Deformable models

Vedrana Dahl, Anders Dahl
DTU Compute
02506 Advanced Image Analysis
March 2020

02506 Part II: Image analysis (segmentation) with geometric priors

Week 5: Markov random fields. Practical exercise: MRF modelling (Exercise 5.2) and bone segmentation (Exercise 5.5).

Week 6 (today): Deformable models. Mumford-Shah functional, Chan-Vese algorithm and snakes. Practical exercise: tracking deformation in a sequence of images.

Week 7: Layered surface detection. Exercise (tentative): detecting layers in bone data. Combining layered surface detection with deformable models.

Deformable models

What to learn? (Exam question related to this exercise)

- Explain the use of deformable models for image segmentation.
- Explain the piecewise-constant Mumford-Shah functional and how it is minimized using Chan-Vese algorithm?
- Explain the difference between the external and the internal forces.
- Explain curve representations used for deformable models, and their advantages/ disadvantages.

Reading material

- Chapter 6 in course note.
- Chan and Vese (2001). "Active contours without edges". TIP 2001.
- Xu, Pham, Prince (2009). "Image Segmentation Using Deformable Models", Handbook of Medical Imaging. Focus is on subsections 3.2.1 and 3.2.4.

Motivation

Demos

- Plusplus
- Amoeba and echiniscus
- Textures
- Nerves

Deformable models for image segmentation

Segmentation approach

1. Define segmentation energy E (optimality criterion given an image and a curve C) such that the desired segmentation has a minimal energy.
2. Derive energy-minimizing curve defromation forces $F = -\nabla E$.
3. Initialize the curve.
4. Make the curve dynamic and deform (evolve) until convergence $\frac{\partial C}{\partial t} = F(C)$.

Forces

- Internal (dependent on the curve, regularizer)
- External (dependent on the image, fidelity term)

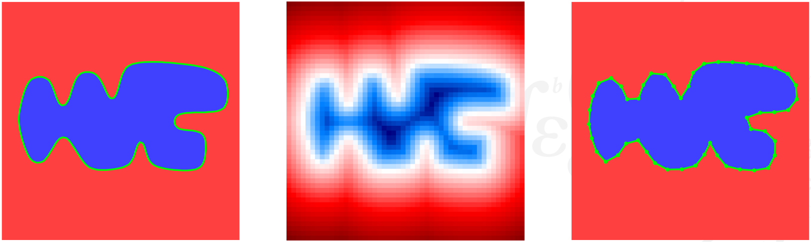
Curve representation

- Implicit (level set)
- Explicit (snakes)

Curve representation

Curves and surfaces

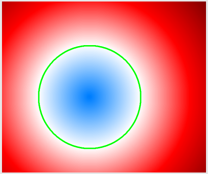
- Implicit discrete curve: level set
- Parametric (explicit) discrete curve: a snake



Curve representation

Deformable implicit curve

(Implicit equation for a circle: $x^2 + y^2 = r^2$)



A discrete implicit curve represented on a 500×600 grid

- Curve deforms by updating the level set function
- Supports topology changes
- No multi-phase support

Deformable parametric curve

(Parametric equation for a circle: $x = r \cos t,\ y = r \sin t,\ t = [0, 2\pi\rangle$)

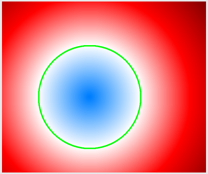
A discrete parametric curve represented by 400 points

- Curve deforms by displacing curve points
- No support for topology changes
- No multi-phase support

Curve representation

Deformable implicit curve

(Implicit equation for a circle: $x^2 + y^2 = r^2$)



A discrete implicit curve represented on a 500×600 grid

- Curve deforms by updating the level set function
- Supports topology changes
- No multi-phase support

Deformable parametric curve

(Parametric equation for a circle: $x = r\cos t,\ y = r\sin t,\ t = [0, 2\pi\rangle$)

A discrete parametric curve represented by 400 points

- Curve deforms by displacing curve points
- No support for topology changes
- No multi-phase support

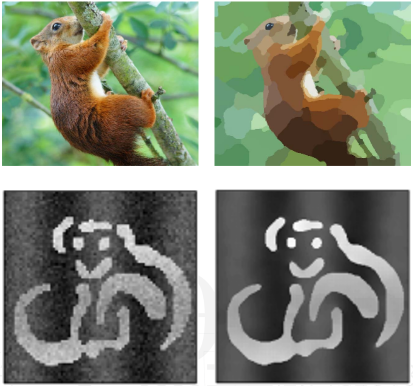
Mumford-Shah functional

- One of the most studied problems in image analysis
- General case

$$E(J,\Gamma) = \lambda \int_\Omega (J-I)^2 \mathrm{d}x + \int_{\Omega\setminus\Gamma} |\nabla I|^2 \mathrm{d}x + \mu \int_\Gamma \mathrm{d}s$$

- Piecewise-constant case

$$E(J,\Gamma) = \lambda \int_\Omega (J-I)^2 \mathrm{d}x + \mu \int_\Gamma \mathrm{d}s$$



- How to find the solution which minimizes the functional? Many approaches proposed, most based on special cases, approximations and relaxations.
- In the piecewise-constant case, curves have to be closed and Γ is implicitly given by J, so we can formulate $E(J)$.

Chan-Vese algorithm: contours without edges

A slightly simplified problem

- Consider a forground-background image (with unknown forground intensity m_F and background intensity m_B). The task is to separate the forground from the background.
- Find a curve which separates image into Ω_{in} and Ω_{out}, and find intensities m_{in} and m_{out} which minimize the energy

$$E = \int_{\Omega_{\mathrm{in}}} (m_{\mathrm{in}} - I)^2 \mathrm{d}x + \int_{\Omega_{\mathrm{out}}} (m_{\mathrm{out}} - I)^2 \mathrm{d}x + \mu \int_{\Gamma} \mathrm{d}s \ .$$

$$E = E_{\mathrm{ext}} + E_{\mathrm{int}}$$

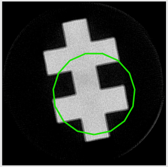
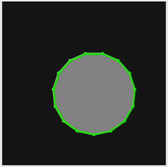
Chan-Vese algorithm

- Initialize a curve.
- Iterate until convergence between:
  - For given curve find m_{in} and m_{out} to minimize E.
  - For given m_{in} and m_{out} deform the curve to minimize E.
- Chan and Vese derived a solution for a level-set (implicit) curve using a variational approach.

Chan-Vese algorithm, for snakes

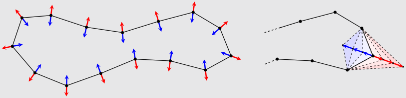
Optimal m_{in} and m_{out} given a curve

$$E_{\text{ext}} = \int_{\Omega_{\text{in}}} (m_{\text{in}} - I)^2 \mathrm{d}x + \int_{\Omega_{\text{out}}} (m_{\text{in}} - I)^2 \mathrm{d}x$$

- How to find m which minimizes $\sum_x (m - I(x))^2$?
- Minimizers are mean intensities inside and outside the curve.

Curve deformation given m_{in} and m_{out}



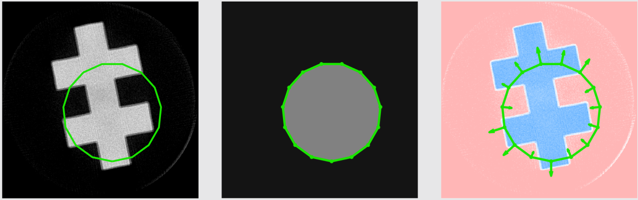
- We consider only displacement in normal direction.
- Area affected by displacement is proportional to the length of the displacement.

- continues -

Chan-Vese algorithm, for snakes

Curve deformation given m_{in} and m_{out} (continuation)

$$E_{\text{ext}} = \int_{\Omega_{\text{in}}} (m_{\text{in}} - I)^2 \mathrm{d}x + \int_{\Omega_{\text{out}}} (m_{\text{out}} - I)^2 \mathrm{d}x$$



- For a snake point i, how to find the (signed) length of an outward normal displacement d_i which minimizes E_{ext}?
- For d_i, the change in energy is proportional to

$$\left((m_{\text{in}} - I_i)^2 - (m_{\text{out}} - I_i)^2\right) d_i = (m_{\text{in}} - m_{\text{out}})\,(m_{\text{in}} + m_{\text{out}} - 2I_i)\, d_i$$

- Displacement which minimizes energy is given by the force

$$F_{\text{ext}} = \tau(m_{\text{in}} - m_{\text{out}})\,(2I - m_{\text{in}} - m_{\text{out}})\, N$$

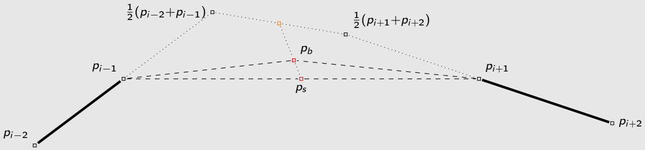
Snakes regularization

For a discrete snake

$$E_{\text{INT}} = \alpha \sum_{i \text{ (circular)}} ||p_{i+1} - p_i||^2 + \beta \sum_{i \text{ (circular)}} ||p_{i+1} - 2p_i + p_{i-1}||^2$$

- Stretching energy: sum of squared segment lengths
- Bending energy: sum of squared change between segments

For one point i



$$E_{\text{stretching}} = ||p_{i+1} - p_i||^2 + ||p_i - p_{i-1}||^2$$

$$E_{\text{bending}} = ||p_{i+2} - 2p_{i+1} + p_i||^2 + ||p_{i+1} - 2p_i + p_{i-1}||^2 + ||p_i - 2p_{i-1} + p_{i-2}||^2$$

- Optimal position for minimal stretching: $p_s = \frac{1}{2}(p_{i+1} + p_{i-1})$
- Optimal position for minimal bending: $p_b = \frac{1}{6}(-p_{i+2} + 4p_{i+1} + 4p_{i-1} - p_{i-2})$

Snakes regularization

Forward and backward

- Regularization corresponds to filtering (smoothing) the curve with filters for the first and the second derivative, i.e. a filter

$$\alpha \begin{bmatrix} 1 & -2 & 1 \end{bmatrix} + \beta \begin{bmatrix} -1 & 4 & -6 & 4 & -1 \end{bmatrix}$$

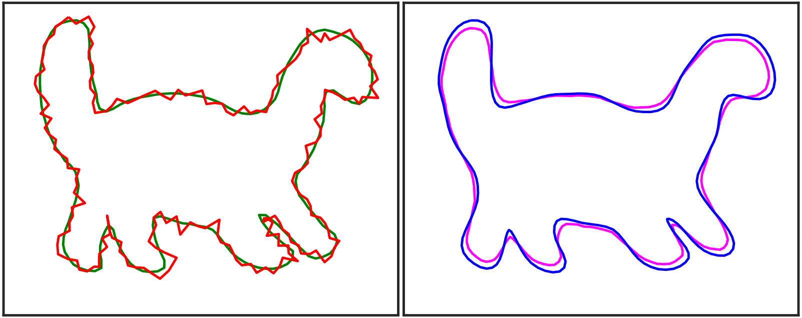
- Filtering can be achieved by a matrix multiplication

$$\mathbf{X}^{\text{new}} = (\mathbf{I} + \lambda \mathbf{L})\mathbf{X}$$

  where $\mathbf{L}$ is a N-times-N matrix containing the kernel and λ is a parameter controling the magnitude of displacement.
- For larger values of λ the curve will start oscillating, but using a small λ requires many iterations of the smoothing step for a noticeable result.
- Stability issues can be avoided by evaluating the displacement on the new snake $\mathbf{X}^{\text{new}}$. In other words, we can use an implicit (backwards Euler) approach where we use $\mathbf{X}^{\text{new}} = \mathbf{X} + \lambda \mathbf{L} \mathbf{X}^{\text{new}}$ leading to

$$\mathbf{X}^{\text{new}} = (\mathbf{I} - \lambda \mathbf{L})^{-1}\mathbf{X}$$

Snakes regularization



Dino, noisy dino, and smoothing results with different α and β weights.

Now: exercise

6.4 Exercise: Segmentation and tracking

- Suggestion: start by segmenting plusplus.
- Suggestion: start by hard-coding intensities of the foreground and the background.
- Visualize every step of deformation.

One iteration of the reduced algorithm

- `values = I(snake)` % Retrieve values, plot this (initially)
- `force = values - threshold` % Compute force
- `snake = snake + stepsize * force * normals` % Displace snake
- % Smooth snake
- % Fix snake
- % Plot snake