HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

KHOA CÔNG NGHỆ THÔNG TIN 1

HỌC PHẦN XỬ LÝ ẢNH

Lớp chuyên ngành: D22CNPM02

Nhóm bài tập lớn: 02

ĐỀ TÀI: GHÉP ẢNH PANORAMA

Giảng viên: Phạm Hoàng Việt

Danh sách thành viên nhóm:

Phạm Quang Minh	B22DCCN544
Đoàn Thảo Vân	B22DCCN890

Hà Nội, 12/2015

LỜI NÓI ĐẦU

Trước hết, nhóm em xin gửi lời tri ân sâu sắc và chân thành nhất đến thầy **Phạm Hoàng Việt**, người đã trực tiếp giảng dạy và hướng dẫn nhóm em trong suốt quá trình thực hiện bài báo cáo môn **Xử Lý Ảnh**.

Trong quá trình học tập và nghiên cứu, thầy luôn tận tâm giải đáp thắc mắc, định hướng nội dung và tạo mọi điều kiện thuận lợi để nhóm em có thể tiếp cận kiến thức một cách rõ ràng và hiệu quả nhất. Không chỉ truyền đạt những kiến thức chuyên môn giá trị, thầy còn chia sẻ nhiều kinh nghiệm thực tiễn, giúp nhóm em tự tin hơn trong quá trình tìm hiểu và triển khai đề tài.

Nhờ sự đồng hành và chỉ dẫn của thầy, nhóm em đã có cơ hội củng cố tư duy, mở rộng kiến thức và hoàn thiện kỹ năng nghiên cứu. Mặc dù đã cố gắng hết khả năng, bài báo cáo không tránh khỏi những thiếu sót do kinh nghiệm còn hạn chế. Nhóm em rất mong nhận được những nhận xét, góp ý từ thầy để có thể cải thiện và hoàn thiện hơn trong tương lai.

Một lần nữa, nhóm em xin gửi lời cảm ơn chân thành đến thầy vì sự tận tâm, nhiệt huyết và những hỗ trợ quý báu trong suốt quá trình học tập và thực hiện đề tài. Đây sẽ là những hành trang quan trọng giúp nhóm em tiếp tục phát triển trong học tập và nghiên cứu sau này.

Nhóm em xin trân trọng cảm ơn thầy!

GIỚI THIỆU ĐỀ TÀI

Ghép ảnh Panorama là một kỹ thuật trong xử lý ảnh nhằm kết hợp nhiều bức ảnh riêng lẻ có vùng chồng lắp nhau thành một hình ảnh có góc nhìn rộng hơn, tạo ra một khung cảnh toàn cảnh (panoramic view). Phương pháp này được ứng dụng phổ biến trong nhiếp ảnh, bản đồ số, thị giác máy tính và các hệ thống thực tế ảo.

Trong quá trình ghép ảnh, hệ thống cần thực hiện các bước quan trọng như phát hiện đặc trưng (feature detection), mô tả đặc trưng (feature description), ghép cặp đặc trưng (feature matching), ước lượng biến đổi hình học giữa các ảnh (homography) và cuối cùng là căn chỉnh – trộn ghép các ảnh để tạo ra sản phẩm hoàn chỉnh.

Việc thực hiện đề tài này giúp chúng em hiểu rõ hơn các kỹ thuật cốt lõi trong xử lý ảnh và thị giác máy tính, đặc biệt là các thuật toán như SIFT, ORB, BFMatcher, FLANN hay RANSAC. Thông qua đó, nhóm em có thể xây dựng mô hình tự động ghép ảnh Panorama, đồng thời rèn luyện kỹ năng lập trình và phân tích thuật toán.

MỤC LỤC

DANH SÁCH HÌNH ẢNH

DANH SÁCH BẢNG BIỂU

A. Nội dung

I. SIFT (Scale-Invariant Feature Transform)

1.1. Tổng quan về SIFT

SIFT (Scale-Invariant Feature Transform) là một trong những thuật toán phát hiện và mô tả điểm đặc trưng phổ biến nhất trong thị giác máy tính. Thuật toán này cho phép phát hiện các điểm đặc trưng bất biến với tỉ lệ, xoay và một số biến đổi ánh sáng, rất phù hợp cho các bài toán ghép ảnh (image stitching) nhằm tạo ảnh panorama.

1.2. Keypoint Detection (Phát hiện điểm đặc trưng)

1.2.1. Tổng quan

Keypoint Detection (phát hiện điểm đặc trưng) là một lĩnh vực quan trọng trong thị giác máy tính, đóng vai trò then chốt trong nhiều ứng dụng như nhận dạng vật thể, theo dõi chuyển động, ghép ảnh panorama. Kỹ thuật này nhằm xác định các điểm đặc biệt, có tính phân biệt cao trên ảnh hoặc video, giúp mô tả và nhận diện đối tượng một cách hiệu quả.

Mục tiêu chính của Keypoint Detection là phát hiện và xác định vị trí các điểm nổi bật, đại diện cho cấu trúc hoặc hình dạng đặc trưng của vật thể hoặc cảnh. Các điểm này thường là góc, cạnh hoặc vùng có sự thay đổi cường độ sắc nét. Dữ liệu keypoint sau đó được sử dụng để:

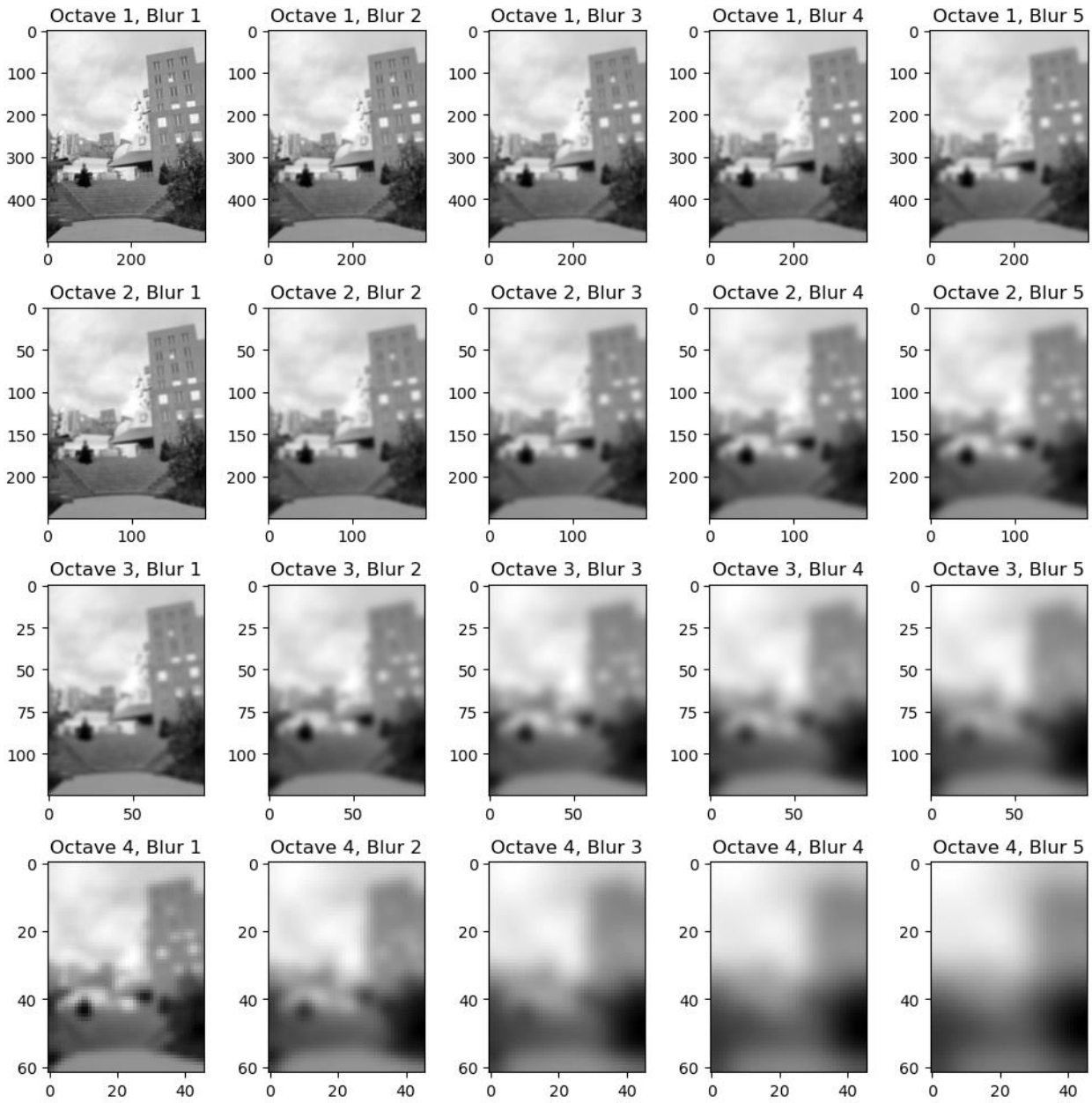
- Nhận dạng và phân loại vật thể.
- Ghép nối các ảnh khác nhau thành ảnh panorama.
- Xác định tư thế người (pose estimation) trong các ứng dụng phân tích hành vi hoặc tương tác người-máy.
- Hỗ trợ trong hệ thống định vị và bản đồ hóa cho robot và thiết bị di động.
- Theo dõi đối tượng trong video.

1.2.2. Các bước xác định điểm đặc trưng của ảnh

1.2.2.1. Xây dựng hình ảnh không gian tỉ lệ(Scale-space)

Trong giai đoạn xây dựng không gian tỉ lệ của thuật toán SIFT, ảnh đầu vào được làm mờ bằng bộ lọc Gaussian với nhiều giá trị sigma khác nhau nhằm tạo ra các lớp ảnh có mức độ chi tiết từ thô đến tinh. Các lớp Gaussian này được sắp xếp thành từng octave, mỗi octave biểu diễn một thang tỉ lệ riêng biệt để mô tả ảnh ở nhiều kích thước quan sát.

Việc xây dựng không gian tỉ lệ giúp hệ thống có khả năng phát hiện các điểm đặc trưng ổn định ngay cả khi ảnh bị phóng to hoặc thu nhỏ, bởi các cấu trúc quan trọng của ảnh luôn xuất hiện tại mức sigma tương ứng trong một octave nhất định. Đây là nền tảng cho các bước tiếp theo như tạo Difference of Gaussian và xác định keypoint bền vững theo tỉ lệ.



Hình 1 Hình ảnh minh họa hình ảnh được chia theo các octave với độ blur khác nhau

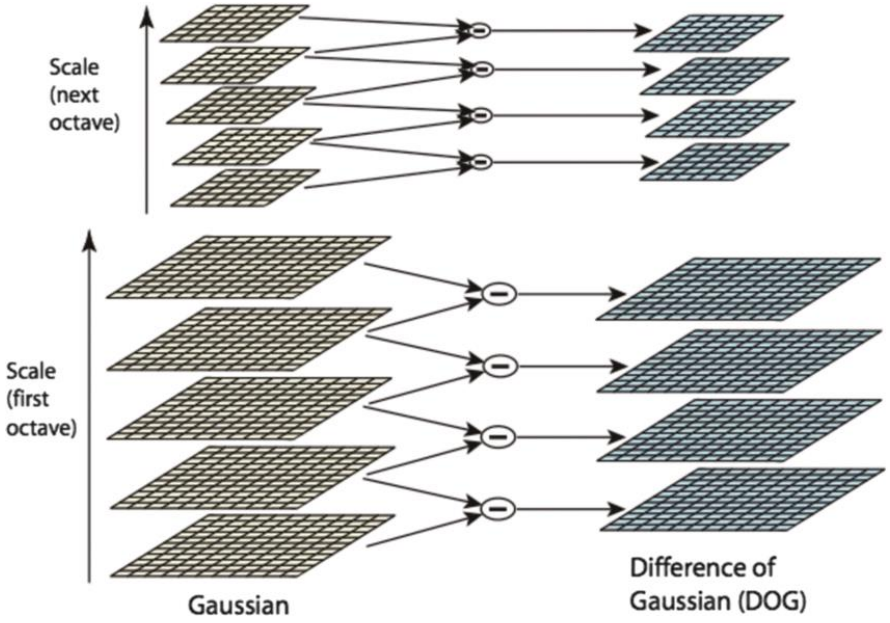
1.2.2.2. Tính hiệu khác của Gaussian (Difference of Gaussian - DoG)

Sau khi hoàn tất việc xây dựng không gian tỉ lệ Gaussian, bước tiếp theo trong thuật toán SIFT là tạo ra các ảnh Difference of Gaussian (DoG) thông qua phép tính hiệu số giữa hai lớp ảnh Gaussian liên tiếp trong cùng một octave. Cụ thể, với mỗi octave, ta thực hiện phép trừ pixel theo từng cặp ảnh Gaussian có giá trị sigma kế cận nhau, từ đó thu được một tập hợp các ảnh DoG biểu diễn sự thay đổi cường độ giữa các mức làm mờ khác nhau.

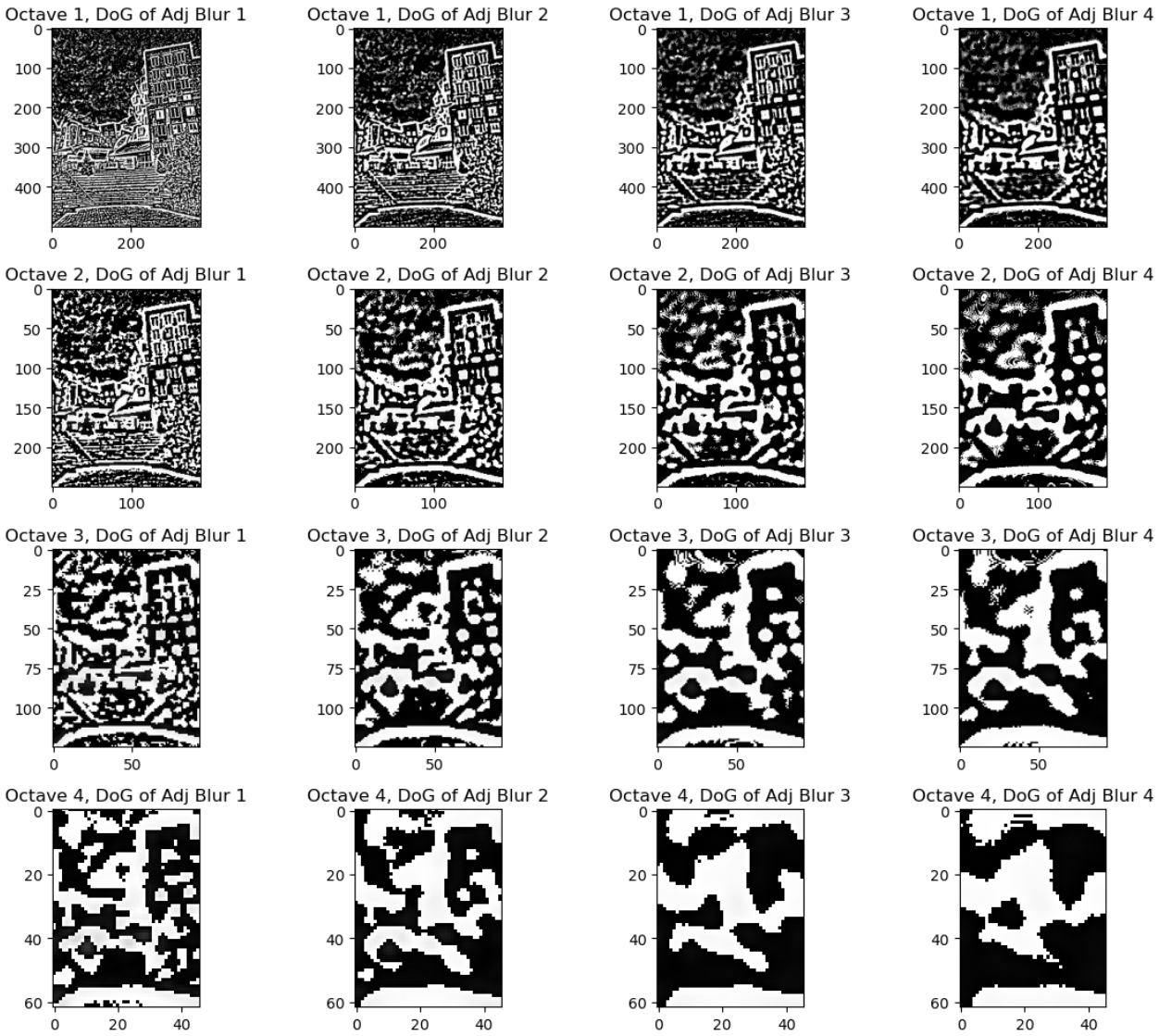
Về mặt lý thuyết, ảnh DoG là một phép xấp xỉ hiệu quả của toán tử Laplacian of Gaussian (LoG), vốn được sử dụng để phát hiện các vùng blob – những vùng có sự thay đổi cường độ nổi bật theo không gian. Nhờ tính chất này, ảnh DoG cho phép hệ thống xác định các điểm có sự biến đổi cường độ sắc nét, tức là những vị trí mà tín hiệu ảnh có phản ứng cực đại hoặc cực tiểu qua nhiều tỉ lệ quan sát. Đây chính là các ứng viên tiềm năng để trở

thành keypoint – những điểm đặc trưng bền vững, ổn định và có khả năng phân biệt cao trong quá trình so khớp ảnh.

Việc sử dụng DoG thay vì LoG không chỉ giảm đáng kể chi phí tính toán mà còn duy trì độ chính xác trong việc phát hiện các cấu trúc quan trọng của ảnh, đóng vai trò nền tảng cho giai đoạn định vị keypoint chính xác ở các bước tiếp theo.



Hình 2 Hình ảnh minh họa tìm kiếm Different of Gaussiaan



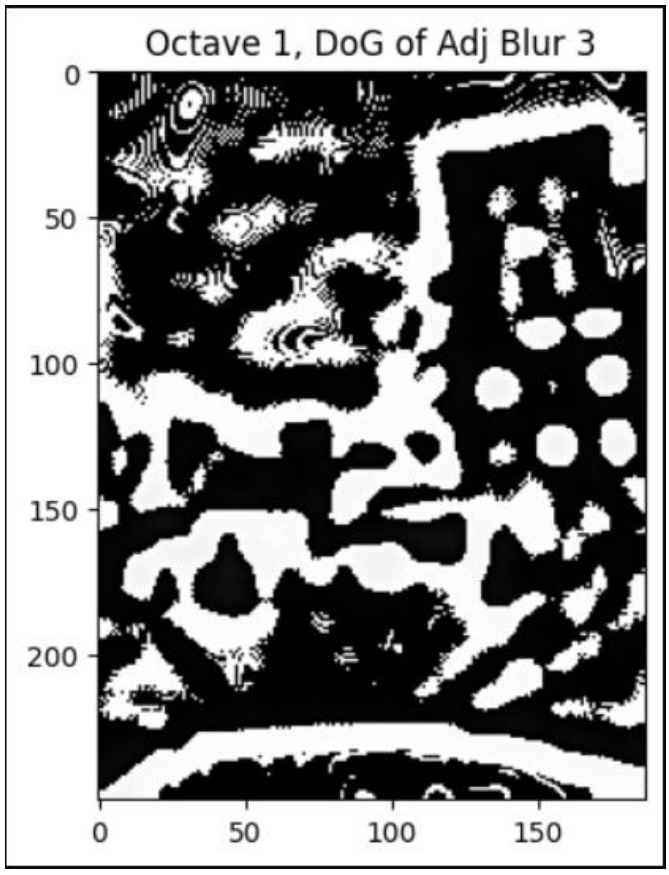
Hình 3 Hình ảnh minh họa kết quả của DoG

1.2.2.3. Phát hiện điểm cực trị (Extrema detection)

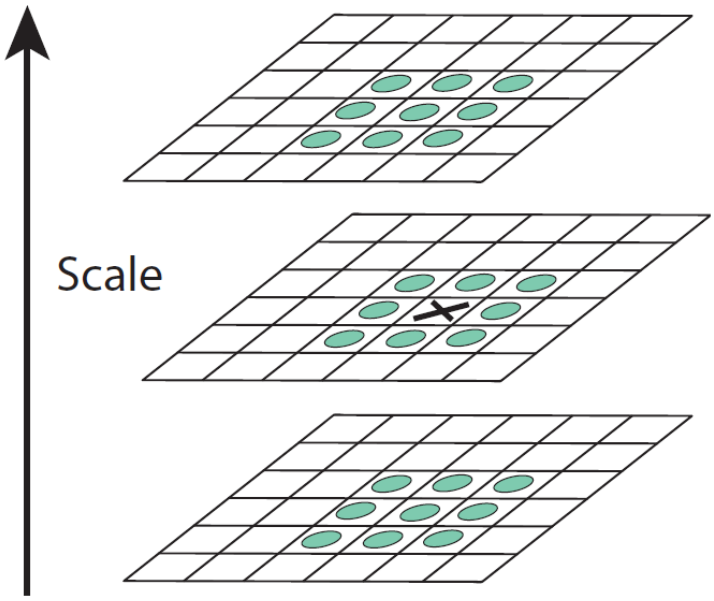
Sau khi thu được tập hợp các ảnh Difference of Gaussian, giai đoạn phát hiện keypoint được thực hiện thông qua việc tìm kiếm các điểm cực trị cục bộ trong không gian ba chiều bao gồm cả vị trí không gian và tỉ lệ. Cụ thể, với mỗi điểm ảnh trong một lớp DoG, hệ thống tiến hành so sánh giá trị của điểm đó với 26 điểm lân cận xung quanh, bao gồm 8 điểm nằm trong cùng lớp DoG hiện tại (các điểm láng giềng theo không gian 2D), 9 điểm nằm trong lớp DoG ngay phía trên (tỉ lệ lớn hơn), và 9 điểm nằm trong lớp DoG ngay phía dưới (tỉ lệ nhỏ hơn). Một điểm được coi là cực trị nếu giá trị cường độ của nó lớn hơn hoặc nhỏ hơn tất cả 26 điểm lân cận này, tức là nó đạt cực đại hoặc cực tiểu cục bộ trong không gian ba chiều DoG.

Lý do sử dụng 26 điểm lân cận thay vì chỉ 9 điểm (trong cùng một lớp) hay 27 điểm (bao gồm cả chính điểm đang xét) là để đảm bảo tính bất biến theo tỉ lệ của keypoint. Nếu chỉ xét 9 điểm lân cận trong cùng một lớp, thuật toán sẽ chỉ phát hiện được các cực trị theo

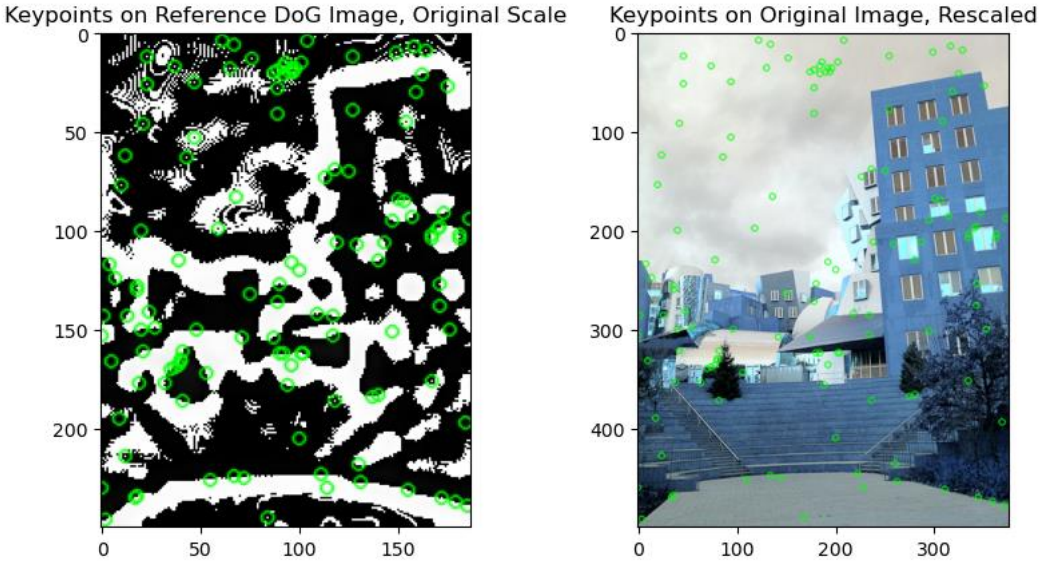
không gian 2D mà bỏ qua chiều tỉ lệ, dẫn đến việc các keypoint không ổn định khi ảnh bị phóng to hoặc thu nhỏ. Ngược lại, nếu bao gồm cả chính điểm đang xét (tổng 27 điểm), phép so sánh trở nên vô nghĩa vì một điểm luôn bằng chính nó. Do đó, việc sử dụng đúng 26 điểm lân cận – loại trừ chính điểm đang xét nhưng bao phủ đầy đủ cả không gian lân cận 3D (chiều rộng, chiều cao và chiều tỉ lệ) – cho phép thuật toán xác định các điểm cực trị thực sự, vừa nổi bật về mặt không gian vừa ổn định qua nhiều tỉ lệ quan sát.



Hình 4 Hình ảnh minh họa của một DoG bất kỳ



Hình 5 Hình ảnh minh họa tìm kiếm cực trị của các pixel



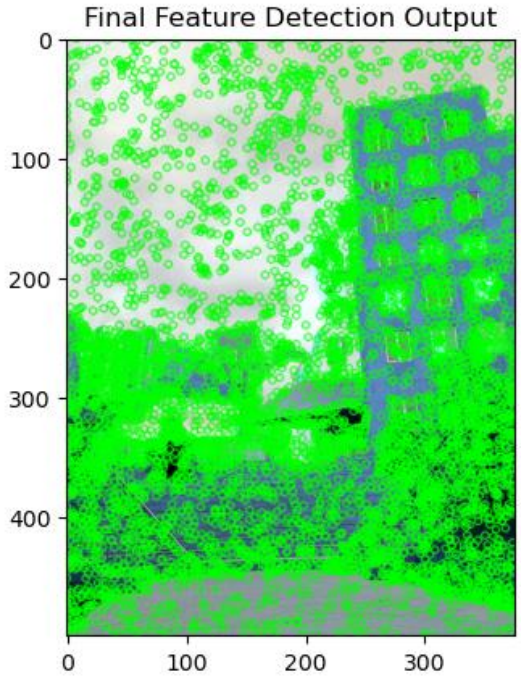
Hình 6 Hình ảnh minh họa sau khi tìm được các điểm cực trị trong một DoG

1.2.2.4. Lọc và xác định vị trí keypoint chính xác

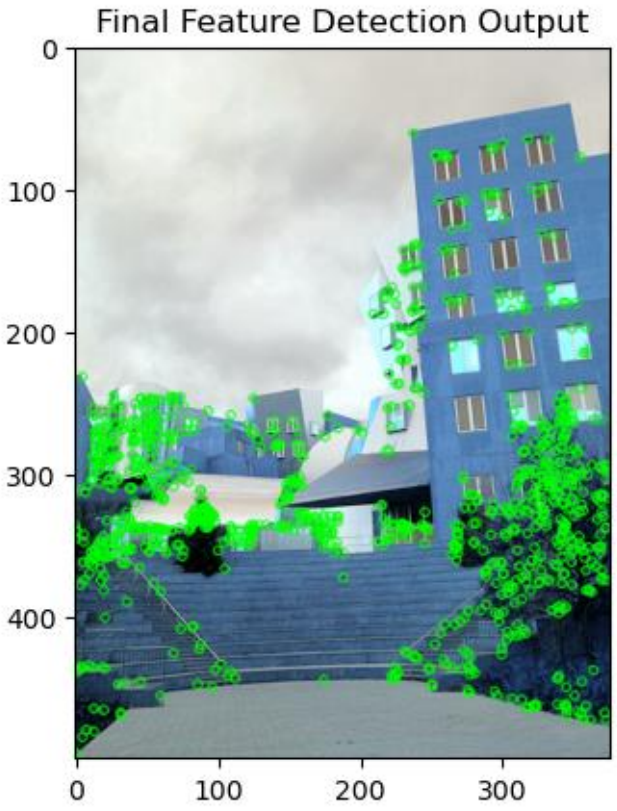
Sau khi xác định được các điểm cực trị tiềm năng, bước tinh chỉnh keypoint được thực hiện nhằm loại bỏ các điểm không ổn định và nâng cao độ chính xác về vị trí. Cụ thể, các điểm có độ tương phản thấp – tức là giá trị DoG quá nhỏ – sẽ bị loại bỏ vì chúng dễ bị ảnh hưởng bởi nhiễu và không mang lại thông tin đặc trưng rõ ràng.

Đồng thời, các điểm nằm trên các cạnh yếu hoặc biên không ổn định cũng bị loại trừ thông qua phân tích ma trận Hessian, giúp phân biệt các điểm góc thực sự với các điểm trên cạnh có độ cong thấp. Ngoài ra, vị trí của mỗi keypoint được tinh chỉnh đến mức độ dưới

pixel (sub-pixel) bằng phương pháp nội suy Taylor, nhằm đảm bảo tọa độ chính xác hơn so với vị trí rời rạc ban đầu. Qua quá trình này, chỉ những keypoint có độ tương phản cao, nằm tại các vị trí góc ổn định và được định vị chính xác mới được giữ lại, tạo nền tảng vững chắc cho bước mô tả đặc trưng tiếp theo.



Hình 7 Hình ảnh minh họa sau khi kiếm được tất cả keypoint trong ảnh



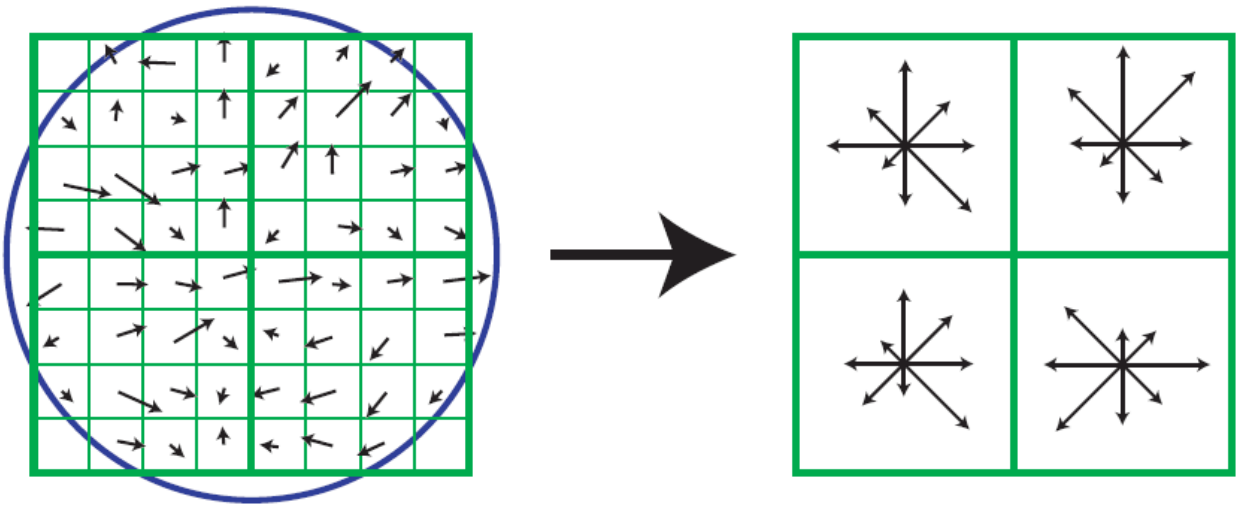
Hình 8 Hình ảnh minh họa các keypoint trong ảnh khi loại bỏ các điểm biên, điểm phẳng

1.3. Descriptor Generation (Tạo mô tả)

1.3.1. Tổng quan

Sau khi phát hiện được vị trí và hướng của mỗi keypoint, SIFT tiến hành tạo descriptor nhằm biểu diễn đặc trưng cục bộ xung quanh điểm đó dưới dạng một vector số. Mục tiêu của descriptor là đảm bảo tính bất biến với các phép biến đổi hình học như tỉ lệ và xoay, đồng thời có khả năng chịu đựng sự thay đổi về ánh sáng và phân biệt rõ ràng giữa các keypoint khác nhau.

Để đạt được điều này, vùng lân cận 16×16 pixel xung quanh keypoint được chia thành lưới 4×4 ô con, mỗi ô chứa 8 bin hướng gradient, tạo thành một histogram mô tả phân bố cường độ và hướng biến thiên cục bộ. Vector descriptor cuối cùng có 128 chiều (4×4×8), được chuẩn hóa để giảm ảnh hưởng của độ sáng toàn cục. Nhờ cấu trúc này, descriptor SIFT vừa nắm bắt được thông tin hình học quan trọng của vùng ảnh, vừa duy trì độ ổn định cao khi ảnh bị biến đổi, đóng vai trò then chốt trong quá trình so khớp điểm giữa các ảnh khác nhau.



Hình 9 Hình ảnh minh họa kết quả của bước keypoint descriptor

1.3.2. Các bước tạo mô tả

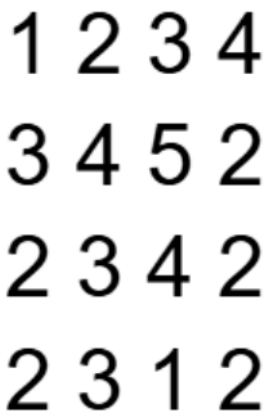
1.3.2.1. Chọn vùng lân cận xung quanh keypoint

Để xây dựng descriptor cho mỗi keypoint, hệ thống trước tiên xác định một vùng vuông có kích thước 16×16 pixel xung quanh keypoint trong ảnh Gaussian tương ứng. Vùng này được xoay theo hướng chủ đạo đã được gán cho keypoint ở bước trước, nhằm đảm bảo rằng descriptor không bị thay đổi khi ảnh bị xoay. Nhờ việc chuẩn hóa hướng này, mọi keypoint đều được biểu diễn trong cùng một hệ tọa độ cục bộ, giúp descriptor trở nên bất biến với phép xoay và tạo điều kiện cho quá trình so khớp chính xác giữa các ảnh có góc nhìn khác nhau.

1.3.2.2. Chia vùng lân cận thành các ô con (subregions)

Vùng 16×16 pixel được chia thành lưới 4×4 ô con, mỗi ô có kích thước 4×4 pixel, tạo thành tổng cộng 16 ô. Việc phân chia này giúp descriptor nắm bắt được cấu trúc không gian

cục bộ một cách chi tiết hơn, đồng thời giảm độ nhạy cảm với nhiễu và biến dạng nhỏ. Thay vì mô tả toàn bộ vùng bằng một histogram duy nhất, mỗi ô sẽ đóng góp thông tin riêng biệt về phân bố gradient tại vị trí đó, tạo nên một biểu diễn phong phú và ổn định hơn cho keypoint.



Hình 10 Hình ảnh ví dụ về một ô kích thước 4x4 pixel

1.3.2.3. Tính gradient biên độ và hướng cho từng pixel

Trước khi tính toán gradient cho mỗi pixel trong vùng descriptor, hệ thống cần xác định đạo hàm riêng theo trục x và trục y từ dữ liệu ảnh rời rạc. Đối với các pixel nằm bên trong vùng ảnh, phương pháp khác biệt trung tâm (central difference) được sử dụng để ước lượng đạo hàm bằng cách lấy hiệu giữa giá trị pixel bên phải và bên trái (cho trục x), hoặc bên trên và bên dưới (cho trục y), sau đó chia cho khoảng cách giữa chúng. Tuy nhiên, tại các pixel nằm trên biên của ảnh, do không có đủ điểm lân cận, phương pháp khác biệt xuôi (forward difference) hoặc khác biệt ngược (backward difference) được áp dụng để tránh truy xuất ngoài vùng ảnh hợp lệ. Cách tiếp cận này đảm bảo tính toán gradient được thực hiện một cách chính xác và an toàn trên toàn bộ vùng descriptor.

- Khác biệt trung tâm (áp dụng cho $1 \le x \le W-2$, $1 \le y \le H-2$):

$$L_x(x, y) = I(x+1, y) - I(x-1, y)$$

$$L_y(x, y) = I(x, y+1) - I(x, y-1)$$

- Tại biên trái (x=0) dùng khác biệt xuôi (forward difference):

$$L_x(0, y) = I(1, y) - I(0, y)$$

- Tại biên phải (x=W−1) dùng khác biệt ngược (backward difference):

$$L_x(W-1, y) = I(W-1, y) - I(W-2, y)$$

- Tại biên trên (y=0) và biên dưới (y=H−1) tương tự cho Ly :

$$L_y(x,0) = I(x,1) - I(x,0), \qquad L_y(x,H-1) = I(x,H-1) - I(x,H-2)$$

Đối với mỗi pixel trong vùng 16×16, hệ thống tính toán hai thành phần quan trọng của gradient: biên độ (magnitude) và hướng (orientation). Biên độ gradient phản ánh cường độ thay đổi cục bộ của ảnh tại điểm đó, trong khi hướng gradient chỉ ra phương mà cường độ thay đổi mạnh nhất. Hai giá trị này thường được tính thông qua đạo hàm Sobel đơn giản, trong đó biên độ được xác định bằng căn bậc hai của tổng bình phương các đạo hàm theo x và y, còn hướng được tính bằng hàm arctan2 của tỉ số giữa chúng. Thông tin gradient này đóng vai trò nền tảng để xây dựng histogram hướng cho từng ô con trong bước tiếp theo.

Biên độ Gradient:

$$m = \sqrt{L_x^2 + L_y^2}$$

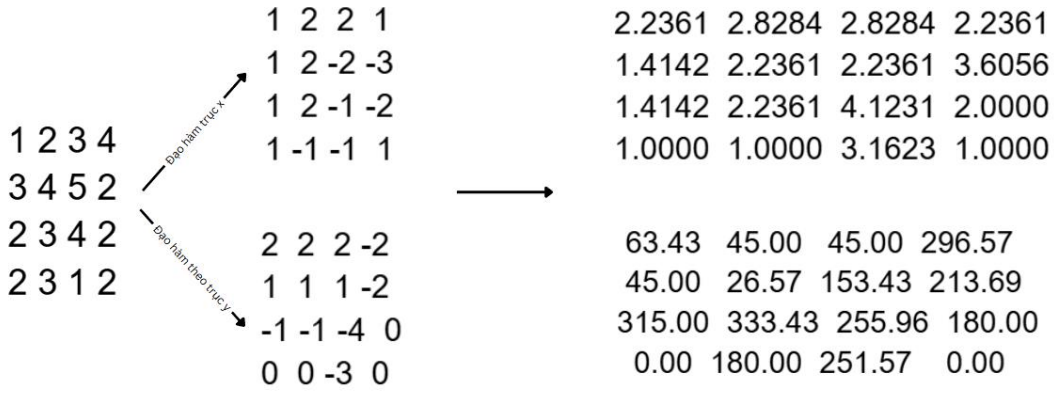
Hướng Gradient:

$$\theta = \tan^{-1}\left(\frac{L_y}{L_x}\right)$$

Trong đó:

- Lx: đạo hàm theo trục x (hướng ngang), với công thức
- Ly: đạo hàm theo trục y (hướng dọc)

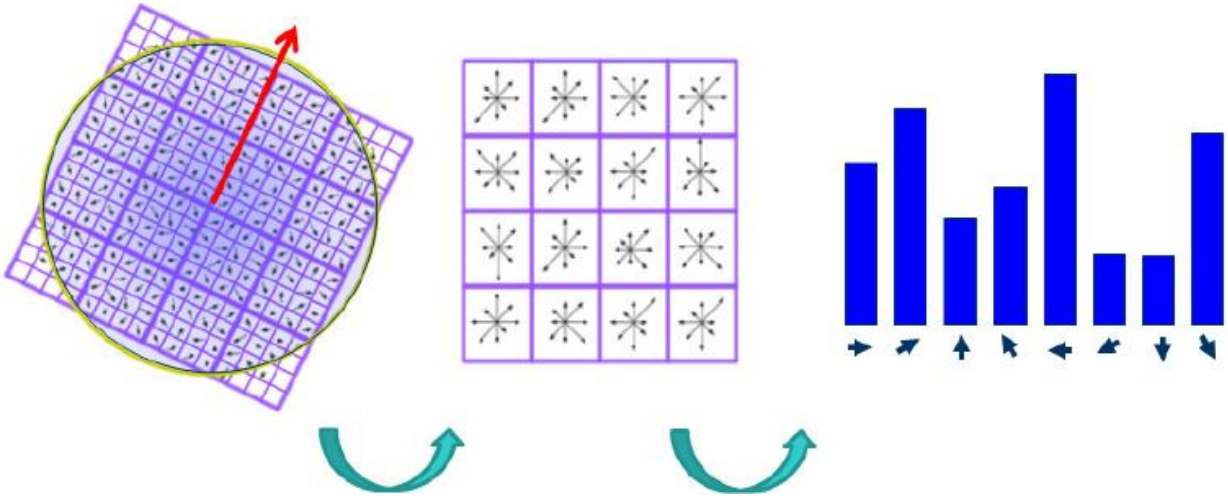
Ví dụ:



1.3.2.4. Tạo histogram hướng trong từng ô con

Mỗi ô 4×4 pixel sẽ tạo ra một histogram gồm 8 bins, tương ứng với 8 hướng gradient chính (0°, 45°, 90°, 135°, 180°, 225°, 270° và 315°). Mỗi pixel trong ô đóng góp vào các bin dựa trên hướng gradient của nó, với trọng số được tính theo hai yếu tố: biên độ gradient tại pixel đó và trọng số Gaussian phụ thuộc vào khoảng cách từ pixel đến tâm keypoint. Việc áp dụng trọng số Gaussian giúp giảm ảnh hưởng của các pixel nằm xa keypoint, đồng thời tăng cường vai trò của các pixel gần tâm – nơi chứa thông tin đặc trưng quan trọng

nhất. Nhờ cơ chế này, descriptor tập trung vào vùng cục bộ có giá trị cao và ít nhạy cảm với nhiễu hoặc biến dạng ở biên ngoài.



Hình 11 Hình ảnh mô trả tạo histogram theo các bin khác nhau

1.3.2.5. Ghép tất cả histogram thành vector 128 chiều

Từ 16 ô con, mỗi ô đóng góp một histogram 8 bins, tổng cộng tạo ra 16 × 8 = 128 giá trị. Các histogram này được ghép nối tuần tự thành một vector descriptor duy nhất có 128 chiều, đại diện cho đặc trưng hoàn chỉnh của keypoint. Vector 128 chiều này nắm bắt đầy đủ thông tin về phân bố gradient cục bộ theo cả không gian và hướng, đồng thời được chuẩn hóa để đảm bảo tính bất biến với sự thay đổi ánh sáng. Đây chính là đặc trưng cuối cùng được sử dụng để so khớp giữa các keypoint trên các ảnh khác nhau trong quá trình ghép ảnh panorama.

1.3.2.6. Chuẩn hóa (Normalization)

Để giảm ảnh hưởng của sự thay đổi ánh sáng hoặc độ tương phản, vector descriptor 128 chiều được chuẩn hóa qua ba bước tuần tự. Đầu tiên, vector được chuẩn hóa theo chuẩn L2 (Euclidean normalization) để đưa độ lớn tổng thể về đơn vị. Tiếp theo, các giá trị lớn hơn 0.2 được cắt ngưỡng (clipping) nhằm hạn chế ảnh hưởng quá mức của các gradient cục bộ mạnh, thường xuất hiện tại các cạnh sắc nét hoặc nhiễu. Cuối cùng, vector được chuẩn hóa lại lần thứ hai để duy trì tính nhất quán. Quy trình này giúp descriptor trở nên ổn định hơn trước các biến đổi về ánh sáng toàn cục hoặc cục bộ, đảm bảo khả năng so khớp chính xác giữa các ảnh chụp trong điều kiện chiếu sáng khác nhau.

Công thức tính Norm:

$$\|\mathbf{v}\| = \sqrt{\sum_{i=1}^{n} v_i^2}$$

Chuẩn hóa từng phần tử:

$$v_i' = \frac{v_i}{\|\mathbf{v}\|}$$

Cắt ngưỡng:

$$v_i'' = \begin{cases} v_i', & \text{nếu } v_i' \leq T \\ T, & \text{nếu } v_i' > T \end{cases}$$

II. Feature Matching (Ghép cặp đặc trưng)

Feature Matching (Ghép cặp đặc trưng) là quá trình tìm và ghép các điểm đặc trưng giống nhau giữa hai hình ảnh khác nhau.

Giả sử ta đã extract được hai tập keypoint trên hai ảnh là:

$$S_1 = \{k_1, k_2, ..., k_n\}$$

và

$$S_2 = \{k_1', k_2', ..., k_m'\}$$

Trong đó:

- S1 là tập keypoint của ảnh thứ nhất, gồm n keypoint.
- S2 là tập keypoint của ảnh thứ hai, gồm m keypoint.

Mỗi keypoint k_i và k_j' đều có Descriptor tương ứng mô tả đặc trưng cục bộ của vùng ảnh quanh keypoint đó.

Ta cần tiến hành so khớp keypoint trong 2 tập này với nhau, tìm ra các cặp keypoint tương ứng trên 2 ảnh. Người ta thường dùng khoảng cách Euclid giữa 2 Descriptor của 2 keypoint để đo độ sai khác giữa 2 keypoint đó.

Để match 2 tập keypoint với nhau, ta có thể dùng 1 trong 2 thuật toán FLANN maching hoặc Bruce Force Maching.

2.1. Brute Force Matching

BF Matching có nghĩa là phải tính toán vét cạn, ta phải tính Euclid distance giữa 1 keypoint k_i trong S_1 với tất cả các điểm trong S_2, từ đó tìm ra các cặp điểm match nhau nhất giữa 2 tập.

Giả sử:

- Ảnh A có n descriptor
- Ảnh B có m descriptor

BruteForceMatcher sẽ:

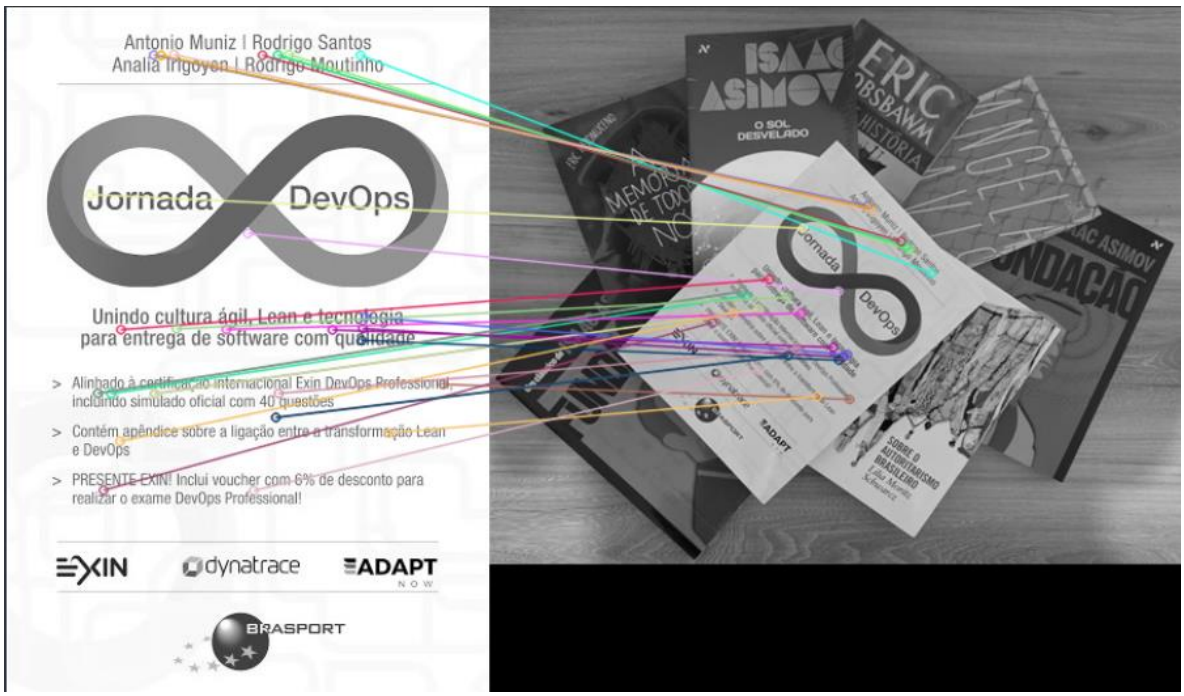
1. Lấy từng descriptor trong ảnh A
2. Tính khoảng cách (Euclidean) đến mọi descriptor của ảnh B
3. Chọn ra descriptor gần nhất (hoặc k gần nhất)

Triển khai thuật toán bằng thư viện **cv2.BFMatcher**:

```
2  # Initiate SIFT detector
3  sift = cv.SIFT_create()
4
5  # find the keypoints and descriptors with SIFT
6  kp1, des1 = sift.detectAndCompute(img1, None)
7  kp2, des2 = sift.detectAndCompute(img2, None)
8
9  # BFMatcher with default params
10 bf = cv.BFMatcher()
11 matches = bf.knnMatch(des1, des2, k=2)
12
13 # Apply ratio test
14 good = []
15 for m, n in matches:
16     if m.distance < 0.75 * n.distance:
17         good.append([m])
```

Hình 12 Hình ảnh minh họa code match

Kết quả:



Hình 13 Hình ảnh minh họa kết quả sau khi sử dụng hàm

2.2. Fast Library for Approximate Nearest Neighbors (FLANN)

FLANN (Fast Library for Approximate Nearest Neighbors) là một thư viện trong OpenCV dùng để tìm điểm gần nhất một cách nhanh hơn BFMatcher, đặc biệt hiệu quả khi: Số lượng mô tả đặc trưng (descriptors) rất lớn, hình ảnh lớn hoặc có nhiều điểm đặc trưng, cần tốc độ nhanh và xấp xỉ (approximate).

Nó không so sánh từng cặp như BFMatcher → mà dùng cấu trúc dữ liệu tối ưu như: KD tree, Hierarchical k-means tree để tìm neighbor.

Triển khai thuật toán bằng thư viện **cv2.FlannBasedMatcher:**

```
# find the keypoints and descriptors with SIFT
kp1, des1 = sift.detectAndCompute(img1, None)
kp2, des2 = sift.detectAndCompute(img2, None)

# FLANN parameters
FLANN_INDEX_KDTREE = 1
index_params = dict(algorithm=FLANN_INDEX_KDTREE, trees=5)
search_params = dict(checks=50)  # or pass empty dictionary

flann = cv.FlannBasedMatcher(index_params, search_params)

matches = flann.knnMatch(des1, des2, k=2)

# Need to draw only good matches, so create a mask
matchesMask = [[0, 0] for i in range(len(matches))]

# ratio test as per Lowe's paper
for i, (m, n) in enumerate(matches):
    if m.distance < 0.7 * n.distance:
        matchesMask[i] = [1, 0]
```

Hình 14 Hình ảnh minh họa code sử dụng Flann

Kết quả:

Hình 15 Hình ảnh minh họa kết quả sử dụng Flann

FLANN thường cho nhiều match hơn BFMatcher vì nó sử dụng cơ chế tìm kiếm "xấp xỉ" (approximate) thay vì tìm chính xác tuyệt đối. FLANN cũng thường áp dụng KNN để lấy nhiều hàng xóm gần nhất cho mỗi descriptor, trong khi BFMatcher (đặc biệt khi dùng cross-check) chỉ nhận một cặp khớp duy nhất. Nhờ đó, FLANN giữ lại nhiều cặp tương đồng hơn, đặc biệt khi số lượng đặc trưng lớn.

III. Homography (Ma trận biến đổi)

3.1. Tổng quan

Trong toán học, Homography là sự dịch chuyển sử dụng phép chiếu hình học, hay nói cách khác nó là sự kết hợp của cặp điểm trong phép chiếu phối cảnh. Ảnh thực trong không gian ba chiều có thể biến đổi về không gian ảnh bằng phép chiếu thông qua ma trận biến đổi Homography hay còn gọi là ma trận H. Các phép chiếu biến đổi thông qua ma trận Homography không đảm bảo về kích thước và góc của vật được chiếu, nhưng lại đảm bảo về tỉ lệ.

Homography là một ánh xạ từ mặt phẳng đối tượng đến mặt phẳng ảnh. Ma trận Homography thường có liên quan đến các công việc xử lý giữa hai ảnh bất kì và có ứng dụng rất rộng rãi trong các công tác sửa ảnh, ghép ảnh, tính toán sự chuyển động, xoay hay dịch chuyển giữa hai ảnh.

3.2. Công thức tính Homography

Homography là ma trận H kích thước 3×3, ánh xạ một điểm từ ảnh 1 sang ảnh 2 theo dạng:

$$x' = Hx$$

Trong đó:

- X là điểm trong ảnh thứ nhất(Tọa độ đồng nhất - homogeneous)
- X' là điểm trong ảnh thứ hai
- H: ma trận homography 3x3, xác định tới một hệ số tỉ lệ(scale factor)

Dạng đầy đủ của công thức:

$$\begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

Tọa độ thực trong ảnh thứ hai:

$$u' = \frac{h_{11}u + h_{12}v + h_{13}}{h_{31}u + h_{32}v + h_{33}}$$

$$v' = \frac{h_{21}u + h_{22}v + h_{23}}{h_{31}u + h_{32}v + h_{33}}$$

Tính chất quan trọng:

- Homography có 8 bậc tự do (9 phần tử nhưng chuẩn hóa theo scale → chia cho một hệ số).
- Cần ít nhất 4 cặp điểm tương ứng để tính được!
- Ánh xạ được các biến đổi: tịnh tiến, xoay, phóng to, thu nhỏ, biến đổi phối cảnh.
- Chỉ áp dụng chính xác khi tất cả điểm nằm trên cùng một mặt phẳng 3D hoặc camera chỉ quay quanh tâm mà không dịch chuyển.

IV. RANSAC (Random Sample Consensus)

4.1. Tổng quan

RANSAC (Random Sample Consensus) là một thuật toán lặp mạnh mẽ được thiết kế để ước lượng các mô hình toán học như homography, fundamental matrix, đường thẳng hoặc mặt phẳng từ tập dữ liệu chứa nhiều nhiễu và điểm ngoại lai (outliers). Khác với các phương pháp bình phương tối thiểu (least-squares) truyền thống vốn rất nhạy cảm với outliers và dễ bị lệch kết quả, RANSAC hoạt động theo nguyên lý chọn lọc: chỉ sử dụng một tập con nhỏ các điểm dữ liệu "tốt" (inliers) để xây dựng mô hình ứng viên, sau đó đánh giá mức độ phù hợp của mô hình đó với toàn bộ tập dữ liệu. Qua nhiều vòng lặp ngẫu nhiên, RANSAC xác định được mô hình có số lượng inliers lớn nhất, từ đó loại bỏ hiệu quả ảnh hưởng của các điểm khớp sai trong quá trình tính toán homography cho ghép ảnh panorama.

4.2. Ý tưởng của RANSAC

Ý tưởng cốt lõi của RANSAC dựa trên quy trình lặp đơn giản nhưng hiệu quả. Đầu tiên, thuật toán chọn ngẫu nhiên một tập con nhỏ các điểm dữ liệu, gọi là minimal sample, đủ để ước lượng mô hình – ví dụ, 4 cặp điểm tương ứng để tính homography bằng phương pháp DLT.

Tiếp theo, mô hình ứng viên được xây dựng từ tập con này và được đánh giá bằng cách đo lường mức độ phù hợp với toàn bộ dữ liệu, thông qua việc đếm số lượng điểm có sai số dưới ngưỡng cho phép (inliers). Quá trình này được lặp lại nhiều lần với các tập con ngẫu nhiên khác nhau, và cuối cùng mô hình có số lượng inliers cao nhất được chọn làm kết quả cuối cùng. Nhờ cơ chế này, RANSAC loại bỏ hiệu quả các outliers mà không cần biết trước phân bố của chúng.

4.3. Quy trình chi tiết của RANSAC

4.3.1. Khởi tạo tham số

Thuật toán RANSAC yêu cầu thiết lập một số tham số quan trọng để kiểm soát quá trình tìm kiếm mô hình tối ưu.

Tham số N xác định số vòng lặp thực hiện, thường dao động từ vài trăm đến vài nghìn tùy thuộc vào tỉ lệ outliers dự kiến trong dữ liệu. Tham số k là số điểm tối thiểu cần thiết để ước lượng mô hình – đối với homography, k = 4 cặp điểm tương ứng. Ngưỡng sai số t được sử dụng để phân loại một điểm là inlier hay outlier dựa trên khoảng cách từ điểm đó đến mô hình dự đoán. Tham số d xác định số lượng inliers tối thiểu mà một mô hình cần có để được coi là hợp lệ. Trong suốt quá trình lặp, hệ thống lưu trữ mô hình tốt nhất H* cùng với tập inliers tương ứng, và cập nhật chúng mỗi khi tìm thấy mô hình có số lượng inliers cao hơn.

4.3.2. Lặp RANSAC và tính ma trận Homography

4.3.2.1. Chọn các cặp diểm

Thuật toán RANSAC thực hiện quy trình lặp qua N vòng, trong đó mỗi vòng lặp bao gồm các bước chọn mẫu, ước lượng mô hình và đánh giá inliers. Tuy nhiên, cần làm rõ rằng trong cài đặt chuẩn của RANSAC, mỗi vòng lặp đều chọn ngẫu nhiên độc lập k điểm (với homography là 4 cặp điểm) từ toàn bộ tập dữ liệu ban đầu, không phụ thuộc vào kết quả của các vòng lặp trước. Việc chọn mẫu ngẫu nhiên hoàn toàn mới mỗi lần giúp đảm bảo khả năng khám phá không gian giải pháp rộng hơn và tránh bị kẹt ở các tập inliers cục bộ.

Sau khi ước lượng mô hình từ 4 điểm ngẫu nhiên, hệ thống kiểm tra toàn bộ tập dữ liệu để đếm số lượng inliers và lưu lại mô hình tốt nhất có số inliers cao nhất qua tất cả các vòng lặp. Chỉ sau khi hoàn thành N vòng lặp, tập inliers cuối cùng từ mô hình tốt nhất mới được sử dụng để tinh chỉnh lại homography bằng phương pháp least-squares, nâng cao độ chính xác của kết quả.

4.3.2.2. Cách tính ma trận Homography

a. Thiết lập phương trình

Với mỗi cặp điểm, ta có:

$$(u_i, v_i) \leftrightarrow (u_i', v_i')$$

Khi này u', v' sẽ tính theo công thức:

$$u_i' = \frac{h_{11}u_i + h_{12}v_i + h_{13}}{h_{31}u_i + h_{32}v_i + h_{33}}$$

$$v_i' = \frac{h_{21}u_i + h_{22}v_i + h_{23}}{h_{31}u_i + h_{32}v_i + h_{33}}$$

Nhân chéo để loại mẫu số:

$$u_i' h_{31}u_i + u_i' h_{32}v_i + u_i' h_{33} = h_{11}u_i + h_{12}v_i + h_{13}$$

$$v_i' h_{31}u_i + v_i' h_{32}v_i + v_i' h_{33} = h_{21}u_i + h_{22}v_i + h_{23}$$

b. Viết lại thành dạng ma trận

Mỗi cặp điểm tạo ra 2 phương trình, dạng:

$$\begin{bmatrix} -u_i & -v_i & -1 & 0 & 0 & 0 & u_i'u_i & u_i'v_i & u_i' \\ 0 & 0 & 0 & -u_i & -v_i & -1 & v_i'u_i & v_i'v_i & v_i' \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{bmatrix} = 0$$

Lúc này hệ phương trình tổng quát:

$$Ah = 0$$

Trong đó:

- A: ma trận 2n*9
- H: vector gồm 9 phần tử của H

c. Ước lượng mô hình tạm thời bằng DLT

Sử dụng 4 cặp điểm vừa chọn ngẫu nhiên để tính ma trận Homography H thông qua phương pháp Direct Linear Transformation (DLT). Mỗi cặp điểm tương ứng (x_i, y_i) trong ảnh nguồn và (x'_i, y'_i) trong ảnh đích tạo ra hai phương trình tuyến tính ràng buộc các phần

tử của ma trận H. Với 4 cặp điểm, hệ thống thu được 8 phương trình độc lập, đủ để giải hệ phương trình tuyến tính thuần nhất Ah = 0, trong đó A là ma trận hệ số 8×9 và h là vector chứa 9 phần tử của ma trận H được sắp xếp thành cột.

Nghiệm của hệ này được tìm bằng phân tích SVD (Singular Value Decomposition) trên ma trận A, lấy vector riêng tương ứng với giá trị riêng nhỏ nhất. Ma trận H thu được từ DLT là mô hình ứng viên dùng để đánh giá số lượng inliers trong bước tiếp theo.

c. Tính sai số chiếu lại (Reprojection Error)

Với mỗi cặp điểm tương ứng trong toàn bộ tập dữ liệu, hệ thống chiếu điểm (x, y) từ ảnh nguồn qua ma trận Homography H để thu được điểm dự đoán (x', y') trong ảnh đích.

$$(x, y) \leftrightarrow (x', y')$$

Sai số được tính bằng khoảng cách Euclidean giữa điểm dự đoán này và vị trí thực tế của điểm tương ứng trong ảnh đích.

$$e = \sqrt{(x' - \hat{x}')^2 + (y' - \hat{y}')^2}$$

Nếu sai số e nhỏ hơn ngưỡng t đã thiết lập, điểm đó được phân loại là inlier – tức là phù hợp với mô hình hiện tại. Ngược lại, nếu e ≥ t, điểm được coi là outlier và bị loại bỏ khỏi tập hỗ trợ mô hình. Quá trình này được thực hiện cho tất cả các cặp điểm, cho phép đánh giá chất lượng của mô hình Homography ứng viên thông qua tổng số inliers thu được.

d. Đánh giá mô hình

Sau khi phân loại toàn bộ các cặp điểm, hệ thống đếm tổng số inliers của mô hình Homography hiện tại. Nếu số lượng inliers này lớn hơn số lượng inliers tốt nhất đã ghi nhận từ các vòng lặp trước, hệ thống sẽ cập nhật mô hình tốt nhất bằng cách lưu lại ma trận H hiện tại cùng với tập inliers tương ứng. Cơ chế cập nhật này đảm bảo rằng sau N vòng lặp, RANSAC sẽ giữ lại mô hình có khả năng giải thích nhiều dữ liệu nhất, tức là mô hình ít bị ảnh hưởng nhất bởi các điểm khớp sai và phản ánh đúng nhất phép biến đổi hình học giữa hai ảnh.

$$H_{\text{best}} = H$$

$$\text{Inliers}_{\text{best}} = \text{tập inlier hiện tại}$$

e. Tính lại ma trận Homography lần cuối cùng

Sau khi hoàn thành N vòng lặp và xác định được tập inliers tốt nhất, ma trận Homography được tính toán lại lần cuối cùng bằng cách sử dụng toàn bộ các inliers thay vì chỉ 4 điểm ngẫu nhiên ban đầu. Việc tái ước lượng này thường được thực hiện thông qua phương pháp least-squares hoặc DLT áp dụng trên tập inliers đầy đủ, giúp tận dụng tối đa thông tin từ các điểm khớp chính xác và giảm thiểu sai số tích lũy. Ma trận H cuối cùng thu

được sẽ chính xác hơn đáng kể so với các mô hình ứng viên trong quá trình lặp, đồng thời vẫn đảm bảo tính ổn định nhờ loại bỏ hoàn toàn ảnh hưởng của outliers, tạo nền tảng vững chắc cho bước ghép ảnh panorama tiếp theo.

V. Image Warping, Blending và Cropping (Biến đổi ảnh, Trộn ảnh và cắt viền)

5.1. Image Warping.

5.1.1. Khái niệm

Image Warping là quá trình biến đổi hình học một ảnh từ hệ tọa độ ban đầu sang một hình dạng hoặc vị trí mới trong hệ tọa độ khác, thông qua một phép biến đổi hình học (geometric transformation). Các phép biến đổi này có thể là: biến đổi affine, projective (homography), biến dạng phi tuyến, hoặc ánh xạ theo ma trận.

Mục tiêu của warping là ánh xạ mỗi điểm ảnh (x,y) trong ảnh gốc sang vị trí mới (x′,y′) trong ảnh đích thông qua một hàm biến đổi:

$$(x', y') = T(x, y)$$

5.1.2. Vài trò của Image Warping trong ghép ảnh Panorama

Trong quá trình ghép ảnh panorama, mục tiêu là kết hợp nhiều ảnh chụp từ các góc nhìn khác nhau thành một ảnh toàn cảnh liền mạch. Tuy nhiên, do các ảnh được chụp ở vị trí và góc khác nhau, các điểm ảnh tương ứng sẽ không nằm đúng vị trí trên cùng một mặt phẳng.

Image Warping chính là bước biến đổi hình học để đưa các ảnh này về cùng một hệ tọa độ tham chiếu chung, sao cho các điểm tương ứng trùng khớp, tạo tiền đề cho việc ghép nối ảnh liền mạch.

5.1.3. Quá trình Image Warping trong panorama

a. Tính kích thước Canvas

Khi ghép nhiều ảnh lại, ảnh kết quả (panorama) sẽ có kích thước lớn hơn mỗi ảnh riêng lẻ. Canvas là khung ảnh tổng thể chứa toàn bộ các ảnh đã warp và ghép nối. Kích thước canvas phải đủ lớn để chứa toàn bộ ảnh, kể cả những vùng ảnh bị dịch chuyển hoặc biến dạng do warp.

Để đơn giản hóa tìm kích thước canvas của nhiều ảnh khác nhau thì sẽ quy về tìm canvas của 2 ảnh bất kỳ.

Ở ảnh thứ nhất, ảnh tham chiếu giữ nguyên hệ tọa độ gốc, kích thước W1*H1, các điểm của góc ảnh là:

$$P_1 = \{(0,0), (W_1,0), (W_1,H_1), (0,H_1)\}$$

Ở ảnh thứ hai, ảnh này sẽ được warp về hệ tọa độ ảnh tham chiếu bằng ma trận Homography H, lấy 4 điểm góc của ảnh thứ hai, ta có:

$$P_2 = \{(0,0), (W_2,0), (W_2,H_2), (0,H_2)\}$$

Áp dụng ma trận H lên các điểm góc này:

$$p_i' = H \cdot p_i, \quad p_i \in P_2$$

Tương tự, khi chuẩn các điểm góc từ ảnh nguồn sang ảnh đích cũng sẽ theo một tỉ lệ nhất định, lúc này sẽ chuẩn hóa tạo độ về:

$$(x_i', y_i') = \left(\frac{p_{i,x}'}{p_{i,w}'}, \frac{p_{i,y}'}{p_{i,w}'} \right)$$

Lúc này, tập hợp tất cả các điểm góc từ cả hai ảnh này sẽ là:

$$S = P_1 \cup \{p_i'\}$$

Khi đó sẽ tính được các điểm tương ứng sau:

$$x_{min} = \min_{(x,y)\in S} x, \quad x_{max} = \max_{(x,y)\in S} x$$

$$y_{min} = \min_{(x,y)\in S} y, \quad y_{max} = \max_{(x,y)\in S} y$$

Lúc này, kích thước canvas theo chiều rộng và chiều cao tương ứng sẽ là:

$$\text{canvas\_width} = \lceil x_{max} - x_{min} \rceil$$

$$\text{canvas\_height} = \lceil y_{max} - y_{min} \rceil$$

b. Tìm offset

Sau khi tính được kích thước canvas chứa đủ cả ảnh tham chiếu và ảnh thứ hai đã warp, bước tiếp theo là xác định offset để dịch chuyển tọa độ ảnh sao cho toàn bộ ảnh nằm gọn trong vùng tọa độ dương của canvas.

Lý do cần Offset là khi warp ảnh thứ hai, các điểm ảnh có thể có tọa độ âm, Canvas được định nghĩa trong không gian tọa độ bắt đầu từ (0,0) ở góc trên bên trái. Để tránh ảnh bị cắt hoặc nằm ngoài vùng hiển thị, cần dịch chuyển tất cả ảnh (ảnh tham chiếu và ảnh đã warp) sao cho điểm ảnh nhỏ nhất có tọa độ $x \geq 0, y \geq 0$.

Giả sử sau khi lấy tập hợp điểm góc của hai ảnh:

$$x_{min} = \min_{(x,y)} x, \quad y_{min} = \min_{(x,y)} y$$

Offset được tính bằng:

$$\text{offsetX} = \begin{cases} -x_{min}, & \text{nếu } x_{min} < 0 \\ 0, & \text{ngược lại} \end{cases}$$

$$\text{offsetY} = \begin{cases} -y_{min}, & \text{nếu } y_{min} < 0 \\ 0, & \text{ngược lại} \end{cases}$$

Ma trận dịch chuyển 2D dưới dạng Homogeneous:

$$T = \begin{bmatrix} 1 & 0 & \text{offsetX} \\ 0 & 1 & \text{offsetY} \\ 0 & 0 & 1 \end{bmatrix}$$

Homography ban đầu H ánh xạ từ ảnh thứ hai → ảnh thứ nhất (hệ tọa độ ảnh tham chiếu). Nhưng canvas mới đã dịch sang phải và xuống dưới một đoạn offset, nên Homography mới phải được điều chỉnh:

$$H_{\text{new}} = T \cdot H$$

c. Chiếu ảnh nguồn sang ảnh đích

Sau khi tìm được canvas, offset và ma trận homography H mới, tiến hành ánh xạ ảnh nguồn sang ảnh đích bằng phép chiếu

$$\begin{pmatrix} x' \\ y' \\ w \end{pmatrix} = H \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Vì khi chiếu ảnh từ ảnh nguồn sang ảnh đích theo một tỉ lệ nhất định, nên vị trí điểm ảnh mới trong ảnh đích sẽ được tình bằng:

$$(x', y') = \left(\frac{x'}{w}, \frac{y'}{w} \right)$$

d. Tạo mask cho hai arnh trên Canvas

Sau khi warp ảnh nguồn và đặt ảnh tham chiếu lên canvas, cần xác định vùng nào trong canvas thuộc:

- Chỉ ảnh A (ảnh tham chiếu)
- Chỉ ảnh B (ảnh warp)

- Cả hai ảnh (vùng overlap)

Mục đích của bước mask là chuẩn bị cho giai đoạn blending, giúp vùng chồng lấp giữa hai ảnh được hòa trộn mượt, tránh thấy đường nối rõ ràng.

Xác định mask của ảnh A (ảnh tham chiếu): Khi đặt ảnh A vào canvas, đánh dấu những pixel có dữ liệu (pixel khác 0), tạo một ảnh mask A cùng kích thước canvas:

$$\text{maskA}(x,y) = \begin{cases} 1, & \text{nếu pixel canvas(x,y) đến từ ảnh A} \\ 0, & \text{ngược lại} \end{cases}$$

Xác định mask của ảnh B (ảnh warp): Tương tự, khi warp ảnh B vào canvas bằng Hnew , đánh dấu vùng pixel mà ảnh B đóng góp.

$$\text{maskB}(x,y) = \begin{cases} 1, & \text{nếu pixel canvas(x,y) đến từ ảnh B} \\ 0, & \text{ngược lại} \end{cases}$$

Xác định vùng overlap: Vùng overlap được xác định bằng phép toán:

- Nếu overlap = 1 → điểm này được cả hai ảnh đóng góp → cần blending.
- Nếu overlap = 0 → lấy pixel từ ảnh A hoặc B tùy theo mask.

$$\text{overlap}(x,y) = \text{maskA}(x,y) \cdot \text{maskB}(x,y)$$

Như vậy, sau khi mask xong sẽ có bảng quy tắc chọn pixel như sau:

Vùng	Lấy pixel từ	Lý do
maskA = 1, maskB = 0	Ảnh A	Chỉ A có dữ liệu
maskA = 0, maskB = 1	Ảnh B	Chỉ B có dữ liệu
maskA = 1, maskB = 1	Blending A & B	Tránh đường nối, giữ sự mượt mà

Bảng 1. Bảng quy tắc chọn pixel

5.2. Image Blending.

5.2.1. Tổng quan

Sau khi đã xác định được vùng dữ liệu của từng ảnh và vùng chồng lấp (overlap), bước cuối cùng là blending để làm mượt phần biên giữa hai ảnh. Mục tiêu của blending là loại bỏ đường nối rõ rệt (visible seam), đảm bảo hai ảnh hòa trộn tự nhiên thành một panorama thống nhất.

Trong ghép ảnh panorama, có một kỹ thuật blending phổ biến là F**eathering (linear blending)** – đơn giản, nhanh, phù hợp cho hầu hết bài làm.Ngoài ra, còn một kỹ thuật ít phổ biến nhưng đơn giản đó là kỹ thuật Hard Cut

5.2.2. Hard Cut Blending

a. Tổng quan

Hard Cut (còn gọi là binary blending hoặc naive seam-cut) là kỹ thuật hòa trộn ảnh đơn giản nhất trong ghép ảnh panorama. Thay vì trộn giá trị pixel giữa hai ảnh, Hard Cut chỉ chọn pixel từ một ảnh duy nhất tại mỗi vị trí trong vùng chồng lặp (overlap).

Nói cách khác, ở vùng overlap, không hòa trộn màu mà cắt theo một đường biên cố định để quyết định pixel được lấy từ ảnh A hay ảnh B.

Hard Cut dùng một đường cắt đơn giản, thường là đường thẳng đứng tại giữa vùng overlap.

$$\text{Seam} = \frac{\min(X_{\text{overlap}}) + \max(X_{\text{overlap}})}{2}$$

Lúc này, quy tắc chọn pixel sẽ được quy định như sau:

- Nếu pixel không ở overlap: Lấy từ ảnh nào có dữ liệu
- Nếu pixel nằm bên trái đường seam: Pixel = A(x, y)
- Nếu pixel nằm bên phải đường seam: Pixel = B(x, y)

b. Ưu nhược điểm

Trong ghép ảnh panorama, kỹ thuật Hard Cut có một ưu điểm nổi bật là phương pháp này rất đơn giản và tốc độ xử lý nhanh. Do không cần thực hiện các phép tính phức tạp liên quan đến phân bố mức xám hay gradient như những kỹ thuật hòa trộn tiên tiến hơn, Hard Cut cho phép hệ thống lựa chọn pixel trực tiếp từ một trong hai ảnh tại vùng chồng lặp. Điều này giúp giảm thiểu thời gian tính toán, phù hợp với các hệ thống thời gian thực hoặc các mô hình thử nghiệm ban đầu.

Tuy nhiên, Hard Cut cũng tồn tại nhiều hạn chế khiến phương pháp này ít được sử dụng trong các hệ thống ghép ảnh chất lượng cao. Điểm yếu lớn nhất của Hard Cut là đường tiếp giáp giữa hai ảnh rất dễ bị lộ rõ nếu có sự khác biệt nhỏ về độ sáng, màu sắc hoặc phơi sáng giữa các ảnh đầu vào. Bên cạnh đó, khi có các vật thể lớn nằm trong vùng chồng lặp, kỹ thuật Hard Cut có thể "cắt đôi" vật thể này, dẫn đến hiện tượng méo hình hoặc gián đoạn thị giác. Chính vì những hạn chế đó, Hard Cut thường chỉ phù hợp trong các trường hợp cần tốc độ và không yêu cầu chất lượng hình ảnh cao, còn trong thực tế, người ta ưu tiên sử dụng các kỹ thuật blending nâng cao hơn để đạt được kết quả mượt mà và tự nhiên hơn.

5.2.3. Feathering (Linear Blending)

a. Tổng quan

Đây là phương pháp blending được dùng nhiều trong pipeline panorama cơ bản. Ý tưởng chính là tại vùng overlap, mỗi pixel được tính bằng tổ hợp tuyến tính của pixel từ ảnh A và B. Ảnh càng gần phía nào thì trọng số càng lớn về phía đó.

Công thức chung:

$$I(x,y) = w(x,y) \cdot A(x,y) + (1 - w(x,y)) \cdot B(x,y)$$

Trong đó:

- A(x,y): pixel từ ảnh A
- B(x,y): pixel từ ảnh B
- w(x,y)∈[0,1]: trọng số thay đổi theo vị trí trong vùng overlap

Khi này trong số w(x, y) sẽ được tính dựa trên khoảng cách vùng chống lần theo trục X từ x(start) đến x(end) như sau:

$$w(x) = \frac{x_{end} - x}{x_{end} - x_{start}}$$

Dựa trên công thức, ta có:

- Pixel bên trái overlap → w≈1 (ưu tiên ảnh A)
- Pixel bên phải overlap → w≈0 (ưu tiên ảnh B)
- Giữa overlap → gradually blending

b. Ưu nhược điểm

Phương pháp Feathering có ưu điểm rõ rệt ở sự đơn giản và dễ triển khai. Kỹ thuật này chỉ yêu cầu xây dựng một mặt nạ trọng số thay đổi tuyến tính trong vùng chồng lặp. Quá trình tính toán cũng diễn ra nhanh chóng do bản chất của Feathering chỉ là phép nội suy tuyến tính giữa hai pixel tương ứng. Khi hai ảnh đầu vào có điều kiện ánh sáng, độ sáng và màu sắc tương đối đồng nhất, Feathering mang lại hiệu quả hòa trộn tốt, giúp làm mờ dần ranh giới giữa hai ảnh và tạo nên sự chuyển tiếp mềm mại, tự nhiên.

Mặc dù vậy, Feathering vẫn tồn tại nhiều nhược điểm khiến kỹ thuật này không phải lúc nào cũng cho ra kết quả tối ưu. Trong trường hợp hai ảnh có sự khác biệt lớn về ánh sáng hoặc màu sắc, đường nối giữa chúng vẫn có khả năng bị lộ và tạo cảm giác không liền mạch, bởi thuật toán không thực hiện điều chỉnh ánh sáng hay cân bằng màu. Feathering cũng không xử lý hiệu quả các vùng phức tạp như bóng đổ, texture nhiều chi tiết hoặc chênh lệch phơi sáng, dẫn đến hiện tượng nhòe, loang màu hoặc mất chi tiết tại các khu vực này. Do đó, Feathering chỉ phù hợp với những bộ ảnh đầu vào có sự tương đồng cao, còn trong các tình huống thực tế với nhiều biến thiên ánh sáng, các phương pháp mạnh hơn như multi-band blending thường được ưu tiên sử dụng.

5.2.4. So sánh hai kỹ thuật

Kỹ thuật	Mực độ min	Tính toán	Chất lượng
Hard Cut	Thấp	Rất nhanh	Dễ lộ seam
Feathering	Trung bình	Nhánh	Seam mượt hơn

Bảng 2. Bảng so sánh hai kỹ thuật blending

5.3. Cropping (Cắt viền)

Sau khi hoàn thành bước blending để hòa trộn mượt mà các ảnh trong vùng chồng lấp, ảnh panorama thu được thường có kích thước lớn với nhiều vùng biên thừa, không chứa dữ liệu hình ảnh hợp lệ (thường là vùng đen hoặc méo). Do đó, bước cropping được thực hiện để tối ưu hóa khung hình cuối cùng và cải thiện chất lượng ảnh.

Cropping trong ghép ảnh panorama nhằm mục đích loại bỏ các vùng biên không cần thiết, bao gồm những phần canvas trống, thường là các pixel màu đen do quá trình warp ảnh không phủ kín toàn bộ canvas. Ngoài ra, nó còn giúp loại bỏ các vùng bị méo hoặc biến

dạng quá mức ở rìa ảnh, từ đó tạo ra một bức ảnh panorama có kích thước hình chữ nhật chuẩn, thuận tiện cho việc sử dụng và hiển thị.

Phương pháp thực hiện cropping thường bắt đầu bằng việc xác định vùng chứa dữ liệu hợp lệ trong ảnh panorama, tức là những vùng pixel có giá trị màu khác với màu nền. Từ vùng dữ liệu này, ta sẽ tính toán bounding box lớn nhất hoặc hình chữ nhật lớn nhất có thể chèn vừa vào vùng hợp lệ đó. Sau khi xác định được vùng phù hợp, ảnh sẽ được cắt theo vùng này để loại bỏ các phần thừa, giúp ảnh trở nên gọn gàng và dễ nhìn hơn.

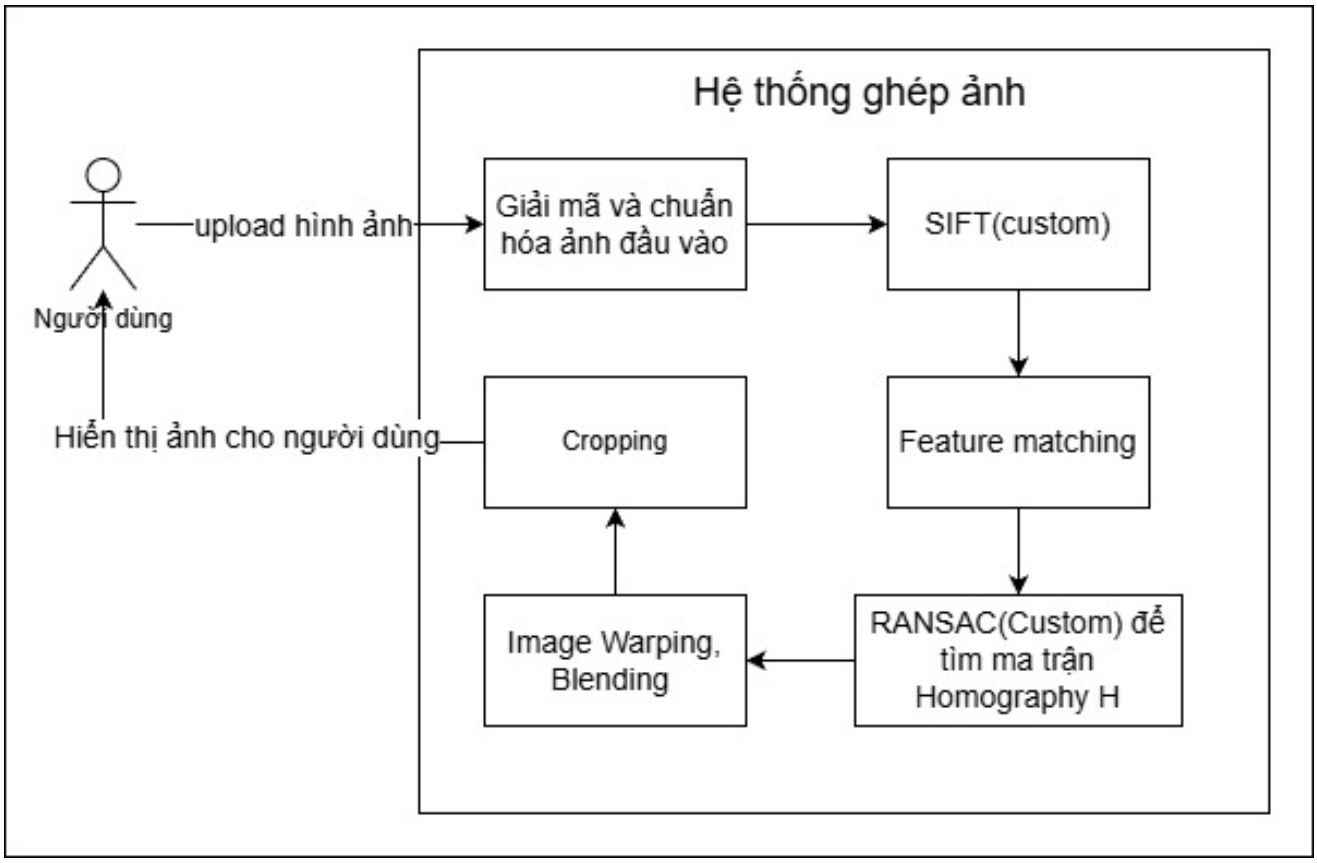
Tuy nhiên, cần lưu ý rằng nếu quá trình cropping được thực hiện quá gắt, có thể làm mất một phần cảnh hoặc vật thể quan trọng nằm ở mép ảnh, ảnh hưởng đến nội dung panorama. Ngược lại, nếu không thực hiện cropping, ảnh sẽ giữ lại vùng biên thừa, gây mất thẩm mỹ và khó khăn khi hiển thị hoặc xử lý sau này. Vì vậy, cropping thường được xem là một bước bắt buộc trong quy trình xử lý panorama để đảm bảo ảnh đầu ra vừa đẹp vừa dễ sử dụng.

B. Triển khai và kết quả

VI. Triển khai

Hệ thống được triển khai nhằm xây dựng một pipeline ghép ảnh panorama hoàn toàn tự cài đặt, không dựa vào các hàm có sẵn như SIFT, findHomography hay warpPerspective của thư viện OpenCV. Chuỗi xử lý gồm các bước chính: phát hiện và mô tả điểm đặc trưng bằng SIFT tự cài, ghép đặc trưng sử dụng thuật toán KNN kèm theo lọc Lowe's ratio, ước lượng ma trận Homography thông qua phương pháp DLT với chuẩn hóa Hartley kết hợp RANSAC để loại bỏ ngoại lai, biến đổi phối cảnh ảnh bằng ánh xạ ngược (backward mapping) với nội suy song tuyến tính, và cuối cùng là hòa trộn vùng chồng lấp theo gradient alpha cùng bước cắt viền đen để tạo ra ảnh panorama hoàn chỉnh.

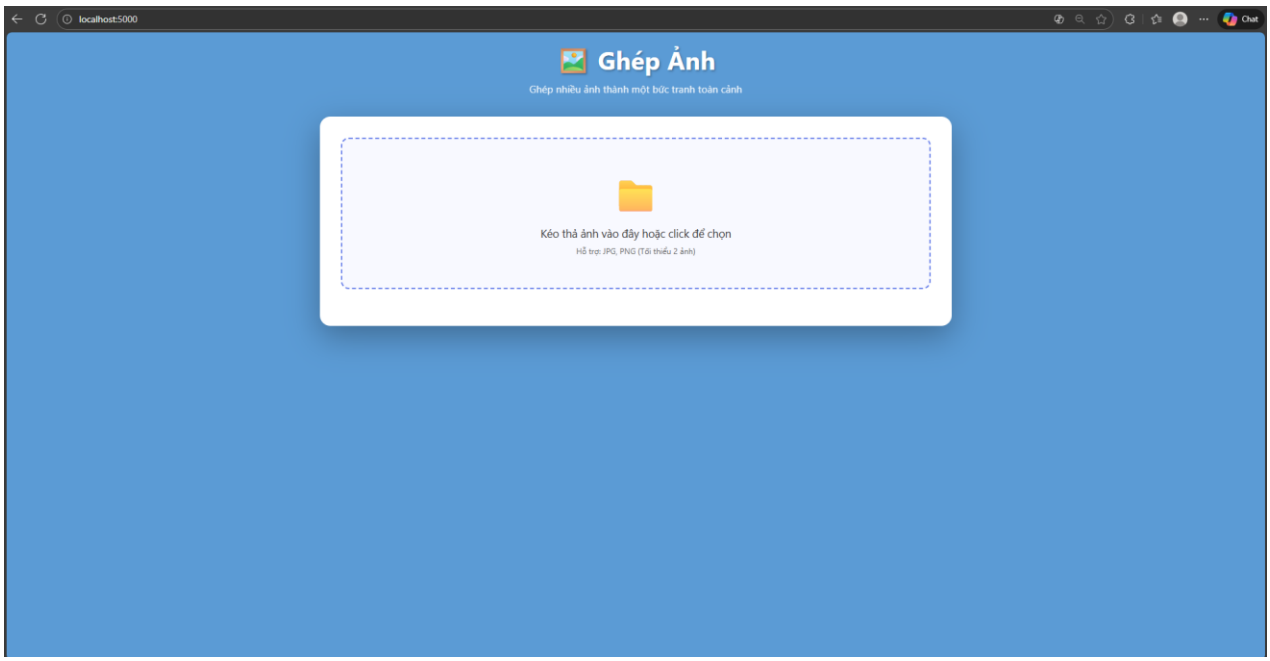
Sơ đồ tổng quan hệ thống



Hình 16 Hình ảnh sơ đồ tổng quan hệ thống ghép ảnh

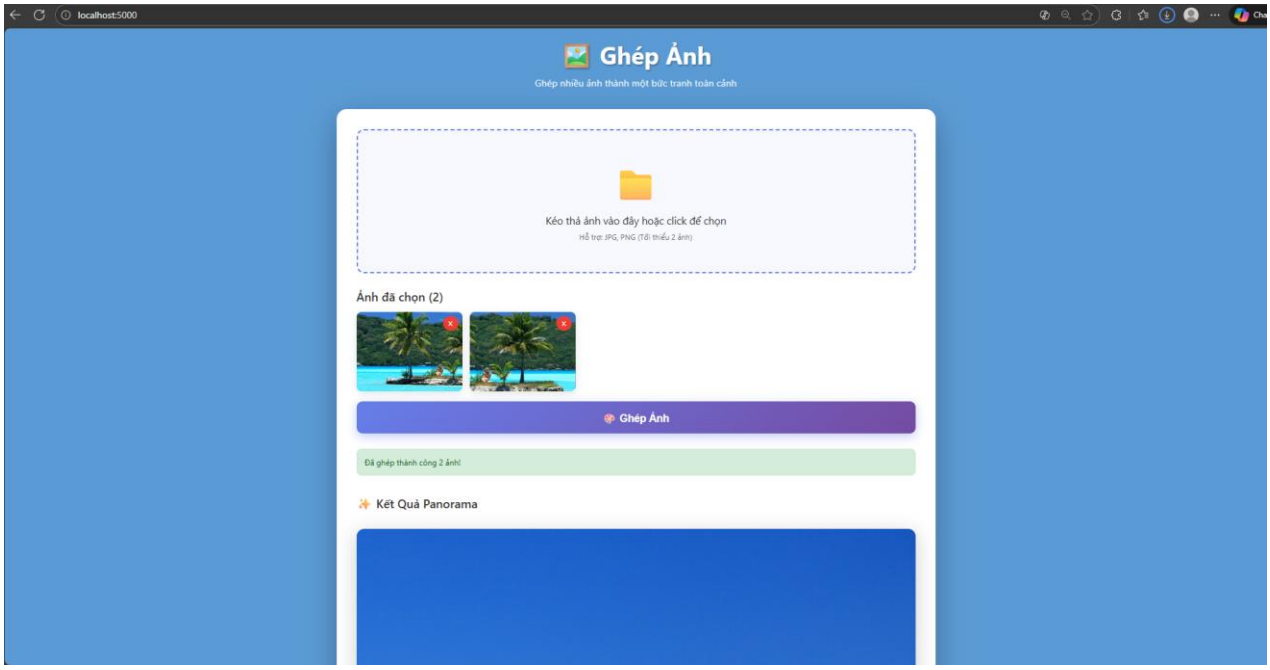
VII. Kết quả triển khai

Trang web ghép ảnh panorama là một ứng dụng cho phép người dùng tải lên nhiều ảnh chụp từ các góc khác nhau và tự động xử lý để tạo ra một bức ảnh panorama liền mạch, rộng hơn. Hệ thống sử dụng các thuật toán xử lý ảnh tiên tiến để phát hiện điểm đặc trưng, ghép nối ảnh dựa trên các điểm tương đồng, ước lượng ma trận biến đổi phối cảnh, và cuối cùng là hòa trộn vùng chồng lấp để tạo ra ảnh toàn cảnh mượt mà, không có đường nối rõ ràng.

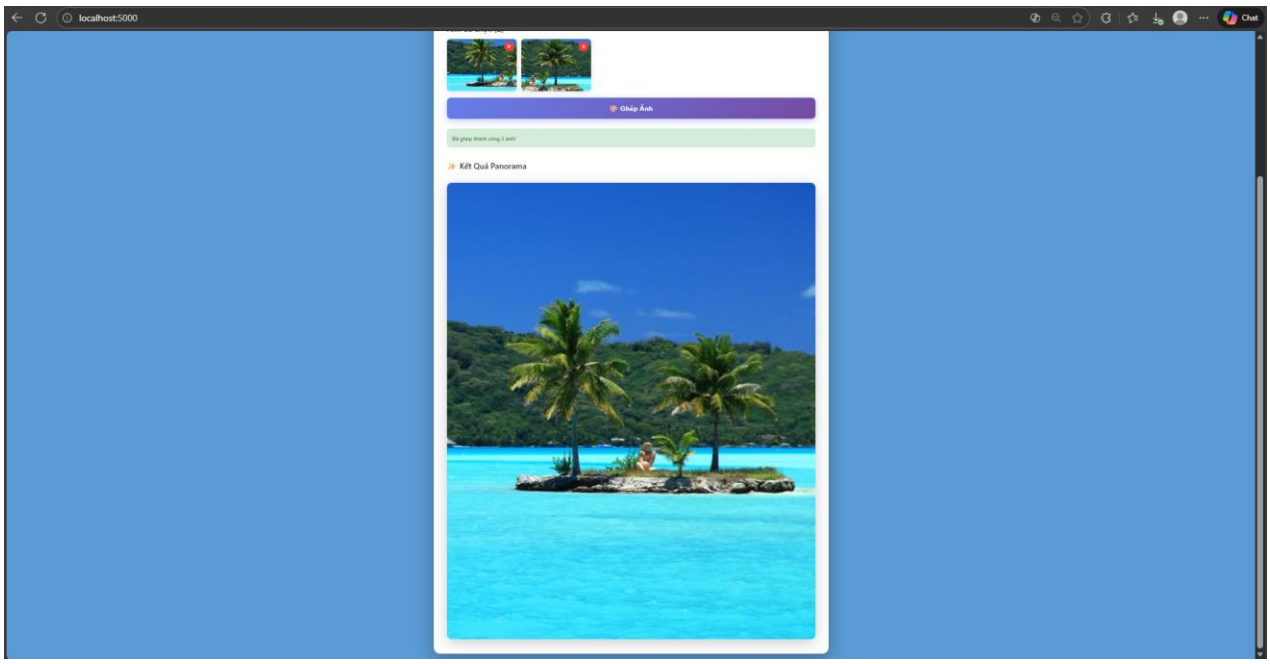


Hình 17 Hình ảnh trang web ghép ảnh

Ghép 2 ảnh bất kỳ:

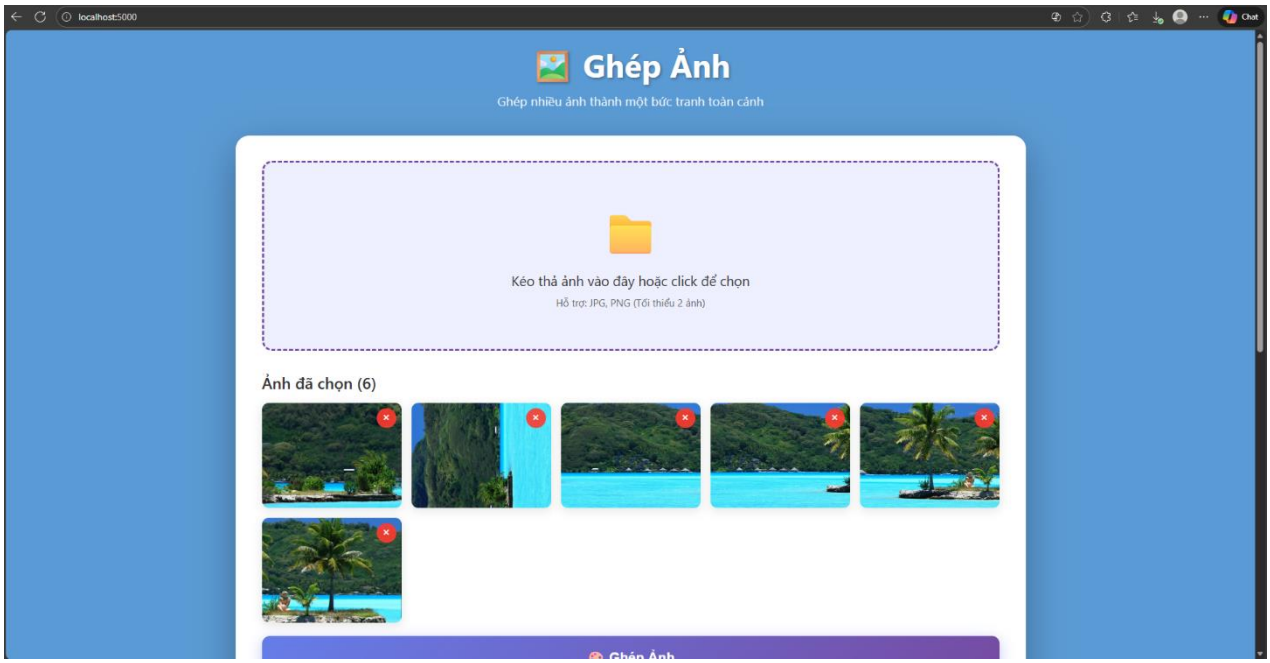


Hình 18 Hình ảnh trang web ghép hai ảnh bất kỳ

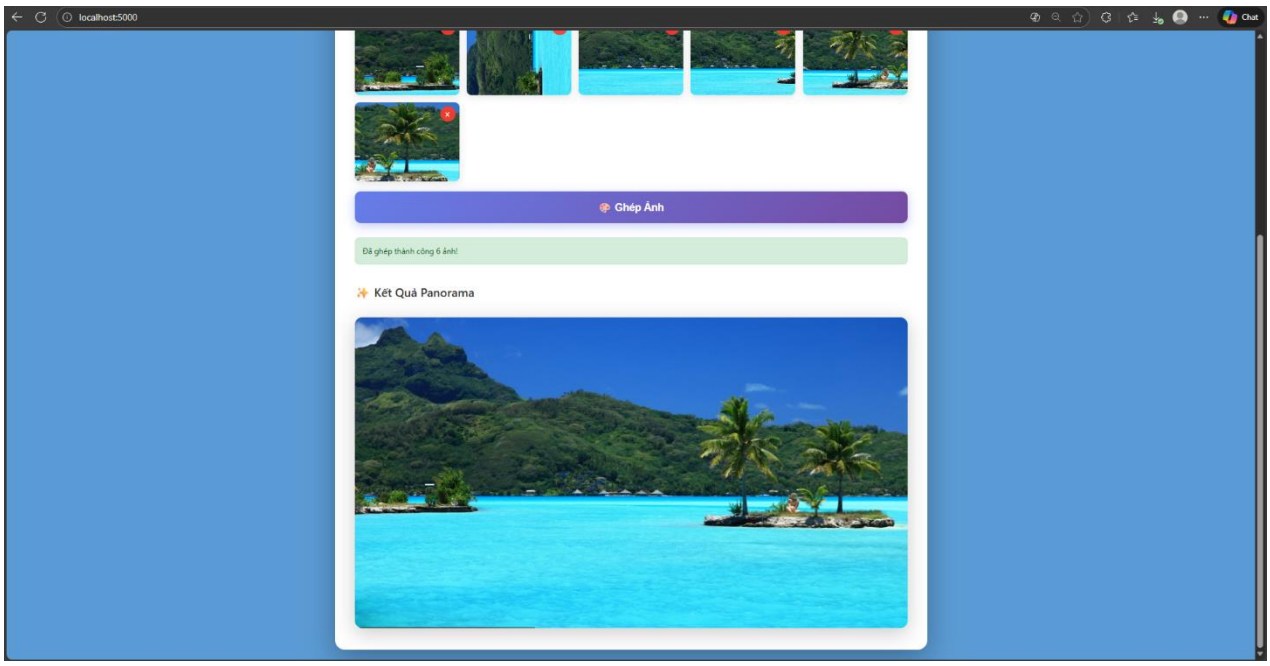


Hình 19 Hình ảnh trang web hiển thị kết quả sau khi ghép hai ảnh bất kỳ

Ghép nhiều(6) ảnh cùng lúc:

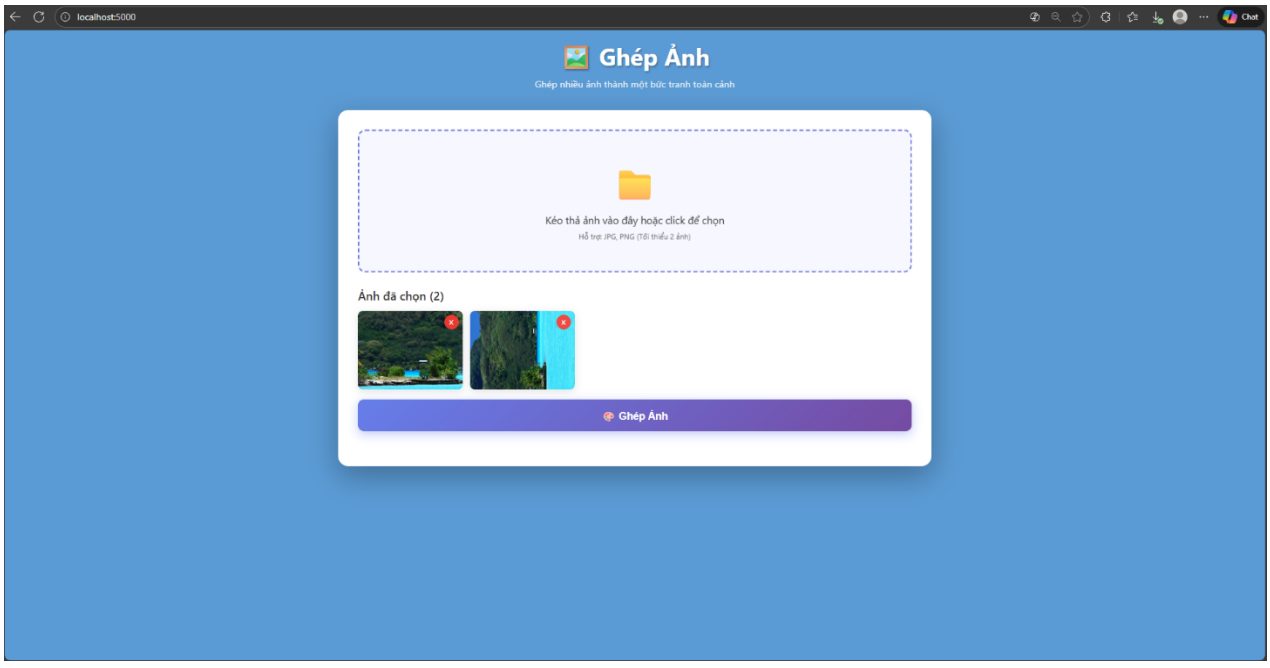


Hình 20 Hình ảnh trang web sau khi tải 6 hình ảnh lên

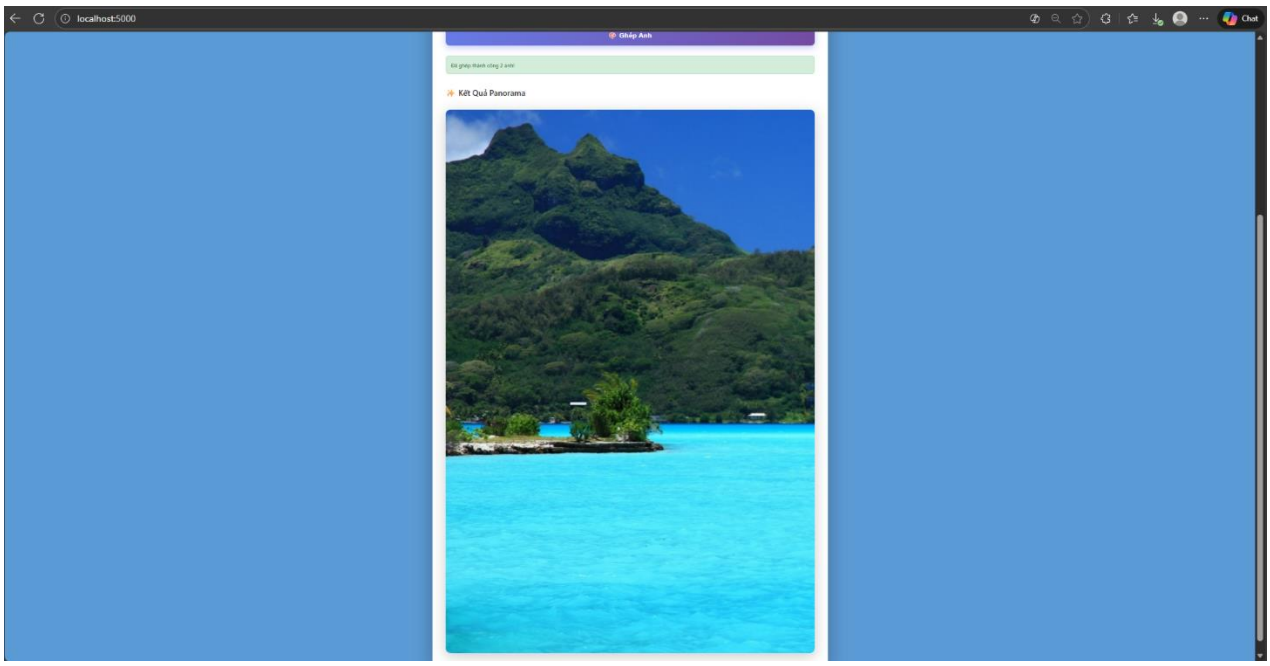


Hình 21 Hình ảnh trang web hiển thị kết quả sau khi ghép 6 hình ảnh cùng lúc

Ghép ảnh trong đó có 1 ảnh thẳng đúng, một ảnh nằm ngang:



Hình 22 Hình ảnh trang web sau khi tải một ảnh dọc, một ảnh ngang



Hình 23 Hình ảnh trang web hiển thị kết quả sau khi ghép một hình ảnh dọc một hình ảnh ngang

C. HƯỚNG PHÁT TRIỂN TIẾP THEO

Cải thiện chất lượng ghép ảnh có thể đạt được bằng việc áp dụng các thuật toán blend tiên tiến như multi-band blending, giúp giảm thiểu hiện tượng đường nối (seam) và sự lệch màu giữa các ảnh ghép. Phương pháp này mang lại kết quả liền mạch, tự nhiên hơn, đồng thời nâng cao trải nghiệm thị giác cho người dùng.

Về mặt hiệu năng, tối ưu tốc độ xử lý là yếu tố then chốt. Việc sử dụng các thuật toán tìm kiếm đặc trưng nhanh hơn hoặc tận dụng xử lý song song (parallel processing) sẽ giúp ghép ảnh theo thời gian thực, đáp ứng nhu cầu xử lý nhanh chóng trong các ứng dụng hiện đại. Ngoài ra, mở rộng sang ghép ảnh cho video hoặc panorama 360° sẽ phục vụ hiệu quả cho các lĩnh vực thực tế ảo (VR) và thực tế tăng cường (AR), mang đến trải nghiệm đa chiều và sống động hơn.

Đồng thời, việc ứng dụng học sâu (Deep Learning) với các mạng CNN hoặc transformer trong phát hiện và ghép đặc trưng góp phần nâng cao độ chính xác, đặc biệt trong những trường hợp ảnh có nhiều vùng đồng nhất hoặc thay đổi ánh sáng phức tạp. Tích hợp những công nghệ này vào các ứng dụng thực tiễn trên điện thoại hoặc web sẽ giúp người dùng dễ dàng tạo panorama tự động từ bộ sưu tập ảnh cá nhân, mang lại trải nghiệm tiện lợi và thân thiện.

D. TÀI LIỆU THAM KHẢO

[1] https://www.researchgate.net/figure/Difference-of-Gaussian-is-obtained-as-the-difference-of-Gaussian-blurring-of-an-image_fig4_312107986

[2] https://github.com/reinbugnot/keypoint-detection-sift

https://medium.com/@navekshasood/image-stitching-to-create-a-panorama-5e030ecc8f7

[3] Đỗ Năng Toàn, Nguyễn Tất Thắng, Đào Thị Thuý Quỳnh, Bài giảng Xử lý ảnh, Học viện Công nghệ Bưu chính viễn thông, 2023.

[4] https://docs.opencv.org/4.x/d9/dab/tutorial_homography.html

[5] https://www.researchgate.net/figure/Difference-of-Gaussian-is-obtained-as-the-difference-of-Gaussian-blurring-of-an-image_fig4_312107986

[6] https://docs.opencv.org/4.x/dc/dc3/tutorial_py_matcher.html