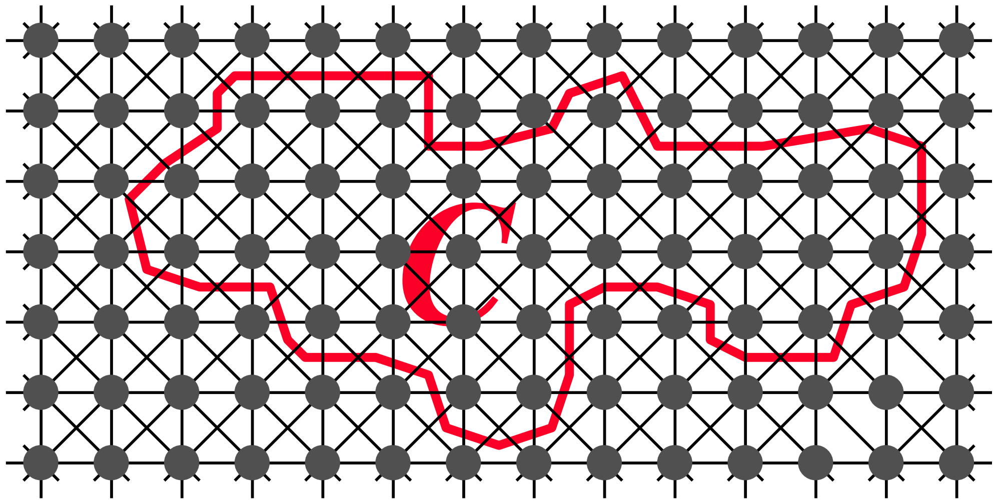


$$\| C \| = \sum_{e \in C} w_e$$

contour's "length"

- "*Cut metrics*" and "*Riemannian metrics*" allow to compute contour "length" in 2D (or "area" in 3D)

Geo-cuts [Boykov,Kolmogorov'03]: Combining graph cuts and geodesic active contours



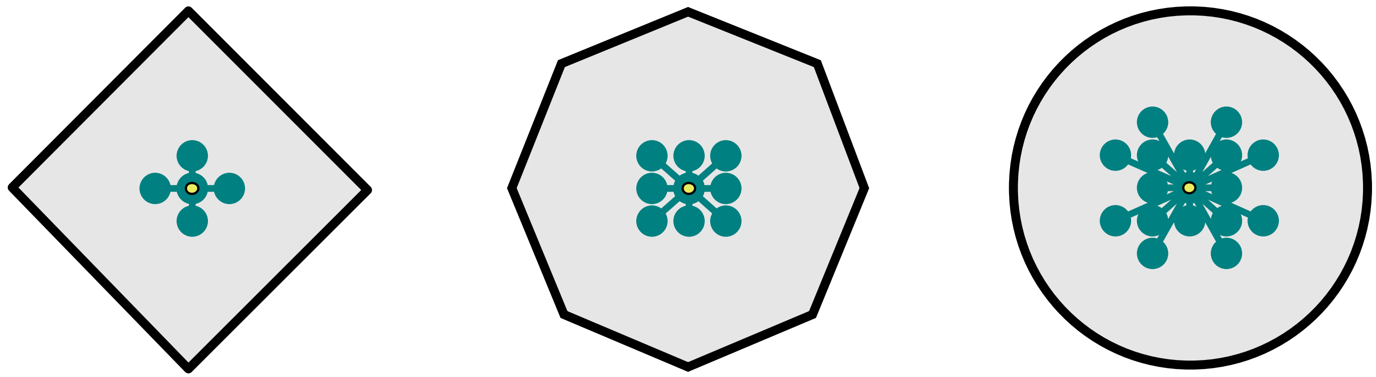
$$\|C\| = \sum_{e \in C} w_e$$

Given geometric functional, e.g. $E(C) = \int_C g(C(s), \vec{N})\ ds$

construct graph such that $E(C) \approx \|C\| \equiv \sum_{e \in C} w_e$

"Distance maps" for cut metric

Euclidean metric



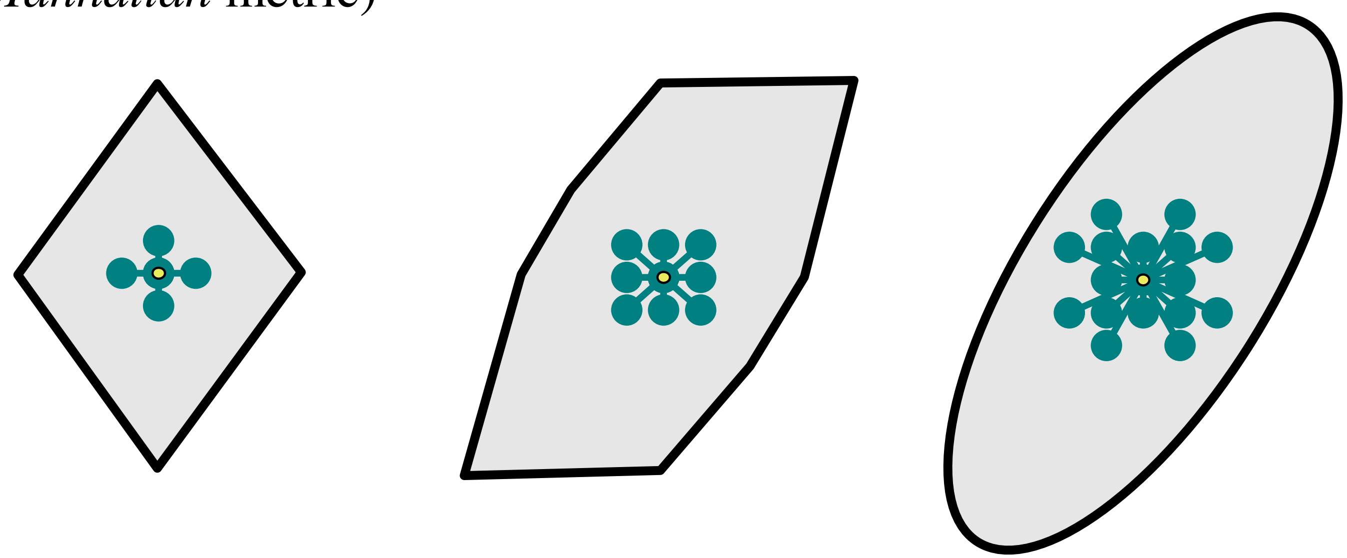
"standard" 4-neighborhoods (*Manhattan* metric)

8-neighborhoods

256-neighborhoods

Riemannian metric

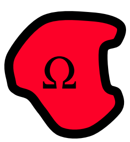
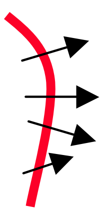
$D(p)$=const



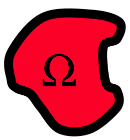
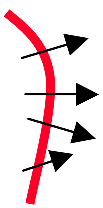
What metrics can be approximated?

- Question: What continuous functionals can be approximated with geo-cuts?

- [Kolmogorov,Boykov’05]:

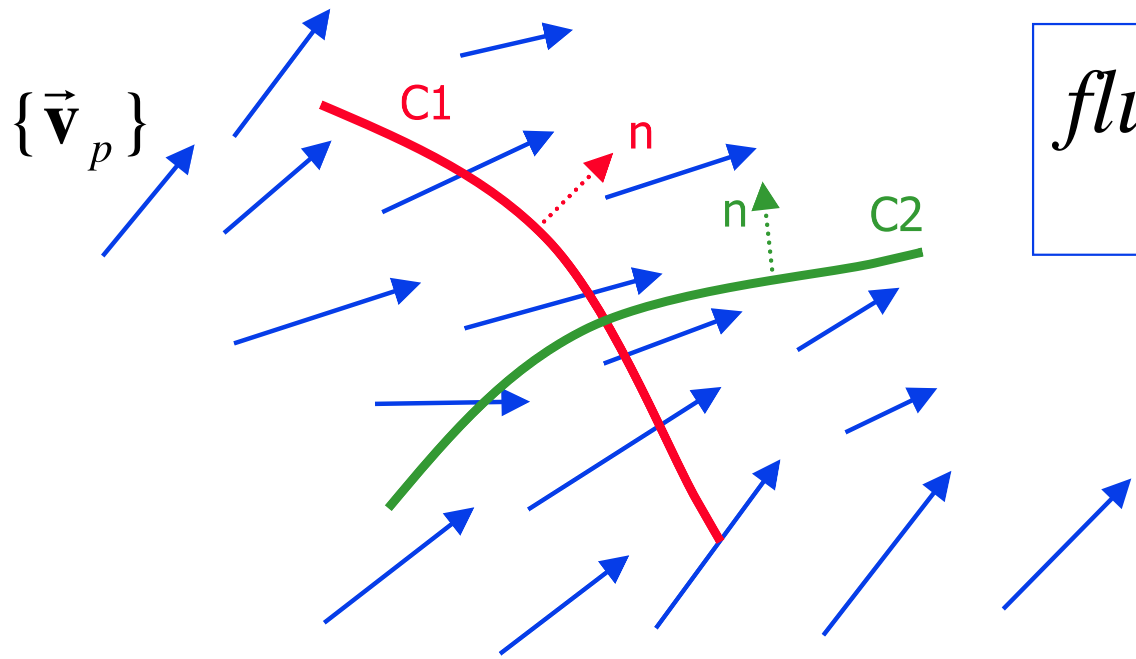
	• Geometric length (e.g. Riemannian) – Distance map for g() is convex & symmetric	$E(C) = \int_C g(\cdot)\ ds$
 	• Regional bias	$+ \iint_\Omega f\ da$
	• Flux of a given vector field	$+ \int_C (\vec{\mathbf{v}} \cdot \vec{N})\ ds$

Geometric measures used in *level set* segmentation [Acknowledgement: Ron Kimmel’s presentation]

		functional	evolution equation
	weighted arc-length	$E(C)=\int_C g(\cdot)\ ds$	$C_t=\left(g-\nabla g\cdot\vec{N}\right)\vec{N}$
	weighted area	$E(C)=\iint_\Omega f\ da$	$C_t=f\ \vec{N}$
	alignment (flux)	$E(C)=\int_C(\vec{\mathbf{v}}\cdot\vec{N})\ ds$	$C_t=-(\mathrm{div}\,\vec{\mathbf{v}})\ \vec{N}$
	robust alignment	$E(C)=\int_C -\lvert\vec{\mathbf{v}}\cdot\vec{N}\rvert\ ds$	$C_t=\mathrm{sign}(\vec{\mathbf{v}}\cdot\vec{N})(\mathrm{div}\,\vec{\mathbf{v}})\ \vec{N}$

Flux

- *vector field*: some vector $\vec{\mathbf{v}}_p$ defined at each point p
  - "stream of water" with a given speed at each location
- *flux*: "amount of water" passing through a given contour



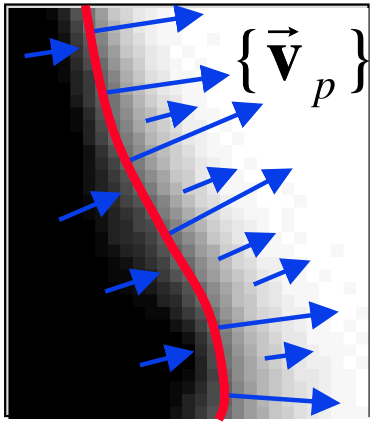
$$flux(C) = \int_C (\vec{\mathbf{v}} \cdot \vec{N})\, ds$$

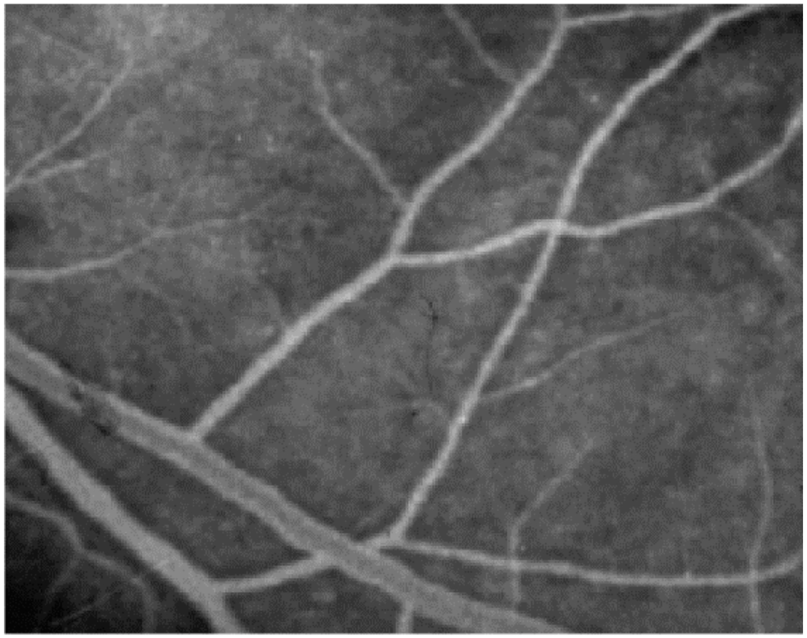
flux(C1) > flux(C2)

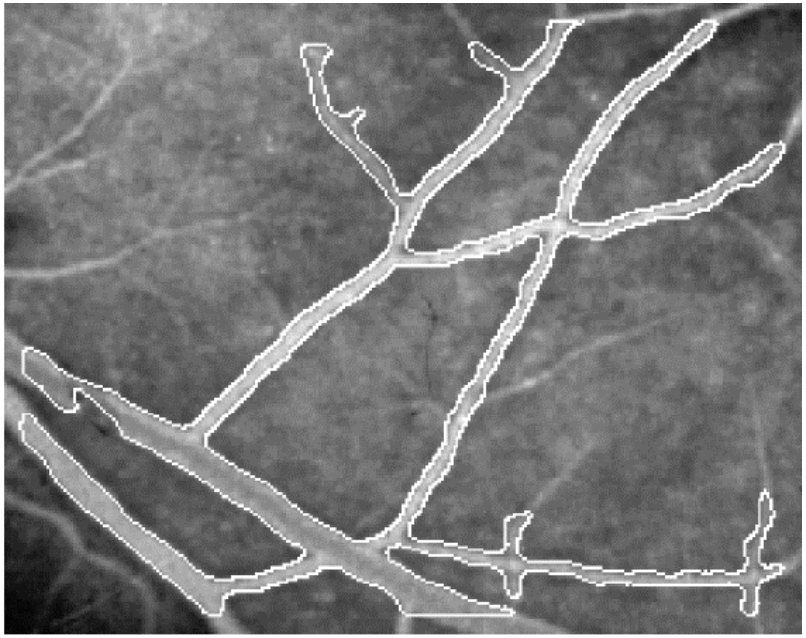
- Changes sign with orientation

Segmentation of thin objects [Vasilevskiy,Siddiqi'02]

- Vector field: $\vec{\mathbf{v}} = \nabla I$

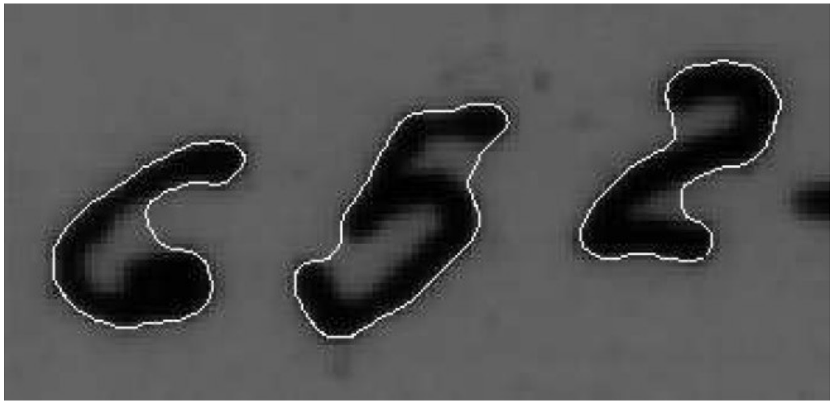




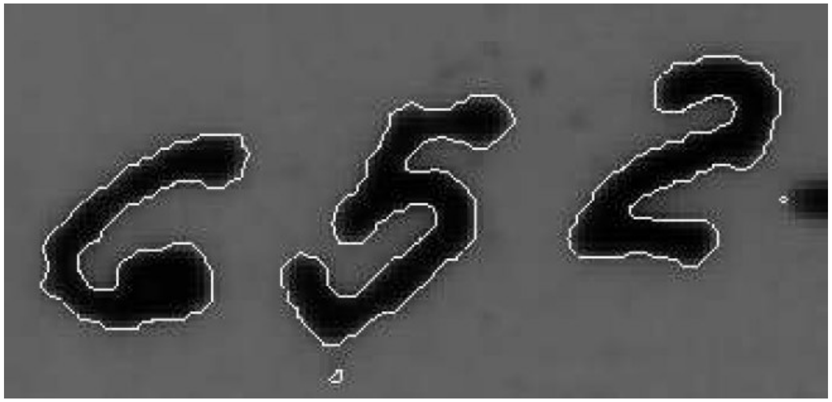


Riemannian length + Flux [Kimmel,Bruckstein'03]

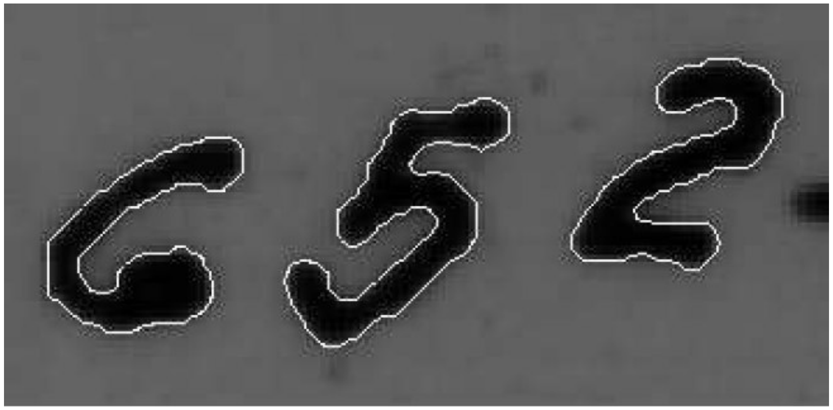
Riemannian length



Flux of ∇I



Riemannian length + Flux



Robust alignment

$$E(C) = \int_C (\nabla I \cdot \vec{N}) \; ds$$

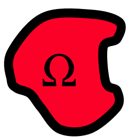
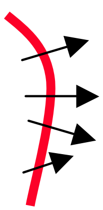
assumes bright object, dark background

$$E(C) = \int_C - | \nabla I \cdot \vec{N} | \; ds$$

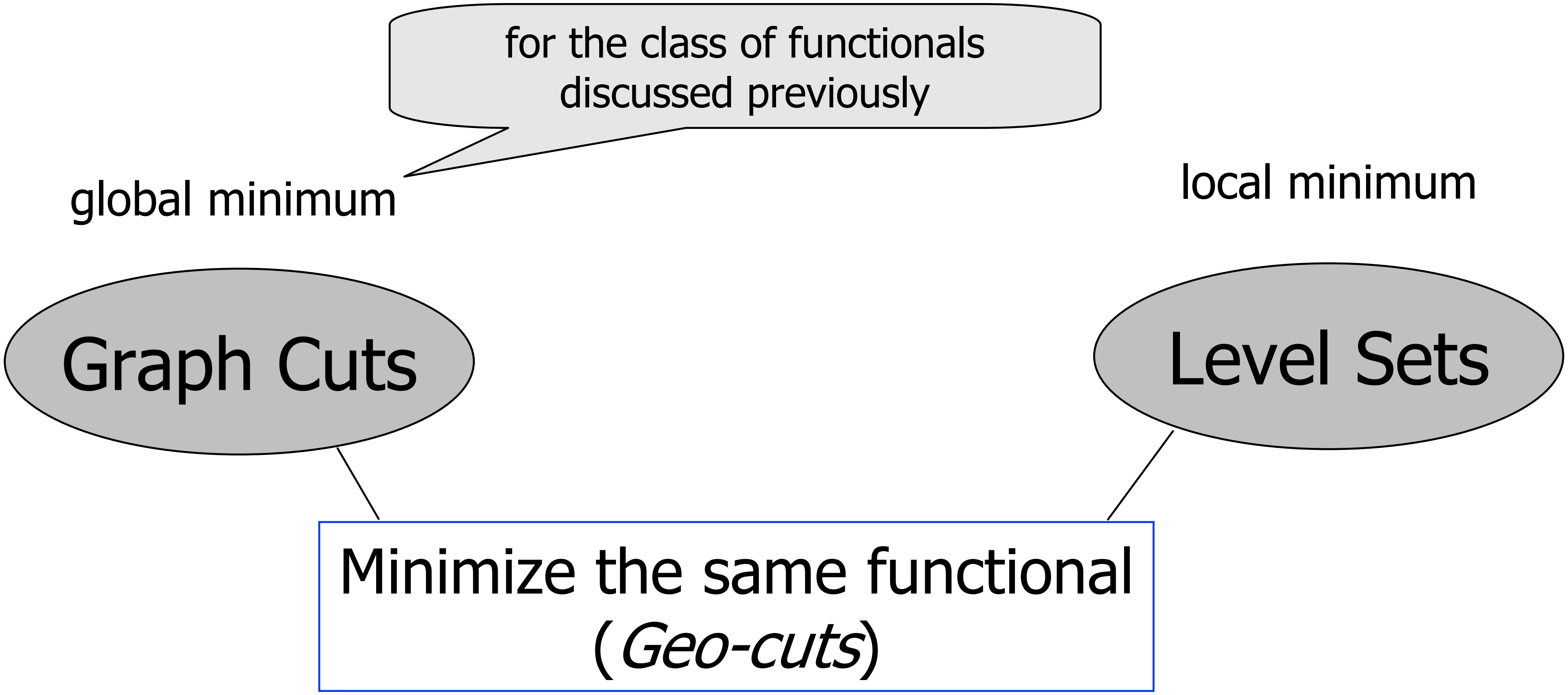
no such assumption

"Robust alignment"
[Kimmel,Bruckstein'03]

Geometric measures used in *level set* segmentation [Acknowledgement: Ron Kimmel's presentation]

	functional		
	weighted arc-length	$E(C) = \int_C g(\cdot)\ ds$	same as in geo-cuts
	weighted area	$E(C) = \iint_\Omega f\ da$	same as in geo-cuts
	alignment (flux)	$E(C) = \int_C (\vec{\mathbf{v}} \cdot \vec{N})\ ds$	same as in geo-cuts
	robust alignment	$E(C) = \int_C - \mid \vec{\mathbf{v}} \cdot \vec{N} \mid\ ds$	non-submodular

Graph cuts vs. level sets for geodesic active contours



- Connection only approximate: $E(C) \approx ||C||$
- Even stronger connection: *continuous maxflow*

Continuous maxflow

- [Iri’79],[Strang’83],[Appleton,Talbot’03]
  - Analogue of discrete maxflow
  - Solves continuous problem in subset of R^n
  - Flow = vector field

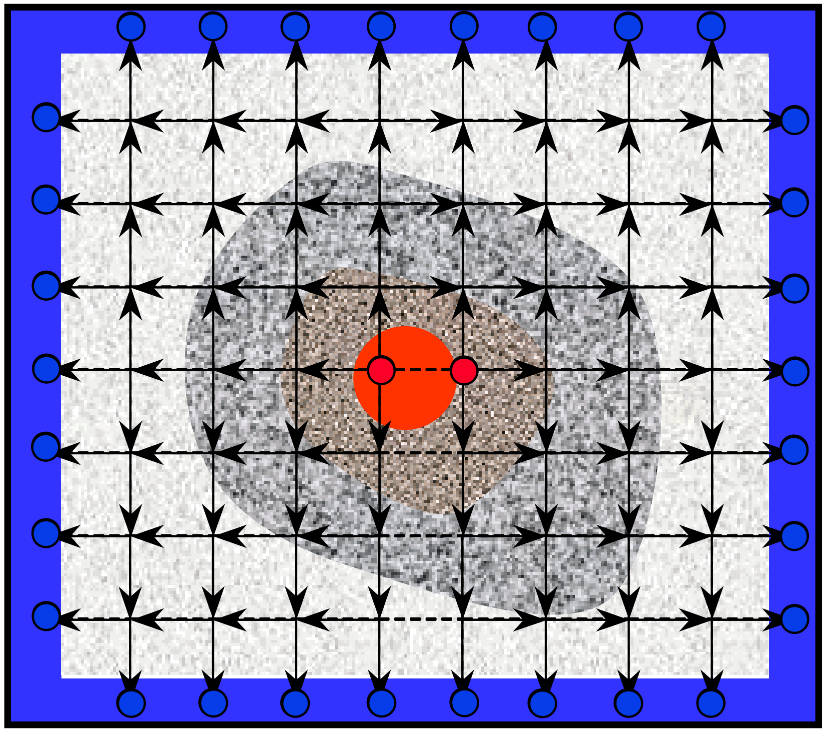
Discrete maxflow

- Flow defined on graph edges
  - $f_{pq} = -f_{qp}$

Capacity constraint: $f_{pq} \leq w_{pq}$

Flow conservation: $\sum_q f_{pq} = 0$

(for $p \neq s,t$)



4-neighbourhood system

<u>Maximize flow out of the source(s)</u>: $\sum_q f_{sq} \rightarrow \max$

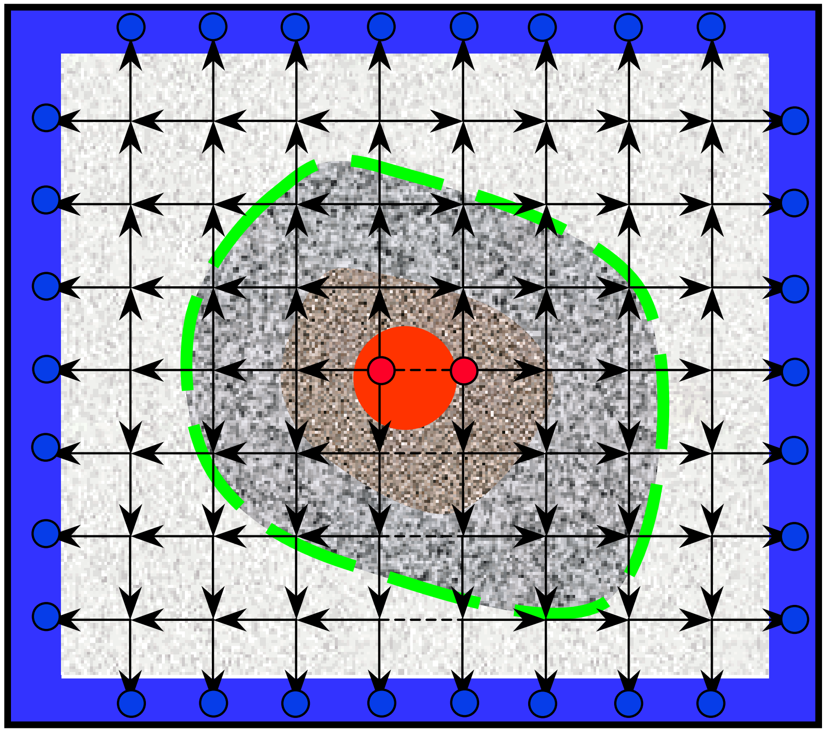
Discrete maxflow

- Flow defined on graph edges
  - $f_{pq} = -f_{qp}$

Capacity constraint: $f_{pq} \leq w_{pq}$

Flow conservation: $\sum_q f_{pq} = 0$

(for $p \neq s,t$)

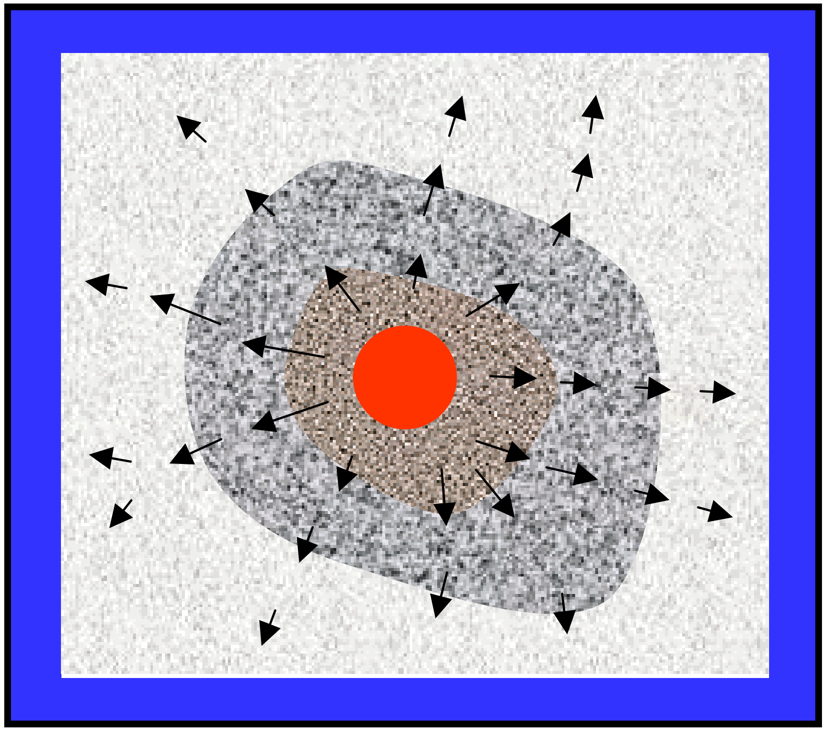


[Ford&Fulkerson theorem]:

<u>Maximum flow saturates minimum cut</u>

Continuous maxflow

- Flow = vector field



Capacity constraint: $|\vec{f}_p| \leq g$

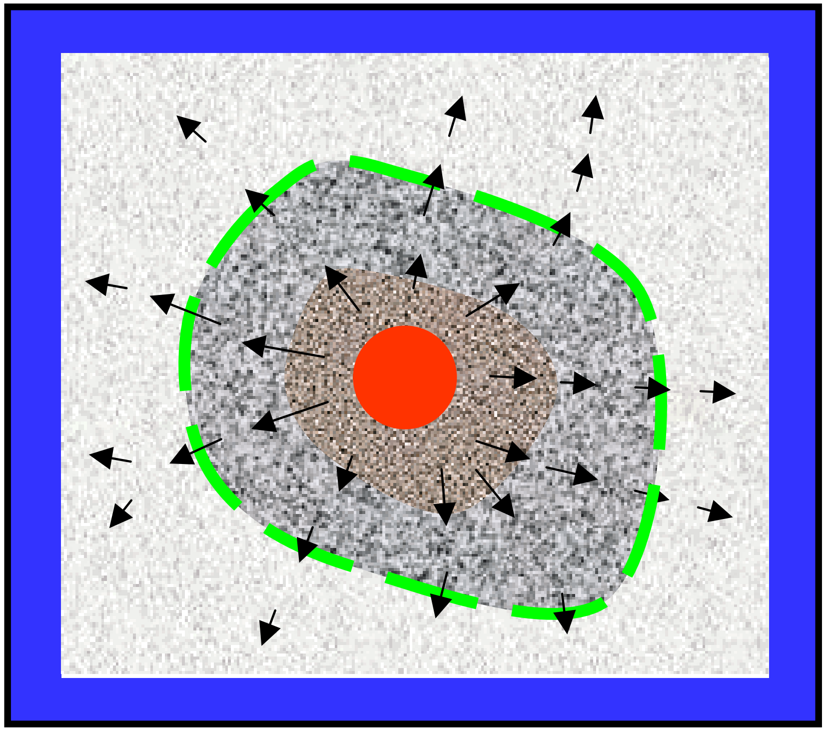
Flow conservation: $\operatorname{div} \vec{f}_p = 0$

(for $p \notin s,t$)

Maximize flow out of the source: $\int_s (\operatorname{div} \vec{f}_p)\, da \rightarrow \max$

Continuous maxflow

- Flow = vector field



Capacity constraint: $|\vec{f}_p| \le g$

Flow conservation: $\operatorname{div} \vec{f}_p = 0$

(for $p \notin s,t$)

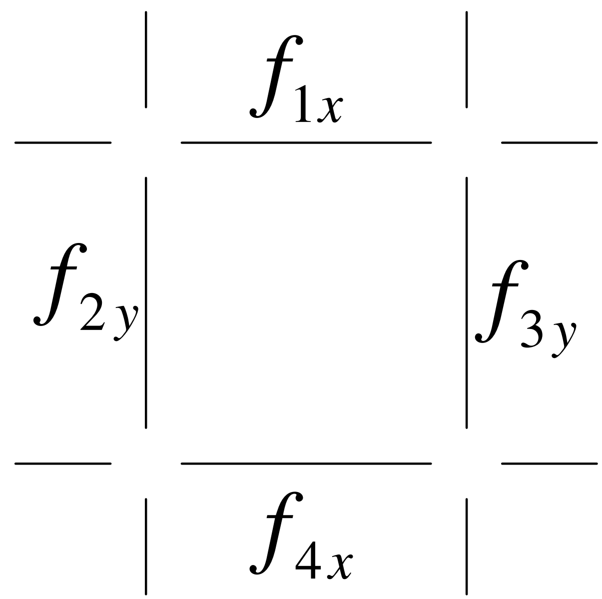
Maximum flow saturates minimum cut

$$\vec{f}_p = g_p \vec{N} \quad (\text{ for } p \in C^*)$$

Solving continuous maxflow

- [Appleton,Talbot'03,06]: numerical algorithm
  - Vector field stored on *edges*
    - Horizontal edges => *x*-component
    - Vertical edges => *y*-component
  - Flow conservation similar to the discrete case
  - <u>But</u> - capacity constraint:

$$f_x^{\ 2} + f_y^{\ 2} \leq g^2$$



  - Report 0.1 pixels accuracy (on average)