Introduction to Image Segmentation:

Part 1: binary image labeling
discrete (and other) methods

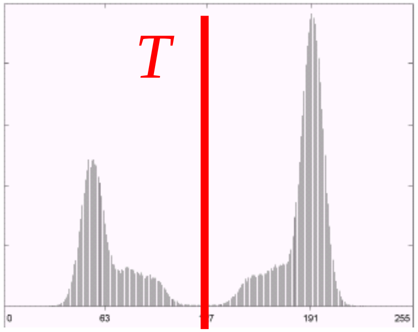
Yuri Boykov

University of Western Ontario

Introduction to Image Segmentation

- motivation for optimization-based approach
- active contours, level-sets, graph cut, etc.
- implicit/explicit representation of boundaries
- objective functions (energies)
  - physics, geometry, statistics, information theory
  - set functions and submodularity (graph cuts)
- *part II*: from binary to multi-label problems

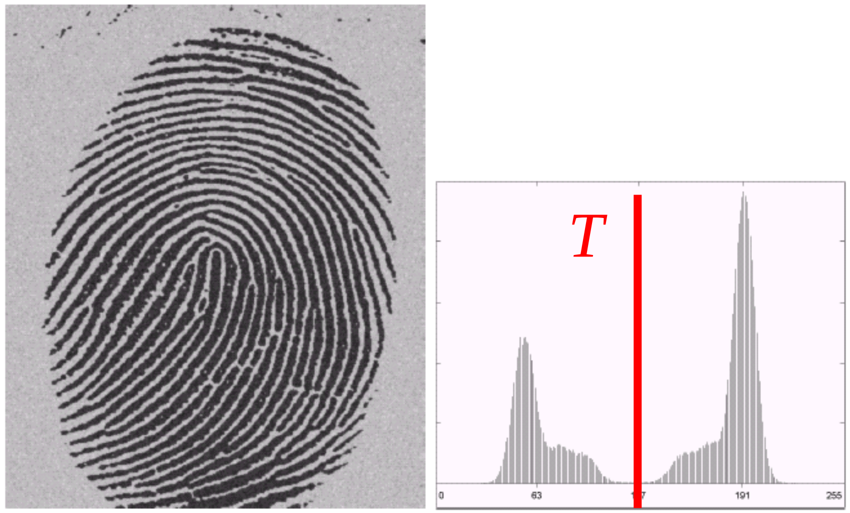
Thresholding



a b
c

FIGURE 10.29 (a) Original image. (b) Image histogram. (c) Result of segmentation with the threshold estimated by iteration. (Original courtesy of the National Institute of Standards and Technology.)

Thresholding



a b
c

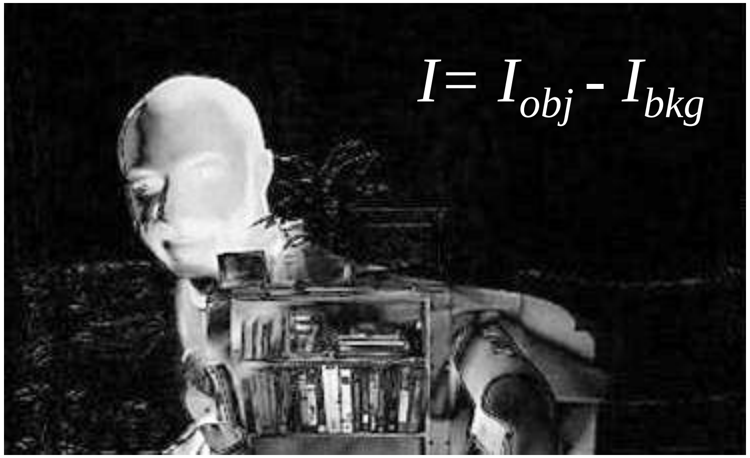
FIGURE 10.29 (a) Original image. (b) Image histogram. (c) Result of segmentation with the threshold estimated by iteration. (Original courtesy of the National Institute of Standards and Technology.)

$$S=\{\, p : I_p < T \,\}$$

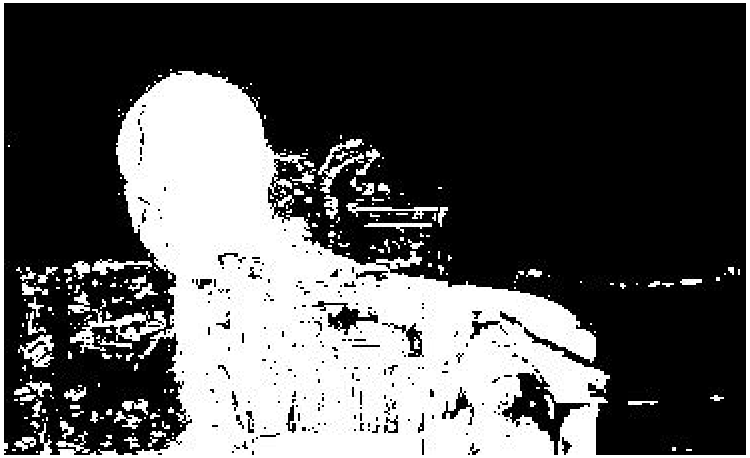
Thresholding

-

=



Threshold intensities above T

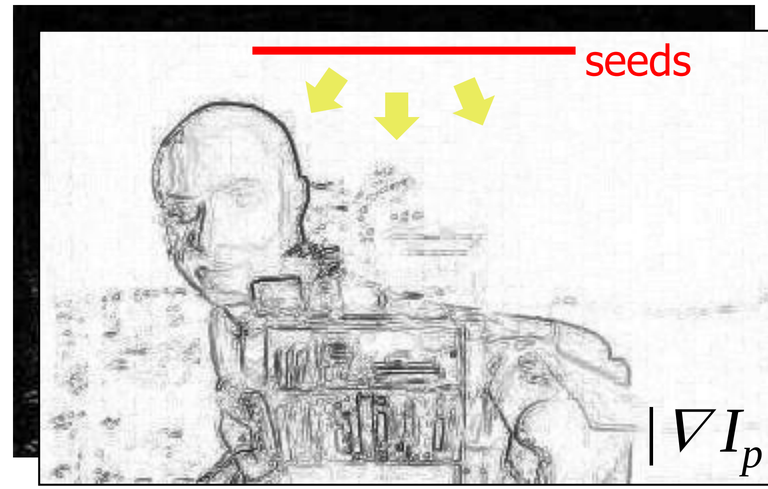


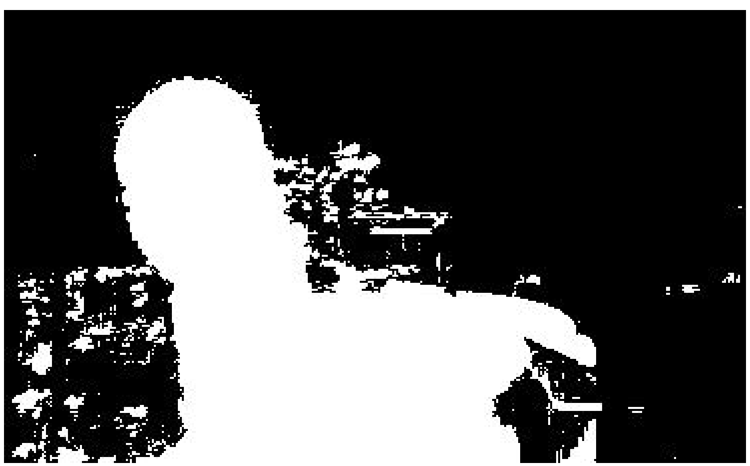
$S = \{ p : I_p > T \}$ ← **segment's region property**

Region growing

-

=





Breadth First Search (seeds) :

$$|\nabla I_p| < T$$

$$p \in \partial S \quad \Rightarrow \quad |\nabla I_p| > T \quad \longleftarrow$$ **segment's boundary property**

Region growing

“leakage”

Good segmentation S ?

- Objective function must be specified

Quality function

Cost function

Loss function $$E(S) \; : \; 2^P \; \rightarrow \; \Re$$

"Energy"

Regularization functional

Segmentation becomes an **optimization problem**: $S = arg\ min\ E(S)$

Good segmentation S ?

- Objective function must be specified

Quality function

Cost function

“Energy”

Regularization functional

$$E(S) = E_1(S) + \ldots + E_n(S)$$

combining different constraints
e.g. on **region** and **boundary**

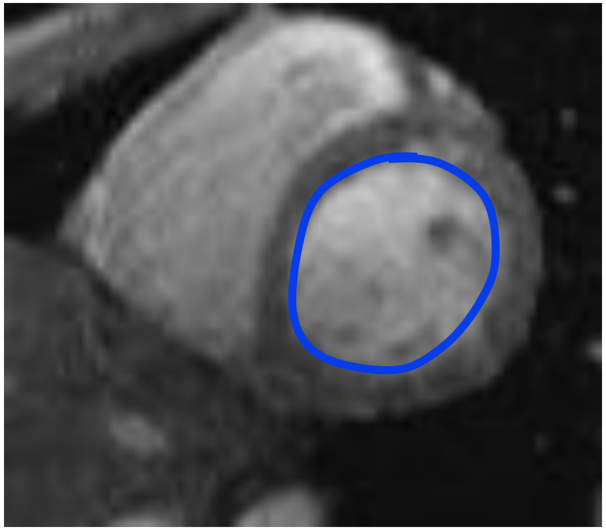
Common segmentation techniques

boundary-based	*region-based*	*both region & boundary*
region-growing	thresholding	geodesic
intelligent scissors (live-wire)		active contours (e.g. level-sets)
active contours (snakes)		MRF (e.g. graph-cuts)
watersheds		random walker

optimization-based

Active contours - snakes

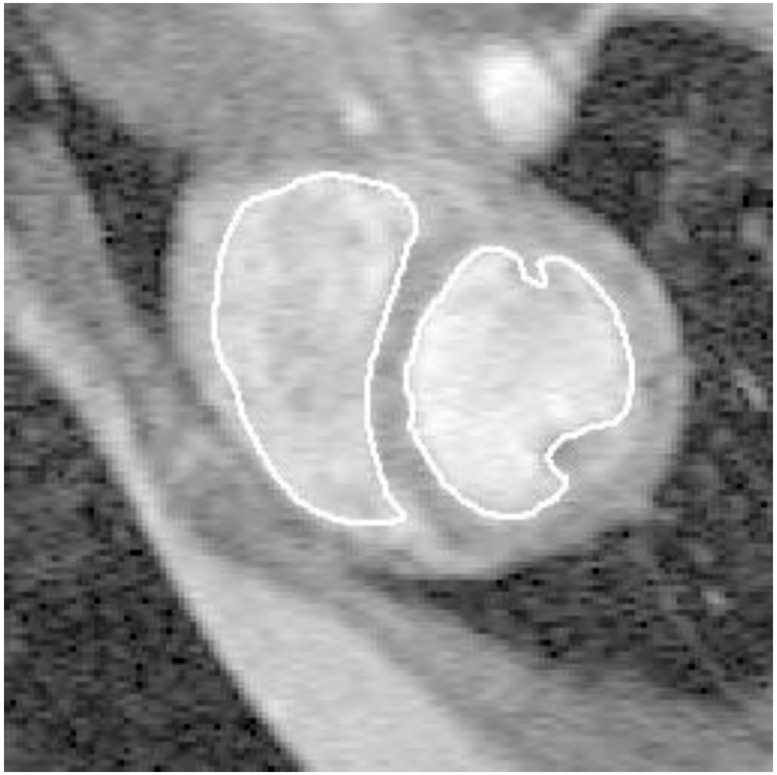
[Kass, Witkin, Terzopoulos 1987]



Given: initial contour (model) near desirable object

Goal: evolve the contour to fit exact object boundary

Tracking via active contours



Tracking Heart Ventricles

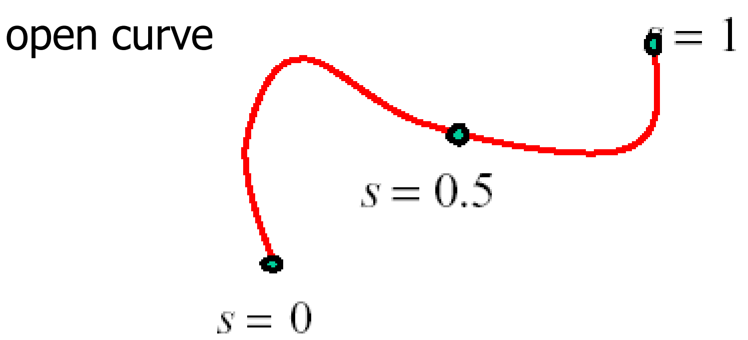
Active contours - snakes

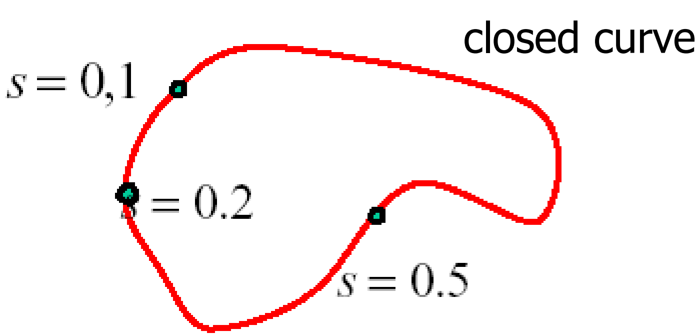
Parametric Curve Representation (continuous case)

A curve can be represented by 2 functions

parameter

$$\mathbf{v}(s) = (x(s), y(s)) \qquad 0 \leq s \leq 1$$





$$C = \{ \mathrm{v}(s) \mid s \in [0,1] \} \quad \in \ \Re^{\infty}$$

Snake Energy

$$E(C) = E_{in}(C) + E_{ex}(C)$$

internal energy encourages smoothness or any particular shape

external energy encourages curve onto image structures (e.g. image edges)

Active contours - snakes
(continuous case)

- *internal* energy (physics of elastic band)

$$E_{in}(C) = \alpha \cdot \boxed{\int_0^1 \left|\frac{d\mathbf{v}}{ds}\right|^2 ds} + \beta \cdot \boxed{\int_0^1 \left|\frac{d^2\mathbf{v}}{d^2 s}\right|^2 ds}$$

elasticity / stretching stiffness / bending

- *external* energy (from image)

$$E_{ex}(C) = \boxed{-\int_0^1 |\nabla I(\mathbf{v}(s))|^2 ds}$$

proximity to image edges

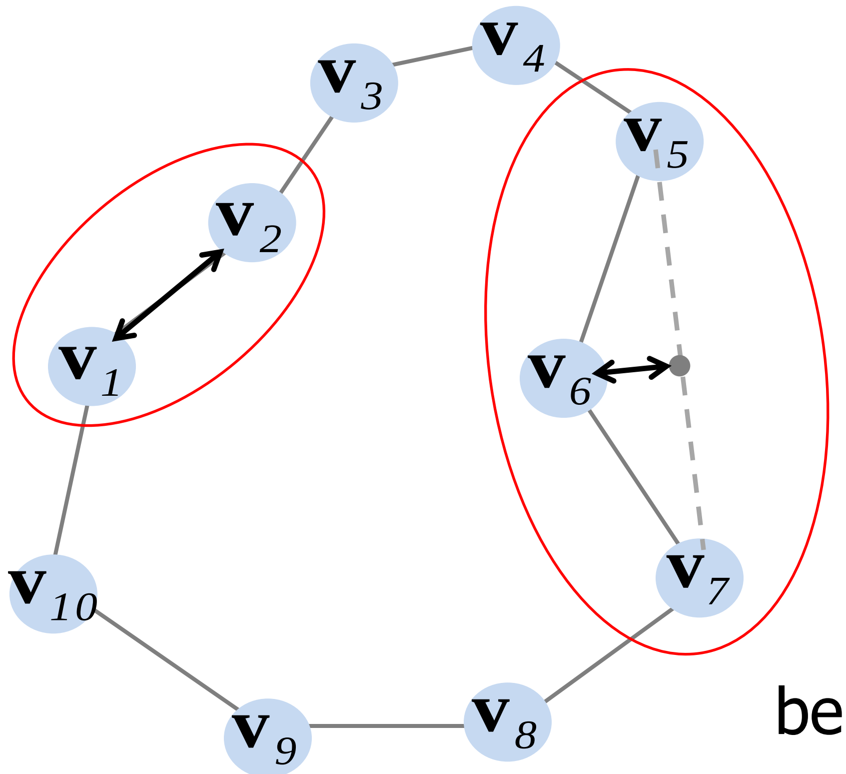
Active contours – snakes
(discrete case)

$$\mathbf{C} = (\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{n-1}) \in \Re^{2n}$$

$$\mathbf{v}_i = (x_i, y_i)$$

elastic energy
(elasticity)

$$\frac{dv}{ds} \approx \frac{v_{i+1} - v_{i-1}}{2}$$



bending energy
(stiffness)

$$\frac{d^2 v}{ds^2} \approx (v_{i+1} - v_i) - (v_i - v_{i-1}) = v_{i+1} - 2v_i + v_{i-1}$$

Basic Elastic Snake

$$E = \alpha \cdot \int_0^1 \left| \frac{dv}{ds} \right|^2 ds \;-\; \int_0^1 |\nabla I(v(s))|^2 ds$$

continuous case

$\mathbf{C} = \{ \mathbf{v}(s) \,|\, s \in [0,1] \}$

$$E = \alpha \cdot \sum_{i=0}^{n-1} | v_{i+1} - v_i |^2 \;-\; \sum_{i=0}^{n-1} |\nabla I(v_i)|^2$$

discrete case

$\mathbf{C} = \{ \mathbf{v}_i \,|\, 0 \le i < n \}$

elastic smoothness term
(interior energy)

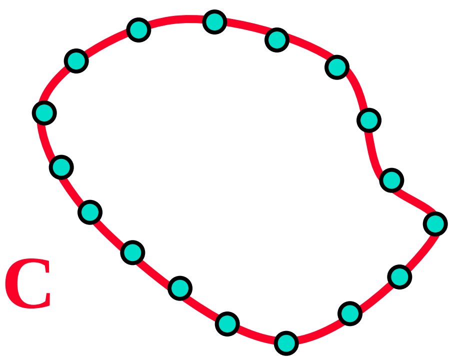
image data term
(exterior energy)

Snakes - gradient descent

simple elastic snake energy

$$E(\overbrace{x_0,\ldots,x_{n-1},y_0,\ldots,y_{n-1}}^{\mathbf{C}}) = -\sum_{i=0}^{n-1} |I_x(x_i,y_i)|^2 + |I_y(x_i,y_i)|^2 + \alpha \cdot \sum_{i=0}^{n-1} (x_{i+1}-x_i)^2 + (y_{i+1}-y_i)^2$$

here, *energy* is a function of 2n variables



update equation for the whole snake

$$\mathbf{C}' = \mathbf{C} - \nabla E \cdot \Delta t$$

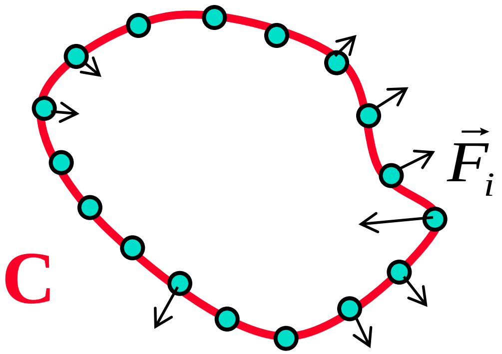
$$\begin{pmatrix} x'_0 \\ y'_0 \\ \ldots \\ x'_{n-1} \\ y'_{n-1} \end{pmatrix} = \begin{pmatrix} x_0 \\ y_0 \\ \ldots \\ x_{n-1} \\ y_{n-1} \end{pmatrix} - \begin{pmatrix} \frac{\partial E}{\partial x_0} \\ \frac{\partial E}{\partial y_0} \\ \ldots \\ \frac{\partial E}{\partial x_{n-1}} \\ \frac{\partial E}{\partial y_{n-1}} \end{pmatrix} \cdot \Delta t$$

Snakes - gradient descent

simple elastic snake energy

$$E(\overbrace{x_0,\ldots,x_{n-1},y_0,\ldots,y_{n-1}}^{C}) = -\sum_{i=0}^{n-1} |I_x(x_i,y_i)|^2 + |I_y(x_i,y_i)|^2 + \alpha\cdot\sum_{i=0}^{n-1}(x_{i+1}-x_i)^2+(y_{i+1}-y_i)^2$$

here, *energy* is a function of 2n variables



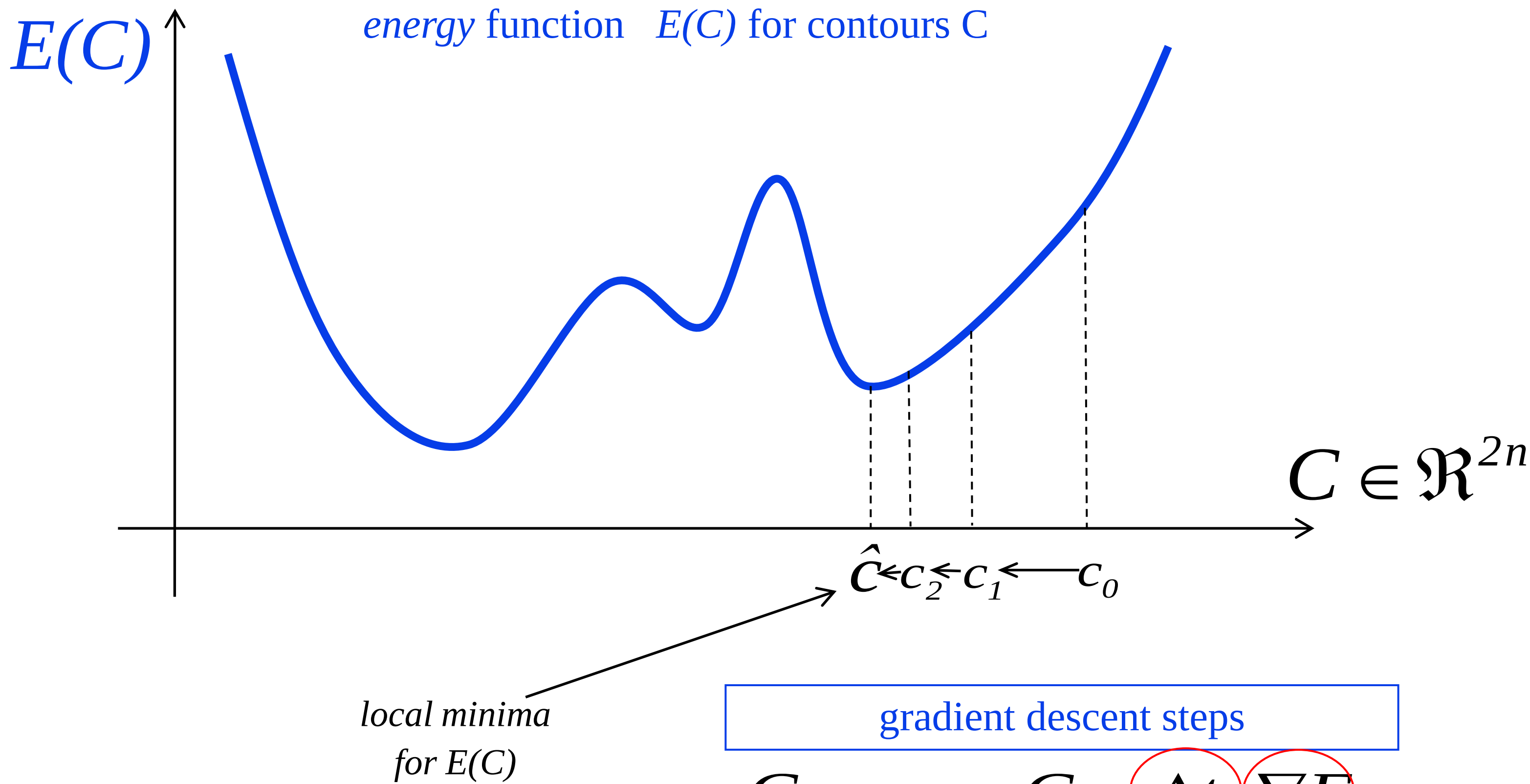
update equation for each node

$$\mathbf{v}'_i = \mathbf{v}_i + \vec{F}_i \cdot \Delta t$$

$$\vec{F}_i = -\begin{bmatrix} \frac{\partial E}{\partial x_i} \\ \frac{\partial E}{\partial y_i} \end{bmatrix}$$

$$\begin{pmatrix} x'_0 \\ y'_0 \\ \cdots \\ x'_{n-1} \\ y'_{n-1} \end{pmatrix} = \begin{pmatrix} x_0 \\ y_0 \\ \cdots \\ x_{n-1} \\ y_{n-1} \end{pmatrix} - \begin{pmatrix} \frac{\partial E}{\partial x_0} \\ \frac{\partial E}{\partial y_0} \\ \cdots \\ \frac{\partial E}{\partial x_{n-1}} \\ \frac{\partial E}{\partial y_{n-1}} \end{pmatrix} \cdot \Delta t$$

Snakes - gradient descent



gradient descent steps

$$C_{i+1} = C_i - \Delta t \cdot \nabla E$$

step size could be tricky

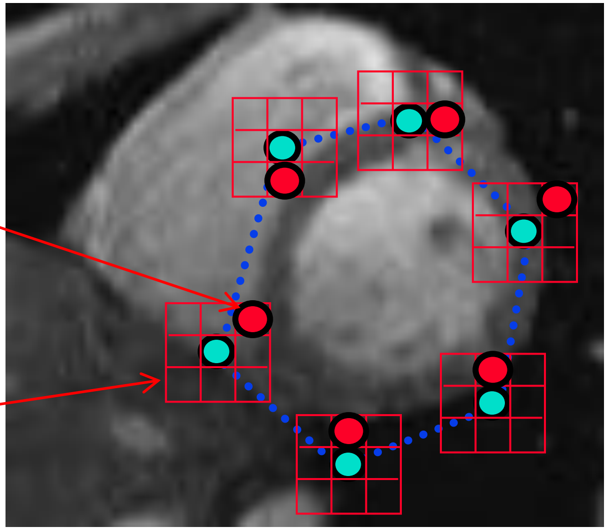
second derivative of image intensities

Snakes – dynamic programming (DP)

[Amini, Weymouth, Jain, 1990]

optimal state

possible states for each control point



$$\mathbf{v}_i = (x_i, y_i)$$

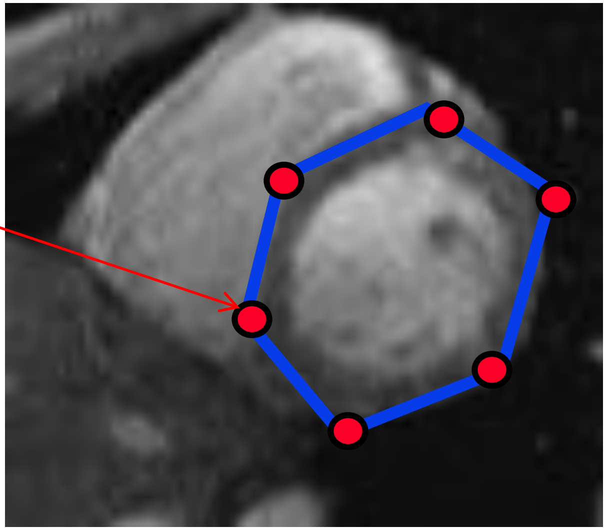
Elastic energy - **pairwise interactions**

$$E(v_1, v_2, ..., v_n) = E_1(v_1, v_2) + E_2(v_2, v_3) + ... + E_{n-1}(v_{n-1}, v_n)$$

Energy E can be minimized via Dynamic Programming

Snakes – dynamic programming (DP)

[Amini, Weymouth, Jain, 1990]



optimal state

Advantages: no 2^{nd} derivatives
explicit step size control

Elastic energy - **pairwise interactions**

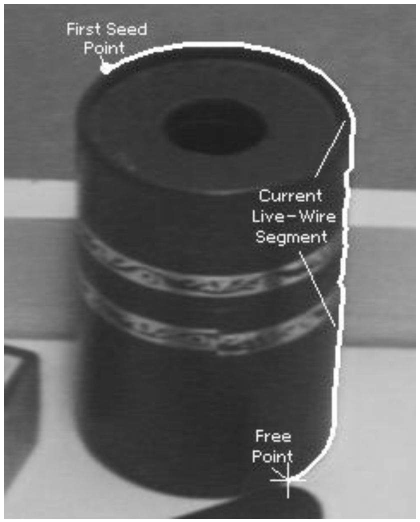
$$E(v_1, v_2, \ldots, v_n) = E_1(v_1, v_2) + E_2(v_2, v_3) + \ldots + E_{n-1}(v_{n-1}, v_n)$$

Energy E can be minimized via Dynamic Programming

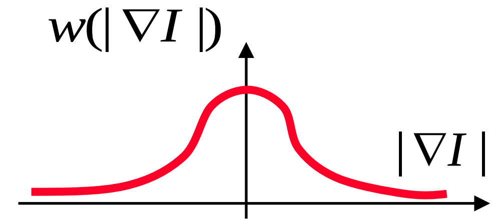
Iterate... until optimal position for each point is the center of the box, (local minimum condition)

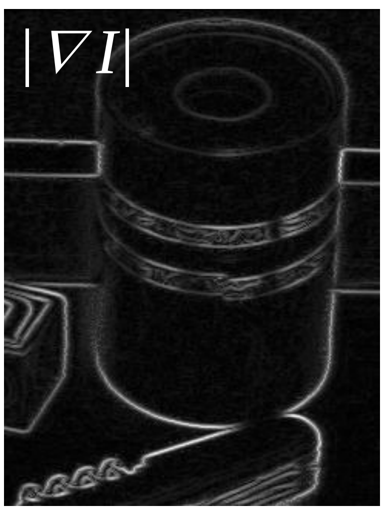
Another example of DP

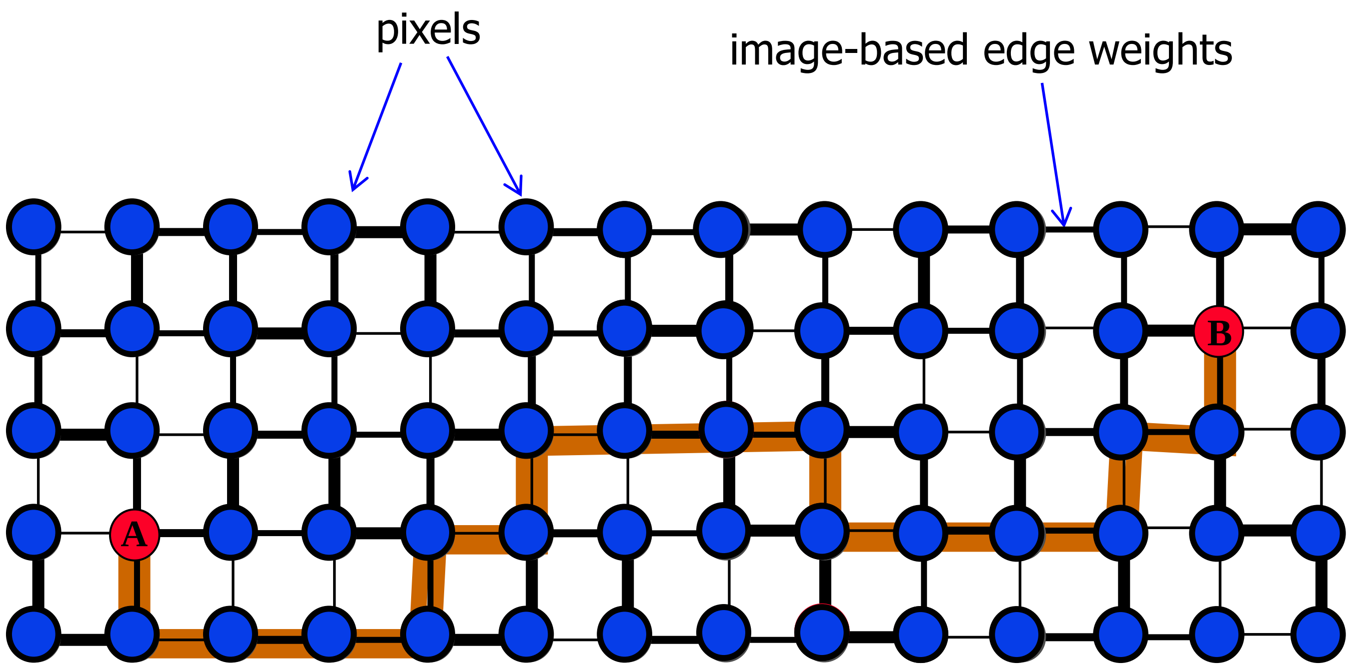
"Live wire" or *"intelligent scissors"*



[Barrett and Mortensen 1996]



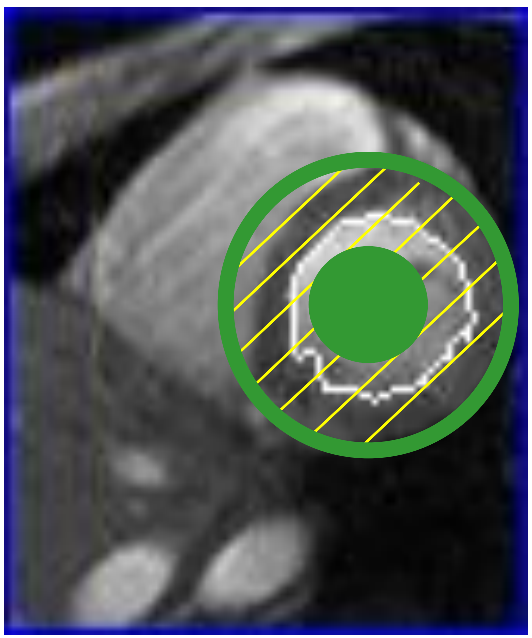




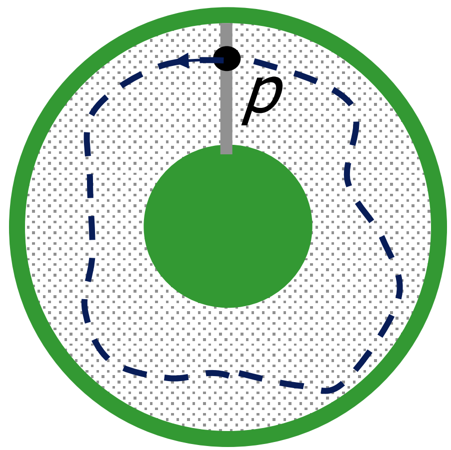
shortest path algorithm (Dijkstra)

shortest path on a 2D graph ⇔ *graph cut*

Example:
find the shortest closed contour in a given domain of a graph

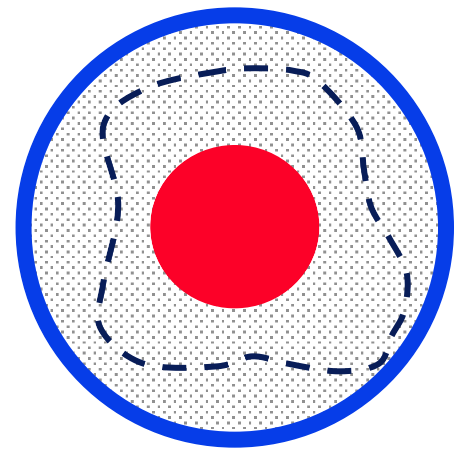


Shortest paths approach



Compute the *shortest path* *p -> p* for a point *p*. Repeat for all points on the gray line. Then choose the optimal contour.

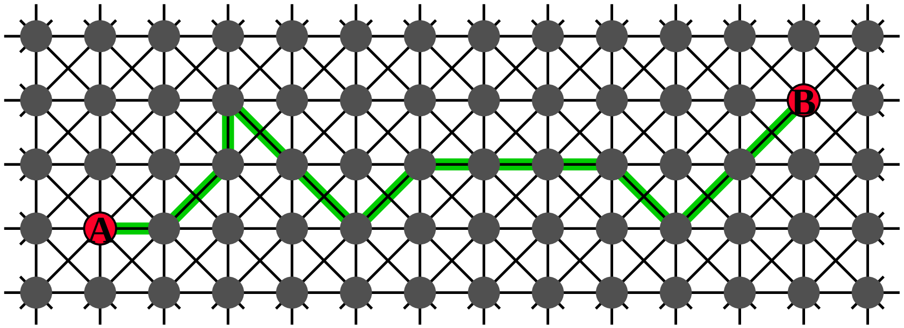
Graph Cuts approach



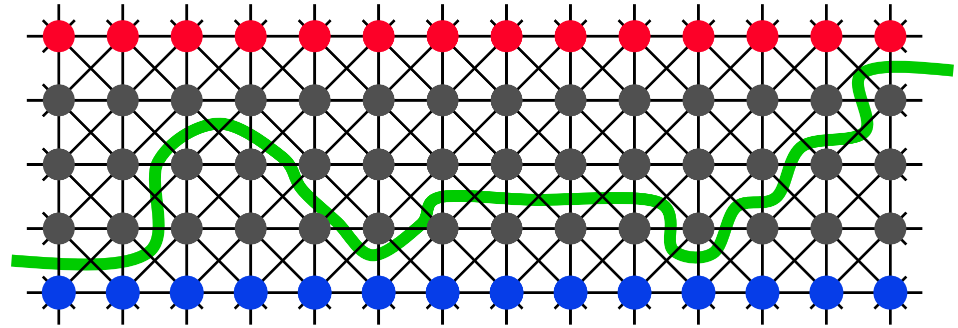
Compute the *minimum cut* that separates red region from blue region

graph cuts vs. *shortest paths*

- On 2D grids *graph cuts* and *shortest paths* give optimal 1D contours.



A Path connects points

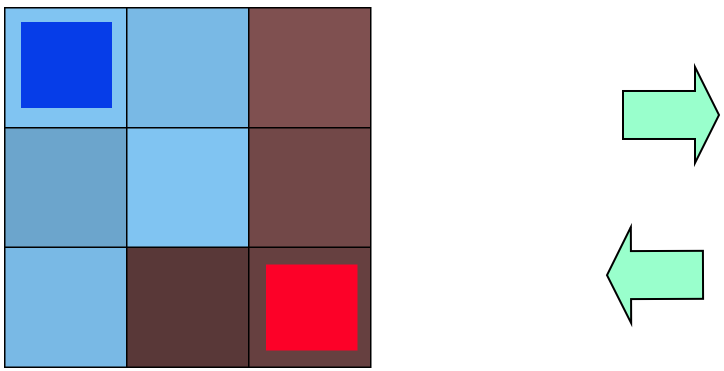


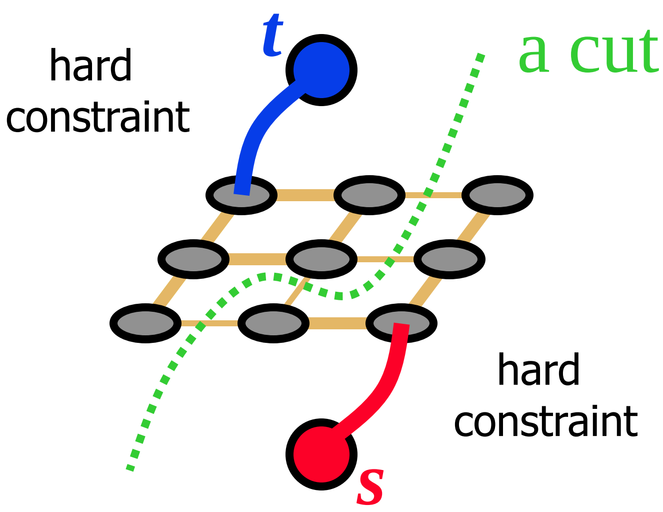
A Cut separates regions

- *Shortest paths* still give optimal 1-D **contours** on N-D grids
- *Min-cuts* give optimal **hyper-surfaces** on N-D grids

Graph cut

[Boykov and Jolly 2001]

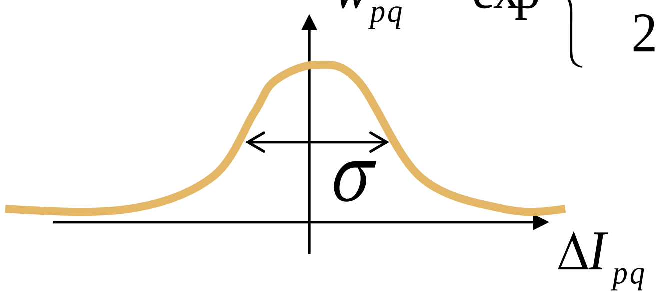




Minimum cost cut can be computed in polynomial time

(max-flow/min-cut algorithms)

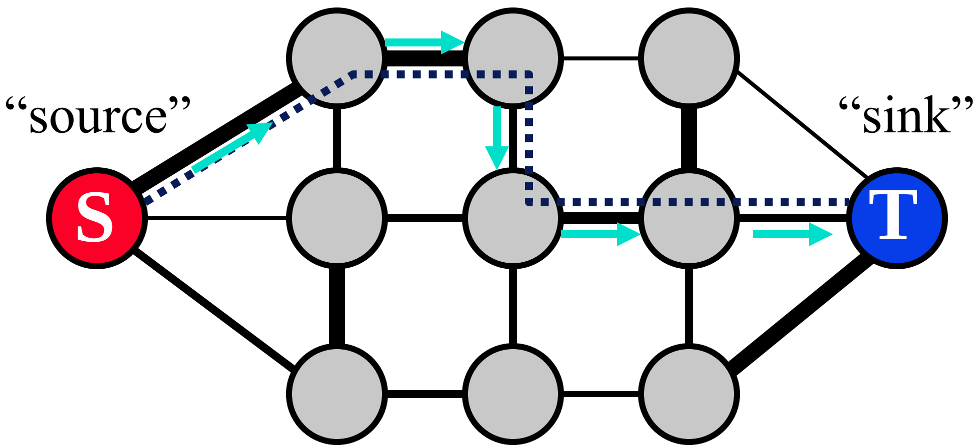
$$w_{pq} = \exp\left\{-\frac{\Delta I_{pq}}{2\sigma^2}\right\}$$



Minimum *s-t* cuts algorithms

- Augmenting paths [Ford & Fulkerson, 1962]
  - heuristically tuned to grids [Boykov&Kolmogorov 2003]
- Push-relabel [Goldberg-Tarjan, 1986]
  - good choice for denser grids, e.g. in 3D
- Preflow [Hochbaum, 2003]
  - also competitive

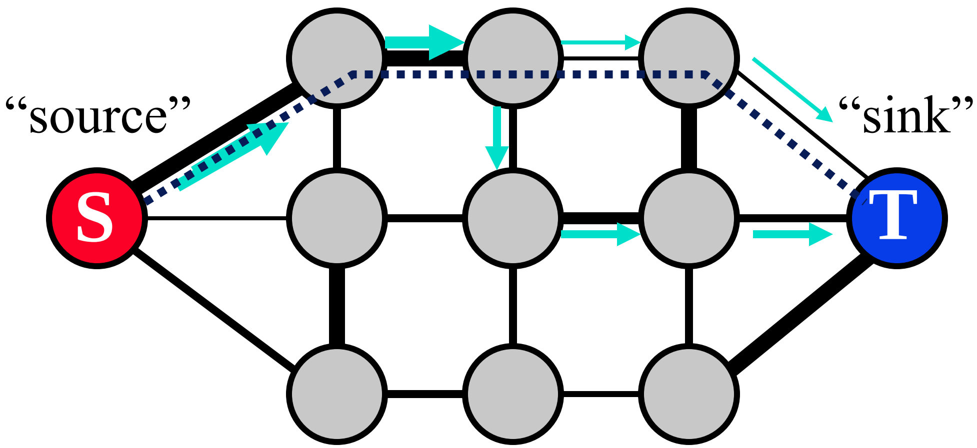
"Augmenting Paths"



A graph with two terminals

- Find a path from S to T along non-saturated edges
- Increase flow along this path until some edge saturates

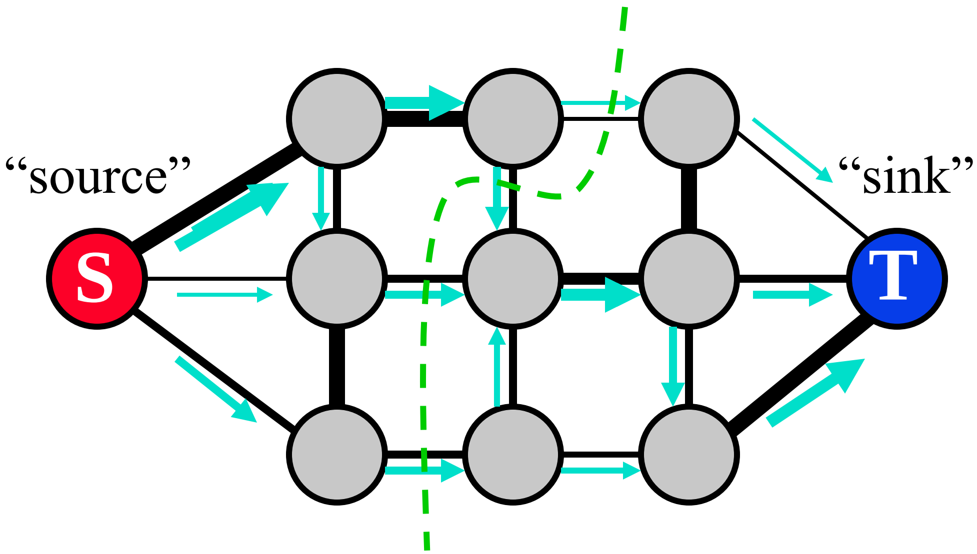
"Augmenting Paths"



A graph with two terminals

- Find a path from S to T along non-saturated edges
- Increase flow along this path until some edge saturates
- Find next path…
- Increase flow…

"Augmenting Paths"



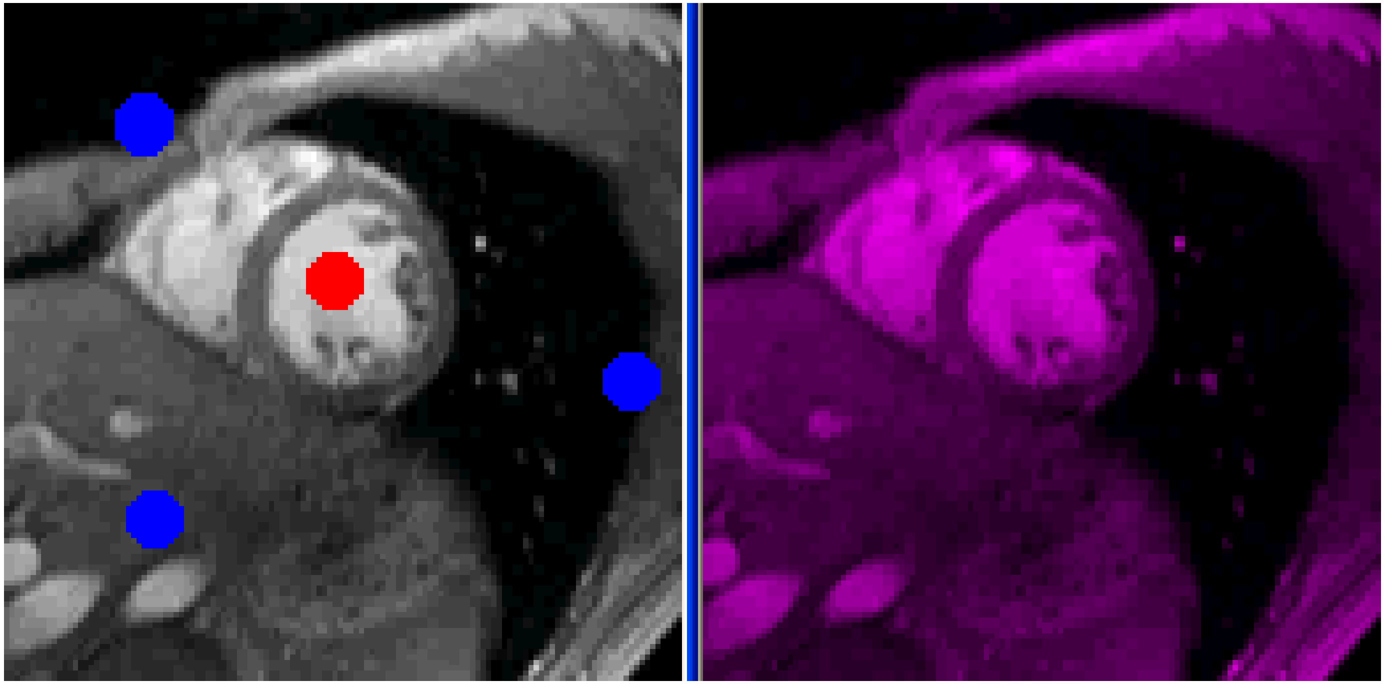
A graph with two terminals

MAX FLOW ⇔ MIN CUT

- Find a path from S to T along non-saturated edges
- Increase flow along this path until some edge saturates

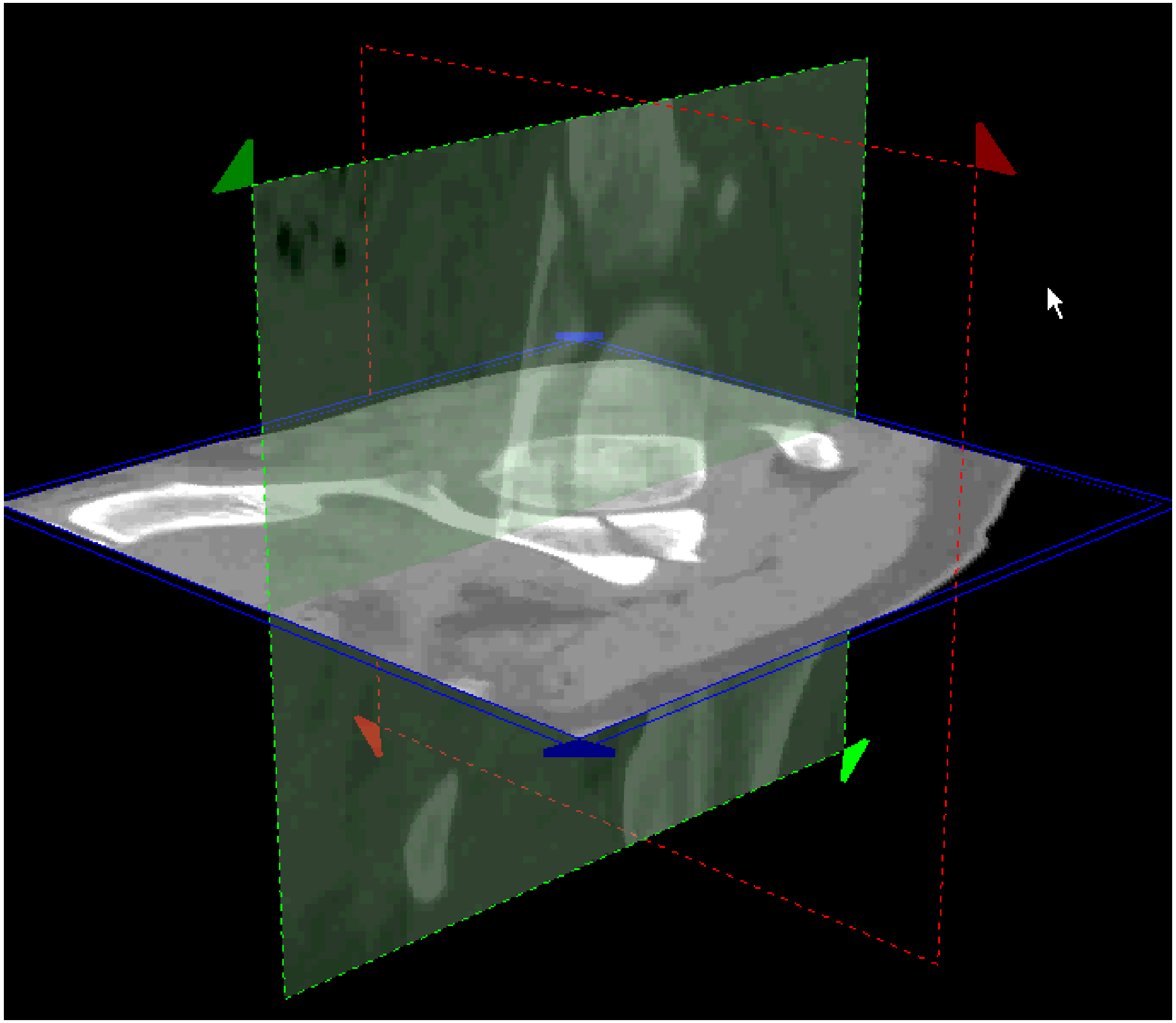
Iterate until … all paths from S to T have at least one saturated edge

Optimal boundary in 2D



"max-flow = min-cut"

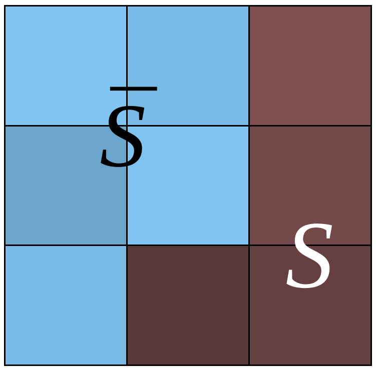
Optimal boundary in 3D



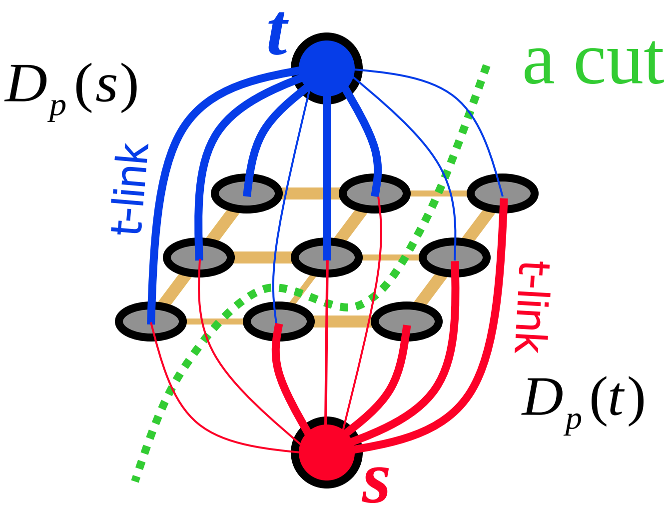
3D bone segmentation (real time screen capture)

Graph cut (region + boundary)

[Boykov and Jolly 2001]



segmentation



assume I^s and I^t are known "expected" intensities of object and background

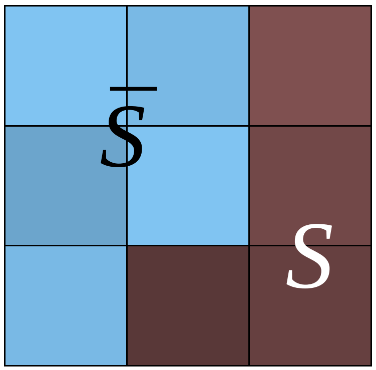
$$D_p(s) = |I_p - I^s|$$
$$D_p(t) = |I_p - I^t|$$

example of soft regional constraints

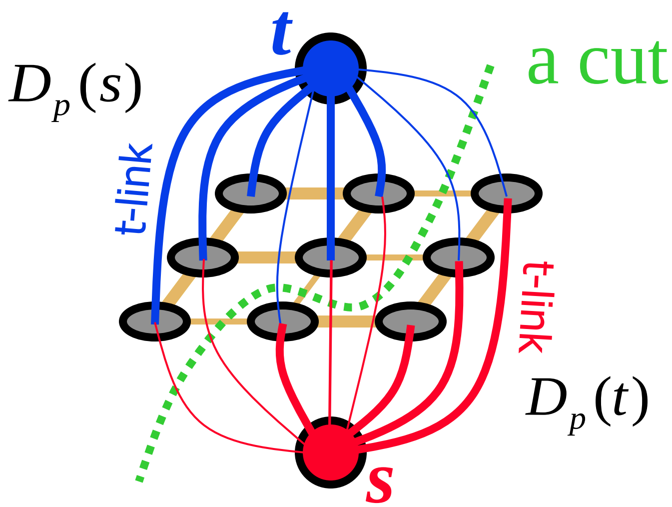
NOTE: seeds were hard constrains on segment's region

Graph cut (region + boundary)

[Boykov and Jolly 2001]



segmentation



in general, assume known intensities distributions of object and background

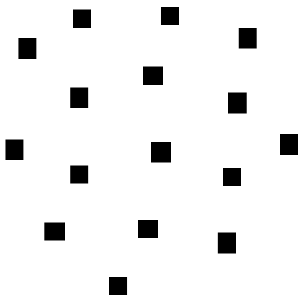
$$D_p(s) = -\ln Pr(I_p \mid s)$$

$$D_p(t) = -\ln Pr(I_p \mid t)$$

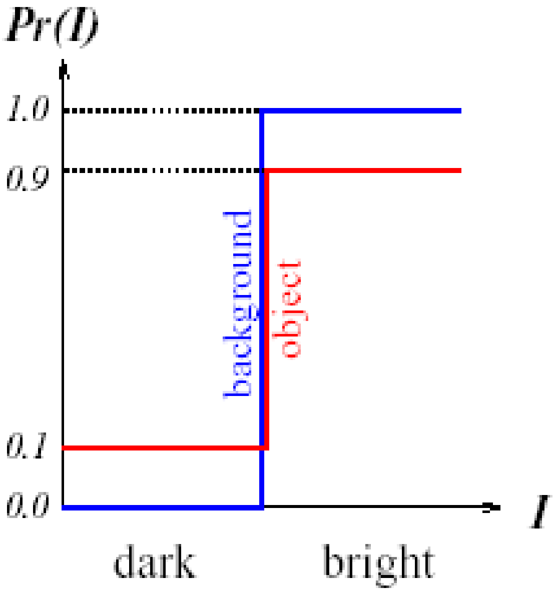
example of soft regional constraints

NOTE: seeds were hard constrains on segment's region

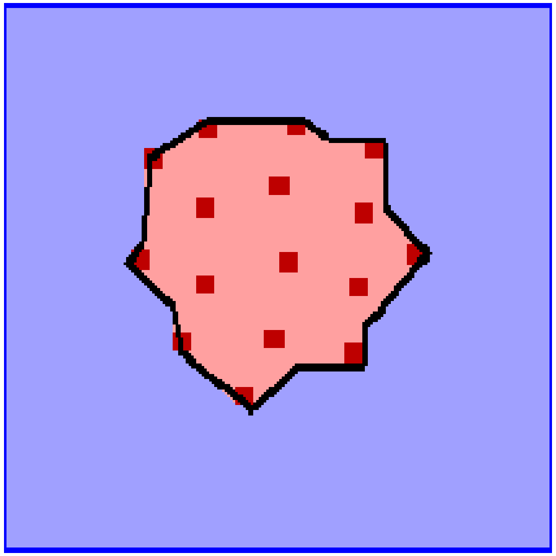
Graph cut (region + boundary)



(a) Original image



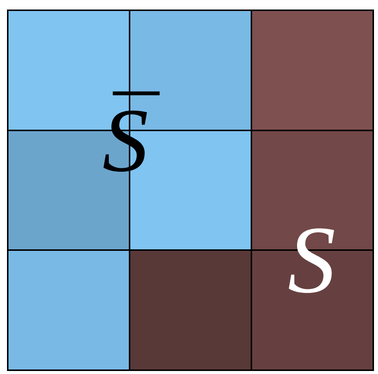
(b) Intensity histograms



(c) Optimal segmentation

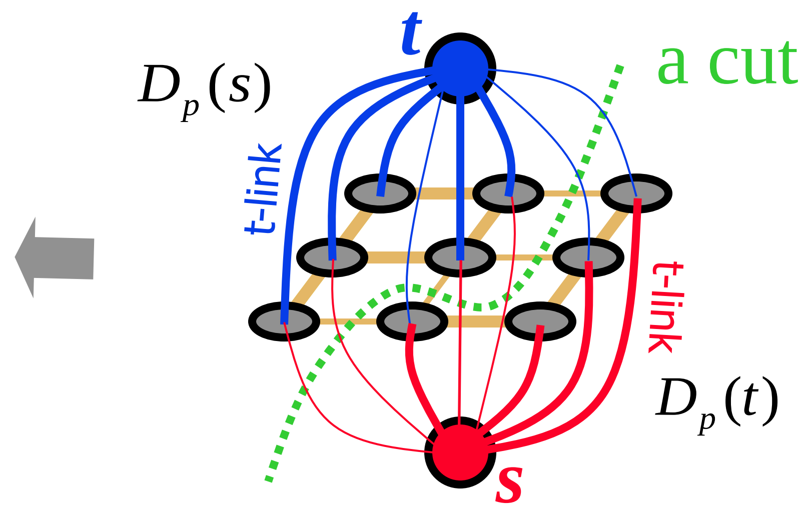
Graph cuts as energy optimization for S

[Boykov and Jolly 2001]



segmentation ⇔ cut

$S_p \in \{0,1\}$



$$E(S) = \sum_p D_p(S_p) + \sum_{pq \in N} w_{pq} \cdot [S_p \neq S_q]$$

cost of severed ***t-links***	cost of severed ***n-links***
unary terms	**pair-wise** terms
regional properties of S	**boundary** smoothness for S

Unary potentials as linear term wrt. $S_p \in \{0,1\}$

unary terms

$$\sum_p D_p(S_p) = \sum_{p \in S} D_p(1) + \sum_{p \in \bar{S}} D_p(0)$$

$$= \sum_p \left(D_p(1) \cdot S_p + D_p(0) \cdot (1 - S_p) \right)$$

$$= const + \sum_p \underbrace{\left(D_p(1) - D_p(0) \right)}_{g(p)} \cdot S_p = \langle \mathbf{g}, S \rangle$$

In general,…

k-arity potentials are *k*-th order polynomial

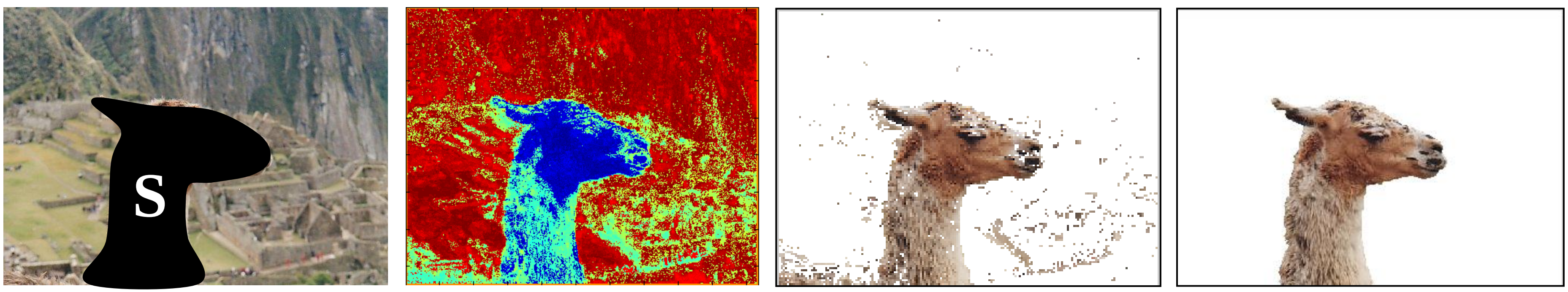
pair-wise terms

$$\sum_{pq \in N} w_{pq} \cdot [S_p \neq S_q] \;=\; \sum_{pq \in N} w_{pq} \cdot \left(S_p \cdot (1 - S_q) + (1 - S_p) \cdot S_q \right)$$

quadratic polynomial wrt. S_p

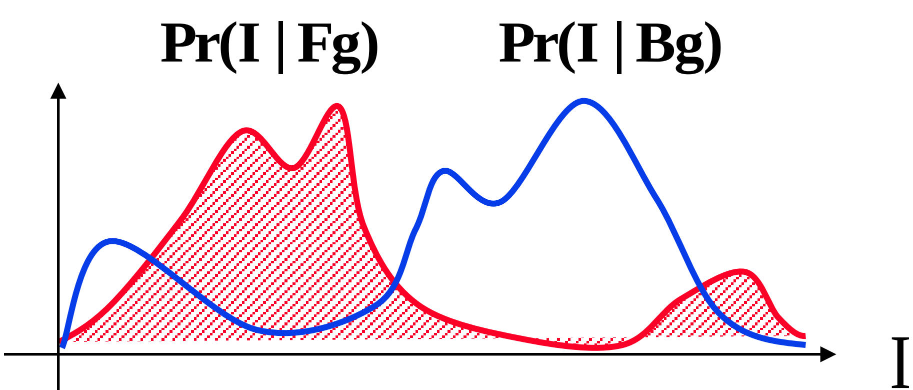
Graph cuts vs Thresholding

$$E(S) = \sum_{p \in S} g(p) + B(S)$$



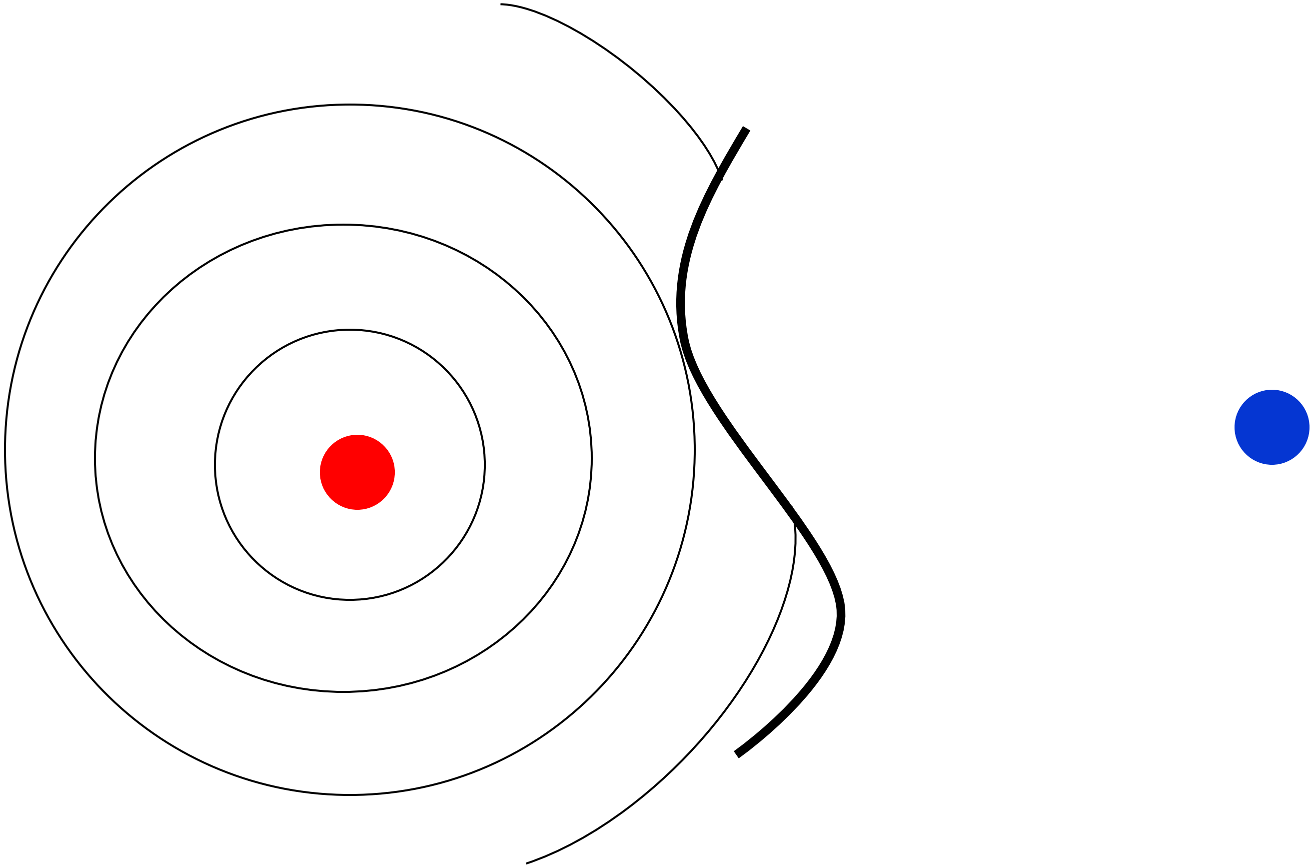
thresholding

graph cut [BJ, 2001]



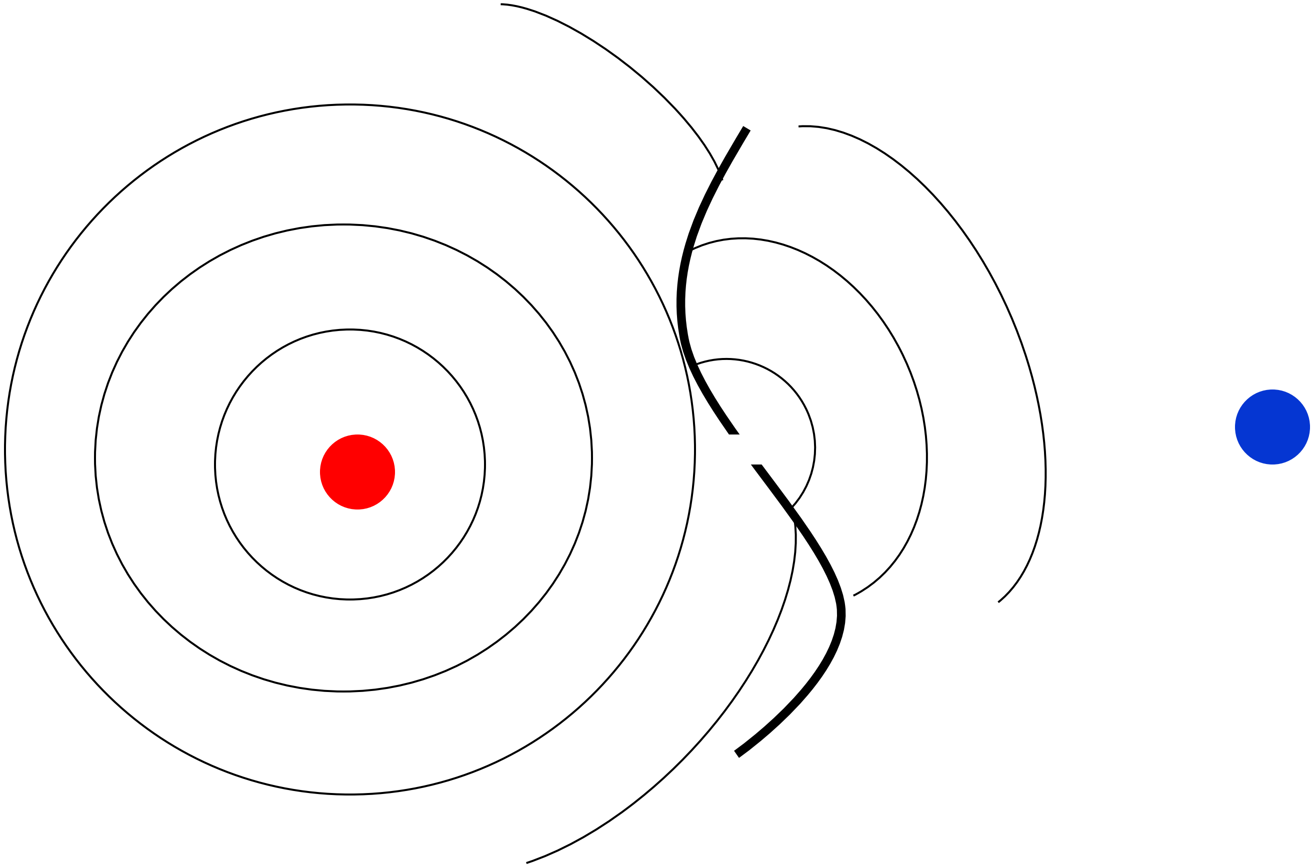
$$g(p) = -\ln\left(\frac{Pr(I(p)|\, fg)}{Pr(I(p)|\, bg)}\right)$$

Graph cuts vs Region Growing



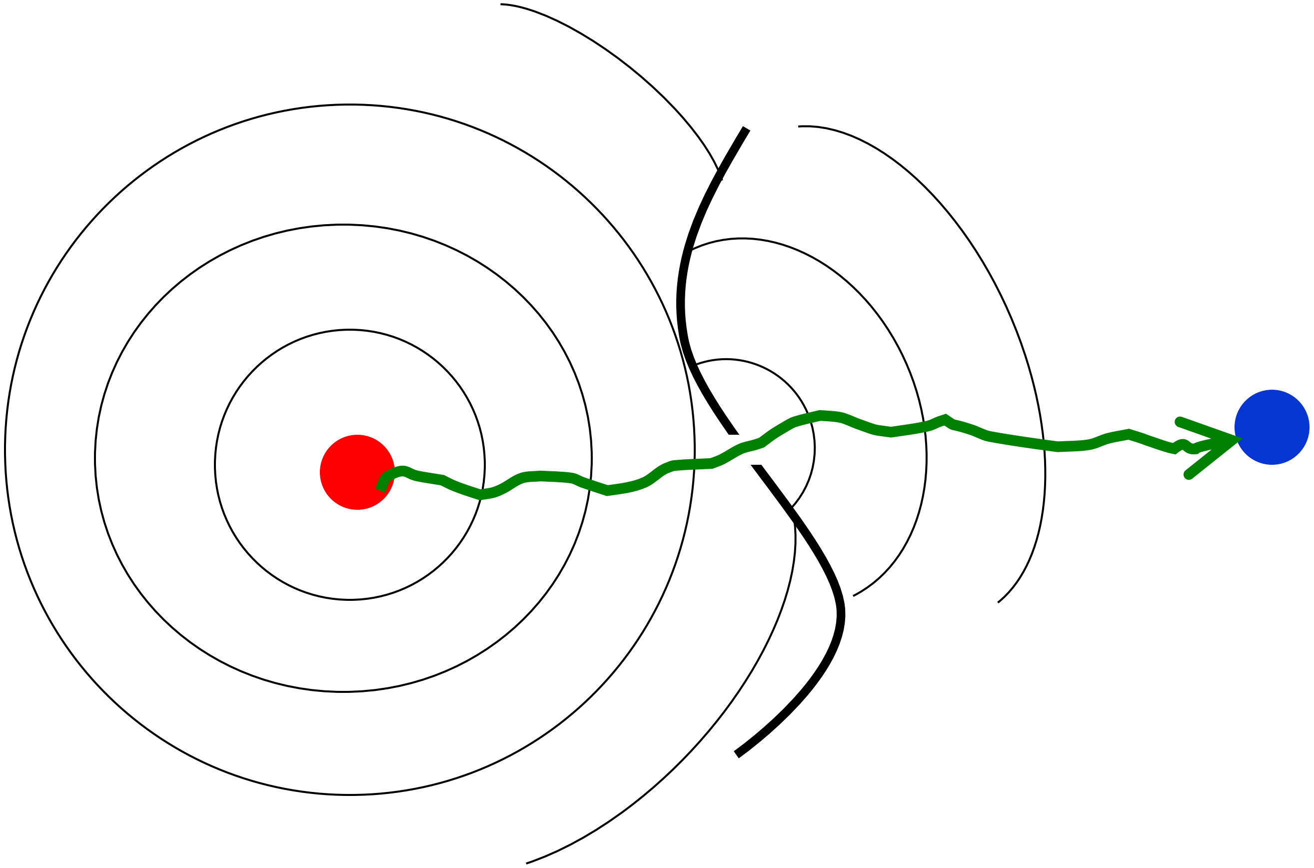
like "region growing"

Graph cuts vs Region Growing



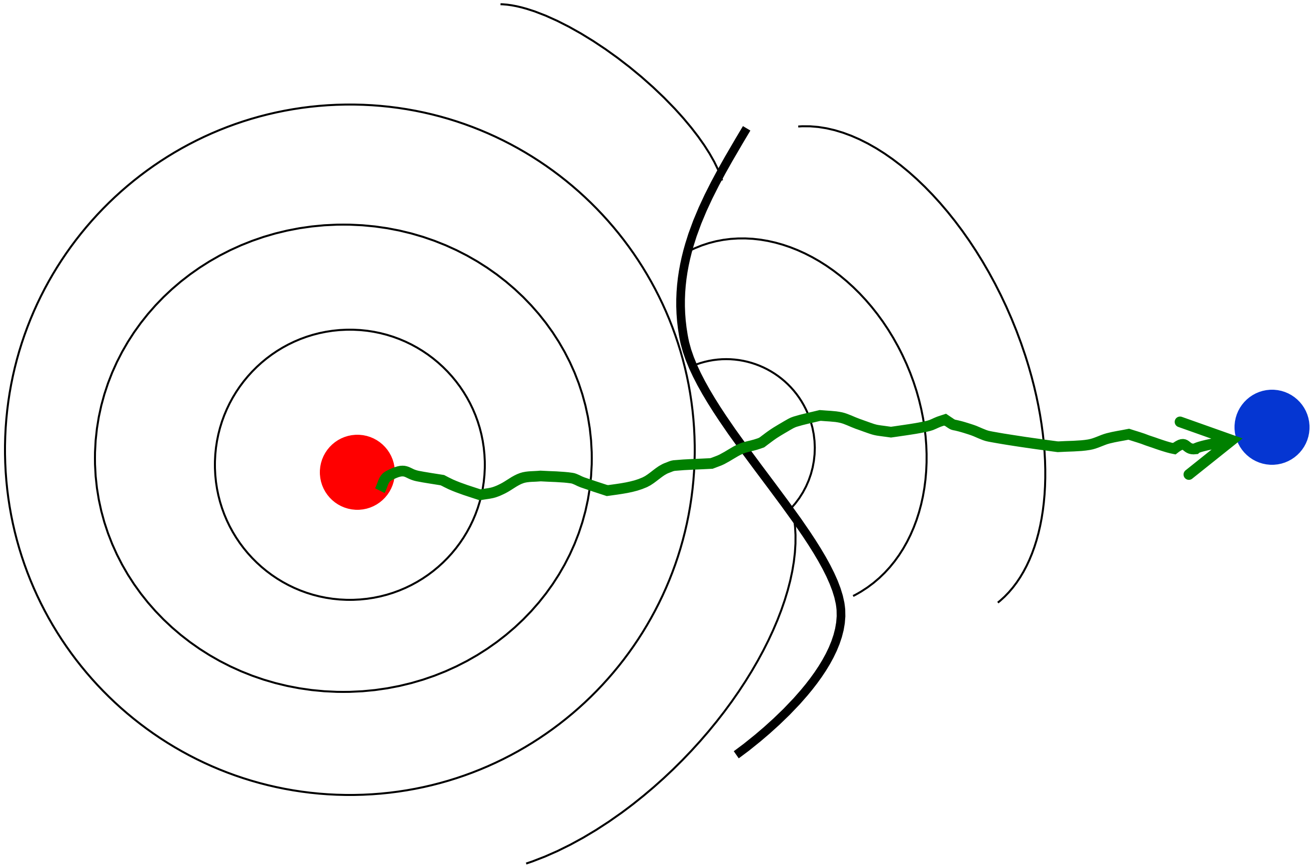
like "region growing"

Graph cuts vs Region Growing



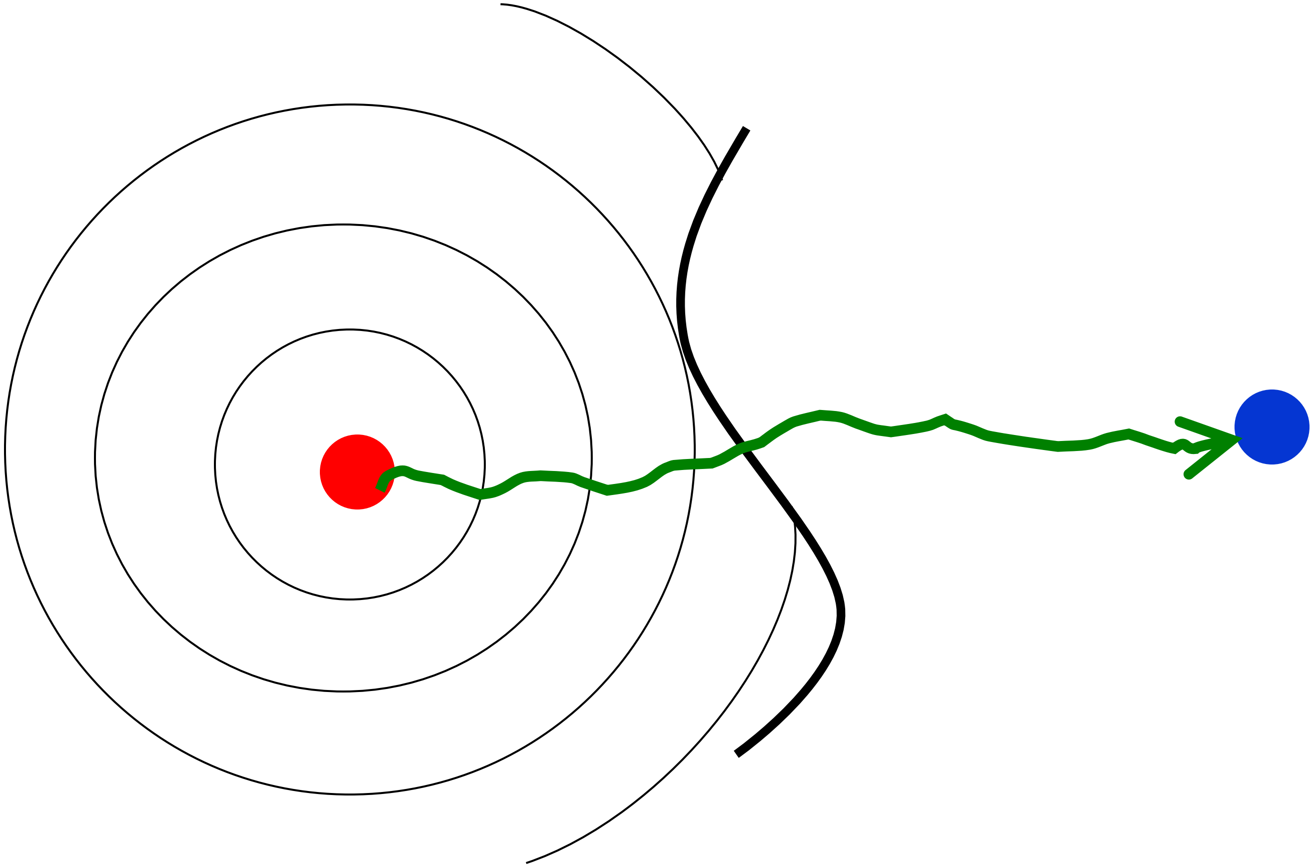
like “region growing”

Graph cuts vs Region Growing

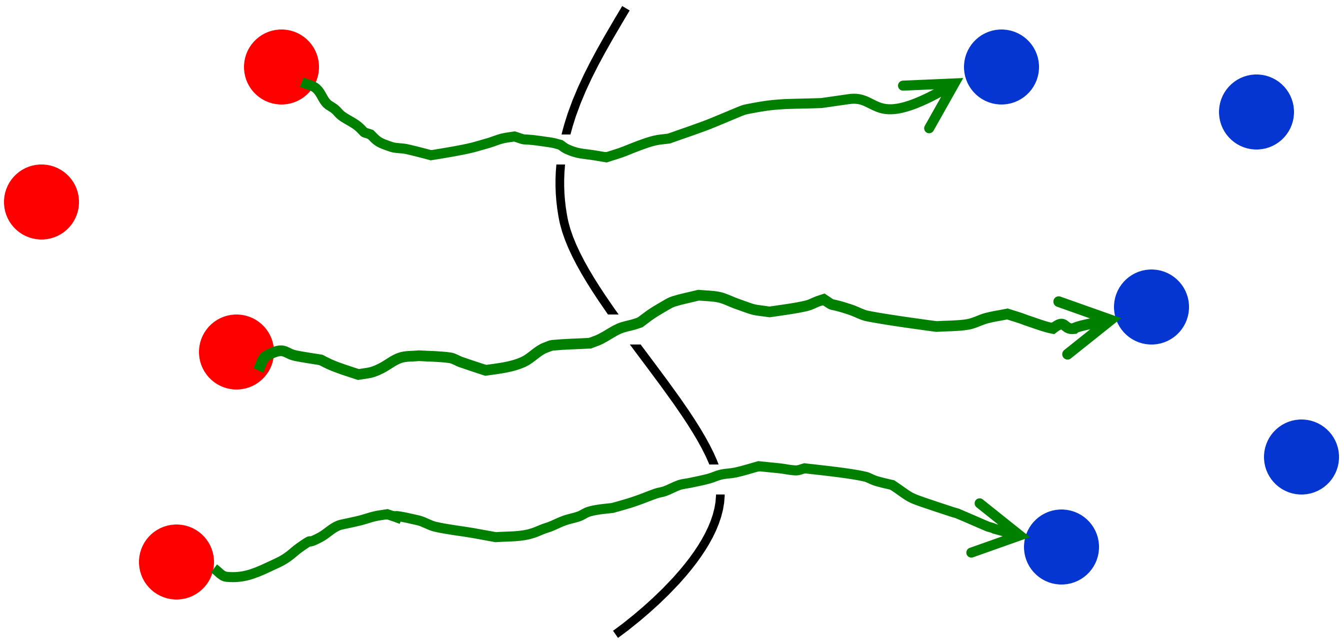


~~like “region growing”~~

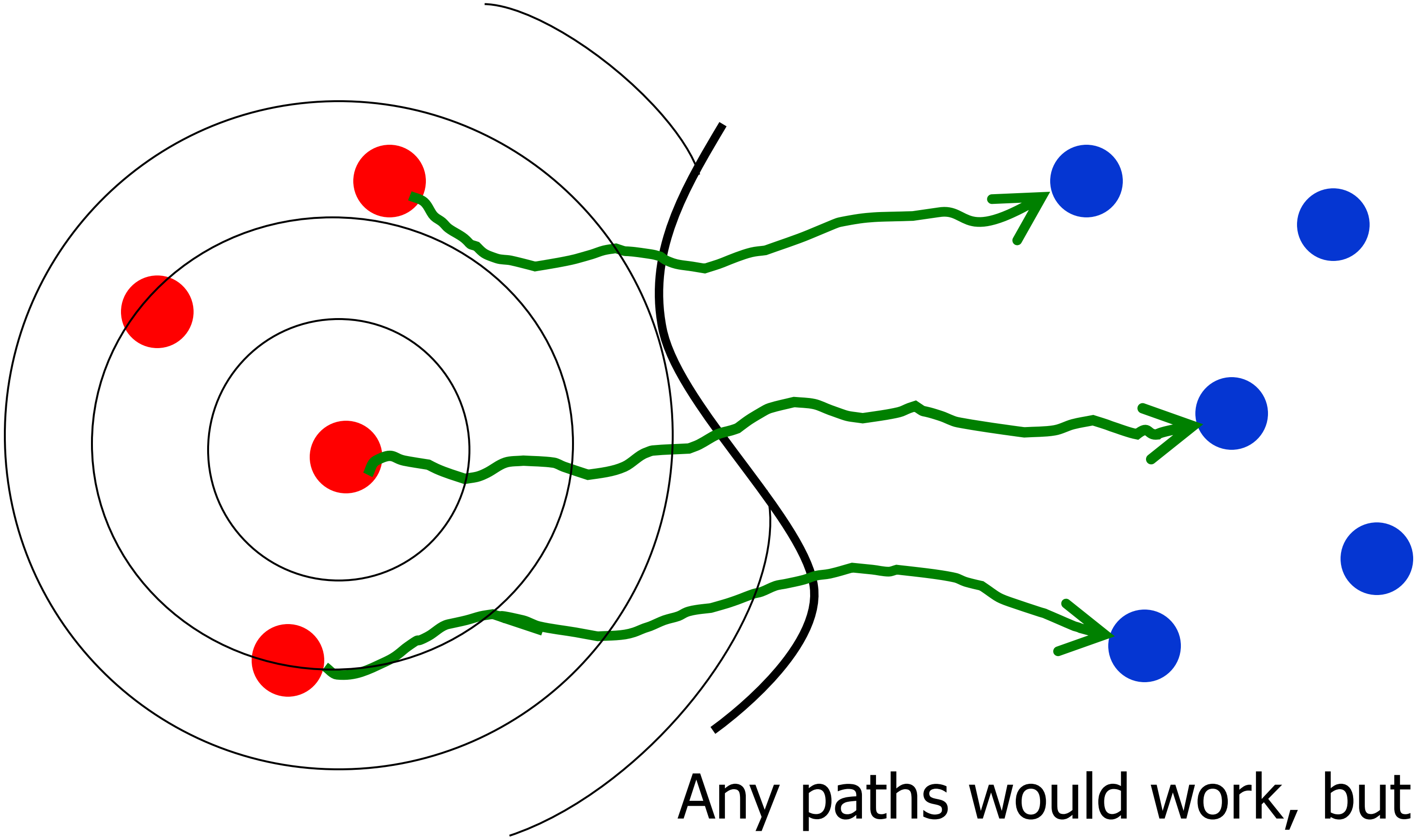
Graph cuts



Graph cuts 2

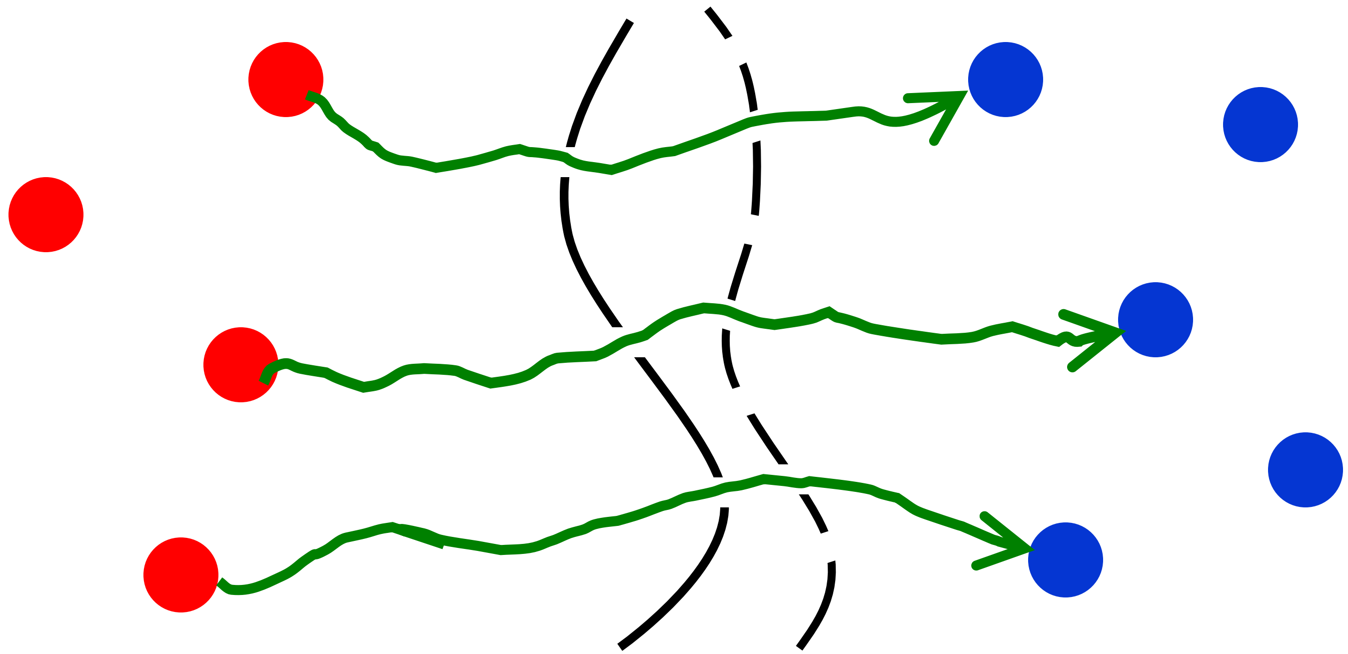


Graph cuts 2

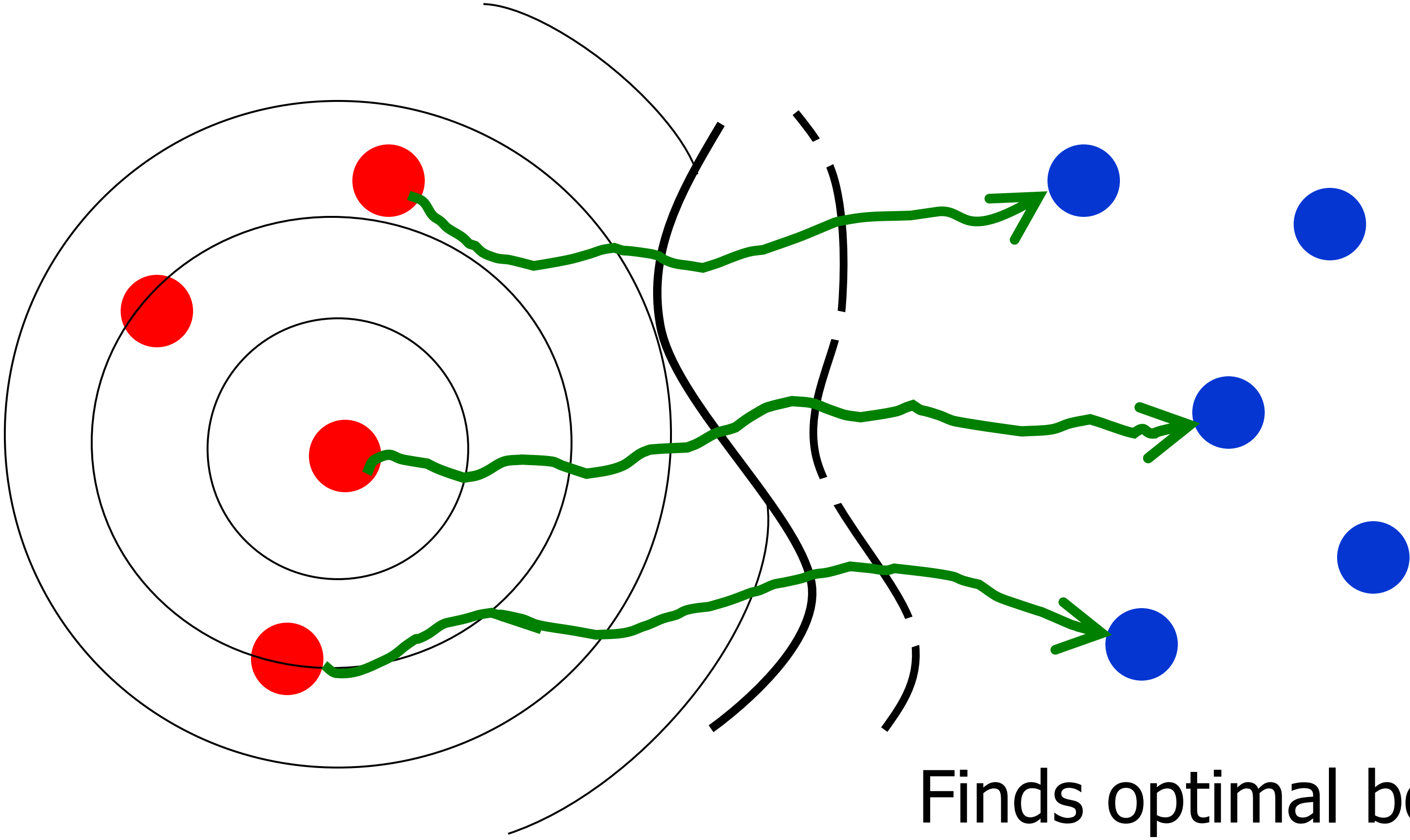


Any paths would work, but
shorter paths give faster algorithms
(in theory and practice)

Graph cuts 3



Graph cuts 3



Finds optimal boundary
(least number of *holes*)

=> Energy minimization

‘Smoothness’ of segmentation boundary

snakes (physics-based contours)

geodesic contours

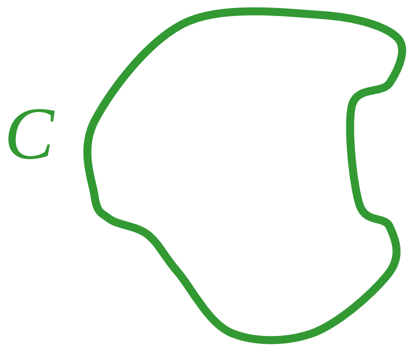
graph cuts

Note: many common distance-based methods do not optimize segmentation boundary directly
(fuzzy connectivity, geodesic Voronoi cells, random walker)

Discrete vs. continuous energies

Geodesic contours

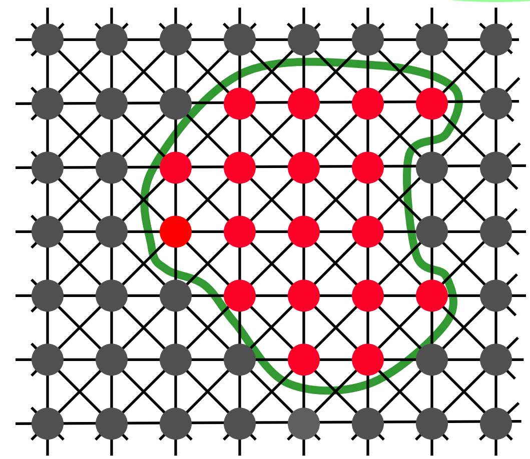
$$E(S) = \int_S g(p)\, dp + \int_{\partial S} w_s \; ds$$



[Caselles, Kimmel, Sapiro, 1997] (level-sets)
[Chan, Esidoglu, Nikolova, 2006] (convex)

Graph cuts

$$E(S) = \sum_{p \in S} g_p + \sum_{p,q} w_{pq} \delta(S_p \neq S_q)$$



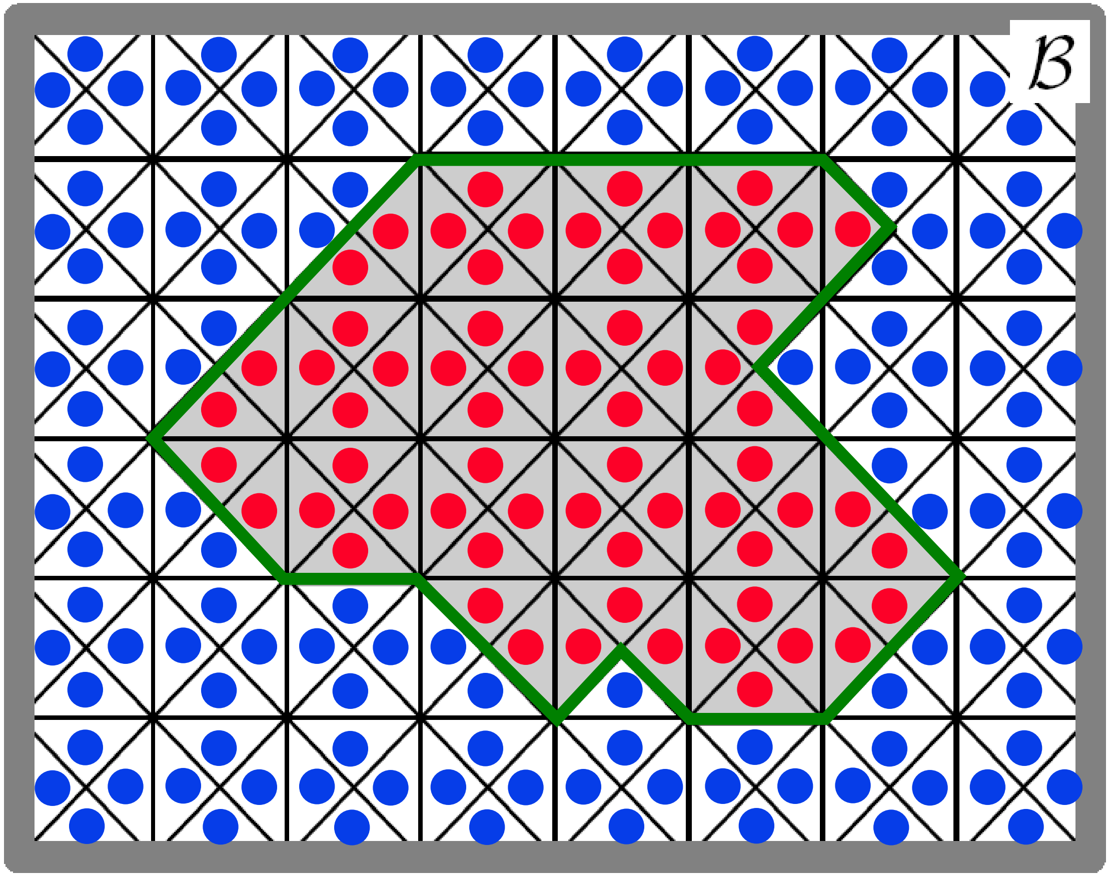
$S_p \in \{0,1\}$

[Boykov and Jolly 2001]

Both incorporate segmentation cues:

- Regional bias
- Boundary smoothness and alignment to image edges

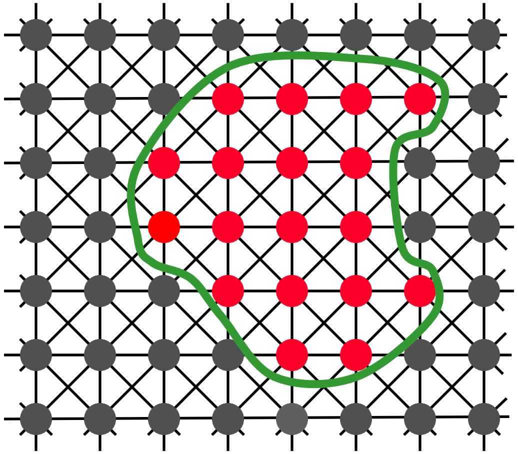
Graph cuts on a complex and boundary of S



John Sullivan'90, Kirsanov&Gortler'04

Graph cuts on a grid and boundary of S

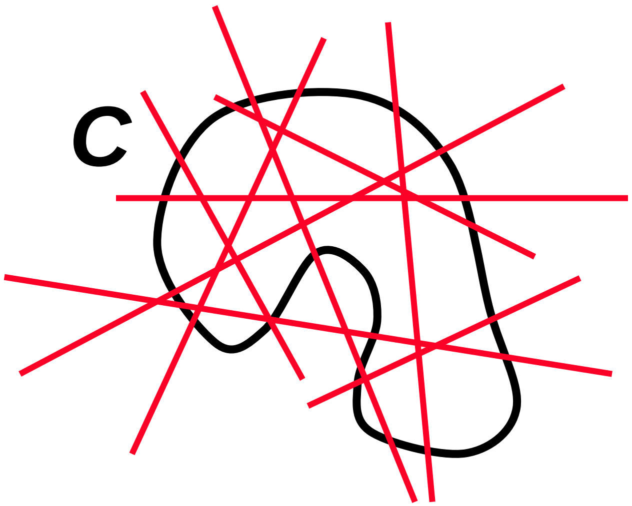
$S_p \in \{0,1\}$

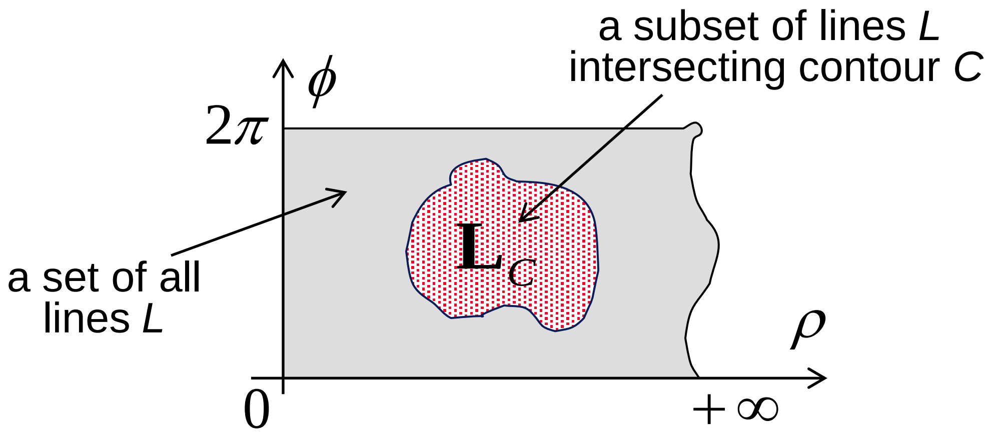


$$B(S) = \sum_{e \in C} |e|$$

- Severed n-links can approximate geometric length of contour ***C***
  [Boykov&Kolmogorov, ICCV 2003]
- This result fundamentally relies on ideas of ***Integral Geometry*** **(**also known as ***Probabilistic Geometry***) originally developed in 1930's.
  - e.g. **Blaschke**, **Santalo**, **Gelfand**

Integral geometry approach to *length*





probability that a "randomly drown" line intersects *C*

Euclidean length of **C** :

$$\| C \|_{\varepsilon} = \frac{1}{2} \int n_L \cdot d\rho \cdot d\phi$$

Cauchy-Crofton formula

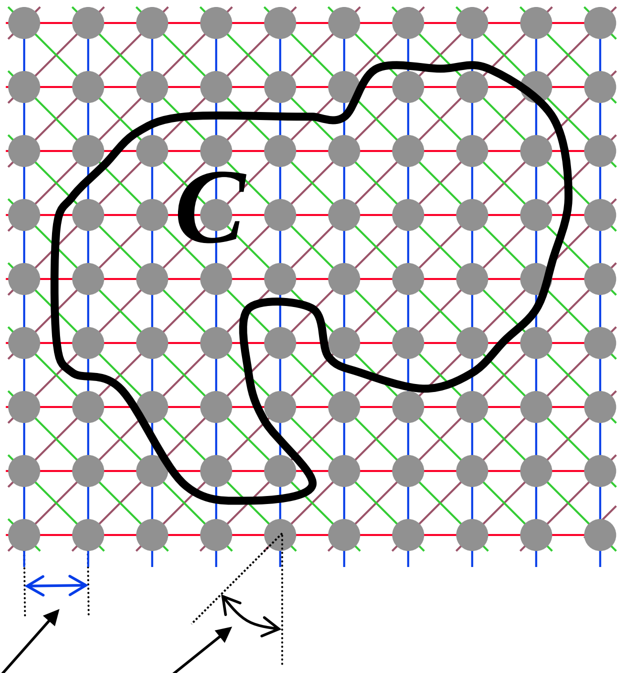
the number of times line *L* intersects *C*

Graph cuts and integral geometry

Graph nodes are imbedded in R2 in a grid-like fashion

Edges of any regular neighborhood system generate families of lines

{—, /, |, \}



$$\| C \|_{\varepsilon} \approx \tfrac{1}{2} \sum_k n_k \cdot \Delta\rho_k \cdot \Delta\phi_k = \| C \|_{gc}$$

Euclidean length

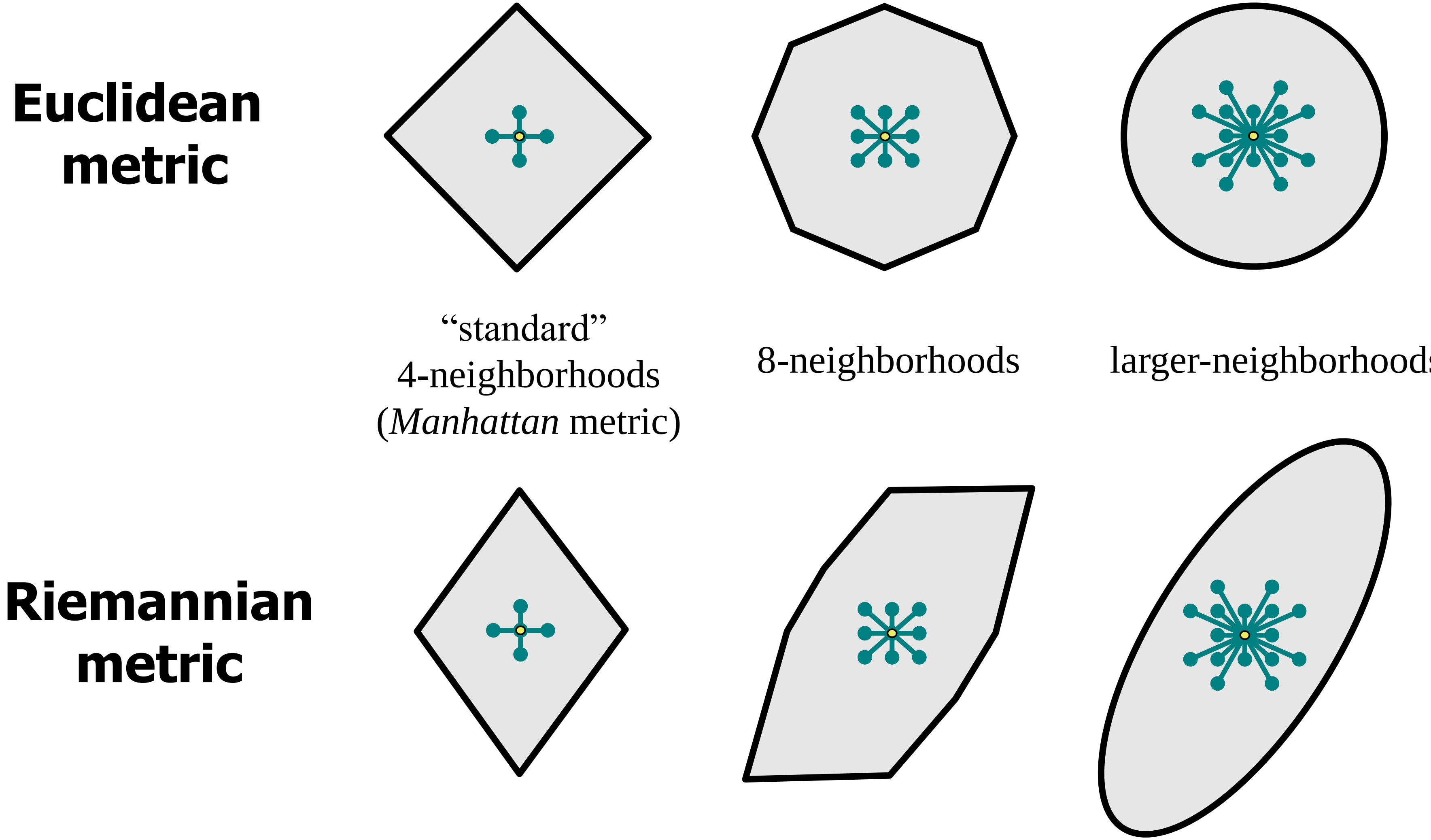
the number of edges of family k intersecting *C*

graph cut cost for edge weights:

$$w_k = \frac{\Delta\rho_k \cdot \Delta\phi_k}{2}$$

Length can be estimated without computing any derivatives

Metrication errors



Metrication errors

4-neighborhood

8-neighborhood

Differential vs. integral approach to length

Differential geometry

$$\| C \|_{\varepsilon} = \int_0^1 C'_t \cdot dt$$

Parametric (explicit) contour representation

$$\| C \|_{\varepsilon} = \int_{\Omega} | \nabla u | \, dx$$

Level-set function representation

Integral geometry

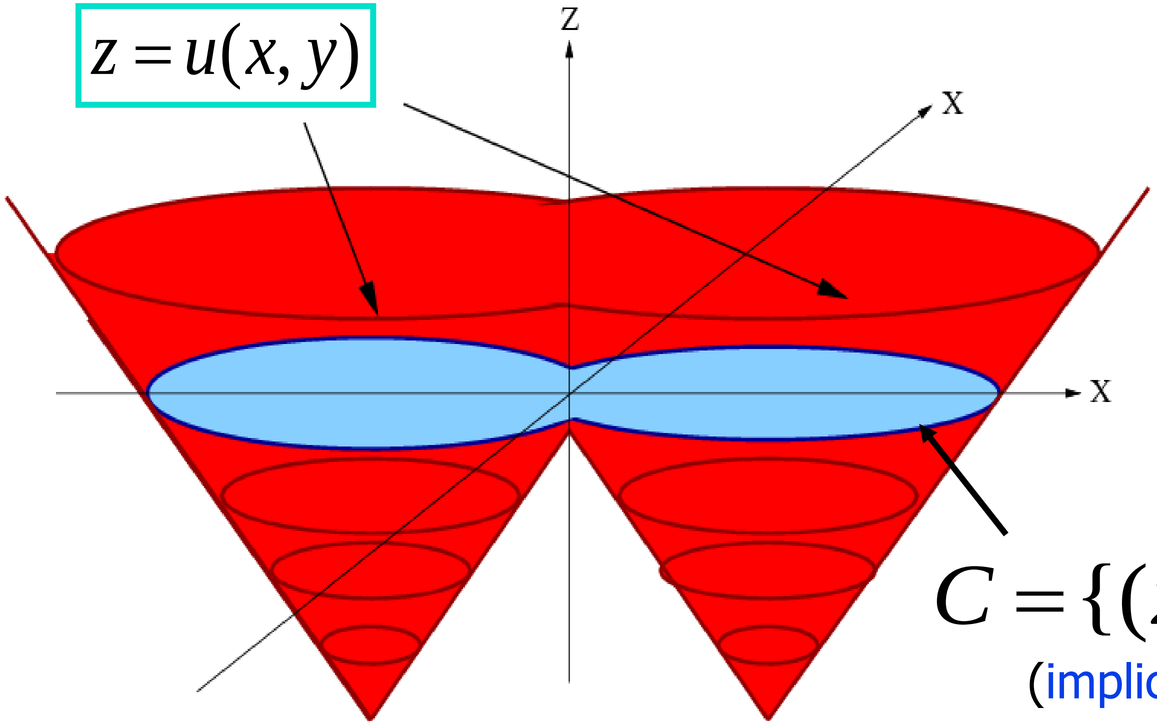
$$\| C \|_{\varepsilon} = \frac{1}{2} \int n_L \cdot d\rho \cdot d\phi$$

Cauchy-Crofton formula

implicit (region-based) representation of contours

Implicit (region-based) surface representation via *level-sets*

$z = u(x, y)$



$$C = \{(x, y) : u(x, y) = 0\}$$

(implicit contour representation)

[Dervieux, Thomasset, 79, 81] [Osher, Sethian, 89]

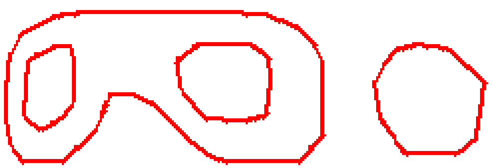
Note: 0.5-level set for $u : \Re^n \to [0,1]$ in convex formulations

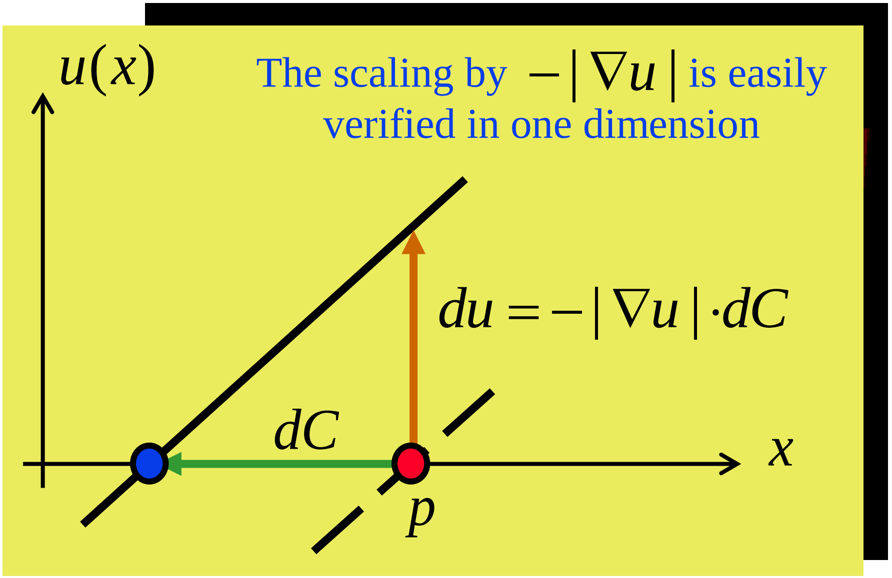
[Chan, Esidoglu, Nikolova, 2006] (convex)

Implicit (region-based) surface representation via *level-sets*

$$d\vec{C} = \beta \cdot \vec{N} \quad \Leftrightarrow \quad du_p = -\beta_p \cdot |\nabla u_p|$$

[Dervieux, Thomasset, 79, 81] [Osher, Sethian, 89]

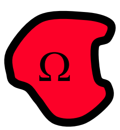
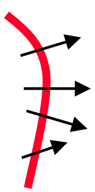




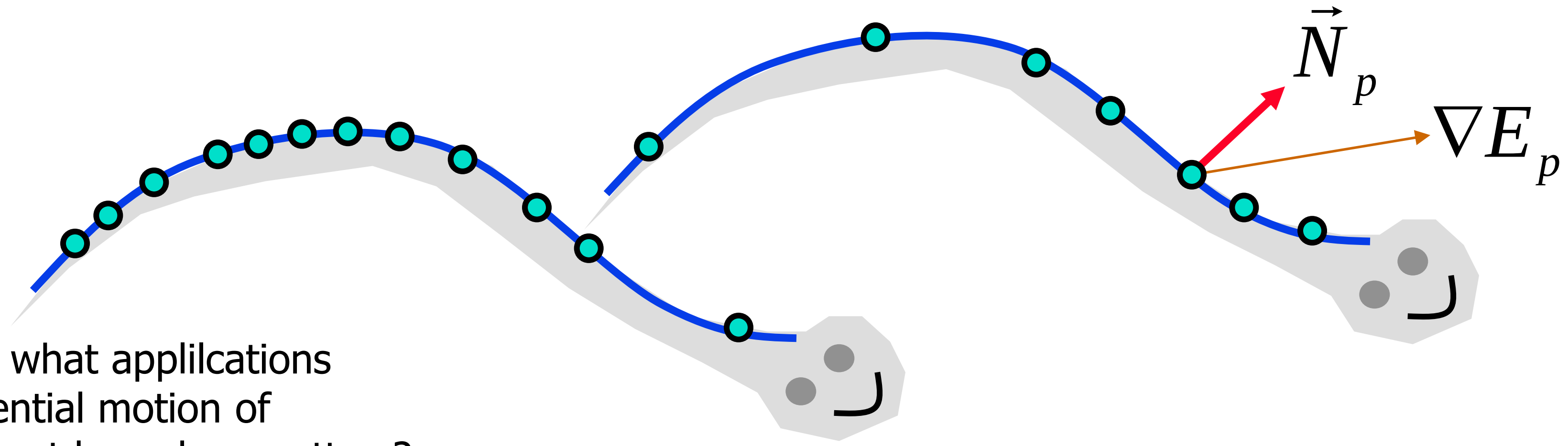
For example, *level sets* can compute contour's gradient descent $d\vec{C} = -dt \cdot \nabla E$ for **geometric energies** of contour C where ∇E is collinear with normal $\vec{N}$

Implicit (region-based) surface representation via *level-sets*

Geometric measures commonly used in segmentation

		<u>Functional *E(C)*</u>	<u>gradient descent evolution</u> $dC = \beta \cdot \vec{N}$
	weighted length	$E(C) = \int_C g(\cdot)\ ds$	$\beta \sim g \cdot \kappa - \langle \nabla g, \vec{N} \rangle$
	weighted area	$E(C) = \iint_\Omega f\ da$	$\beta \sim f$
	alignment (flux)	$E(C) = \int_C \langle \vec{\mathbf{v}}, \vec{N} \rangle\ ds$	$\beta \sim \mathrm{div}(\vec{\mathbf{v}})$

Note: physic-based energy of snake depends on contour parameterization



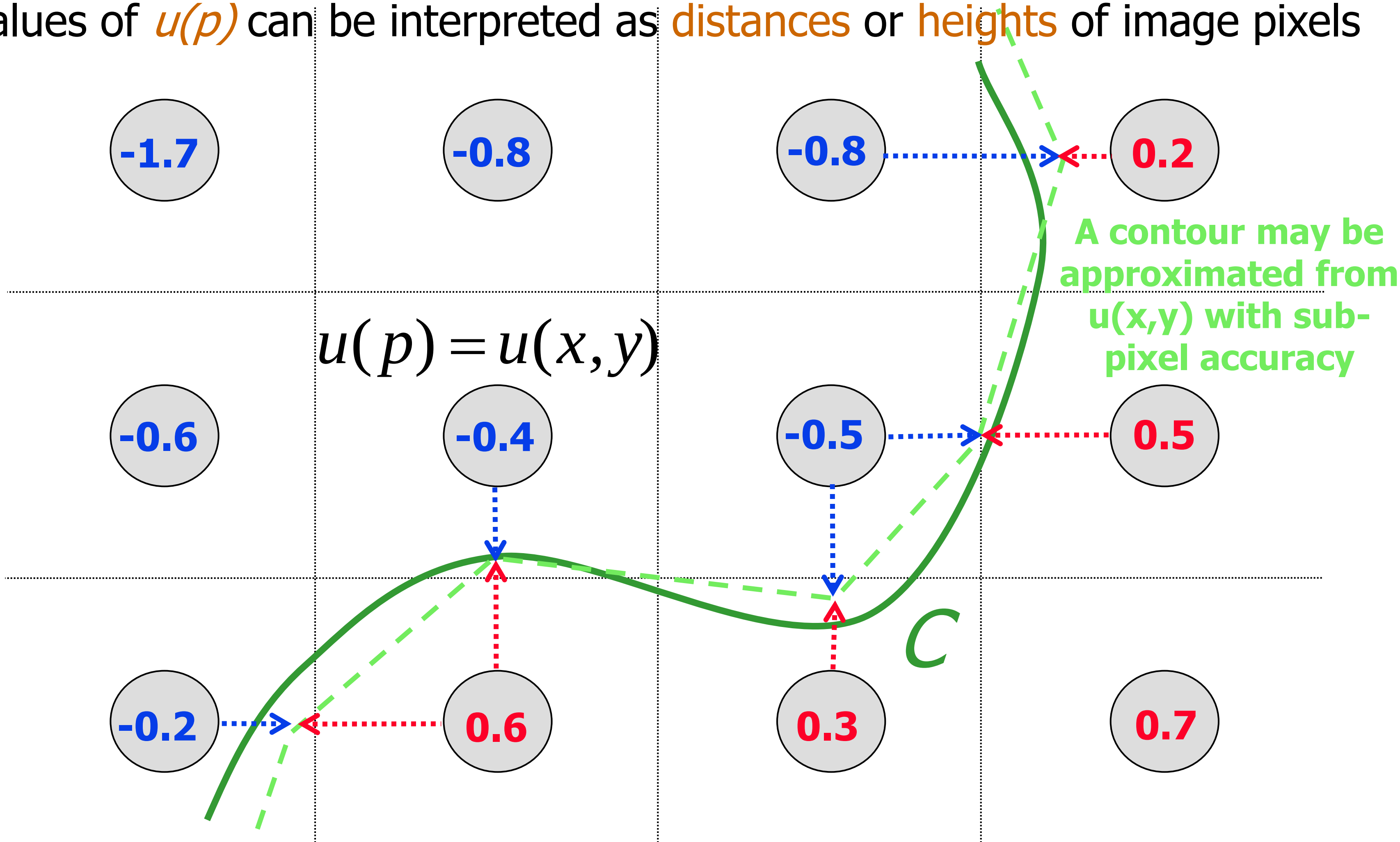
Q: in what applilcations tangential motion of segment boundary matters?

Gradient descent of snake *C* produces tangential motion, which is "invisible" in segmentation.

In contrast, geometric energies give ∇E collinear with $\vec{N}$
can be minimized via **level-sets**

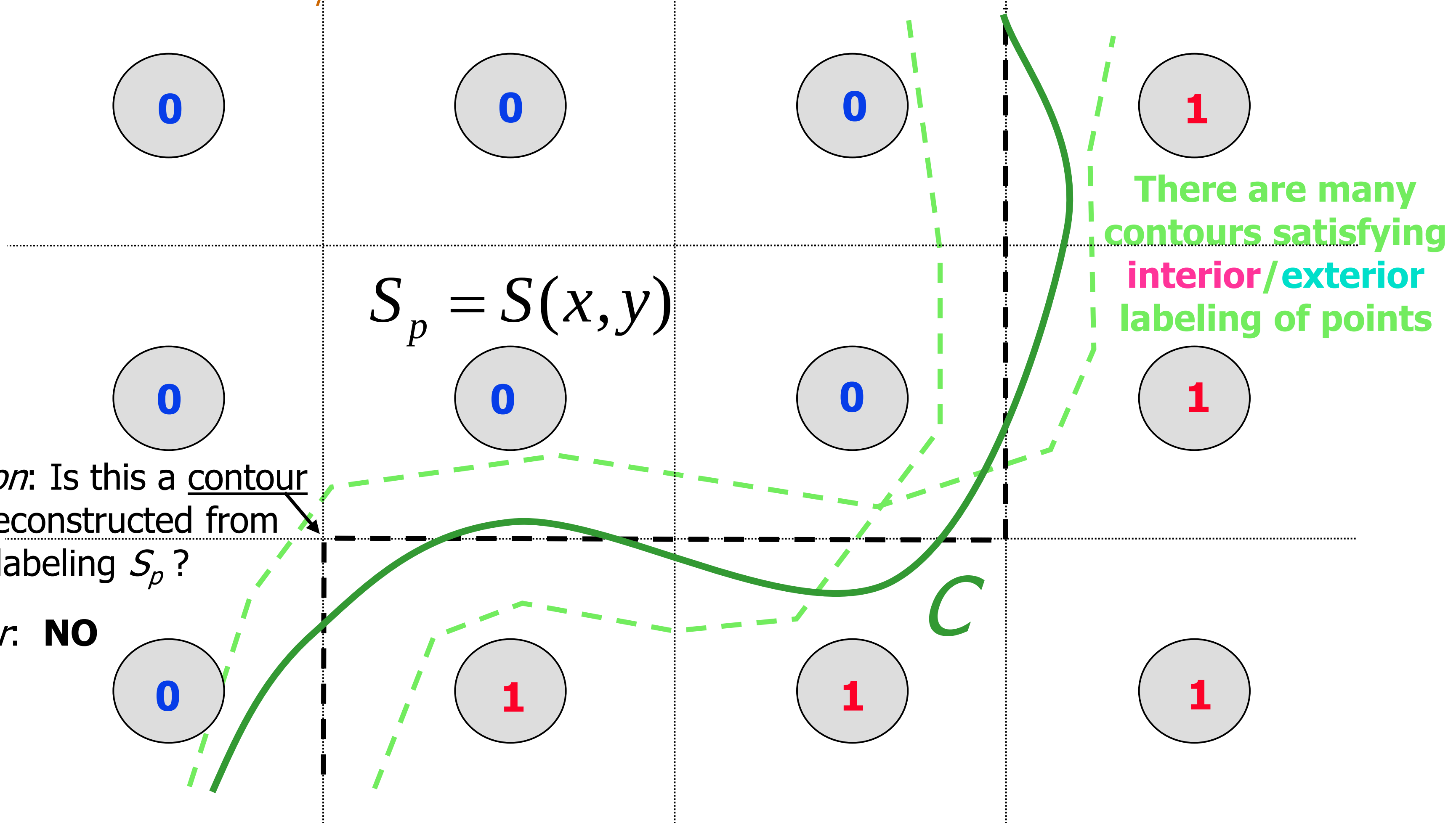
Implicit (region-based) surface representation via *level-sets*

- Level set function *u(p)* is normally stored on image pixels
- Values of *u(p)* can be interpreted as distances or heights of image pixels



Implicit (region-based) surface representation via *graph-cuts*

- Graph cuts represent surfaces via binary labeling S_p of each graph node
- Binary values of S_p indicate interior or exterior points (e.g. pixel centers)



Question: Is this a contour to be reconstructed from binary labeling S_p ?

Answer: **NO**

Contour/surface representations (summary)

Implicit (area-based)	Explicit (boundary-based)
Level sets (geodesic active contours, convex geometric energies)	Snakes (physics-based band model)
	Live-wire (shortest paths on graphs)
Graph cuts (grids or complexes)	Graph cuts on complexes

Different ways to look at energy of graph cuts

1: Posterior energy (MAP-MRF)

$$E(S) \;=\; \sum_{p} -\ln Pr(D_p \mid S_p) \;+\; \sum_{pq \in N} V_{pq}(S_p, S_q)$$

log-likelihoods log of prior

2: Approximating continuous surface functional

$$E(S) = \int_{S} g(\cdot)\, ds \;+\; \int_{\partial S} \langle \vec{N}, \vec{v}_s \rangle \, ds \;+\; \int_{\partial S} w(s)\, ds$$

regional term flux boundary length

3: Submodular set function

$$E(S) \;=\; \sum_{A} E_A(S_A) \qquad \text{for } S_A = \{S_p \mid p \in A\}$$

factors

Submodular functions

- Edmonds 1970

Lattice $(\mathcal{L}, \wedge, \vee)$ - set of elements with *inf* and *sup* operations

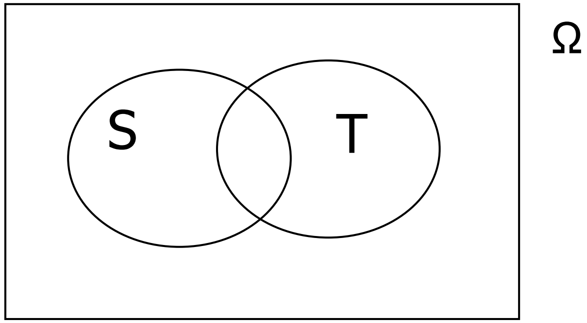
$$S, T \in \mathcal{L} \quad \Rightarrow \quad S \wedge T \in \mathcal{L} \quad S \vee T \in \mathcal{L}$$

Function $\mathrm{E} : \mathcal{L} \rightarrow \Re$ s called **submodular** if for any $S, T \in \mathcal{L}$

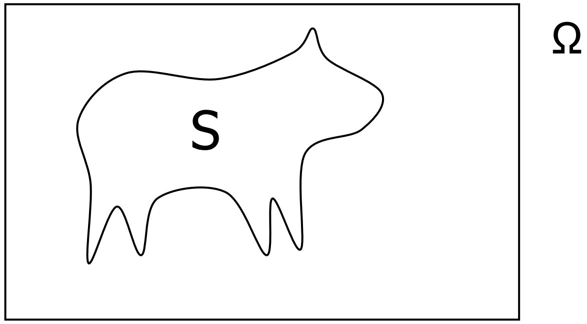
$$E(S \wedge T) + E(S \vee T) \leq E(S) + E(T)$$

Lattice of sets and set functions

Assume set Ω, then $(2^{\Omega}, \cap, \cup)$ is a lattice of subsets



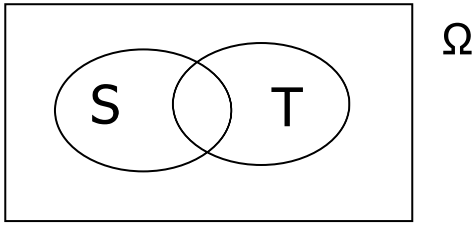
NOTE: if Ω is a set of pixels then any (binary) segmentation energy *E(S)* is a set function $E: \; 2^{\Omega} \rightarrow \Re$



Submodular set functions

Set function $\mathrm{E} : 2^{\Omega} \rightarrow \Re$ is **submodular** if for any $S, T \subseteq \Omega$

$$E(S \cap T) + E(S \cup T) \leq E(S) + E(T)$$

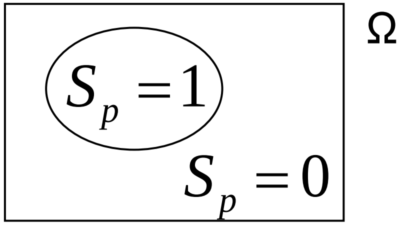


Significance: any submodular set function can be globally optimized in polynomial time $O\left(|\Omega|^9\right)$ [Grotschel et al.1981,88, Schrijver 2000]

Submodular set functions

Sets are conveniently represented by binary indicator variables

$$S \subset \Omega \quad \leftrightarrow \quad \left\{ S_p \in \{0,1\} \mid p \in \Omega \right\}$$



Thus, set functions $\mathrm{E} : 2^{\Omega} \to \Re$ can be represented as

$$E(S) = E(S_1, S_2, \ldots, S_{|\Omega|})$$

Define $S_A = \left\{ S_p \mid p \in A \right\}$, a *restriction* of S to any subset $A \subseteq \Omega$ and consider *projections* $E(S_A \mid S_{\Omega \setminus A})$ of energy E onto subsets A

Set function $E(S)$ is **submodular** iff for any pair $p, q \in \Omega$

$$E(\mathbf{0},\mathbf{0} \mid S_{\Omega \setminus pq}) + E(\mathbf{1},\mathbf{1} \mid S_{\Omega \setminus pq}) \leq E(\mathbf{1},\mathbf{0} \mid S_{\Omega \setminus pq}) + E(\mathbf{0},\mathbf{1} \mid S_{\Omega \setminus pq})$$

Graph cuts for minimization of submodular set functions

Assume set Ω and *2nd*-order (quadratic) function

$$E(S) = \sum_{(pq)\in N} E_{pq}(S_p, S_q) \qquad S_p, S_q \in \{0,1\}$$

Indicator variables

Function *E(S)* is **submodular** if for any $(p,q) \in N$

$$E_{pq}(\mathbf{0,0}) + E_{pq}(\mathbf{1,1}) \le E_{pq}(\mathbf{1,0}) + E_{pq}(\mathbf{0,1})$$

Significance: submodular 2nd-order boolean (set) function can be globally optimized in polynomial time by **graph cuts**

[Hammer 1968, Pickard&Ratliff 1973] $O\left(|N| \cdot |\Omega|^2\right)$

[Boros&Hammer 2000, Kolmogorov&Zabih2003]

Graph cuts for approximating continuous surface functionals

Submodular quadratic boolean functions on a grid can approximate continuous geometric functionals

$$E(S) = \int_{\partial S} g(\cdot)\, ds + \int_{\partial S} \langle \vec{N}, \vec{v}_x \rangle\, ds + \int_{S} f(p)\, dp$$

Geometric length
any convex, symmetric metric ***g***
e.g. Riemannian

Flux
any vector field **v**

Regional bias
any scalar function ***f***

[Boykov&Kolmogorov, ICCV 2003]
[Kolmogorov&Boykov, ICCV 2005]

Graph cuts for minimization of posterior energy (MRF)

Assume **Gibbs distribution** over binary variables $S_p \in \{0,1\}$

$$Pr(S_1,\ldots,S_n) \propto exp\left(-\sum_A \underbrace{E_A(S_A)}_{\text{factors}}\right) \qquad S_A = \{S_p \mid p \in A\}$$

Theorem [*Boykov, Delong, Kolmogorov, Veksler* in unpublished book 2013?]

Any pair of random variables S_p and S_p are **positively correlated** iff function

boolean (set) function $E(S) = \sum_A E_A(S_A)$ is **submodular**

That is, submodularity implies MRF with "smoothness" prior

Graph cuts for minimization of posterior energy (MRF)

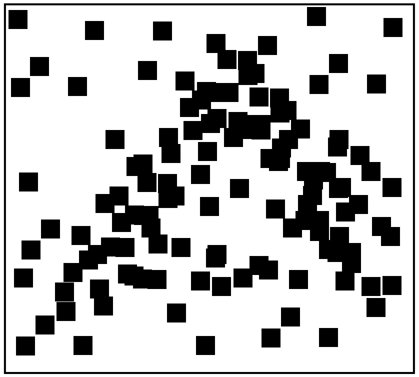
$$E(L) = \sum_{p} (I_p - L_p)^2 + \sum_{pq \in N} V_{pq}(L_p, L_q)$$

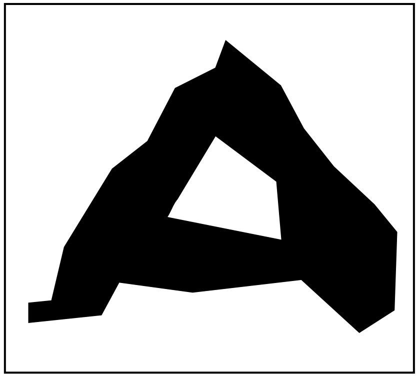
$L_p \in \{0,1\}$

Log-Likelihood (data term)

Spatial prior (regularization)

$I_p \in \{0,1\}$





binary image restoration

[Greig at. al., IJRSSB, 1989]

Graph cuts for minimization of posterior energy (MRF)

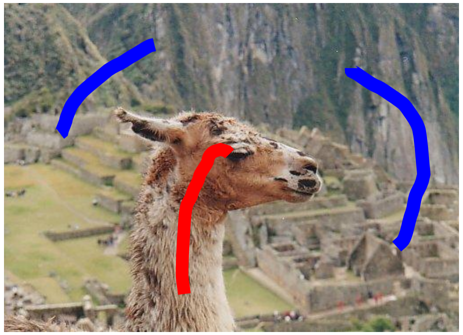
$$E(L\,|\,\theta_0,\theta_1) \;=\; \sum_p -\ln \Pr(I_p\,|\,\theta_{L_p}) \;+\; \sum_{pq\in N} w_{pq}[L_p \neq L_q] \qquad L_p \in \{0,1\}$$

Log-Likelihood (data term)

Spatial prior (regularization)

assuming known

$I_p \in RGB$



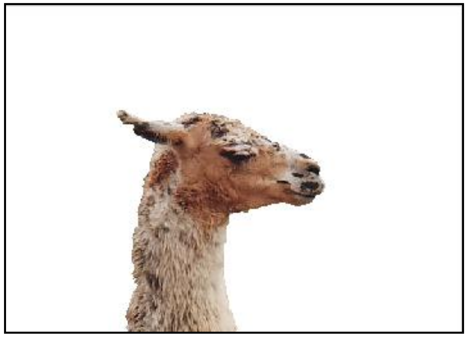


image segmentation, graph cut [Boykov&Jolly, ICCV2001]

Beyond given appearance models

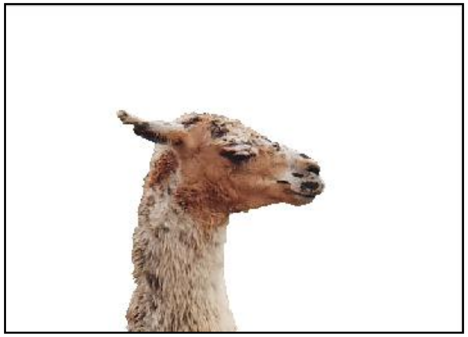
$$E(L,\theta_0,\theta_1) = \sum_p -\ln \Pr(I_p \mid \theta_{L_p}) + \sum_{pq \in N} w_{pq}[L_p \neq L_q] \qquad L_p \in \{0,1\}$$

extra variables

Log-Likelihood (data term)

Spatial prior (regularization)

$I_p \in RGB$



Models θ_0, θ_1 can be iteratively re-estimated

iterative image segmentation, Grabcut
(block coordinate descent $L \leftrightarrow \theta_0, \theta_1$)
[Rother, et al. SIGGRAPH2006]

Beyond submodularity

- Many useful non-submodular set functions *E(S)*
  - in binary segmentation with learned interaction potentials
  - in the context of binary moves for multi-label (later)

- QPBO (partial optimality) [survey Kolmogorov&Rother, 2007]
- LP relaxations [Schlezinger, Komodakis, Kolmogorov, Savchinsky,…]
- Message passing, e.g. TRWS [Kolmogorov]
- active area of research…

Beyond submodularity

Deconvolution

image I blurred with mean kernel

$$E(L) = \sum_p \left(I_p - \tfrac{1}{|B|} \cdot \sum_{q \in B_p} L_q\right)^2 + \sum_{pq \in N} w[L_p \neq L_q]$$

non-submodular quadratic term

submodular quadratic term

Beyond loglikelihoods and length-based smoothness

$$E(S)= E_1(S)+\ldots+ E_n(S)$$

- Shape bias
  - star-shape (one click) [Veksler 2008]
  - shape statistics [Cremers 2003]
  - box prior [Lempitsky 2009]
- Curvature of the boundary (like bending in snakes)
- Cardinality constraints
- Distribution constraints
- Sparsity or MDL prior, label costs
- Many others….

Beyond linear combination of terms

- Ratios are also used

$$E(S) = \frac{E_1(S)}{E_2(S)}$$

  - Normalized cuts [Shi, Malik, 2000]
  - Minimum Ratio cycles [Jarmin Ishkawa, 2001]
  - Ratio regions [Cox et al, 1996]
  - Parametric max-flow applications [Kolmogorov et al 2007]

Segmentation principles

interactive **vs.** **unsupervised**

- Boundary seeds
  - Livewire (intelligent scissors)
- Region seeds
  - Graph cuts (intelligent paint)
  - Distance (Voronoi-like cells)
- Bounding box
  - Grabcut [Rother et al]
- Center seeds
  - Star shape [Veksler]
- Many other options…

- Normalized cuts [Shi Malik]
- Mean-shift [Comaniciu]
- MDL [Zhu&Yuille]
- Entropy of appearance
- Add enough constraints:
  - Saliency
  - Shape
  - Known appearance
  - Texture

Differences maybe minor

interactive ? unsupervised

$$E(S,\theta_0,\theta_1) = \sum_p -\ln \Pr(I_p \mid \theta_{S_p}) + \sum_{pq\in N} w_{pq}[S_p \neq S_q]$$

Grabcut energy [Rother et al]

$$E(S) = \underbrace{|S|\cdot H(S)}_{\text{entropy}} + \underbrace{|\bar{S}|\, H(\bar{S})}_{\text{entropy}} + \sum_{pq\in N} w_{pq}[S_p \neq S_q]$$

unsupervised image segmentation energy

NOTE: Grabcut converges to a local minima near the initial box.

Summary

Covered basics of:

- Thresholding, region growing
- Snakes, active contours
- Geodesic contours
- Graph cuts (binary labeling, MRF)

Implicit surface representation
Global optimization is possible

Not-Covered:

- Ratio functionals
- Normalized cuts
- Watersheds
- Random walker
- Many others…

To be covered later:

- High-order models
- Multi-label segmentation
- Model fitting