Active Contours
–Snakes and the Level Set Method

Niels Chr. Overgaard

2013-09-27

Outline

- What is an Image?
- What is an Active Contour?
- Deforming Force: The Edge Map
- The Snake Model
- The Level Set Representation
- The Chan-Vese Segmentation Model

Outline

- What is an Image?
- What is an Active Contour?
- Deforming Force: The Edge Map
- The Snake Model
- The Level Set Representation
- The Chan-Vese Segmentation Model

Outline

- What is an Image?
- What is an Active Contour?
- Deforming Force: The Edge Map
- The Snake Model
- The Level Set Representation
- The Chan-Vese Segmentation Model

Outline

- What is an Image?
- What is an Active Contour?
- Deforming Force: The Edge Map
- The Snake Model
- The Level Set Representation
- The Chan-Vese Segmentation Model

Outline

- What is an Image?
- What is an Active Contour?
- Deforming Force: The Edge Map
- The Snake Model
- The Level Set Representation
- The Chan-Vese Segmentation Model

Outline

- What is an Image?
- What is an Active Contour?
- Deforming Force: The Edge Map
- The Snake Model
- The Level Set Representation
- The Chan-Vese Segmentation Model

What is an Image?

- Consider a rectangular subset of the plane:

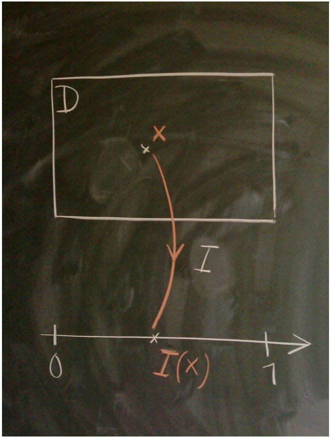
  $$D = [0, a] \times [0, b].$$

  $D \subset \mathbf{R}^2$ is called the image domain.

- A gray scale image is a function

  $$I : D \to [0, 1]$$

- $\boldsymbol{x} = (x, y) \in D$ is called a pixel or a point. $I(\boldsymbol{x})$ is the gray level of I at $\boldsymbol{x}$.



- A digital image is a regular sampling of I.

What is an Image?

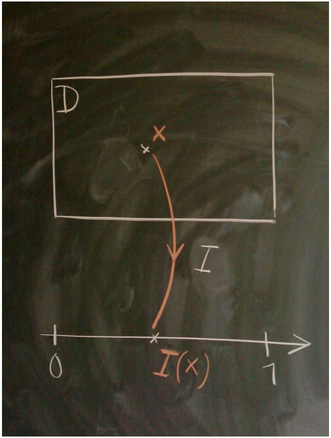
- Consider a rectangular subset of the plane:

$$D = [0, a] \times [0, b].$$

  $D \subset \mathbf{R}^2$ is called the image domain.
- A gray scale image is a function

$$I : D \to [0, 1]$$

- $\boldsymbol{x} = (x, y) \in D$ is called a pixel or a point. $I(\boldsymbol{x})$ is the gray level of I at $\boldsymbol{x}$.



- A digital image is a regular sampling of I.

What is an Image?

- Consider a rectangular subset of the plane:

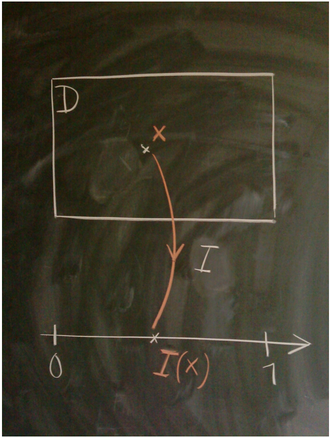
$$D = [0, a] \times [0, b].$$

  $D \subset \mathbf{R}^2$ is called the image domain.

- A gray scale image is a function

$$I : D \to [0, 1]$$

- $\boldsymbol{x} = (x, y) \in D$ is called a pixel or a point. $I(\boldsymbol{x})$ is the gray level of I at $\boldsymbol{x}$.



- A digital image is a regular sampling of I.

What is an Image?

- Consider a rectangular subset of the plane:

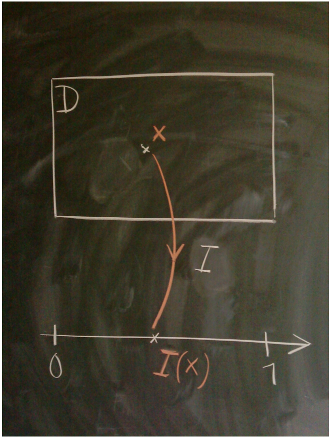
$$D = [0, a] \times [0, b].$$

  $D \subset \mathbf{R}^2$ is called the image domain.

- A gray scale image is a function

$$I : D \to [0, 1]$$

- $\boldsymbol{x} = (x, y) \in D$ is called a pixel or a point. $I(\boldsymbol{x})$ is the gray level of I at $\boldsymbol{x}$.



- A digital image is a regular sampling of I.

What is an Active Contour?

Image segmentation with an active contour:

xyz

Deforming Force Field – The Edge Map

Compute the image gradient:

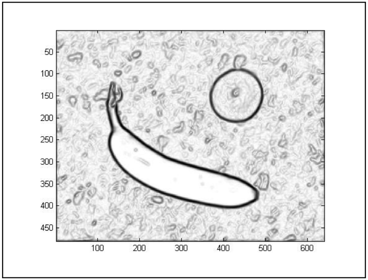
$$\nabla I(\boldsymbol{x}) = \left(\frac{\partial I(\boldsymbol{x})}{\partial x}, \frac{\partial I(\boldsymbol{x})}{\partial y}\right)$$

An edge map is a function based on the image gradient, e.g.

$$V(\boldsymbol{x}) = -|\nabla I(\boldsymbol{x})|$$

or

$$V(\boldsymbol{x}) = \frac{1}{1 + |\nabla I(\boldsymbol{x})|}.$$



The deforming force is (minus) the gradient of the edge map: $F = -\nabla V$.

The Snake Energy

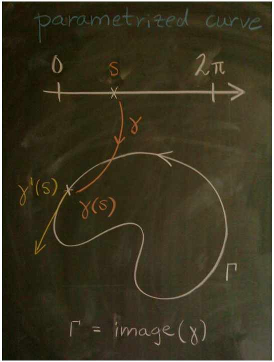
- The contour is a parametrized curve:

$$\gamma : [0, 2\pi] \to D \subset \mathbf{R}^2.$$

Thus $\gamma(s)$ is a pixel in D for every parameter $s, 0 \le s \le 2\pi$.

- The snake energy is a function (or functional) whose argument is an entire contour γ:

$$E[\gamma] = \alpha \int_0^{2\pi} \frac{1}{2} |\gamma'(s)|^2 \, ds + \beta \int_0^{2\pi} V(\gamma(s)) \, ds$$



The Snake Energy

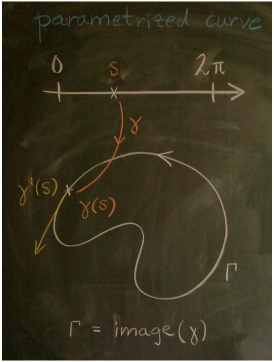
- The contour is a parametrized curve:

$$\gamma : [0, 2\pi] \to D \subset \mathbf{R}^2.$$

Thus $\gamma(s)$ is a pixel in D for every parameter $s, 0 \le s \le 2\pi$.

- The snake energy is a function (or functional) whose argument is an entire contour γ:

$$E[\gamma] = \alpha \int_0^{2\pi} \frac{1}{2} |\gamma'(s)|^2 \, ds + \beta \int_0^{2\pi} V(\gamma(s)) \, ds$$



Minimization of the Snake Energy

- The mathematical problem:

$$\text{Segmentation} = \text{The optimal contour} = \arg\min_\gamma E[\gamma]$$

- Structure:

$$E[\gamma] = E_{\text{reg}}[\gamma] + E_{\text{data}}[\gamma].$$

  - E_{reg} –Regularizing term. Will shorten the contour and make it more smooth.
  - E_{data} –External energy, fidelity term or data term. Is small at image features e.g. edges.
  - α, β –Parameters used to weight the relative importance of E_{int} and E_{ext}.

Minimization of the Snake Energy

- The mathematical problem:

$$\text{Segmentation} = \text{The optimal contour} = \arg\min_\gamma E[\gamma]$$

- Structure:

$$E[\gamma] = E_{\text{reg}}[\gamma] + E_{\text{data}}[\gamma].$$

  - E_{reg} –Regularizing term. Will shorten the contour and make it more smooth.
  - E_{data} –External energy, fidelity term or data term. Is small at image features e.g. edges.
  - α, β –Parameters used to weight the relative importance of E_{int} and E_{ext}.

Minimization of the Snake Energy

- The mathematical problem:

$$\text{Segmentation} = \text{The optimal contour} = \arg\min_{\gamma} E[\gamma]$$

- Structure:

$$E[\gamma] = E_{\text{reg}}[\gamma] + E_{\text{data}}[\gamma].$$

  - E_{reg} –Regularizing term. Will shorten the contour and make it more smooth.
  - E_{data} –External energy, fidelity term or data term. Is small at image features e.g. edges.
  - α, β –Parameters used to weight the relative importance of E_{int} and E_{ext}

Minimization of the Snake Energy

- The mathematical problem:

$$\text{Segmentation} = \text{The optimal contour} = \arg\min_\gamma E[\gamma]$$

- Structure:

$$E[\gamma] = E_{\text{reg}}[\gamma] + E_{\text{data}}[\gamma].$$

  - E_{reg} –Regularizing term. Will shorten the contour and make it more smooth.
  - E_{data} –External energy, fidelity term or data term. Is small at image features e.g. edges.
  - α, β –Parameters used to weight the relative importance of E_{int} and E_{ext}

Minimization of the Snake Energy

- The mathematical problem:

$$\text{Segmentation} = \text{The optimal contour} = \arg\min_{\gamma} E[\gamma]$$

- Structure:

$$E[\gamma] = E_{\text{reg}}[\gamma] + E_{\text{data}}[\gamma].$$

  - E_{reg} –Regularizing term. Will shorten the contour and make it more smooth.
  - E_{data} –External energy, fidelity term or data term. Is small at image features e.g. edges.
  - α, β –Parameters used to weight the relative importance of E_{int} and E_{ext}.

Steepest Descent

- Consider time-dependent contours:

$$\gamma = \gamma(s;t) = \gamma_t(s), \qquad t \geq 0.$$

- The snake energy is minimized using gradient descent:

$$\dot{\gamma}_t = -\nabla E[\gamma_t], \qquad (\ \dot{}\ = \frac{d}{dt})$$

Here $\nabla E[\gamma]$ is the variational gradient/derivative computed at the contour γ.

- For the snake energy the variational gradient is

$$\nabla E[\gamma](s) = -\alpha\gamma''(s) + \beta\nabla V(\gamma(s)), \qquad (\ ' = \frac{d}{ds})$$

Derivation requires the theory from calculus of variations.

Steepest Descent

- Consider time-dependent contours:

$$\gamma = \gamma(s; t) = \gamma_t(s), \qquad t \geq 0.$$

- The snake energy is minimized using gradient descent:

$$\dot{\gamma}_t = -\nabla E[\gamma_t], \qquad (\,\dot{}\, = \frac{d}{dt})$$

Here $\nabla E[\gamma]$ is the variational gradient/derivative computed at the contour γ.

- For the snake energy the variational gradient is

$$\nabla E[\gamma](s) = -\alpha\gamma''(s) + \beta\nabla V(\gamma(s)), \qquad (\,'\, = \frac{d}{ds})$$

Derivation requires the theory from calculus of variations.

Steepest Descent

- Consider time-dependent contours:

$$\gamma = \gamma(s;t) = \gamma_t(s), \qquad t \geq 0.$$

- The snake energy is minimized using gradient descent:

$$\dot{\gamma}_t = -\nabla E[\gamma_t], \qquad (\ \dot{}\ = \frac{d}{dt})$$

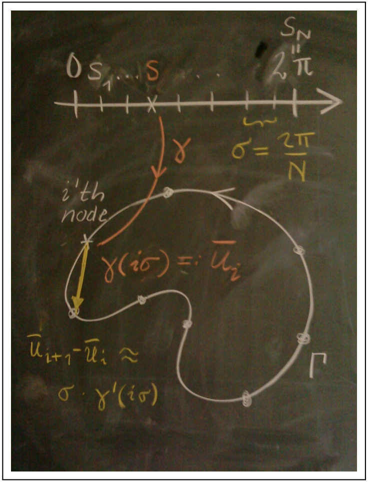
Here $\nabla E[\gamma]$ is the variational gradient/derivative computed at the contour γ.

- For the snake energy the variational gradient is

$$\nabla E[\gamma](s) = -\alpha\gamma''(s) + \beta\nabla V(\gamma(s)), \qquad (\ {}'\ = \frac{d}{ds})$$

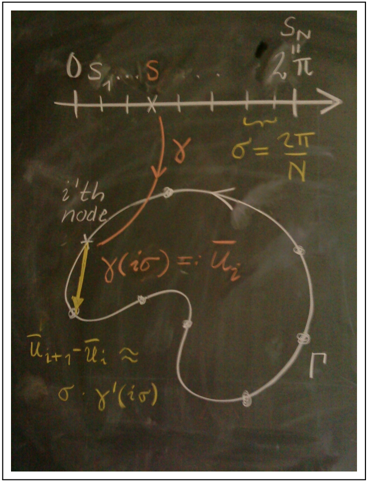
Derivation requires the theory from calculus of variations.

Discretizing the Snake I



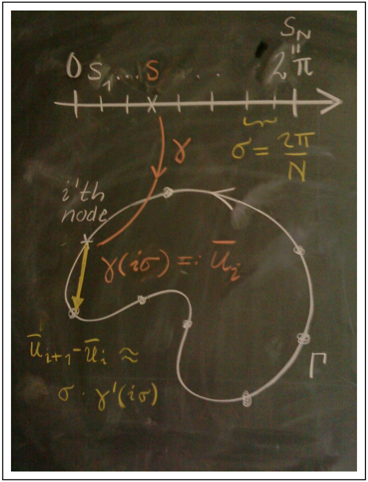
- N = number of discretization points in $[0, 2\pi]$
- $\sigma = 2\pi/N$ corresponding step length.
- $s_i = i\sigma$, $i = 1, \ldots, N$ discretization points.
- τ = time step. $t^k = k\tau$, $k = 0, 1, 2, \ldots$ time points.
- $\mathbf{u}_i^k = \gamma(s_i, t^k) = \gamma(i\sigma, k\tau)$ = i'th node.
- $\mathbf{u}^k = (\mathbf{u}_1^k, \ldots, \mathbf{u}_N^k)^T$ = discretized contour.

Discretizing the Snake I



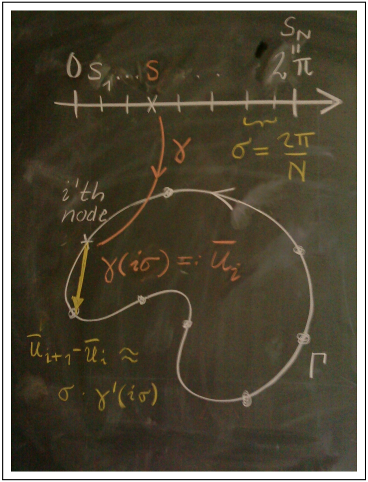
- N = number of discretization points in $[0, 2\pi]$
- $\sigma = 2\pi/N$ corresponding step length.
- $s_i = i\sigma$, $i = 1, \ldots, N$ discretization points.
- τ = time step. $t^k = k\tau$, $k = 0, 1, 2, \ldots$ time points.
- $\mathbf{u}_i^k = \gamma(s_i, t^k) = \gamma(i\sigma, k\tau)$ = i'th node.
- $\mathbf{u}^k = (\mathbf{u}_1^k, \ldots, \mathbf{u}_N^k)^T =$ discretized contour.

Discretizing the Snake I



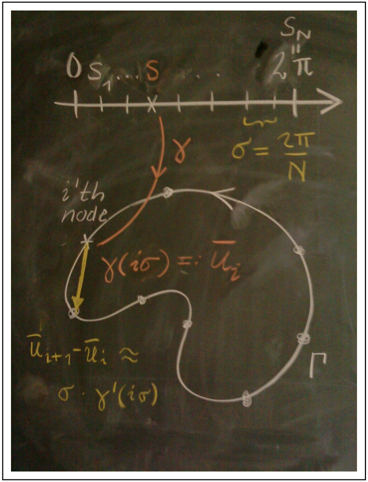
- N = number of discretization points in $[0, 2\pi]$
- $\sigma = 2\pi/N$ corresponding step length.
- $s_i = i\sigma$, $i = 1, \ldots, N$ discretization points.
- τ = time step. $t^k = k\tau$, $k = 0, 1, 2, \ldots$ time points.
- $\mathbf{u}_i^k = \gamma(s_i, t^k) = \gamma(i\sigma, k\tau)$ = i'th node.
- $\mathbf{u}^k = (\mathbf{u}_1^k, \ldots, \mathbf{u}_N^k)^T$ = discretized contour.

Discretizing the Snake I



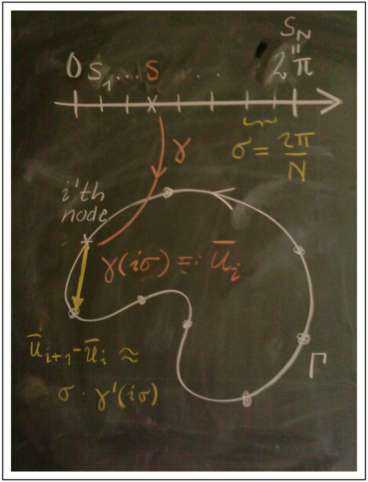
- N = number of discretization points in $[0, 2\pi]$
- $\sigma = 2\pi/N$ corresponding step length.
- $s_i = i\sigma$, $i = 1, \ldots, N$ discretization points.
- τ = time step. $t^k = k\tau$, $k = 0, 1, 2, \ldots$ time points.
- $\mathbf{u}_i^k = \gamma(s_i, t^k) = \gamma(i\sigma, k\tau)$ = i'th node.
- $\mathbf{u}^k = (\mathbf{u}_1^k, \ldots, \mathbf{u}_N^k)^T =$ discretized contour.

Discretizing the Snake *I*



- N = number of discretization points in $[0, 2\pi]$
- $\sigma = 2\pi/N$ corresponding step length.
- $s_i = i\sigma$, $i = 1, \ldots, N$ discretization points.
- τ = time step. $t^k = k\tau$, $k = 0, 1, 2, \ldots$ time points.
- $\mathbf{u}_i^k = \gamma(s_i, t^k) = \gamma(i\sigma, k\tau)$ = i'th node.
- $\mathbf{u}^k = (\mathbf{u}_1^k, \ldots, \mathbf{u}_N^k)^T =$ discretized contour.

Discretizing the Snake I



- N = number of discretization points in $[0, 2\pi]$
- $\sigma = 2\pi/N$ corresponding step length.
- $s_i = i\sigma$, $i = 1, \ldots, N$ discretization points.
- τ = time step. $t^k = k\tau$, $k = 0, 1, 2, \ldots$ time points.
- $\mathbf{u}_i^k = \gamma(s_i, t^k) = \gamma(i\sigma, k\tau)$ $=$ i'th node.
- $\mathbf{u}^k = (\mathbf{u}_1^k, \ldots, \mathbf{u}_N^k)^T =$ discretized contour.

Discretizing the Snake Energy *II*

If we use the finite differences for the velocity:

$$\gamma'(i\sigma) = \frac{\mathbf{u}_{i+1} - \mathbf{u}_i}{\sigma} = (D\mathbf{u})_i$$

where $D =$ forward difference operator.
(Obs! Cyclic indexing; $\mathbf{u}_{N+1}$ is interpreted as $\mathbf{u}_1$)

Discretized snake energy:

$$E[\mathbf{u}] = \alpha \sum_{i=1}^{N} \frac{1}{2} \left| \frac{\mathbf{u}_{i+1} - \mathbf{u}_i}{\sigma} \right|^2 \sigma + \beta \sum_{i=1}^{N} V(\mathbf{u}_i)\sigma$$

Discretizing the Snake Energy *II*

If we use the finite differences for the velocity:

$$\gamma'(i\sigma) = \frac{\mathbf{u}_{i+1} - \mathbf{u}_i}{\sigma} = (D\mathbf{u})_i$$

where $D =$ forward difference operator.
(Obs! Cyclic indexing; $\mathbf{u}_{N+1}$ is interpreted as $\mathbf{u}_1$)

Discretized snake energy:

$$E[\mathbf{u}] = \alpha \sum_{i=1}^{N} \frac{1}{2} \left| \frac{\mathbf{u}_{i+1} - \mathbf{u}_i}{\sigma} \right|^2 \sigma + \beta \sum_{i=1}^{N} V(\mathbf{u}_i)\sigma$$

Gradient of the Snake Energy

We throw away the common factor σ and redefine:

$$E[\mathbf{u}] = \alpha \sum_{i=1}^{N} \frac{1}{2} \left| \frac{\mathbf{u}_{i+1} - \mathbf{u}_i}{\sigma} \right|^2 + \beta \sum_{i=1}^{N} V(\mathbf{u}_i)$$

Compute the partial derivative wrt. the i'th node:

$$\frac{\partial}{\partial \mathbf{u}_i} E[\mathbf{u}] = -\alpha \frac{\mathbf{u}_{i+1} - 2\mathbf{u}_i - \mathbf{u}_{i-1}}{\sigma^2} + \beta \nabla V(\mathbf{u}_i).$$

The i'th component of the gradient of E may be written

$$\frac{\partial}{\partial \mathbf{u}_i} E[\mathbf{u}] = -\alpha (D^2 \mathbf{u})_i + \beta \nabla V(\mathbf{u}_i).$$

Gradient of the Snake Energy

We throw away the common factor σ and redefine:

$$E[\mathbf{u}] = \alpha \sum_{i=1}^{N} \frac{1}{2} \left| \frac{\mathbf{u}_{i+1} - \mathbf{u}_i}{\sigma} \right|^2 + \beta \sum_{i=1}^{N} V(\mathbf{u}_i)$$

Compute the partial derivative wrt. the i'th node:

$$\frac{\partial}{\partial \mathbf{u}_i} E[\mathbf{u}] = -\alpha \frac{\mathbf{u}_{i+1} - 2\mathbf{u}_i - \mathbf{u}_{i-1}}{\sigma^2} + \beta \nabla V(\mathbf{u}_i).$$

The i'th component of the gradient of E may be written

$$\frac{\partial}{\partial \mathbf{u}_i} E[\mathbf{u}] = -\alpha (D^2 \mathbf{u})_i + \beta \nabla V(\mathbf{u}_i).$$

Gradient of the Snake Energy

We throw away the common factor σ and redefine:

$$E[\mathbf{u}] = \alpha \sum_{i=1}^{N} \frac{1}{2} \left| \frac{\mathbf{u}_{i+1} - \mathbf{u}_i}{\sigma} \right|^2 + \beta \sum_{i=1}^{N} V(\mathbf{u}_i)$$

Compute the partial derivative wrt. the i'th node:

$$\frac{\partial}{\partial \mathbf{u}_i} E[\mathbf{u}] = -\alpha \frac{\mathbf{u}_{i+1} - 2\mathbf{u}_i - \mathbf{u}_{i-1}}{\sigma^2} + \beta \nabla V(\mathbf{u}_i).$$

The i'th component of the gradient of E may be written

$$\frac{\partial}{\partial \mathbf{u}_i} E[\mathbf{u}] = -\alpha (D^2 \mathbf{u})_i + \beta \nabla V(\mathbf{u}_i).$$

Numerics *I*: Two Descent Strategies

We use forward differences for the time derivative:

$$\dot{\gamma}(i\sigma, k\tau) = \frac{\mathbf{u}_i^{k+1} - \mathbf{u}_i^k}{\tau}$$

Explicit Euler Scheme:

$$\frac{\mathbf{u}_i^{k+1} - \mathbf{u}_i^k}{\tau} = -\alpha(D^2\mathbf{u}^k)_i + \beta\nabla V(\mathbf{u}_i^k)$$

Semi-implicit Scheme:

$$\frac{\mathbf{u}_i^{k+1} - \mathbf{u}_i^k}{\tau} + \alpha(D^2\mathbf{u}^{k+1})_i = \beta\nabla V(\mathbf{u}_i^k)$$

Notice that the external force is non-linear.

Numerics *I*: Two Descent Strategies

We use forward differences for the time derivative:

$$\dot{\gamma}(i\sigma, k\tau) = \frac{\mathbf{u}_i^{k+1} - \mathbf{u}_i^k}{\tau}$$

Explicit Euler Scheme:

$$\frac{\mathbf{u}_i^{k+1} - \mathbf{u}_i^k}{\tau} = -\alpha(D^2\mathbf{u}^k)_i + \beta\nabla V(\mathbf{u}_i^k)$$

Semi-implicit Scheme:

$$\frac{\mathbf{u}_i^{k+1} - \mathbf{u}_i^k}{\tau} + \alpha(D^2\mathbf{u}^{k+1})_i = \beta\nabla V(\mathbf{u}_i^k)$$

Notice that the external force is non-linear.

Numerics I: Two Descent Strategies

We use forward differences for the time derivative:

$$\dot{\gamma}(i\sigma, k\tau) = \frac{\mathbf{u}_i^{k+1} - \mathbf{u}_i^k}{\tau}$$

Explicit Euler Scheme:

$$\frac{\mathbf{u}_i^{k+1} - \mathbf{u}_i^k}{\tau} = -\alpha(D^2\mathbf{u}^k)_i + \beta\nabla V(\mathbf{u}_i^k)$$

Semi-implicit Scheme:

$$\frac{\mathbf{u}_i^{k+1} - \mathbf{u}_i^k}{\tau} + \alpha(D^2\mathbf{u}^{k+1})_i = \beta\nabla V(\mathbf{u}_i^k)$$

Notice that the external force is non-linear.

Numerics *II*: Evolving the Contour

The discrete second derivative is computed as

$$D^2\mathbf{u} = \frac{1}{2}A\mathbf{u}, \qquad A = \begin{bmatrix} -2 & 1 & 0 & \dots & 0 & 1 \\ 1 & -2 & 1 & \dots & 0 & 0 \\ 0 & 1 & -2 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 1 & 0 & 0 & \dots & 1 & -2 \end{bmatrix}$$

Define $F(\mathbf{u}) = (F_1(\mathbf{u}), \dots F_N(\mathbf{u}))^T$ where $F_i(\mathbf{u}) = -\nabla V(\mathbf{u}_i)$, $i = 1, \dots, N$.

Rearrange the explicit Euler vector equation as:

$$\left(I - \frac{\alpha\tau}{\sigma^2}A\right)\mathbf{u}^{k+1} = \mathbf{u}^k + \beta\tau F(\mathbf{u}^k).$$

The contour is evolved $\mathbf{u}^k \longrightarrow \mathbf{u}^{k+1}$ by solving this equation.

Numerics *II*: Evolving the Contour

The discrete second derivative is computed as

$$D^2\mathbf{u} = \frac{1}{2}A\mathbf{u}, \qquad A = \begin{bmatrix} -2 & 1 & 0 & \dots & 0 & 1 \\ 1 & -2 & 1 & \dots & 0 & 0 \\ 0 & 1 & -2 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 1 & 0 & 0 & \dots & 1 & -2 \end{bmatrix}$$

Define $F(\mathbf{u}) = (F_1(\mathbf{u}), \dots F_N(\mathbf{u}))^T$ where $F_i(\mathbf{u}) = -\nabla V(\mathbf{u}_i)$, $i = 1, \dots, N$.

Rearrange the explicit Euler vector equation as:

$$\left(I - \frac{\alpha\tau}{\sigma^2}A\right)\mathbf{u}^{k+1} = \mathbf{u}^k + \beta\tau F(\mathbf{u}^k).$$

The contour is evolved $\mathbf{u}^k \longrightarrow \mathbf{u}^{k+1}$ by solving this equation.

Numerics *II*: Evolving the Contour

The discrete second derivative is computed as

$$D^2\mathbf{u} = \frac{1}{2}A\mathbf{u}, \qquad A = \begin{bmatrix} -2 & 1 & 0 & \dots & 0 & 1 \\ 1 & -2 & 1 & \dots & 0 & 0 \\ 0 & 1 & -2 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 1 & 0 & 0 & \dots & 1 & -2 \end{bmatrix}$$

Define $F(\mathbf{u}) = (F_1(\mathbf{u}), \dots F_N(\mathbf{u}))^T$ where $F_i(\mathbf{u}) = -\nabla V(\mathbf{u}_i)$, $i = 1, \dots, N$.

Rearrange the explicit Euler vector equation as:

$$\left(I - \frac{\alpha\tau}{\sigma^2}A\right)\mathbf{u}^{k+1} = \mathbf{u}^k + \beta\tau F(\mathbf{u}^k).$$

The contour is evolved $\mathbf{u}^k \longrightarrow \mathbf{u}^{k+1}$ by solving this equation.

Stop Criterion and Parameter Settings

...?

Examples: Success and Failure of the Snake Model

xyz

xyz

Local Optima: Trivial Equilibrium Solutions

At equilibrium (steady state) $\mathbf{u}^{k+1} = \mathbf{u}^k$. Thus equilibrium $\mathbf{u}$ satisfies:

$$\Big(I - \frac{\alpha\tau}{\sigma^2}A\Big)\mathbf{u} = \mathbf{u} + \beta\tau F(\mathbf{u}).$$

Suppose $\mathbf{u} = (\mathbf{c}, \mathbf{c}, \ldots, \mathbf{c})^T$ for some $\mathbf{c} \in \mathbf{R}^2$, then $\mathbf{u} = \mathbf{1}\mathbf{c}$ where $\mathbf{1} = (1, 1, \ldots, 1)^T$.

$$\text{Since} \quad A\mathbf{1} = 0 \quad \text{we find} \quad \Big(I - \frac{\alpha\tau}{\sigma^2}A\Big)\mathbf{u} = \mathbf{u}.$$

Suppose also that $\nabla V(\mathbf{c}) = 0$, then $F(\mathbf{u}) = 0$ and the right hand side of the steady state equation becomes

$$\mathbf{u} + \beta\tau F(\mathbf{u}) = \mathbf{u}.$$

Thus $\mathbf{u} = (\mathbf{c}, \mathbf{c}, \ldots, \mathbf{c})^T$ is a (trivial) equilibrium.

Local Optima: Trivial Equilibrium Solutions

At equilibrium (steady state) $\mathbf{u}^{k+1} = \mathbf{u}^k$. Thus equilibrium $\mathbf{u}$ satisfies:

$$\left(I - \frac{\alpha\tau}{\sigma^2}A\right)\mathbf{u} = \mathbf{u} + \beta\tau F(\mathbf{u}).$$

Suppose $\mathbf{u} = (\mathbf{c}, \mathbf{c}, \dots, \mathbf{c})^T$ for some $\mathbf{c} \in \mathbf{R}^2$, then $\mathbf{u} = \mathbf{1}\mathbf{c}$ where $\mathbf{1} = (1, 1, \dots, 1)^T$.

$$\text{Since} \quad A\mathbf{1} = 0 \quad \text{we find} \quad \left(I - \frac{\alpha\tau}{\sigma^2}A\right)\mathbf{u} = \mathbf{u}.$$

Suppose also that $\nabla V(\mathbf{c}) = 0$, then $F(\mathbf{u}) = 0$ and the right hand side of the steady state equation becomes

$$\mathbf{u} + \beta\tau F(\mathbf{u}) = \mathbf{u}.$$

Thus $\mathbf{u} = (\mathbf{c}, \mathbf{c}, \dots, \mathbf{c})^T$ is a (trivial) equilibrium.

Local Optima: Trivial Equilibrium Solutions

At equilibrium (steady state) $\mathbf{u}^{k+1} = \mathbf{u}^k$. Thus equilibrium $\mathbf{u}$ satisfies:

$$\left(I - \frac{\alpha\tau}{\sigma^2}A\right)\mathbf{u} = \mathbf{u} + \beta\tau F(\mathbf{u}).$$

Suppose $\mathbf{u} = (\mathbf{c}, \mathbf{c}, \ldots, \mathbf{c})^T$ for some $\mathbf{c} \in \mathbf{R}^2$, then $\mathbf{u} = \mathbf{1}\mathbf{c}$ where $\mathbf{1} = (1, 1, \ldots, 1)^T$.

$$\text{Since} \quad A\mathbf{1} = 0 \quad \text{we find} \quad \left(I - \frac{\alpha\tau}{\sigma^2}A\right)\mathbf{u} = \mathbf{u}.$$

Suppose also that $\nabla V(\mathbf{c}) = 0$, then $F(\mathbf{u}) = 0$ and the right hand side of the steady state equation becomes

$$\mathbf{u} + \beta\tau F(\mathbf{u}) = \mathbf{u}.$$

Thus $\mathbf{u} = (\mathbf{c}, \mathbf{c}, \ldots, \mathbf{c})^T$ is a (trivial) equilibrium.

The Level Set Method

- Parametric representation of the unit circle (snake):

$$\gamma(s) = (\cos(s), \sin(s)), \quad 0 \leq s \leq 2\pi.$$

- Implicit representation of the unit circle (level set):

$$\Gamma = \{\boldsymbol{x} = (x, y) \in \mathbf{R}^2 | x^2 + y^2 - 1 = 0\}.$$

- Generally, a closed curve Γ in the plane is given by:

$$\Gamma = \{\boldsymbol{x} = (x, y) \in \mathbf{R}^2 | \phi(x, y) = 0\}.$$

Here ϕ is called the level set function.

The Level Set Method

- Parametric representation of the unit circle (snake):

$$\gamma(s) = (\cos(s), \sin(s)), \quad 0 \leq s \leq 2\pi.$$

- Implicit representation of the unit circle (level set):

$$\Gamma = \{\boldsymbol{x} = (x, y) \in \mathbf{R}^2 | x^2 + y^2 - 1 = 0\}.$$

- Generally, a closed curve Γ in the plane is given by:

$$\Gamma = \{\boldsymbol{x} = (x, y) \in \mathbf{R}^2 | \phi(x, y) = 0\}.$$

Here ϕ is called the level set function.

The Level Set Method

- Parametric representation of the unit circle (snake):

$$\gamma(s) = (\cos(s), \sin(s)), \quad 0 \leq s \leq 2\pi.$$

- Implicit representation of the unit circle (level set):

$$\Gamma = \{\boldsymbol{x} = (x, y) \in \mathbf{R}^2 | x^2 + y^2 - 1 = 0\}.$$

- Generally, a closed curve Γ in the plane is given by:

$$\Gamma = \{\boldsymbol{x} = (x, y) \in \mathbf{R}^2 | \phi(x, y) = 0\}.$$

Here ϕ is called the level set function.

Computing Geometric Entities from ϕ

- Outward unit normal:

$$\mathbf{n}(\boldsymbol{x}) = \frac{\nabla\phi(\boldsymbol{x})}{|\nabla\phi(\boldsymbol{x})|}, \quad \boldsymbol{x} \in \Gamma.$$

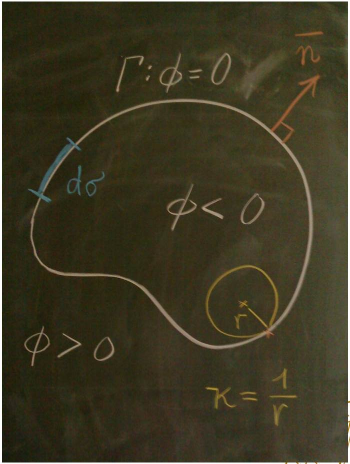
- Curvature of contour at $\boldsymbol{x} \in \Gamma$:

$$\kappa(\boldsymbol{x}) = \operatorname{div}\left(\frac{\nabla\phi(\boldsymbol{x})}{|\nabla\phi(\boldsymbol{x})|}\right).$$

- Curve length element $d\sigma$:

$$d\sigma = \delta(\phi(\boldsymbol{x}))|\nabla\phi(\boldsymbol{x})|\,dx$$

where $\delta(z)$ is the Dirac delta function on $\mathbf{R}$.



Computing Geometric Entities from ϕ

- Outward unit normal:

$$\mathbf{n}(\boldsymbol{x}) = \frac{\nabla\phi(\boldsymbol{x})}{|\nabla\phi(\boldsymbol{x})|}, \quad \boldsymbol{x} \in \Gamma.$$

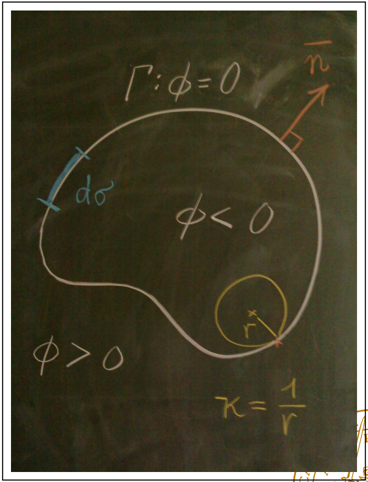
- Curvature of contour at $\boldsymbol{x} \in \Gamma$:

$$\kappa(\boldsymbol{x}) = \operatorname{div}\left(\frac{\nabla\phi(\boldsymbol{x})}{|\nabla\phi(\boldsymbol{x})|}\right).$$

- Curve length element $d\sigma$:

$$d\sigma = \delta(\phi(\boldsymbol{x}))|\nabla\phi(\boldsymbol{x})|\,dx$$

where $\delta(z)$ is the Dirac delta function on $\mathbf{R}$.



Computing Geometric Entities from ϕ

- Outward unit normal:

$$\mathbf{n}(\boldsymbol{x}) = \frac{\nabla\phi(\boldsymbol{x})}{|\nabla\phi(\boldsymbol{x})|}, \quad \boldsymbol{x} \in \Gamma.$$

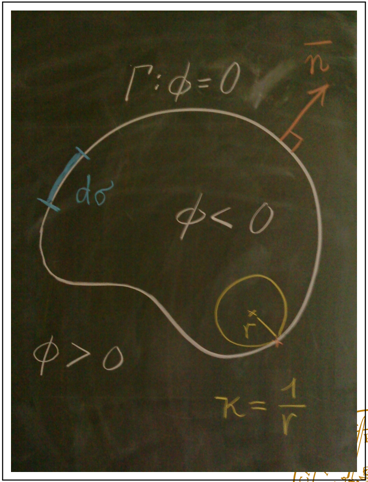
- Curvature of contour at $\boldsymbol{x} \in \Gamma$:

$$\kappa(\boldsymbol{x}) = \operatorname{div}\left(\frac{\nabla\phi(\boldsymbol{x})}{|\nabla\phi(\boldsymbol{x})|}\right).$$

- Curve length element $d\sigma$:

$$d\sigma = \delta(\phi(\boldsymbol{x}))|\nabla\phi(\boldsymbol{x})|\, dx$$

where $\delta(z)$ is the Dirac delta function on $\mathbf{R}$.



Kinematics

A moving contour is represented by a time-dependent level set function:

$$\Gamma(t) = \{\boldsymbol{x} \in \mathbf{R}^2 | \phi(\boldsymbol{x}, t) = 0\}.$$

The normal velocity of $\Gamma(t)$ is given by the formula:

$$\frac{d}{dt}\Gamma(t) = -\frac{\partial \phi(\boldsymbol{x}, t)/\partial t}{|\nabla \phi(\boldsymbol{x}, t)|}, \quad \boldsymbol{x} \in \Gamma(t).$$

To see this, consider a particle $t \mapsto \gamma(t)$ moving along with the contour $\Gamma(t)$. Then clearly $\phi(\gamma(t), t) = 0$ for all t. Compute and interpret the derivative of this identity!

Kinematics

A moving contour is represented by a time-dependent level set function:

$$\Gamma(t) = \{\boldsymbol{x} \in \mathbf{R}^2 | \phi(\boldsymbol{x}, t) = 0\}.$$

The normal velocity of $\Gamma(t)$ is given by the formula:

$$\frac{d}{dt}\Gamma(t) = -\frac{\partial \phi(\boldsymbol{x}, t)/\partial t}{|\nabla \phi(\boldsymbol{x}, t)|}, \quad \boldsymbol{x} \in \Gamma(t).$$

To see this, consider a particle $t \mapsto \gamma(t)$ moving along with the contour $\Gamma(t)$. Then clearly $\phi(\gamma(t), t) = 0$ for all t. Compute and interpret the derivative of this identity!

Kinematics

A moving contour is represented by a time-dependent level set function:

$$\Gamma(t) = \{\boldsymbol{x} \in \mathbf{R}^2 | \phi(\boldsymbol{x}, t) = 0\}.$$

The normal velocity of $\Gamma(t)$ is given by the formula:

$$\frac{d}{dt}\Gamma(t) = -\frac{\partial\phi(\boldsymbol{x}, t)/\partial t}{|\nabla\phi(\boldsymbol{x}, t)|}, \quad \boldsymbol{x} \in \Gamma(t).$$

To see this, consider a particle $t \mapsto \gamma(t)$ moving along with the contour $\Gamma(t)$. Then clearly $\phi(\gamma(t), t) = 0$ for all t. Compute and interpret the derivative of this identity!

The Chan-Vese Model I.

Segment gray scale image
$I : D \to [0,1]$.

Model:

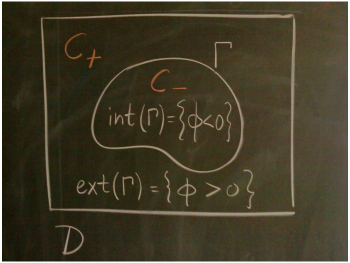
- Piecewise constant image

$$I_{\text{model}}(\boldsymbol{x}) = \begin{cases} c_- & \text{if } \phi(\boldsymbol{x}) < 0 \\ c_+ & \text{if } \phi(\boldsymbol{x}) > 0 \end{cases}$$

- The constant areas separated by a contour

$$\Gamma = \{\boldsymbol{x} | \phi(\boldsymbol{x}) = 0\}$$

  Γ should be as short as possible.



The Chan-Vese Model I.

Segment gray scale image
$I : D \to [0, 1]$.

Model:

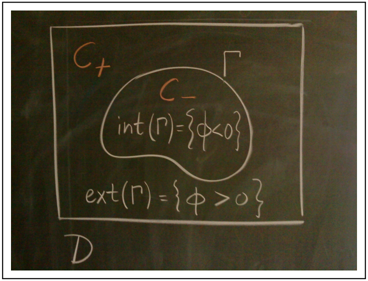
- Piecewise constant image

$$I_{\text{model}}(\boldsymbol{x}) = \begin{cases} c_- & \text{if } \phi(\boldsymbol{x}) < 0 \\ c_+ & \text{if } \phi(\boldsymbol{x}) > 0 \end{cases}$$

- The constant areas separated by a contour

$$\Gamma = \{\boldsymbol{x} | \phi(\boldsymbol{x}) = 0\}$$

  Γ should be as short as possible.



The Chan-Vese Model I.

Segment gray scale image
$I : D \to [0,1]$.

Model:

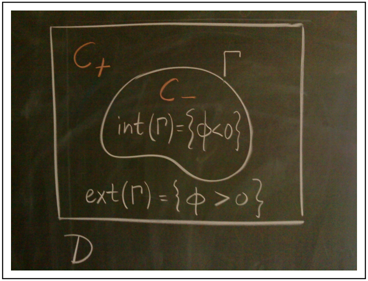
- Piecewise constant image

$$I_{\text{model}}(\boldsymbol{x}) = \begin{cases} c_- & \text{if } \phi(\boldsymbol{x}) < 0 \\ c_+ & \text{if } \phi(\boldsymbol{x}) > 0 \end{cases}$$

- The constant areas separated by a contour

$$\Gamma = \{\boldsymbol{x} | \phi(\boldsymbol{x}) = 0\}$$

Γ should be as short as possible.



The Chan-Vese Model *II*.

Given a gray scale image $I : D \to [0,1]$ that we want to segment.

The segmentation is described by the contour Γ and the gray levels c_-, c_+ which minimize the following functional:

$$E(\Gamma, c_-, c_+) = \alpha \int_\Gamma d\sigma + \int_{\mathrm{int}(\Gamma)} (I(x) - c_-)^2\, dx + \int_{\mathrm{ext}(\Gamma)} (I(x) - c_+)^2\, dx$$

where $I = I(x)$ denotes the grayscale image.

That is, we consider the variational problem:

$$(\Gamma^*, c_-^*, c_+^*) = \mathrm{argmin}_{\Gamma, c_-, c_+} E_{\mathrm{CV}}(\Gamma, c_-, c_+).$$

Structure of E: Again $E = E_{\mathrm{reg}} + E_{\mathrm{data}}$. Here we have

- E_{reg} = curve length – the regularizing term.
- E_{data} = squared distance model image/observed image — the fidelity term.

The Chan-Vese Model *II*.

Given a gray scale image $I : D \to [0,1]$ that we want to segment.

The segmentation is described by the contour Γ and the gray levels c_-, c_+ which minimize the following functional:

$$E(\Gamma, c_-, c_+) = \alpha \int_\Gamma d\sigma + \int_{\mathrm{int}(\Gamma)} (I(x) - c_-)^2\,dx + \int_{\mathrm{ext}(\Gamma)} (I(x) - c_+)^2\,dx$$

where $I = I(x)$ denotes the grayscale image.

That is, we consider the variational problem:

$$(\Gamma^*, c_-^*, c_+^*) = \mathrm{argmin}_{\Gamma, c_-, c_+} E_{\mathrm{CV}}(\Gamma, c_-, c_+).$$

Structure of E: Again $E = E_{\mathrm{reg}} + E_{\mathrm{data}}$. Here we have

- E_{reg} = curve length – the regularizing term.
- E_{data} = squared distance model image/observed image — the fidelity term.

The Chan-Vese Model *II*.

Given a gray scale image $I : D \to [0,1]$ that we want to segment.

The segmentation is described by the contour Γ and the gray levels c_-, c_+ which minimize the following functional:

$$E(\Gamma, c_-, c_+) = \alpha \int_\Gamma d\sigma + \int_{\mathrm{int}(\Gamma)} (I(x) - c_-)^2\, dx + \int_{\mathrm{ext}(\Gamma)} (I(x) - c_+)^2\, dx$$

where $I = I(x)$ denotes the grayscale image.

That is, we consider the variational problem:

$$(\Gamma^*, c_-^*, c_+^*) = \mathrm{argmin}_{\Gamma, c_-, c_+} E_{\mathrm{CV}}(\Gamma, c_-, c_+).$$

Structure of E: Again $E = E_{\mathrm{reg}} + E_{\mathrm{data}}$. Here we have

- E_{reg} = curve length – the regularizing term.
- E_{data} = squared distance model image/observed image — the fidelity term.

The Chan-Vese Model *II*.

Given a gray scale image $I : D \to [0,1]$ that we want to segment.

The segmentation is described by the contour Γ and the gray levels c_-, c_+ which minimize the following functional:

$$E(\Gamma, c_-, c_+) = \alpha \int_\Gamma d\sigma + \int_{\mathrm{int}(\Gamma)} (I(x) - c_-)^2\, dx + \int_{\mathrm{ext}(\Gamma)} (I(x) - c_+)^2\, dx$$

where $I = I(x)$ denotes the grayscale image.

That is, we consider the variational problem:

$$(\Gamma^*, c_-^*, c_+^*) = \mathrm{argmin}_{\Gamma, c_-, c_+} E_{\mathrm{CV}}(\Gamma, c_-, c_+).$$

Structure of E: Again $E = E_{\mathrm{reg}} + E_{\mathrm{data}}$. Here we have

- E_{reg} = curve length – the regularizing term.
- E_{data} = squared distance model image/observed image — the fidelity term.

The Chan-Vese Model *II*.

Given a gray scale image $I : D \to [0,1]$ that we want to segment.

The segmentation is described by the contour Γ and the gray levels c_-, c_+ which minimize the following functional:

$$E(\Gamma, c_-, c_+) = \alpha \int_\Gamma d\sigma + \int_{\text{int}(\Gamma)} (I(x) - c_-)^2 \, dx + \int_{\text{ext}(\Gamma)} (I(x) - c_+)^2 \, dx$$

where $I = I(x)$ denotes the grayscale image.

That is, we consider the variational problem:

$$(\Gamma^*, c_-^*, c_+^*) = \text{argmin}_{\Gamma, c_-, c_+} E_{\text{CV}}(\Gamma, c_-, c_+).$$

Structure of E: Again $E = E_{\text{reg}} + E_{\text{data}}$. Here we have

- E_{reg} = curve length – the regularizing term.
- E_{data} = squared distance model image/observed image — the fidelity term.

The Chan-Vese Model *II*.

Given a gray scale image $I : D \to [0,1]$ that we want to segment.

The segmentation is described by the contour Γ and the gray levels c_-, c_+ which minimize the following functional:

$$E(\Gamma, c_-, c_+) = \alpha \int_\Gamma d\sigma + \int_{\mathrm{int}(\Gamma)} (I(x) - c_-)^2 \, dx + \int_{\mathrm{ext}(\Gamma)} (I(x) - c_+)^2 \, dx$$

where $I = I(x)$ denotes the grayscale image.

That is, we consider the variational problem:

$$(\Gamma^*, c_-^*, c_+^*) = \mathrm{argmin}_{\Gamma, c_-, c_+} E_{\mathrm{CV}}(\Gamma, c_-, c_+).$$

Structure of E: Again $E = E_{\mathrm{reg}} + E_{\mathrm{data}}$. Here we have

- E_{reg} = curve length – the regularizing term.
- E_{data} = squared distance model image/observed image — the fidelity term.

Computing the Optimal Gray Levels

If the contour Γ is fixed, then the optimal gray level inside the contour must satisfy

$$0 = \frac{\partial}{\partial c_-} E(\Gamma, c_-, c_+) = -2 \int_{\mathrm{int}(\Gamma)} (I(x) - c_-)\, dx$$

Hence

$$c_- = \frac{\int_{\mathrm{int}(\Gamma)} I(x)\, dx}{\int_{\mathrm{int}(\Gamma)} dx} \quad \text{similarly} \quad c_+ = \frac{\int_{\mathrm{ext}(\Gamma)} I(x)\, dx}{\int_{\mathrm{ext}(\Gamma)} dx}$$

That is, the mean gray levels inside resp. outside the contour Γ.

Computing the Optimal Gray Levels

If the contour Γ is fixed, then the optimal gray level inside the contour must satisfy

$$0 = \frac{\partial}{\partial c_-} E(\Gamma, c_-, c_+) = -2 \int_{\mathrm{int}(\Gamma)} (I(x) - c_-)\, dx$$

Hence

$$c_- = \frac{\int_{\mathrm{int}(\Gamma)} I(x)\, dx}{\int_{\mathrm{int}(\Gamma)} dx} \quad \text{similarly} \quad c_+ = \frac{\int_{\mathrm{ext}(\Gamma)} I(x)\, dx}{\int_{\mathrm{ext}(\Gamma)} dx}$$

That is, the mean gray levels inside resp. outside the contour Γ.

Computing the Optimal Contour: Gradient Descent

The gradient descent search for the optimal Γ is the gradient descent equation:

$$\frac{d}{dt}\Gamma(t) = -\nabla E(\Gamma(t)), \quad \Gamma(0) = \Gamma_0 \text{ given.} \tag{$*$}$$

Here $\nabla E(\Gamma)$ is the (L^2-)shape gradient of E at Γ.

Since the normal velocity is $d\Gamma(t)/dt = -(\partial\phi/\partial t)/|\nabla\phi|$ Eq. $(*)$ becomes

$$-\frac{\partial\phi/\partial t}{|\nabla\phi|} = -\nabla E(\phi)$$

or

$$\frac{\partial\phi}{\partial t} - \nabla E(\phi)|\nabla\phi| = 0, \quad \phi(\boldsymbol{x}, 0) = \phi_0(\boldsymbol{x}).$$

This PDE is known as the level set equation.

Computing the Optimal Contour: Gradient Descent

The gradient descent search for the optimal Γ is the gradient descent equation:

$$\frac{d}{dt}\Gamma(t) = -\nabla E(\Gamma(t)), \quad \Gamma(0) = \Gamma_0 \text{ given.} \qquad (*)$$

Here $\nabla E(\Gamma)$ is the (L^2-)shape gradient of E at Γ.

Since the normal velocity is $d\Gamma(t)/dt = -(\partial\phi/\partial t)/|\nabla\phi|$ Eq. $(*)$ becomes

$$-\frac{\partial\phi/\partial t}{|\nabla\phi|} = -\nabla E(\phi)$$

or

$$\frac{\partial\phi}{\partial t} - \nabla E(\phi)|\nabla\phi| = 0, \quad \phi(\boldsymbol{x}, 0) = \phi_0(\boldsymbol{x}).$$

This PDE is known as the level set equation.

The Shape Gradient and The Level Set Eq.

The shape gradient of the Chan-Vese functional is

$$\nabla E(\Gamma) = \alpha\kappa + (I - c_-)^2 - (I - c_+)^2$$

This result can be obtained in several ways.

The level set equation for the Chan-Vese Model is:

$$\frac{\partial \phi}{\partial t} = \left[\alpha\kappa + (I - c_-)^2 - (I - c_+)^2\right] |\nabla \phi|$$

where $\phi(\boldsymbol{x}, 0) = \phi_0(\boldsymbol{x})$ and

$$c_- = \frac{\int (1 - H(\phi)) I \, dx}{\int (1 - H(\phi)) \, dx} \quad \text{and} \quad c_+ = \frac{\int H(\phi) I \, dx}{\int H(\phi) \, dx}$$

$H(z)$ denotes the Heaviside function of $\mathbf{R}$.

The Shape Gradient and The Level Set Eq.

The shape gradient of the Chan-Vese functional is

$$\nabla E(\Gamma) = \alpha\kappa + (I - c_-)^2 - (I - c_+)^2$$

This result can be obtained in several ways.
The level set equation for the Chan-Vese Model is:

$$\frac{\partial \phi}{\partial t} = \left[\alpha\kappa + (I - c_-)^2 - (I - c_+)^2\right] |\nabla \phi|$$

where $\phi(\boldsymbol{x}, 0) = \phi_0(\boldsymbol{x})$ and

$$c_- = \frac{\int (1 - H(\phi)) I \, dx}{\int (1 - H(\phi)) \, dx} \quad \text{and} \quad c_+ = \frac{\int H(\phi) I \, dx}{\int H(\phi) \, dx}$$

$H(z)$ denotes the Heaviside function of $\mathbf{R}$.

Examples: Segmentation in on and two pieces

xyz
Movie: Segmentation of a banana.

xyz
Movie: Split and merge – segmentation of two bananas.

Thank you for your attention!