Lecture 21:

How to minimise $J(f)$

We consider the problem of finding f that minimizes $J(f)$.

In the discrete case, J depends on $f(x,y)$ for $\begin{matrix} x = 1, 2, \ldots, N \\ y = 1, 2, \ldots, N \end{matrix}$.

Consider a time-dependent image $f(x,y;\underbrace{t}_{\text{time}})$. Assuming that $f(x,y;t)$ satisfies:

$$\frac{df(\cdot,\cdot;t)}{dt} = -\nabla J(f(\cdot,\cdot;t)) \qquad (**)$$

We can show that $J(f(\;\;;t))$ decreases as t increases.

Note that:

$$\frac{d}{dt} J(f(\cdot,\cdot;t)) = \nabla J(f(\cdot,\cdot;t)) \cdot \frac{d}{dt} f(\cdot,\cdot;t) = -\nabla J(f(\cdot,\cdot;t)) \cdot \nabla J(f(\cdot,\cdot;t))$$

$$= -\left|\nabla J(f(\cdot,\cdot;t))\right|^2 \leq 0.$$

$\therefore$ $J(f(\cdot,\cdot;t))$ is decreasing as t increases!!

In the discrete case,

$$\frac{f^{n+1} - f^n}{\Delta t} = -\nabla J(f^n)$$

(Gradient descent algorithm)

For the ROF model:

$$
\begin{aligned}
&\frac{f^{n+1}(x,y) - f^n(x,y)}{\Delta t} \\
&= -(f^n(x,y) - g(x,y)) + \lambda \frac{f^n(x+1,y) - f^n(x,y)}{\sqrt{(f^n(x+1,y) - f^n(x,y))^2 + (f^n(x,y+1) - f^n(x,y))^2}} \\
&- \lambda \frac{f^n(x,y) - f^n(x-1,y)}{\sqrt{(f^n(x,y) - f^n(x-1,y))^2 + (f^n(x-1,y+1) - f^n(x-1,y))^2}} \\
&+ \lambda \frac{f^n(x,y+1) - f^n(x,y)}{\sqrt{(f^n(x+1,y) - f^n(x,y))^2 + (f^n(x,y+1) - f^n(x,y))^2}} \\
&- \lambda \frac{f^n(x,y) - f^n(x,y-1)}{\sqrt{(f^n(x+1,y-1) - f^n(x,y-1))^2 + (f^n(x,y) - f^n(x,y-1))^2}}
\end{aligned}
$$

Discretization of ∇J

(Gradient descent algorithm for ROF)

In the continuous case, consider:

$$E(f) = \int_\Omega (f-g)^2 \, dx\, dy + \lambda \int |\nabla f| \, dx\, dy$$

We want to find a sequence $f_0 = g$, $f_1, \ldots, f_n, \ldots$ such that:

$$E(f_0) \geq E(f_1) \geq \ldots \geq E(f_n) \geq E(f_{n+1}) \geq \ldots$$

Define $s(\varepsilon) \overset{\text{def}}{:=} E(\overbrace{f_n + \varepsilon v}^{f_{n+1}})$ for some suitable v. (Assuming that $\nabla f_n = 0$ on $\partial\Omega$)

For small $\varepsilon \geq 0$ by Taylor expansion,

$$\underset{E(f_{n+1})}{\underset{\parallel}{s(\varepsilon)}} = \underset{E(f_n)}{\underset{\parallel}{s(0)}} + s'(0)\,\underset{0}{\underset{\vee}{\varepsilon}} + \underbrace{O(\varepsilon^2)}_{\text{neglible}}$$

If $s'(0) \leq 0$, then $E(f_{n+1}) \leq E(f_n)$

Now, $$\frac{d}{d\varepsilon}\Big|_{\varepsilon=0} s(\varepsilon) = \frac{d}{d\varepsilon}\Big|_{\varepsilon=0}\left[\int_\Omega (f_n + \varepsilon v - g)^2 \, dx\, dy + \lambda \int_\Omega \underset{\sqrt{(\nabla f_n + \varepsilon \nabla v)\cdot(\nabla f_n + \varepsilon \nabla v)}}{\underset{\parallel}{|\nabla f_n + \varepsilon \nabla v|}} \, dx\, dy\right]$$

$$\therefore\ S'(0) = \int_\Omega (f_n - g)\, v \, dx\,dy + \lambda \int_\Omega \frac{\nabla f_n \cdot \nabla v}{\underbrace{\sqrt{\nabla f_n \cdot \nabla f_n}}_{|\nabla f_n|}} \, dx\,dy$$

$$= \int_\Omega (f_n - g)\, v \, dx\,dy - \lambda \int_\Omega \nabla \cdot \left(\frac{\nabla f_n}{|\nabla f_n|} \right) v \, dx\,dy + \cancelto{0}{\lambda \int_{\partial\Omega} \frac{\nabla f_n}{|\nabla f_n|} \cdot \vec{n} \, v \, ds}$$

$$= \int_\Omega \left[(f_n - g) - \lambda \nabla \cdot \left(\frac{\nabla f_n}{|\nabla f_n|} \right) \right] v \, dx\,dy$$

Put $v = -\left[(f_n - g) - \lambda \nabla \cdot \left(\frac{\nabla f_n}{|\nabla f_n|} \right) \right]$. Then: $S'(0) \leq 0$.

$\therefore$ Gradient descent algorithm:

$$f^{n+1} = f^n + \varepsilon \underbrace{\left[-\left((f_n - g) - \lambda \nabla \cdot \left(\frac{\nabla f_n}{|\nabla f_n|} \right) \right) \right]}_{v} \quad \text{for } n = 0, 1, 2, \ldots$$

This is called the gradient descent algorithm.

Image deblurring in the spatial domain

(Spatial) Deblurring / denoising model:

Consider: $\underset{\text{Observed}}{g} = \underset{\text{Degradation}}{h} * f + \underset{\text{noise}}{n}$

We aim to find f that minimizes:

$$J(f) = \frac{1}{2}\iint_D \left(h * f(x,y) - g(x,y) \right)^2 dx\,dy + \lambda \iint_D |\nabla f|\, dx\, dy$$

Goal: Use gradient descent method

Note that:

$$\iint_D h*f(x,y)\, g(x,y)\, dx dy = \iint_D \left[\iint_D h(\alpha,\beta)\, f(x-\alpha, y-\beta)\, d\alpha\, d\beta \right] g(x,y)\, dx\, dy$$

$$= \iint_D \left[\iint_D f(\underbrace{x-\alpha}_{X}, \underbrace{y-\beta}_{Y})\, g(x,y)\, dx dy \right] h(\alpha,\beta)\, d\alpha\, d\beta$$

Let $X = x - \alpha$, $Y = y - \beta$.

$$= \iint_D \left[\iint_D f(X,Y)\, g(X+\alpha, Y+\beta)\, dx dy \right] h(\alpha,\beta)\, d\alpha\, d\beta$$

$$= \iint_D f(X,Y) \underbrace{\left[\iint_D h(\alpha,\beta)\, g(\alpha+X, \beta+Y)\, d\alpha\, d\beta \right]}_{\tilde{H}(g)(X,Y)} dX\, dY$$

To minimize $J(f)$ using gradient descent, consider: $s(\varepsilon) = J(f + \varepsilon w)$. Then: $\left.\frac{d}{d\varepsilon}\right|_{\varepsilon=0} s(\varepsilon) = 0$.

$$\therefore \quad s'(0) = \frac{1}{2}\iint_D \left.\frac{d}{d\varepsilon}\right|_{\varepsilon=0} (h*f + \varepsilon h*w - g)^2 \, dx\,dy + \lambda \iint_D \left.\frac{d}{d\varepsilon}\right|_{\varepsilon=0} |\nabla f + \varepsilon \nabla w| \, dx\,dy$$

$$= \iint_D (h*f - g)\, h*w \, dx\,dy - \lambda \iint_D \nabla \cdot \left(\frac{1}{|\nabla f|} \nabla f\right) w \, dx\,dy$$

$$= \iint_D \tilde{H}(h*f - g)(x,y)\, w(x,y)\, dx\,dy - \lambda \iint_D \nabla \cdot \left(\frac{1}{|\nabla f|}\nabla f\right)(x,y)\; w(x,y)\, dx\,dy$$

$\therefore$ Gradient descent direction:

$$w(x,y) = -\tilde{H}(h*f - g)(x,y) + \lambda \nabla \cdot \left(\frac{1}{|\nabla f|} \nabla f\right)(x,y)$$

Algorithm:

$$\frac{f^{n+1} - f^n}{\Delta t} = -\tilde{H}(h*f^n - g) + \lambda \underbrace{\nabla \cdot \left(\frac{1}{|\nabla f^n|} \nabla f^n\right)}_{\text{Discrete discretization of partial derivatives.}}$$

Image segmentation

Basic idea of Image Segmentation:

Task 1: extract sets of points describing the boundaries / edges of objects;

Task 2: Find a binary image ("black and white") so that "white" color represents the objects.

Information from the image: ($I : \Omega \to \mathbb{R}$; Ω = image domain)

Edge detector: $V : \Omega \to \mathbb{R}$ such that $V(\vec{x})$ is small if $\vec{x}$ is on the edges of the object.

Example 1: $V(\vec{x}) = -|\nabla I(\vec{x})|$

On edges, $\nabla I(\vec{x})$ is big $\Rightarrow -|\nabla I(\vec{x})|$ is small

$V(\vec{x}) = -|\nabla I(\vec{x})| = 0$ in the interior of the object.

In the discrete case, $\frac{\partial I}{\partial x}$ and $\frac{\partial I}{\partial y}$ (hence ∇I) can be computed by linear filtering with filters: $\begin{pmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$ and $\begin{pmatrix} 0 & 0 & 0 \\ -1 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$

Example 2: $V(\vec{x}) = \dfrac{1}{1+|\nabla I(\vec{x})|}$

Segmentation models:

1. (Explicit) Parameterized curve evolution (Active contour model)

   Goal: Find a parameterized curve $\gamma : [0, 2\pi] \to \Omega$ such that it represents the boundary of the object.

2. (Implicit) Level set model:

   Find a function $\varphi : \Omega \to \mathbb{R}$ such that $\varphi(\vec{x}) > 0$ if $\vec{x}$ is inside the object and $\varphi(\vec{x}) < 0$ if $\vec{x}$ is outside the object.

$\therefore$ $\underbrace{\varphi^{-1}(\{0\}) = \{\vec{x} \in \Omega : \varphi(\vec{x}) = 0\}}_{\text{zero level set of } \varphi.}$ is a set of points on the boundary of the object.

This segmentation method is called the Level set method!

Active contour model (Kass, Witkin, Terzopoulous)

Let $I : \Omega \to \mathbb{R}$ be the image. $\subseteq \mathbb{R}^2$

Goal: Find $\gamma : [0, 2\pi] \to \Omega$, which lies on the boundary of the object.

Assume the boundary is a simple closed curve. Then: $\gamma(0) = \gamma(2\pi)$

Let $V : \Omega \to \mathbb{R}$ be the edge detector. We consider the snake model to find γ that minimizes the snake energy:

$$E_{snake}(\gamma) = \int_0^{2\pi} \underbrace{\frac{1}{2} |\gamma'(s)|^2}_{\text{enhance smoothness of } \gamma(s)} ds + \beta \int_0^{2\pi} \underbrace{V(\gamma(s))}_{\text{Find } \gamma(s) \text{ that lies on the boundary.}} ds$$

Goal: Use gradient descent algorithm to obtain γ.

Remark: $\gamma = (\varphi_1, \varphi_2)$. $\therefore$ $\gamma'(s) = (\varphi_1'(s), \varphi_2'(s))$. $\Rightarrow |\gamma'(s)|^2 = (\varphi_1'(s))^2 + (\varphi_2'(s))^2$.

Starting from γ^0 = initial curve (e.g. circle)

Iteratively look for $\gamma^1, \gamma^2, \dots, \gamma^n, \gamma^{n+1}, \dots$ such that:

$$E_{snake}(\gamma^{n+1}) \leq E_{snake}(\gamma^n).$$

Given γ^n, define $\gamma^{n+1} = \gamma^n + \varepsilon \varphi$ (Perturbation of γ^n)

with $\varphi(0) = \varphi(2\pi)$

Need to find φ such that $\displaystyle \frac{d}{d\varepsilon}\Big|_{\varepsilon=0} E_{snake}(\overbrace{\gamma^n + \varepsilon \varphi}^{\varphi^{n+1}}) < 0.$

(Then: $\displaystyle E(\gamma^{n+1}) = E(\gamma^n) + \varepsilon \left(\frac{d}{d\varepsilon}\Big|_{\varepsilon=0} E_{snake}(\gamma^n + \varepsilon\varphi) + \mathcal{O}(\varepsilon^2) \right)$)

$$\frac{d}{d\varepsilon}\Big|_{\varepsilon=0} E_{snake}(\gamma^n + \varepsilon \varphi) = \frac{d}{d\varepsilon}\Big|_{\varepsilon=0} \int_0^{2\pi} \frac{1}{2} \overbrace{\left| (\gamma^n)'(s) + \varepsilon \varphi'(s) \right|^2}^{\left((\gamma^n)'(s) + \varepsilon \varphi'(s)\right) \cdot \left((\gamma^n)'(s) + \varepsilon \varphi'(s)\right)} ds + \beta \frac{d}{d\varepsilon}\Big|_{\varepsilon=0} \int_0^{2\pi} V(\gamma^n(s) + \varepsilon \varphi(s))\, ds$$

$$= \int_0^{2\pi} (\gamma^n)'(s) \cdot \varphi'(s)\, ds + \beta \int_0^{2\pi} \nabla V(\gamma^n(s)) \cdot \varphi(s)\, ds$$

$$= -\int_0^{2\pi} (\gamma^n)''(s) \cdot \varphi(s)\, ds + \cancelto{0}{(\gamma^n)'(s)\, \varphi(s) \Big|_0^{2\pi}} + \beta \int_0^{2\pi} \nabla V(\gamma^n(s)) \cdot \varphi(s)\, ds$$

$$= \int_0^{2\pi} \left(-(\gamma^n)''(s) + \beta \nabla V(\gamma^n(s)) \right) \cdot \varphi(s)\, ds$$

In order that $\left.\frac{d}{d\varepsilon}\right|_{\varepsilon=0} E_{snake}(\gamma^{n+1}) < 0$ (decreasing), we choose:

$$\varphi(s) = (\gamma^n)''(s) - \beta \nabla V(\gamma^n(s))$$

Hence, we modify γ^n by:

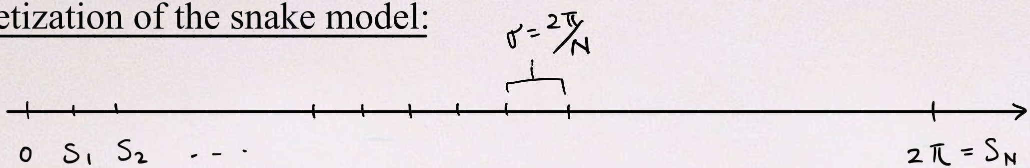
$$\gamma^{n+1}(s) = \gamma^n(s) + \varepsilon\left[(\gamma^n)''(s) - \beta \nabla V(\gamma^n(s))\right] \quad \text{for some } \varepsilon > 0.$$

$$\frac{\gamma^{n+1}(s) - \gamma^n(s)}{\varepsilon} = \underbrace{(\gamma^n)''(s) - \beta \nabla V(\gamma^n(s))}_{\text{denote it by: } -\nabla E(\gamma^n(s))}$$

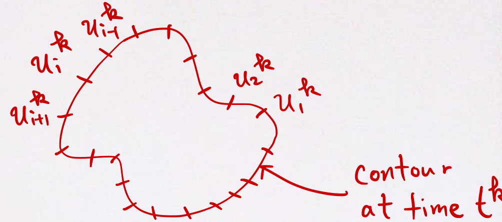
<u>Remark:</u> In the continuous setting, we aim to obtain a time-dependent contour: $\gamma_t(s) = \gamma(s;t)$ such that:

$$\frac{d}{dt}\gamma_t(s) = -\nabla E(\gamma_t(s)).$$

Discretization of the snake model:



Let N = number of discrete points in $[0, 2\pi]$



Let $\sigma = \frac{2\pi}{N}$ = step length.

$s_i = i\sigma$ for $i = 1, 2, \dots, N$

$t^k = k\tau$ for $k = 1, 2, \dots$ (τ = time step)

$u_i^k = \gamma(s_i; t^k) = \gamma(i\sigma; k\tau) = i^{th}$ node of the contour at time t^k

$$= \begin{pmatrix} (u_i^k)_x \\ (u_i^k)_y \end{pmatrix}$$

Define: $$u^k = \begin{pmatrix} (u_1^k)_x & (u_1^k)_y \\ \vdots & \\ (u_i^k)_x & (u_i^k)_y \\ \vdots & \\ (u_N^k)_x & (u_N^k)_y \end{pmatrix} = \begin{pmatrix} | & | & & | \\ u_1^k & u_2^k & \dots & u_N^k \\ | & | & & | \end{pmatrix}^T \in M_{N\times 2}(\mathbb{R})$$

u^k is called the discrete closed curve.

The discrete derivative can be approximated by:

$$\gamma_k'(\underset{\overset{\|}{i\sigma}}{s_i}) \approx \frac{u_{i+1}^k - u_i^k}{\sigma} \qquad i = 1, 2, \dots, N$$

(Here, $u_{N+1}^k = u_N^k$ and $u_0^k = u_N^k$. ($\because$ contour is closed))

Thus, the discrete snake energy can be written as:

$$E_{snake}(\underset{\overset{\cap}{M_{N\times 2}(\mathbb{R})}}{u}) = \sum_{i=1}^{N} \frac{1}{2} \left| \frac{u_{i+1} - u_i}{\sigma} \right|^2 \underbrace{\sigma}_{ds} + \beta \sum_{i=1}^{N} V(u_i)\,\sigma$$

where $u = (u_1\ u_2\ \dots\ u_N)^T \in M_{N\times 2}(\mathbb{R})$ is a discrete closed curve.

To simplify, we throw away σ to obtain:

$$E_{snake}(u) = \sum_{i=1}^{N} \frac{1}{2} \left| \frac{\overset{\in \mathbb{R}^2}{u_{i+1}} - \overset{\in \mathbb{R}^2}{u_i}}{\sigma} \right|^2 + \beta \sum_{i=1}^{N} V(u_i)$$

Note that E_{snake} is a multi-variable function depending on:

$u_{1x}, u_{1y}, u_{2x}, u_{2y}, \dots, u_{Nx}, u_{Ny}$, where $u_i = \begin{pmatrix} u_{ix} \\ u_{iy} \end{pmatrix} \in \mathbb{R}^2$.

To minimize Esnake, we compute the gradient of Esnake.

$$\text{Gradient of } E_{snake} = \nabla E_{snake} = \begin{pmatrix} \frac{\partial E_{snake}}{\partial u_{1x}} \\ \frac{\partial E_{snake}}{\partial u_{1y}} \\ \vdots \\ \frac{\partial E_{snake}}{\partial u_{ix}} \\ \frac{\partial E_{snake}}{\partial u_{iy}} \\ \vdots \end{pmatrix}$$

For simplicity, we can rewrite ∇E_{snake} as:

$$\nabla E_{snake} = \left(\frac{\partial E_{snake}}{\partial u_1}, \frac{\partial E_{snake}}{\partial u_2}, \dots, \frac{\partial E_{snake}}{\partial u_N} \right)^T = \begin{pmatrix} \frac{\partial E_{snake}}{\partial u_{1x}} & \frac{\partial E_{snake}}{\partial u_{1y}} \\ \vdots & \\ \frac{\partial E_{snake}}{\partial u_{ix}} & \frac{\partial E_{snake}}{\partial u_{iy}} \\ \vdots & \end{pmatrix}$$

$$\frac{\partial E_{snake}}{\partial u_1} \overset{def}{=} \begin{pmatrix} \frac{\partial E_{snake}}{\partial u_{1x}} \\ \frac{\partial E_{snake}}{\partial u_{1y}} \end{pmatrix}, \qquad \frac{\partial E_{snake}}{\partial u_2} \overset{def}{=} \begin{pmatrix} \frac{\partial E_{snake}}{\partial u_{2x}} \\ \frac{\partial E_{snake}}{\partial u_{2y}} \end{pmatrix}$$

Gradient descent:

$$M_{N\times 2} \ni \frac{u^{k+1} - u^{k}}{\tau} = -\underbrace{\nabla E_{snake}}_{\in M_{N\times 2}}, \qquad u^{k} \in M_{N\times 2}$$

(Dimension agrees)

Here,

$$\frac{\partial E_{snake}}{\partial u_i} = \begin{pmatrix} \frac{\partial E_{snake}}{\partial u_{ix}} \\ \frac{\partial E_{snake}}{\partial u_{iy}} \end{pmatrix}.$$

Hence

$$\frac{\partial V}{\partial u_i} = \begin{pmatrix} \frac{\partial V}{\partial u_{ix}} \\ \frac{\partial V}{\partial u_{iy}} \end{pmatrix} = \nabla V(u_i)$$

Recall that: $E_{snake}(u) = \sum_{i=1}^{N} \frac{1}{2} \left| \frac{u_{i+1} - u_i}{\sigma} \right|^2 + \beta \sum_{i=1}^{N} V(u_i)$

$$\therefore \frac{\partial E_{snake}}{\partial u_i} = -\left(\frac{u_{i+1} - u_i}{\sigma^2} \right) + \left(\frac{u_i - u_{i-1}}{\sigma^2} \right) + \beta \nabla V(u_i)$$

$$= \frac{-u_{i+1} + 2u_i - u_{i-1}}{\sigma^2} + \beta \nabla V(u_i)$$

The gradient descent algorithm can be written as:

$$\frac{u^{k+1} - u^k}{\tau} = -\nabla E_{snake}(u^k) = -\begin{pmatrix} \frac{\partial L_{snake}}{\partial u_{1x}^k} & \frac{\partial E_{snake}}{\partial u_{1y}^k} \\ & \vdots & \\ \frac{\partial E_{snake}}{\partial u_{ix}^k} & \frac{\partial E_{snake}}{\partial u_{iy}^k} \\ & \vdots & \end{pmatrix}$$

or $$\frac{u_i^{k+1} - u_i^k}{\tau} = \frac{u_{i+1}^k - 2u_i^k + u_{i-1}^k}{\sigma^2} - \beta \nabla V(u_i^k) \quad \text{for } i = 1, 2, \dots, N.$$

(Explicit Euler Scheme)

Remark:

- Sometimes, the semi-implicit scheme is used:

$$\frac{u_i^{k+1} - u_i^{k}}{\tau} = \frac{u_{i+1}^{k+1} - 2u_i^{k+1} + u_{i-1}^{k+1}}{\delta^2} - \beta \nabla V(u_i^k) \quad \text{for } i = 1, 2, \dots, N$$

- The explicit scheme can be written in matrix form as:

$$\frac{u^{k+1} - u^{k}}{\tau} = D u^{k} + \beta F(u^k)$$

where
$$D = \begin{pmatrix} -2 & 1 & 0 & \cdots & 0 & 1 \\ 1 & -2 & 1 & \cdots & 0 & 0 \\ 0 & 1 & -2 & 1 \cdots & 0 & 0 \\ & & \ddots & \ddots & \ddots & \\ 1 & 0 & \cdots & 0 & 1 & -2 \end{pmatrix} ; \quad F(u) = \begin{pmatrix} -\nabla V(u_1) & -\nabla V(u_2) & \cdots & \nabla V(u_N) \end{pmatrix}^T$$

$$u = \begin{pmatrix} u_1 & u_2 & \cdots & u_N \end{pmatrix}^T$$

- The semi-implicit scheme can be written in matrix form as:

$$\frac{u^{k+1} - u^{k}}{\tau} = D u^{k+1} + \beta F(u^k)$$

Remark:

- τ and σ have to be choosen carefully!

- F is called the driving force

- $(F(u))_i = -\nabla V(u_i)$. The edge detector is usually smoothed out by:

$$\tilde{V} = G * V \quad (G = \text{Gaussian function})$$

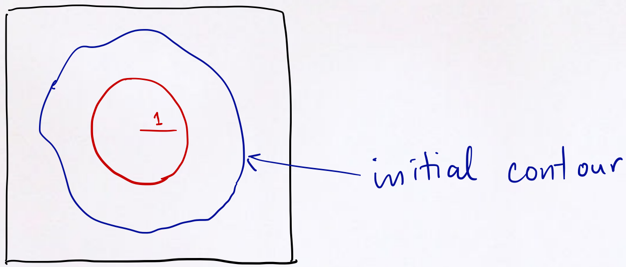
- Active contour is sensitive to noises (depends on edge detector)

- Active contour model cannot handle topological change.

Example 1: Let $I : \Omega \to \mathbb{R}$ be an image and $V : \Omega \to \mathbb{R}$ is the edge detector defined by:

$$V(\vec{p}) = V((p_1, p_2)) = \begin{cases} p_1^2 + p_2^2 & \text{if } p_1^2 + p_2^2 \geq 1 \\ 1 & \text{if } p_1^2 + p_2^2 < 1 \end{cases}$$

Assuming that the initial contour encloses the unit circle. Explain (intuitively) why the active contour model converges to a circle $C : \{(x, y) \in \mathbb{R} : x^2 + y^2 = 1\}$.



Solution: We cansider the explicit Euler model first.

$$\frac{u^{n+1}-u^n}{\tau} = \frac{Du^n}{\sigma^2} + \alpha F(u^n)$$

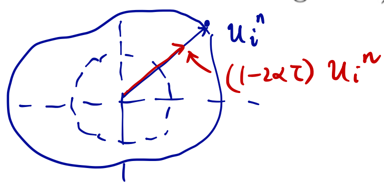
where $(F(u))_i = -\nabla V(u_i) = -\nabla V((u_{i1}, u_{i2})) = -2(u_{i1}, u_{i2})$.

Hence, if $u^n = \begin{pmatrix} u_{11}^n & u_{12}^n \\ u_{21}^n & u_{22}^n \\ \vdots & \vdots \\ u_{N1}^n & u_{N2}^n \end{pmatrix}$, then $F(u^n) = -2 \begin{pmatrix} u_{11}^n & u_{12}^n \\ u_{21}^n & u_{22}^n \\ \vdots & \vdots \\ u_{N1}^n & u_{N2}^n \end{pmatrix}$ (contour force). Our model becomes

$$u^{n+1} = u^n + \tau \frac{Du^n}{\sigma^2} - 2\alpha\tau \begin{pmatrix} u_{11}^n & u_{12}^n \\ u_{21}^n & u_{22}^n \\ \vdots & \vdots \\ u_{N1}^n & u_{N2}^n \end{pmatrix}$$

As $(F(u^n))_i = -\nabla V(u_{i1}^n, u_{i2}^n) = -2(u_{i1}^n, u_{i2}^n)$, the force attracts the contour to the circle. Also, $F = \vec{0}$ in the interior region since $\nabla V = 0$ (V = constant in the interior region C).

Now, $u^{n+1} = (1-2\alpha\tau) \begin{pmatrix} u_{11}^n & u_{12}^n \\ u_{21}^n & u_{22}^n \\ \vdots & \vdots \\ u_{N1}^n & u_{N2}^n \end{pmatrix} + \tau \frac{Du^n}{\sigma^2}$.



For semi-implicit scheme:

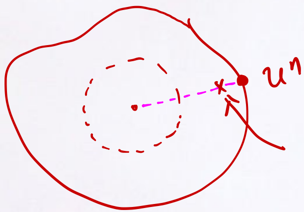
For semi-implicit scheme, the model is given by:

$$\frac{u^{n+1}-u^n}{\tau} = \frac{Du^{n+1}}{\sigma^2} + \alpha F(u^n)$$

Hence, we have $\left(I - \frac{\tau}{\sigma^2}D\right)u^{n+1} = (1-2\alpha\tau)u^n$

If τ is comparatively small, $\left(I - \frac{\tau}{\sigma^2}D\right) \approx I$.

The right hand side draws the contour to the unit circle.



$u^{n+1} \approx (1-2\alpha\tau)u^n$

<u>**Example 2**</u>: Consider the following discrete curve evolution model:

$$\frac{u^{n+1} - u^n}{\tau} = Au^n + \alpha F(u^n)$$

where $(F(u))_i = -\nabla V(u_i)$ and $V(\vec{x}) = V((x, y)) = x^2 + y^2$,

$$A = \begin{pmatrix} \sigma(1) & & & \\ & \sigma(2) & & \\ & & \ddots & \\ & & & \sigma(N) \end{pmatrix} \text{ and } \sigma(j) < 0 \text{ for all } j.$$

Prove that $\{u^n\}_{n=1}^{\infty}$ converges to a point curve $u = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{pmatrix}$.

Solution:

$$u^{n+1} = \left(I + \tau \begin{pmatrix} \sigma(1) & & & \\ & \sigma(2) & & \\ & & \ddots & \\ & & & \sigma(N) \end{pmatrix} - 2\alpha\tau I \right) u^n$$

$$= \begin{pmatrix} 1+\tau\sigma(1)-2\alpha\tau & & & \\ & 1+\tau\sigma(2)-2\alpha\tau & & \\ & & \ddots & \\ & & & 1+\tau\sigma(N)-2\alpha\tau \end{pmatrix} u^n$$

Assume τ is small enough such that $1+\tau\sigma(j)-2\alpha\tau > 0$ for all j.
Then, $0 < 1 + \underbrace{\tau\sigma(j)}_{-ve} - 2\alpha\tau < 1$ for all j. Easy to check:

$$u^n = \begin{pmatrix} \underbrace{(1+\tau\sigma(1)-2\alpha\tau)^n}_{\to 0} & & & \\ & \underbrace{(1+\tau\sigma(2)-2\alpha\tau)^n}_{\to 0} & & \\ & & \ddots & \\ & & & \underbrace{(1+\tau\sigma(N)-2\alpha\tau)^n}_{\to 0} \end{pmatrix} u^0$$

Then, $u^n \to \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{pmatrix}$ when $n \to \infty$

For semi-implicit scheme, we have:

$$\frac{u^{n+1} - u^n}{\tau} = Au^{n+1} + \alpha F(u^n)$$

Then,

$$\left(I - \tau \begin{pmatrix} \sigma(1) & & & \\ & \sigma(2) & & \\ & & \ddots & \\ & & & \sigma(N) \end{pmatrix} \right) u^{n+1} = (1 - 2\alpha\tau)u^n$$

$$\Rightarrow u^{n+1} = (1 - 2\alpha\tau) \begin{pmatrix} \frac{1}{1-\tau\sigma(1)} & & & \\ & \frac{1}{1-\tau\sigma(2)} & & \\ & & \ddots & \\ & & & \frac{1}{1-\tau\sigma(N)} \end{pmatrix} u^n$$

$$\Rightarrow u^n = \begin{pmatrix} \left(\frac{1-2\alpha\tau}{1-\tau\sigma(1)}\right)^n & & & \\ & \left(\frac{1-2\alpha\tau}{1-\tau\sigma(2)}\right)^n & & \\ & & \ddots & \\ & & & \left(\frac{1-2\alpha\tau}{1-\tau\sigma(N)}\right)^n \end{pmatrix} u^0$$

$$\therefore u^n \to \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{pmatrix} \text{ if } 1 - 2\alpha\tau > 0$$

since $1 - \tau\sigma(j) > 1$ for all j.

Explicit scheme:

Need $|1+\tau\sigma(j) - 2\alpha\tau| < 1$

Need smaller τ

Implicit scheme:

Need $|1-2\alpha\tau| < 1$

Allow larger τ.

$-1 < 1 + \tau\sigma(j) - 2\alpha\tau < 1$

$\Rightarrow \tau(2\alpha - \sigma(j)) < 2$

$\Rightarrow \tau < \frac{2}{2\alpha - \sigma(j)}$

$-1 < 1 - 2\alpha\tau$

$2\alpha\tau < 2$

$\tau < \frac{2}{2\alpha}$