Algorithms for Image Analysis

Beyond Snakes:

Explicit vs. Implicit representation of contours

Level-sets

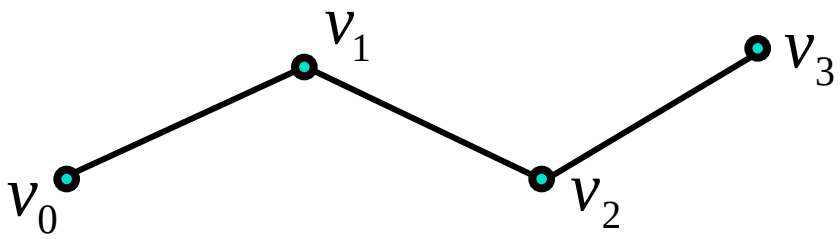
Geodesic Active Contours

Beyond the Snakes

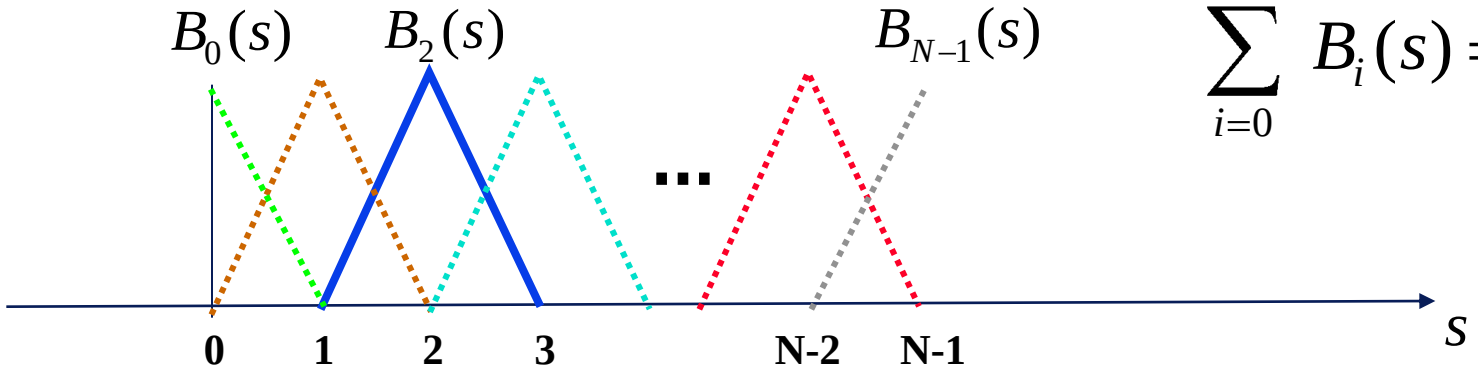
Examples of Representation Models for Contours and Surfaces

- Parametric (snakes)
  - Explicit (point-based)
  - Spline snakes
- Non-parametric (implicit models)
  - Level-sets (mainly for Geometric Active Contours)
  - Graph-based

Previous (polygonal) snakes can be seen as



$$C(s) = \sum_{i=0}^{N-1} v_i \cdot B_i(s) \qquad 0 \le s \le (N-1)$$

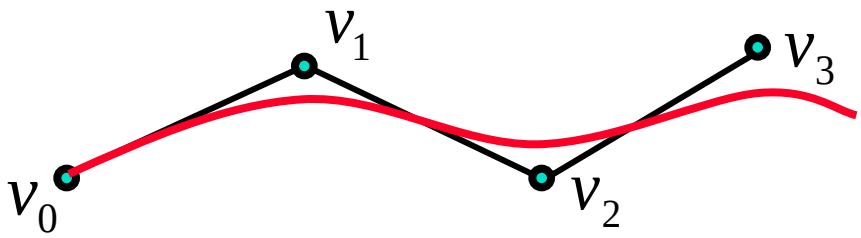


$$\sum_{i=0}^{N-1} B_i(s) = 1 \qquad \forall s$$

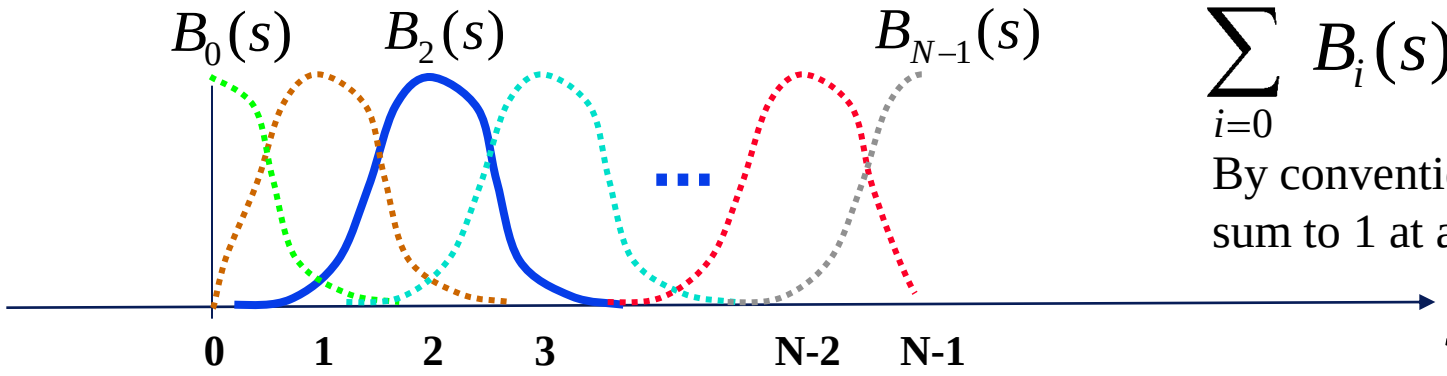
- Contour as a linear combination of N basis functions

Other parametric representation models for contours:

B-spline snakes (See A. Blake and M. Isard, "*Active Contours*")



$$C(s) = \sum_{i=0}^{N-1} v_i \cdot B_i(s) \qquad 0 \le s \le (N-1)$$



$$\sum_{i=0}^{N-1} B_i(s) = 1 \qquad \forall s$$

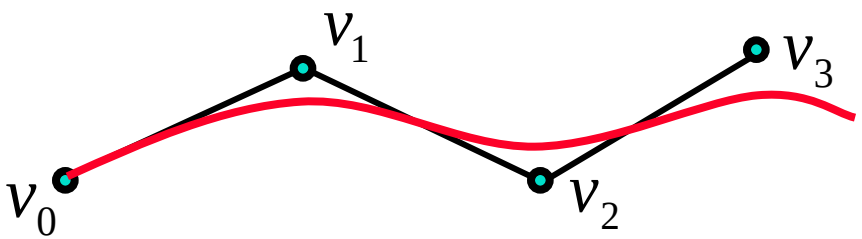
By convention, B-splines sum to 1 at all points s

- Each $B_i(s)$ is typically a simple combination of polynomials

Other parametric representation models for contours:

B-spline snakes (See A. Blake and M. Isard, "*Active Contours*")

Explicit representation of contour: $C(s) = \sum_{i=0}^{N-1} v_i \cdot B_i(s)$



Smooth (differentiable) curve passes close to the poly-line joining the knot (control) points

B-spline snakes - still parametric representation of continuous contours via finite number of parameters:

$$\{v_0, v_1, v_2, \ldots, v_{N-1}\}$$

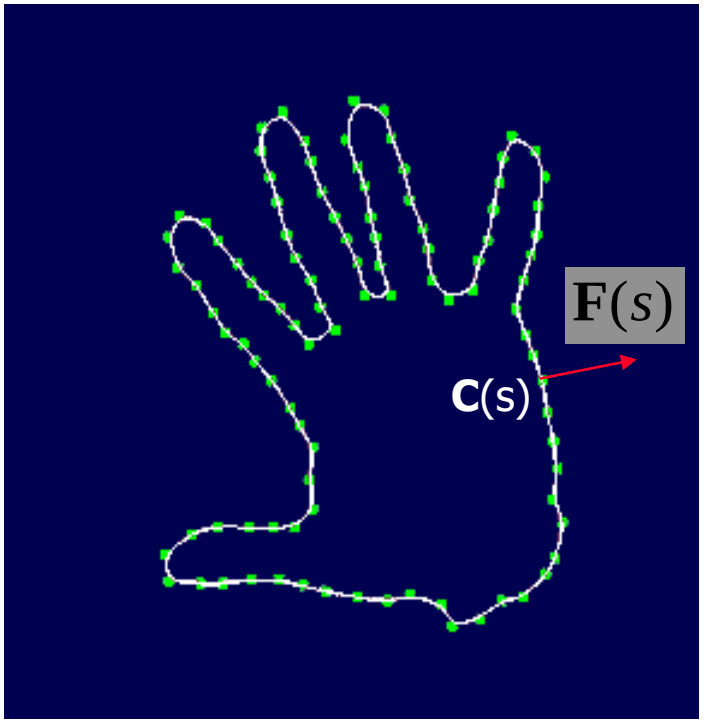
[a slide borrowed from Daniel Cremers]

Evolution of Explicit Boundaries

$$\mathbf{C}(s) = \sum_{j=0}^{N-1} \mathbf{v}_j \cdot B_j(s)$$

control points

basis functions



$$d\mathbf{C} = \sum_j d\mathbf{v}_j \cdot B_j(s) = \mathbf{F}(s) \cdot dt$$

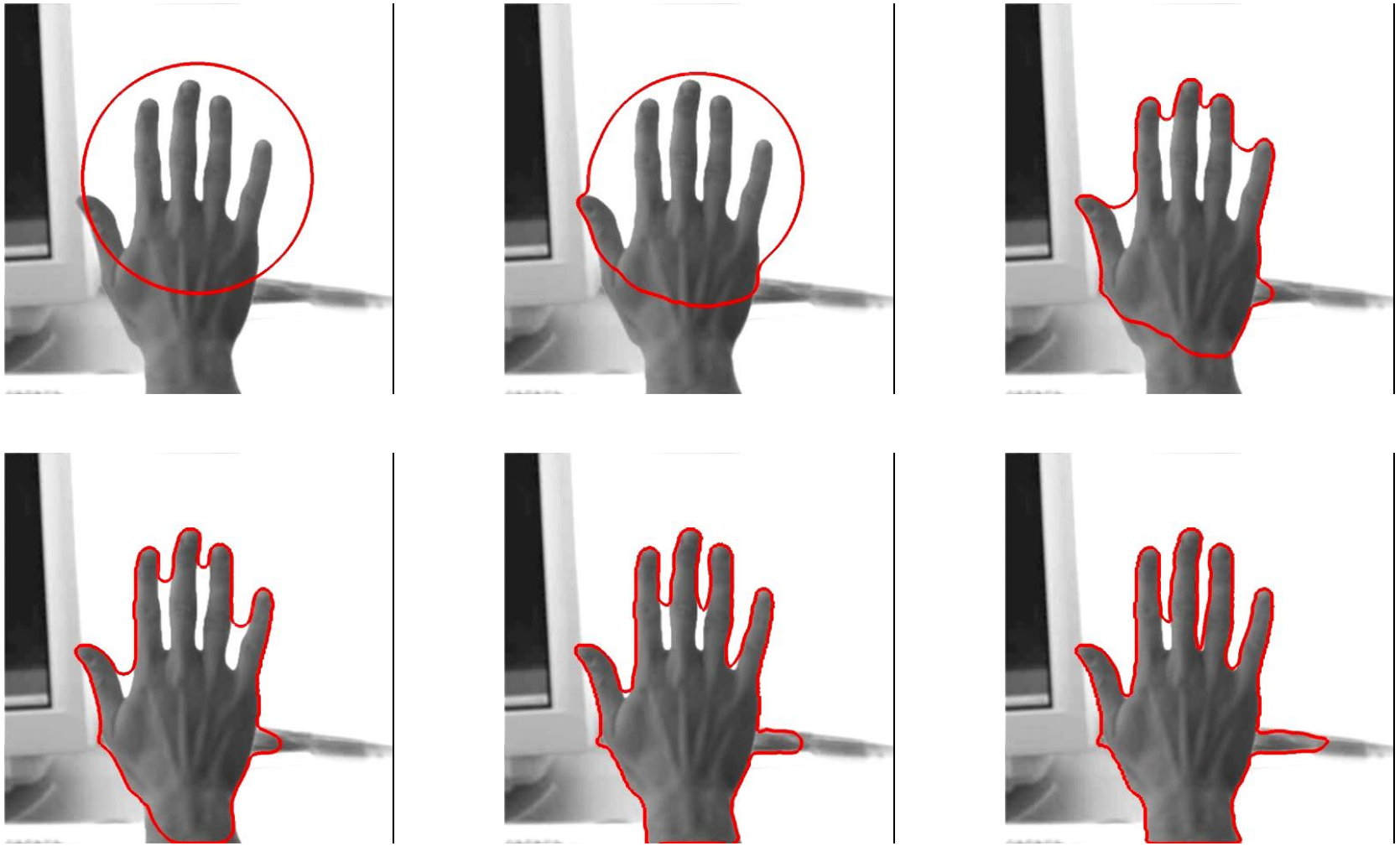
some given force

$$\sum_j d\mathbf{v}_j \cdot \underbrace{\langle B_i, B_j \rangle}_{B_{ij}} = \underbrace{\langle B_i, \mathbf{F} \rangle}_{\mathbf{b}_i} \cdot dt \quad \Rightarrow \quad \boxed{\vec{d\mathbf{v}} = B^{-1}\vec{\mathbf{b}} \cdot dt}$$

control point evolution

[a slide borrowed from Daniel Cremers]

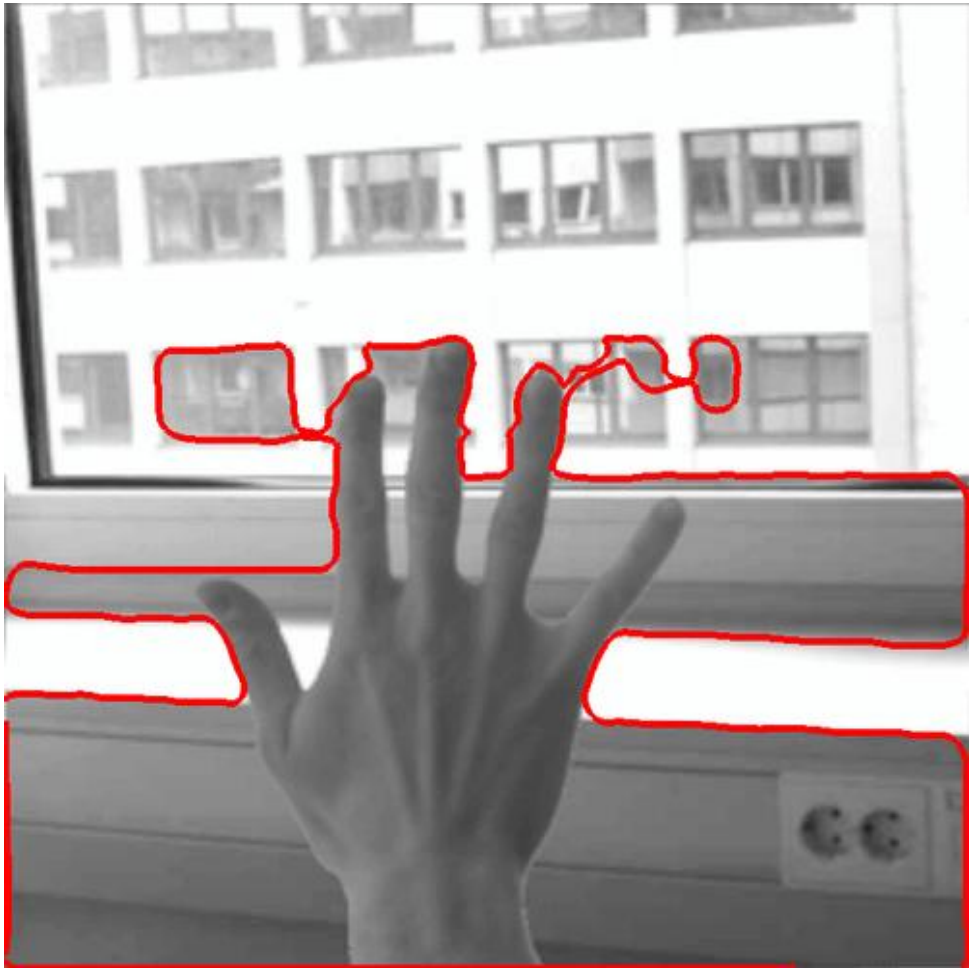
Evolution of Explicit Boundaries



Cremers, Tischhäuser, Weickert, Schnörr, "Diffusion Snakes", IJCV '02

[a slide borrowed from Daniel Cremers]

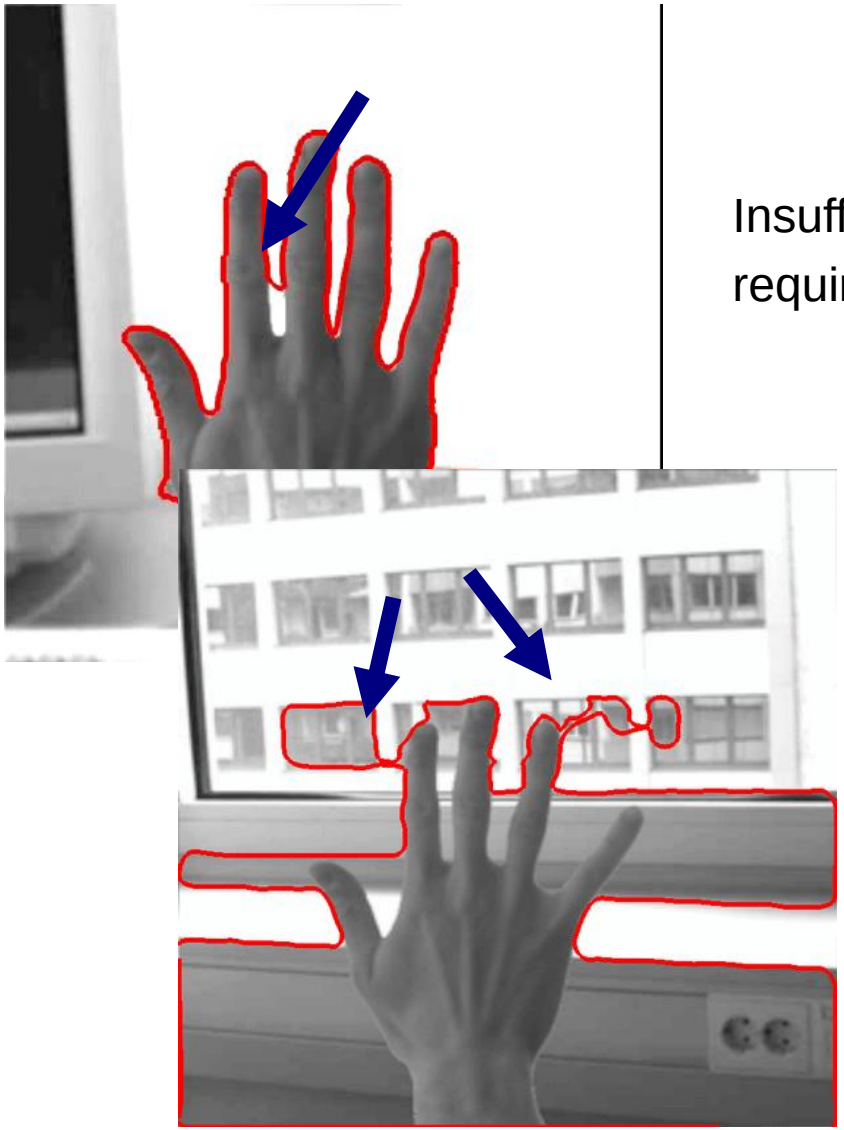
Evolution of Explicit Boundaries



Cremers, Tischhäuser, Weickert, Schnörr, "Diffusion Snakes", IJCV '02

[a slide borrowed from Daniel Cremers]

Limitations of Explicit Representations

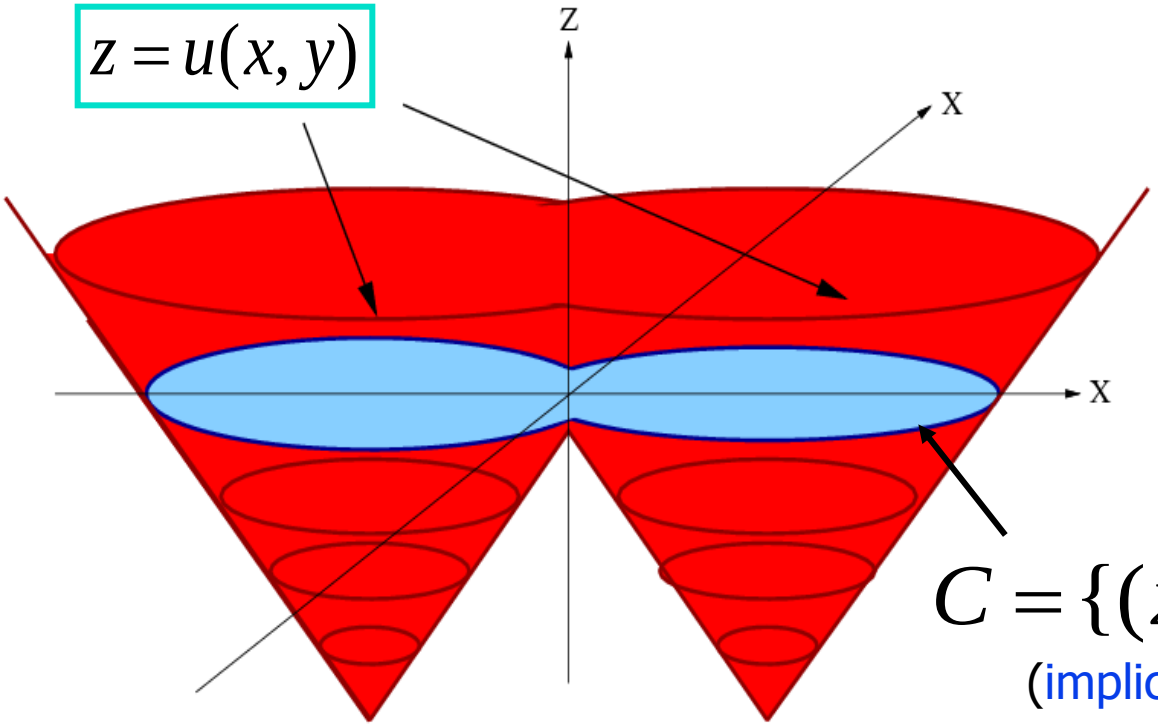


Insufficient resolution / control point density
requires control point regridding mechanisms

Fixed topology
requires heuristic splitting mechanisms

Non-parametric implicit representation of contours via *Level Sets*

The University of Western Ontario



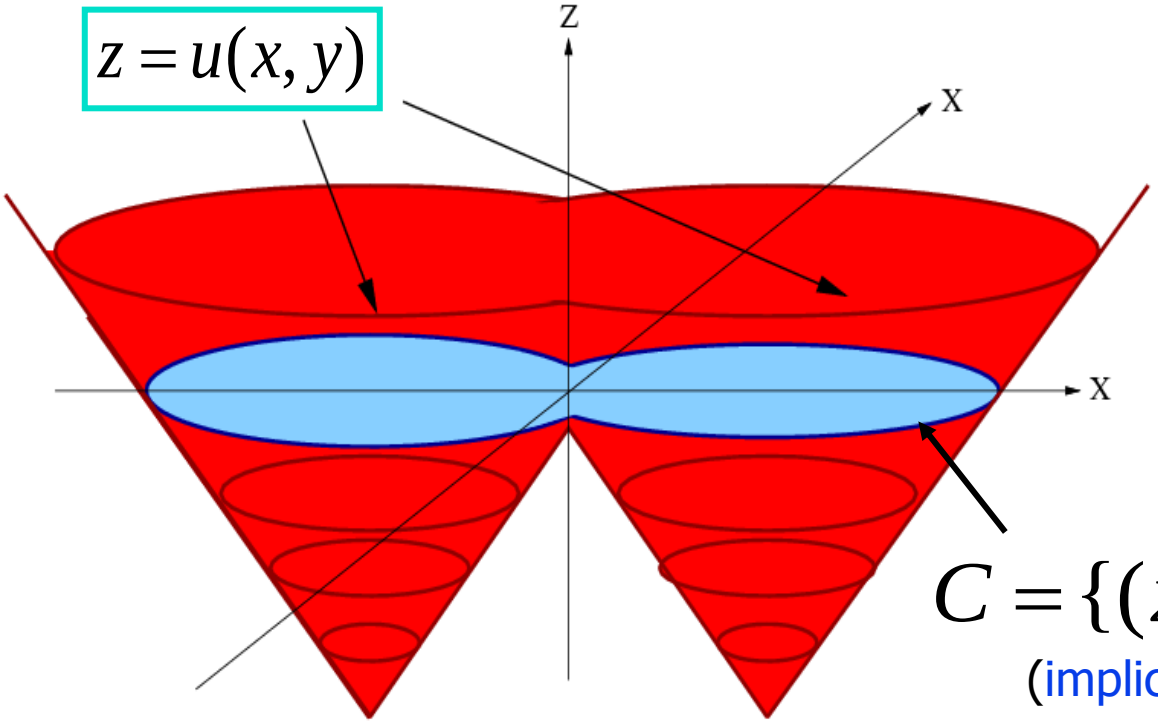
$$C = \{(x, y) : u(x, y) = 0\}$$

(implicit contour representation)

- Let contour *C* be a zero-level set of some function *u(x,y)*
- Function *u(x,y)* is called a *level-set* function

Non-parametric implicit representation of contours via *Level Sets*

The University of Western Ontario



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

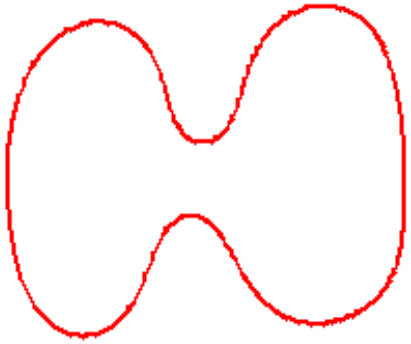
(implicit contour representation)

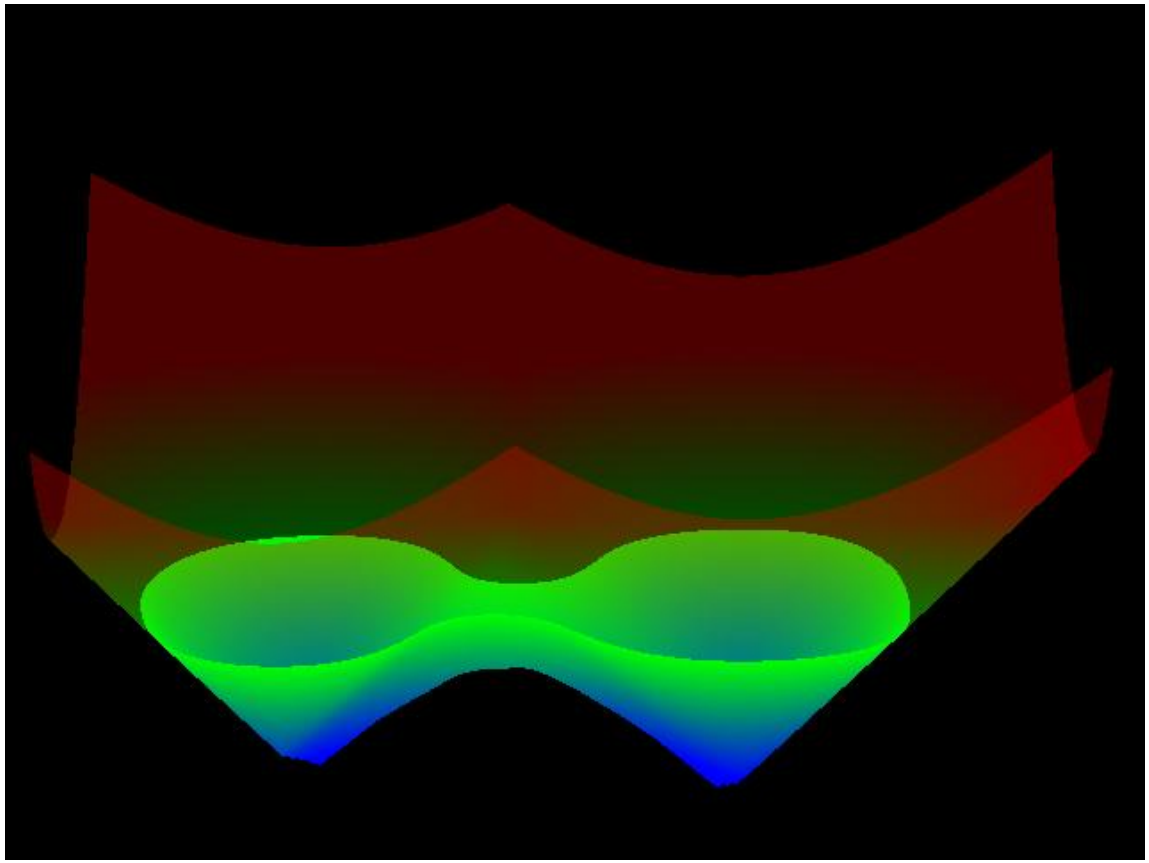
- For a given *u(x,y)* one can get *C* by thresholding
  Contour interior are points with negative values of *u(x,y)*
- How to get some level set function *u(x,y)* for a given *C*?

  Signed distance map: $u(x, y) = \pm \text{dist}(x, y, C)$

[Visualization is courtesy of O. Juan]

Example of distance map



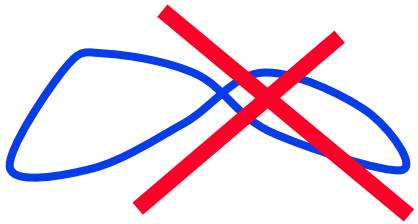


Advantages of *level-set* representation of contours

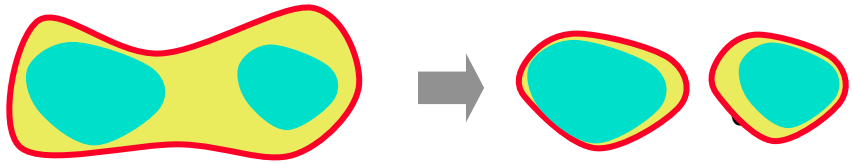
The University of Western Ontario

- Level set functions can represent contours corresponding to object boundaries with arbitrary topological properties (holes, isolated parts)

- No self intersections

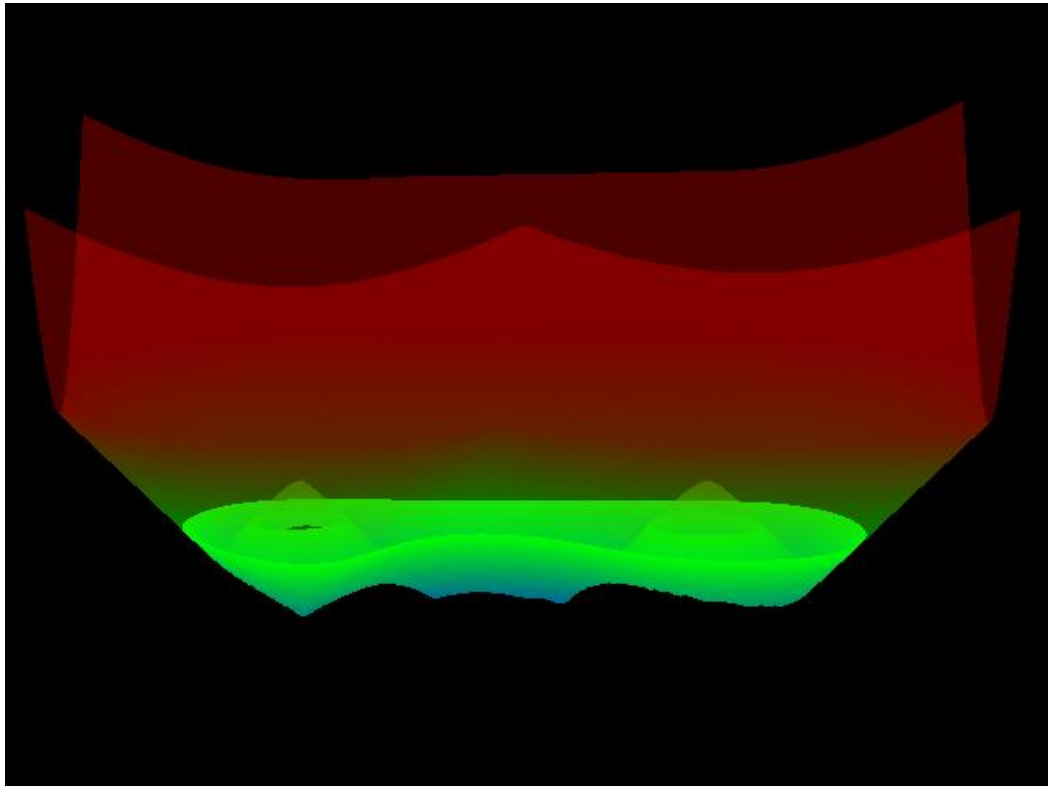


- Can change topology as it evolves



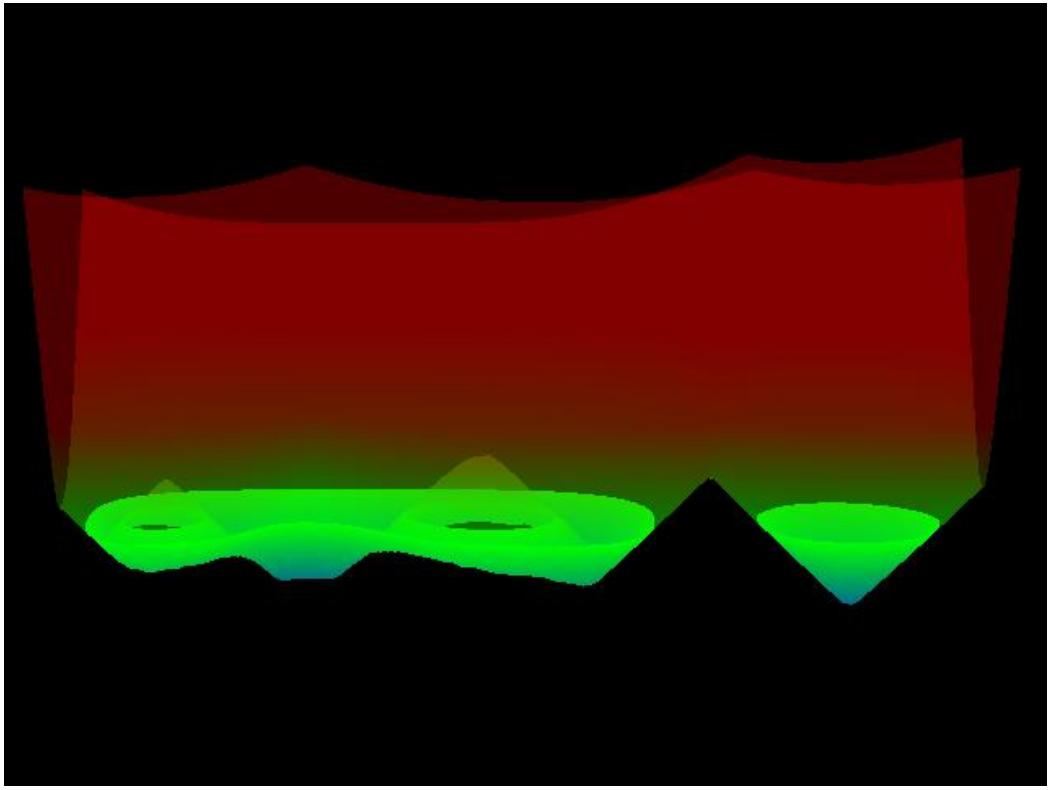
[Visualization is courtesy of O. Juan]

Some more complex examples



[Visualization is courtesy of O. Juan]

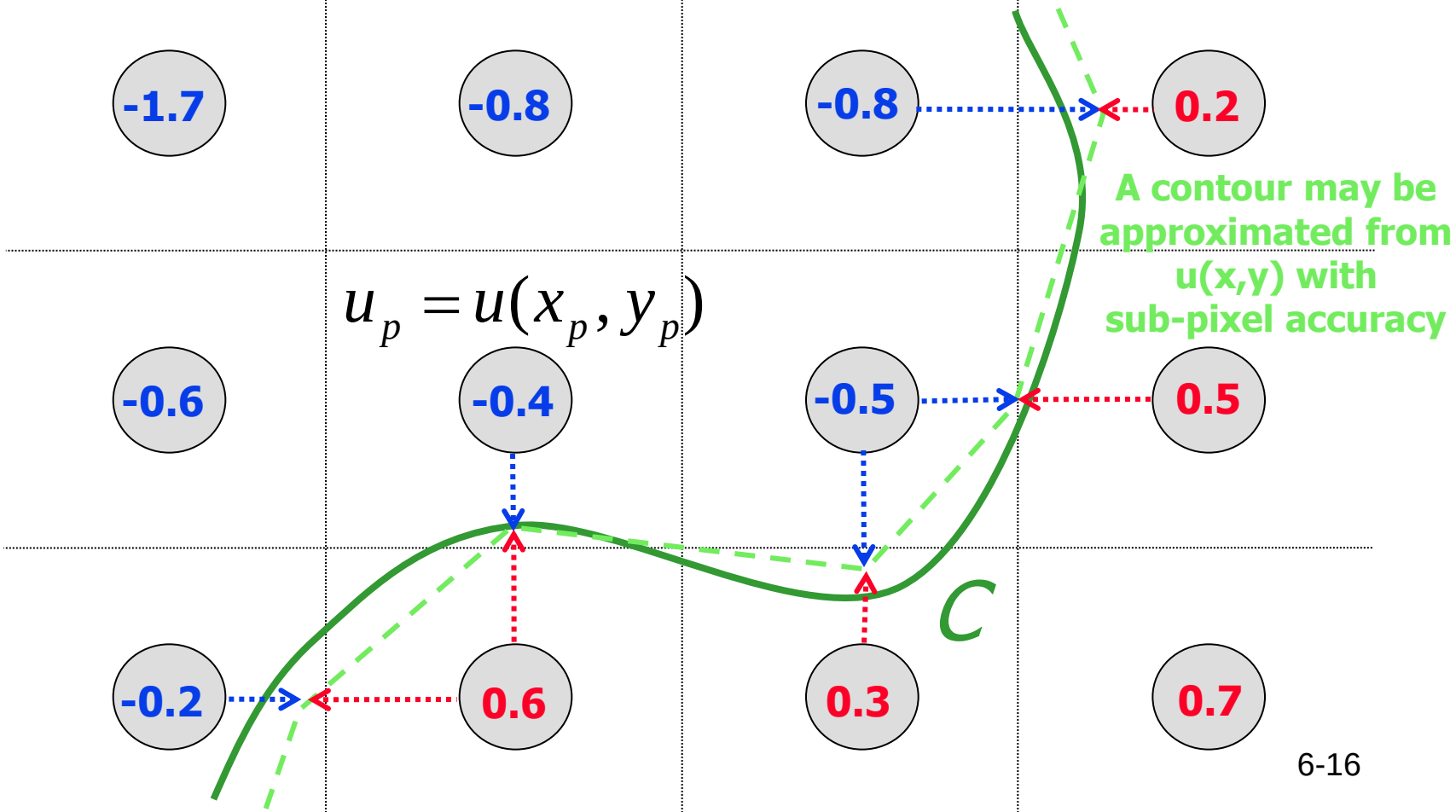
Some more complex examples



Level-sets, how does that work?

- easier then you may think

- Level set function *u(x,y)* is normally discretized/stored over image pixels
- Values of *u(p)* can be interpreted as distances or heights of image pixels



Dervieux, Thomasset, ’79, ’81, Osher and Sethian, 89

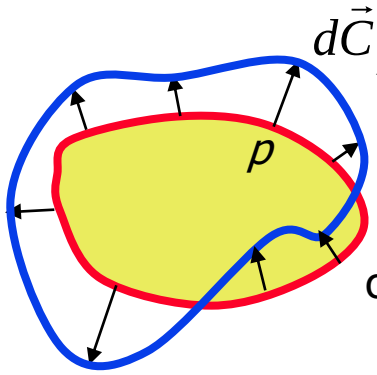
Contour evolution via *Level Sets*

Consider some motion or evolution of contour C

$$d\vec{C}_p = \beta_p \cdot \vec{N}_p$$

motion of contour points

contour normals



$$\| d\vec{C}_p \| = | \beta_p |$$

speed

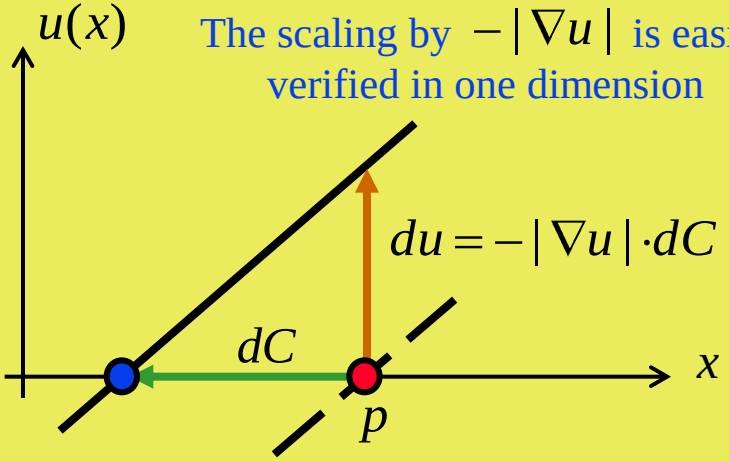
Note: contour’s motion can be detected visually only in the direction orthogonal to the contour (i.e. $\vec{N}$). **Tangential motion is geometrically irrelevant.**

FACT: contour evolution above can be implicitly replicated by updating level-set function *u(x,y)* as follows:

$$du_p = -\beta_p \cdot | \nabla u_p |$$

change of “pixel’s height”

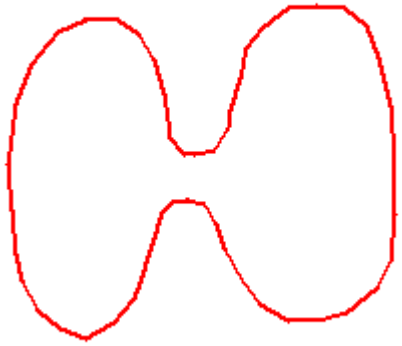
gradient of u(x,y)

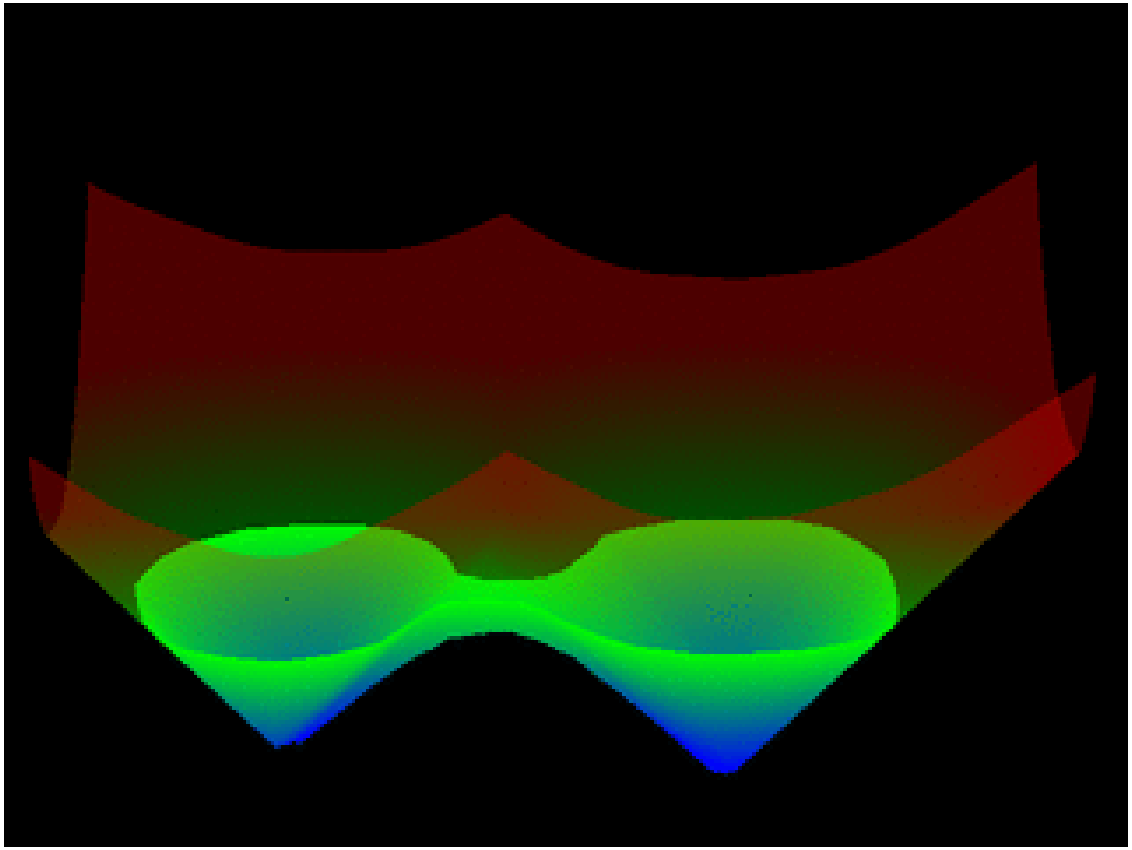


[Visualization is courtesy of O. Juan]

Simple evolution

$$d\vec{C}_p = -\vec{N}_p \quad \Longleftrightarrow \quad du_p = |\nabla u_p|$$





Where could *contour speed* β come from in image segmentation?

The University of Western Ontario

$$dC = \beta \cdot \vec{N}$$

this geometric contour motion formula can be seen as a mathematical way of describing generic form of *region growing*

- Empirical approach: chose β based on some image heuristics
  - BOUNDARY BIAS: contour speed β could be large far from object boundaries and approach zero near high image gradients in the image
  - REGIONAL BIAS: contour speed β at pixel p could measure how much contour C likes ($\beta>0$) or dislikes ($\beta<0$) to have p in its interior

BAD IDEA: empirical choices of β are often equivalent to "region growing" or "thresholding", ... **examples?**

- no energy formulation => no quality or convergence guarantees

BETTER APPROACH: formulate an energy functional and optimize it

- generally much safer way to robust and numerically stable algorithms

May contour motion correspond to some energy minimization?

The University of Western Ontario

visible motion

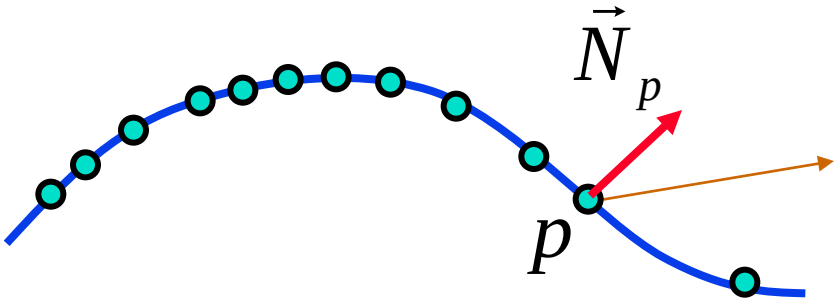
$$dC = \beta \cdot \vec{N}$$

?

gradient descent w.r.t. energy *E*

$$dC = -dt \cdot \nabla E$$

(as in slide 5-42)



$$\nabla E_p = \begin{Bmatrix} \frac{dE}{dx_p} \\ \frac{dE}{dy_p} \end{Bmatrix}$$

POTENTIAL PROBLEMS:

- Energy gradient ∇E may not be orthogonal to contour creating invisible geometrically-irrelevant motion,… **example?**
- Contour parameterization (e.g. control points selection) may affect gradient ∇E and, therefore, visible contour motion

What if energy E evaluates only appearance of contour C ?

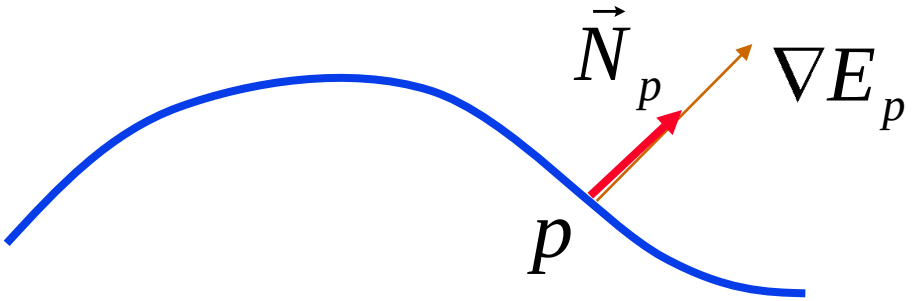
The University of Western Ontario

visible motion

$$dC = \beta \cdot \vec{N}$$

gradient descent w.r.t. energy *E*

$$dC = -dt \cdot \nabla E$$

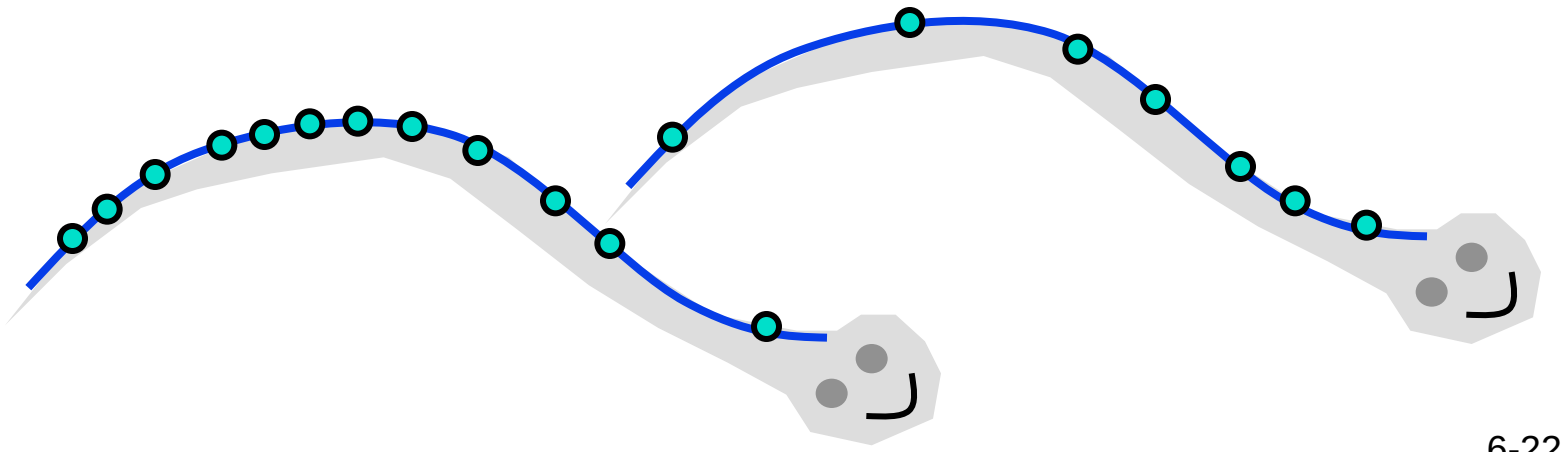


- Energy $E(C)$ is *geometric* if it evaluates only geometric properties of contour C which are visually observable and independent of contour representation or parameterization choice
  - then gradient descent can generate only visible motion (**Why?**)
  - **implicit** representation of contours via **level-sets** fits well (**Why?**)

MAIN MOTIVATION FOR USING GEOMETRIC ENERGIES IN IMAGE ANALYSIS:

appearance/geometry of contour *C* defines segmentation and that is all that really matters in the context of image analysis

invisible parameters of contours are irrelevant, in the end.



Bringing geometry into snakes:
first proposed by Caselles et al. in 1993-95

Elastic Snake
(Kass, Witkin, Terzopoulos, 87)
motivated by physics

energy of elastic band with external force

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

for uniform parameterization $C:[0,1] \rightarrow R^2$

- Reparameterization $s = f(s')$ changes the energy form and affects contours behavior
  - like choosing different irregularly spaced control points for a discrete snake

Geodesic Active Contours
(Caselles, Kimmel, Shapiro, 95)
motivated by geometry

Riemannian length of *C*

$$E = \int_0^L g(C(s))ds = \int_C g_p \, ds$$

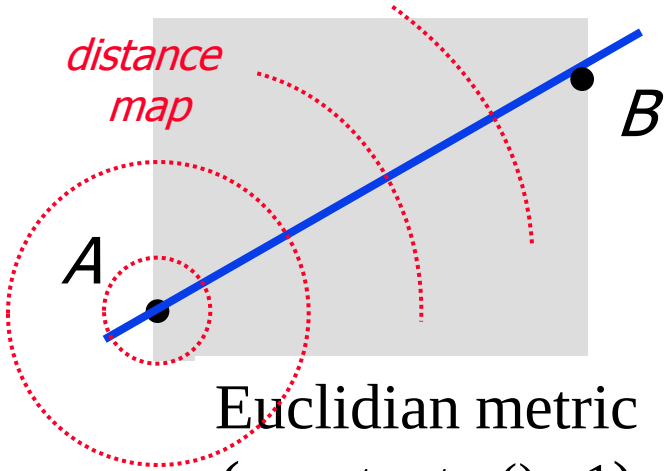
for arc-length parameterization $C:[0,L] \rightarrow R^2$

- if $g = 1$ then *E(C)* is simply Euclidean length of contour C
- if function $g \geq 0$ weights each ds interval => "weighted" (Riemannian) length of C

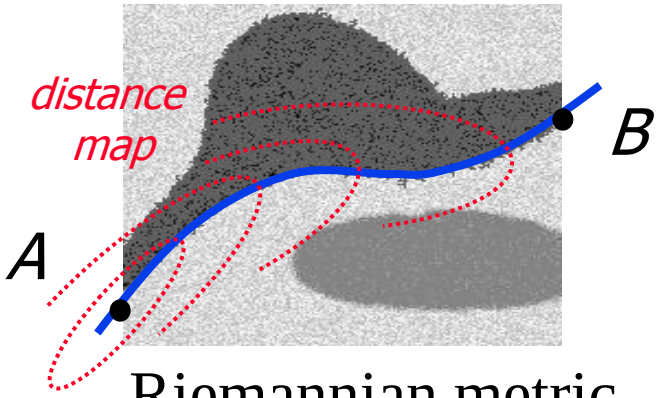
Caselles, Kimmel, Shapiro, 95

Geodesic Active Contours

- Energy $E(C) = \int_A^B g(|\nabla I_{C(s)}|)ds$ = "weighted length" of C
- *Geodesic:* Shortest curve between two points (wrt. E).



Euclidian metric
(constant $g() = 1$)



Riemannian metric
(space varying $g(|\nabla I|) \geq 0$)

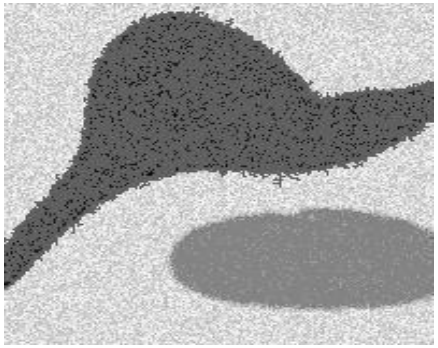
E(C) = Euclidean length of *C*

E(C) = image-weighted length of *C*
(very similar to live-wire)

Caselles et al.'93, Caselles et al.'95, Kichenassamy et al. '95

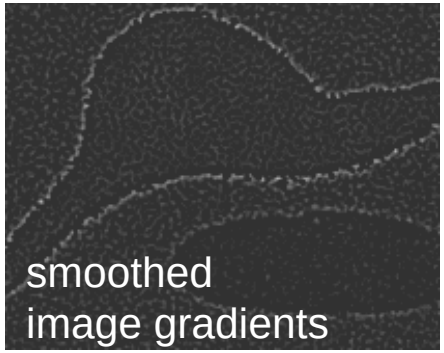
Geodesic Active Contours

Image I



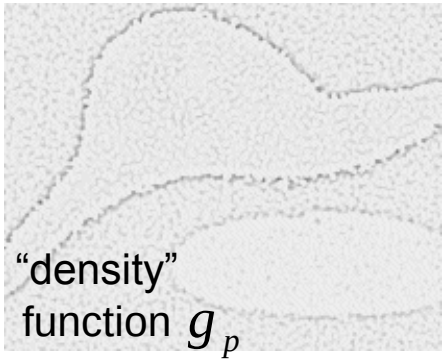
$\Rightarrow$

$\nabla I_\sigma = \nabla * H_\sigma * I$



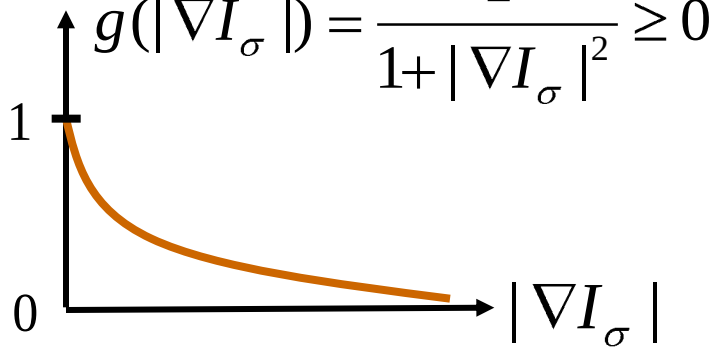
$\Rightarrow$

$g_p = g(|\nabla I_\sigma(p)|)$



Function $g()$ transforms image gradients into density function over image pixels g_p. Scalar function g_p defines non-Euclidean metric for measuring "length" of any curve or path C in the image $\|C\|_g = \int_C g_p \, ds$.

$$g(|\nabla I_\sigma|) = \frac{1}{1+|\nabla I_\sigma|^2} \geq 0$$

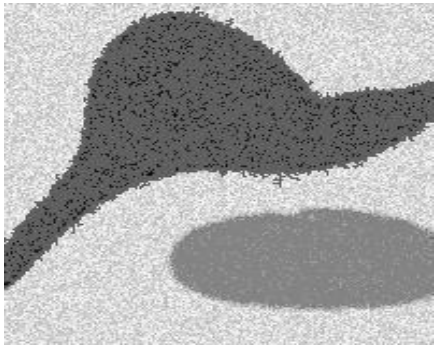


NOTE: *live-wire* uses a similar approach by defining discrete image "metric" via graph edge-weights $\|C\|_w = \sum_{e \in C} w_e$

Caselles et al.'93, Caselles et al.'95, Kichenassamy et al. '95

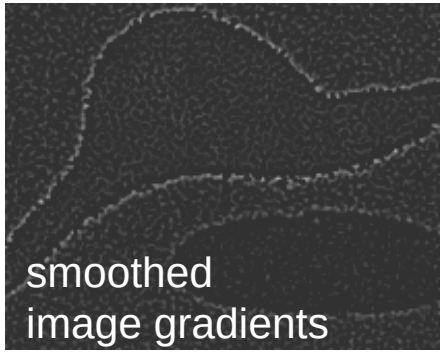
Geodesic Active Contours

Image I



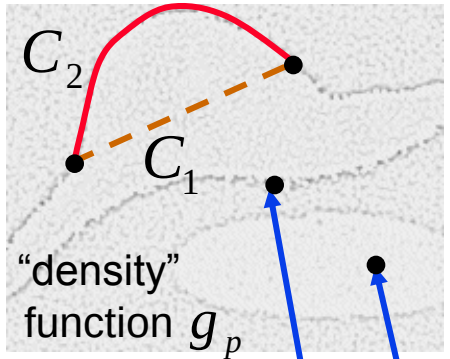
$\Rightarrow$

$\nabla I_\sigma = \nabla * H_\sigma * I$



$\Rightarrow$

$g_p = g(|\nabla I_\sigma(p)|)$

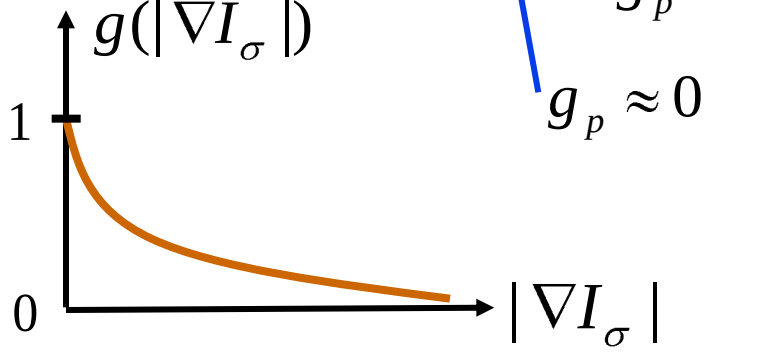


$$\|C\|_g = \int_C g_p \cdot ds$$

For strong enough image edges
and for appropriate *g we have*

$$E(C_2) \;?\; E(C_1)$$

$$\underbrace{\int_{C_2} (\approx 0) \cdot ds}_{} \;<<\; \underbrace{\int_{C_1} (1) \cdot ds}_{}$$



Other examples of *g()* ?

...restriction: $g(\cdot) \geq 0 \wedge g'(\cdot) \leq 0$

Caselles et al.'93, Caselles et al.'95, Kichenassamy et al. '95

Implementing Geodesic Active Contours

$$\| C \|_g = \int_C g_p \cdot ds$$

gradient descent

$$dC_p = -dt \cdot \nabla E_p$$

FACT:
for any g>=0

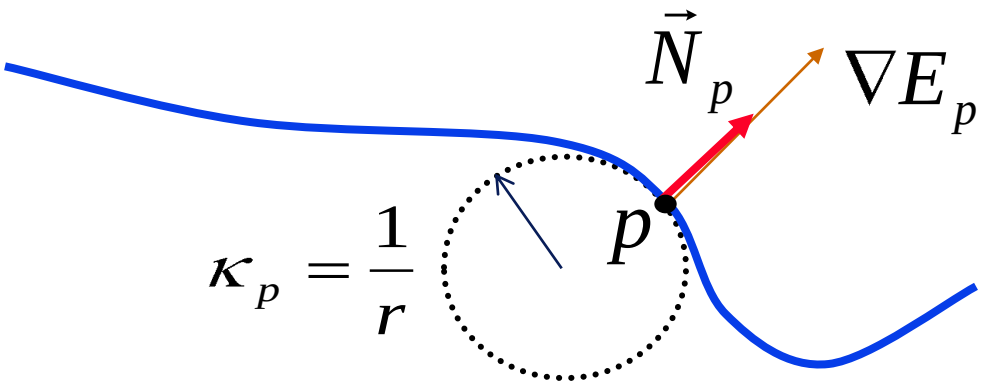
dot product

"closed" formula

$$\nabla E_p = (-g_p \cdot \kappa_p + \langle \nabla g_p, \vec{N}_p \rangle) \cdot \vec{N}_p$$

curvature of contour C

normal of contour C



NO tangential (geometrically invisible) motion

$$dC_p \sim \nabla E_p \sim \vec{N}_p$$

Caselles et al.'93, Caselles et al.'95, Kichenassamy et al. '95

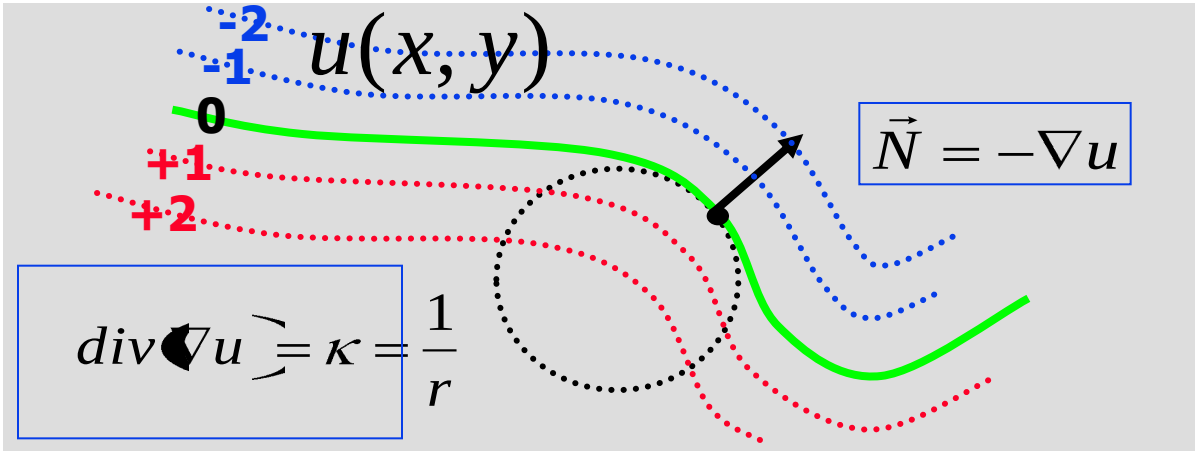
Implementing Geodesic Active Contours

- Gradient Descent $dC = -dt \cdot \nabla E = dt \cdot (g \cdot \kappa - \langle \nabla g, \vec{N} \rangle) \cdot \vec{N}$

$$\Rightarrow \quad dC = \beta \cdot \vec{N}$$
$$\Rightarrow \quad du = -\beta$$

via level-sets

SOME USEFUL FACTS



$$\Rightarrow \quad du = -dt \cdot \left(g \cdot div\left(\frac{\nabla u}{|\nabla u|}\right) + \langle \nabla g, \nabla u \rangle\right)$$

Caselles et al.'93, Caselles et al.'95, Kichenassamy et al. '95

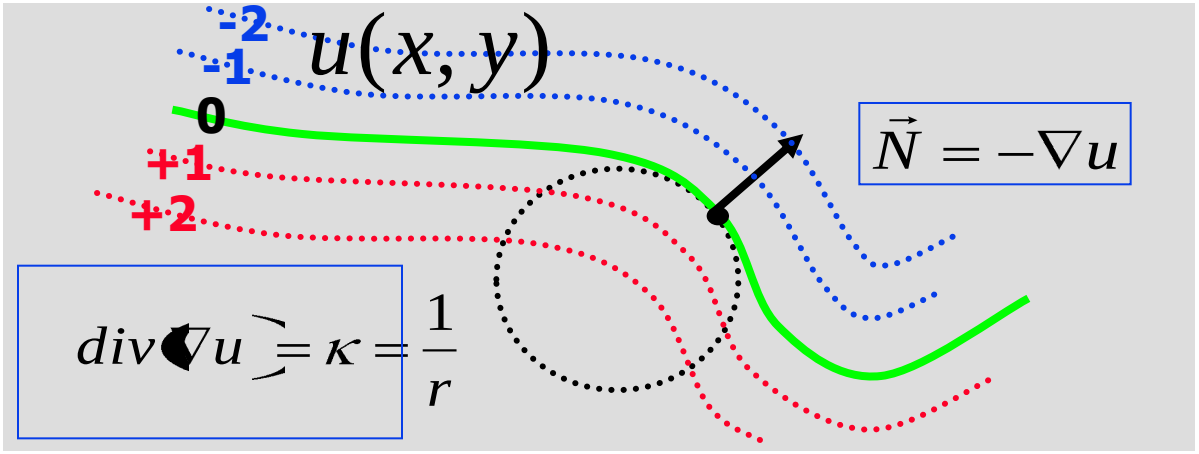
Implementing Geodesic Active Contours

- Gradient Descent $dC = -dt \cdot \nabla E = dt \cdot (g \cdot \kappa - \langle \nabla g, \vec{N} \rangle) \cdot \vec{N}$

$$\Rightarrow \quad dC = \beta \cdot \vec{N}$$
$$\Rightarrow \quad du = -\beta$$

via level-sets

SOME USEFUL FACTS



$$\Rightarrow \quad du = -dt \cdot div\left(g \cdot \frac{\nabla u}{|\nabla u|}\right)$$

Caselles et al.'93, Caselles et al.'95, Kichenassamy et al. '95

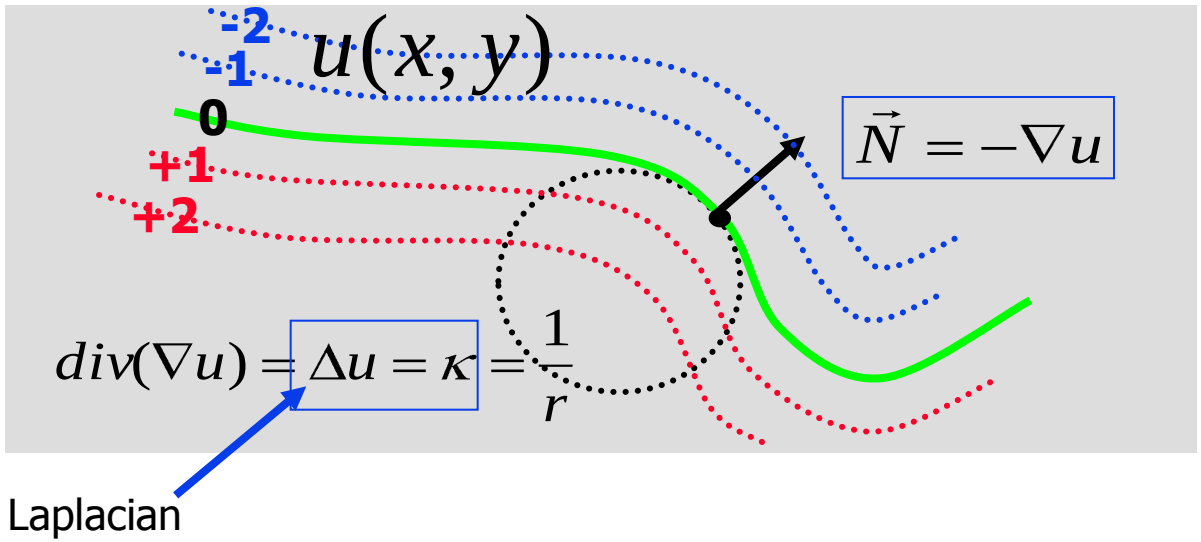
Implementing Geodesic Active Contours

- Gradient Descent $dC = -dt \cdot \nabla E = dt \cdot (g \cdot \kappa - \langle \nabla g, \vec{N} \rangle) \cdot \vec{N}$

$$\Rightarrow \quad dC = \beta \cdot \vec{N}$$
$$\Rightarrow \quad du = -\beta$$

via level-sets

SOME USEFUL FACTS



if *u(x,y)* is a signed distance map, then

$$|\nabla u| = 1$$

$$\Rightarrow \quad du = -dt \cdot div(g \cdot \nabla u)$$

[example from Goldenberg, Kimmel, Rivlin, Rudzsky, IEEE TIP '01]

Geodesic Active Contours via Level-sets

$$u_p \mathrel{+}= (-dt) \cdot \left(\frac{\partial (g \cdot \nabla_x u)}{\partial x} + \frac{\partial (g \cdot \nabla_y u)}{\partial y} \right)$$

Simple Example:
Mean Curvature evolution

The University of Western Ontario

If no image gradients, then $g = 1$

$$\| C \|_g = \int_C g_p \cdot ds = \int_C 1 \cdot ds = \| C \|_\varepsilon$$

Our contour energy is its *Euclidean* length

- Gradient Descent: $g = 1 \quad \Rightarrow \quad \nabla g = 0$

$$\Rightarrow \quad dC = -dt \cdot \nabla E = dt \cdot \kappa \cdot \vec{N}$$

via level-sets $\quad u_p \; += \; (-dt) \cdot \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right)$

Laplacian Δu

Simple Example: *Mean Curvature* evolution

The University of Western Ontario

Gradient descent of contour w.r.t. *Euclidean* length

$$u_p \mathrel{+}= (-dt) \cdot \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right)$$

Laplacian

Simple Example: *Mean Curvature* evolution

The University of Western Ontario

Gradient descent of contour w.r.t. *Euclidean* length

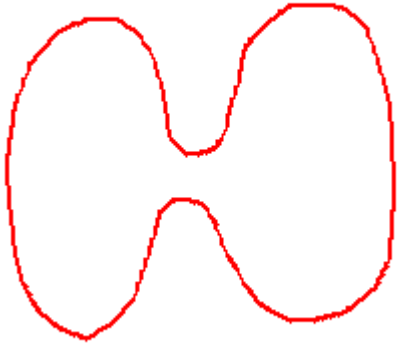
$$u_p \mathrel{+}= (-dt) \cdot \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right)$$

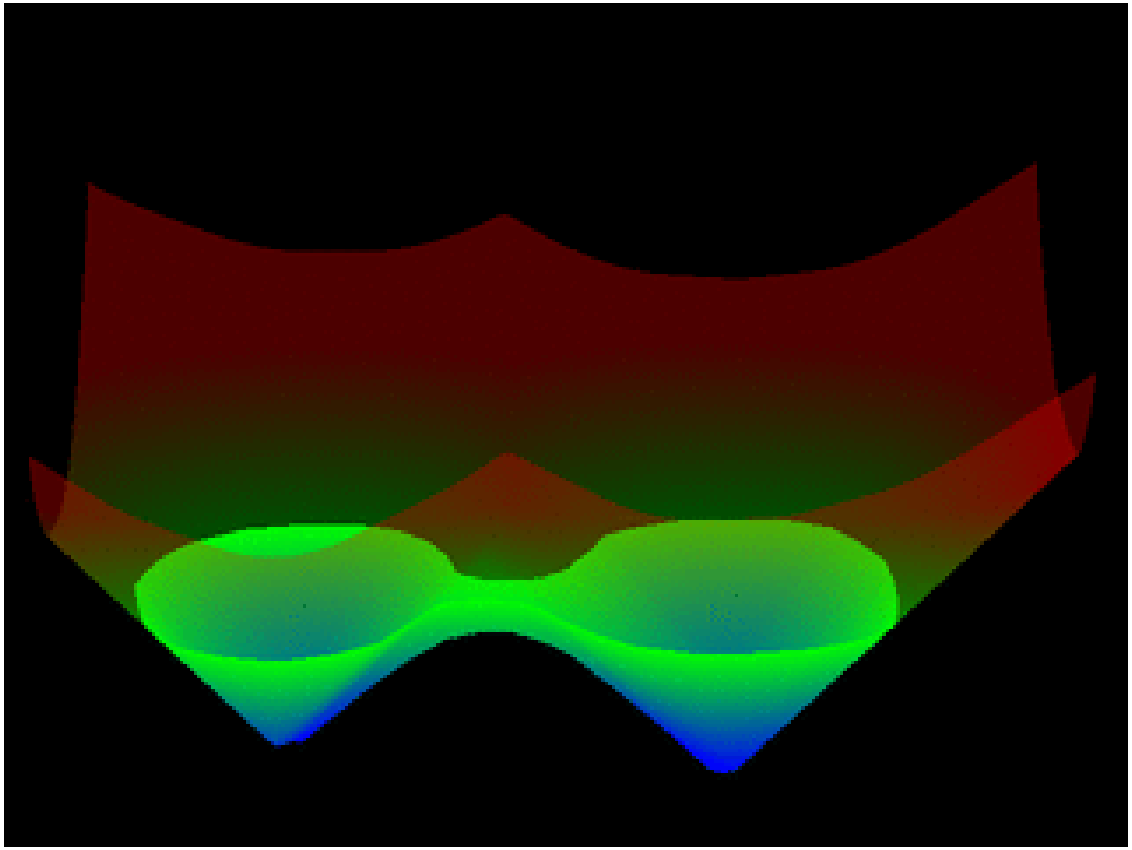
Laplacian

[Visualization is courtesy of O. Juan]

More examples of Mean Curvature Motion

$$d\vec{C} = -\kappa\vec{N}$$

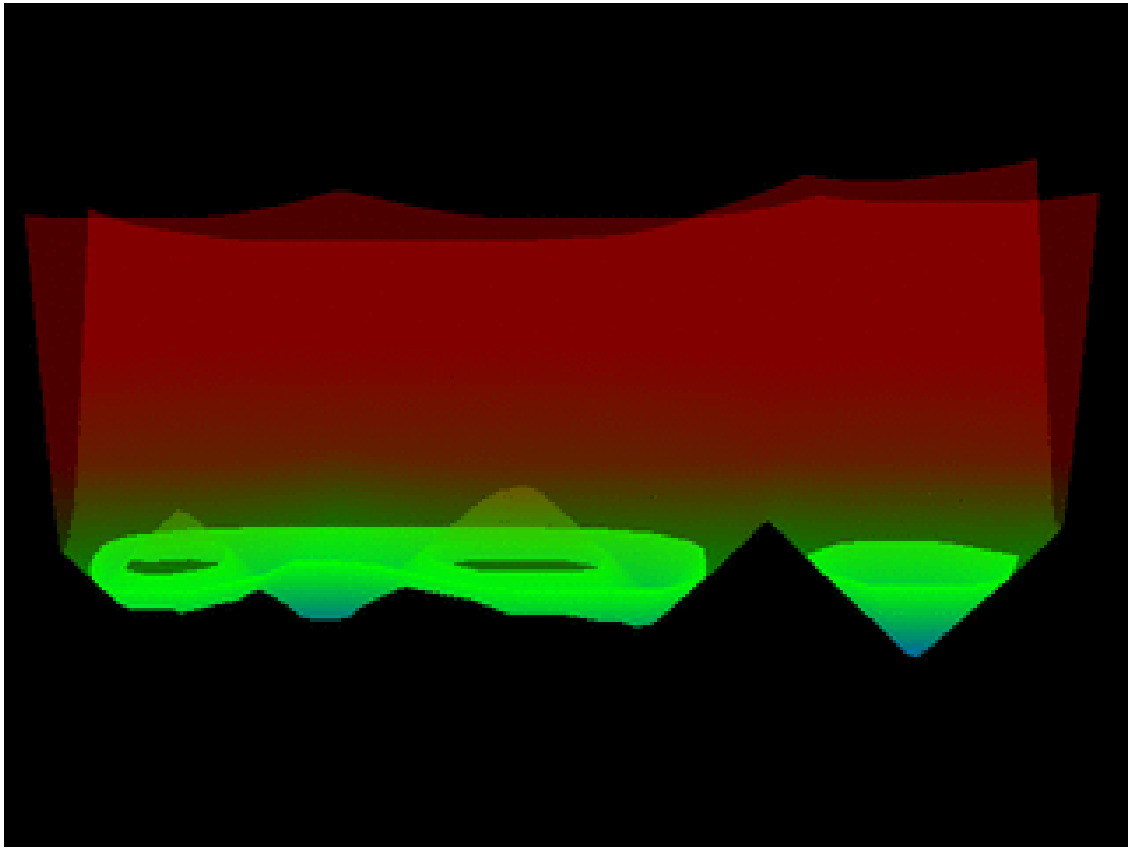




[Visualization is courtesy of O. Juan]

More examples of Mean Curvature Motion

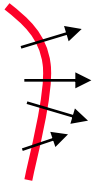
$$d\vec{C} = -\kappa\vec{N}$$



Other geometric energy functionals besides *length* [courtesy of Ron Kimmel]

The University of Western Ontario

Geometric measures commonly used in segmentation

	Functional *E(C)*	**gradient descent evolution** $dC = \beta \cdot \vec{N}$
 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)	$E(C) = \int_C \langle \vec{\mathbf{v}}, \vec{N} \rangle\ ds$	$\beta \sim \mathrm{div}(\vec{\mathbf{v}})$

Optimizing Geometric Functionals $E(C)$

- Differential approach:
  - Gradient descent
    - Explicit $dC = -dt \cdot \nabla E = \beta \cdot \vec{N}$
    - Implicit (level-sets) $du = -\beta$
  - **Local optimization**
- Integral approach:
  - DP (2D), remember *livewire*
  - **Graph Cuts** (N-D) – starting next lecture
  - **Global optimization** **(combinatorial graph algorithms)**

Continuous Global Optimization

- TV approach for image segmentation:

$$E(u) = \iint f_{ob} \cdot u + f_{bg} \cdot (1-u) + |\nabla u| dx_1 dx_2$$

with $u(x) \in \{0,1\}$ cannot be solved directly.

- E(u) is convex if we allow $u(x) \in [0,1]$
- The globally optimal $u^*(x)$ may have values $\notin \{0,1\}$
- **Threshold Theorem:** (Chan, Esedoglu, Nikolova, SIAM 2006)

$$u_\Theta(x) = \begin{cases} 0 & , \text{if } u^*(x) < \Theta \\ 1 & , \text{if } u^*(x) \geq \Theta \end{cases}$$

is also globally optimal for any $\Theta \in (0,1)$