

*“Don't expect to be easy to work, because if you're easy to do, your heart will be flattered and arrogant.”*  
\_ Buddha



# APT: PDE (Math Models)

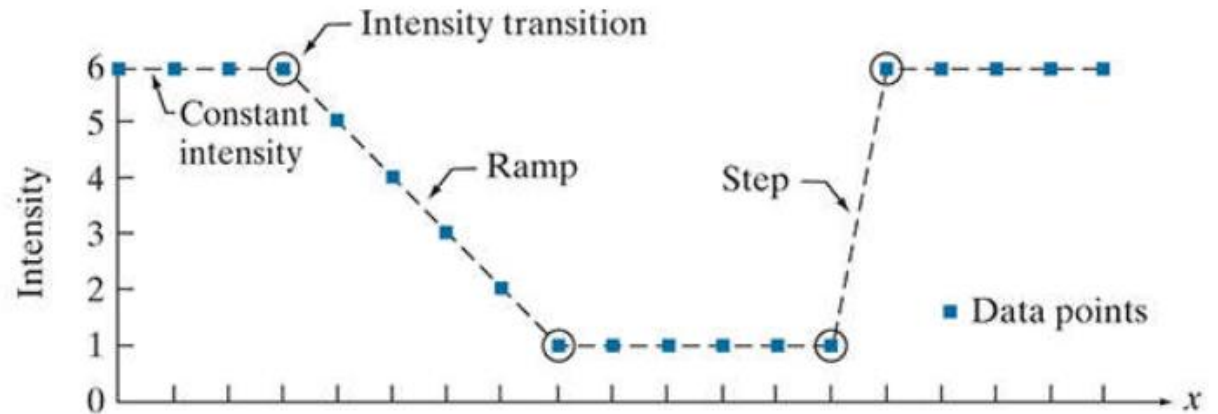
Presenter: Dr. Ha Viet Uyen Synh.



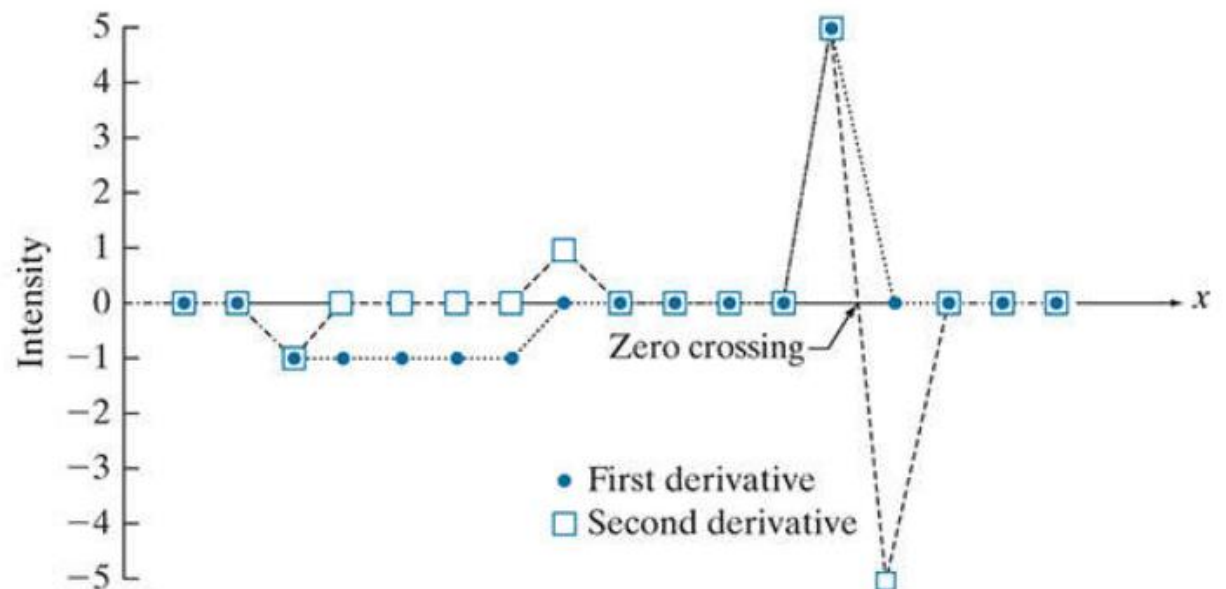
Part C

# **SHARPENING SPATIAL FILTERS**

# 1. Laplacian Filter



Values of scan line	6	6	6	6	5	4	3	2	1	1	1	1	1	1	6	6	6	6	6	$x$
1st derivative	0	0	-1	-1	-1	-1	-1	0	0	0	0	0	0	5	0	0	0	0	0	
2nd derivative	0	0	-1	0	0	0	0	1	0	0	0	0	0	5	-5	0	0	0	0	



# Laplacian Filter

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

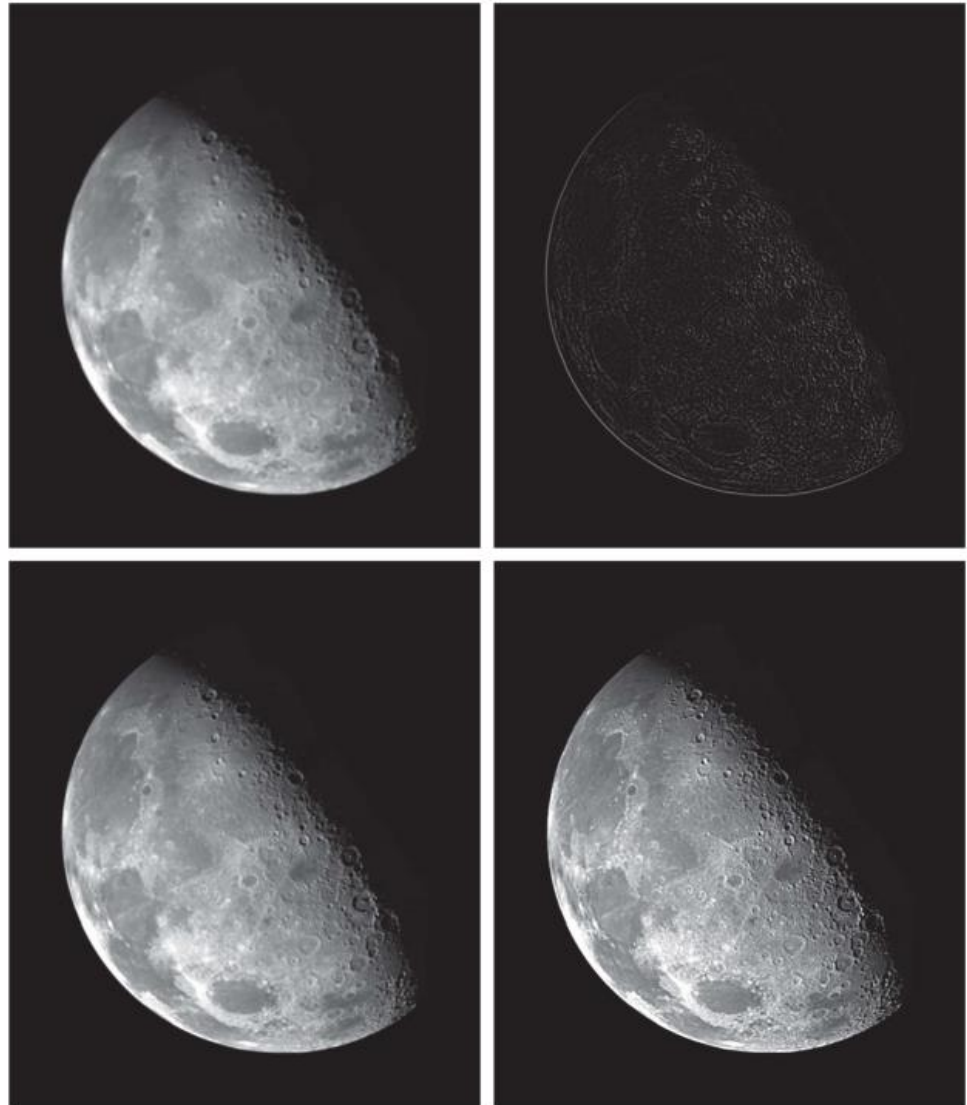
$$\frac{\partial^2 f}{\partial x^2} = f(x+1, y) + f(x-1, y) - 2f(x, y)$$

$$\frac{\partial^2 f}{\partial y^2} = f(x, y+1) + f(x, y-1) - 2f(x, y)$$

$$\nabla^2 f(x, y) = f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y)$$

0	1	0	1	1	1	0	-1	0	-1	-1	-1
1	-4	1	1	-8	1	-1	4	-1	-1	8	-1
0	1	0	1	1	1	0	-1	0	-1	-1	-1

# Image sharpening using the Laplacian



$$g(x,y) = f(x,y) + c[\nabla^2 f(x,y)]$$



## 2. Unsharp Masking and Highboost Filtering

Unsharp masking consists of the following steps:

1. Blur the original image.
2. Subtract the blurred image from the original (the resulting difference is called the mask.)
3. Add the mask to the original.

$$g_{\text{mask}}(x, y) = f(x, y) - \bar{f}(x, y)$$

where  $\bar{f}(x, y)$  is a blurred image.

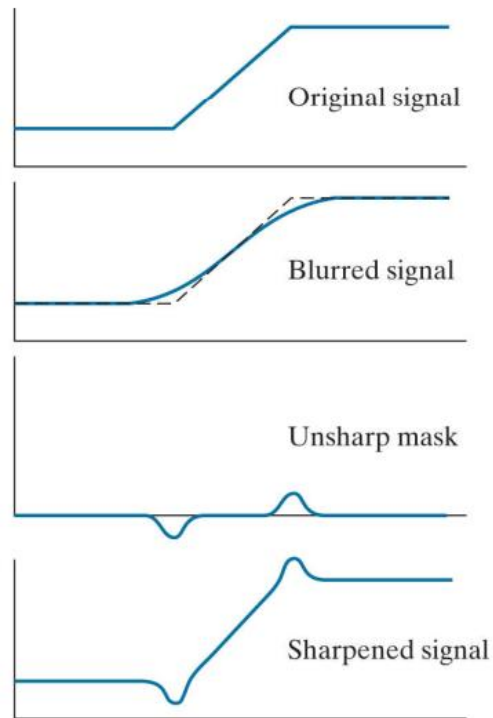
$$g(x, y) = f(x, y) + kg_{\text{mask}}(x, y)$$

where we included a weight, for generality.

$k = 1$  : Unsharp Masking

$k > 1$  : Highboost Filtering

# Image sharpening using the Unsharp Masking



$k=1$



$k=2$



$k=3$



# 3. Gradient

$$\nabla f \equiv \text{grad}(f) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

$$M(x, y) = \| \nabla f \| = \text{mag}(\nabla f) = \sqrt{g_x^2 + g_y^2}$$

$$M(x, y) \approx |g_x| + |g_y|$$

$$g_x = (z_9 - z_5) \quad g_y = (z_8 - z_6)$$

$$M(x, y) = [(z_9 - z_5)^2 + (z_8 - z_6)^2]^{1/2}$$

$$M(x, y) \approx |z_9 - z_5| + |z_8 - z_6|$$

$$g_x = \frac{\partial f}{\partial x} = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)$$

$$g_y = \frac{\partial f}{\partial y} = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)$$

$$M(x, y) = [g_x^2 + g_y^2]^{\frac{1}{2}} = [[(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)]^2 + [(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)]^2]^{\frac{1}{2}}$$

$z_1$	$z_2$	$z_3$
$z_4$	$z_5$	$z_6$
$z_7$	$z_8$	$z_9$

-1	0	0	-1
0	1	1	0

-1	-2	-1	-1	0	1
0	0	0	-2	0	2
1	2	1	-1	0	1





PART D

# **APPLICATIONS**

# Optical Flow

## Problem

Let  $\mathbf{x} \equiv (x, y, \underline{t})^T$  be a position  $(x, y)$  in the frame at time  $t$ ,  $I(\mathbf{x})$  is the grey value at position  $(x, y)$  in the frame at time  $t$ . Let  $\mathbf{w} \equiv (u, v, \underline{I})^T$  be a motion vector at the time  $t$ .





# Optical Flow

Phương trình bảo toàn cường độ sáng (Brightness Constancy Constraint Equation - BCCE) trong **Optical Flow** xuất phát từ giả định rằng cường độ của một điểm ảnh không thay đổi theo thời gian khi nó di chuyển trong một chuỗi hình ảnh.

---

## 1. Giả định cơ bản

Khi một vật thể di chuyển trong một cảnh, ta giả định rằng cường độ sáng của mỗi điểm ảnh vẫn giữ nguyên, tức là nếu một điểm ảnh  $(x, y)$  tại thời điểm  $t$  có giá trị cường độ là  $I(x, y, t)$ , thì sau một khoảng thời gian nhỏ  $dt$ , điểm ảnh đó di chuyển đến vị trí  $(x + dx, y + dy)$  và có giá trị cường độ tương tự:

$$I(x, y, t) = I(x + dx, y + dy, t + dt)$$

Điều này có nghĩa là không có sự thay đổi đáng kể về ánh sáng hoặc tính chất của vật thể trong quá trình chuyển động.

## 2. Áp dụng khai triển Taylor

Chúng ta áp dụng khai triển Taylor bậc nhất cho hàm  $I(x, y, t)$  tại điểm  $(x + dx, y + dy, t + dt)$ :

$$I(x + dx, y + dy, t + dt) \approx I(x, y, t) + \frac{\partial I}{\partial x} dx + \frac{\partial I}{\partial y} dy + \frac{\partial I}{\partial t} dt$$

Vì theo giả định bảo toàn cường độ sáng:

$$I(x, y, t) = I(x + dx, y + dy, t + dt)$$

Nên lấy hiệu hai vế:

$$I(x + dx, y + dy, t + dt) - I(x, y, t) = 0$$

Suy ra:

$$\frac{\partial I}{\partial x} dx + \frac{\partial I}{\partial y} dy + \frac{\partial I}{\partial t} dt = 0$$

Chia cả hai vế cho  $dt$ :

$$I_x \frac{dx}{dt} + I_y \frac{dy}{dt} + I_t = 0$$

Do:

$$\frac{dx}{dt} = u, \quad \frac{dy}{dt} = v$$

ta có phương trình:

$$I_x u + I_y v + I_t = 0$$

Đây chính là **phương trình bảo toàn cường độ sáng**, là nền tảng của phương pháp Horn-Schunck và Lucas-Kanade trong Optical Flow.



# 1. Horn-Schunck's method

## 1. Công thức chính của Horn-Schunck

Horn-Schunck dựa trên phương trình bảo toàn cường độ sáng:

$$I_x u + I_y v + I_t = 0$$

trong đó:

- $I_x, I_y$  là đạo hàm của ảnh theo hướng x và y.
- $I_t$  là đạo hàm của ảnh theo thời gian (sự thay đổi cường độ pixel giữa hai khung hình).
- $u, v$  là các thành phần của vector vận tốc cần tìm.

Do phương trình trên có một **ấn số ít hơn phương trình** (ill-posed problem), Horn-Schunck bổ sung thêm một ràng buộc về độ trơn:

$$E = \iint ((I_x u + I_y v + I_t)^2 + \lambda(\|\nabla u\|^2 + \|\nabla v\|^2)) dx dy$$

với:

- Thành phần đầu tiên: giữ cho nghiệm của phương trình bảo toàn sáng đúng.
- Thành phần thứ hai: đảm bảo trường vận tốc trơn, với hệ số  $\lambda$  kiểm soát mức độ trơn.



## 2. Phương trình Euler-Lagrange

Tối thiểu hoá năng lượng  $E$  bằng cách lấy đạo hàm theo  $u$  và  $v$ , ta thu được hệ phương trình đạo hàm riêng:

$$I_x(I_x u + I_y v + I_t) - \lambda \nabla^2 u = 0$$

$$I_y(I_x u + I_y v + I_t) - \lambda \nabla^2 v = 0$$

trong đó  $\nabla^2 u$  và  $\nabla^2 v$  là Laplacian của  $u$  và  $v$ , được tính như:

$$\nabla^2 u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}$$

$$\nabla^2 v = \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}$$

### 3. Biến đổi thành dạng số (Numerical Methods)

Bây giờ, ta cần chuyển phương trình trên sang dạng rời rạc để có thể lập trình trên máy tính.

#### Bước 1: Xấp xỉ đạo hàm ảnh

- Đạo hàm không gian  $I_x, I_y$  có thể tính bằng bộ lọc Sobel:

$$I_x = \frac{1}{8} \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * I$$

$$I_y = \frac{1}{8} \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} * I$$

- Đạo hàm thời gian  $I_t$  có thể lấy bằng hiệu trung bình giữa hai ảnh liên tiếp:

$$I_t = \frac{1}{2}(I_{t+1} - I_t)$$



## Bước 2: Xấp xỉ toán tử Laplacian

Toán tử Laplacian của  $u$  và  $v$  được tính bằng trung bình lân cận 4 hướng:

$$\nabla^2 u = \bar{u} - u$$

$$\nabla^2 v = \bar{v} - v$$

trong đó  $\bar{u}, \bar{v}$  là giá trị trung bình của  $u$  và  $v$  từ 4 điểm lân cận:

$$\bar{u}_{i,j} = \frac{1}{4}(u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1})$$

$$\bar{v}_{i,j} = \frac{1}{4}(v_{i+1,j} + v_{i-1,j} + v_{i,j+1} + v_{i,j-1})$$

## Bước 3: Giải hệ phương trình bằng phép lặp

Sau khi xấp xỉ, ta có công thức cập nhật  $u, v$  theo từng bước lặp:

$$u^{(n+1)} = \bar{u}^{(n)} - \frac{I_x(I_x \bar{u}^{(n)} + I_y \bar{v}^{(n)} + I_t)}{\lambda + I_x^2 + I_y^2}$$

$$v^{(n+1)} = \bar{v}^{(n)} - \frac{I_y(I_x \bar{u}^{(n)} + I_y \bar{v}^{(n)} + I_t)}{\lambda + I_x^2 + I_y^2}$$

Quá trình này được lặp lại cho đến khi  $u$  và  $v$  hội tụ hoặc đạt đến số vòng lặp tối đa.



## 2. Thomas Brox's method

### Bước 1: Ý tưởng chính (chi tiết hơn)

Phương pháp của Brox tối ưu hóa một hàm năng lượng kết hợp:

1. **Brightness Constancy:** Giả định cường độ sáng không đổi giữa hai khung hình.
2. **Gradient Constancy:** Giả định gradient ảnh không đổi, giúp xử lý các thay đổi ánh sáng.
3. **Smoothness với Total Variation:** Đảm bảo trường vận tốc mượt mà nhưng vẫn giữ được các biên bất liên tục (edges).
4. **Coarse-to-Fine:** Dùng kim tự tháp ảnh để xử lý chuyển động lớn, bắt đầu từ độ phân giải thấp và tinh chỉnh dần.

Hàm năng lượng tổng quát:

$$E(u, v) = E_{\text{data}}(u, v) + \alpha E_{\text{smooth}}(u, v)$$

- $E_{\text{data}}$ : Đo sự khác biệt giữa hai khung hình.
- $E_{\text{smooth}}$ : Ràng buộc mượt mà.
- $\alpha$ : Hệ số cân bằng.

## Bước 2: Hàm năng lượng (Energy Functional)

Brox định nghĩa hàm năng lượng cần tối ưu như sau:

$$E(u, v) = E_{\text{data}}(u, v) + \alpha E_{\text{smooth}}(u, v)$$

### 2.1. Data Term ( $E_{\text{data}}$ )

Data term kết hợp Brightness Constancy và Gradient Constancy:

$$E_{\text{data}}(u, v) = \int_{\Omega} [\Psi(|I(x+w) - I(x)|^2) + \gamma \Psi(|\nabla I(x+w) - \nabla I(x)|^2)] dx dy$$

Trong đó:

- $w = (u, v)$ : Trường Optical Flow.
- $I(x) = I(x, y, t)$ ,  $I(x+w) = I(x+u, y+v, t+1)$ : Hai khung hình liên tiếp.
- $\nabla I = (I_x, I_y)$ : Gradient của ảnh.
- $\Psi(s^2) = \sqrt{s^2 + \epsilon^2}$ : Hàm robust (L1-norm xấp xỉ) để giảm ảnh hưởng của nhiễu và ngoại lai (outliers), với  $\epsilon$  nhỏ (thường  $10^{-3}$ ).
- $\gamma$ : Hệ số cân bằng giữa brightness và gradient terms.

### 2.2. Smoothness Term ( $E_{\text{smooth}}$ )

Smoothness term sử dụng Total Variation để bảo toàn bất liên tục:

$$E_{\text{smooth}}(u, v) = \int_{\Omega} \Psi(|\nabla u|^2 + |\nabla v|^2) dx dy$$

- $|\nabla u|^2 = u_x^2 + u_y^2$ ,  $|\nabla v|^2 = v_x^2 + v_y^2$ : Độ lớn gradient của  $u$  và  $v$ .
- $\alpha$ : Hệ số điều chỉnh mức độ mượt mà.

Mục tiêu: Tìm  $u, v$  sao cho  $E(u, v)$  nhỏ nhất.



### Bước 3: Phương trình Euler-Lagrange

Để tối ưu  $E(u, v)$ , ta tính đạo hàm biến phân (variational derivative) và đặt bằng 0. Vì hàm  $\Psi$  không tuyến tính và  $I(x + w)$  phụ thuộc phi tuyến vào  $u, v$ , ta cần một phương pháp lặp để giải.

Tuy nhiên, trước hết, hãy xem dạng tổng quát của phương trình Euler-Lagrange:

- Đối với  $u$ :  $\frac{\partial}{\partial u} E_{\text{data}} - \text{div} (\Psi'(|\nabla u|^2 + |\nabla v|^2) \nabla u) = 0$
- Đối với  $v$ :  $\frac{\partial}{\partial v} E_{\text{data}} - \text{div} (\Psi'(|\nabla u|^2 + |\nabla v|^2) \nabla v) = 0$

Phần  $E_{\text{data}}$  phức tạp vì  $I(x + w)$  và  $\nabla I(x + w)$  là phi tuyến. Brox giải quyết bằng cách dùng chiến lược warping và lặp cố định (fixed-point iteration).

#### Bước 4: Xấp xỉ số và lặp cố định

Vì  $E(u, v)$  là phi tuyến (do  $I_2(x + u, y + v)$  và  $\Psi$ ), Brox dùng hai vòng lặp:

1. **Outer Iteration:** Cập nhật warping  $I_2$  dựa trên  $w^k$ .
2. **Inner Iteration:** Tính  $du, dv$  bằng cách tuyến tính hóa và giải hệ phương trình.

##### 4.1. Xấp xỉ Incremental Flow

Tại vòng lặp  $k$ :

$$I_2(x + w^{k+1}) \approx I_2(x + w^k) + I_{2x}(x + w^k)du + I_{2y}(x + w^k)dv$$

$$\nabla I_2(x + w^{k+1}) \approx \nabla I_2(x + w^k) + \nabla(\nabla I_2(x + w^k)) \cdot (du, dv)$$

- $I_{2x}, I_{2y}$ : Gradient của  $I_2$  tại  $x + w^k$ .

Thay vào  $E_{\text{data}}$ :

$$E_{\text{data}}(du, dv) = \int \Psi(|I_2(x + w^k) + I_{2x}du + I_{2y}dv - I_1|^2) + \gamma \Psi(|\nabla I_2(x + w^k) + \nabla I_{2x}du + \nabla I_{2y}dv - \nabla I_1|^2)$$

##### 4.2. Phương trình Euler-Lagrange


Đạo hàm biến phân cho  $du$  (tương tự cho  $dv$ ):

$$\Psi'_c(I_{2x}(I_2^k + I_{2x}du + I_{2y}dv - I_1)) + \gamma \Psi'_g(\nabla I_{2x} \cdot (\nabla I_2^k + \nabla I_{2x}du + \nabla I_{2y}dv - \nabla I_1)) - \alpha \text{div}(\Psi'_s \nabla du) = 0$$

- $\Psi'_c = \frac{1}{\sqrt{(I_2^k - I_1)^2 + \epsilon^2}}$ : Trọng số của brightness term.
- $\Psi'_g$ : Trọng số của gradient term.
- $\Psi'_s = \frac{1}{\sqrt{|\nabla u|^2 + |\nabla v|^2 + \epsilon^2}}$ : Trọng số smoothness.

##### 4.3. Lặp cố định

- **Outer loop:** Cập nhật  $I_2^k$  bằng warping.
- **Inner loop:** Giải  $du, dv$  bằng phương pháp lặp (SOR hoặc Gauss-Seidel).



## Bước 5: Công thức xấp xỉ số chi tiết để lập trình

Mục tiêu là biến phương trình Euler-Lagrange từ bước 4 thành một dạng rời rạc có thể tính toán trên lưới pixel  $(i, j)$ . Ta cần:

1. Rời rạc hóa các thành phần gradient và divergence.
2. Tính toán các trọng số  $\Psi'$  tại mỗi pixel.
3. Áp dụng phương pháp lặp cố định để cập nhật  $du$  và  $dv$ .

### 5.1. Rời rạc hóa trên lưới pixel

Giả sử ảnh có kích thước  $H \times W$ , các biến được biểu diễn dưới dạng ma trận:

- $I_1(i, j), I_2(i, j)$ : Giá trị cường độ tại pixel  $(i, j)$  của hai khung hình.
- $I_{2x}(i, j), I_{2y}(i, j)$ : Gradient của  $I_2$  sau khi warped.
- $u(i, j), v(i, j)$ : Thành phần flow tổng tại vòng lặp  $k$ .
- $du(i, j), dv(i, j)$ : Incremental flow cần tính.

## 5.2. Công thức Euler-Lagrange rời rạc

Phương trình tổng quát từ bước 4:

$$\Psi'_c(I_{2x}(I_2^k + I_{2x}du + I_{2y}dv - I_1)) + \gamma\Psi'_g(\nabla I_{2x} \cdot (\nabla I_2^k + \nabla I_{2x}du + \nabla I_{2y}dv - \nabla I_1)) - \alpha \operatorname{div}(\Psi'_s \nabla du) = 0$$

Ta sẽ rời rạc từng thành phần:

### (a) Data Term - Brightness Constancy

- Residual:  $\rho_c(i, j) = I_2^k(i, j) + I_{2x}(i, j)du(i, j) + I_{2y}(i, j)dv(i, j) - I_1(i, j)$
- Trọng số:  $\Psi'_c(i, j) = \frac{1}{\sqrt{\rho_c(i, j)^2 + \epsilon^2}}$
- Đạo hàm theo  $du$ :  $\Psi'_c(i, j)I_{2x}(i, j)\rho_c(i, j)$

### (b) Data Term - Gradient Constancy

- Gradient residual:
  - $\rho_{gx}(i, j) = I_{2xx}(i, j)du(i, j) + I_{2xy}(i, j)dv(i, j) + I_{2x}(i, j) - I_{1x}(i, j)$
  - $\rho_{gy}(i, j) = I_{2yx}(i, j)du(i, j) + I_{2yy}(i, j)dv(i, j) + I_{2y}(i, j) - I_{1y}(i, j)$
- Trọng số:  $\Psi'_g(i, j) = \frac{1}{\sqrt{\rho_{gx}(i, j)^2 + \rho_{gy}(i, j)^2 + \epsilon^2}}$
- Đạo hàm theo  $du$ :  $\gamma\Psi'_g(i, j)(I_{2xx}(i, j)\rho_{gx}(i, j) + I_{2yx}(i, j)\rho_{gy}(i, j))$



### (c) Smoothness Term

- Gradient của  $du$  và  $dv$  (dùng finite difference):
  - $du_x(i, j) \approx du(i+1, j) - du(i, j)$ ,  $du_y(i, j) \approx du(i, j+1) - du(i, j)$
  - Tương tự cho  $dv_x, dv_y$ .
- Magnitude:  $s(i, j) = du_x(i, j)^2 + du_y(i, j)^2 + dv_x(i, j)^2 + dv_y(i, j)^2$
- Trọng số:  $\Psi'_s(i, j) = \frac{1}{\sqrt{s(i, j) + \epsilon^2}}$
- Divergence (rời rạc hóa):
  - $\text{div}(\Psi'_s \nabla du)(i, j) \approx (\Psi'_s(i, j) du_x(i, j) - \Psi'_s(i-1, j) du_x(i-1, j)) + (\Psi'_s(i, j) du_y(i, j) - \Psi'_s(i, j-1) du_y(i, j-1))$

### 5.3. Công thức lặp cố định rời rạc

Ta viết lại phương trình cho  $du$  tại pixel  $(i, j)$ :

$$\Psi'_c I_{2x} \rho_c + \gamma \Psi'_g (I_{2xx} \rho_{gx} + I_{2yx} \rho_{gy}) - \alpha \text{div}(\Psi'_s \nabla du) = 0$$

Dùng phương pháp lặp Gauss-Seidel:

$$du^{n+1}(i, j) = du^n(i, j) + \omega \frac{-\Psi'_c(i, j) I_{2x}(i, j) \rho_c^n(i, j) - \gamma \Psi'_g(i, j) (I_{2xx}(i, j) \rho_{gx}^n(i, j) + I_{2yx}(i, j) \rho_{gy}^n(i, j)) + \alpha \text{div}(\Psi'_s \nabla du^n)(i, j)}{\Psi'_c(i, j) I_{2x}(i, j)^2 + \gamma \Psi'_g(i, j) (I_{2xx}(i, j)^2 + I_{2yx}(i, j)^2) + \alpha \Psi'_s(i, j)}$$

- $\omega$ : Hệ số thư giãn (thường 1.0-1.9 cho SOR).
- Tương tự cho  $dv$ .



**Any Questions?**