Threshold Dynamics

with Applications to Image Processing

Louis Ly James Murphy Max Stolarski

University of Texas at Austin

November 11, 2016

Multiphase Mean Curvature Flow

Investigate algorithms for simulating the mean curvature flow of networks of interfaces under arbitrary surface tensions.

Setup

A domain D is partitioned into N closed sets Σ_i called *phases*. We want to develop algorithms for simulating the L^2-gradient flow of the energy

$$E(\Sigma_1, ..., \Sigma_N) = \sum_{i,j=1}^{N} \sigma_{i,j} \text{Area}(\partial\Sigma_i \cap \partial\Sigma_j)$$

Here, $\sigma_{i,j}$ are given *surface tensions* that satisfy

- $\sigma_{i,j} = \sigma_{j,i}$
- $\sigma_{i,i} = 0$
- $\sigma_{i,j} > 0 \quad \text{for } i \neq j$

Results/Constraints

1. At "smooth" points, $\partial\Sigma_i \cap \partial\Sigma_j$ moves by mean curvature flow.
2. At triple junctions $p \in \partial\Sigma_i \cap \partial\Sigma_j \cap \partial\Sigma_k$, the *Herring angle condition* holds:

$$\sigma_{i,j} n_{i,j}(p) + \sigma_{j,k} n_{j,k}(p) + \sigma_{k,l} n_{k,l}(p) = 0$$

where $n_{i,j}$ denotes the unit normal to $\partial\Sigma_i \cap \partial\Sigma_j$ (pointing from Σ_i into Σ_j). Equivalently, in terms of the opening angles $\theta_1, \theta_2, \theta_3$,

$$\frac{\sin(\theta_1)}{\sigma_{2,3}} = \frac{\sin(\theta_2)}{\sigma_{1,3}} = \frac{\sin(\theta_3)}{\sigma_{1,2}}$$

MBO Algorithm

Let $G_{\delta t}(x)$ denote the heat kernel

$$G_{\delta t}(x) = \frac{1}{(4\pi\delta t)^{d/2}} e^{-\|x\|^2/4\delta t}$$

Given the partition $\Sigma_1^k, ..., \Sigma_N^k$ at time $t = (\delta t)k$, obtain the partition Σ_i^{k+1} at the next time step as follows:

Step 1. (Convolution)

$$\phi_i^k = G_{\delta t} * \mathbf{1}_{\Sigma_i^k}$$

Step 2. (Thresholding)

$$\Sigma_i^{k+1} = \{x \in D | \phi_i^k(x) > \phi_j^k(x) \text{ for all } j \neq i\}$$

MBO Algorithm

Let $G_{\delta t}(x)$ denote the heat kernel

$$G_{\delta t}(x) = \frac{1}{(4\pi\delta t)^{d/2}} e^{-\|x\|^2/4\delta t}$$

Given the partition $\Sigma_1^k, ..., \Sigma_N^k$ at time $t = (\delta t)k$, obtain the partition Σ_i^{k+1} at the next time step as follows:

Step 1. (Convolution)

$$\phi_i^k = G_{\delta t} * \mathbf{1}_{\Sigma_i^k}$$

Step 2. (Thresholding)

$$\Sigma_i^{k+1} = \{x \in D | \phi_i^k(x) > \phi_j^k(x) \text{ for all } j \neq i\}$$

Problem: This method only ensures the Herring angle condition for surface tensions $\sigma_{i,j} = 1$ for all $i \neq j$.

Approximate Energy

$$Area(\partial\Sigma_i \cap \partial\Sigma_j) \approx \frac{1}{\sqrt{\delta t}} \int \mathbf{1}_i G_{\delta t} * \mathbf{1}_j dx$$

$$\Longrightarrow \; E(\Sigma_1, ..., \Sigma_N) \approx E_{\delta t}(\Sigma_1, ..., \Sigma_N) = \sum_{i,j=1}^{N} \sigma_{i,j} \frac{1}{\sqrt{\delta t}} \int \mathbf{1}_{\Sigma_i} G_{\delta t} * \mathbf{1}_{\Sigma_j} dx$$

More generally, for $\mathbf{u} = (u_1, ..., u_N) : D \to [0,1]^N$ such that $\sum_j u_j \equiv 1$, let

$$E_{\delta t}(\mathbf{u}) = \sum_{i,j=1}^{N} \sigma_{i,j} \frac{1}{\sqrt{\delta t}} \int u_i G_{\delta t} * u_j dx$$

The linearization of $E_{\delta t}$ at $\mathbf{u}^*$ is denoted

$$\mathcal{L}_{E_{\delta t}}(\mathbf{u}^*, \mathbf{u}) = \frac{2}{\sqrt{\delta t}} \sum_{i=1}^{N} \int u_i \left(\sum_{j \neq i} \sigma_{i,j} G_{\delta t} * u_j^* \right) dx$$

EO Algorithm

The EO-Algorithm minimizes the linearization of the approximate energy to obtain a threshold dynamics scheme for multiphase mean curvature motion.
Namely, given the partition $\Sigma_1^k, ..., \Sigma_N^k$ at time $t = (\delta t)k$, obtain the partition at the next time step as follows:

Step 1. (Convolution)

$$\phi_i^k = \sum_{j=1}^{N} \sigma_{i,j} G_{\delta t} * \mathbf{1}_{\Sigma_j^k}$$

Step 2. (Thresholding)

$$\Sigma_i^{k+1} = \{x \in D | \phi_i^k(x) < \phi_j^k(x) \text{ for all } j \neq i\}$$

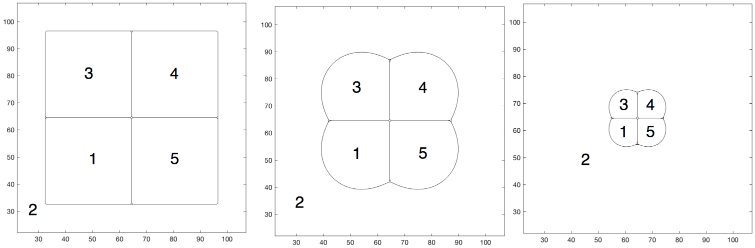
Stability

We say the surface tensions $\sigma_{i,j}$ are *conditionally negative semi-definite* if

$$\sum_{i=1}^{N} \xi_i = 0 \implies \sum_{i,j=1}^{N} \sigma_{i,j}\xi_i\xi_j \leq 0$$

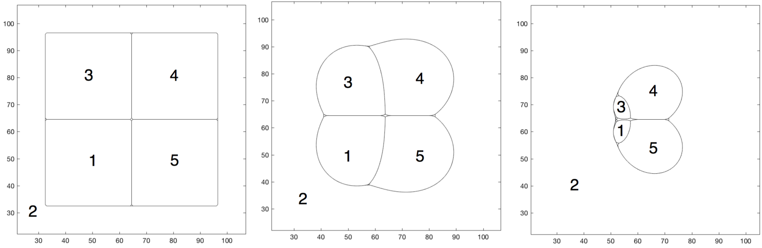
For such surface tensions, the EO-Algorithm is unconditionally gradient stable, i.e. each iteration decreases the approximate energy $E_{\delta t}$.
In applications, many surface tensions are conditionally negative semi-definite (e.g. Read-Shockley type surface tensions).

Demo



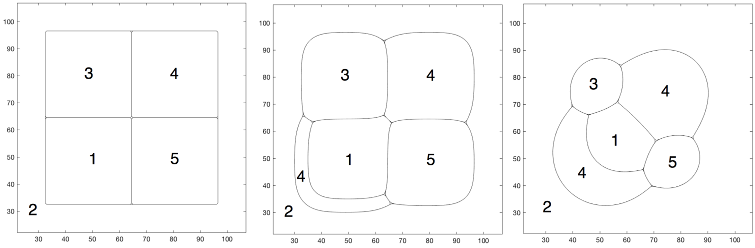
$$(\sigma_{i,j}) = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{pmatrix}$$

Different surface tensions $\sigma_{i,j} \rightsquigarrow$ different dynamics



$$(\sigma_{i,j}) = \begin{pmatrix} 0 & 1.5 & 1.5 & 1 & 1 \\ 1.5 & 0 & 1.5 & 1 & 1 \\ 1.5 & 1.5 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1.5 \\ 1 & 1 & 1 & 1.5 & 0 \end{pmatrix}$$

Nucleation/Wetting



$$(\sigma_{i,j}) \sim \begin{pmatrix} 0 & 1.5 & 1 & 0.5 & 1 \\ 1.5 & 0 & 1 & 0.5 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 0.5 & 0.5 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{pmatrix}$$

Image Segmentation

- Given an input image, how to determine the foreground vs. background?
- More generally, how to separate an image into n regions?

Mumford-Shah Model

Energy minimization approach:

$$E_{\mathrm{MS}}(u, \Gamma) = \int_{\Omega \setminus \Gamma} \|\nabla u\|^2 \, dx + \mu \cdot \mathrm{Len}(\Gamma) + \lambda \int_{\Omega} (u - f)^2 \, dx$$

where

- $\mu, \lambda > 0$
- $f : \Omega \to \mathbb{R}$ input image
- $\Gamma \subseteq \Omega$ closed set, union of finite number of curves, the boundaries of regions
- u piecewise smooth approximation of f

Mumford-Shah Model

Energy minimization approach:

$$E_{\mathrm{MS}}(u,\Gamma) = \int_{\Omega\setminus\Gamma} \|\nabla u\|^2\, dx + \mu \cdot \mathrm{Len}(\Gamma) + \lambda \int_{\Omega} (u-f)^2\, dx$$

where

- $\mu, \lambda > 0$
- $f : \Omega \to \mathbb{R}$ input image
- $\Gamma \subseteq \Omega$ closed set, union of finite number of curves, the boundaries of regions
- u piecewise smooth approximation of f

Terms penalize for lack of smoothness in regions, length of boundaries of regions, and lack of image fidelity.

Mumford-Shah Model (cont.)

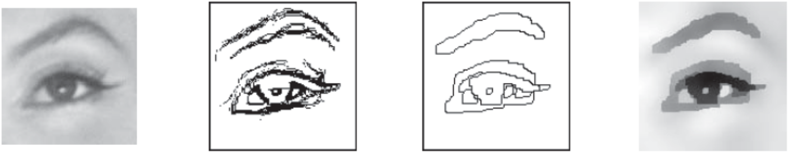


Image credit [4], left to right:

- original
- high contrast areas ($\|\nabla f\|^2$ large)
- Γ boundaries shown
- smooth approximation u shown

Mumford-Shah Model (cont.)

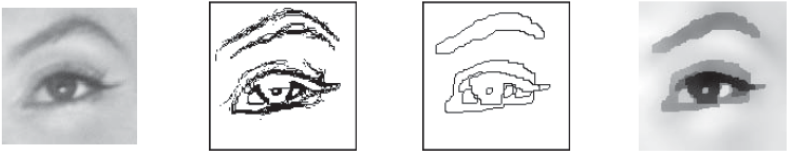


Image credit [4], left to right:

- original
- high contrast areas ($\|\nabla f\|^2$ large)
- Γ boundaries shown
- smooth approximation u shown

Works well, but numerically too difficult for practical applications.

Chan-Vese Model

Allow input image $f : \Omega \to \mathbb{R}^d$, e.g. $f = (R, G, B, A)$.
Simplify Mumford-Shah by taking u piecewise constant,

$$E_{\mathrm{CV}}(\Omega_1, \ldots, \Omega_n, C_1, \ldots, C_n) = \sum_{i=1}^{n} \int_{\Omega_i} \|C_i - f\|^2 \, dx + \lambda \cdot \mathrm{Len}(\partial \Omega_i)$$

where

- $\lambda > 0$
- $\Omega_1, \ldots, \Omega_n$ partition Ω
- $u = C_i$ on Ω_i.

Chan-Vese Model (cont.)

Rewrite with $u_i = 1_{\Omega_i}, g_i = \| C_i - f \|^2$,

$$E_{\mathrm{CV}}(\Omega_1, \ldots, \Omega_n, C_1, \ldots, C_n) = \sum_{i=1}^{n} \int_{\Omega} u_i g_i \, dx + \lambda \cdot \mathrm{Len}(\partial \Omega_i)$$

Chan-Vese Model (cont.)

Rewrite with $u_i = 1_{\Omega_i}, g_i = \| C_i - f \|^2$,

$$E_{\mathrm{CV}}(\Omega_1, \ldots, \Omega_n, C_1, \ldots, C_n) = \sum_{i=1}^{n} \int_{\Omega} u_i g_i \, dx + \lambda \cdot \mathrm{Len}(\partial \Omega_i)$$

Best (minimizing) choice for C_i is

$$C_i = \frac{\int_{\Omega} u_i f \, dx}{\int_{\Omega} u_i \, dx}$$

Chan-Vese Model (cont.)

Rewrite with $u_i = 1_{\Omega_i}, g_i = \| C_i - f \|^2$,

$$E_{\mathrm{CV}}(\Omega_1, \ldots, \Omega_n, C_1, \ldots, C_n) = \sum_{i=1}^{n} \int_{\Omega} u_i g_i \, dx + \lambda \cdot \mathrm{Len}(\partial \Omega_i)$$

Best (minimizing) choice for C_i is

$$C_i = \frac{\int_{\Omega} u_i f \, dx}{\int_{\Omega} u_i \, dx}$$

But still numerically too difficult to find best $\Omega_1, \ldots, \Omega_n$.

Approximate Chan-Vese Model

Two simplifications

- Allow $0 \leq u_i \leq 1$ and $\sum_{i=1}^{n} u_i = 1$ to make problem convex

Approximate Chan-Vese Model

Two simplifications

- Allow $0 \leq u_i \leq 1$ and $\sum_{i=1}^{n} u_i = 1$ to make problem convex
- $\mathrm{Len}(\partial\Omega_i) \approx \sum_{j=1, j\neq i}^{n} \sqrt{\frac{\pi}{\delta t}} \int_{\Omega} u_i G_{\delta t} * u_j \, dx$, $G_{\delta t}$ heat kernel
  - $G_{\delta t} = \frac{1}{4\pi\delta t} \exp\left(-\frac{\|x\|^2}{4\delta t}\right)$
  - $\delta t \ll 1$

Approximate Chan-Vese Model

Two simplifications

- Allow $0 \leq u_i \leq 1$ and $\sum_{i=1}^n u_i = 1$ to make problem convex
- $\mathrm{Len}(\partial\Omega_i) \approx \sum_{j=1, j\neq i}^n \sqrt{\frac{\pi}{\delta t}} \int_\Omega u_i G_{\delta t} * u_j \, dx$, $G_{\delta t}$ heat kernel
  - $G_{\delta t} = \frac{1}{4\pi\delta t} \exp\left(-\frac{\|x\|^2}{4\delta t}\right)$
  - $\delta t \ll 1$

Problem becomes minimizing

$$\mathcal{E}^{\delta t}(u_1, \ldots, u_n) = \sum_{i=1}^n \int_\Omega \left(u_i g_i + \lambda \sum_{j=1, j\neq i}^n \sqrt{\frac{\pi}{\delta t}} u_i G_{\delta t} * u_j \right) dx$$

Approximate Chan-Vese Model

Two simplifications

- Allow $0 \leq u_i \leq 1$ and $\sum_{i=1}^{n} u_i = 1$ to make problem convex
- $\mathrm{Len}(\partial\Omega_i) \approx \sum_{j=1, j\neq i}^{n} \sqrt{\frac{\pi}{\delta t}} \int_\Omega u_i G_{\delta t} * u_j \, dx$, $G_{\delta t}$ heat kernel
  - $G_{\delta t} = \frac{1}{4\pi\delta t} \exp\left(-\frac{\|x\|^2}{4\delta t}\right)$
  - $\delta t \ll 1$

Problem becomes minimizing

$$\mathcal{E}^{\delta t}(u_1, \ldots, u_n) = \sum_{i=1}^{n} \int_\Omega \left(u_i g_i + \lambda \sum_{j=1, j\neq i}^{n} \sqrt{\frac{\pi}{\delta t}} u_i G_{\delta t} * u_j \right) dx$$

It turns out minimum will have $u_i \in \{0, 1\}$ for all i.

Approximate Chan-Vese Model Iteration Scheme

Given $u^k = (u_1^k, \ldots, u_n^k)$ some iteration, linearize

$$\mathcal{E}^{\delta t}(u_1, \ldots, u_n) \approx \mathcal{E}^{\delta t}(u_1^k, \ldots, u_n^k) + \mathcal{L}(u_1 - u_1^k, \ldots, u_n - u_n^k, u_1^k, \ldots, u_n^k)$$

Approximate Chan-Vese Model Iteration Scheme

Given $u^k = (u_1^k, \ldots, u_n^k)$ some iteration, linearize

$$\mathcal{E}^{\delta t}(u_1, \ldots, u_n) \approx \mathcal{E}^{\delta t}(u_1^k, \ldots, u_n^k) + \mathcal{L}(u_1 - u_1^k, \ldots, u_n - u_n^k, u_1^k, \ldots, u_n^k)$$

where

- $\mathcal{L}(u_1, \ldots, u_n, u_1^k, \ldots, u_n^k) = \sum_{i=1}^n \int_\Omega u_i \phi_i^k \, dx$

Approximate Chan-Vese Model Iteration Scheme

Given $u^k = (u_1^k, \ldots, u_n^k)$ some iteration, linearize

$$\mathcal{E}^{\delta t}(u_1, \ldots, u_n) \approx \mathcal{E}^{\delta t}(u_1^k, \ldots, u_n^k) + \mathcal{L}(u_1 - u_1^k, \ldots, u_n - u_n^k, u_1^k, \ldots, u_n^k)$$

where

- $\mathcal{L}(u_1, \ldots, u_n, u_1^k, \ldots, u_n^k) = \sum_{i=1}^n \int_\Omega u_i \phi_i^k \, dx$
- $\phi_i^k = g_i^k + \frac{2\lambda\sqrt{\pi}}{\sqrt{\delta t}}(1 - G_{\delta t} * u_j^k)$
- $g_i^k = \left\| C_i^k - f \right\|^2$
- $C_i^k = \frac{\int_\Omega u_i^k f \, dx}{\int_\Omega u_i^k \, dx}$

Approximate Chan-Vese Model Iteration Scheme

Given $u^k = (u_1^k, \ldots, u_n^k)$ some iteration, linearize

$$\mathcal{E}^{\delta t}(u_1, \ldots, u_n) \approx \mathcal{E}^{\delta t}(u_1^k, \ldots, u_n^k) + \mathcal{L}(u_1 - u_1^k, \ldots, u_n - u_n^k, u_1^k, \ldots, u_n^k)$$

where

- $\mathcal{L}(u_1, \ldots, u_n, u_1^k, \ldots, u_n^k) = \sum_{i=1}^n \int_\Omega u_i \phi_i^k \, dx$
- $\phi_i^k = g_i^k + \frac{2\lambda\sqrt{\pi}}{\sqrt{\delta t}}(1 - G_{\delta t} * u_j^k)$
- $g_i^k = \left\| C_i^k - f \right\|^2$
- $C_i^k = \frac{\int_\Omega u_i^k f \, dx}{\int_\Omega u_i^k \, dx}$

Minimize $\mathcal{L}$ to determine u^{k+1}.

Approximate Chan-Vese Model Iteration Scheme (cont.)

$$\mathcal{L}(u_1, \ldots, u_n, u_1^k, \ldots, u_n^k) = \sum_{i=1}^{n} \int_\Omega u_i \phi_i^k \, dx$$

Minimum attained at

$$u_i^{k+1} = \begin{cases} 1 & \phi_i^k(x) = \min_j \phi_j^k(x) \\ 0 & \text{else} \end{cases}$$

Minimum can be computed in parallel for each x.
This iteration scheme has decreasing energy.

Threshold Dynamics Formulation

Iteration scheme suggests algorithm

Step 0. Given initial partition $\Omega_1^0, \dots, \Omega_n^0$ of Ω define corresponding $u_i^0 = 1_{\Omega_i^0}$. Pick tolerance $\tau > 0$.

Threshold Dynamics Formulation

Iteration scheme suggests algorithm

Step 0. Given initial partition $\Omega_1^0, \ldots, \Omega_n^0$ of Ω define corresponding $u_i^0 = 1_{\Omega_i^0}$. Pick tolerance $\tau > 0$.

Step 1. Given $u^k = (u_1^k, \ldots, u_n^k)$, compute C_i^k, g_i^k, ϕ_i^k as before. Use FFT to compute convolutions.

Threshold Dynamics Formulation

Iteration scheme suggests algorithm

Step 0. Given initial partition $\Omega_1^0, \ldots, \Omega_n^0$ of Ω define corresponding $u_i^0 = 1_{\Omega_i^0}$. Pick tolerance $\tau > 0$.

Step 1. Given $u^k = (u_1^k, \ldots, u_n^k)$, compute C_i^k, g_i^k, ϕ_i^k as before. Use FFT to compute convolutions.

Step 2. Thresholding: let $\Omega_i^{k+1} = \{x : \phi_i^k(x) < \min_{j \neq i} \phi_j^k(x)\}$ and define corresponding $u_i^{k+1} = 1_{\Omega_i^{k+1}}$.

Threshold Dynamics Formulation

Iteration scheme suggests algorithm

Step 0. Given initial partition $\Omega_1^0, \ldots, \Omega_n^0$ of Ω define corresponding $u_i^0 = 1_{\Omega_i^0}$. Pick tolerance $\tau > 0$.

Step 1. Given $u^k = (u_1^k, \ldots, u_n^k)$, compute C_i^k, g_i^k, ϕ_i^k as before. Use FFT to compute convolutions.

Step 2. Thresholding: let $\Omega_i^{k+1} = \{x : \phi_i^k(x) < \min_{j \neq i} \phi_j^k(x)\}$ and define corresponding $u_i^{k+1} = 1_{\Omega_i^{k+1}}$.

Step 3. Compute normalized difference of successive iterations

$$e^{k+1} = \frac{1}{|\Omega|} \int_\Omega \sum_{i=1}^n \left\| u_i^{k+1} - u_i^k \right\|^2 dx$$

If $e^{k+1} \leq \tau$ stop, else goto step 1.

Threshold Dynamics Formulation (cont.)

$d = 1, n = 2, \delta t = 0.03$ and $\lambda = 0.01$. Convergence after 15 iterations in 0.1188 seconds, image credit [1].

Demo

References

[1] D. Wang, H. Li, X. Wei, and X. Wang
An Efficient Iterative Thresholding Method For Image Segmentation
arXiv preprint arXiv:1608.01431

[2] S. Esedoḡlu, F. Otto
Threshold Dynamics for Networks with Arbitrary Surface Tensions
Communications on Pure and Applied Mathematics

[3] B. Merriman, J.K. Bence, S. J. Osher,
Diffusion Generated Motion by Mean Curvature
Proceedings of the Computational Crystal Growers Workshop

[4] Wikipedia
Mumford-Shah functional
en.wikipedia.org/wiki/Mumford-Shah_functional