ECCV 2006 tutorial on

Graph Cuts vs. Level Sets

part III

Connecting Graph Cuts and Level Sets

Yuri Boykov  Daniel Cremers  Vladimir Kolmogorov
University of Western Ontario  University of Bonn  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
- Integral and differential approaches
- Learning and shape prior in graph cuts and level-sets
Connecting graph cuts and level sets

- Integral and differential approaches
  - Integral vs. differential geometry
    - Implicit surface representation via level sets and graph cuts
    - Sub-pixel accuracy vs. non-deterministic surface
  - Differential and integral solutions for surface evolution PDEs
    - Gradient flow as a sequence of optimal small step
    - L2 distance between contours/surfaces
    - PDE-cuts (pluses and minuses)
    - Spatio-temporal approach
    - Shortcomings of narrow band cuts and DP snakes
Integral and differential approaches:

Implicit (region-based) surface representation

- 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

$$u_p = u(x_p, y_p)$$

A contour may be approximated from $u(x,y)$ with sub-pixel accuracy
Integral and differential approaches:
Implicit (region-based) surface representation

- Graph cuts represent surfaces via binary function $c(p)$ on image pixels
- Two values of $c(p)$ indicate interior and exterior labeling of pixel centers

Question: Is this a contour to be reconstructed from binary labeling $c(x,y)$? 
Answer: NO
Integral and differential approaches:

Implicit (region-based) surface representation

- Both level-sets and graph cuts use region-based implicit representation of contours

- Level-set function $u(p)$ allows to approximately reconstruct a contour with *sub-pixel accuracy*

- Graph cuts use a “non-deterministic” representation of contours. No particular contour satisfying given pixel labeling is fixed
Integral and differential approaches:

Sub-pixel accuracy

- Level-set function $u(p)$ allows to approximately restore a contour
  - with “sub-pixel accuracy”

- Graph cuts do not identify any particular contour among those that satisfy the pixel labeling
  - no “sub-pixel accuracy”
Integral and differential approaches:

Sub-pixel accuracy,… what for?

- “Super Resolution”
  - … if original data does not have sufficient resolution.

- In any case, one can use a regular grid of acceptable resolution which can be either finer or courser than the data.

- Now-days images often have fairly high resolution and pixel-size segmentation accuracy is more than enough for many applications.
Integral and differential approaches:

Sub-pixel accuracy,... who cares, who does not, and why?

- **Level-sets need sub-pixel accuracy for a technical reason:**
  - Explicit estimation of contour derivatives (e.g. curvature) is an intrinsic part of variational optimization techniques of differential geometry.

  e.g. curvature flow equation

  \[ C_t = \kappa \cdot \tilde{N} \]

  \[ \frac{\partial u}{\partial t} = \kappa \cdot |\nabla u| \]

  - explicit (*snakes*)
  - implicit (*level-sets*)

- **Graph cuts methods DO NOT use any surface derivatives in their inner workings**
  - **sub-pixel accuracy is unnecessary for graph cuts to work**
Integral and differential approaches:

Contour length in differential geometry?

\[ C(t) : [0,1] \rightarrow \mathbb{R}^2 \]

\[ \| C \|_\varepsilon = \sup \left\{ \sum_{i=1}^{n} \| C(t_i) - C(t_{i-1}) \|_\varepsilon : n > 0, \ 0 \leq t_0 \leq t_1 \leq \ldots \leq t_n \leq 1 \right\} \]

- Limit of finite differences approximation
Integral and differential approaches:

Contour length in differential geometry?

\[
C(t) : [0,1] \rightarrow \mathbb{R}^2
\]

If \( C'_{t_0} = \lim_{t \to t_0} \frac{\| C(t) - C(t_0) \|_\varepsilon}{| t - t_0 |} \) then

\[
\| C \|_\varepsilon = \int_0^1 C'_t \cdot dt
\]

- This is standard **Differential Geometry** approach to length
- Variational optimization gives standard *mean curvature flow*

\[
\frac{dC}{dt} = \kappa \cdot \vec{N} \quad \Rightarrow \quad \frac{du}{dt} = \kappa \cdot | \nabla u |
\]

as in level-sets
Integral and differential approaches:

How do graph cuts evaluate contour length?

- As mentioned earlier, the cost of a cut 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 and differential approaches:

**Integral geometry approach to length**

Euclidean length of $C$:

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

Cauchy-Crofton formula

The number of times line $L$ intersects $C$
Integral and differential approaches:

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
\{\_, \_/\, \|\, \_\}

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

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
Differential vs. integral approach to length

Differential geometry

\[ \| C \|_{\epsilon} = \int_{0}^{1} C'_t \cdot dt \]
\[ \| C \|_{\epsilon} = \int_{\Omega} |\nabla u| \, dx \]

Integral geometry

\[ \| C \|_{\epsilon} = \frac{1}{2} \int n_L \cdot d\rho \cdot d\phi \]

Parametric contour representation

Level-set function representation

Cauchy-Crofton formula
Integral and differential approaches:

Graph cuts and integral geometry

- Min-cut/max-flow algorithms find **globally optimal cut**

- In the most general case of directed graphs, a cost of **n-links** is a linear combination of **geometric length** and **flux** of a given vector field, e.g. **Riemannian**

  while **t-links** can implement any **regional bias**

[Boykov&Kolmogorov, ICCV 2003]
[Kolmogorov&Boykov, ICCV 2005]
Integral and differential approaches:

From global to local optimization

- In some problems local minima is desirable
  - when global minima is a trivial solution
  - when a good initial solution is known
  - many “shape prior” techniques rely on intermediate solutions (Daniel will explain more)

**differential approach**

- **Level-sets** is a variational optimization technique computing *gradient flow* evolution of contours converging to a local minima.

**integral approach**

- In fact, **graph cuts** can be also converted into a local optimization method.
Integral and differential approaches:

**Gradient flow of a contour for energy $F(C)$**

- Contour $C$ is a point in the space of all contours

![Diagram showing two contours $C$ and $C'$, with $C'$ being the best contour in the neighborhood of $C$.]

- **Gradient flow** evolution implies infinitesimal step in the space of contours giving the largest energy decrease among all small steps of the same size.
Integral and differential approaches:

**Differential approach to gradient flow**

- Level-sets and other *differential methods* for computing *gradient flow* of a contour explicitly estimate local motion (speed) at each point.

\[
\frac{dC}{dt} = \kappa \cdot \hat{N}
\]

and

\[
\frac{\partial u}{\partial t} = \kappa \cdot |\nabla u|
\]

- Local speed could be proportional to local curvature.
- e.g. *mean curvature flow* minimizing Euclidean length.
Integral and differential approaches:

Integral approach to gradient flow

- Discrete and continuous max-flow algorithms can “directly” compute an optimal step $C'$ in the small neighborhood of $C$.  

- **integral** approach to estimating contour evolution.
Integral and differential approaches:

Measuring distance between contours

- What is a small “neighborhood” of contour $C$?

$$\| C - C' \| \leq \varepsilon$$

- Typically, gradient flow is based on $L_2$ metric in the space of contours

*Boykov, Kolmogorov, Cremers, Delong, ECCV ’06*
Integral and differential approaches:

Measuring $L_2$ distance between contours

**Differential framework**

$$d \tilde{C}_s$$

**Integral framework**

$$D_0(p)$$

Euclidean distance map from $C_0$

$$\text{dist}^2(C, C_0) = \langle dC, dC \rangle = \int_{C_0} |dC_s|^2 ds$$

$$\text{dist}^2(C, C_0) = 2 \cdot \int_{\Delta C} D_0(p) \, dp$$

*Boykov, Kolmogorov, Cremers, Delong, ECCV ’06*
Integral and differential approaches:

**Integral approach to gradient flow**

\[
\min_{C : \text{dist}(C, C_0) = \varepsilon} F(C)
\]

\[
\min_C F(C) + \lambda \cdot \text{dist}^2(C, C_0)
\]

- Penalty for moving away from the current position
  - converts global optimization of \( F(C) \) into gradient descent (flow)

- There is a connection between \( \lambda \) and time

Boykov, Kolmogorov, Cremers, Delong, ECCV '06
Integral and differential approaches:

Integral approach to gradient flow

\[ E(C) = F(C) + \frac{1}{2(t-t_0)} \cdot \text{dist}^2(C, C_0) \]

Minimization of this energy is equivalent to solving a standard gradient flow equation:

\[ \frac{dE}{dC} = \frac{dF}{dC} + \frac{(C - C_0)}{(t-t_0)} \]

\[ t \rightarrow t_0 \quad \Rightarrow \quad \frac{dC}{dt} = -\frac{dF}{dC} \]

\[ E(C) \text{ can be minimized globally via discrete or continuous max-flow algorithms} \]

*Boykov, Kolmogorov, Cremers, Delong, ECCV '06*
Integral and differential approaches:

PDE cuts

Compute minimum cut for different values of time parameter $t$

$$E(C) = F(C) + \frac{1}{2(t - t_0)} \cdot dist^2(C, C_0)$$

- A sequence of cuts $C_0, C_1, C_2, \ldots, C_n$
- Transition times $t_0, t_1, t_2, \ldots, t_n$

$$F(C_0) > F(C_1) > F(C_2) > \ldots > F(C_n)$$

Initial solution smallest detectable step global minima

Local minima criteria: ......................
Integral and differential approaches:

Gradient flows via discrete graph cuts

\[ F(C) = I I C I I \epsilon \]

Under mean curvature motion any contour should converge to a circle before collapsing into a point.

4-grid

8-grid

16-grid

Boykov, Kolmogorov, Cremers, Delong, ECCV '06
Integral and differential approaches:

Gradient flows via discrete graph cuts

\[ F(C) = \| C \|_\epsilon \]

Under mean curvature motion a point on a contour Moves with a speed proportional to local curvature

NOTE: straight sides of the sausage should not move until the sausage collapses into a circle from the top and the bottom

Boykov, Kolmogorov, Cremers, Delong, ECCV ’06
Integral and differential approaches:

Gradient flows via discrete graph cuts

Theoretically, this plot should be

\[ r(t) = \sqrt{\text{const} - 2t} \]

Empirical plot for radius of a circle vs. time under mean curvature motion

Boykov, Kolmogorov, Cremers, Delong, ECCV '06
Integral and differential approaches:

PDE cuts for image based metric
Integral and differential approaches:

Gradient flows via discrete graph cuts

\[ F(C) = \| C \|_\varepsilon \]

mean curvature motion in 3D
Integral and differential approaches:

Gradient flows via discrete graph cuts

\[ F(C) = \| C \|_\varepsilon \quad \text{mean curvature motion in 3D} \]
Integral and differential approaches:

Gradient flows via discrete graph cuts

\[ F(C) = \| \cdot \|_\varepsilon \]

mean curvature motion in 3D
Integral and differential approaches:

Gradient flows via discrete graph cuts

\[ F(C) = \|C\|_{\varepsilon} \quad \text{mean curvature motion in 3D} \]
Integral and differential approaches:

Gradient flows via discrete graph cuts

\[ F(C) = \| C \|_\varepsilon \]  

mean curvature motion in 3D
Integral and differential approaches:

Gradient flows via discrete graph cuts

\[ F(C) = \| C \|_\varepsilon \]  
mean curvature motion in 3D
Integral and differential approaches:

Gradient flows via discrete graph cuts

\[ F(C) = ||C||_\varepsilon \]

mean curvature motion in 3D
Integral and differential approaches:

Gradient flows via discrete graph cuts

\[ F(C) = \| C \|_\varepsilon \] mean curvature motion in 3D
Integral and differential approaches:

Gradient flows via discrete graph cuts

\[ F(C) = \| I(C) \| \]  

mean curvature motion in 3D
Integral and differential approaches:

Gradient flows via discrete graph cuts

\[ F(C) = \| C \|_\varepsilon \]

mean curvature motion in 3D
Integral and differential approaches:

Gradient flows via discrete graph cuts

\[ F(C) = \| C \|_\epsilon \quad \text{mean curvature motion in 3D} \]
Integral and differential approaches:

Gradient flows via discrete graph cuts

\[ F(C) = \| C \|_\epsilon \quad \text{mean curvature motion in 3D} \]
Integral and differential approaches:

Gradient flows via discrete graph cuts

\[ F(C) = \| C \|_\epsilon \quad \text{mean curvature motion in 3D} \]
Integral and differential approaches:

Gradient flows via discrete graph cuts

mean curvature motion
Integral and differential approaches:

Earlier discrete methods for local optima

- **Banded graph cuts [Xu et al., CVPR 03]**
  - Binary 0-1 metric on the space of contours
    - Thresholding Hausdorff distance between contours
  - Jerky motion
  - Produces “erosion” in case of the *sausage* example
  - $r(t) = \text{const} - t$ in case of a *collapsing circle* example

- **DP-snakes [Amini et al., PAMI 1990]**
  - Explicit boundary representation
    - Constrained topology, non-geometric energy
  - Their method gives L1 metric on the space of contours
    - This is easy to correct based on insights in [BKCD, ECCV 2006]
  - 2D only
Integral and differential approaches:

**PDE cuts, pluses and minuses**

- Efficient binary search for $dt$ (reuses residual graph)
  - No guessing for choosing time step is required
- No oscillatory motion, guaranteed energy decrease
- Does not need to estimate surface derivatives
- Should reset distance map to better approximate gradient flow in $L2$ metric
- Can not produce arbitrarily small (sub-pixel) motion
- “Frying pan” artifact: small motion may be ignored if surface has large variation in curvature
Integral and differential approaches:

Summary

- Level-sets are based on ideas from **differential geometry**
  - sub-pixel accuracy, estimates derivatives
- Graph cuts use **integral geometry** to estimate length
  - no sub-pixel accuracy, but derivatives are unnecessary

- Level sets compute gradient flow by estimating local **differential motion** (speed) of contour points
  - derivatives (e.g. curvature) are estimated at every point
- Discrete or continuous max-flow algorithms directly estimate **integral motion** of a contour as a whole.
  - no derivatives at contour points are estimated