



ĐẠI HỌC ĐÀ NẴNG

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG VIỆT - HÀN
VIETNAM - KOREA UNIVERSITY OF INFORMATION AND COMMUNICATION TECHNOLOGY
한-베정보통신기술대학교

Chương 4. Phát hiện đặc trưng ảnh và So khớp (Feature Detection and Matching)

TS. PHẠM NGUYỄN MINH NHỰT

Email: pnmnhut@vku.udn.vn

Phone: 0903.501.421

- 1. Khái niệm phát hiện và so khớp đặc trưng ảnh
- 2. Đặc trưng ảnh
- 3. Phát hiện đặc trưng ảnh và so khớp
- 4. Ảnh Panorama

1. Khái niệm phát hiện đặc trưng ảnh và so khớp

- Phát hiện đặc trưng (feature detection) và so khớp/đối sánh (Matching) ảnh là một nhiệm vụ quan trọng trong nhiều ứng dụng thị giác máy tính, chẳng hạn như:
 - Cấu trúc từ chuyển động → Mô hình hóa 3D thông qua chuyển động vật thể 2D



<https://images.google.com/>

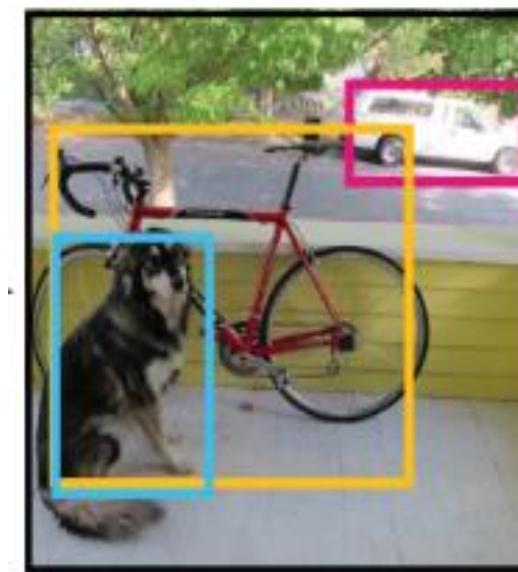


Search by image

Search Google with an image instead of text. Try dragging an image here.

Paste image URL Upload an image

Search by image



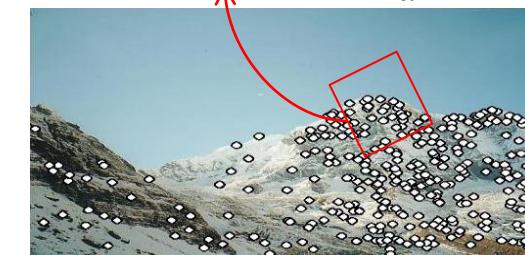
- Truy vấn hình ảnh
- Phát hiện đối tượng
- Điều hướng Robot
- Lưu vết đối tượng
- ...

1. Khái niệm phát hiện đặc trưng ảnh và so khớp

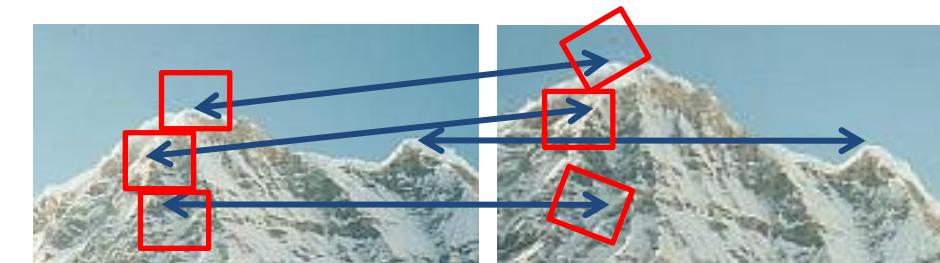
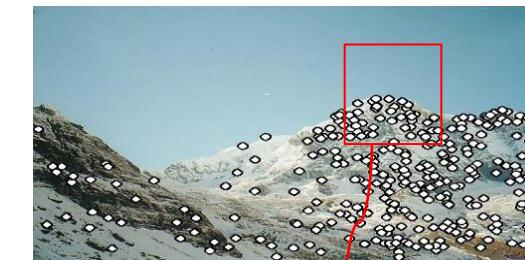
- Thành phần chính của phát hiện và so khớp đặc trưng

- Phát hiện điểm quan tâm (đặc trưng): Xác định các điểm cần quan tâm
- Mô tả: Trích xuất Bộ mô tả vectơ đặc trưng xung quanh mỗi điểm quan tâm.
- So khớp/đối sánh: Xác định sự tương quan giữa các Bộ mô tả trong hai khung nhìn (ảnh)

$$\mathbf{x}_1 = [x_1^{(1)}, \dots, x_d^{(1)}]$$



$$\mathbf{x}_2 = [x_1^{(2)}, \dots, x_d^{(2)}]$$



- Đặc trưng ảnh (Image Feature) là một phần thông tin của ảnh dùng để tính toán cho một ứng dụng nhất định.
- Các đặc trưng có thể được phân thành hai loại chính:
 - Loại 1: Đặc trưng cạnh/biên ảnh (image edge) là hướng cục bộ để nhận biết ranh giới giữa các đối tượng trong ảnh
 - Loại 2: Đặc trưng keypoint là những điểm quan tâm nằm ở các vị trí cụ thể của ảnh, chẳng hạn như: đỉnh núi, góc tòa nhà, ô cửa ... và được mô tả bằng các patch.
- Phát hiện (detect) các đặc trưng sử dụng các phép toán lân cận (lọc cải thiện, lọc khôi phục ảnh) hoặc áp dụng kỹ thuật phát hiện đối tượng (object detection) trên ảnh.

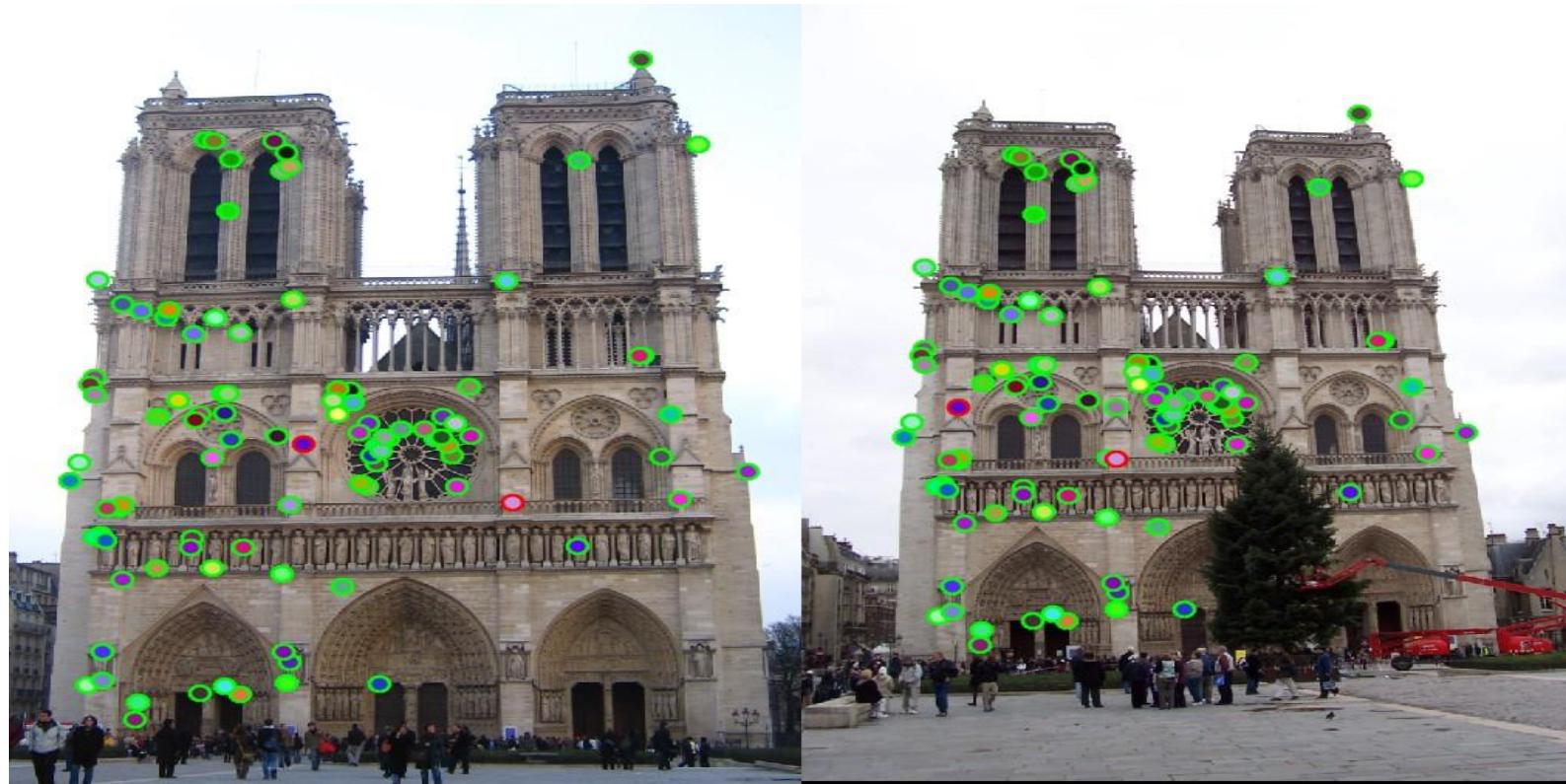


- Loại 1: Phát hiện đặc trưng cạnh/biên ảnh

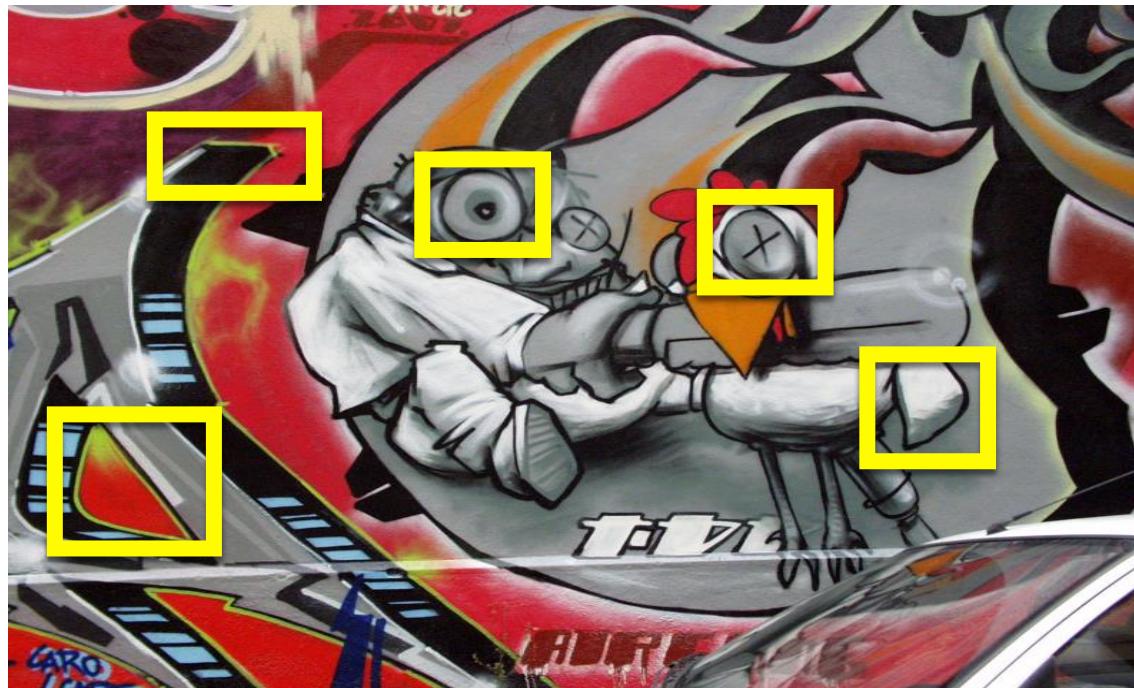
- Dựa trên sự thay đổi đột ngột về cường độ mức xám → phát hiện biên/cạnh ảnh
 - Sử dụng bộ lọc đạo hàm bậc 1, bậc 2
 - Toán tử Roberts
 - Toán tử Laplacian
 - Toán tử Sobel
 - ...
- Trình bày trong phần lọc sắc nét ảnh

- Loại 2: Phát hiện điểm đặc trưng (keypoint)

- Điểm quan tâm (điểm đặc trưng): là điểm được thể hiện trong kết cấu của ảnh, là điểm mà hướng cạnh (edge) của đối tượng thay đổi đột ngột hoặc là giao điểm giữa hai hoặc nhiều đoạn cạnh (edge)



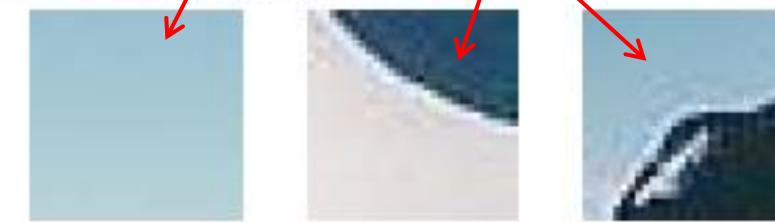
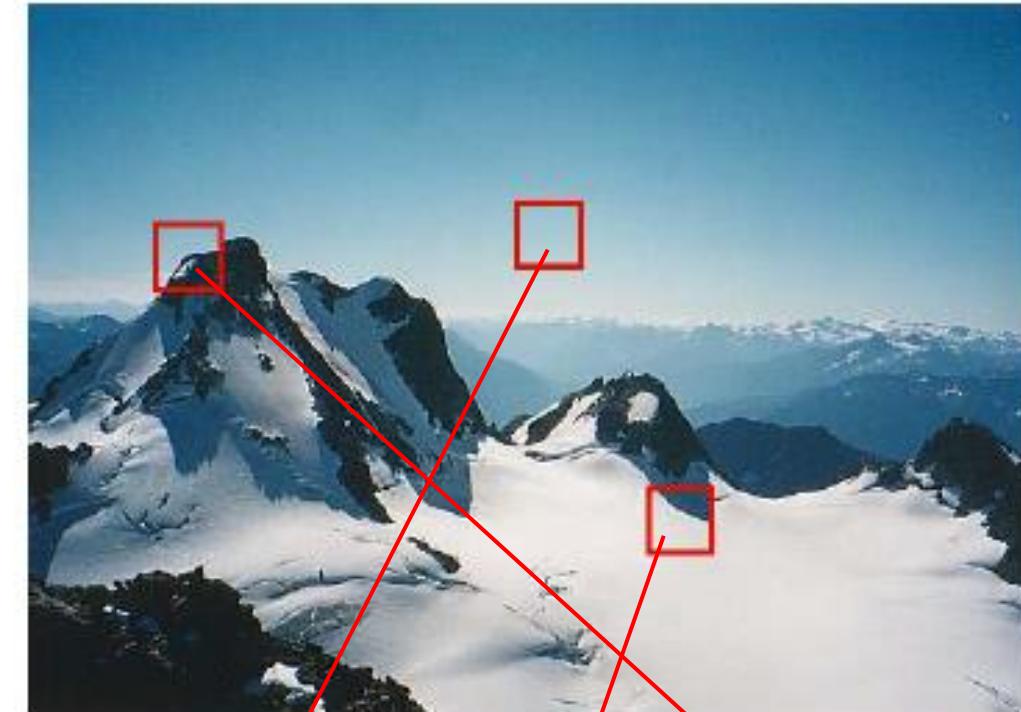
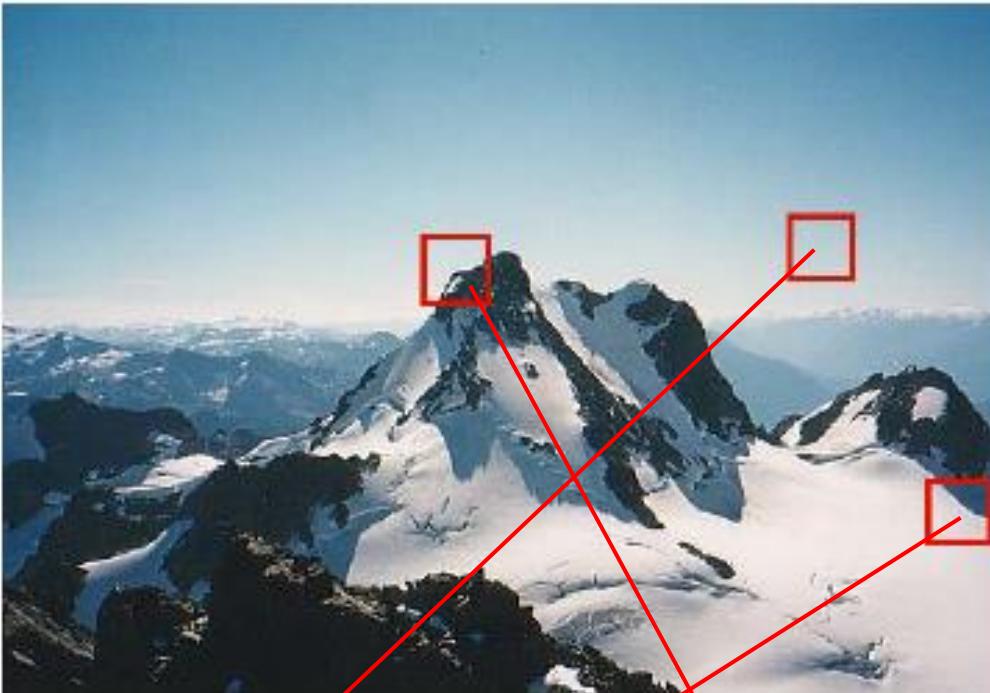
- Đặc điểm của điểm đặc trưng (điểm quan tâm):
 - Nó có một vị trí được xác định rõ ràng trong không gian ảnh hoặc được vị trí hóa tốt.
 - Nó ổn định dưới các nhiễu cục bộ và toàn cục
 - Có khả năng tính toán tin cậy → Nên được cung cấp khả năng phát hiện hiệu quả.



- Mục đích phát hiện điểm đặc trưng
 - Các điểm đặc trưng được sử dụng để tìm tập các vị trí tương ứng trong các ảnh khác nhau
 - → Căn chỉnh ảnh khi ghép các ảnh
 - → Đối sánh (so khớp) các ảnh
 - → Nhận dạng ảnh
- Phương pháp phát hiện điểm đặc trưng và sự tương quan của chúng: 2 phương pháp
 - PP1: Phát hiện điểm đặc trưng bằng cách sử dụng kỹ thuật tìm kiếm cục bộ, chẳng hạn như sự tương quan hoặc bình phương nhỏ nhất (MSE) giữa chúng
 - PP2: Phát hiện các điểm đặc trưng độc lập với ảnh đang xem xét và sau đó so khớp các đặc trưng dựa trên sự xuất hiện cục bộ của nó

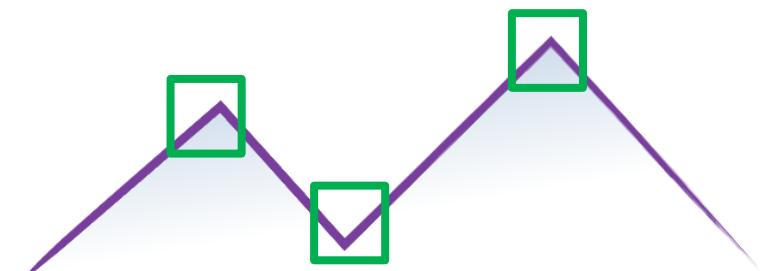
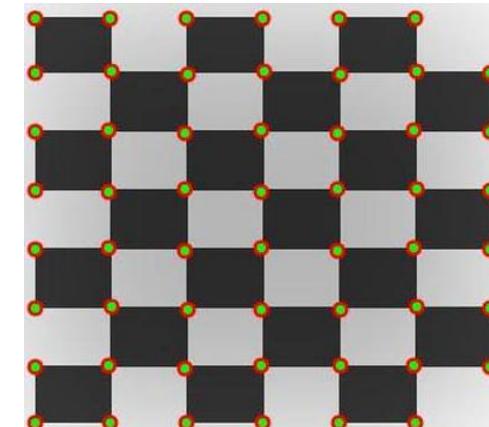
- **Patch:** là một cửa sổ hình chữ nhật chứa các điểm ảnh gần kề nhau. Cửa sổ này đủ lớn để có thể chứa được các thành phần để có thể mô tả được vật thể trong ảnh.
- **Ví dụ:**
 - Ảnh mặt người, các patch đủ lớn để chứa được các phần của khuôn mặt như mắt, mũi, miệng...
 - Ảnh là ô tô, các patch có thể chứa là bánh xe, khung xe, cửa xe
- **Hai patch gần giống nhau khi:**
 - Khoảng cách Euclide giữa hai vector tạo bởi hai patch đó càng nhỏ
 - Cùng nằm trong cụm (cluster) khi sử dụng phân cụm K-mean
 - ...

- Ví dụ: 02 ảnh với các patch



- Một số thuật toán phát hiện điểm đặc trưng, mô tả điểm đặc trưng
 - Thuật toán HCD (Harris Corner Detector): Phát hiện điểm đặc trưng
 - Thuật toán FAST (Features from Accelerated Segment Test): Phát hiện điểm đặc trưng
 - Thuật toán BRIEF(Binary Robust Independent Elementary Features): Mô tả điểm đặc trưng
 - Thuật toán SIFT(Scale Invariant Feature Transform): Phát hiện và mô tả điểm đặc trưng
 - Thuật toán SURF (Speeded-Up Robust Features): Phát hiện và mô tả điểm đặc trưng
 - Thuật toán ORB (Oriented FAST and Rotated BRIEF): Phát hiện và mô tả điểm đặc trưng

- Thuật toán HDC (Harris Corner Detector):
 - Là thuật toán **phát hiện góc (điểm đặc trưng)** của ảnh.
 - Được giới thiệu lần đầu tiên bởi Chris Harris và Mike Stephens vào năm 1988
- Góc (corner):
 - Góc là một điểm có vùng lân cận cục bộ theo hai hướng cạnh (biên ảnh) khác nhau
 - → một góc là điểm giao nhau của hai cạnh, trong đó một cạnh là sự thay đổi đột ngột về mức xám của ảnh.
 - Góc không thay đổi khi dịch chuyển, xoay, thay đổi mức xám → làm điểm đặc trưng của ảnh.



- Phát hiện góc:

- Xét một cửa sổ nhỏ (patch) xung quanh mỗi pixel $p(x,y)$ trong một ảnh.
- Phân biệt các patch khác nhau bằng cách di chuyển patch trong ảnh và đo giá trị thay đổi $E(u,v)$ của patch theo công thức (Gọi đây là giá trị Harris).

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

- Trong đó:
 - $w(x, y)$: cửa sổ (patch), thường là hàm Gaussian
 - u, v là tọa độ của mỗi pixel lân cận với điểm (x, y) trong cửa sổ (patch) $3 \times 3; 5 \times 5\dots$
 - $I(x, y)$ là giá trị mức xám của pixel $p(x, y)$
 - $I(x+u, y+v)$: giá trị mức xám thay đổi khi di chuyển đến điểm $(x+u, y+v)$

- Từ công thức

$$E(u, v) = \sum_{x,y \in w} w(x, y)[I(x + u, y + v) - I(x, y)]^2 \quad (1)$$

- Áp dụng công thức Taylor rút gọn biểu thức:

$$I(x + u, y + v) \approx I(x, y) + u \frac{\partial I}{\partial x} + v \frac{\partial I}{\partial y} = I(x, y) + uI_x + vI_y \quad (2)$$

- Thay (2) vào (1), triển khai và rút gọn:

$$E(u, v) \approx \sum_{x,y \in w} w(x, y)(u^2 I_x^2 + 2uI_x I_y + v^2 I_y^2) \quad (3)$$

- Biểu diễn (3) dưới dạng ma trận:

$$E(u, v) \approx [u \quad v] M \begin{bmatrix} u \\ v \end{bmatrix} \text{ với } M = \sum w(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \quad (4)$$

- Từ công thức (4):

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

Bộ lọc Sobel

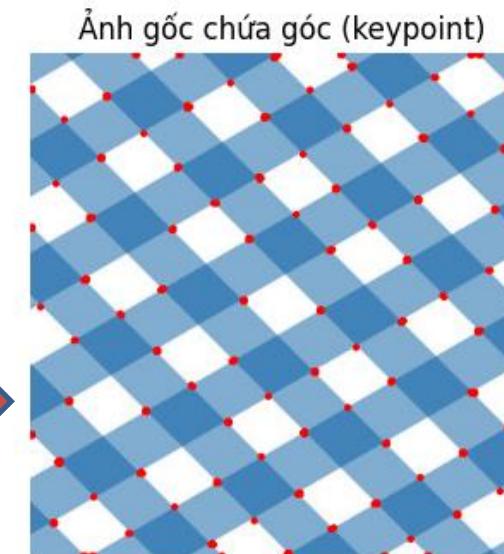
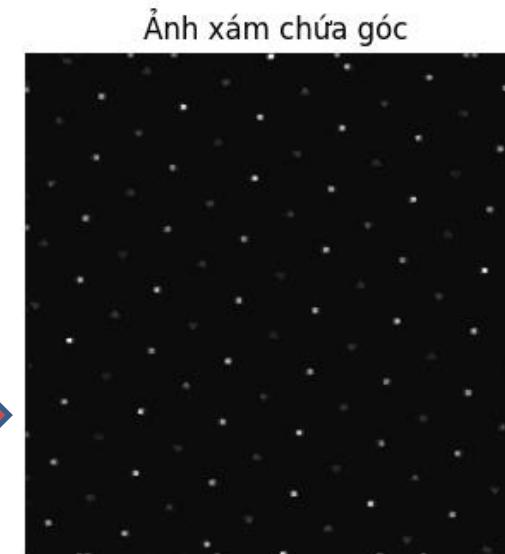
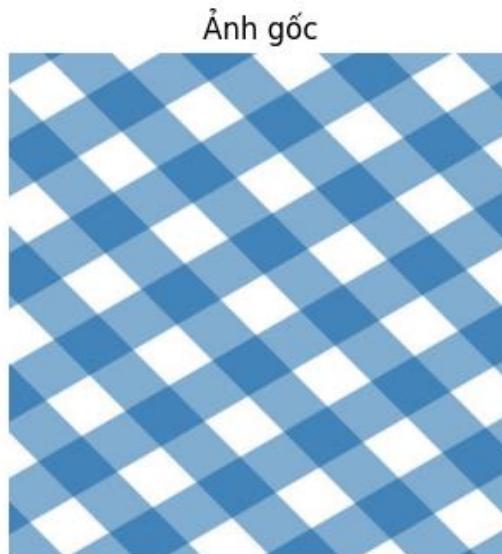
- thu được M có 2 giá trị riêng: λ_{\min} và λ_{\max}
- Vì mong muốn là $E(u, v)$ càng lớn → nghĩa là $\lambda_{\max} > \lambda_{\min}$ và λ_{\min} càng lớn
- Để đạt được điều đó thì sử dụng công thức Harris Response Calculation (HRC), như sau:

$$f = \frac{\lambda_{\max} \lambda_{\min}}{\lambda_{\max} - \lambda_{\min}} \text{ càng lớn} \quad (5)$$

- Nhận thấy: các điểm tại góc thỏa mãn công thức (5)

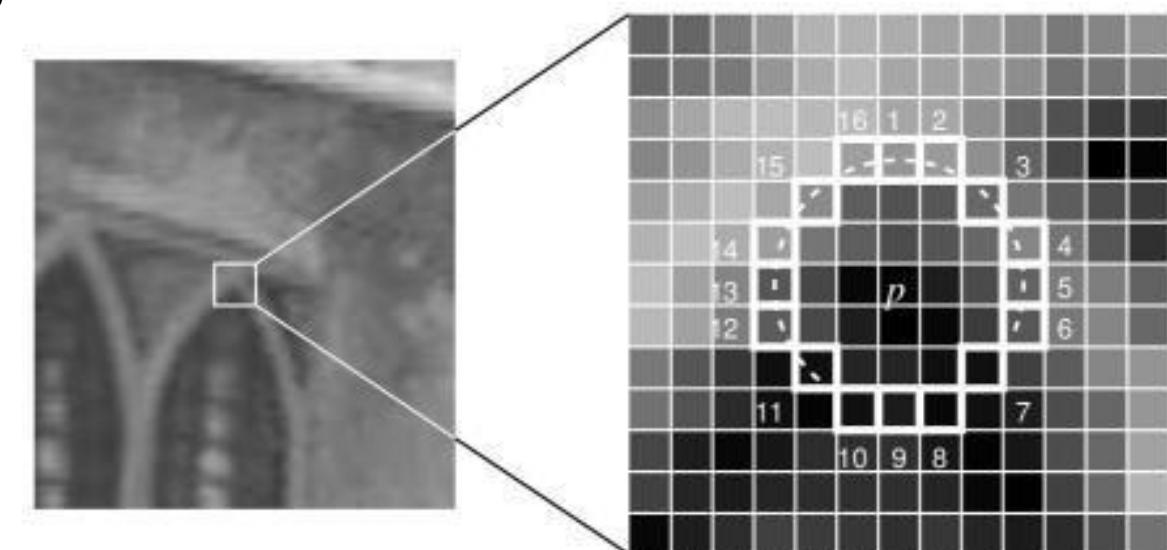
- Các bước thuật toán HDC:
 - **Bước 1:** Chuyển ảnh gốc sang ảnh đa cấp xám
 - **Bước 2:** Áp dụng bộ lọc Gaussian để làm mịn
 - **Bước 3:** Áp dụng toán tử Sobel để tìm các giá trị gradient x và y cho mọi pixel trong ảnh
 - **Bước 4:** Với mỗi pixel p trong ảnh xám, thiết lập một cửa sổ 3×3 xung quanh nó và tính hàm $E(u,v)$. Gọi đây là giá trị Harris của nó.
 - **Bước 5:** Tìm tất cả các pixel vượt quá một ngưỡng nhất định và là cực đại cục bộ trong một cửa sổ nhất định
 - **Bước 6:** Những pixel đáp ứng các tiêu chí trong Bước 5 → điểm đặc trưng.

Ví dụ phát hiện đặc trưng ảnh bằng Thuật toán HDC



- Thuật toán FAST (Features from Accelerated Segment Test):
 - Là thuật toán **phát hiện góc** (điểm đặc trưng)
 - Thuật toán FAST được phát triển bởi Edward Rosten và Tom Drummond và được xuất bản vào năm 2006.
- **Ưu điểm:**
 - Đạt được hiệu suất vượt trội về thời gian tính toán.
 - FAST rất thích hợp cho ứng dụng xử lý video thời gian thực vì hiệu suất tốc độ cao

- Các bước của thuật toán FAST:
 - **Bước 1:** Chọn một pixel p trong ảnh để xét nó là một điểm đặc trưng hay không → Gọi giá trị mức xám của nó là L_p .
 - **Bước 2:** Chọn giá trị ngưỡng thích hợp T .
 - **Bước 3:** Xét một vòng tròn 16 pixel xung quanh p (bán kính = 3) → pixel p là một góc (điểm đặc trưng) nếu tồn tại một tập n pixel liền kề (trong vòng tròn 16 pixel). Tất cả đều sáng hơn $L_p + T$ hoặc tối hơn $L_p - T$
 - **Bước 4:** Lặp lại tất cả các pixel trong ảnh nếu thỏa mãn điều kiện Bước 3 → là điểm đặc trưng



3. Phát hiện đặc trưng ảnh: Thuật toán FAST – Ví dụ

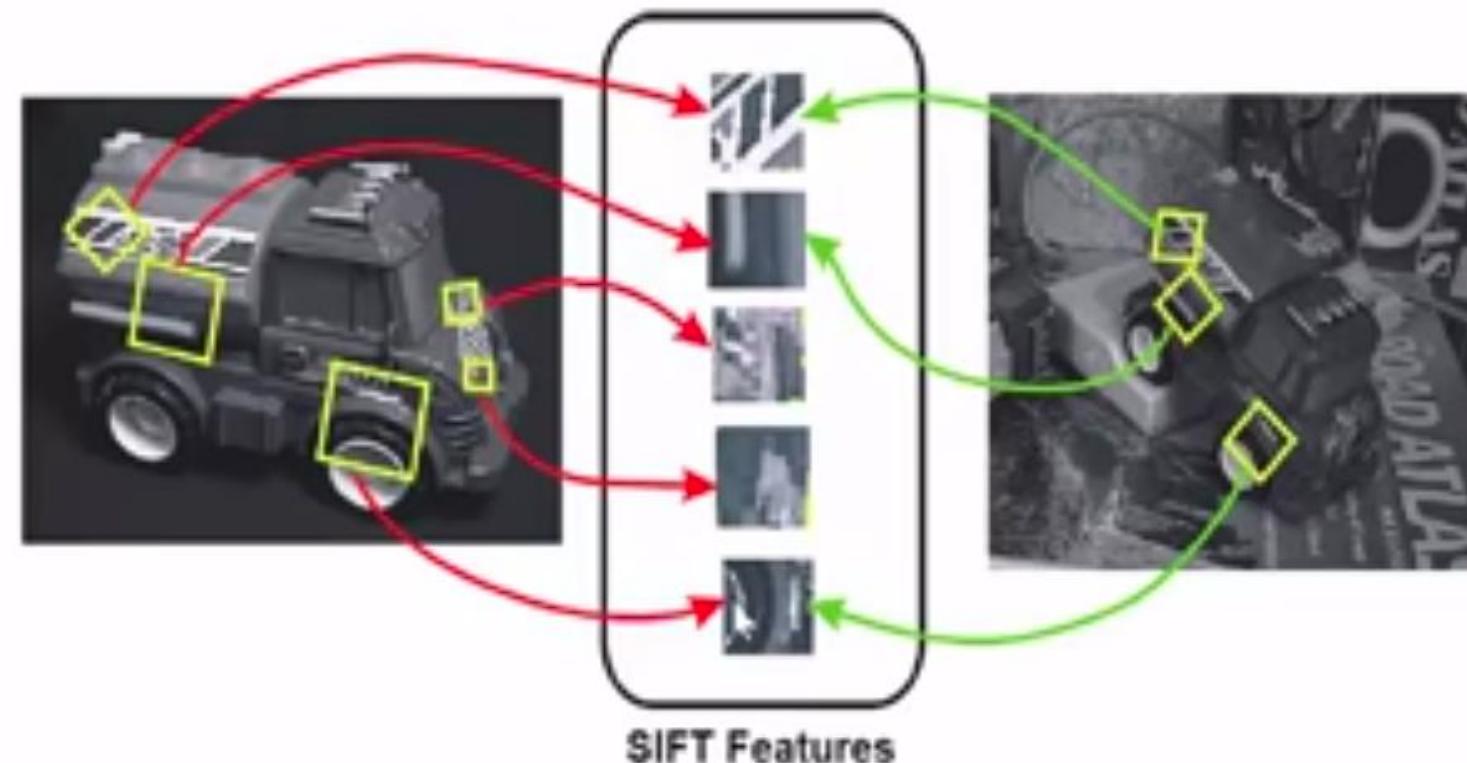
- Phát hiện điểm đặc trưng từ thuật toán FAST và vẽ trên ảnh



- Thuật toán BRIEF (Binary Robust Independent Elementary Features):
 - Sau khi sử dụng Thuật toán HDC, FAST để phát hiện điểm đặc trưng (**góc - corner**), cần mô tả các điểm đặc trưng → so khớp.
 - Thuật toán BRIEF không phát hiện điểm đặc trưng mà **chỉ tạo ra mô tả điểm đặc trưng** từ các thuật toán phát hiện đặc trưng
- Các bước thuật toán BRIEF:
 - **Bước 1:** Tạo các patch với tâm là điểm góc (điểm đặc trưng)
 - **Bước 2:** Mã hóa các patch bằng cách chuyển các patch thành một vectơ nhị phân đặc trưng gồm một chuỗi các bit 1 và 0, có độ dài 128 – 512 bit. Cụ thể:
 - Bước 2.1: Sử dụng bộ lọc Gaussian để làm mịn các patch
 - Bước 2.2: Chuyển các patch làm mịn thành các vec tơ nhị phân đặc trưng
- **Chú ý:** So khớp điểm đặc trưng: Dựa trên tỷ lệ láng giềng gần nhất (Nearest Neighbor Distance Ratio) của các điểm đặc trưng (**xem trong nội dung Thuật toán SIFT phần sau**)

4. Phát hiện và mô tả đặc trưng ảnh: Thuật toán SIFT

- Thuật toán SIFT(Scale Invariant Feature Transform):
 - Được công bố lần đầu tiên vào năm 2004, bởi D.Lowe, Đại học British Columbia.
 - Thuật toán SIFT **phát hiện và mô tả điểm đặc trưng** không ảnh hưởng đến tỷ lệ và góc quay của hình ảnh.

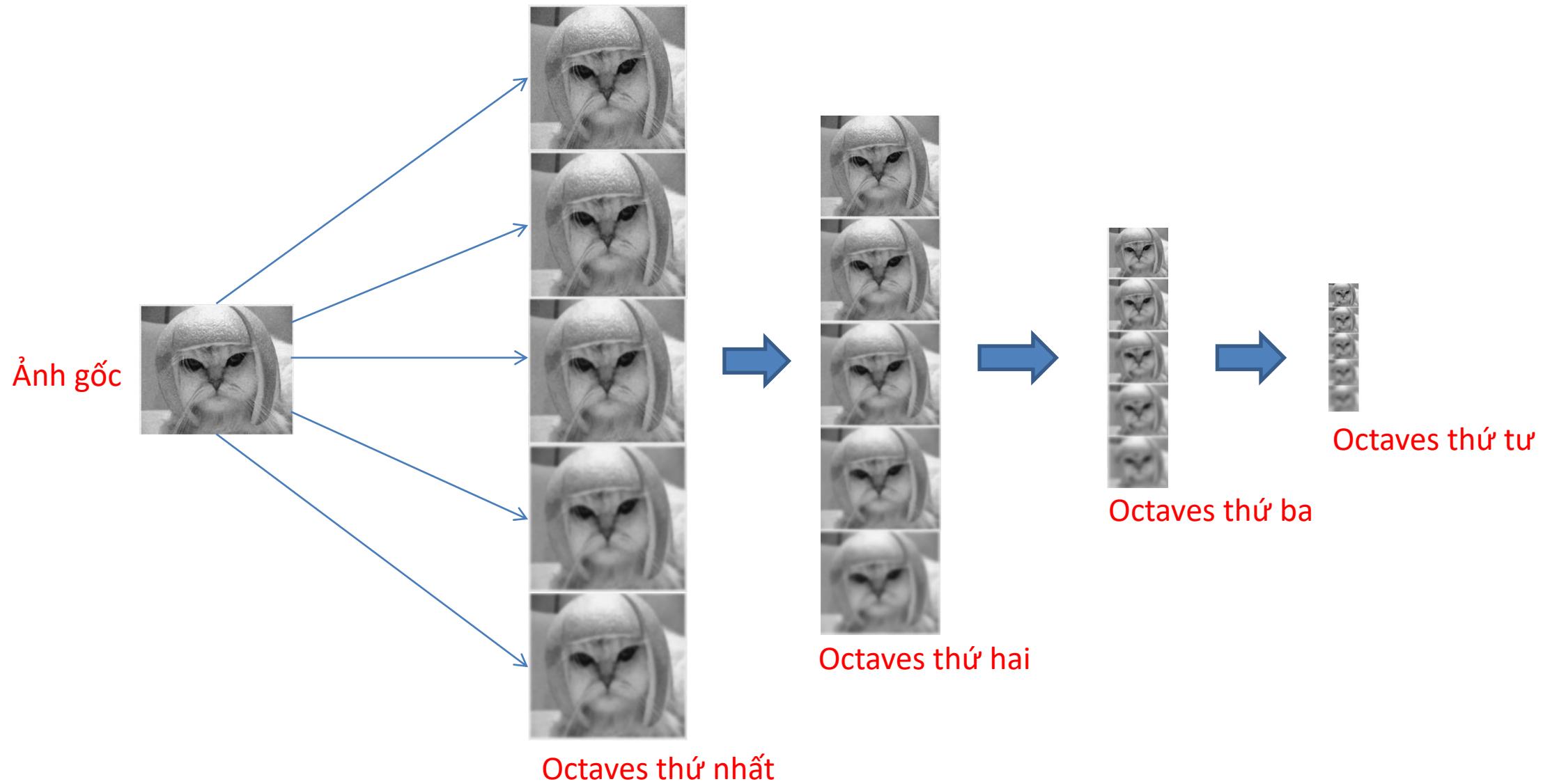


- Các bước của thuật toán SIFT(Scale Invariant Feature Transform)
 - **Bước 1: Tìm kiếm các điểm keypoint tiềm năng:**
 - 1.1 Xác định các điểm cực trị trên **không gian tỷ lệ** của ảnh
 - 1.2 Tìm kiếm các điểm keypoint tiềm năng trên tất cả các không gian tỷ lệ của ảnh.
 - **Bước 2: Định vị điểm keypoint:**
 - Định vị chính xác các điểm keypoint bằng hàm kiểm tra.
 - **Bước 3: Xác định hướng của điểm keypoint**
 - **Bước 4: Tạo bộ mô tả điểm keypoint:**
 - Mô tả các điểm keypoint dưới dạng vectơ đặc trưng nhiều chiều.
- Lưu ý: Sau khi phát hiện các điểm keypoint và tạo bảng mô tả → So khớp điểm keypoint

- Không gian tỷ lệ

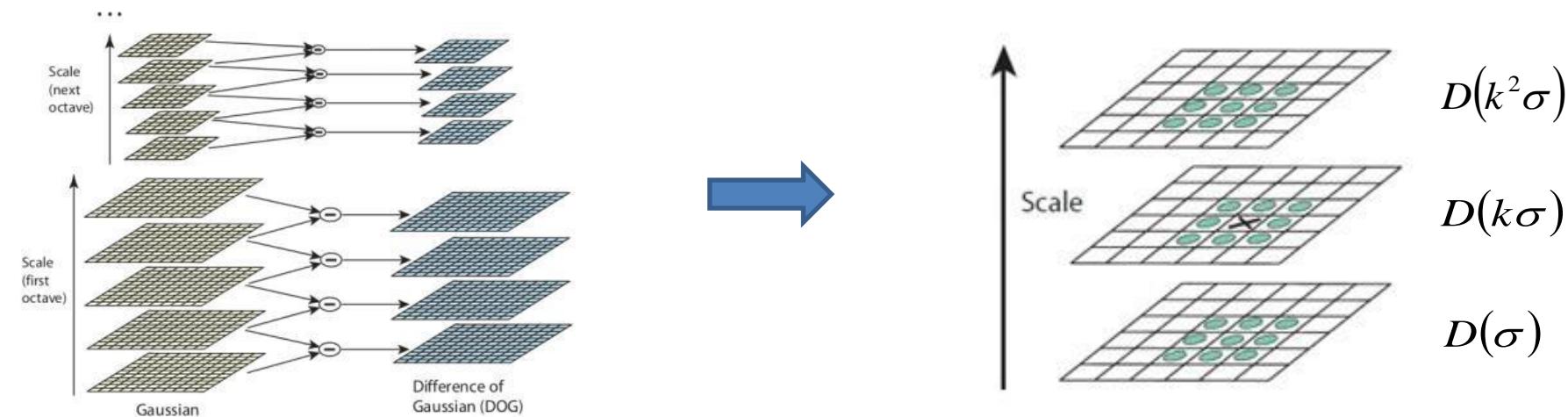
- Không gian tỷ lệ của một ảnh là một hàm $L(x,y,\sigma)$ được tạo ra từ tích chập của bộ lọc Gaussian để làm mờ ở các tỷ lệ khác nhau với ảnh đầu vào.
- Không gian tỷ lệ bao gồm các Octaves, mỗi Octaves là ảnh làm mờ dần từ ảnh đầu vào. Số lượng Octaves và tỷ lệ phụ thuộc kích thước của ảnh gốc
 - Mỗi Octaves bước sau có kích thước bằng một nữa ảnh trước đó
 - Chỉ cần Octaves thứ tư hoặc thứ năm là phù hợp
- Làm mờ (làm mịn):** $L(x,y,\sigma) = G(x,y,\sigma) * I(x,y)$
 - $L(x,y,\sigma)$ là một ảnh mờ
 - $G(x,y,\sigma)$ là toán tử Gaussian $\rightarrow G(x,y,\sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}$
 - $I(x,y)$ là ảnh đầu vào
 - σ là tham số "tỷ lệ" \rightarrow lượng mờ, giá trị càng lớn, độ mờ càng lớn.

Ví dụ về Không gian tỷ lệ và các Octaves



- Chi tiết các bước của thuật toán SIFT

- Bước 1: Tìm kiếm điểm keypoint tiềm năng** → Xác định điểm cực trị trên không gian tỷ lệ
 - Gọi σ là hệ số "tỷ lệ" → lượng mờ → Lượng mờ trong ảnh tiếp theo sẽ là $k\sigma$ (k là hằng số bất kỳ)
 - $\rightarrow L(x,y,k\sigma) = G(x,y,k\sigma) * I(x,y)$
 - \rightarrow Tìm các điểm cực trị của hàm DoG : $D(x,y,\sigma) = (G(x,y,\sigma) - G(x,y,k\sigma)) * I(x,y)$
 - Tìm điểm cực trị: so sánh các điểm trên ảnh DoG so sánh với 8 điểm lân cận và 9 điểm ảnh DoG trước và 9 điểm ảnh DoG sau → điểm cực trị là điểm keypoint tiềm năng



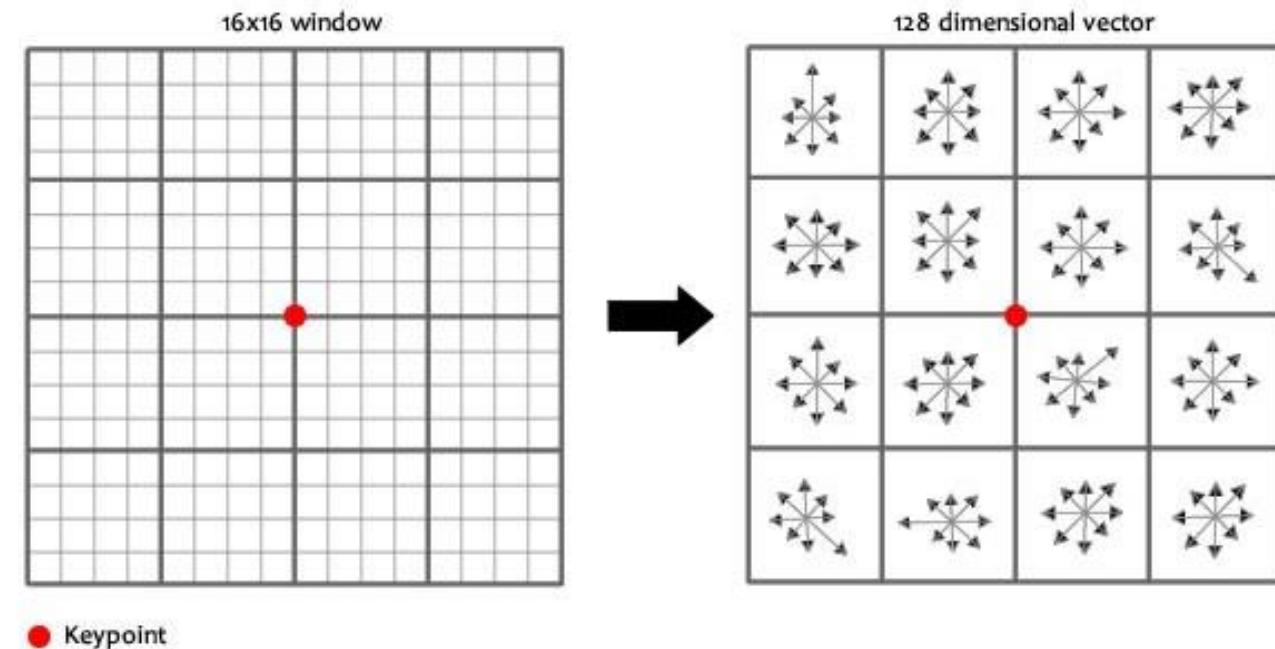
- Chi tiết các bước của thuật toán SIFT
 - Bước 2: Định vị điểm keypoint:** Mỗi điểm keypoint tiềm năng sẽ được đánh giá xem có giữ lại không:
 - Loại bỏ các điểm keypoint có độ tương phản thấp
 - Loại bỏ các điểm keypoint dọc theo các biên (cạnh) do dễ bị nhiễu
 - Bước 3: Xác định hướng cho điểm keypoint:** tính toán biểu đồ hướng Gradient trong vùng láng giềng của điểm keypoint.
 - Độ lớn và hướng của các điểm keypoint xác định theo công thức:

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2}$$

$$\theta(x,y) = \tan^{-1}((L(x,y+1) - L(x,y-1)) / ((L(x+1,y) - L(x-1,y))))$$

4. Phát hiện và mô tả đặc trưng ảnh: Thuật toán SIFT

- Chi tiết các bước của thuật toán SIFT
 - Bước 4: Tạo bộ mô tả điểm keypoint:** Mô tả các điểm keypoint dưới dạng vectơ đặc trưng nhiều chiều.
 - Một cửa sổ 16×16 xung quanh keypoint và chia thành 16 khối phụ có kích thước 4×4 .
 - Mỗi khối phụ có 8 hướng $\rightarrow 4 \times 4 \times 6 = 128$ chiều \rightarrow **Mỗi điểm keypoint được biểu diễn bằng 1 vectơ đặc trưng 128 chiều**



- Phát hiện điểm đặc trưng từ thuật toán SIFT và vẽ trên ảnh

Ảnh gốc



Ảnh chứa keypoint - Thuật toán SIFT



4. Phát hiện và mô tả đặc trưng ảnh: Thuật toán SURF

- Thuật toán SURF (Speeded-Up Robust Features):
 - Được công bố lần đầu tiên vào năm 2009, bởi luật án tiến sĩ của Zurich
 - Được bổ sung bởi H. Bay, A. Ess, T. Tuytelaars, và L. Van Gool
 - Thuật toán SURF **phát hiện và mô tả điểm đặc trưng**.
 - Thực hiện nhanh, áp dụng để biểu diễn và so sánh các hình ảnh tương tự nhau
 - Sử dụng các cửa sổ lọc
 - → cho phép các ứng dụng thời gian thực: phát hiện và nhận dạng đối tượng

4. Phát hiện và mô tả đặc trưng ảnh: Thuật toán SURF

- Các bước thuật toán SURF (Speeded-Up Robust Features): Tương tự như SIFT
 - **Bước 1: Tìm kiếm các điểm keypoint tiềm năng**
 - **Bước 2: Định vị điểm keypoint**
 - **Bước 3: Xác định hướng của điểm keypoint**
 - **Bước 4: Tạo bộ mô tả điểm keypoint**

- Chi tiết các bước thuật toán SURF:

- Bước 1: Tìm kiếm các điểm keypoint tiềm năng**

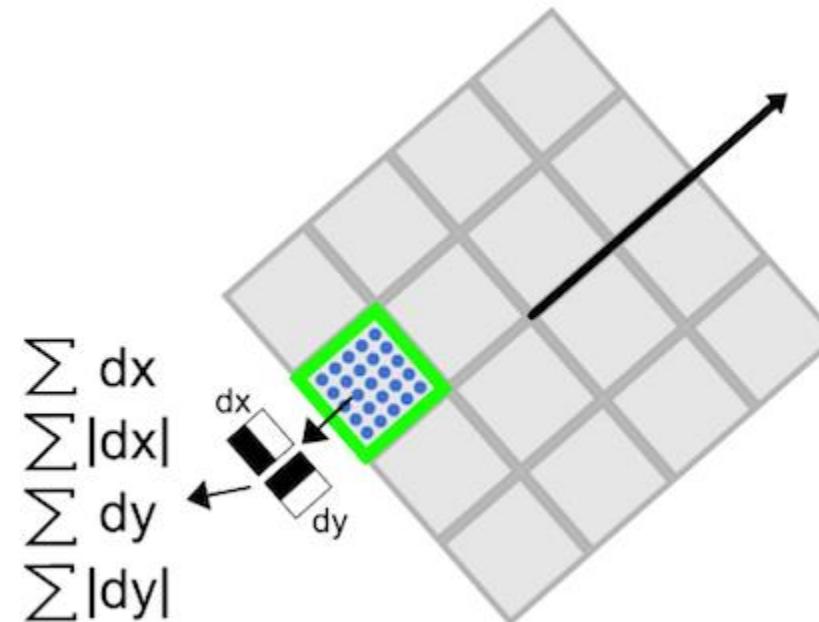
- Giống như thuật toán SIFT nhưng để tạo ra không gian tỷ lệ sử dụng bộ lọc hình vuông → tốc độ xử lý sẽ được cải thiện đáng kể

- Bước 2: Định vị điểm keypoint**

- Dựa trên ma trận Hessian:
$$H(p, \sigma) = \begin{pmatrix} L_{xx}(p, \sigma) & L_{xy}(p, \sigma) \\ L_{yx}(p, \sigma) & L_{yy}(p, \sigma) \end{pmatrix}$$
- Trong đó:
 - p(x,y) là điểm ảnh trong ảnh I(x,y)
 - σ hệ số tỷ lệ (scale)
 - $L_{xx}(p, \sigma)$, $L_{xy}(p, \sigma)$ và $L_{yy}(p, \sigma)$ là tích chập của đạo hàm bậc hai Gaussian với ảnh I(x,y) tại điểm p

4. Phát hiện và mô tả đặc trưng ảnh: Thuật toán SURF

- Chi tiết các bước thuật toán SURF:
 - Bước 3: Xác định hướng của điểm keypoint**
 - Sử dụng phương pháp phản hồi Haar-wavelet theo hướng x và y
 - Bước 4: Tạo bộ mô tả điểm keypoint**
 - Là một vec tơ đặc trưng $4 \times 4 = 64$ chiều $V = (\sum dx, \sum dy, \sum |dx|, \sum |dy|)$



Ví dụ phát hiện đặc trưng ảnh bằng Thuật toán SURF

- Phát hiện điểm đặc trưng từ thuật toán SURF và vẽ trên ảnh

Ảnh gốc



Ảnh chứa keypoint - Thuật toán SURF



4. Phát hiện và mô tả đặc trưng ảnh: Thuật toán ORB

- Thuật toán ORB (Oriented FAST and Rotated BRIEF)
 - Được công bố vào năm 2011 bởi Ethan Rublee, Vincent Rabaud, Kurt Konolige và Gary R. Bradski.
 - Thuật toán ORB **phát hiện và mô tả điểm đặc trưng**
 - Giải pháp thay thế hiệu quả cho SIFT và SURF (đã được cấp bằng sáng chế).
 - Thực hiện tốt như SIFT trong nhiệm vụ phát hiện đặc trưng (và tốt hơn SURF) trong khi thời gian thực thi nhanh hơn.
 - Được xây dựng dựa trên bộ phát hiện điểm đặc trưng FAST
 - Bổ sung thành phần định hướng nhanh và chính xác vào FAST
 - Cải tiến việc tính toán các đặc trưng BRIEF → Phân tích phương sai và mối tương quan đặc trưng BRIEF

- Phát hiện điểm đặc trưng từ thuật toán ORB và vẽ trên ảnh

Ảnh gốc



Ảnh chứa keypoint - Thuật toán ORB



- Sau khi phát hiện các điểm keypoint và tạo bảng mô tả → So khớp điểm keypoint
 - Các điểm keypoint giữa 2 ảnh được so khớp bằng cách xác định các tỷ lệ khoảng cách láng giềng gần nhất (Nearest Neighbor Distance Ratio - NNDR) của chúng
- $$\text{NNDR} = d_1/d_2$$
- Trong đó:
 - d_1 và d_2 là khoảng cách euclid láng giềng gần nhất và láng giềng thứ hai của điểm keypoint
 - Thường tỷ lệ nhỏ hơn hoặc bằng 0,8 là khớp

$$d(p, q) = \sqrt{\sum_{i=1}^N (q_i - p_i)^2}$$

- **Ảnh Panorama:**
 - Máy ảnh thông thường chỉ chụp với một góc 90 độ nên khó có thể thu lại toàn cảnh không gian mong muốn
 - Ảnh panorama là những ảnh có khung hình rộng (chiều rộng lớn hơn nhiều lần chiều cao) được ghép từ nhiều ảnh nhỏ với nhau
- **Image Stitching hay Photo Stitching:**
 - Kết hợp (ghép) 1 tập các ảnh con để tạo nên 1 ảnh lớn, các ảnh con này có các vùng chồng lấp (overlap) nhau.
 - So khớp 2 ảnh dựa trên các đặc trưng (Feature-based)
- **Tạo ảnh Panorama đối với trường hợp 02 ảnh:**
 - Phát hiện các keypoint cho từng ảnh → So khớp 2 tập keypoint → chọn ra các cặp keypoint phù hợp.
 - Dựa vào các cặp keypoint phù hợp → biến đổi phối cảnh (Perspective Transform) ảnh để 02 ảnh có cùng hệ tọa độ → ghép 02 ảnh lại với nhau để tạo ảnh Panorama.

- Thuật toán Stitching đối với 02 ảnh nguồn và đích
 - **Bước 1:** Phát hiện các điểm keypoint và trích xuất các bộ mô tả keypoint của hai ảnh
 - Tập điểm keypoint, tập bộ mô tả điểm keypoint của ảnh nguồn:
 $S_1 = \{k_1, k_2, \dots k_n\}; DS_1 = \{dk_1, dk_2, \dots dk_n\}$
 - Tập điểm keypoint, tập bộ mô tả điểm keypoint của ảnh đích:
 $S_2 = \{k'_1, k'_2, \dots k'_m\}; DS_2 = \{dk'_1, dk'_2, \dots dk'_m\}$
 - **Bước 2:** So khớp các bộ mô tả giữa hai ảnh DS_1 và DS_2
 - **Bước 3:** Sử dụng thuật toán RANSAC tìm ma trận đồng nhất H
 - **Bước 4:** Dựa trên ma trận H thực hiện biến đổi phối cảnh cho ảnh nguồn để được ảnh nguồn đã được biến đổi tọa độ sang tọa độ ảnh đích
 - **Bước 5:** Ghép ảnh nguồn đã được biến đổi tọa độ với ảnh đích để có được ảnh Panorama

- Thuật toán RANSAC (Random Sample Consensus): → Để xác định ma trận đồng nhất H
 - Từ các cặp điểm so khớp giữa 2 ảnh: $(k_1, k'_1), (k_2, k'_2), (k_3, k'_3) \dots (k_n, k'_m)$
 - Chọn 4 cặp điểm và tính ma trận H bằng cách tính hàm mất (loss)

$$\text{Loss} = \sum_{i=0}^n (\text{distance}(H * k_i, k'_i))$$

- Lặp lại quá trình với số lần lặp đủ lớn. Sau đó chọn H có Loss bé nhất

Ảnh gốc hình 1



Ảnh gốc hình 2



Ảnh so khớp



Ảnh Panorama



Ảnh gốc hình 1



Ảnh gốc hình 2



Ảnh so khớp



Ảnh Panorama



