

COLE.VN
connecting knowledge

Chủ đề:

**Xử lý ảnh (Computer vision) Các kỹ
thuật trích xuất đặt trưng**

Nội dung chính

Giới thiệu

Các phương pháp phát hiện biên

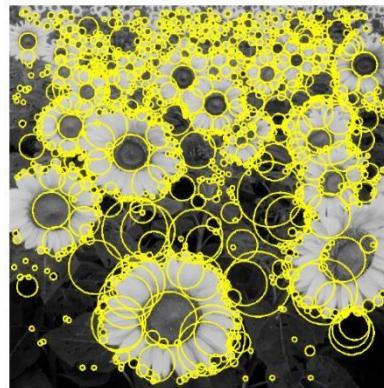
Phát hiện đường thẳng

Giới thiệu

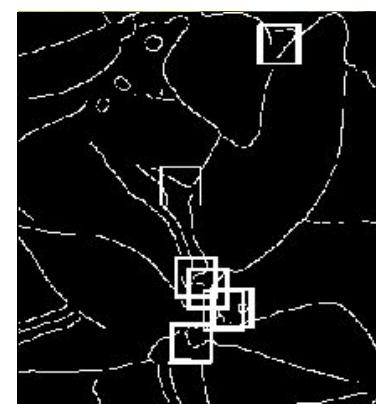
- Vai trò của các local features:
 - Cô đọng hóa nguồn thông tin từ ảnh
 - Đưa vào mô hình để xử lý



Điểm góc (Conner)
 Robot navigation



Điểm đặc trưng
 So sánh ảnh



Mật độ biên
 Đặc trưng đối tượng

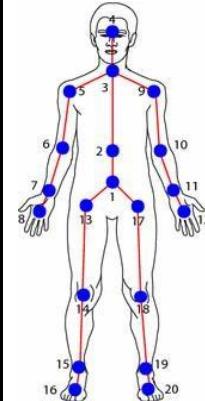


Điểm đặc trưng
 xoay/dịch chuyển

- Một số ứng dụng:
 - So sánh khác biệt giữa hai ảnh
 - Phát hiện đối tượng/object
 - Theo bám đối tượng
 - Làm khớp (stitching/panorama) và tạo ảnh 3-D

Giới thiệu

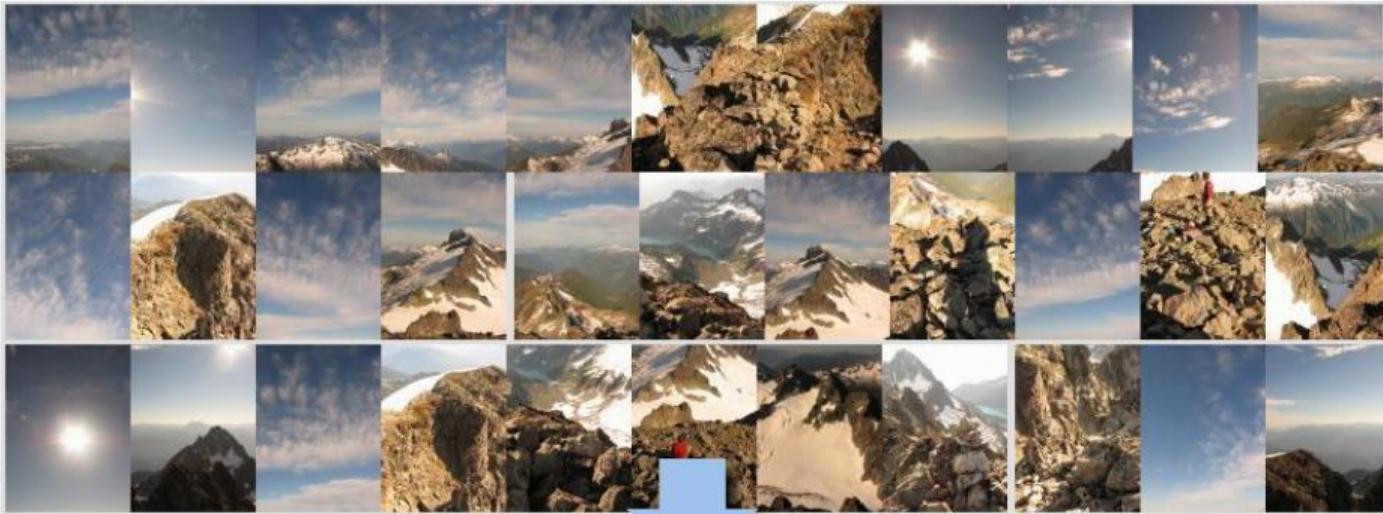
- Vai trò của các local features:
 - Không chỉ được quan sát/trích chọn không gian (spatial) và thời gian (temporal)



- Ứng dụng
 - Nhận dạng hoạt động, tương tác người - máy (HCI)
 - Theo dõi bám đối tượng
 - Thực tại ảo (virtual reality)

Một số ứng dụng

- Automatic Panorama



Credit: Matt Brown

Một số ứng dụng

- Nhận diện các vật thể giống nhau



Schmid and Mohr 1997



Sivic and Zisserman, 2003



Rothganger et al. 2003

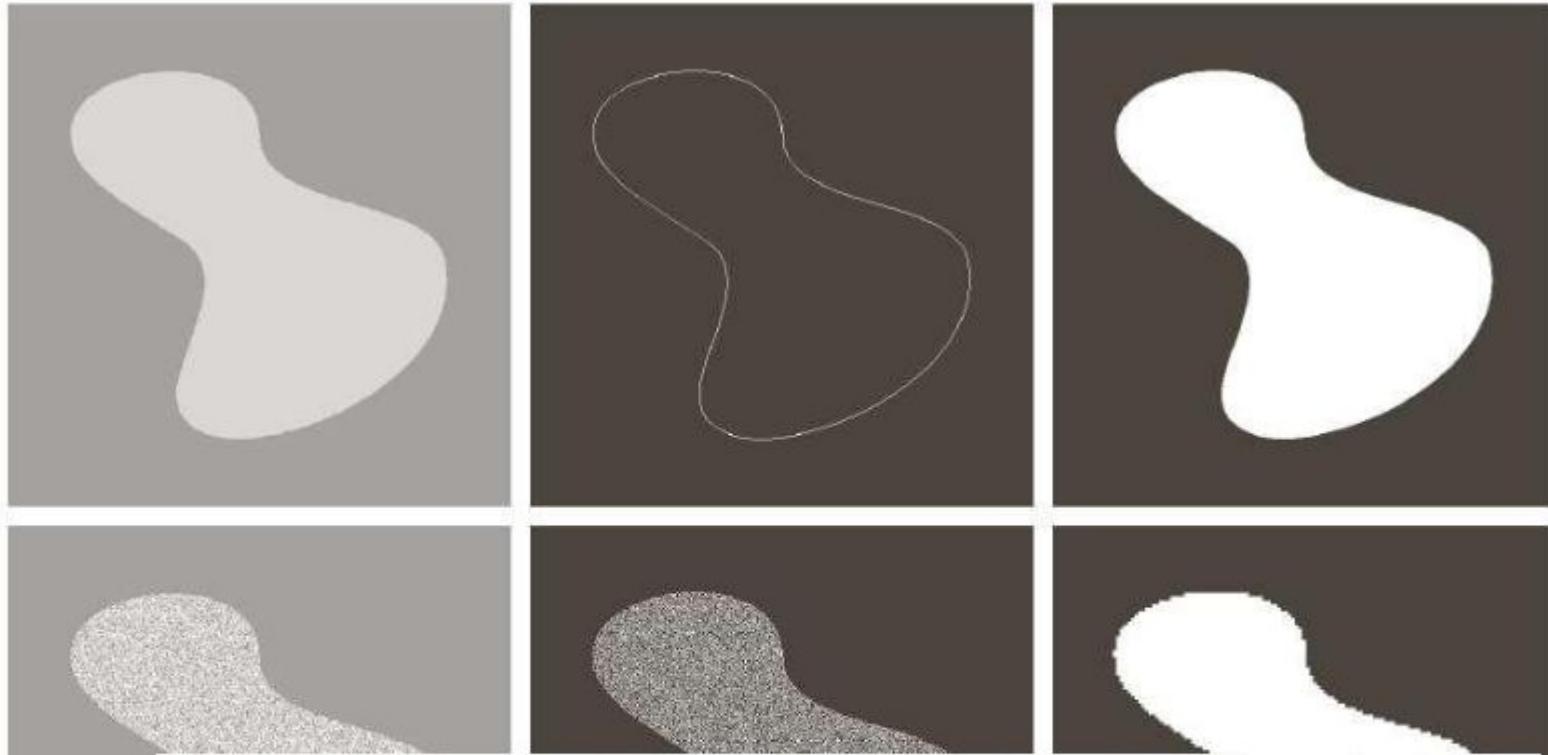


Lowe 2002

Giới thiệu về trích xuất biên

- **Điểm biên:** Một điểm ảnh được coi là điểm biên nếu có sự thay đổi nhanh hoặc đột ngột về mức xám (hoặc màu).
- Ví dụ, trong ảnh nhị phân, điểm đen được gọi là điểm biên nếu lân cận của nó có ít nhất một điểm trắng.
- **Đường biên** còn gọi là đường bao (**boundary**): Là tập hợp các điểm biên liên tiếp.
- Ý nghĩa của **đường biên**
 - Đường biên là một loại đặc trưng cục bộ tiêu biểu trong phân tích, nhận dạng ảnh.
 - Người ta sử dụng biên làm phân cách các vùng xám (hoặc màu) cách biệt. Ngược lại, người ta cũng sử dụng các vùng ảnh để tìm phân cách.
 - Các phương pháp tìm biên
 - **Các phương pháp phát hiện biên trực tiếp**
 - **Phát hiện biên gián tiếp**

Khó khăn của bài toán phát hiện biên

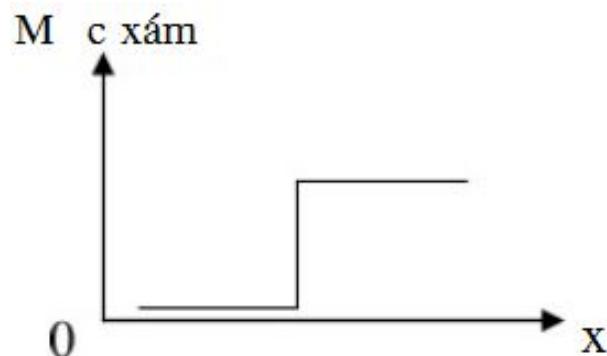


a b c
d e f

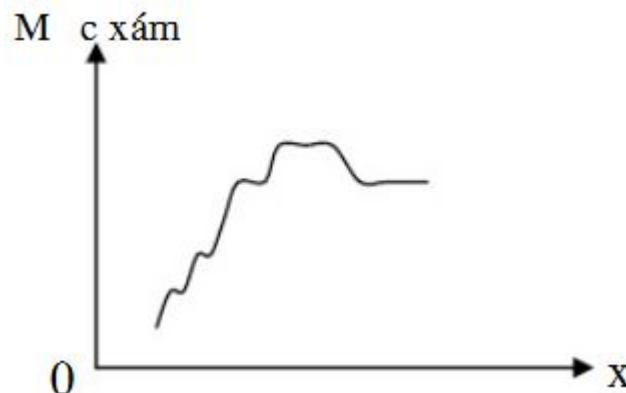
FIGURE 10.1 (a) Image containing a region of constant intensity. (b) Image showing the boundary of the inner region, obtained from intensity discontinuities. (c) Result of segmenting the image into two regions. (d) Image containing a textured region. (e) Result of edge computations. Note the large number of small edges that are connected to the original boundary, making it difficult to find a unique boundary using only edge information. (f) Result of segmentation based on region properties.

Khó khăn của bài toán phát hiện biên

- ◆ Một cách lý tưởng (ảnh đen trắng) thì đồ thị sự biến thiên mức xám của các điểm ảnh khi qua biên như sau:



- ◆ Đối với các ảnh đa mức xám thì đồ thị có dạng

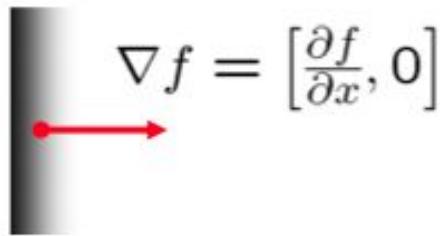


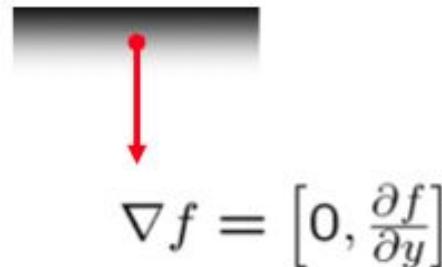
Phát hiện biên trực tiếp

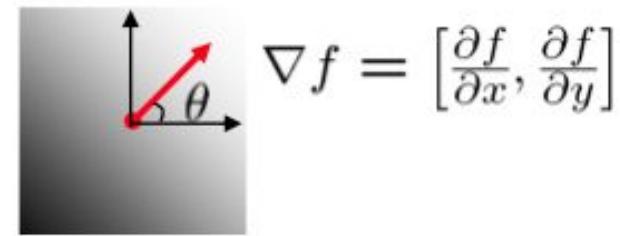
- ◆ Phương pháp này chủ yếu dựa vào sự biến thiên độ sáng của điểm ảnh để làm nổi biên bằng kỹ thuật đạo hàm.
 - Nếu lấy đạo hàm bậc nhất của $f(x,y)$ ta có phương pháp Gradient
 - Nếu lấy đạo hàm bậc hai của $f(x,y)$ ta có phương pháp Laplace
- ◆ **3 kỹ thuật tiêu biểu:**
 - Kỹ thuật phát hiện biên Gradient
 - Kỹ thuật phát hiện biên Laplace
 - Kỹ thuật Canny

Phát hiện biên trực tiếp - Kỹ thuật gradient

- ◆ Gradient là một vectơ có các thành phần biểu thị tốc độ thay đổi mức xám của điểm ảnh (theo hai hướng x,y đối với ảnh 2 chiều) tức là: $\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right]$

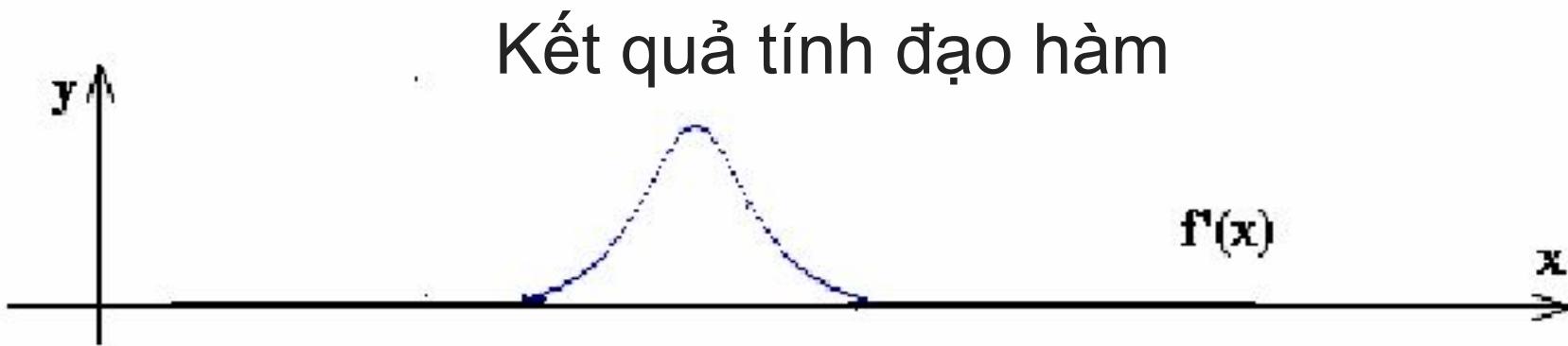
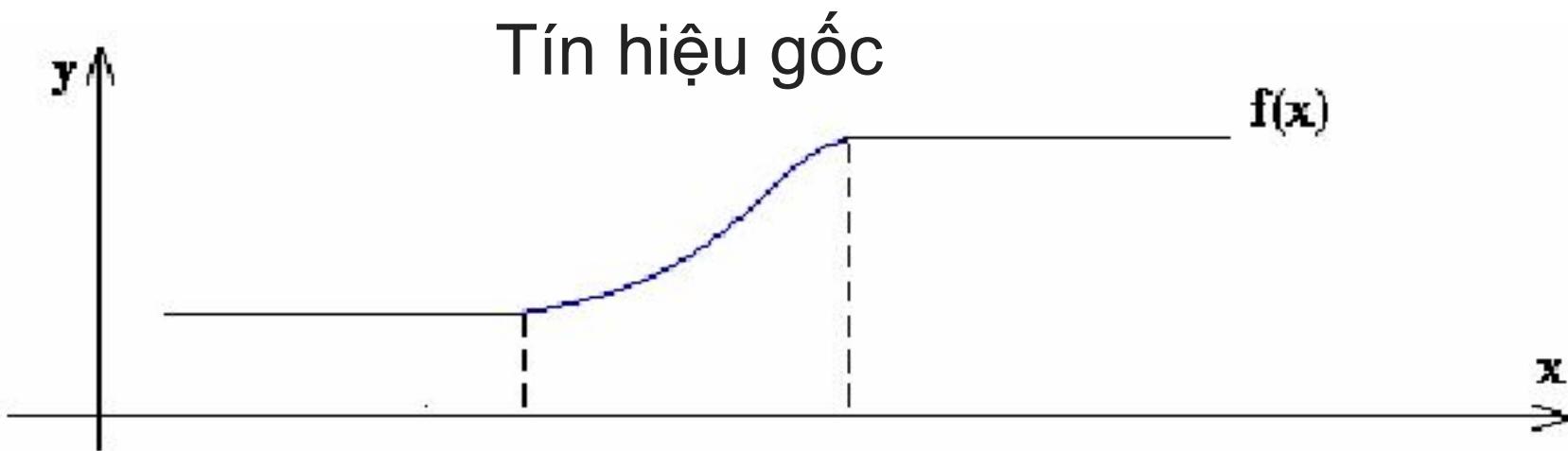

$$\nabla f = \left[\frac{\partial f}{\partial x}, 0 \right]$$


$$\nabla f = \left[0, \frac{\partial f}{\partial y} \right]$$


$$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right]$$

- ◆ Độ mạnh (rõ nét) của một cạnh thường được xác định bằng độ lớn của dốc $\|\nabla f\| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$

Minh họa kết quả phép lấy đạo hàm



Kỹ thuật gradient - phép tính lấy đạo hàm trong ảnh

- ◆ Với biên độ $G(m,n)$ tại điểm (m,n) được tính:

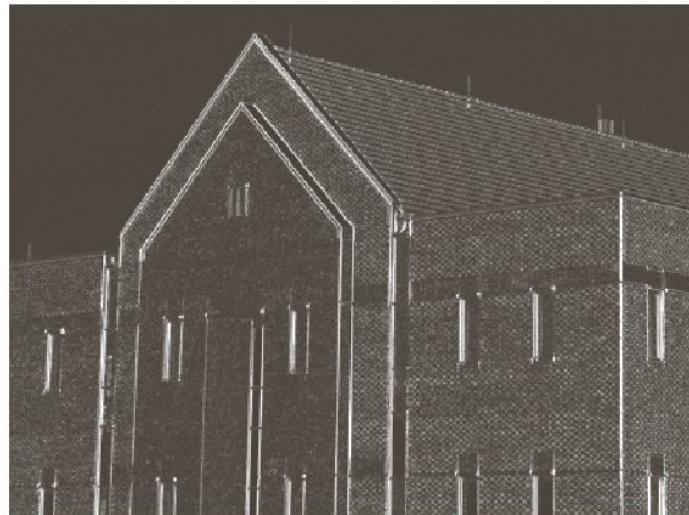
$$G(m,n) = \sqrt{G_x^2(m,n) + G_y^2(m,n)}$$

- ◆ Để giảm độ phức tạp tính toán, $G(m,n)$ được tính gần đúng như sau:

$$G(m,n) = |G_x(m,n)| + |G_y(m,n)|$$

- ◆ Một số toán tử Gradient tiêu biểu như toán tử Prewitt, Sobel, Robert đẳng hướng (Isometric),

Kỹ thuật gradient - kết quả tính đạo hàm



a	b
c	d

FIGURE 10.16
(a) Original image of size 834×1114 pixels, with intensity values scaled to the range $[0, 1]$.
(b) $|g_x|$, the component of the gradient in the x -direction, obtained using the Sobel mask in Fig. 10.14(f) to filter the image.
(c) $|g_y|$, obtained using the mask in Fig. 10.14(g).
(d) The gradient image, $|g_x| + |g_y|$.

Kỹ thuật gradient - kết quả tính đạo hàm



FIGURE 10.18
Same sequence as
in Fig. 10.16, but
with the original
image smoothed
using a 5×5
averaging filter
prior to edge
detection.

Kỹ thuật gradient - kết quả tính đạo hàm



FIGURE 10.20 (a) Thresholded version of the image in Fig. 10.16(d), with the threshold selected as 33% of the highest value in the image; this threshold was just high enough to eliminate most of the brick edges in the gradient image. (b) Thresholded version of the image in Fig. 10.18(d), obtained using a threshold equal to 33% of the highest value in that image.

Kỹ thuật gradient - Bộ lọc Prewitt

- ◆ Kỹ thuật này sử dụng 2 bộ lọc nhập chập xấp xỉ đạo hàm theo 2 hướng x và y là:

$$H_x = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix} \quad H_y = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$$

- ◆ Bước 1: Tính $I \otimes H_x$ và $I \otimes H_y$
- ◆ Bước 2: Tính $(I \otimes H_x) + (I \otimes H_y)$

Kỹ thuật gradient - Bộ lọc Prewitt - ví dụ

$$I = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 5 & 5 & 5 & 5 & 0 & 0 \\ 5 & 5 & 5 & 5 & 0 & 0 \\ 5 & 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Ảnh gốc

$$H_x = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & -1 \end{pmatrix}$$
$$H_y = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$$

Cửa sổ Prewitt

Kỹ thuật gradient - Bộ lọc Prewitt - ví dụ

$$I \otimes H_y = \begin{pmatrix} 15 & 15 & 10 & 5 & * & * \\ 0 & 0 & 0 & 0 & * & * \\ -15 & -15 & -10 & -5 & * & * \\ -15 & -15 & -10 & -5 & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \end{pmatrix} \quad I \otimes H_x = \begin{pmatrix} 0 & 0 & -10 & -10 & * & * \\ 0 & 0 & -15 & -15 & * & * \\ 0 & 0 & -10 & -10 & * & * \\ 0 & 0 & -5 & -5 & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \end{pmatrix}$$

$$I \otimes H_x + I \otimes H_y = \begin{pmatrix} 15 & 15 & 0 & -5 & * & * \\ 0 & 0 & -15 & -15 & * & * \\ -15 & -15 & -20 & -15 & * & * \\ -15 & -15 & -15 & -10 & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \end{pmatrix}$$

Kỹ thuật gradient - Bộ lọc Sobel

- ◆ Tương tự như kỹ thuật Prewitt, nhưng kỹ thuật Sobel sử dụng 2 mặt nạ nhân chập theo 2 hướng x, y là:

$$H_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \quad H_y = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}$$

- ◆ Ngoài ra để phát hiện biên theo đường chéo ta sử dụng 2 mặt nạ:

$$H_1 = \begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix}; \quad H_2 = \begin{bmatrix} -1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

Kỹ thuật gradient - La bàn

Kỹ thuật sử dụng 8 bộ lọc theo 8 hướng chính trên la bàn $0^\circ, 45^\circ, 90^\circ, 135^\circ, 180^\circ, 225^\circ, 270^\circ, 315^\circ$

$$H_1 = \begin{pmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{pmatrix} \quad H_2 = \begin{pmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{pmatrix} \quad H_3 = \begin{pmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{pmatrix}$$

$$H_8 = \begin{pmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{pmatrix} \quad H_4 = \begin{pmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{pmatrix}$$

$$H_7 = \begin{pmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{pmatrix} \quad H_6 = \begin{pmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{pmatrix} \quad H_5 = \begin{pmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{pmatrix}$$

Kỹ thuật phát hiện biên Laplace

- Các phương pháp đánh giá gradient ở trên làm việc khá tốt khi mà độ sáng thay đổi rõ nét.
- Khi mức xám thay đổi chậm, miền chuyển tiếp trải rộng, là phương pháp cho hiệu quả hơn là sử dụng đạo hàm bậc hai Laplace.
- Toán tử Laplace được định nghĩa như sau:

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

- Khai triển toán tử rồi áp dụng xấp xỉ rời rạc

Kỹ thuật phát hiện biên Laplace

- ◆ $\frac{\partial^2 f}{\partial x^2} = \frac{\partial}{\partial x} \left(\frac{\partial f}{\partial x} \right) \approx \frac{\partial}{\partial x} (f(x+1, y) - f(x, y))$
 $\approx [(f(x+1, y) - f(x, y))] - [(f(x, y) - f(x-1, y))]$
 $\approx f(x+1, y) - 2f(x, y) + f(x-1, y)$
- ◆ Tương tự ta có
- ◆ $\frac{\partial^2 f}{\partial y^2} \approx f(x, y+1) - 2f(x, y) + f(x, y-1)$
- ◆ $\Delta^2 f \approx f(x+1, y) + f(x-1, y) - 4f(x, y) + f(x, y+1) + f(x, y-1)$

Tương đương với việc sử dụng ma trận

$$H_1 = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix}$$

Kỹ thuật phát hiện biên Laplace

- Thực tế người ta có thể sử dụng một số ma trận biến thế để xấp xỉ rời rạc đạo hàm bậc 2

$$H = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix}$$

$$H = \begin{pmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{pmatrix}$$

$$H = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix}$$

Kỹ thuật phát hiện biên Laplace

- Do toán tử Laplace là toán tử đạo hàm nên nó làm nổi bật các vùng không liên tục của cấp xám, đồng thời làm yếu đi các vùng phẳng.
- Điều này dẫn đến ảnh kết quả sau khi áp dụng toán tử Laplace sẽ không giữ lại được các chi tiết ban đầu của ảnh.
- Do đó, để khôi phục các chi tiết của ảnh gốc, người ta thường cộng ảnh kết quả với ảnh gốc để cho ra ảnh rõ nét (là ảnh vẫn giữ nguyên những chi tiết ban đầu nhưng các cạnh được làm nổi bật lên).
- Nếu toán tử Laplace có hệ số tâm là âm thì chúng ta sẽ lấy ảnh gốc trừ đi ảnh kết quả sau khi áp dụng toán tử Laplace thay vì cộng.

Kỹ thuật phát hiện biên Canny

- Đây là kỹ thuật cổ điển nhưng đến nay vẫn rất hiệu quả và được sử dụng rộng rãi
- Có khả năng đưa ra đường biên mảnh và phát hiện chính xác với ảnh có nhiễu
- Sau khi áp dụng, ảnh có thể được mã hóa thành các đường cong với công thức toán học



Kỹ thuật phát hiện biên Canny

- Bước 1: Làm trơn ảnh (ma trận đầu vào Gaussian): Giúp loại bỏ bớt nhiễu đầu vào
- Bước 2: Tính gradient bằng ma trận Prewitt

$$G_x = B \otimes H_x$$

$$G_y = B \otimes H_y$$

- Có thể dùng các phép phát hiện bậc một khác (gradient đơn giản, Sobel..)
- Kết quả là 2 ảnh gradient theo hai hướng x và y

$$B = I \otimes H$$

$$H = \frac{1}{159} \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix}$$

Kỹ thuật phát hiện biên Canny

- Bước 3: Tính gradient hướng tại mỗi điểm (i, j)
 - Hướng sẽ được “nguyên hoá” để nằm trong 8 hướng [0...7]
 - Tương đương 8 điểm lân cận của 1 điểm ảnh
- Bước 4: Loại bỏ những điểm không phải là cực đại để xóa bỏ những điểm không phải là biên (tức loại bỏ 1 số cạnh dư thừa)
- Xét (i, j), 0 là gradient hướng tại (i, j), G_1 , G_2 là hai điểm lân cận theo hướng θ .
- Nếu $G(i,j) > G_1$, và $G(i,j) > G_2$ thì mới giữ lại (i,j) ((i,j) là cực đại phương)
- Ngược lại thì xóa (i,j) vì (i,j) là điểm nền

$$G = \sqrt{G_x^2 + G_y^2}$$
$$\Theta = \arctan\left(\frac{G_y}{G_x}\right)$$

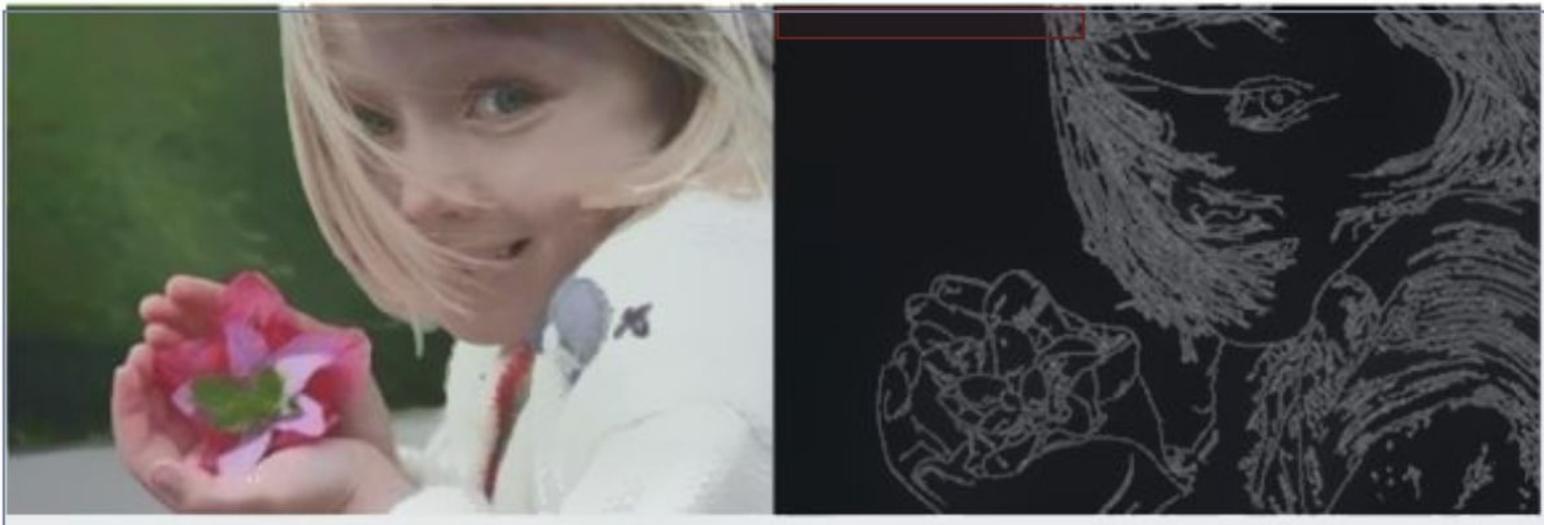
Kỹ thuật phát hiện biên Canny

Bước 5: Phân ngưỡng để tìm biên

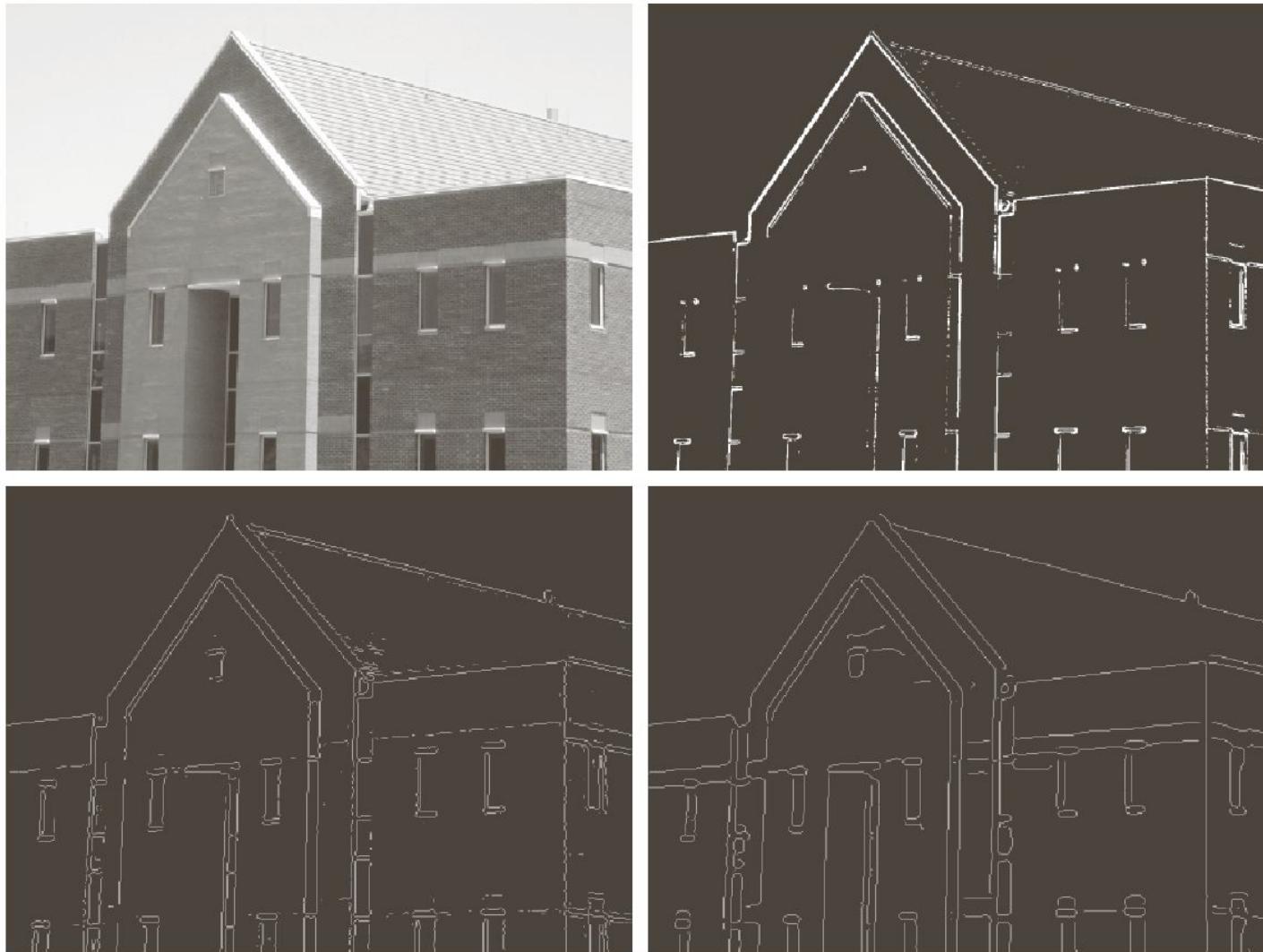
- Điểm có gradient lớn hơn thường có khả năng là biên cao hơn điểm có gradient nhỏ hơn
- Việc chọn ngưỡng để phân loại là rất khó.
- Canny sử dụng phân ngưỡng với độ trễ
- Có hai ngưỡng cao và thấp
- Giả định là biên quan trọng thường nằm trong những đường liên tục trong bức ảnh.

Kỹ thuật phát hiện biên Canny

- Nếu $I(x,y) >$ ngưỡng cao thì giữ lại điểm biên này
- Nếu $I(x,y) <$ ngưỡng thấp thì loại bỏ điểm này.
- Nếu ngưỡng thấp $< I(x,y) <$ ngưỡng cao thì so sánh $I(x,y)$ với giá trị của 8 điểm lân cận. Nếu 1 trong 8 điểm lân cận $>$ ngưỡng cao thì ta giữ lại điểm biên này. Ngược lại thì bỏ điểm biên này.



Kỹ thuật phát hiện biên Canny



a
b
c
d

FIGURE 10.25

(a) Original image of size 834×1114 pixels, with intensity values scaled to the range $[0, 1]$.

(b) Thresholded gradient of smoothed image.

(c) Image obtained using the Marr-Hildreth algorithm.

(d) Image obtained using the Canny algorithm. Note the significant improvement of the Canny image compared to the other two.

Kỹ thuật phát hiện biên cục bộ

- Là phương pháp lọc phát hiện biên dựa vào trung bình cục bộ
- Xác định biên không theo sự biến đổi mà dựa vào trung bình giá trị các điểm lân cận
- Với cửa sổ mxn với tâm là (i,J) thì nếu

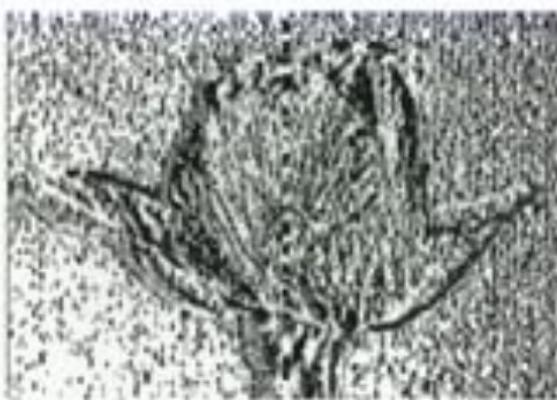
$$\frac{\sum W(i,j)}{m * n} > I(i,j) + \delta$$

→ điểm ảnh I(i,j) sẽ là điểm biên và ngược lại sẽ là điểm nền

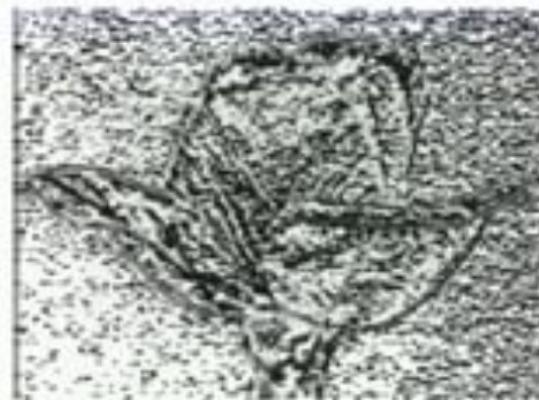
Áp dụng bộ lọc cho các ảnh



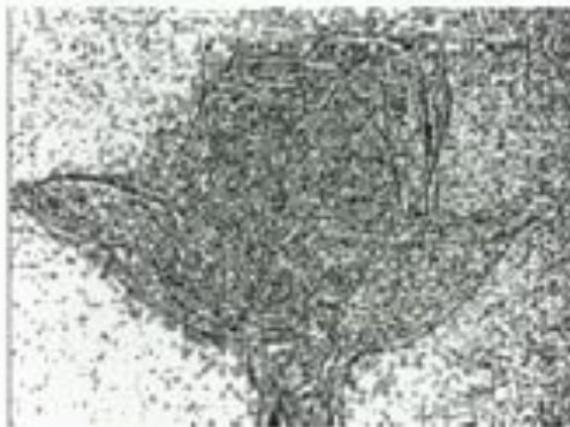
a) Ảnh gốc



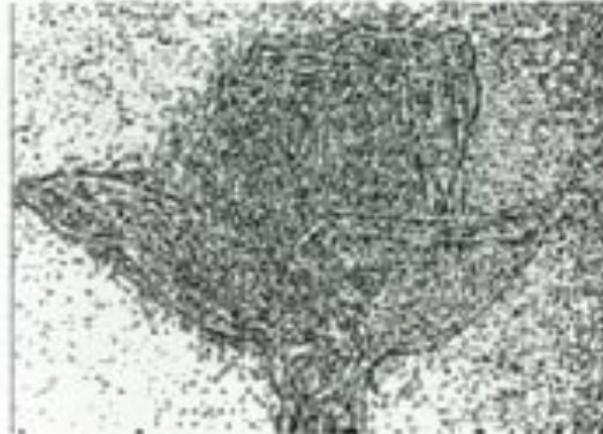
b) Ảnh qua lọc Sobel Hx



c) Ảnh qua lọc Sobel Hy



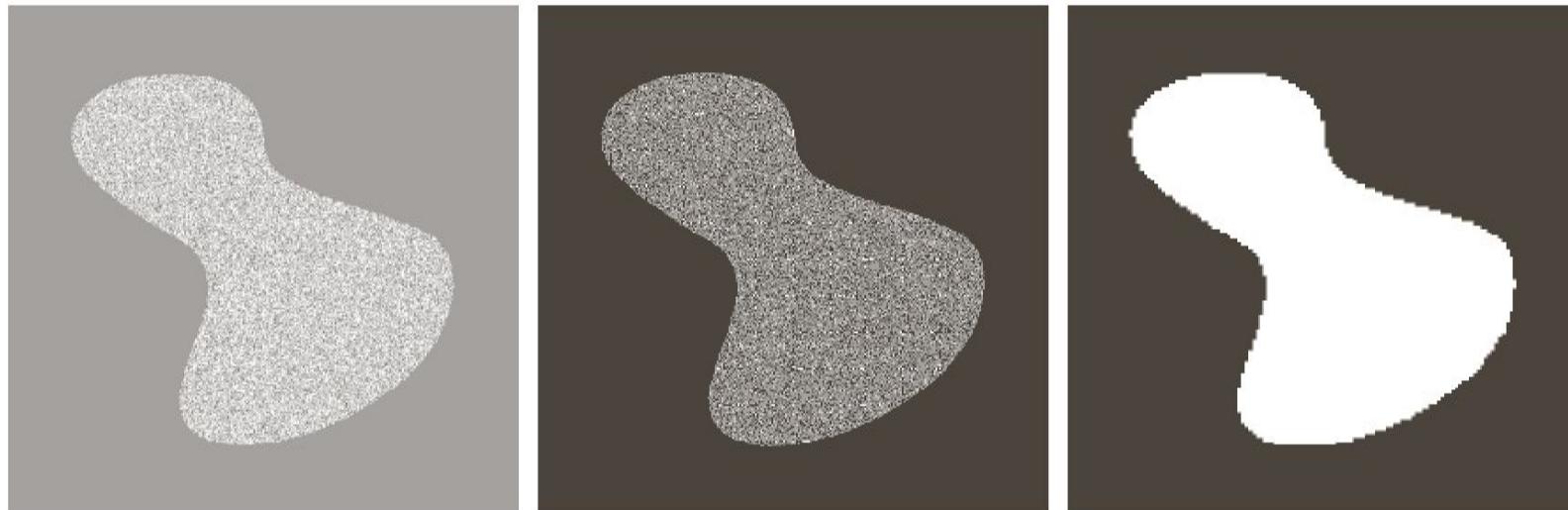
d) Ảnh qua lọc Kirsh



e) Ảnh qua lọc Laplace

Các phương pháp phát hiện biên gián tiếp

- ◆ Phân vùng ảnh dựa vào phép xử lý kết cấu đối tượng, cụ thể là dựa vào sự biến thiên nhỏ và đồng đều của các điểm ảnh thuộc một đối tượng.
- ◆ Dựa trên các vùng, đòi hỏi áp dụng lý thuyết về xử lý kết cấu đối tượng phức tạp và khó cài đặt.



Các điểm 4 và 8-láng giềng

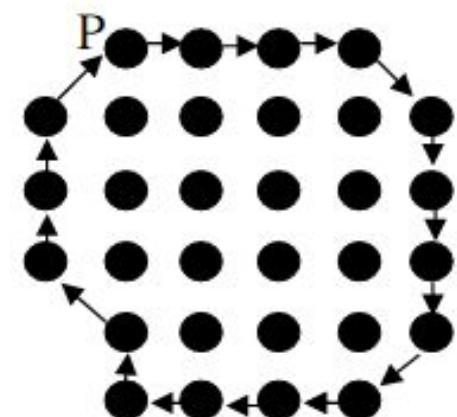
- Với điểm P thì
 - Các điểm P_0, P_2, P_4, P_6 (trên, dưới, phải, trái) là các 4-láng giềng của điểm P
 - Các điểm $P_0, P_1, P_2, P_3, P_4, P_5, P_6, P_7$ (các điểm ở cả 8 hướng) là các 8-láng giềng của P .
-

P_3	P_2	P_1
P_4	P	P_0
P_5	P_6	P_7

Chu tuyến của một đối tượng ảnh

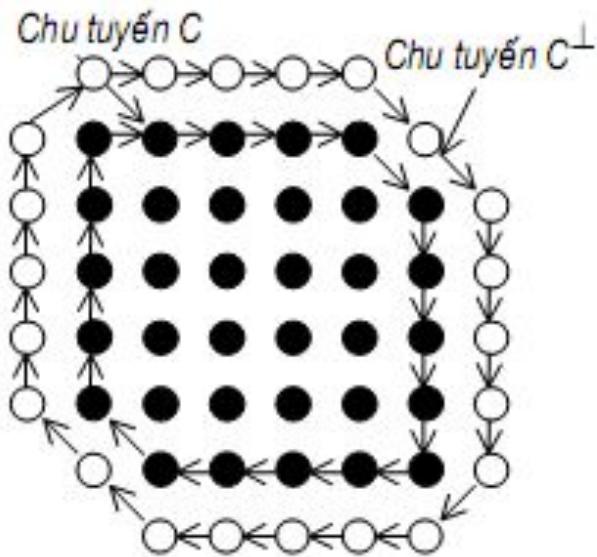
- ◆ Chu tuyến của một đối tượng ảnh là dãy các điểm của đối tượng ảnh P_1, \dots, P_n sao cho
 - P_i và P_{i+1} là các 8-láng giềng của nhau ($i=1, \dots, n-1$)
 - P_1 là 8-láng giềng của P_n ,
 - $\forall i \exists Q$ không thuộc đối tượng ảnh và Q là 4-láng giềng của P_i (hay nói cách khác $\forall i$ thì P_i là biên 4). Kí hiệu $\langle P_1 P_2 \dots P_n \rangle$

Hình bên biểu diễn chu tuyến của ảnh, trong đó, P là điểm khởi đầu chu tuyến.

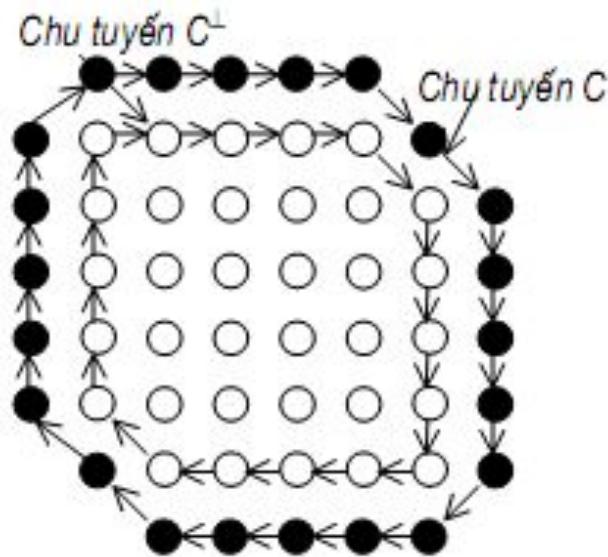


Khái niệm đường bao trong ảnh

Đường bao ngoài



Đường bao trong



Các thuật toán dò biên trên một vùng đều bao gồm các bước sau:

- (1) Xác định điểm biên xuất phát
- (2) Dự báo và xác định điểm biên tiếp theo
- (3) Lặp bước 2 cho đến khi gặp điểm xuất phát

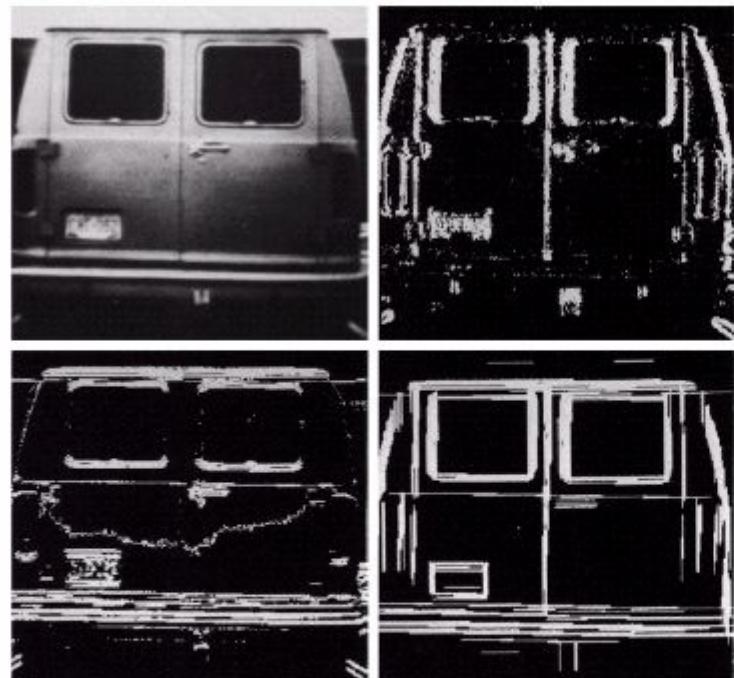
- ◆ Bước 1: Xác định cặp nền-vùng (đen-trắng) xuất phát:
Duyệt ảnh lần lượt từ trên xuống dưới và từ trái sang phải
điểm đem đầu tiên gặp được cùng với điểm trắng trước
đó (theo hướng 4) để tạo nền cặp nền vùng xuất phát
- ◆ Bước 2: Xác định cặp nền-vùng tiếp theo
- ◆ Bước 3: Lựa chọn điểm biên vùng
- ◆ Bước 4: Nếu gặp lại cặp xuất phát thì dừng, nếu không
quay lại bước 2.

3. Phân tích đặc điểm hình dạng dựa trên biên

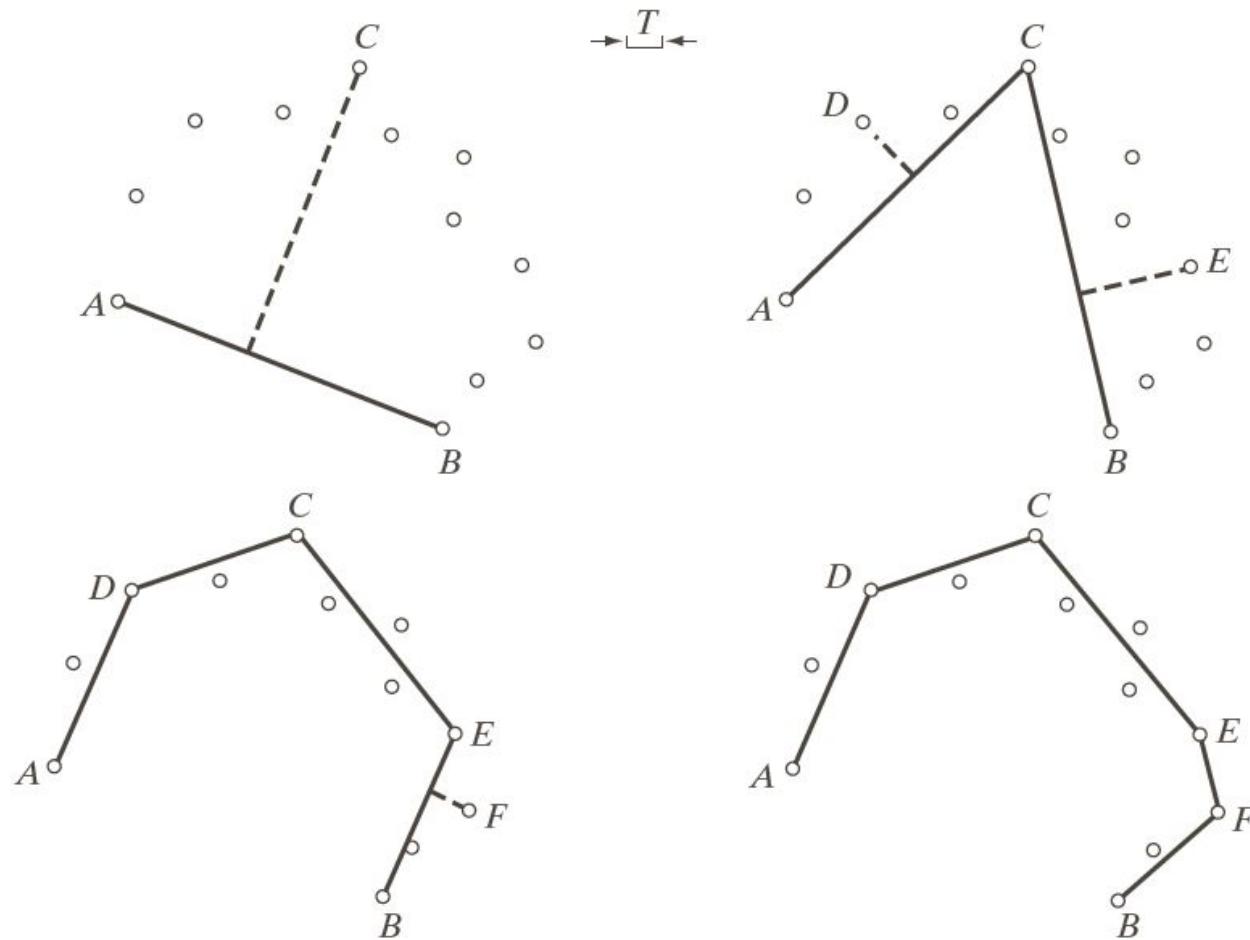
- ◆ Giả sử các điểm biên đã được phát hiện
- ◆ Đường biên phân tách đối tượng (khỏi nền) được hình thành trên cơ sở liên kết các điểm biên (edge Linking). Một số thuật toán :
 - Thông qua xử lý cục bộ
 - Xử lý ở mức vùng
 - Xử lý toàn cục
- ◆ Trích chọn các đặc trưng trên đường biên (bao) phân tích đặc điểm hình khối của đối tượng

Thông qua xử lý cục bộ

- ◆ Giả sử các điểm biên đã được phát hiện
- ◆ Phân tích đặc điểm của điểm biên và các điểm lân cận
- ◆ Các điểm có độ giống nhau cao (dựa trên biên độ và hướng của gradient) liên kết với nhau



Thông qua xử lý ở mức vùng



a	b
c	d

FIGURE 10.28
Illustration of the
iterative
polygonal fit
algorithm.

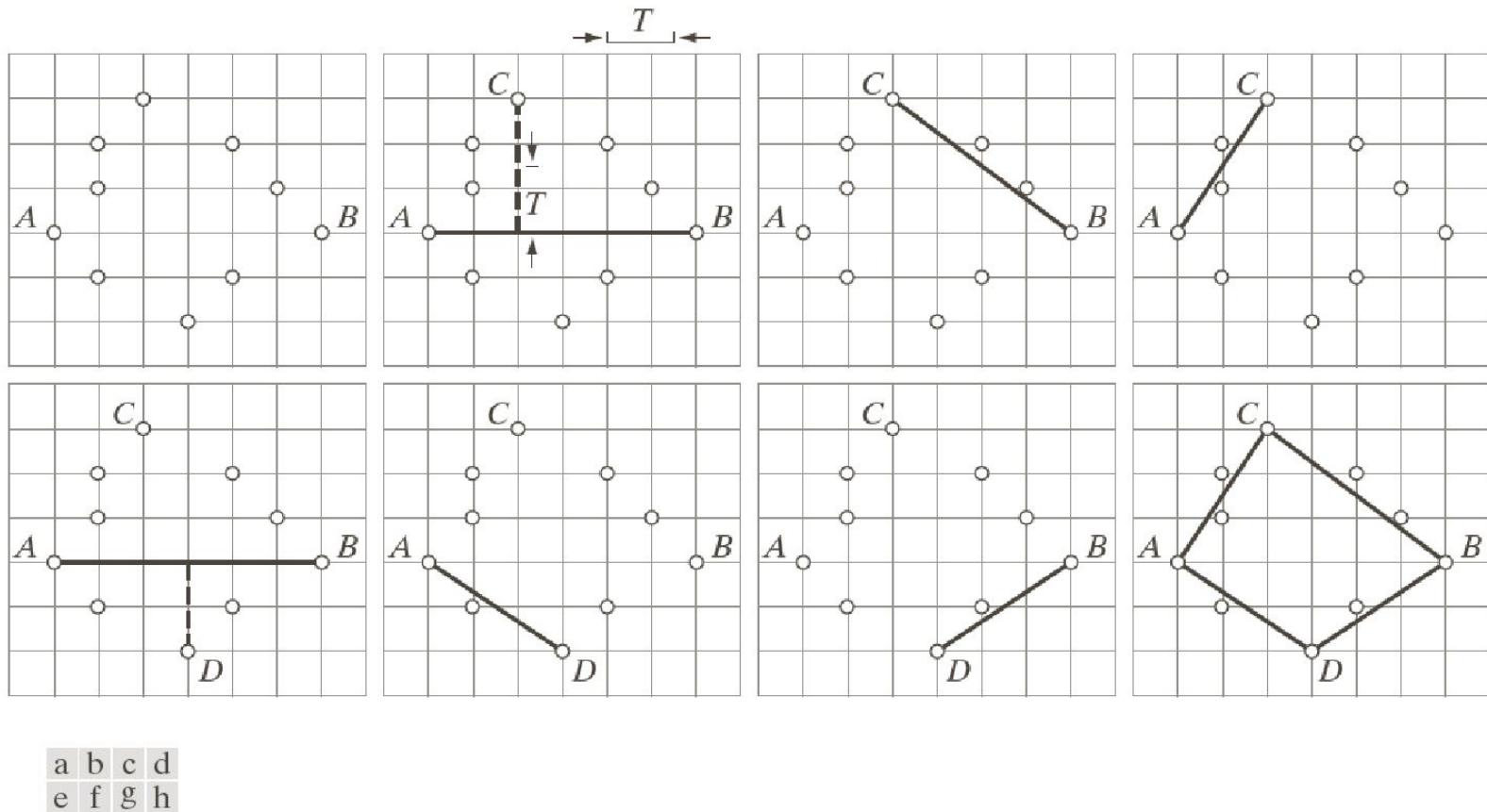
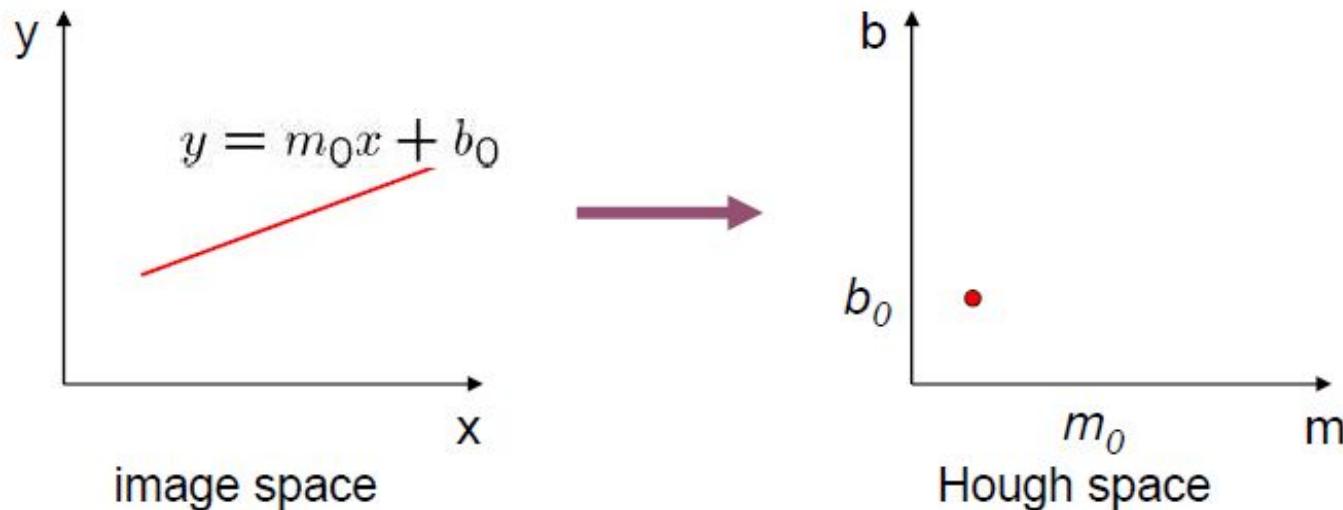


FIGURE 10.29 (a) A set of points in a clockwise path (the points labeled A and B were chosen as the starting vertices). (b) The distance from point C to the line passing through A and B is the largest of all the points between A and B and also passed the threshold test, so C is a new vertex. (d)–(g) Various stages of the algorithm. (h) The final vertices, shown connected with straight lines to form a polygon. Table 10.1 shows step-by-step details.

Thông qua xử lý toàn cục

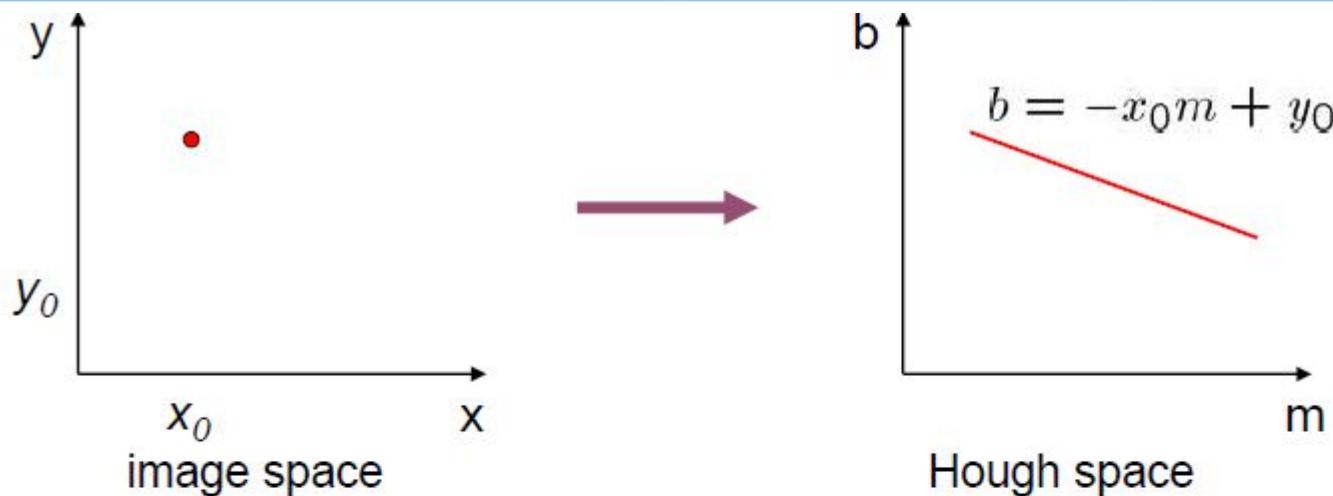
- ◆ Giả sử đã có kết quả phát hiện biên (là các pixel rì rạc)
- ◆ Biết được phương trình hình học của đối tượng cần tìm (đường thẳng, đường tròn, đường ellipise)
- ◆ Thực hiện các thuật toán fitting model (làm khớp)
 - Hough transform
 - RANSAC

Thuật toán Hough transform



- Mỗi quan hệ giữa không gian ảnh (x, y) và không gian Hough (m, b)
 - Một đường thẳng trong ảnh tương ứng với một điểm trong không gian Hough
 - Để chuyển từ không gian ảnh sang không gian Hough:
 - cho một tập hợp các điểm (x, y), tìm tất cả các (m, b) sao cho $y = mx + b$.

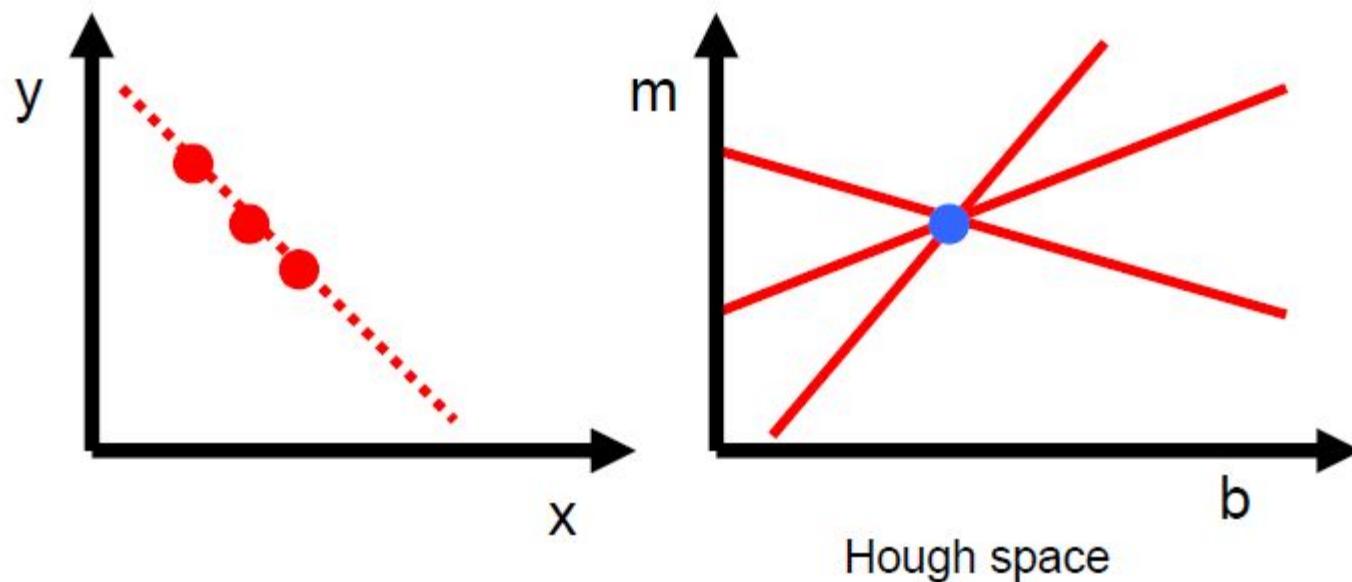
Thuật toán Hough transform



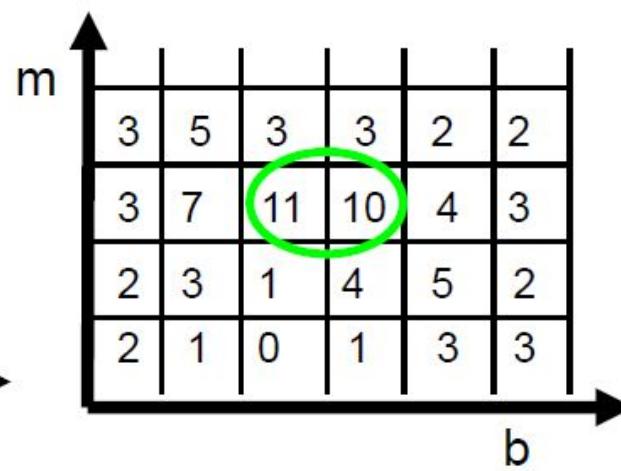
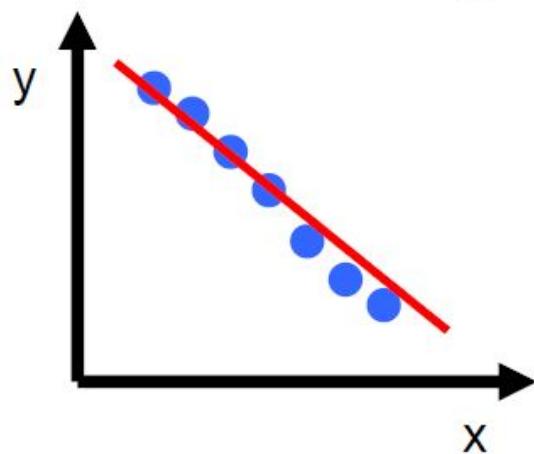
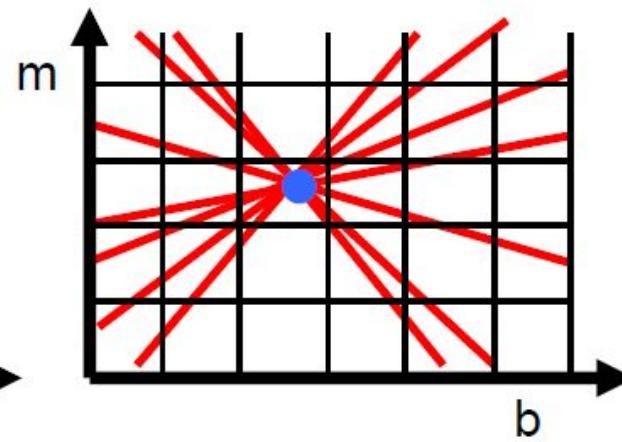
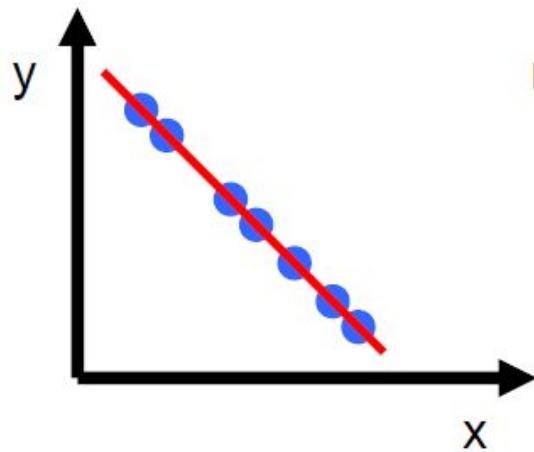
- Mối liên kết giữa không gian ảnh (x, y) và không gian Hough (m, b)
 - Một đường thẳng trong ảnh tương ứng với một điểm trong không gian Hough
 - Để chuyển từ không gian ảnh sang không gian Hough:
 - cho một tập hợp các điểm (x, y) , tìm tất cả các (m, b) sao cho $y = mx + b$
 - Một điểm (X, Y_0) trong không gian ảnh được ánh xạ vào đâu?
 - A: các nghiệm của phương trình $b = -Xom + Y_0$
 - đây là một đường trong không gian Hough

Thuật toán Hough transform

Tìm đường thẳng tốt nhất qua 1 tập hợp các điểm

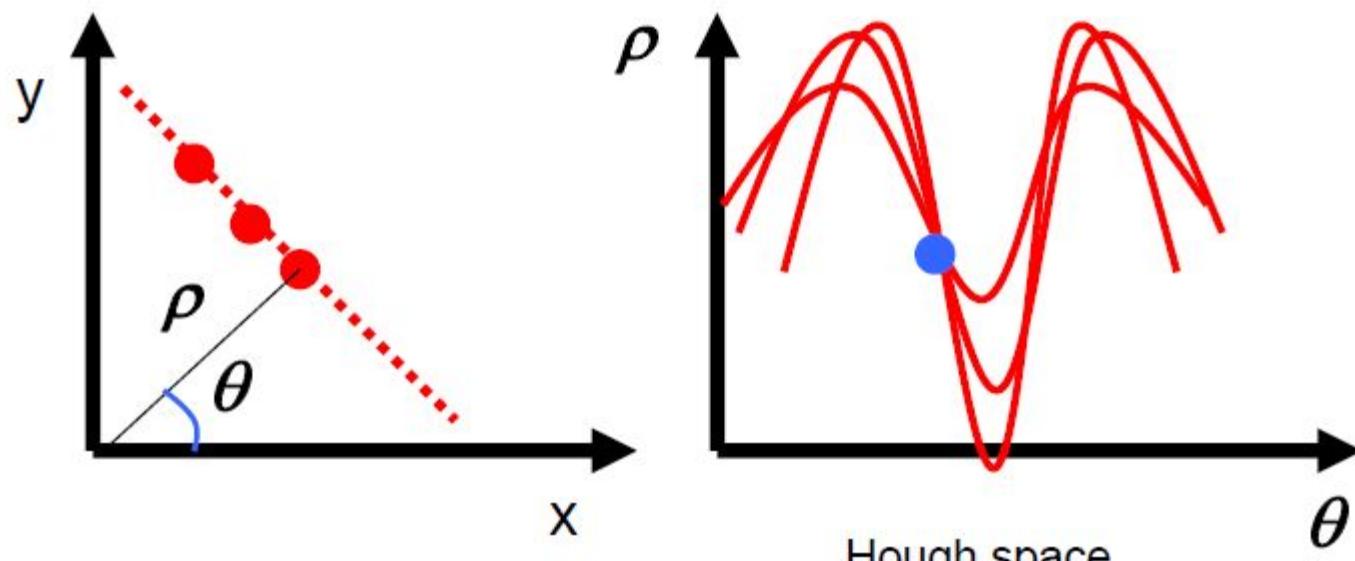


$$y = m x + b$$



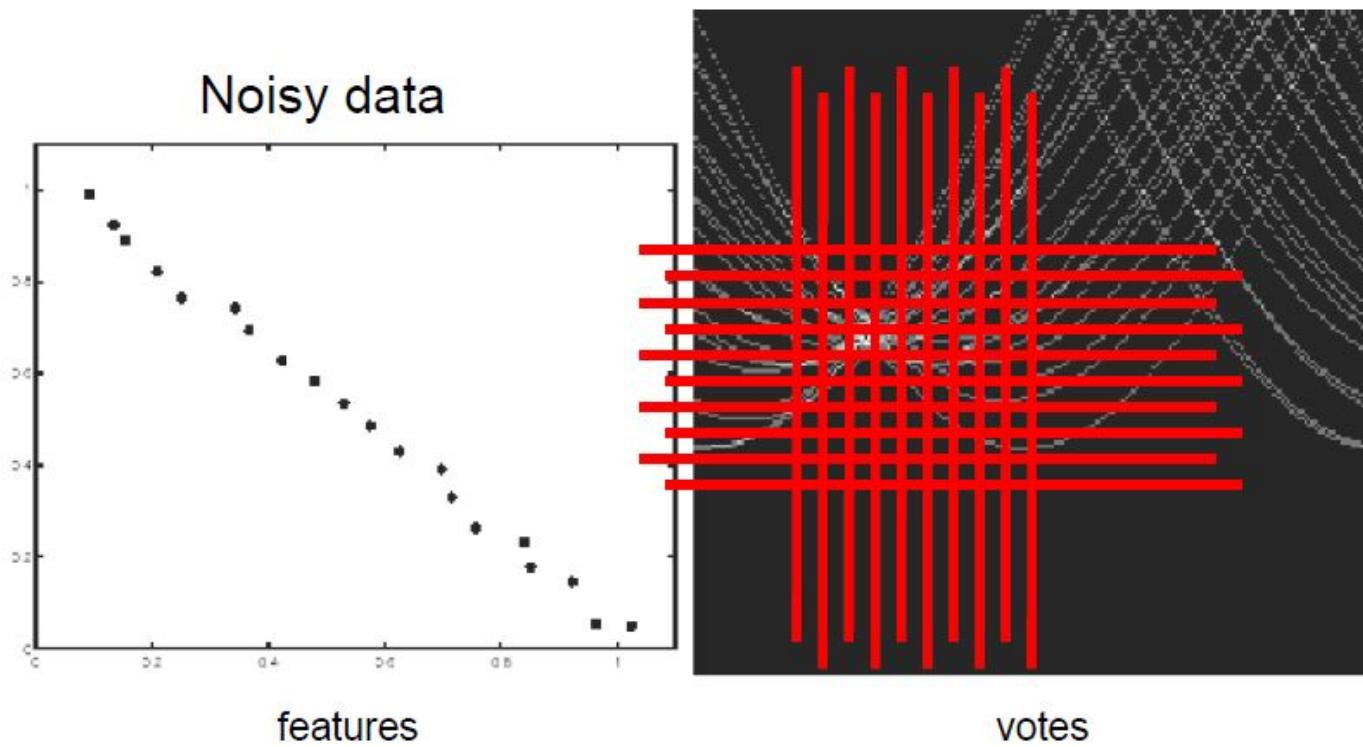
Vấn đề: các tham số [m, b] không bị giới hạn

→ Đổi sang hệ toạ độ cực



$$x \cos \theta + y \sin \theta = \rho$$

Thuật toán Hough transform



Vấn đề về nhiễu trong tập hợp điểm

Ví dụ minh họa

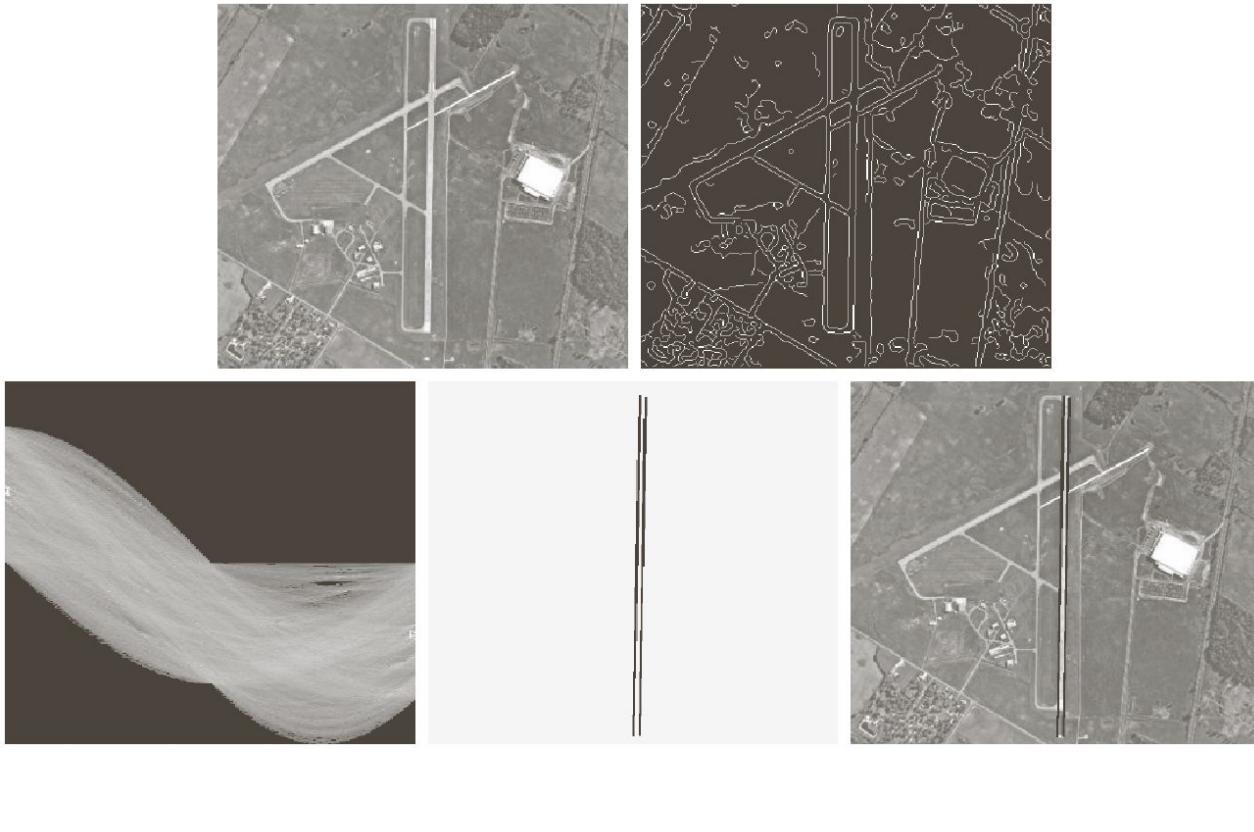


FIGURE 10.34 (a) A 502×564 aerial image of an airport. (b) Edge image obtained using Canny's algorithm. (c) Hough parameter space (the boxes highlight the points associated with long vertical lines). (d) Lines in the image plane corresponding to the points highlighted by the boxes). (e) Lines superimposed on the original image.

Nhận xét về Hough Transform

Lợi:

- Ổn định với các giá trị ngoại lai: Biến đổi Hough là ổn định với các giá trị ngoại lai vì mỗi điểm bầu chọn độc lập, và đóng góp của các điểm riêng lẻ có thể được kết hợp để tìm ra những đường thẳng hoặc hình dạng phù hợp nhất.
- Khá hiệu quả: Biến đổi Hough thường nhanh hơn so với việc thử tất cả các bộ tham số, làm cho nó là một phương pháp hiệu quả để phát hiện đường và hình dạng trong hình ảnh.

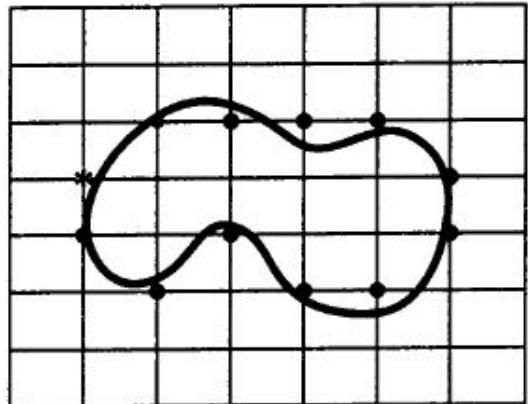
Bất lợi:

- Nhạy cảm với nhiễu: Biến đổi Hough có thể nhạy cảm với nhiễu trong hình ảnh, dẫn đến việc phát hiện sai hoặc không chính xác kết quả.
- Không thích hợp cho nhiều tham số:
 - Biến đổi Hough trở nên ít thực tế khi xử lý nhiều hơn một số tham số (ví dụ: hình dạng có số chiều cao) vì kích thước lưới yêu cầu cho không gian tham số tăng theo cấp số nhân, dẫn đến phức tạp tính toán.

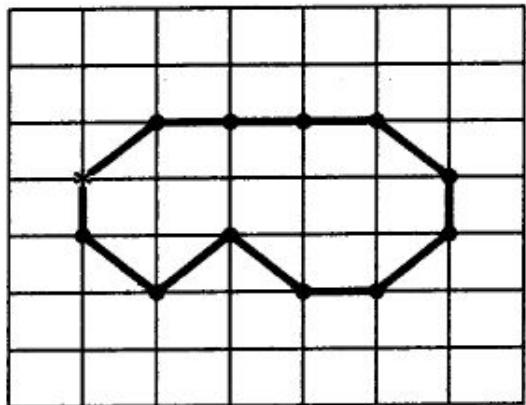
Ứng dụng phổ biến:

- Đường thẳng, đường tròn, eclipse...
- Nhận diện một số đối tượng trong hình ảnh
- Nhận dạng loại đối tượng

Biểu diễn đường bao

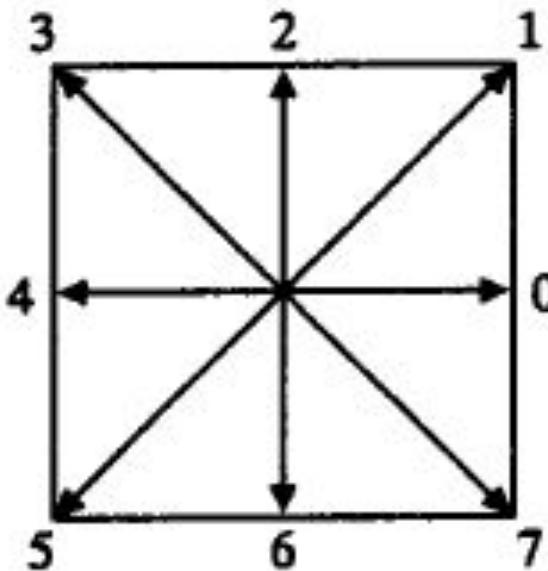


(a) Original curve with grid intersect quantization points.
The asterisk (*) marks the starting point of the curve.



(b) Links associated with the quantization points of (a).

◆ Quy tắc Encode các hướng



(c) Encoding scheme.

100076543532

(d) Chain code for the curve of (a).

Phát hiện góc

- Ngoài các biên thì corners (góc) và interest point (điểm quan tâm) cũng cần được quan tâm
- Một trong những thuật toán mạnh mẽ trong việc tìm kiếm các corners cũng như các image features nhằm phục vụ cho các bài toán như image matching, image stitching for panorama,... là thuật toán của Harris (Harris corner detector - HCD)
- Được phát triển lần đầu vào năm 1988, Chris Harris and Mike Stephens dựa vào sự biến đổi cường độ sáng tại vùng lân cận để phát hiện các góc cạnh



Harris corner detection

- Thuật toán Harris phát biểu: một vùng nhỏ xung quanh các đặc trưng sẽ có 1 sự thay đổi lớn về cường độ sáng nếu một window dịch chuyển 1 đoạn (u, v) từ điểm (x, y) theo bất kì hướng nào.
- Chúng ta có công thức tính lượng thay đổi cường độ sáng khi dịch chuyển 1 đoạn $[u, v]$ như sau:

Trong đó:

W: Cửa sổ w có thể coi = 1
để cho đơn giản trong việc
tính toán nhưng để chuẩn
xác hơn cho kết quả đầu ra
ta có thể coi đây như một
mặt nạ Gauss($3 \times 3, 5 \times 5, \dots$)

$$E(u, v) = \sum_{x, y} w(x, y) [I(x+u, y+v) - I(x, y)]^2$$

Window function

Shifted intensity

Intensity

$I(x, y)$: Cường độ sáng tại điểm x, y

$I(x + u, y + v)$: Cường độ sáng sau khi dịch chuyển cửa sổ đến điểm $(x + u, y + v)$

Harris corner detection

- Ở đây ta sẽ áp dụng biến đổi Taylor để rút gọn lại biểu thức, ta có:
 - $f(x + u, y + v) \approx f(x, y) + ufx(x, y) + vfy(x, y)$

- Suy ra

$$\sum [I(x+u, y+v) - I(x, y)]^2 \approx \sum [I(x, y) + uI_x + vI_y - I(x, y)]^2 \quad \text{First order approx}$$

$$= \sum u^2 I_x^2 + 2uvI_xI_y + v^2 I_y^2$$

$$= \begin{bmatrix} u & v \end{bmatrix} \begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \quad \text{Rewrite as matrix equation}$$

$$= \begin{bmatrix} u & v \end{bmatrix} \left(\sum \begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix} \right) \begin{bmatrix} u \\ v \end{bmatrix}$$

Harris corner detection

- Đến đây ta coi

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

- Như vậy cuối cùng ta có sự thay đổi cường độ sáng:

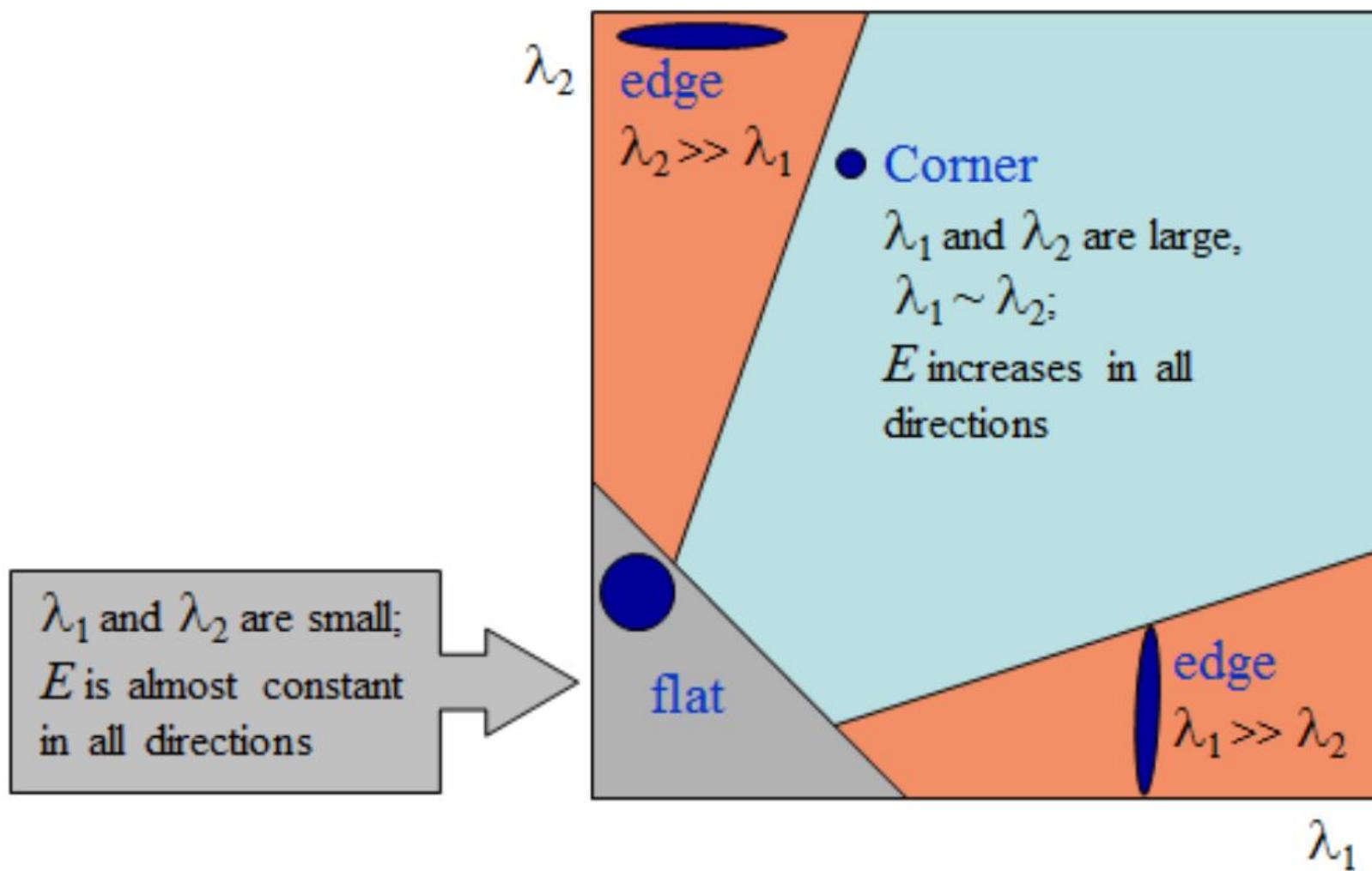
$$E(u, v) \equiv [u, v] M \begin{bmatrix} u \\ v \end{bmatrix}$$

- Như vậy cuối cùng ta có sự thay đổi cường độ sáng:

$$R = \det M - k (\operatorname{trace} M)^2$$

- Trong đó: $\det(M) = AB - C^2 = \lambda_1 \lambda_2$, $\operatorname{trace}(M) = \lambda_1 + \lambda_2$, λ_1, λ_2 là các giá trị bản địa của M tương ứng với độ cong của hàm địa phương quyết định xem điểm này có phải là corners hay không.

Harris corner detection

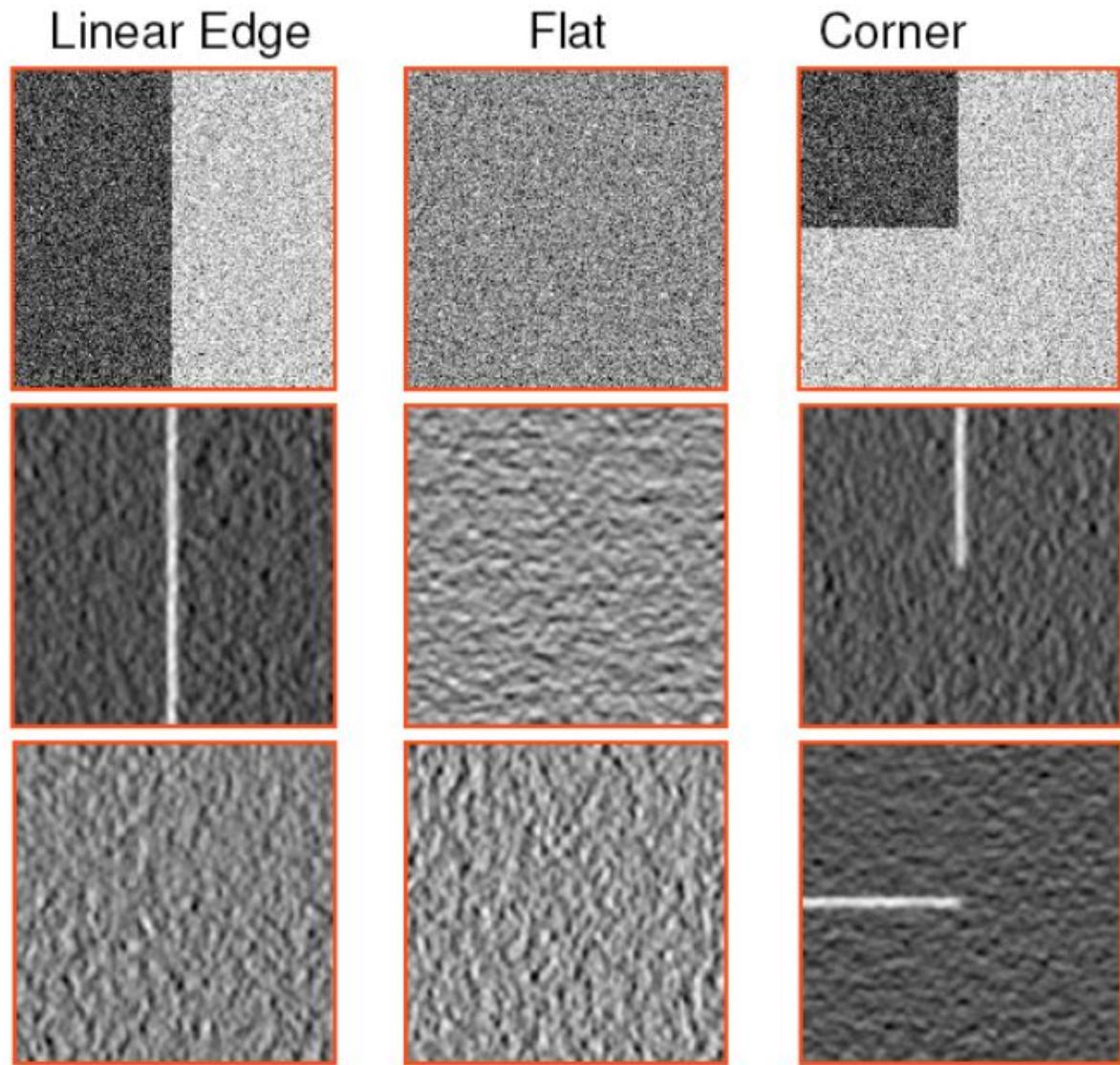


Harris corner detection

Đạo hàm 1 chiều 1
tập các điểm (dx ,
 dy) ta thu được:

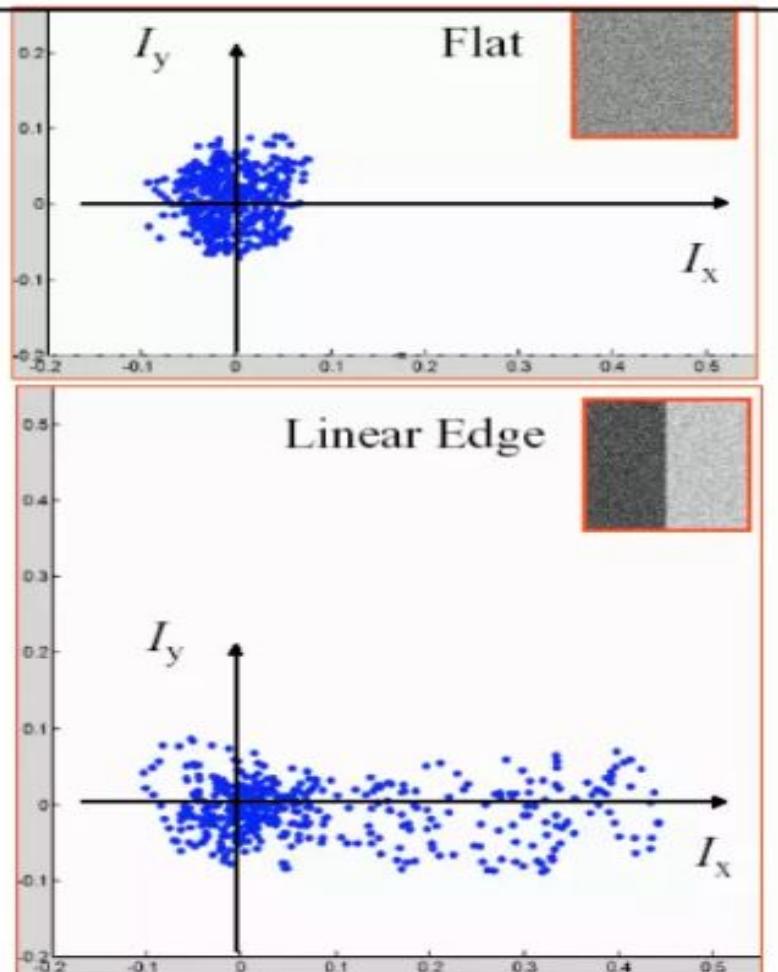
Ở đây ta có thể
nhận ra được sự
khác nhau giữa các
image features theo
các hướng đạo
hàm bậc 1 và đạo
hàm bậc 2.

Y derivative X derivative Input image patch



Harris corner detection - giải thích

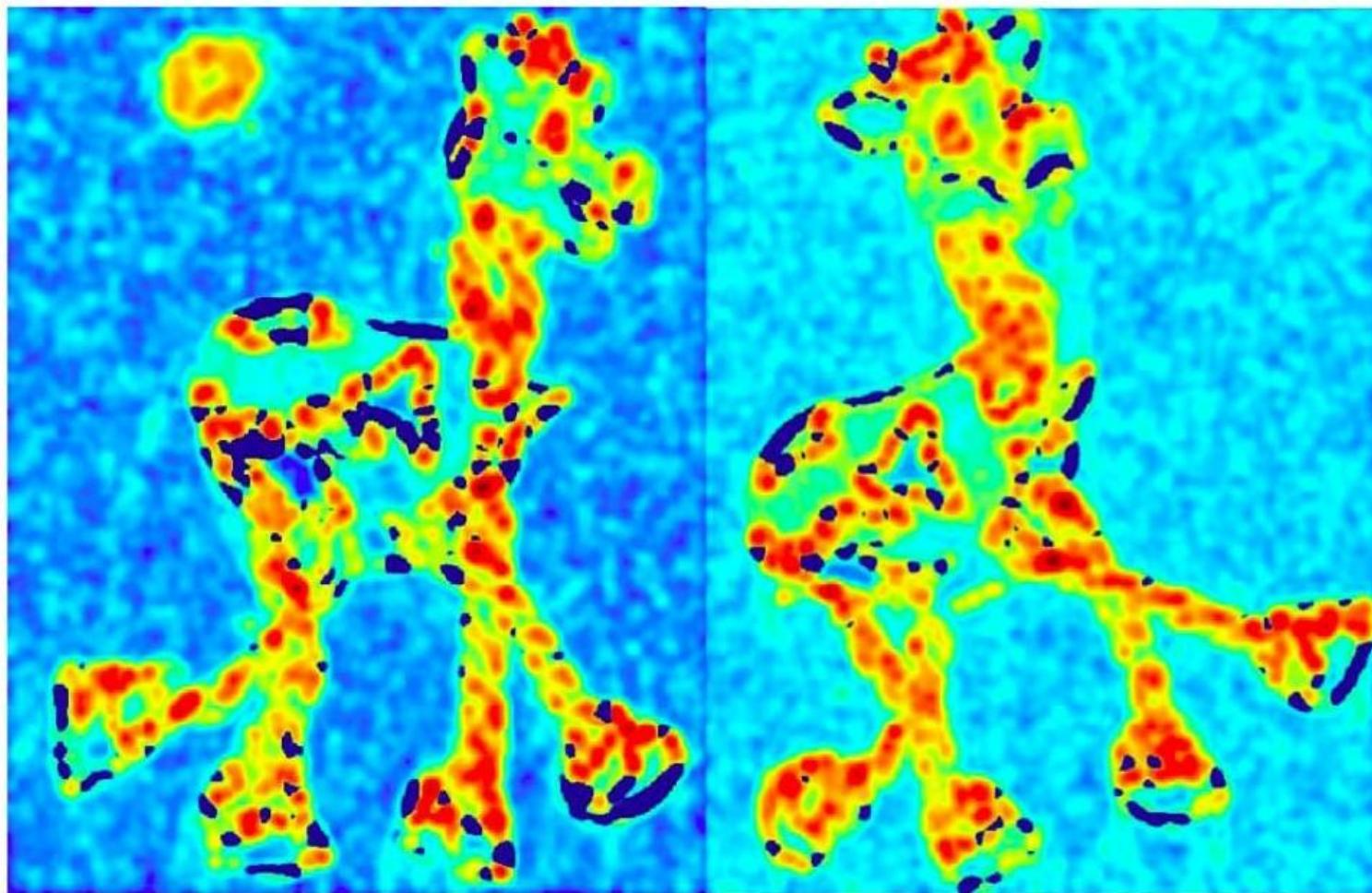
The distribution of the x and y derivatives is very different for all three types of patches



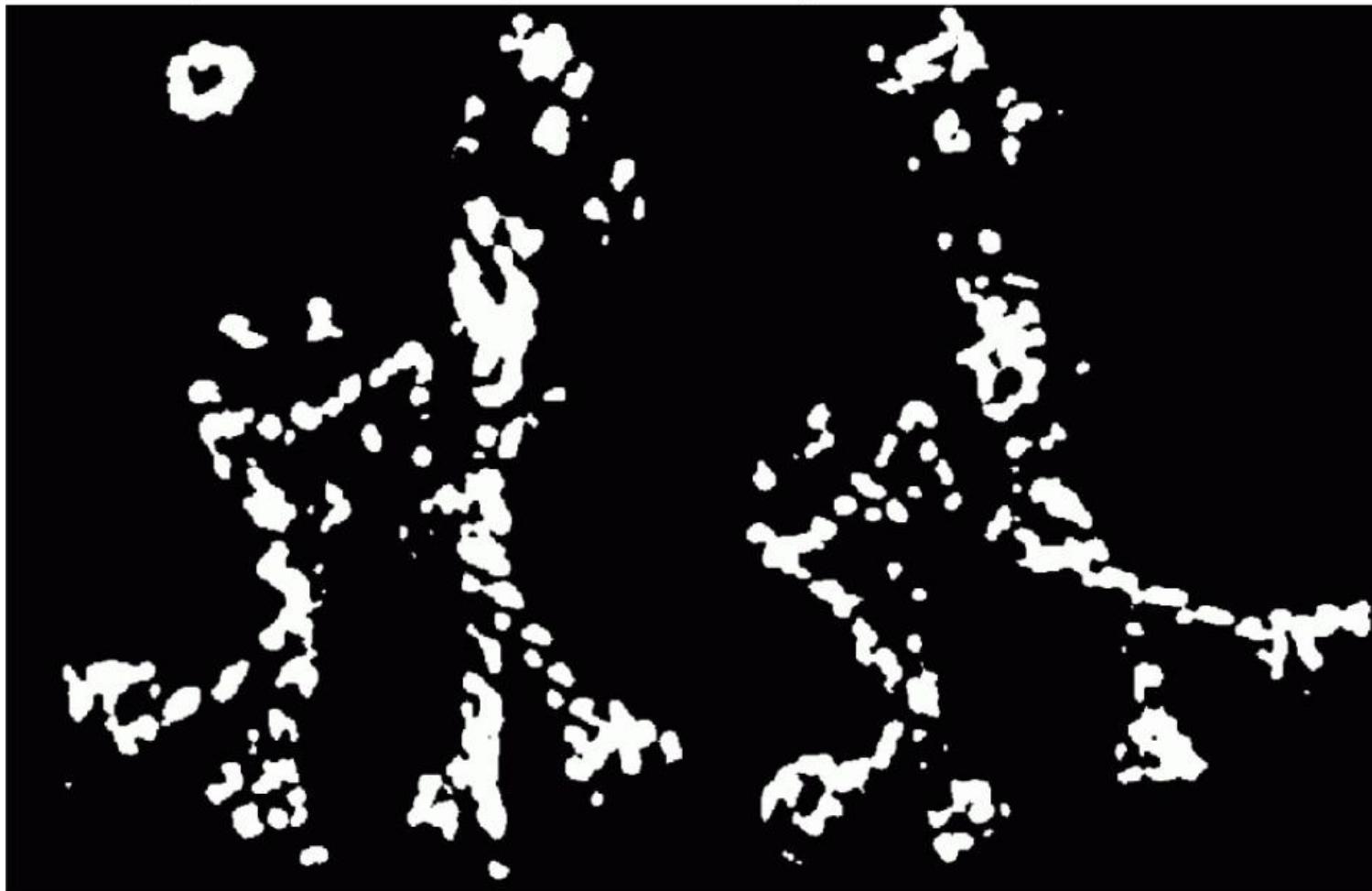
Ví dụ minh họa



Giá trị cornerness score



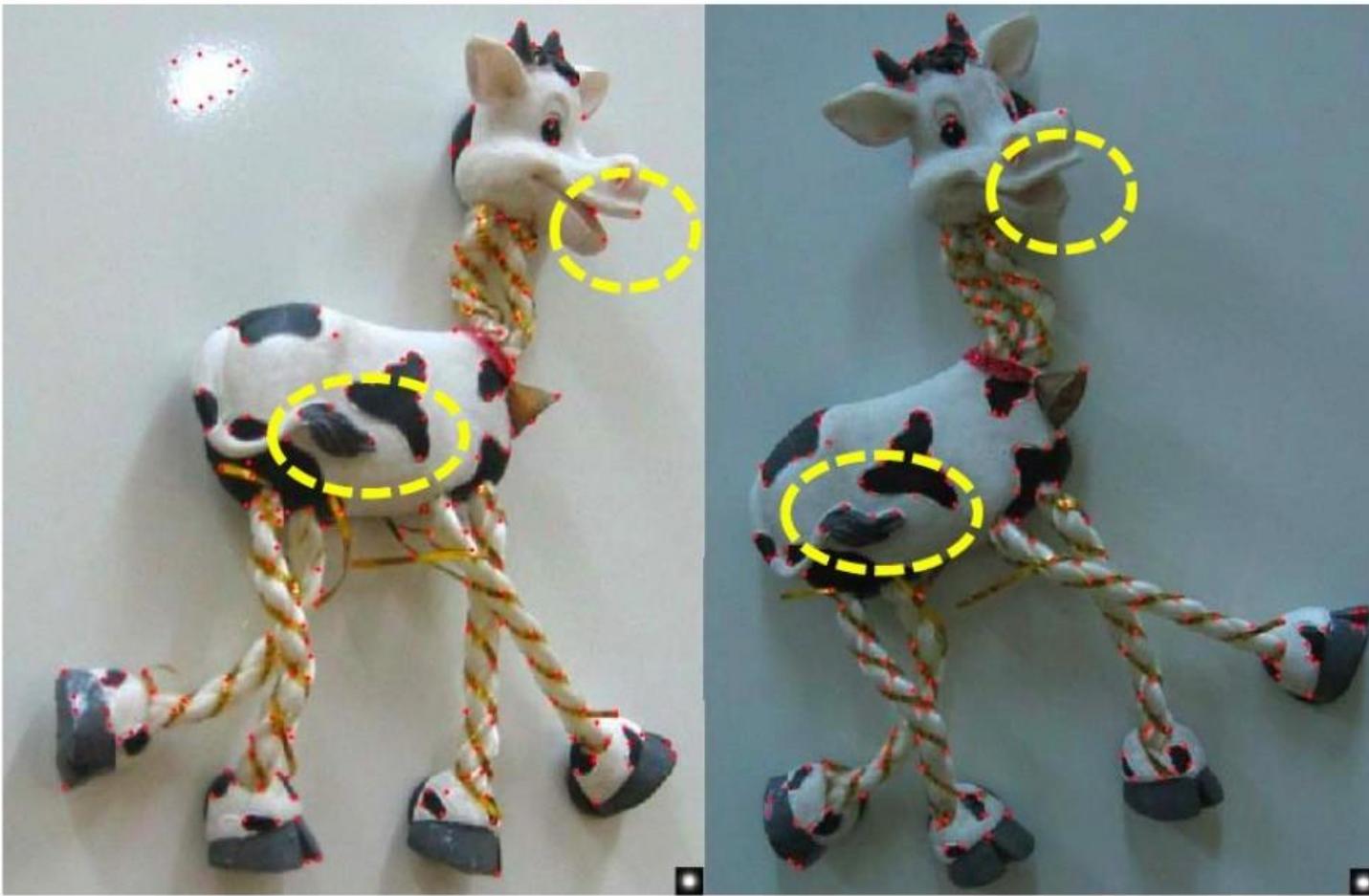
Các điểm local maximal



Các điểm góc sau NMS



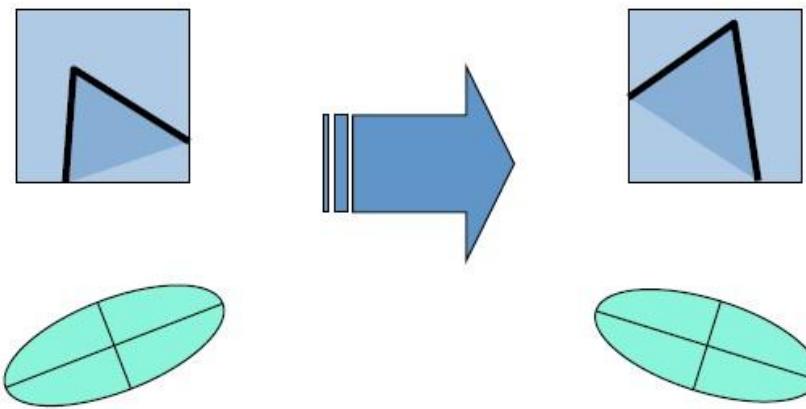
Kết quả các điểm harris corner



Đặc điểm của Harris corner

- Rotation invariant?

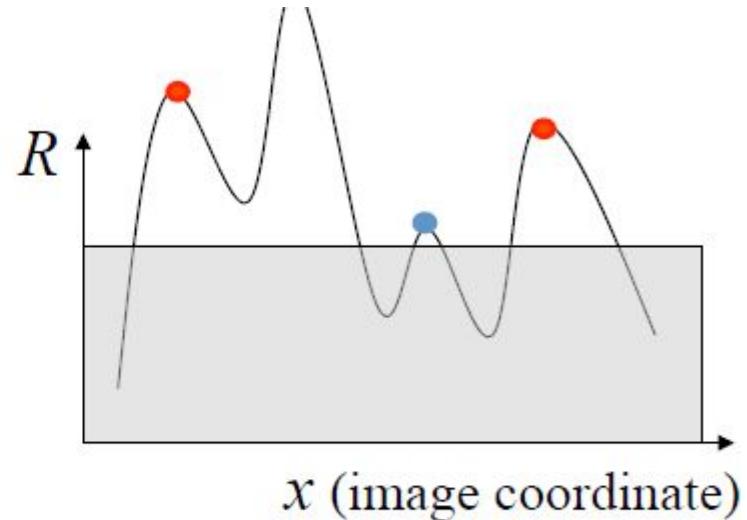
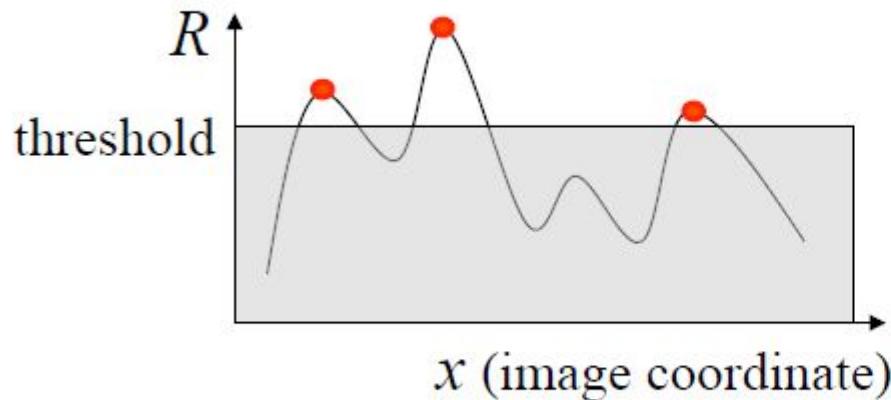
$$\mathbf{A} = w * \begin{bmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{bmatrix} = \mathbf{U} \begin{bmatrix} \lambda_0 & 0 \\ 0 & \lambda_1 \end{bmatrix} \mathbf{U}^T \quad \text{with} \quad \mathbf{A}\mathbf{u}_i = \lambda_i \mathbf{u}_i$$



Ellipse rotates but its shape (i.e. eigenvalues)
remains the same

Đặc điểm của Harris corner

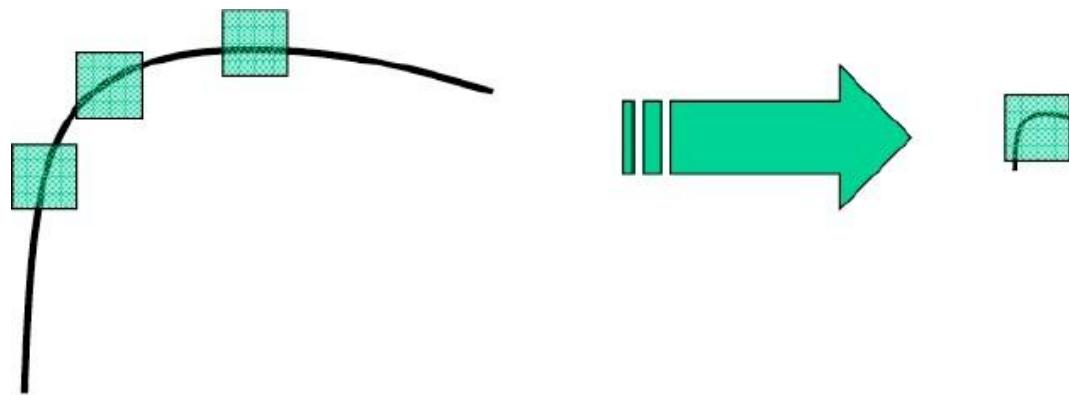
- Bất biến một phần theo cường độ điểm ảnh



Partially invariant to affine intensity change

Đặc điểm của Harris corner

- Không bắt biến theo scale (tỉ lệ)



All points will be
classified as **edges**

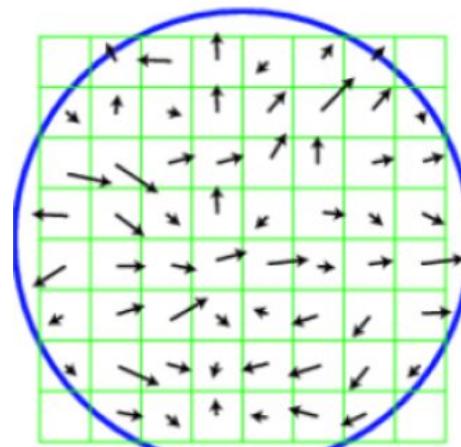
Corner !

Phát hiện/ tìm features

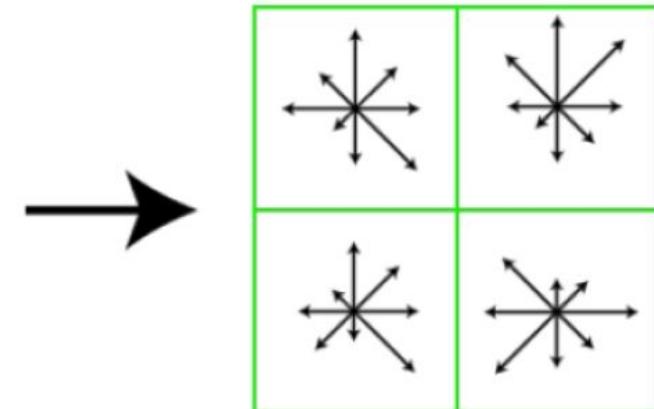
- Sau khi đã tìm được góc, cạnh hay các điểm cần quan tâm, điều tiếp theo cần làm là biểu diễn nó.
- Với bài toán ghép ảnh, ta cần tìm những điểm chung trong 2 tấm ảnh để ghép chúng vào
- Việc biểu diễn các điểm quan tâm theo cách đơn giản và thông minh sẽ giúp các mô hình ghép ảnh rất nhiều
- Một số cách biểu diễn/miêu tả điểm quan tâm:
 - Sift
 - Hog
 - Suft
 - ...

Biểu diễn SIFT (Scale-Invariant Feature Transform)

- Điểm đặc biệt của SIFT nằm ngay trong cái tên của nó Scale-Invariant, tức là nó sẽ đưa ra các kết quả ổn định với những scale của ảnh khác nhau, bên cạnh đó cũng có thể nói giải thuật này có tính rotation-invariant.
- Trong bài báo khoa học "Distinctive Image Features from Scale-Invariant Keypoints", SIFT được đặc tả rõ nét dưới bốn giai đoạn(stages) chính sau:
 - ❖ Scale-space extrema detection.
 - ❖ Keypoint localization.
 - ❖ Orientation assignment.
 - ❖ Keypoint descriptor.



(a) image gradients

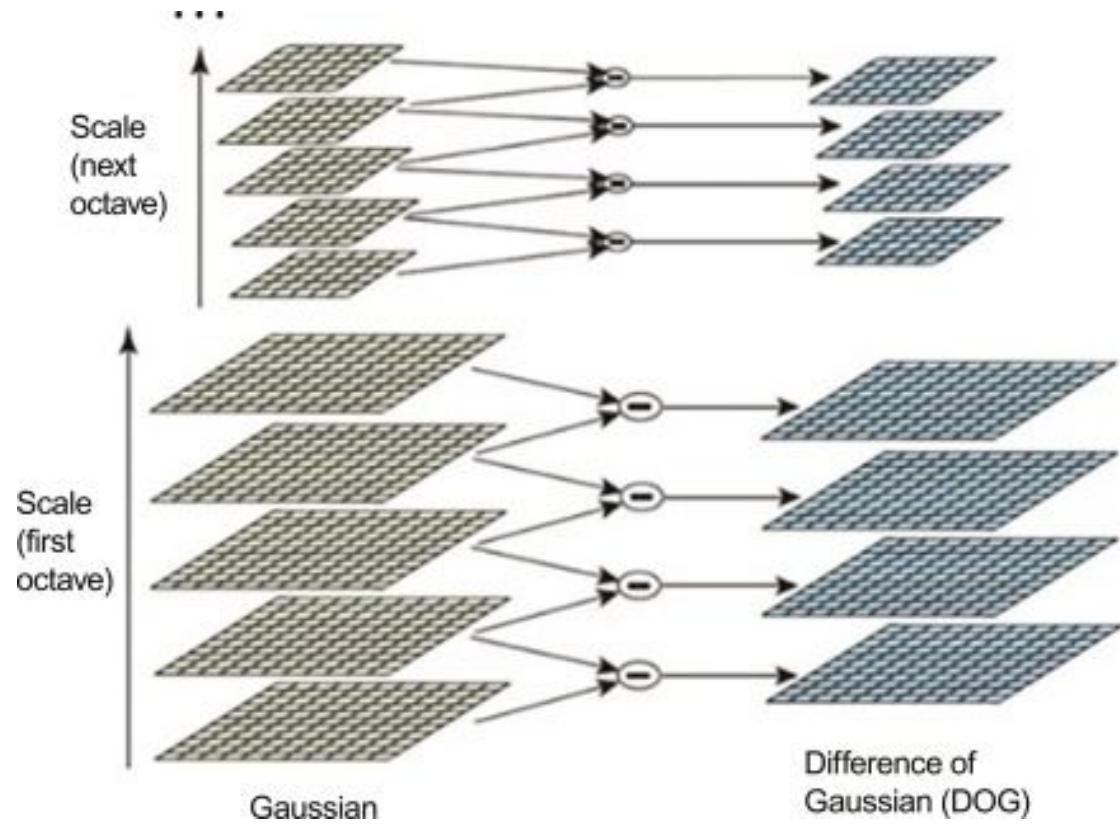


(b) keypoint descriptor

Biểu diễn SIFT (Scale-Invariant Feature Transform)

Scale-space extrema detection

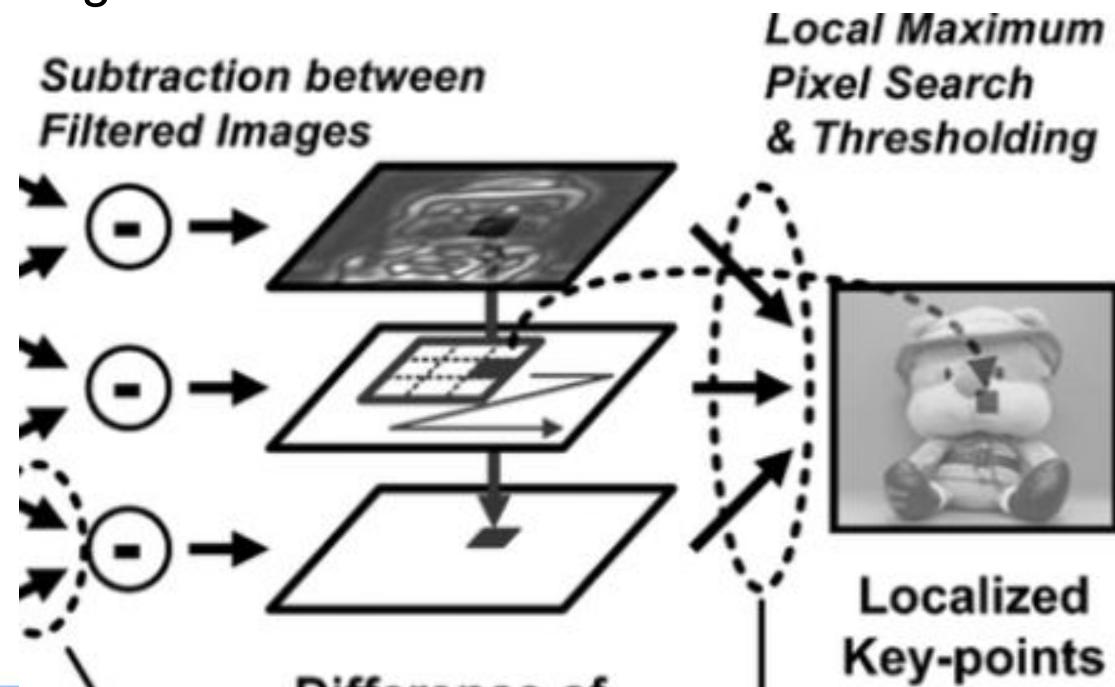
- **Scale-space filtering** được sử dụng để giải quyết vấn đề này. SIFT tính DoG (Difference of Gaussians) trên từng pixel bằng cách lấy diff của Gaussian Blur với 2 σ khác nhau



Biểu diễn SIFT (Scale-Invariant Feature Transform)

Keypoint localization

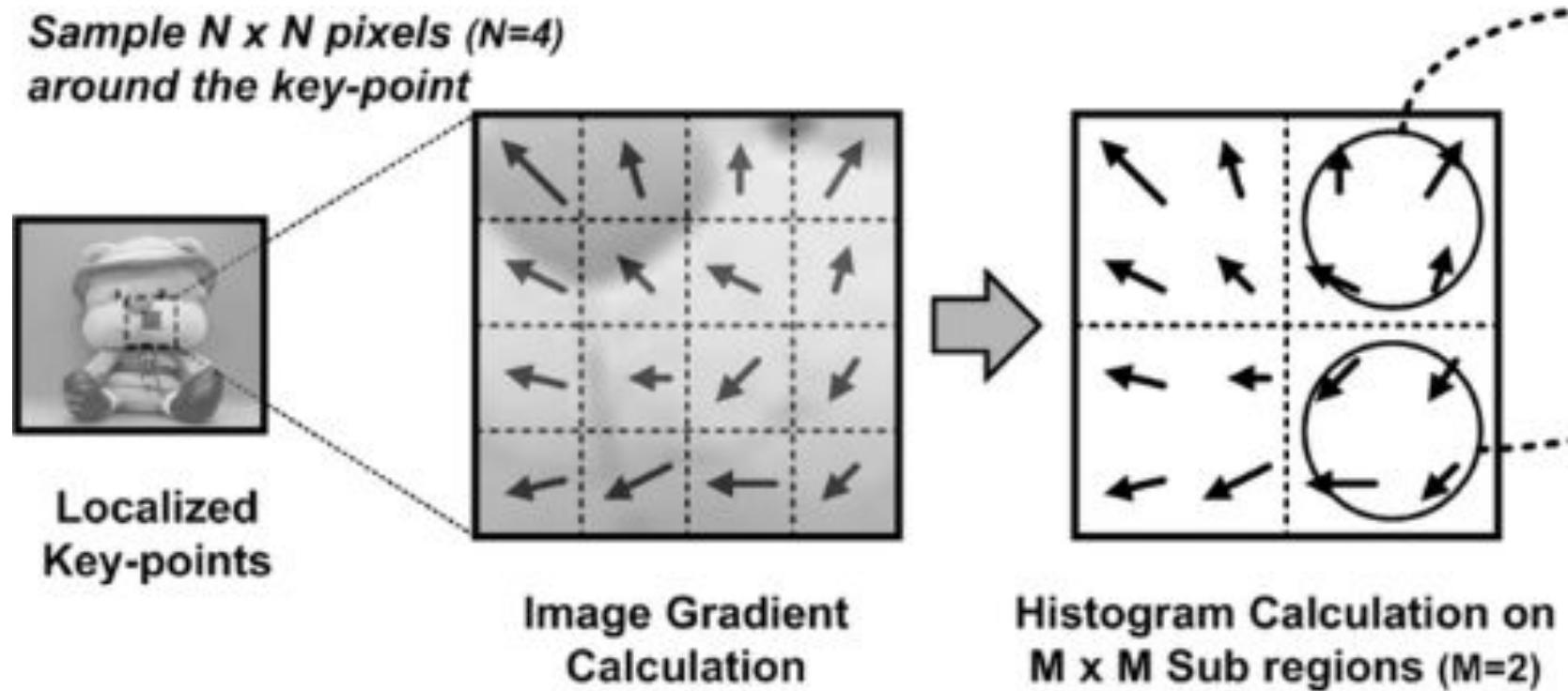
- Với mỗi scale, so sánh từng pixel bởi 8 láng giềng và 9 pixels tương ứng của scale trên và scale dưới. Từ đó có thể tìm ra các potential keypoints dựa trên local maxima và local minima
- Sau khi đã tìm được potential keypoints của ảnh, ta cần lọc, loại bỏ những keypoints có độ tương phản thấp hay những edge keypoints và giữ lại những điểm cần thiết hơn
- Sử dụng ngưỡng (threshold) để lọc là một phương pháp không tồi.



Biểu diễn SIFT (Scale-Invariant Feature Transform)

Orientation assignment

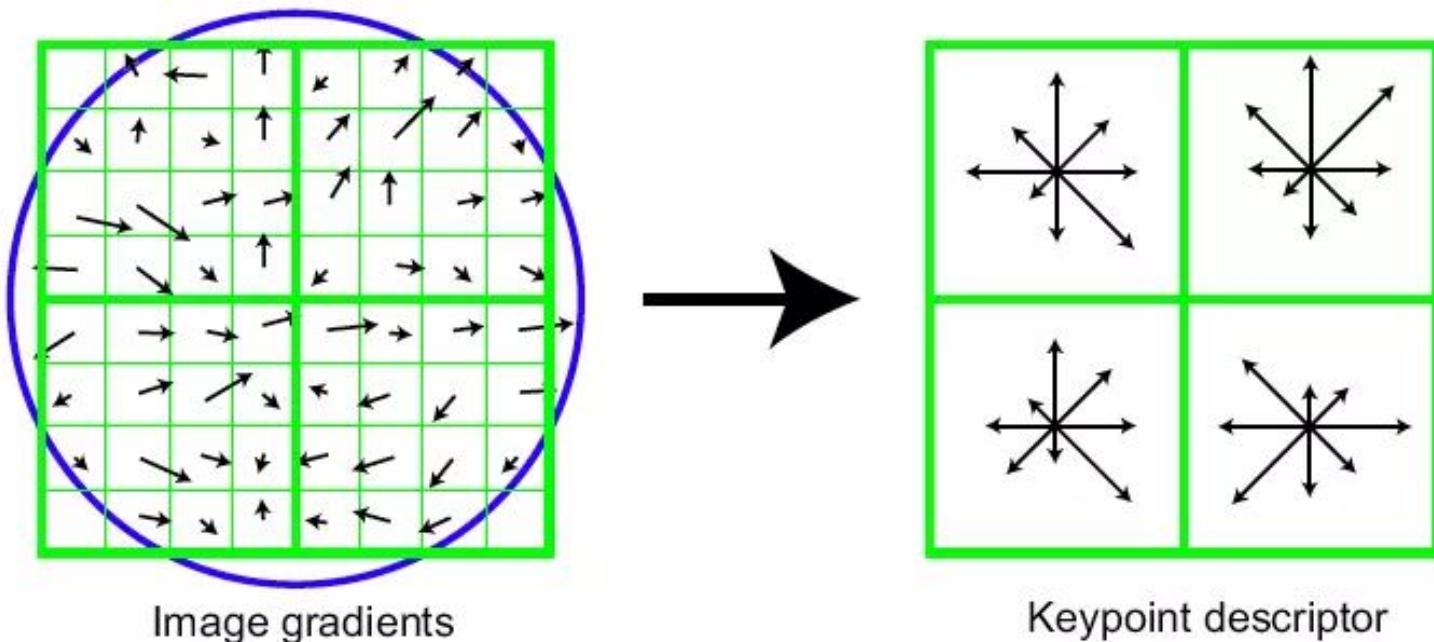
- Một hoặc nhiều orientation (hướng) sẽ được gắn cho từng vị trí keypoint dựa trên image gradient. Các phép tính sau đó trên ảnh sẽ tương ứng với từng orientation, scale và vị trí từng feature.



Biểu diễn SIFT (Scale-Invariant Feature Transform)

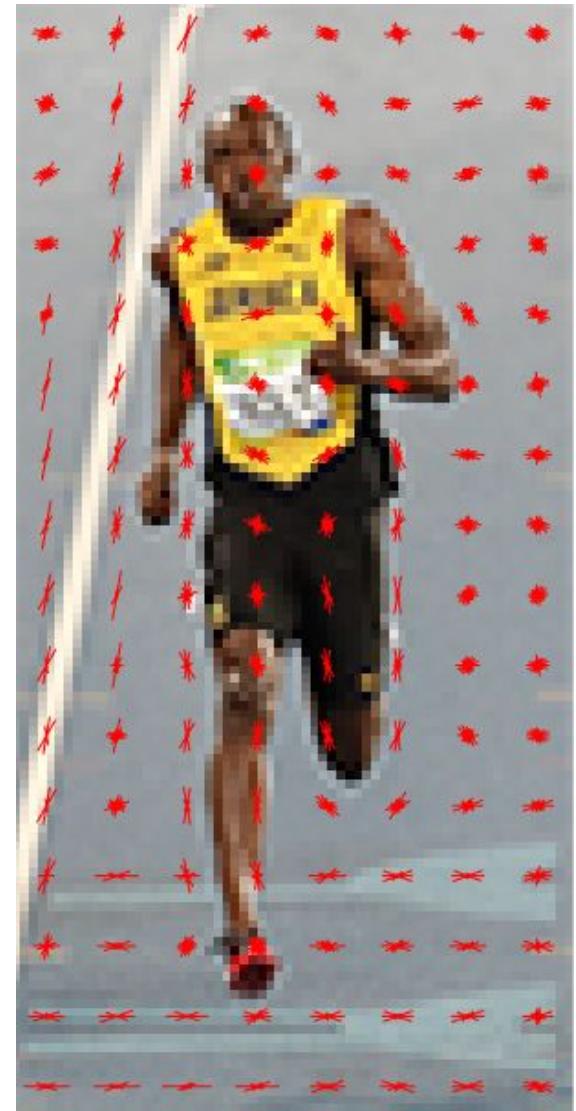
Keypoint descriptor

- SIFT sẽ tính Keypoint descriptor bằng cách lấy 16×16 neighborhood(điểm liền kề) của keypoint đó, rồi chia thành 16 sub-blocks với kích thước 4×4 . Với mỗi sub-block, ta sẽ tạo được 8 bin orientation (như hình dưới). Do đó tất cả sẽ có 128 bin giá trị tương ứng với 1 vector biểu hiện cho keypoint descriptor.



Biểu diễn HOG (Histogram of Oriented Gradients)

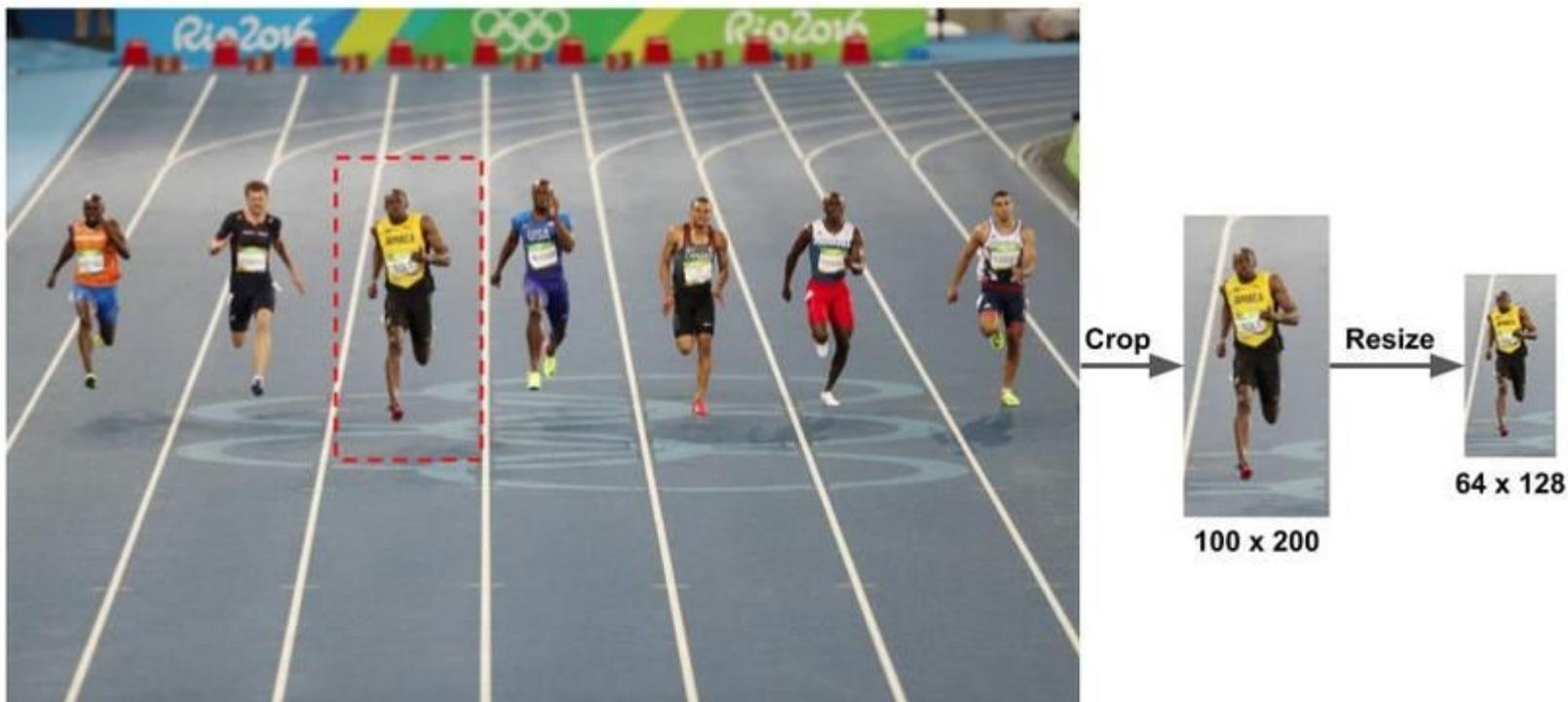
- HOG là viết tắt của Histogram of Oriented Gradient - một loại “feature descriptor”. Mục đích của “feature descriptor” là trừu tượng hóa đối tượng bằng cách trích xuất ra những đặc trưng của đối tượng đó và bỏ đi những thông tin không hữu ích. Vì vậy, HOG được sử dụng chủ yếu để mô tả hình dạng và sự xuất hiện của một đối tượng trong ảnh.
- Thông thường gồm 5 bước:
 - Tiền xử lý
 - Tính gradient
 - Tính vector đặc trưng cho từng cells
 - Chuẩn hóa
 - Tính vector đặc trưng HoG



Biểu diễn HOG (Histogram of Oriented Gradients)

Tiền xử lý

- Trong bài toán này, để thuận tiện cho việc chia đều hình ảnh thành các khối, ô và tính toán đặc trưng ở các bước tiếp theo, chúng ta cần resize kích thước tất cả các hình ảnh trong tập dữ liệu về một kích thước chung.



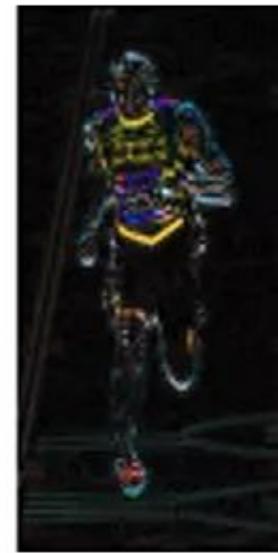
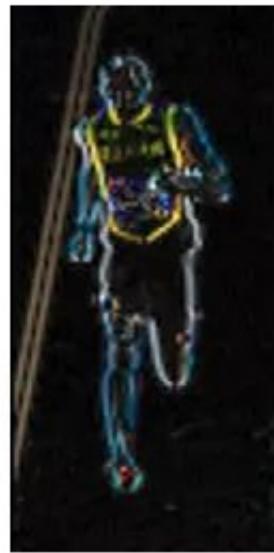
Biểu diễn HOG (Histogram of Oriented Gradients)

Tính Gradient

- Đây là bước đầu tiên, được thực hiện bằng hai phép nhân chập ảnh gốc với 2 chiều, tương ứng với các toán tử lấy đạo hàm theo hai hướng Ox và Oy. Trong đó, 2 hướng tương ứng đó là:

$$D_x = [-1 \ 0 \ 1] \text{ và } D_y = [1 \ 0 \ -1]^T,$$

- Sau bước này, kết quả thu được sẽ là:



Left : Absolute value of x-gradient. Center : Absolute value of y-gradient.

Right : Magnitude of gradient.

Biểu diễn HOG (Histogram of Oriented Gradients)

Tính vector đặc trưng cho từng ô (cells)

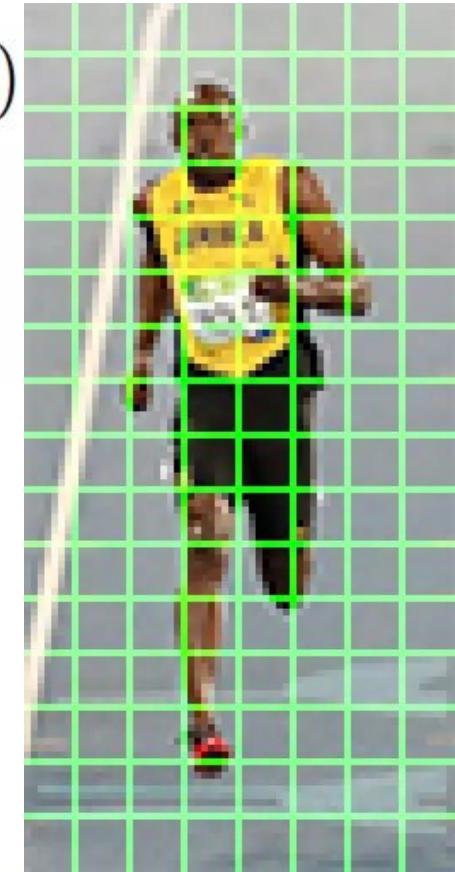
- Để tính toán vector đặc trưng cho từng ô (cell), chúng ta cần chia hình ảnh thành các block, mỗi block lại chia đều thành các cell. Để xác định được số block, chúng ta sẽ sử dụng công thức sau:

$$n_{block_image} = \left(\frac{W_{image} - W_{block} * W_{cell}}{W_{cell}} + 1 \right) * \left(\frac{H_{image} - H_{block} * H_{cell}}{H_{cell}} + 1 \right)$$

trong đó:

W_{image} , W_{block} , W_{cell} : lần lượt là chiều rộng của ảnh, khối, ô

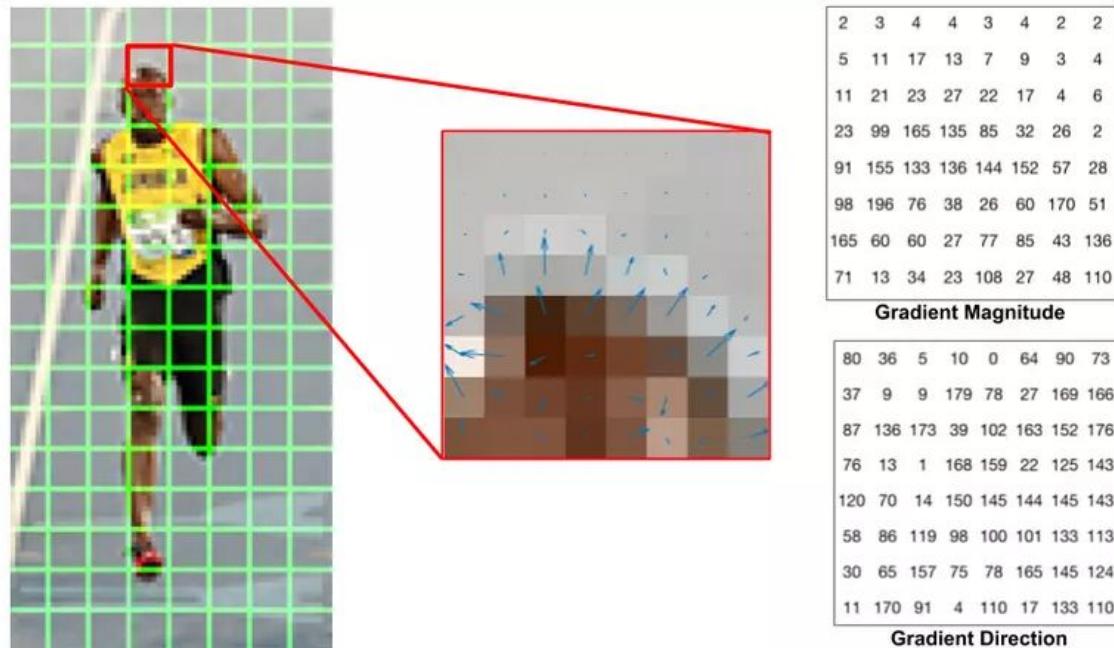
H_{image} , H_{block} , H_{cell} : lần lượt là chiều dài của ảnh, khối, ô



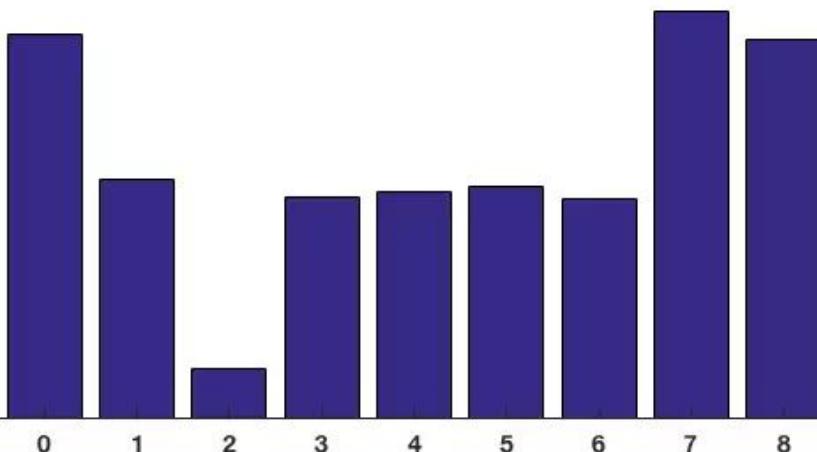
Biểu diễn HOG (Histogram of Oriented Gradients)

Tính vector đặc trưng cho từng ô (cells)

- Tiếp theo, tiến hành tính toán đặc trưng HOG tại mỗi cell sử dụng không gian hướng 9 bin, trường hợp “**unsigned-HOG**”. Hướng gradient sẽ chạy trong khoảng 0 độ đến 180 độ, trung bình 20 độ mỗi bin.
- Tại mỗi cell, xây dựng một biểu đồ cường độ gradient bằng cách vote các pixel vào biểu đồ. Trọng số vote của mỗi pixel phụ thuộc hướng và cường độ gradient (được tính toán từ bước 2) của pixel đó. Ví dụ:



Biểu diễn HOG (Histogram of Oriented Gradients)



80	36	5	10	0	64	90	73
37	9	9	179	78	27	169	166
87	136	173	39	102	163	152	176
76	13	1	168	159	22	125	143
120	70	14	150	145	144	145	143
58	86	119	98	100	101	133	113
30	65	157	75	78	165	145	124
11	170	91	4	110	17	133	116

Gradient Direction

2	3	4	4	3	4	2	2
5	11	17	13	7	9	3	4
11	2	23	27	22	17	4	6
23	99	165	135	85	32	26	2
91	155	133	136	144	152	57	28
98	196	76	38	26	60	170	51
165	60	60	27	77	85	43	136
71	13	34	23	108	27	48	110

Gradient Magnitude



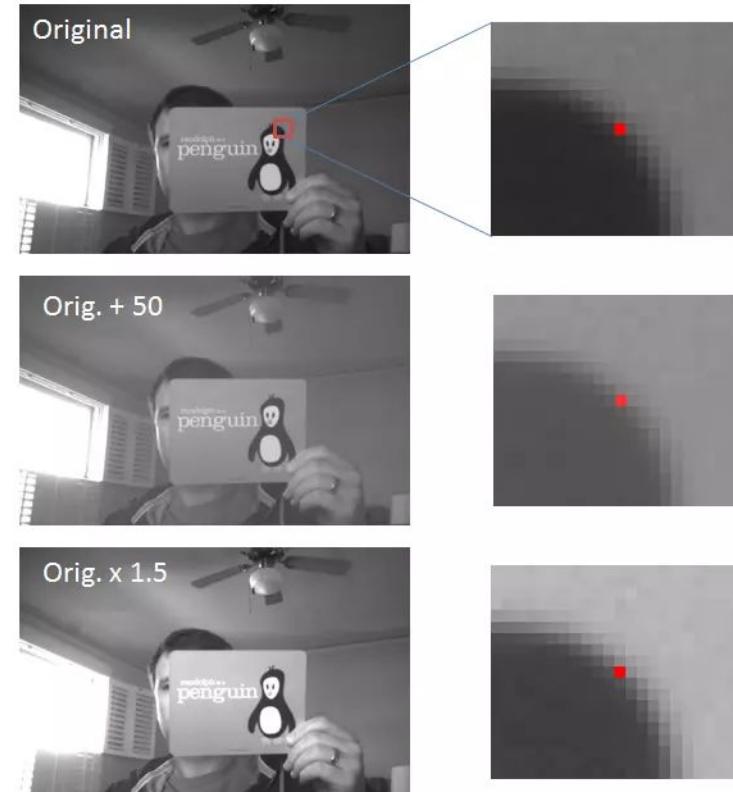
Histogram of Gradients

Biểu diễn HOG (Histogram of Oriented Gradients)

Chuẩn hóa khối (blocks)

- Để tăng cường hiệu năng nhận dạng, các histogram cục bộ sẽ được chuẩn hóa về độ tương phản bằng cách tính một ngưỡng cường độ trong một khối và sử dụng giá trị đó để chuẩn hóa tất cả các ô trong khối. Kết quả sau bước chuẩn hóa sẽ là một vector đặc trưng có tính bất biến cao hơn đối với các thay đổi về điều kiện ánh sáng.
- Đầu tiên, hãy xem xét ảnh hưởng của việc chuẩn hóa tới các vector gradient trong ví dụ sau:

Dễ dàng thấy được, trường hợp thứ ba hiển thị độ tương phản gia tăng.



Biểu diễn HOG (Histogram of Oriented Gradients)

Chuẩn hóa khối (blocks)

- Có nhiều phương pháp có thể được dùng để chuẩn hóa khối. Gọi v là vector cần chuẩn hóa chứa tất cả các histogram của mỗi khối. $\|v\|$ là giá trị chuẩn hóa của v theo các chuẩn $k=1, 3$ và e là một hằng số nhỏ. Khi đó, các giá trị chuẩn hóa có thể tính bằng một trong những công thức sau:
 - Ghép các vector đặc trưng khối sẽ thu được vector đặc trưng R-HOG cho ảnh. Số chiều vector đặc trưng ảnh tính theo công thức :

$$size_{image} = n * size_{block}$$

trong đó :

n là số khối của hình ảnh.

$size_{block}$ là số chiều của vector đặc trưng khối

$$\text{L2-norm: } f = \frac{v}{\sqrt{\|v\|^2 + e^2}}$$

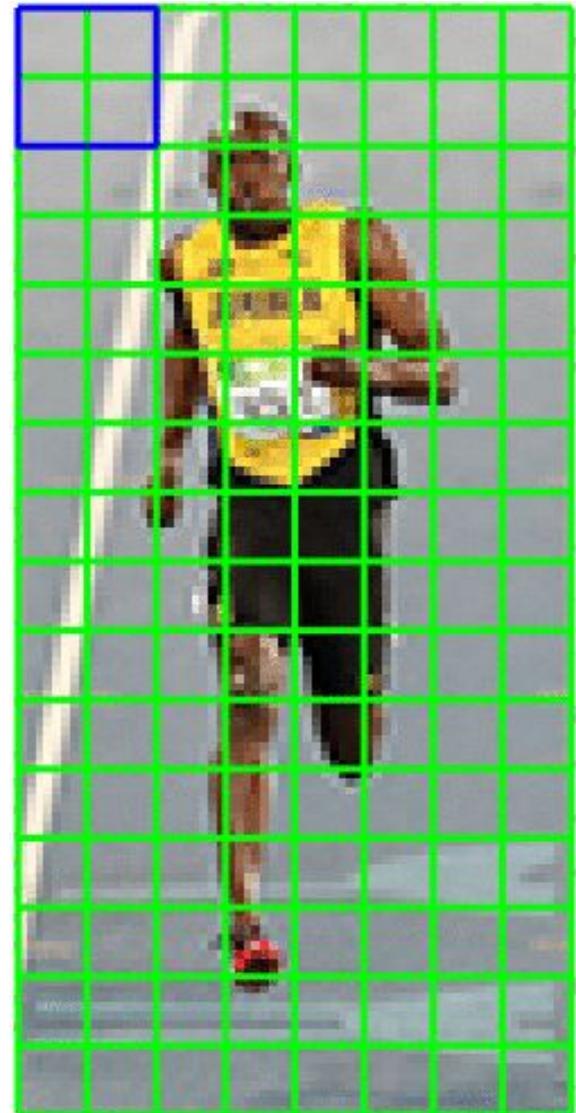
$$\text{L1-norm: } f = \frac{v}{\|v\| + e}$$

$$\text{L1-sqrt: } f = \sqrt{\frac{v}{\|v\|^2 + e}}$$

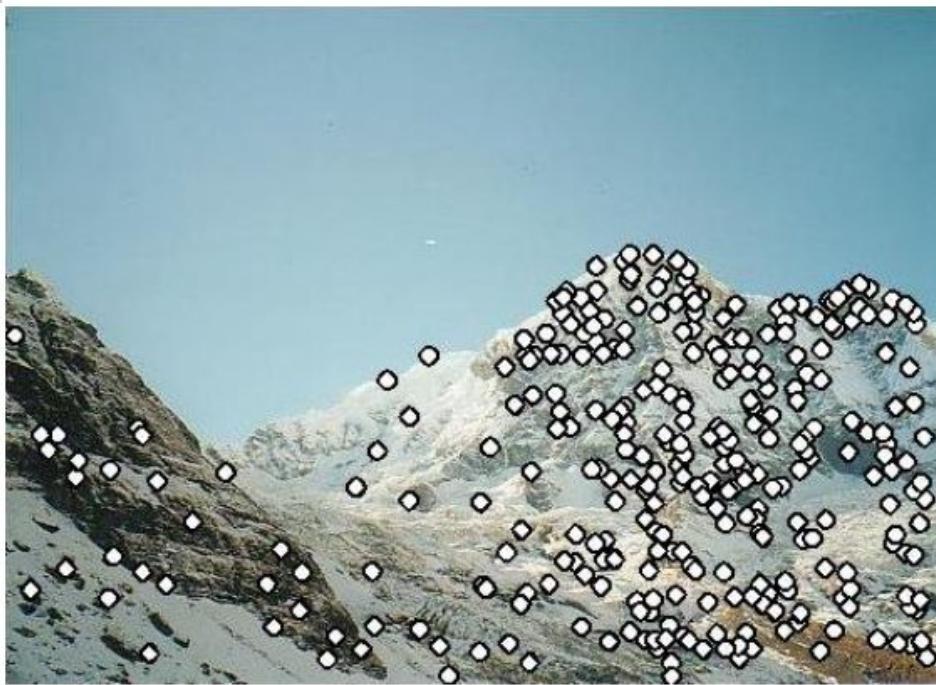
Biểu diễn HOG (Histogram of Oriented Gradients)

Tính toán vector đặc trưng HOG

- Với mỗi hình ảnh kích thước 64×128 , chia thành các block 16×16 chồng nhau, sẽ có 7 block ngang và 15 block dọc, nên sẽ có $7 \times 15 = 105$ blocks.
- Mỗi block gồm 4 cell. Khi áp dụng biểu đồ 9-bin cho mỗi cell, mỗi block sẽ được đại diện bởi một vector có kích thước 36×1 .
- Vì vậy, khi nối tất cả các vector trong một block lại với nhau, ta sẽ thu được vector đặc trưng HOG của ảnh có kích thước $105 \times 36 \times 1 = 3780 \times 1$.



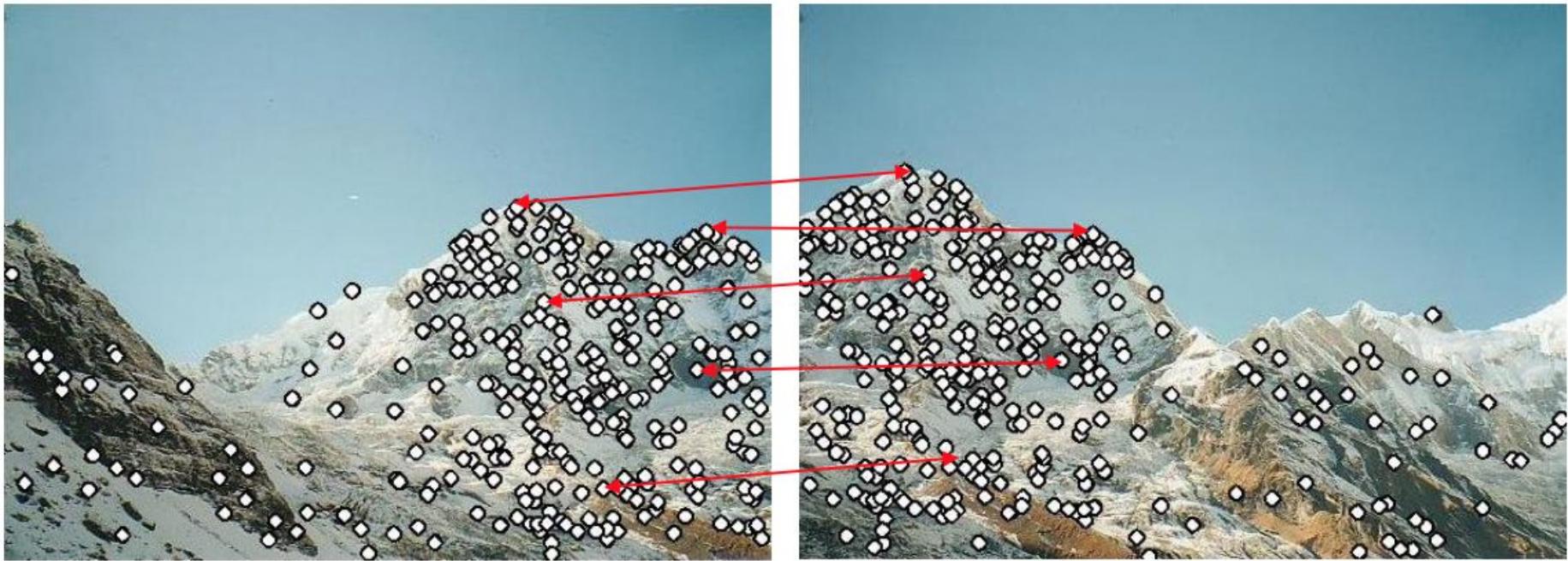
Ứng dụng: ghép ảnh



Quy trình:

- B1: tìm feature points ở cả 2 hình

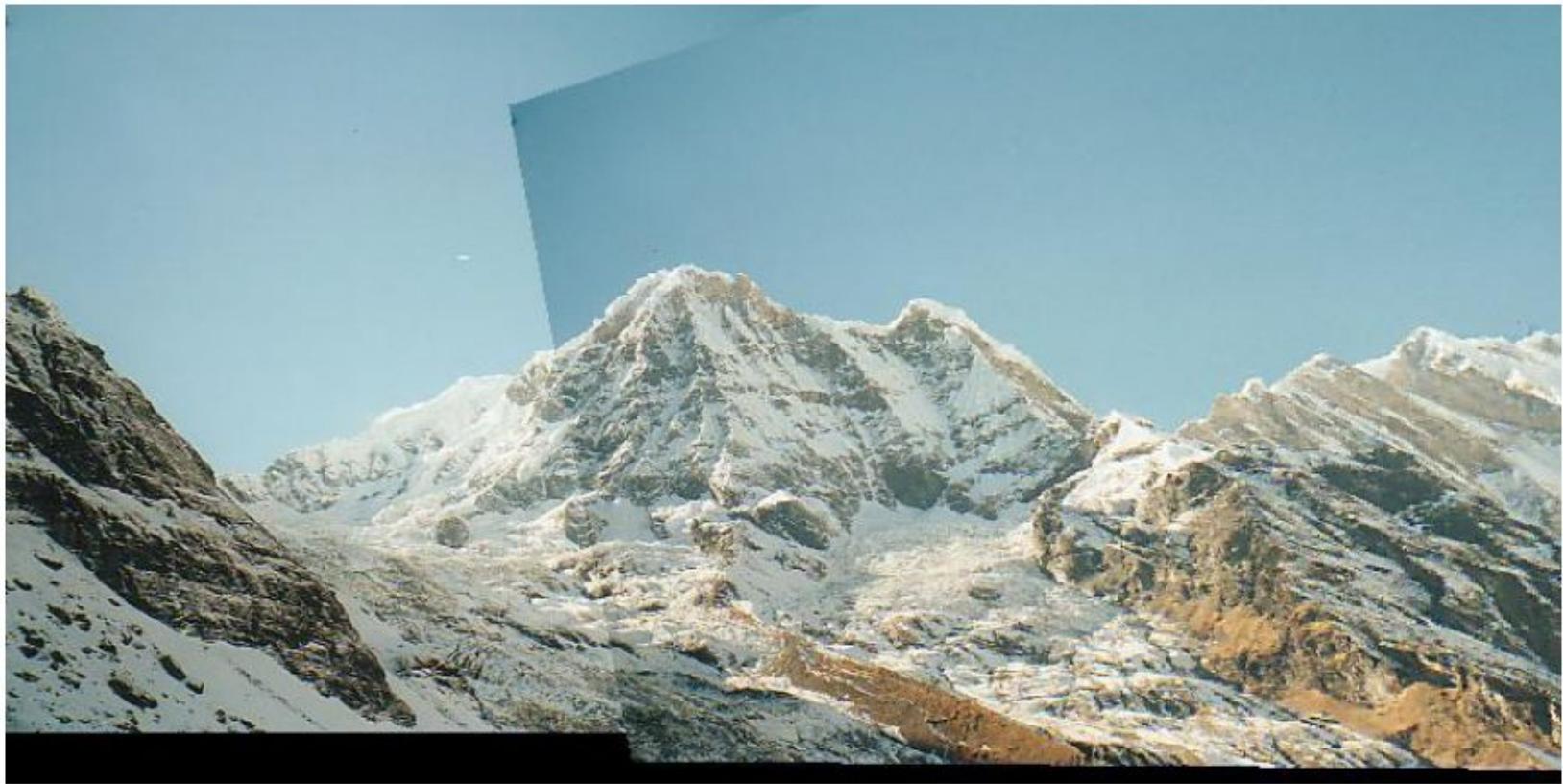
Sản phẩm: ghép ảnh



Quy trình:

- B1: tìm feature points ở cả 2 hình
- B2: Tìm các cặp giống nhau

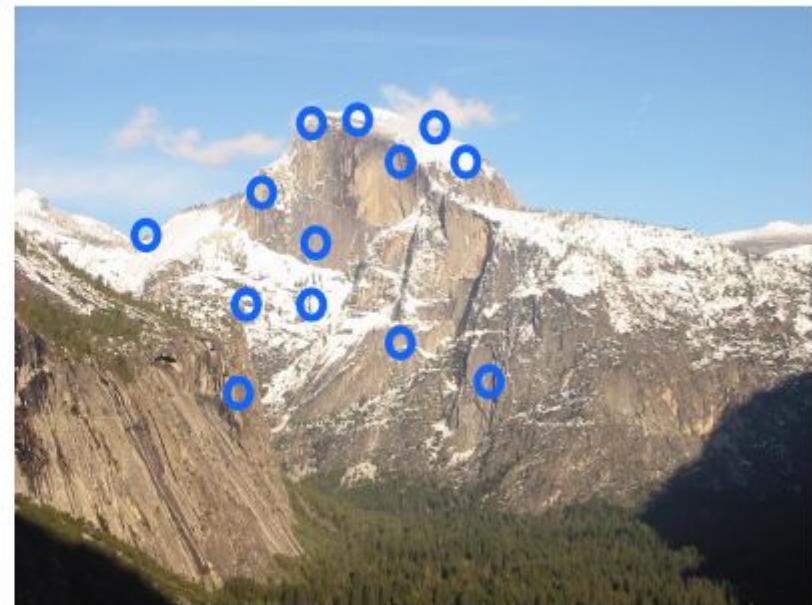
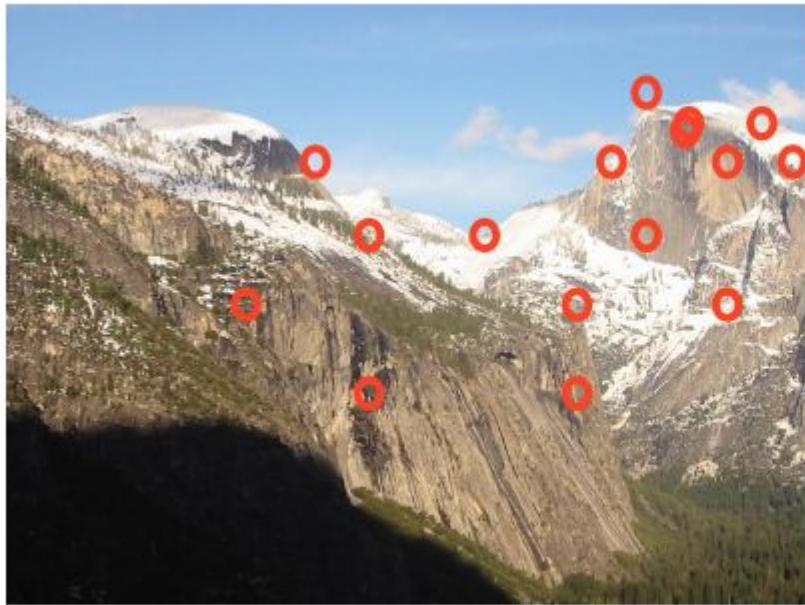
Sản phẩm: ghép ảnh



Quy trình:

- B1: tìm feature points ở cả 2 hình
- B2: Tìm các cặp giống nhau
- B3: Từ các cặp đó, đặt khớp 2 hình ảnh

Các bước xử lý chi tiết

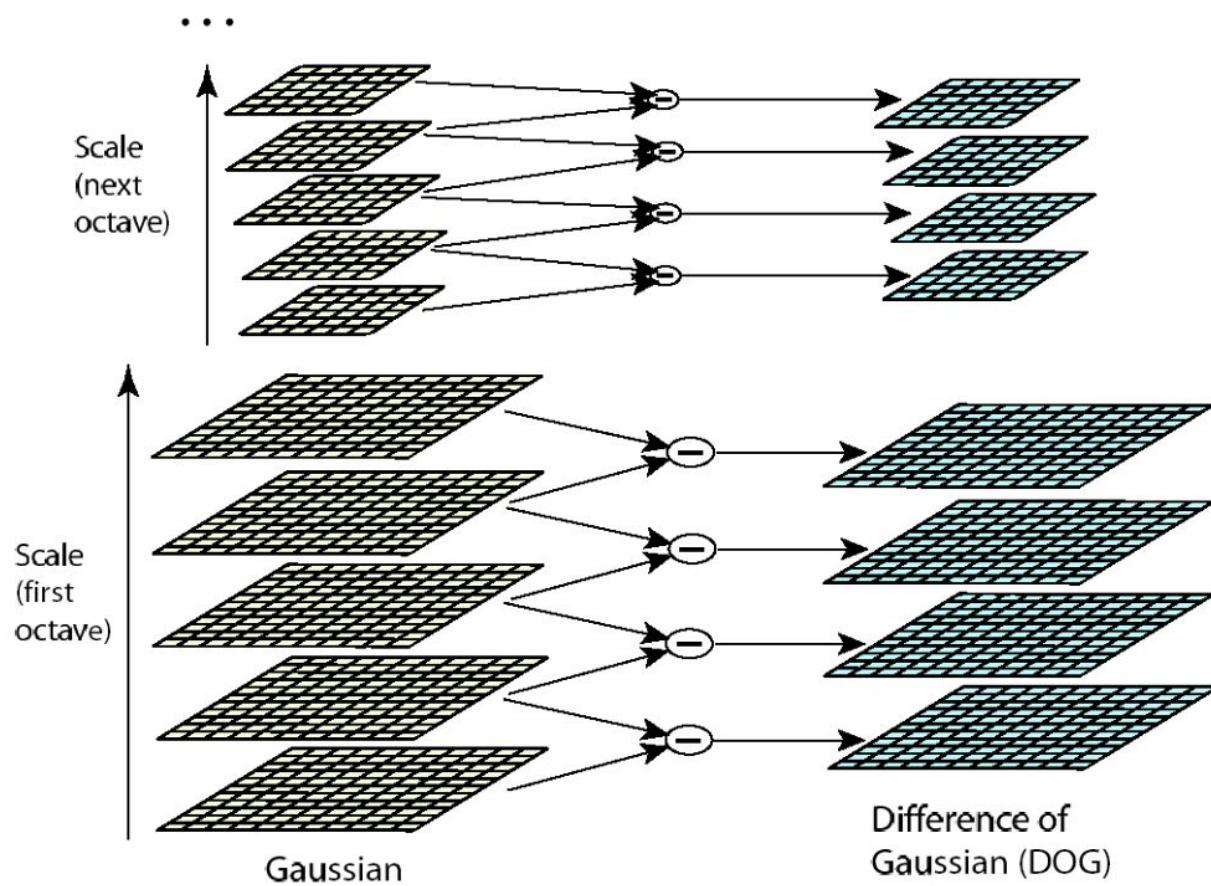


- Tìm các key points

Các bước xử lý chi tiết

- Tìm các key points

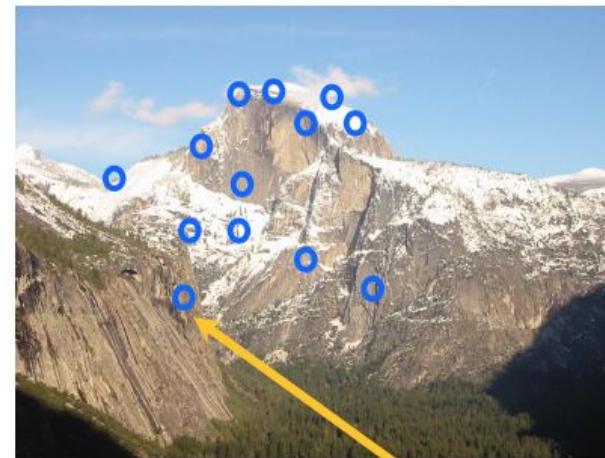
Detect Key Points



Các bước xử lý chi tiết



$(u_1, u_2, \dots, u_{128})$

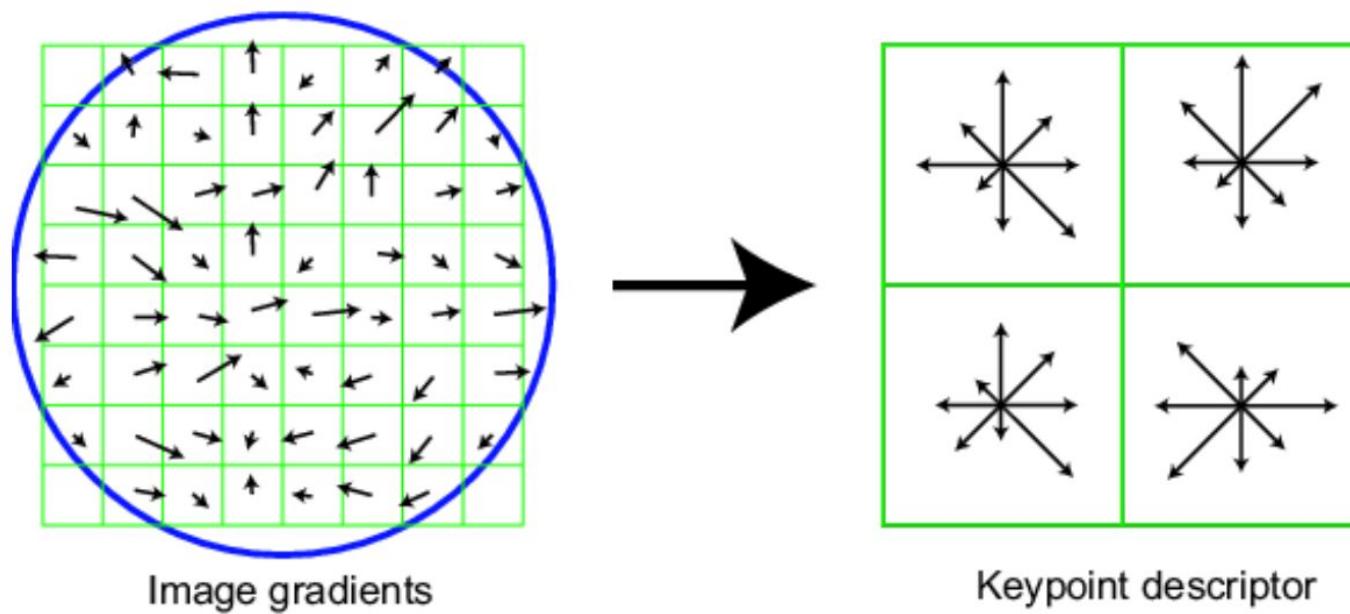


$(v_1, v_2, \dots, v_{128})$

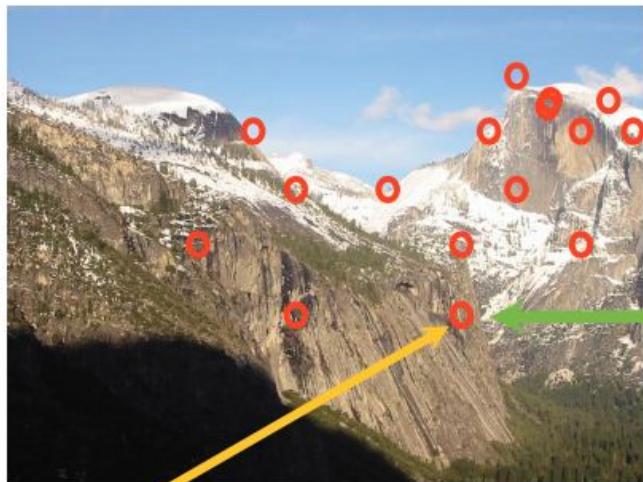
- Xây dựng biểu diễn SIFT

Các bước xử lý chi tiết

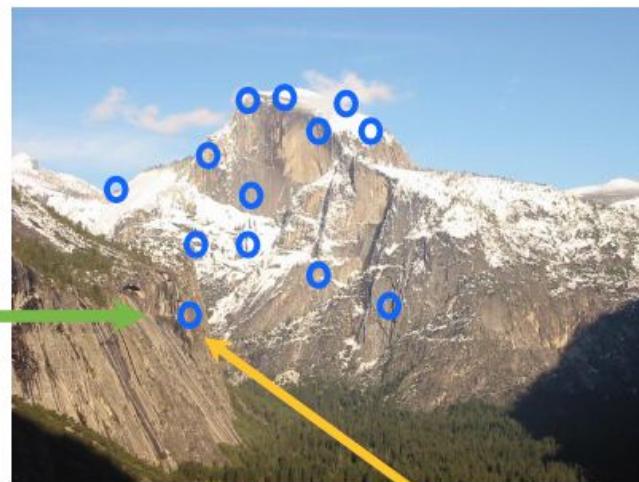
- Xây dựng biểu diễn SIFT



Các bước xử lý chi tiết



$(u_1, u_2, \dots, u_{128})$

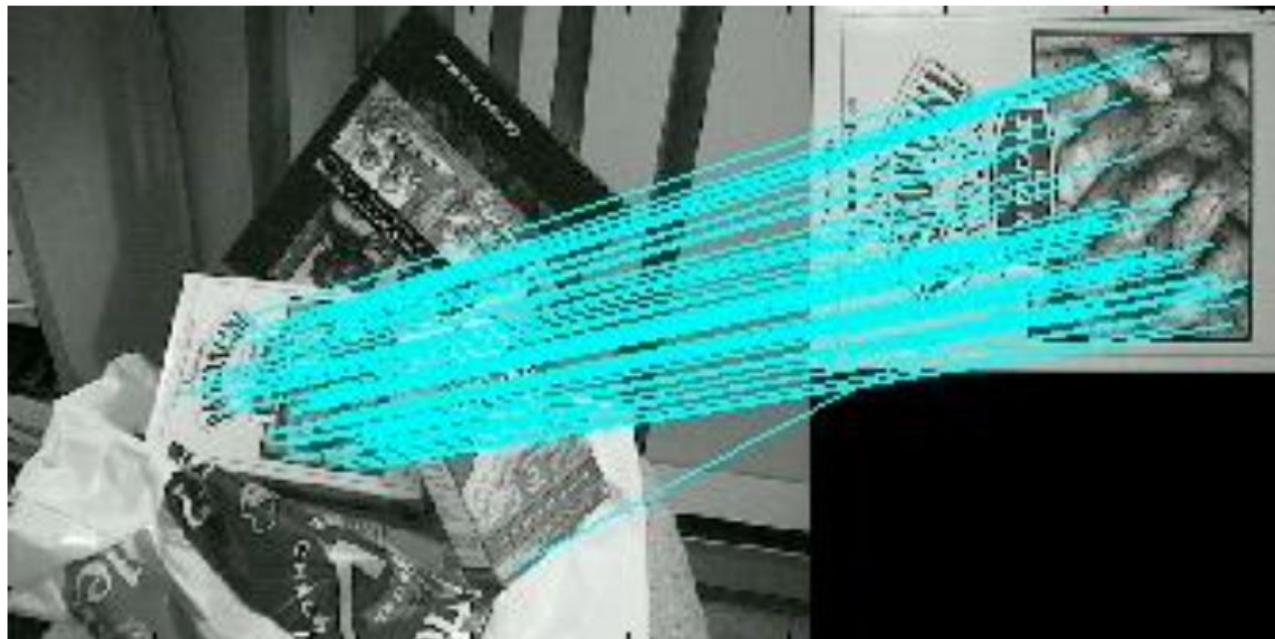


$(v_1, v_2, \dots, v_{128})$

- So sánh và tìm cặp điểm

Các bước xử lý chi tiết

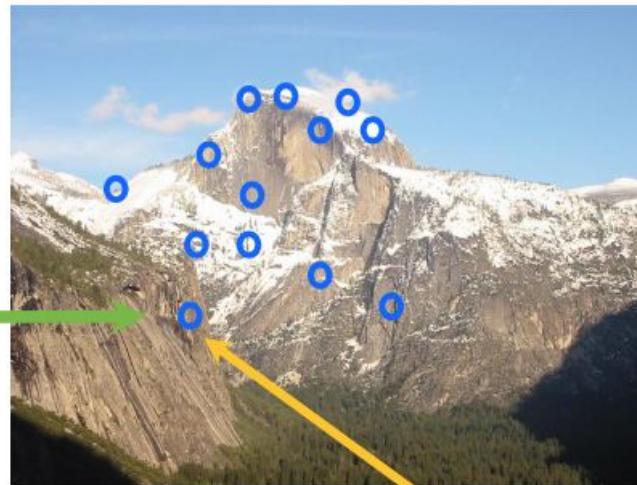
- So sánh và tìm cặp điểm
 - Có thể sử dụng khoảng cách Euclid vì các descriptors chỉ là vector



Các bước xử lý chi tiết



$(u_1, u_2, \dots, u_{128})$



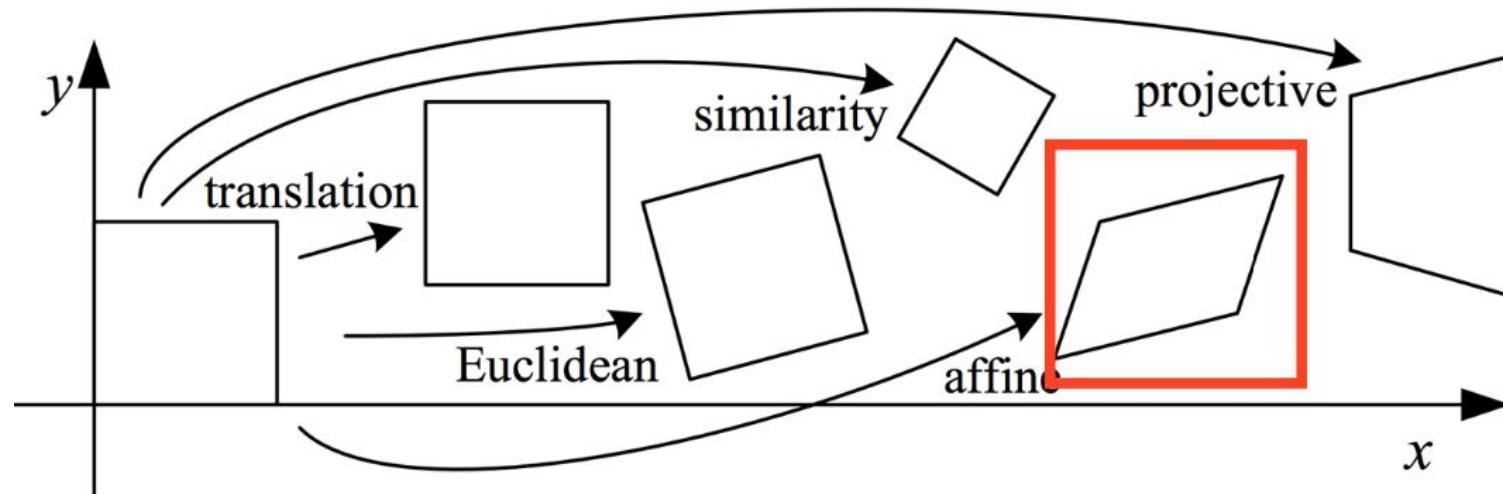
$(v_1, v_2, \dots, v_{128})$

- Tìm các tham số biến đổi phù hợp

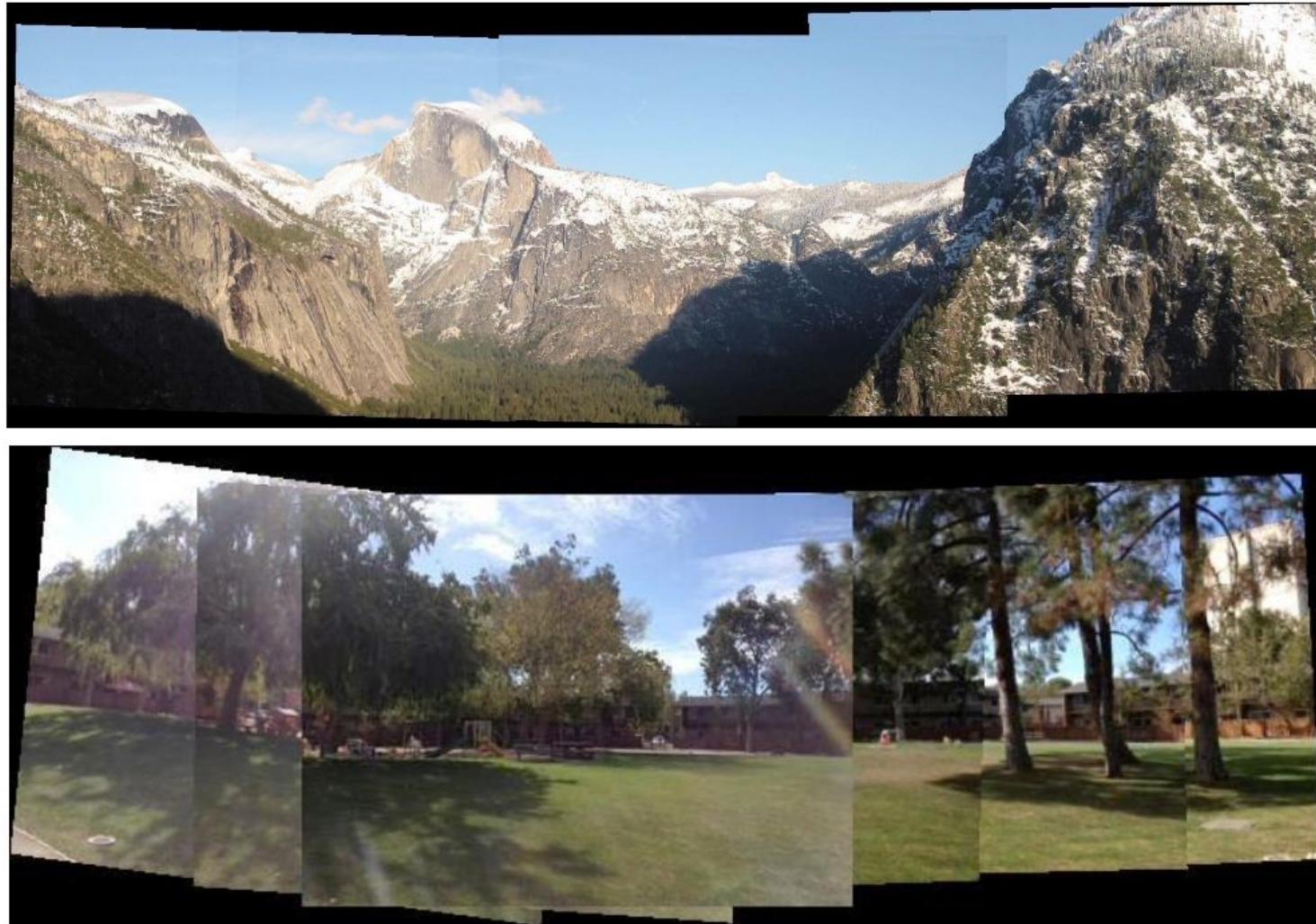
$$T = \begin{bmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ 0 & 0 & 1 \end{bmatrix}$$

Các bước xử lý chi tiết

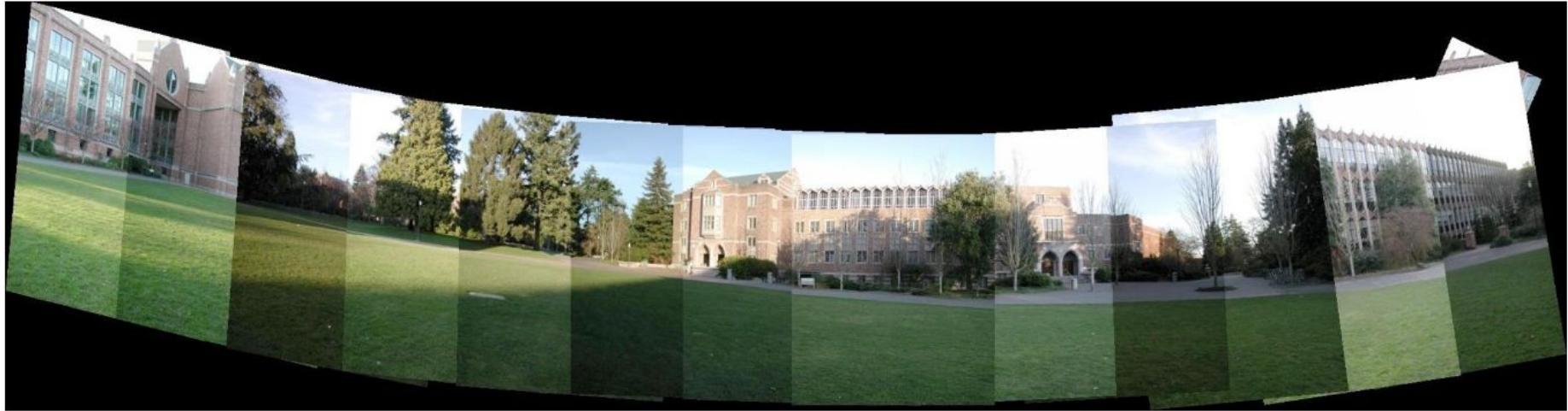
- Tìm các tham số biến đổi phù hợp
 - Biến đổi 2D: affine transformation



Kết quả



Kết quả



Tổng kết buổi học

- Các kĩ thuật trích xuất đặc trưng ảnh: biên, góc, ...
- Các ứng dụng của việc trích xuất đặc trưng ảnh

