TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÀI TẬP LỚN GIỮA KÌ MÔN XỬ LÝ ẢNH SỐ**

**Phân tích và demo thuật toán SIFT**

*Người hướng dẫn*: **GV PHẠM VĂN HUY**

*Người thực hiện*: **TRẦN QUỐC VINH – 52000823**

Lớp **: 20050201**

Khoá  **: 24**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2024**

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÀI TẬP LỚN GIỮA KÌ MÔN XỬ LÝ ẢNH SỐ**

**Phân tích và demo thuật toán SIFT**

*Người hướng dẫn*: **GV PHẠM VĂN HUY**

*Người thực hiện*: **TRẦN QUỐC VINH – 52000823**

Lớp **: 20050201**

Khoá  **: 24**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2024**

LỜI CẢM ƠN

**ĐỒ ÁN ĐƯỢC HOÀN THÀNH**

**TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Tôi xin cam đoan đây là sản phẩm đồ án của riêng tôi / chúng tôi và được sự hướng dẫn của thầy Phạm Văn Huy. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình.** Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày tháng năm*

*Tác giả*

*(ký tên và ghi rõ họ tên)*

*Trần Quốc Vinh*

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

**Phần xác nhận của GV hướng dẫn**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

**Phần đánh giá của GV chấm bài**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

TÓM TẮT

MỤC LỤC

[LỜI CẢM ƠN i](#_Toc182725122)

[PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN iii](#_Toc182725123)

[TÓM TẮT iv](#_Toc182725124)

[MỤC LỤC 1](#_Toc182725125)

[DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT 3](#_Toc182725126)

[DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ 4](#_Toc182725127)

[1 CHƯƠNG 1 – NỀN TẢNG LÝ THUYẾT 5](#_Toc182949050)

[1.1 Tổng quan về xử lý ảnh 5](#_Toc182949051)

[1.1.1 Lý thuyết 5](#_Toc182949052)

[1.1.2 Thách thức trong xử lý ảnh 8](#_Toc182949053)

[1.2 Feature Detection và thuật toán SIFT 9](#_Toc182949054)

[1.2.1 Feature Detection và điểm yếu của các phương pháp xứ lý truyền thống 9](#_Toc182949055)

[1.2.2 Ưu điểm của việc sử dụng SIFT 9](#_Toc182949056)

[2 Chuyên sâu về thuận toán SIFT 10](#_Toc182949057)

[2.1 Tổng quan về quy trình thực hiện SIFT 10](#_Toc182949058)

[2.2 Xây dựng Scale space bằng cách làm mở ảnh 11](#_Toc182949059)

[2.2.1 Áp dụng Gaussian Blur cho Scale Space 11](#_Toc182949060)

[2.2.2 Tăng cường feature trong Scale Space bằng Difference of Gaussian (DoG) 12](#_Toc182949061)

[2.3 Xác định Keypoint trong ảnh bằng cách so sánh từng pixel với hàng xóm của nó 13](#_Toc182949062)

[2.3.1 Tìm ra local maxima and minima 14](#_Toc182949063)

[2.3.2 Loại bỏ các keypoint có độ tương phản thấp ra 14](#_Toc182949064)

[2.4 Gán Hướng (Orientation Assignment) cho từng keypoint 15](#_Toc182949065)

[2.5 Miêu tả Keypoint 17](#_Toc182949066)

[3 Demo ứng dụng SIFT bằng openCV2 19](#_Toc182949067)

DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT

**CÁC KÝ HIỆU**

**CÁC CHỮ VIẾT TẮT**

DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ

**DANH MỤC HÌNH**

[Hình 1. 1: So sánh xử lý ảnh và thị giác máy tính 5](#_Toc182724701)

[Hình 1. 2: Các loại ảnh số thường gặp 6](#_Toc182724702)

[Hình 1. 3: Các kĩ thuật trong Spatial Domain 7](#_Toc182724703)

[Hình 1. 4: Fourier Transformation để tăng cường ảnh 8](#_Toc182724704)

[Hình 1. 5: Feature Detection trong SIFT 10](#_Toc182724705)

[Hình 2. 1: Scale space được xây dựng từ 4 octave (4 bức ảnh giống nhau nhưng khác kích cỡ) 11](#_Toc182724713)

[Hình 2. 2: Tác dụng của Gaussian blur trong SIFT 12](#_Toc182724714)

[Hình 2. 3: cách DoG hoạt động cho Scale space 13](#_Toc182724715)

[Hình 2. 4: Tìm ra local maxima and minima 14](#_Toc182724716)

[Hình 2. 5: các keypoint nằm gần cạnh và không có ý nghĩa 15](#_Toc182724717)

[Hình 2. 6: Tính orientation và magitude cho keypoint dựa vào neighbor 16](#_Toc182724718)

[Hình 2. 7: Histogram dựa vào magnitude và orientation 17](#_Toc182724719)

[Hình 2. 8: Tìm descriptor của một keypoint 18](#_Toc182724720)

[Hình 3. 1: Matching keypoint giữa 2 bức ảnh số 19](#_Toc182724721)

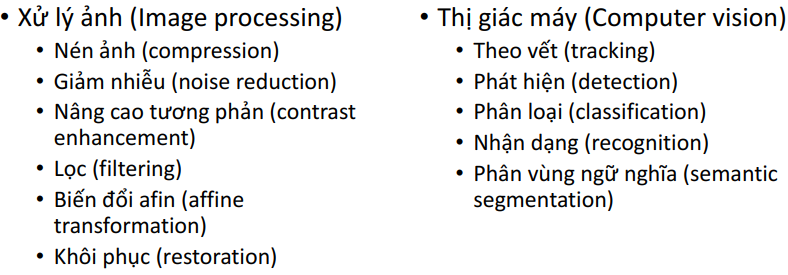
# CHƯƠNG 1 – NỀN TẢNG LÝ THUYẾT

## Tổng quan về xử lý ảnh

### Lý thuyết

Một ảnh số được định nghĩa là một hàm hai chiều, F (x, y), trong đó x và y là các tọa độ không gian, và biên độ của F tại bất kỳ cặp tọa độ (x, y) nào được gọi là cường độ của hình ảnh (pixel intensity) tại điểm đó. Khi các giá trị x, y và biên độ của F là hữu hạn, ta gọi nó là hình ảnh số hóa (digital image). **Nói cách khác, một hình ảnh có thể được xác định bởi một mảng hai chiều, sắp xếp cụ thể theo hàng và cột**

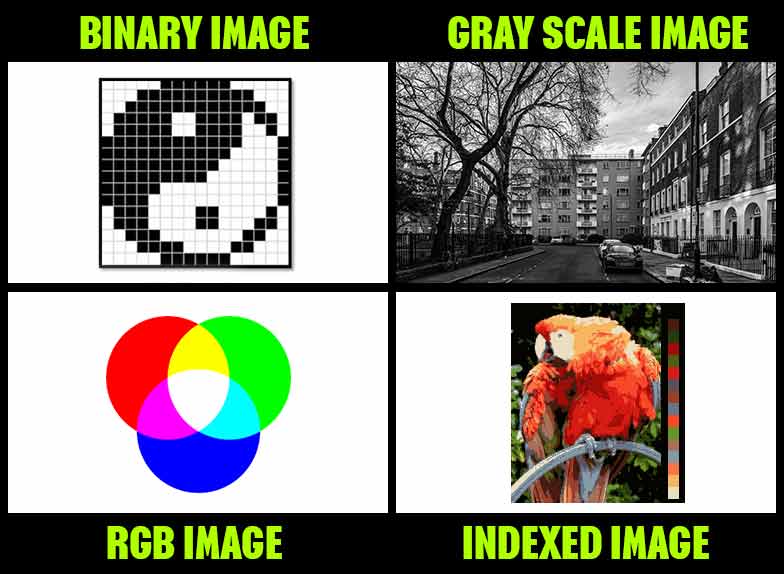
Hình ảnh được biểu thị bằng kích thước (chiều cao và chiều rộng) dựa trên số lượng pixel. Ví dụ: nếu kích thước của hình ảnh là 500 x 400 (chiều rộng x chiều cao), thì tổng số pixel trong hình ảnh là 200000



Hình 1. : So sánh xử lý ảnh và thị giác máy tính

Các loại hình ảnh số thường gặp:

* Ảnh nhị phân: Đúng như tên gọi, ảnh nhị phân chỉ chứa những pixel có giá trị 0 hoặc 1 trong đó 0 chỉ màu đen và 1 chỉ màu trắng. Hình ảnh này còn được gọi là Đơn sắc.
* Ảnh đen-trắng
* Ảnh màu 8 bit (grey image): Đây là định dạng hình ảnh nổi tiếng nhất, có 256 sắc thái màu khác nhau và thường được gọi là hình ảnh thang độ xám. Trong định dạng này, 0 chỉ màu đen, và 255 chỉ màu trắng, và 127 chỉ màu xám
* Ảnh màu 16 bit: Đây là định dạng ảnh màu có 65.536 màu sắc khác nhau, nó còn được gọi là định dạng màu cao. Ở định dạng này, sự phân bố màu sắc không giống như hình ảnh Thang độ xám mà thường được chia thành 3 kênh màu Đỏ, Xanh lục và Xanh lam (RGB)

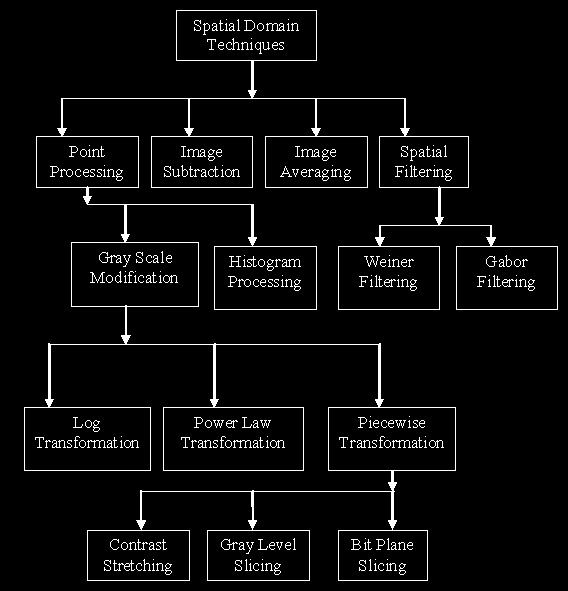


Hình 1. : Các loại ảnh số thường gặp

Xử lý ảnh là một lĩnh vực nghiên cứu liên quan đến việc thao tác trên các hình ảnh số để nâng cao chất lượng, trích xuất thông tin hoặc tạo ra các hình ảnh mới. Đây là nền tảng của nhiều công nghệ hiện đại, từ hình ảnh y tế đến xe tự lái

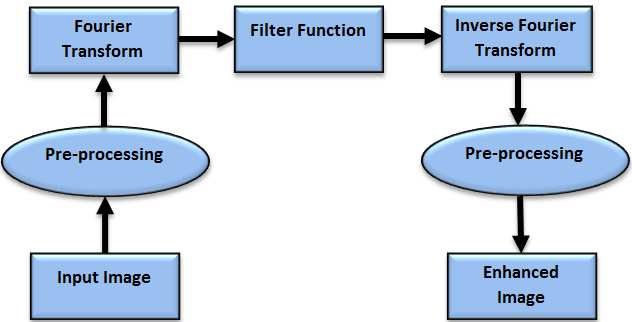
Một số các kĩ thuật trong xử lý ảnh bao gồm

* **Spatial Domain Techniques**:bao gồm các kĩ thuật liên quan đến thao tức, chỉnh sửa như filtering, smoothing, sharpening,.. trực tiếp các pixel trong một bức ảnh nhằm tăng cường, làm mờ,.. ảnh



Hình 1. : Các kĩ thuật trong Spatial Domain

* **Frequency Domain Techniques:** Những kỹ thuật này liên quan đến việc chuyển đổi một hình ảnh từ miền không gian sang miền tần số (Spatial domain to frequency domain), nơi nó được biểu diễn theo các thành phần tần số của nó. Chuyển đổi này cho phép thực hiện các thao tác như giảm nhiễu, nén ảnh và trích xuất đặc trưng. Ví dụ, Fourier Transformation có thể được sử dụng để phân tích nội dung tần số của một hình ảnh



Hình 1. : Fourier Transformation để tăng cường ảnh

### Thách thức trong xử lý ảnh

**Nhiễu hình ảnh:**

* Nhiễu muối tiêu, nhiễu Gaussian, nhiễu đường ngang dọc,... làm giảm chất lượng hình ảnh, gây khó khăn cho các thuật toán xử lý tiếp theo.
* Nguồn gốc nhiễu có thể từ cảm biến, quá trình truyền dẫn, hoặc các yếu tố môi trường.

**Ánh sáng không đồng đều:**

* Ánh sáng quá mạnh hoặc quá yếu, bóng đổ, phản xạ,... làm biến đổi độ sáng và màu sắc của vật thể, ảnh hưởng đến quá trình phân tích hình ảnh.

**Biến dạng hình học:**

* Do góc chụp, ống kính, hoặc quá trình quét, hình ảnh có thể bị méo mó, thay đổi kích thước hoặc tỷ lệ.
* Việc khôi phục lại hình dạng ban đầu của vật thể là một thách thức lớn.

**Phân đoạn và nhận dạng đối tượng:**

* Tách riêng các đối tượng khỏi nền và các đối tượng khác trong hình ảnh là một nhiệm vụ phức tạp, đặc biệt khi các đối tượng có biên mờ, kích thước nhỏ hoặc chồng lấp lên nhau.
* Xác định loại đối tượng trong hình ảnh, đặc biệt khi đối tượng bị che khuất, biến dạng hoặc xuất hiện ở nhiều góc độ khác nhau, là một bài toán khó.

**Xử lý hình ảnh 3D và video:**

* Xử lý hình ảnh 3D đòi hỏi tính toán phức tạp hơn so với hình ảnh 2D, bao gồm các vấn đề như tái tạo bề mặt, ước tính chiều sâu, và phân tích cấu trúc 3D.
* Xử lý video liên quan đến việc xử lý một chuỗi các hình ảnh liên tiếp, đòi hỏi các thuật toán hiệu quả về mặt tính toán để xử lý lượng dữ liệu lớn

## Feature Detection và thuật toán SIFT

### Feature Detection và điểm yếu của các phương pháp xứ lý truyền thống

Feature Detection (phát hiện đặc trưng) là một nhiệm vụ cơ bản trong thị giác máy tính, liên quan đến việc xác định các điểm hoặc vùng đặc biệt (area of interest) trong một hình ảnh mang thông tin quan trọng. Những đặc trưng này có thể được sử dụng cho nhiều ứng dụng xử lý ảnh khác nhau, chẳng hạn như nhận dạng vật thể, khớp nối ảnh,..

**Giới hạn của các phương pháp truyền thống**

Các phương pháp truyền thống, như phát hiện cạnh và phát hiện góc, mặc dù hiệu quả trong một số trường hợp nhất định, nhưng có những hạn chế vốn có.

* **Edge Detection** (Canny Edge Detection): Mặc dù hữu ích để phát hiện các ranh giới giữa các vùng có cường độ khác nhau, nhưng nó nhạy cảm với nhiễu và thay đổi ánh sáng.
* **Corner Detection** (Harris Corner, Moravec Corne, SUSAN Corner,..): Mặc dù hiệu quả trong việc xác định các điểm có độ cong cao, nhưng nó có thể dễ bị nhiễu khi gặp các bức ảnh có textured region (các vùng ảnh có nhiều họa tiết và khó xác định)

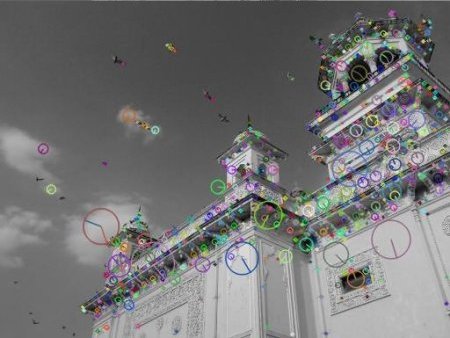
Cả hai phương pháp này đều gặp khó khăn trong việc xử lý khi các bức ảnh cùng một chủ đề nhưng có sự khác biệt về kích cỡ (scale) và từ nhiều góc nhìn khác (rotation), thường dẫn đến việc khó khăn trong trích xuất Feature Detection

### Ưu điểm của việc sử dụng SIFT

The Scale-Invariant Feature Transform (SIFT) là 1 thuật toán để giải quyết hạn chế của các phương pháp xử lý ảnh truyền thống. Trước khi đi vào ưu điểm chính, hãy dành chút thời gian để phân tích tên của thuật toán này. “Scale” trong đây nghĩa là kích cỡ của 1 bức ảnh, “Invariant” tức là không bị ảnh hưởng. Khi hợp 2 từ này lại ta sẽ có “Scale-Invariant” tức là không bị ảnh hưởng bởi kích cỡ của bức ảnh. Cái tên này đã nhấn mạnh rất nhiều ưu điểm mà SIFT mang lại

* **Scale Invariance**: SIFT có thể phát hiện các đặc trưng ở các bức ảnh khác kích cỡ nhưng cùng 1 chủ đề, giúp nó thích hợp để giải quyết bài toán liên quan đến hình ảnh với nhiều kích cỡ/độ phân giải khác nhau
* **Rotation Invariance**: Các đặc trưng của SIFT cũng không bị ảnh hưởng bởi image rotation, điều này đảm bảo SIFT vẫn phát hiện đặc trưng đối tưởng ở nhiều góc chụp khác nhau
* **Illumination Invariance**: SIFT không bị ảnh hưởng với những thay đổi trong điều kiện chiếu sáng, làm cho nó phù hợp với nhiều điều kiện ánh sáng khác nhau.
* **Distinctive Feature**: SIFT tạo ra các đặc trưng rất khác biệt, cho phép khớp chính xác giữa các hình ảnh, ngay cả trong những thay đổi đáng kể trong ảnh

Thuật toán SIFT đã được áp dụng rộng rãi trong nhiều lĩnh vực, bao gồm nhận dạng vật thể, truy xuất hình ảnh và thị giác robot. Độ mạnh mẽ và độ chính xác của nó đã biến nó thành một công cụ giá trị cho nhiều ứng dụng xử lý ảnh



Hình 1. : Feature Detection trong SIFT

# Chuyên sâu về thuận toán SIFT

## Tổng quan về quy trình thực hiện SIFT

Thuật toán SIFT giúp xác định các feature trong một bức ảnh số, thường được gọi là "**Keypoint**" của hình ảnh. Những keypoint này không bị ảnh hưởng khi ảnh có sự thay đổi về góc chụp, độ sáng, độ phân giải. Do đó SIFT được sử dụng cho nhiều ứng dụng thị giác máy tính khác nhau, như khớp nối ảnh, phát hiện vật thể, phát hiện cảnh

Toàn bộ thuật toán SIFT diễn ra trong 4 quá trình chính, có thể chia như sau

* **Xây dựng Scale space bằng cách làm mở ảnh** (Guassian Blur): sử dụng scale space này để đảm bảo các features không bị phụ thuộc vào kích cỡ ảnh
* **Xác định Keypoint trong ảnh bằng cách so sánh từng pixel với hàng xóm của nó**: Xác định keypoint bằng local maxima and minima.
* **Gán Hướng (Orientation Assignment) cho từng keypoint**: đảm bảo các keypoint không bị ảnh hưởng bởi image rotation
* **Định danh Keypoint (Keypoint Descriptor):** gán cho mỗi Keypoint một định danh đặc biệt duy nhất

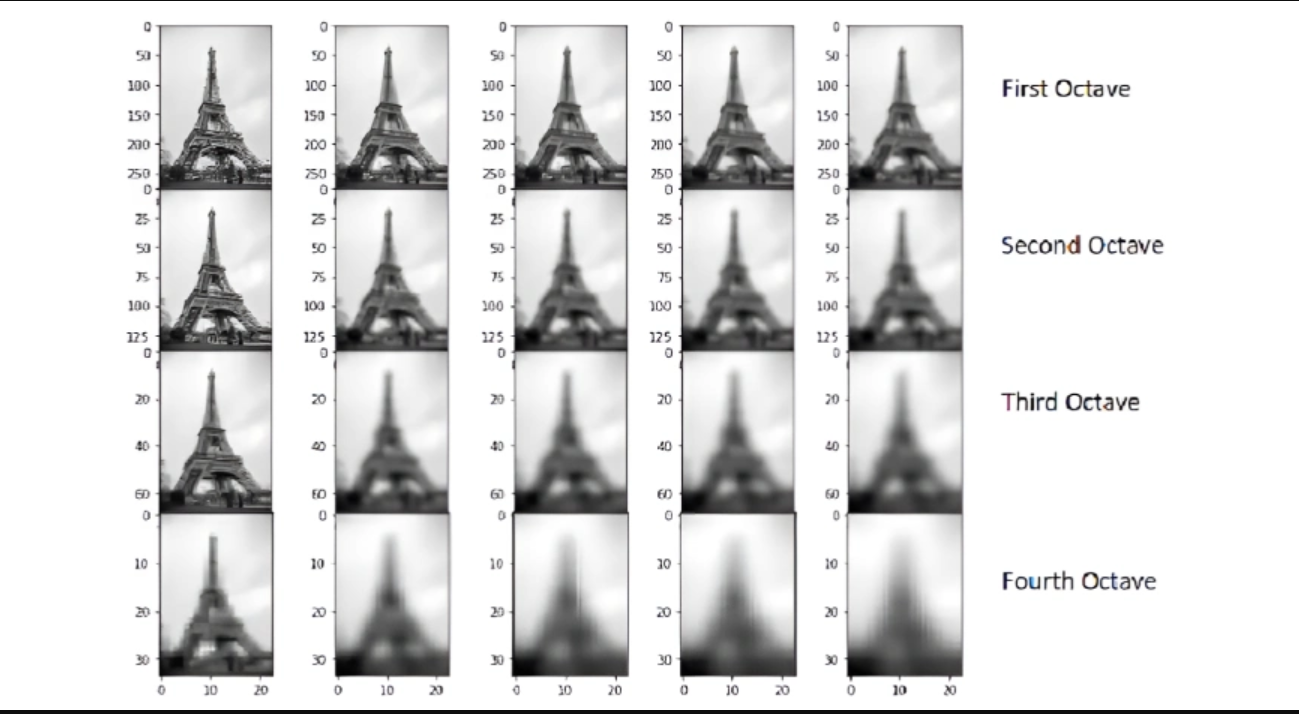
## Xây dựng Scale space bằng cách làm mở ảnh

Áp dụng Gaussian Blur cho Scale Space

Scale space giúp chúng ta xác định được các feature quan trọng nhất trong ảnh đồng thời có 2 ưu điểm như sau

* Xác định được feature bằng cách bỏ qua các điểm nhiễu
* Feature xác định không bị ảnh hưởng bởi scale (kích cỡ) ảnh

Scale space là tập hợp nhiều bức ảnh giống nhau nhưng có kích cỡ khác nhau. Mỗi một kích cỡ ảnh còn được gọi là 1 octave thường sẽ có thêm 4 bức ảnh khác nữa được làm mờ dần từ ảnh gốc

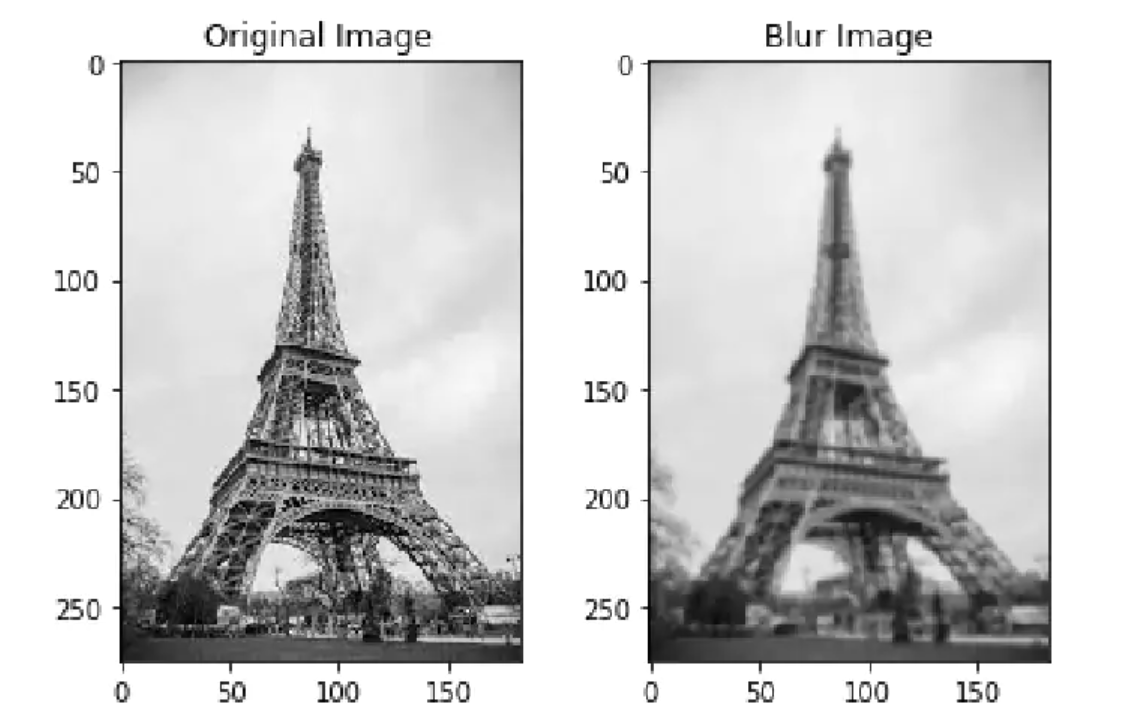


Hình 2. 1: Scale space được xây dựng từ 4 octave (4 bức ảnh giống nhau nhưng khác kích cỡ)

Như ảnh trên, ta thấy được mỗi octave sẽ có thêm 4 bức ảnh được làm mờ từ ảnh gốc

Số octave và ảnh lý tưởng là 4 và 5 (4 octave, 5 ảnh trong từng octave bao gồm cả ảnh gốc). Việc xác định **số lượng octave trong SIFT được xác định bởi kích thước hình ảnh ban đầu và kích thước hình ảnh nhỏ nhất mong muốn**. Mỗi tầng sẽ giảm kích thước hình ảnh đi một nửa theo cả chiều rộng và chiều cao. **Số lượng hình ảnh trong mỗi octave thường được cố định và xác định theo kinh nghiệm (thường là 3 hoặc 5)**.

Một lựa chọn phổ biến là 5 hình ảnh mỗi tầng. Những hình ảnh này được thu được bằng cách tích chập hình ảnh với các bộ lọc Gaussian có giá trị sigma tăng dần. Lý do sử dụng Gaussian Blur là bởi vì nó có thể giúp ta xác định chi tiết quan trọng (cạnh và góc) và loại bỏ nhiễu. Trong ví dụ phía dưới, sau khi blur xong thì thấy chỉ còn các cạnh và góc của tháp Effile là rõ ràng, nhưng chi tiết còn lại đều bị loại bỏ

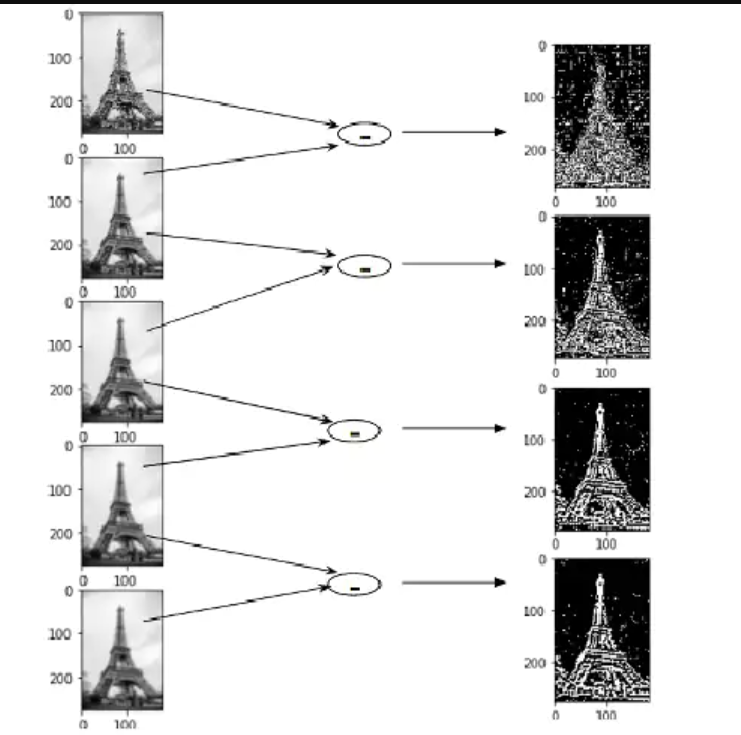


Hình 2. 2: Tác dụng của Gaussian blur trong SIFT

Tăng cường feature trong Scale Space bằng Difference of Gaussian (DoG)

Cho đến nay, chúng ta đã tạo ra các hình ảnh ở nhiều kích cỡ khác nhau (thường được biểu diễn bằng σ) và sử dụng làm Gaussian blur cho từng hình ảnh để giảm nhiễu trong hình ảnh. Tiếp theo, chúng ta sẽ cố gắng tăng cường feature bằng kỹ thuật gọi là DoG

DoG là một thuật toán tăng cường đặc trưng, bao gồm việc trừ một phiên bản mờ của hình ảnh gốc với một phiên bản ít mờ hơn của hình ảnh gốc.DoG tạo ra một tập hợp các hình ảnh khác, cho mỗi octave, bằng cách trừ mỗi hình ảnh với hình ảnh trước đó trong cùng một kích cỡ. Dưới đây là một giải thích trực quan về cách thực hiện DoG



Hình 2. 3: cách DoG hoạt động cho Scale space

## Xác định Keypoint trong ảnh bằng cách so sánh từng pixel với hàng xóm của nó

Một khi Scale space đã được xây dựng, ta bắt đầu đi tìm Keypoint trong ảnh. Bước này bao gồm 2 mục tiêu chính là

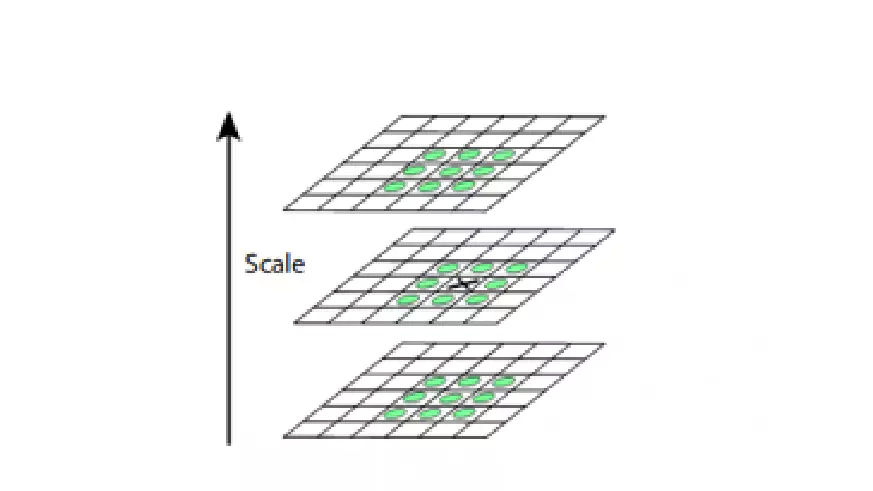
* Tìm ra local maxima and minima.
* Loại bỏ các keypoint có độ tương phản thấp (ít quan trọng) ra

Tìm ra local maxima and minima

Để xác định các điểm cực đại và cực tiểu cục bộ, chúng ta đi qua từng pixel trong ảnh và so sánh nó với các pixel lân cận. Khi chúng ta đề cập "lân cận", điều này không chỉ bao gồm các pixel xung quanh của hình ảnh đó (mà pixel nằm trong đó**) mà còn cả chín pixel cho hình ảnh trước và sau trong octave**.

Điều này có nghĩa là **mỗi giá trị pixel được so sánh với 26 giá trị pixel khác** để tìm xem nó có phải là local maxima and minima hay không. Ví dụ: trong sơ đồ dưới đây, chúng ta có ba hình ảnh từ octave đầu tiên. Pixel được đánh dấu bằng x được so sánh với các pixel lân cận (màu xanh lá cây) và sẽ được chọn làm Keypoint nếu nó có giá trị cao nhất hoặc thấp nhất khi so sánh với 26 pixel khác.

Lý do chọn cao nhất và thấp nhất đơn giản là vì các điểm này thường là pixel quan trọng, thích hợp để làm keypoint feature

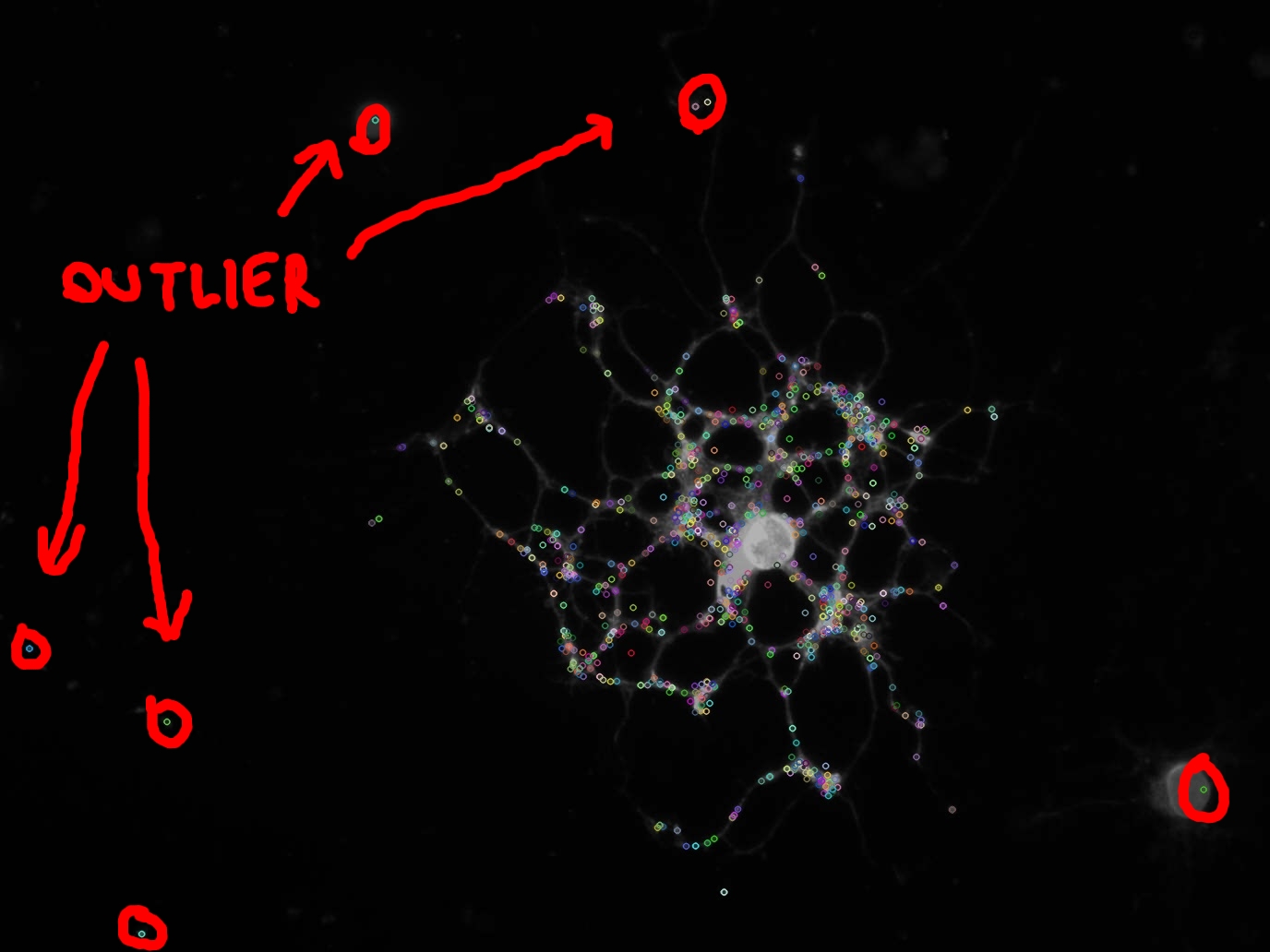


Hình 2. 4: Tìm ra local maxima and minima

Loại bỏ các keypoint có độ tương phản thấp ra

Những keypoint mà chúng ta tìm được bằng cách so sánh với hàng xóm có thể bị ảnh hưởng bởi nhiễu và không mang nhiều ý nghĩa. Do đó, chúng ta sẽ loại bỏ các keypoint có độ tương phản thấp hoặc nằm rất gần cạnh

* **Keypoint có độ tương phản thấp**: khai triển Taylor bậc hai được tính toán cho mỗi điểm khóa. Nếu giá trị kết quả nhỏ hơn 0,03 (về độ lớn), chúng ta loại bỏ keypoint đó
* **Keypoint nằm gần cạnh:** chúng ta thực hiện một kiểm tra để xác định các keypoint. Đây là những keypoint nằm gần cạnh và bị ảnh hưởng bởi nhiễu. Một ma trận Hessian bậc hai được sử dụng để xác định và loại b3 các keypoint như vậy.



Hình 2. 5: các keypoint nằm gần cạnh và không có ý nghĩa

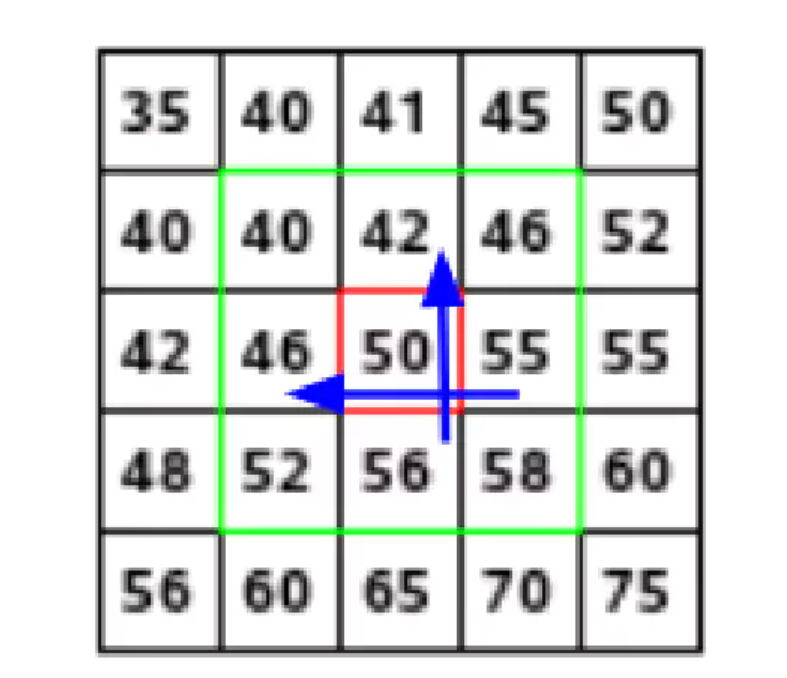
## Gán Hướng (Orientation Assignment) cho từng keypoint

Khi đã có keypoint quan trọng, ta bắt đầu gán hướng (orientation) cho các keypoint đó để chúng không bị ảnh hưởng bởi image rotation. Quá trình này cũng bao gồm 2 bước chính như sau

* Tính độ lớn và hướng (maginitude và orientation)
* Tạo ra biểu đồ histogram cho độ lớn và hướng ấy

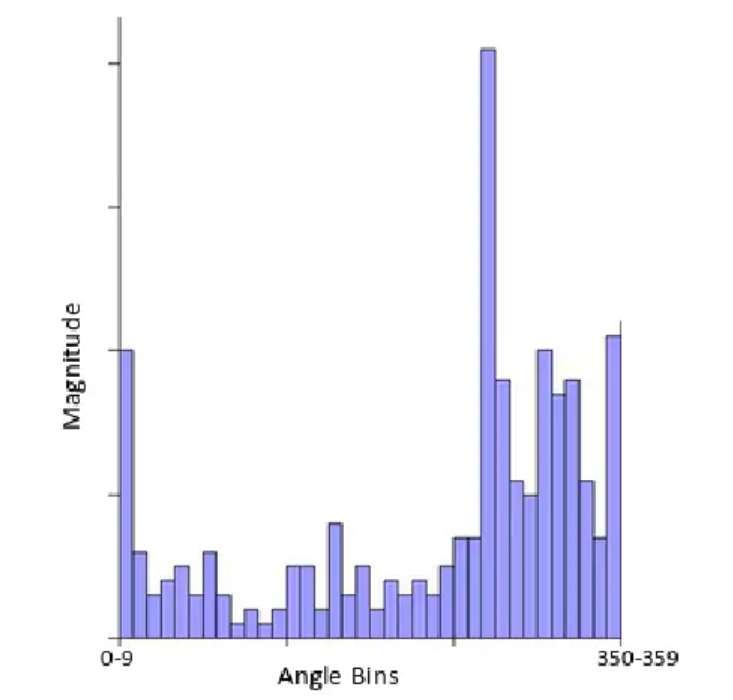
**Việc tính độ lớn sẽ dựa vào gradient và hướng dựa vào arctan của các điểm lân cận xung quanh keypoint đó**

****



Hình 2. 6: Tính orientation và magitude cho keypoint dựa vào neighbor

Sau khi tính 2 yếu tố trên, ta chia bin và tạo ra biểu đồ histogram, Những bin nào có tần số lớn (từ 80-100%) sẽ được chọn để tạo ra thêm keypoint dựa theo hướng và độ lớn đó. Chi tiết cách tính có thể xem thêm tại đây [Khoa học dữ liệu](https://phamdinhkhanh.github.io/2019/11/22/HOG.html)

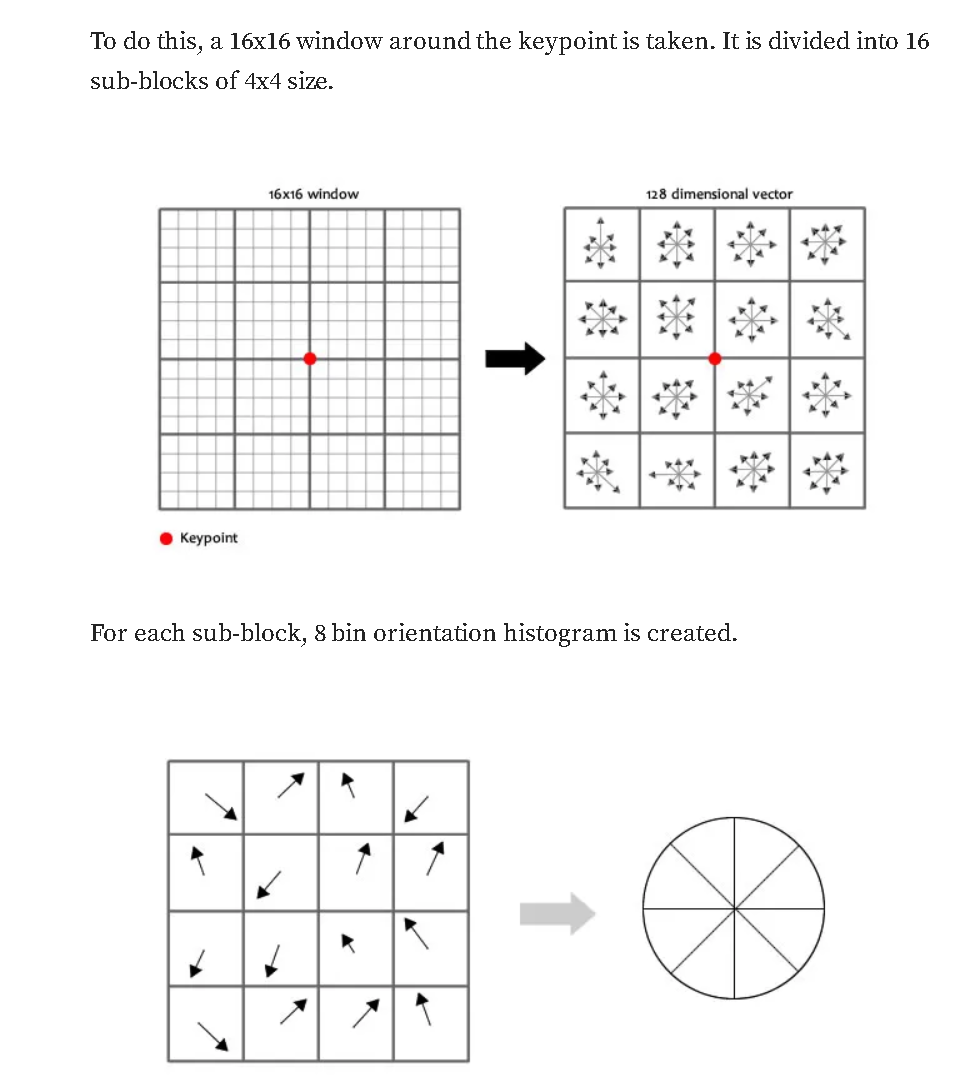


Hình 2. 7: Histogram dựa vào magnitude và orientation

## Miêu tả Keypoint

Bây giờ mỗi keypoint đã có scale, orientation. Tiếp theo ta sẽ tìm descriptor (tạm dịch: đặc trưng) dựa trên các hàng xóm lân cận của keypoint đó. Keypoint descriptor là 1 feature vector có kích cỡ mxm được tìm dựa trên window size 16x16 có tâm là keypoint.

Vùng 16x16 được chia nhỏ thành các sub-window kích cỡ 4x4, mỗi sub-window lại có histogram được tạo ra bởi 8 bin dựa vào orientation



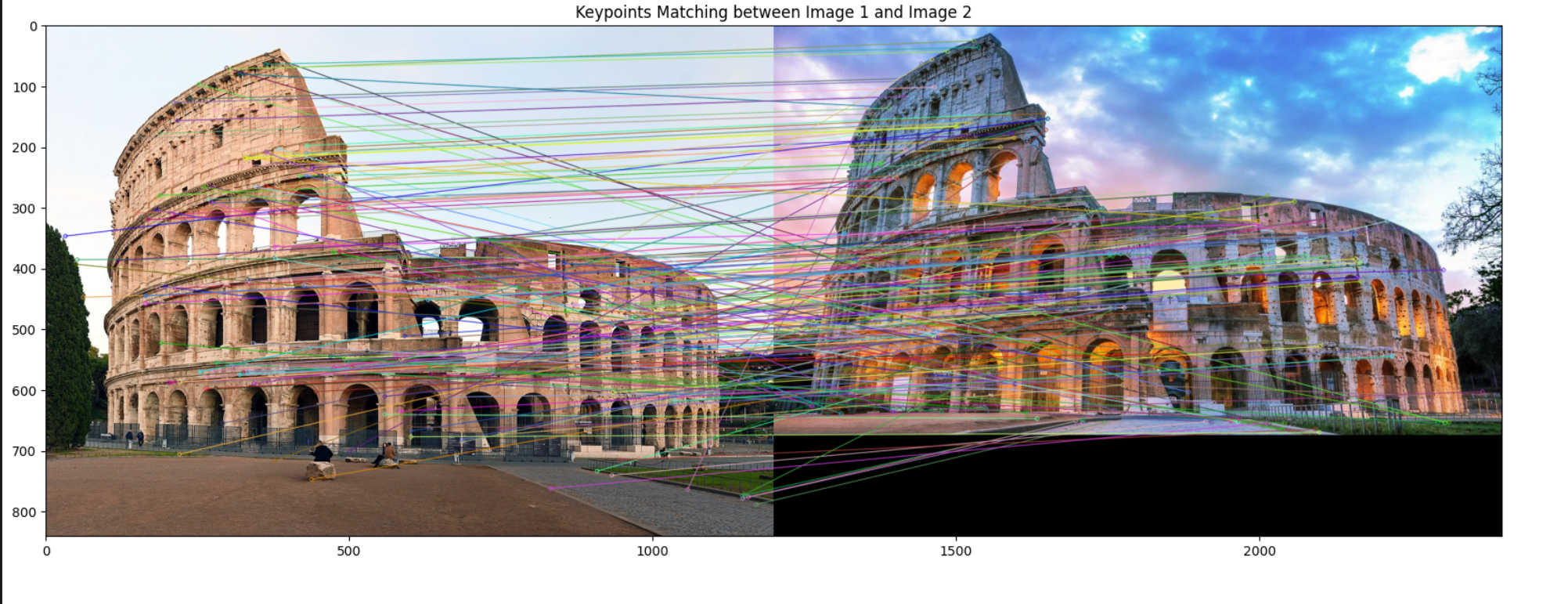
Hình 2. 8: Tìm descriptor của một keypoint

Cuối cùng, một descriptor 4x4 được tạo cho mỗi 16x16 window. **Feature vector còn một số vấn đề cần được chỉnh trước khi biến nó thành “fingerprint” hay còn gọi là “descriptor” cho keypoint**

* Rotation dependence: loại bỏ ảnh hưởng của image rotation trên keypoint bằng cách lấy keypoint’s rotation trừ cho mỗi orientation
* Illumination dependence: loại bỏ ảnh hưởng của độ sáng trên keypoint bằng cách tạo ngưỡng. Bất kì độ lớn nào được tính bằng gradient và lớn hơn 0.2 sẽ được gán bằng 0.2. Tạo ngưỡng giúp giới hạn độ lớn gradient lại
* **Giờ đây chúng ta đã có các keypoint không bị ảnh hưởng bởi scale (dùng scale space), rotation (dùng orientation assignment), illumination (dùng descriptor). Ta có thể dùng các keypoint này để match 2 bức ảnh khác nhau nhưng cùng 1 chủ đề**

# Demo ứng dụng SIFT bằng openCV2

Tìm matching keypoint giữa 2 bức ảnh có cùng chủ đề. Chi tiết xem thêm ở file code



Hình 3. 1: Matching keypoint giữa 2 bức ảnh số

Sau khi đã tìm ra Matching keypoint giữa 2 bức ảnh số, ta có thể ứng dụng chúng vào một số task trong Computer vision như sau

* **Image stiching**: bằng cách xác định matching keypoint giữa 2 bức ảnh, ta có thể ghép 2 bức ảnh lại và tạo ra ảnh panorama
* **Object Recognition**: nhận diện vật thể bằng cách dùng matching keypoint để so sánh 1 bức ảnh mới và 1 bức ảnh đã được nhận diện trước đó
* **3D Reconstruction**: Bằng cách kết hợp thông tin (keypoint) từ nhiều hình ảnh, bạn có thể tái tạo các mô hình 3D của các vật thể hoặc cảnh

**TÀI LIỆU THAM KHẢO**

**Tiếng Việt**

1. [**Xử lý dữ liệu ảnh: Một số kiến thức căn bản - VinBigData**](https://vinbigdata.com/kham-pha/xu-ly-du-lieu-anh-mot-so-kien-thuc-can-ban.html)
2. [**Bài 5: Giới thiệu về xử lý ảnh | Deep Learning cơ bản**](https://nttuan8.com/bai-5-gioi-thieu-ve-xu-ly-anh/)
3. [**Scale Invariant Feature Transform (SIFT)**](https://viblo.asia/p/scale-invariant-feature-transform-sift-6J3Zg0nPlmB)

**Tiếng Anh**

1. [**What is SIFT(Scale Invariant Feature Transform) Algorithm?**](https://www.analyticsvidhya.com/blog/2019/10/detailed-guide-powerful-sift-technique-image-matching-python/#:~:text=The%20SIFT%20(Scale%2DInvariant%20Feature,%2C%20rotation%2C%20and%20affine%20transformations.)
2. [**Introduction to SIFT( Scale Invariant Feature Transform) | by Deep | Medium**](https://medium.com/@deepanshut041/introduction-to-sift-scale-invariant-feature-transform-65d7f3a72d40)
3. [**OpenCV: Introduction to SIFT (Scale-Invariant Feature Transform)**](https://docs.opencv.org/4.x/da/df5/tutorial_py_sift_intro.html)