
MỤC LỤC

LỜI CẢM ƠN	3
LỜI CAM ĐOAN	4
DANH MỤC THUẬT NGỮ VÀ CÁC TỪ VIẾT TẮT	5
DANH MỤC HÌNH ẢNH	6
DANH MỤC BẢNG BIỂU	8
MỞ ĐẦU	9
CHƯƠNG 1: TỔNG QUAN VỀ TRÍCH CHỌN ĐẶC TRƯNG ẢNH SỐ	11
1.1 Các nội dung cơ bản.....	11
1.1.1 Khái niệm đặc trưng ảnh số	11
1.1.2 Tại sao phải trích chọn đặc trưng ảnh số ?	11
1.2 Trích chọn đặc trưng dựa trên điểm nổi bật.....	11
CHƯƠNG 2: CÁC PHƯƠNG PHÁP TRÍCH CHỌN ĐẶC TRƯNG ĐIỂM NỔI BẬT	13
2.1 Thuật toán tìm kiếm góc Harris.....	13
2.1.1 Giới thiệu và ý tưởng.....	13
2.1.2 Thuật toán	14
2.2 Trích chọn đặc trưng cục bộ bất biến SIFT	17
2.2.1 Giới thiệu và định nghĩa.....	17
2.2.2 Thuật toán	18
2.3 Trích chọn đặc trưng SURF	26
2.3.1 Giới thiệu và định nghĩa.....	26
2.3.2 Thuật toán	26
CHƯƠNG 3: ỨNG DỤNG VÀO LĨNH VỰC TẠO ẢNH PANORAMA	33
3.1 Giới thiệu	33
3.2 Quy trình thực hiện.....	34
3.3 So sánh đối chiếu các điểm tương đồng nổi bật.....	35
3.4.1 Cơ sở ban đầu	35
3.4.2 Thực hiện trên nền C++ và OpenCV2.....	36
3.4 Tính ma trận Homography bằng thuật toán RANSAC	39
3.4.1 Vài nét về Homography	39
3.4.2 Tính toán Homography	40
3.4.3 Thực hiện trên nền C++ và OpenCV2.....	44
3.5 Ghép nối ảnh	44
THỰC NGHIỆM TRÊN NỀN TẢNG C++ VÀ OPENCV2	46
4.1 Một vài thông tin ban đầu	46

4.2	Giao diện chương trình	46
4.3	Một số kết quả có được	48
4.3.1.	<i>Tìm kiếm góc Harris, trích chọn đặc trưng SIFT và SURF</i>	48
4.3.2.	<i>So sánh đối chiếu các điểm đặc trưng tương đồng SIFT và SURF</i>	49
4.3.3.	<i>Tạo ảnh Panorama</i>	51
KẾT LUẬN		54
TÀI LIỆU THAM KHẢO		55

LỜI CẢM ƠN

Trước hết, tôi xin chân thành cảm ơn quý thầy cô Trường Đại Học Dân Lập Duy Tân, cũng như Khoa Điện-Điện Tử đã tận tình dạy bảo, truyền đạt kiến thức cho tôi trong suốt quá trình học tập. Đặc biệt là Ths. Nguyễn Lê Mai Duyên, người đã trực tiếp theo sát và chỉ dẫn cho tôi trong quá trình tìm hiểu và thực hiện đề tài này.

Xin được gửi lời cảm ơn đến Ban giám hiệu nhà trường, Khoa Điện – Điện tử đã tạo điều kiện cho tôi được học tập và trau dồi kiến thức trong suốt thời gian qua.

Cũng xin cảm ơn các bạn cùng lớp đã có ý kiến đóng góp, cũng như giúp đỡ tôi trong suốt thời gian học tập vừa qua.

Một lần nữa, xin được gửi lời cảm ơn đến tất cả!

Đà Nẵng, ngày 30 tháng 11 năm 2014

Sinh viên

Lê Đức Thịnh

LỜI CAM ĐOAN

Tôi xin cam đoan kết quả đạt được trong đề tài là sản phẩm của riêng cá nhân tôi, không sao chép lại của người khác. Trong toàn bộ nội dung của đề tài, những điều được trình bày hoặc là của cá nhân hoặc là được tổng hợp từ nhiều nguồn tài liệu. Tất cả các tài liệu tham khảo đều có xuất xứ rõ ràng và được trích dẫn hợp pháp.

Tôi xin hoàn toàn chịu trách nhiệm về lời cam đoan của mình.

Đà Nẵng, ngày 30 tháng 11 năm 2014

Sinh viên

Lê Đắc Thịnh

DANH MỤC THUẬT NGỮ VÀ CÁC TỪ VIẾT TẮT

Thuật ngữ, kí hiệu	Nghĩa
SIFT	Scale-Invariant Feature Transform
SURF	Speed Up Robust Feature
DoG	Difference of Gaussian
RANSAC	Random Sample Consensus
SVD	Singular Value Decomposition
DLT	Direct Linear Transform

DANH MỤC HÌNH ẢNH

Hình 2.1: Các điểm nổi bật trong ảnh

Hình 2.2: Cửa sổ trượt phát hiện góc Harris

Hình 2.3: Minh họa các trường hợp λ_1 và λ_2

Hình 2.4: Kết quả tìm kiếm góc Harris trên nền C++ và OpenCV2

Hình 2.5: Đỉnh tòa nhà VTV Đà Nẵng và những góc Harris

Hình 2.6: Mô tả sự không hoàn thiện của thuật toán Harris

Hình 2.7: Quá trình tính không gian đo (L) và hàm sai khác D

Hình 2.8: Quá trình tìm điểm cực trị trong các hàm sai khác DoG

Hình 2.9: Minh họa các bước của quá trình lựa chọn các điểm nổi bật

Hình 2.10: Mô tả tạo bộ mô tả cục bộ

Hình 2.11: Kết quả tìm kiếm đặc trưng SIFT trên nền C++ và OpenCV2

Hình 2.12: Đỉnh tòa nhà VTV Đà Nẵng và những điểm đặc trưng SIFT(1)

Hình 2.13: Đỉnh tòa nhà VTV Đà Nẵng và những điểm đặc trưng SIFT(2)

Hình 2.14: Xấp xỉ đạo hàm cấp 2 hàm Gaussian bằng hộp lọc

Hình 2.15: Các điểm quan tâm được phát hiện

Hình 2.16: Lọc Haar wavelet để tính sự ảnh hưởng trên hai hướng x và y.

Hình 2.17: Vùng hình tròn xung quanh và hướng đại diện cho điểm đặc trưng

Hình 2.18: 4x4 hình vuông con xung quanh điểm đặc trưng

Hình 2.19: Ví dụ về đốm sáng trên nền tối hơn và đốm tối trên nền sáng hơn

Hình 2.20: Kết quả tìm kiếm đặc trưng SURF trên nền C++ và OpenCV2

Hình 2.21: Đỉnh tòa nhà VTV Đà Nẵng và những điểm đặc trưng SURF(1)

Hình 2.22: Đỉnh tòa nhà VTV Đà Nẵng và những điểm đặc trưng SURF (2)

Hình 3.1: Một ảnh Panorama

Hình 3.2: Minh họa các bước thực hiện thuật toán ghép nối ảnh\

Hình 3.3: Hai hình ảnh được sử dụng so sánh đối chiếu

Hình 3.4: Kết quả sau khi lấy đặc trưng SIFT

Hình 3.5: Kết quả sau khi so sánh đối chiếu các điểm đặc trưng SIFT

Hình 3.6: Kết quả sau khi lấy đặc trưng SURF

Hình 3.7: Kết quả sau khi so sánh đối chiếu các điểm đặc trưng SIFT

Hình 3.8: Kết quả sau khi so sánh đối chiếu các điểm đặc trưng SIFT đối với hình Hình 2.11 và Hình 2.12 ở mục 2.2.3

Hình 3.9: Kết quả sau khi so sánh đối chiếu các điểm đặc trưng SIFT đối với hình Hình 2.11 và Hình 2.12 ở mục 2.2.3

Hình 3.10: Phép chiếu Homography

Hình 3.11: Phép tính Homography từ điểm đặc trưng SIFT

Hình 3.12: Phép tính Homography từ điểm đặc trưng SURF

Hình 3.13: Minh họa ghép nối ảnh

Hình 3.14: Kết quả cuối cùng của phép tạo ảnh Panorama

Hình 4.1: Giao diện chương trình thực nghiệm

Hình 4.2: Hình ảnh đuôi cầu Rồng, Tp Đà Nẵng

Hình 4.3: Hình ảnh cầu Rồng được sử dụng trong thực nghiệm

Hình 4.4: Kết quả so sánh đối chiếu từ điểm đặc trưng SIFT

Hình 4.5: Kết quả so sánh đối chiếu từ điểm đặc trưng SURF

Hình 4.6: Kết quả tính toán ma trận Homography

Hình 4.7: Kết quả của phép tạo ảnh Panorama

Hình 4.8: Hình ảnh Panorama khi sử dụng hàm chức năng của OpenCV2

Hình 4.9: Một kết quả hoàn hảo hơn

DANH MỤC BẢNG BIỂU

Bảng 4.1: So sánh kết quả trích chọn điểm nổi bật

Bảng 4.2: So sánh kết quả so sánh đối chiếu điểm đặc trưng tương đồng

MỞ ĐẦU

Ngày nay, cùng với sự phát triển vượt bậc của xã hội cũng như các lĩnh vực khoa học kỹ thuật, nhu cầu xử lý hình ảnh trên máy tính hay rộng hơn là các nhu cầu có liên quan đến thị giác máy (Computer Vision) lại càng trở nên cấp thiết hơn bao giờ hết.

Nhưng dù nhận được sự hỗ trợ mạnh mẽ từ sự đổi mới rất nhanh chóng của các nền tảng công nghệ mới, việc xử lý lượng thông tin chứa đựng trong hình ảnh nói chung thực sự là một bài toán không hề dễ dàng, nhất là trong bối cảnh các yêu cầu đặt ra cần được thực hiện một cách nhanh chóng và chính xác nhất.

Một trong số những vấn đề rất được chú trọng trong lĩnh vực này chính là: Làm thế nào để nhận biết ảnh số và các vật thể có trong ảnh?

Thực tế, khác với đôi mắt con người, máy tính không làm việc với hình ảnh một cách tổng quát và trực quan, mà trái lại, chúng chỉ có thể tiếp cận hình ảnh và những thông tin chứa trong hình ảnh đó một cách rất hạn chế dưới dạng các ma trận điểm ảnh.

Vì vậy, để thực hiện được nhiệm vụ “*Nhận biết ảnh số và các vật thể có trong ảnh*”, máy tính và các thiết bị kỹ thuật số sẽ phải đi tìm các điểm đặc trưng của ảnh để rồi từ đó đưa ra các biện pháp so sánh, đối chiếu hay biến đổi nhằm thực hiện các yêu cầu được đặt ra.

Trong phạm vi đề tài này, tôi xin được trình bày những nội dung liên quan đến khái niệm “*Trích chọn đặc trưng ảnh số*” và đi sâu hơn là khái niệm “*Trích chọn đặc trưng ảnh số dựa trên điểm nổi bật*” nhằm đưa ra câu trả lời cho những vấn đề ở trên cũng như đề xuất các ứng dụng mang tính thực tiễn có thể phát triển trong tương lai.

Trên cơ sở đó, đề tài cũng sẽ đi vào triển khai ứng dụng “*Tạo ảnh Panorama*” và những vấn đề xung quanh nó để làm rõ thêm vấn đề đã trình bày thông qua ngôn ngữ C++, thư viện mã nguồn mở OpenCV2 và công cụ lập trình Visual Studio 2012.

Cấu trúc của đề tài gồm các phần sau:

Chương 1 trình bày các khái niệm cơ bản về “*Trích chọn đặc trưng ảnh số*” và giới thiệu phương pháp “*Trích chọn đặc trưng ảnh số dựa trên điểm nổi bật*”.

Chương 2 giới thiệu các thuật toán “*Tìm kiếm góc Harris*”, phép “*Trích chọn đặc trưng cục bộ bất biến SIFT*”, phép “*Trích chọn đặc trưng SURF*”.

Chương 3 giới thiệu về lĩnh vực “*Tạo ảnh Panorama*”, phương pháp tạo ảnh dựa trên phương pháp tìm điểm nổi bật đã trình bày ở **Chương 2**, các vấn đề xung quanh như ma trận Homography và cách tính.

Chương 4 bao gồm các hình ảnh về chương trình ứng dụng nhằm đưa ra các kết quả về mặt hình ảnh và con số về các vấn đề đã được trình bày ở các chương trên.

Kết luận nêu ra những kết quả đạt được của đề tài, các mặt hạn chế và hướng phát triển trong tương lai.

CHƯƠNG 1: TỔNG QUAN VỀ TRÍCH CHỌN ĐẶC TRƯNG ẢNH SỐ

1.1 Các nội dung cơ bản

1.1.1 *Khái niệm đặc trưng ảnh số*

Trong phạm vi xử lý ảnh, đặc trưng ảnh số là một phần thông tin của ảnh số thích hợp cho các nhiệm vụ tính toán liên quan đến một ứng dụng nhất định. Những đặc trưng đó có thể là các kết cấu đặc biệt trong ảnh số như các điểm, các cạnh của một đối tượng hoặc một đối tượng nào đó có trong ảnh. Mặt khác, các đặc trưng của ảnh số cũng có thể là kết quả của một phép biến đổi toàn diện hoặc là các phương pháp phát hiện điểm đặc trưng được áp dụng trên toàn bộ ảnh đó[25].

Trích chọn đặc trưng ảnh số là quá trình xử lý làm cho ảnh ban đầu được biến đổi thành các dạng mà máy tính có thể dễ dàng nhận dạng hơn.

1.1.2 *Tại sao phải trích chọn đặc trưng ảnh số ?*

Không giống như thị giác con người, thị giác máy có khả năng nhìn nhận rất hạn chế bởi đối với máy tính, mỗi hình ảnh chỉ là một ma trận các điểm ảnh. Vì vậy, đối với các lĩnh vực có liên quan đến xử lý các lượng lớn hình ảnh, bài toán trích chọn đặc trưng ảnh là rất quan trọng đối với công tác nhận dạng, phân loại dữ liệu hình ảnh hoặc gán gũ hơn là tìm kiếm dữ liệu trong một lượng lớn dữ liệu hình ảnh có sẵn.

Mặt khác, việc trích chọn đặc trưng ảnh số cũng là một công cụ rất có ích trong việc xác định các phần tương đồng hoặc giống nhau của các ảnh số, từ đó có thể phát triển các ứng dụng ghép nối hình ảnh, dựng ảnh 3D hay so sánh một cách dễ dàng và chính xác hơn.

Tóm lại, trích chọn đặc trưng ảnh số là bước xử lý rất căn bản trong xử lý ảnh. Nó không chỉ là giúp máy tính và các thiết bị tiếp xúc với dữ liệu của ảnh một cách dễ dàng hơn mà còn là nền tảng cho những ứng dụng cao cấp trong lĩnh vực này.

1.2 Trích chọn đặc trưng dựa trên điểm nổi bật

Ý tưởng của phương pháp này là thay vì phân tích toàn bộ hình ảnh, ta chỉ cần tập trung vào một vài điểm nhất định trên ảnh đó và thực hiện các phép phân tích cục bộ. Những phương pháp này sẽ hoạt động tốt nếu như ở trên ảnh tồn tại một số lượng vừa đủ các điểm nổi bật bất biến và ổn định để có thể thực hiện việc phân tích cục bộ một cách chính xác[27].

Trong lĩnh vực xử lý ảnh, một điểm được gọi là điểm nổi bật thường được mô tả như sau:

- Rõ ràng về mặt định nghĩa (thường dựa trên cơ sở toán học)
- Có vị trí xác định trong không gian của ảnh

-
- Cấu trúc cục bộ xung quanh điểm nổi bật thường phong phú về thông tin để việc sử dụng điểm nổi bật này giúp đơn giản hóa các bước xử lý cao hơn trong hệ thống thị giác
 - Các điểm này có tính ổn định trước các sự thay đổi cục bộ và toàn diện trên ảnh

Tóm lại, điểm nổi bật của ảnh số được định nghĩa như sau:

Điểm nổi bật trong ảnh là điểm ảnh có chứa nhiều thông tin hơn các điểm ảnh lân cận. Biểu diễn ảnh theo điểm nổi bật sẽ cô đọng hơn, giảm được không gian tìm kiếm trong các bài toán ứng dụng[27].

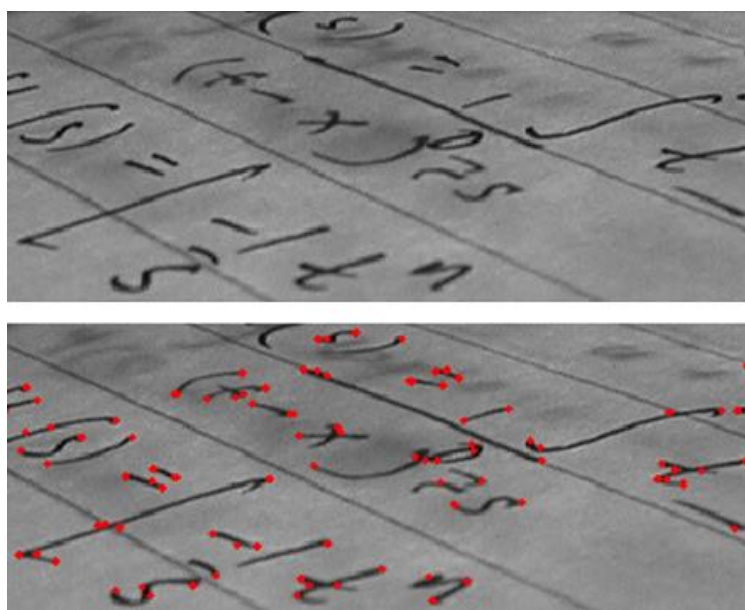
Các điểm nổi bật của ảnh được phân thành một số dạng như: đường biên, đường thẳng, viền tròn và các điểm góc [27].

Đường biên là chuỗi các vị trí mà tại đó có sự thay đổi mạnh mẽ về cường độ điểm ảnh. Đường biên đặc trưng không nhất thiết phải trùng với đường biên vật lý.

Đường thẳng là dải hẹp dọc theo đối tượng và có cùng cường độ điểm ảnh. Có thể coi đường thẳng này là hai đường biên song song.

Viền tròn là các đối tượng đặc có cường độ các điểm ảnh xấp xỉ nhau.

Điểm góc là vị trí gấp khúc của đường biên. Trong một cảnh, điểm góc có thể hiểu là điểm cuối của đường biên.



Hình 2.1: Các điểm nổi bật trong ảnh[25]

Trong chương tiếp theo, luận văn sẽ đi vào tìm hiểu phương pháp trích chọn đặc trưng từ các điểm nổi bật lần lượt theo phương pháp: Thuật toán tìm kiếm góc Harris, Trích chọn đặc trưng cục bộ bất biến SIFT và Trích chọn đặc trưng SURF.

CHƯƠNG 2: CÁC PHƯƠNG PHÁP TRÍCH CHỌN ĐẶC TRƯNG ĐIỂM NỔI BẬT

2.1 Thuật toán tìm kiếm góc Harris

2.1.1 Giới thiệu và ý tưởng

Một trong những phương pháp tìm kiếm điểm nổi bật cơ bản nhất chính là phương pháp tìm kiếm các điểm nổi của đường biên, các điểm nằm trên đường cong mà có độ cong tối đa... hay còn gọi là các điểm góc.

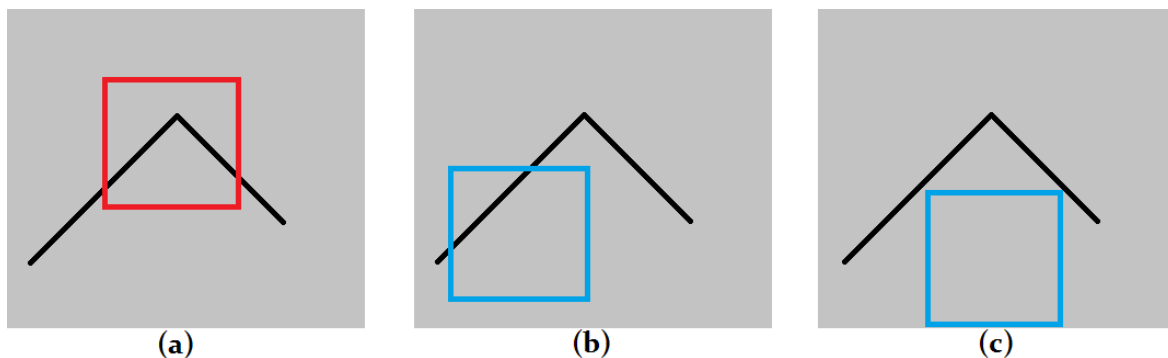
Ban đầu các điểm này được phát hiện nhờ vào “độ nhọn”(sharp) của đường biên: biên của đối tượng được lưu dưới dạng mã xích, góc được phát hiện thông qua việc tìm kiếm những vị trí trên biên bị “uốn” một cách đáng kể. Kỹ thuật phát hiện góc này rất phức tạp và phải triển khai trên nhiều bước.

Sau đó, L.Kitchen[1] đã đề xuất một phương pháp nhằm cô lập các điểm góc dựa vào một số phép toán trên Gradient, phương pháp này tỏ ra ít phức tạp hơn nhưng vẫn chưa đủ hiệu quả.

Tiếp đến, Moravec[2][3] đề xuất hàm phát hiện góc dựa trên việc dùng một cửa sổ và xác định sự thay đổi của trung bình cường độ xám khi dịch chuyển cửa sổ này theo bốn hướng. Nếu cường độ xám chỉ thay đổi theo hai hướng thì ta xác định đó là biên, còn nếu cường độ xám thay đổi theo bất cứ hướng nào thì ta xác định được đó là góc. Tuy nhiên, phương pháp này có một nhược điểm là: cửa sổ trượt là hàm nhị phân, chỉ trượt được theo số hướng nhất định (bốn hướng).

Cuối cùng, Harris và Stephens[4] cải tiến phương pháp của Moravec để tạo ra thuật toán tìm kiếm góc Harris. Nếu như Moravec chỉ sử dụng một cửa sổ để xác định sự thay đổi trung bình cường độ xám bằng cách dịch chuyển cửa sổ này theo bốn hướng[5] thì Harris và Stephens đã làm cho cửa sổ này có thể trượt theo bất kỳ hướng nào bằng cách sử dụng hàm Gaussian và phép khai triển Taylor[4].

Về mặt ý tưởng, thuật toán Harris sẽ tìm kiếm sự thay đổi lớn về cường độ xám theo các hướng khác nhau bằng cách dùng một cửa sổ nhỏ để làm nhiệm vụ rà soát và phát hiện những điểm được định nghĩa là “góc” như minh họa dưới hình sau:



Hình 2.2: Cửa sổ trượt phát hiện góc Harris

Trong Hình 2.2(a), cửa sổ trượt nằm trong vùng hình ảnh có chứa góc, khi ta di chuyển theo bất cứ hướng nào đều có sự thay đổi về cường độ xám.

Trong Hình 2.2(b), cửa sổ trượt nằm trên vùng hình ảnh có chứa cạnh, khi ta di chuyển cửa sổ trượt theo hai hướng của cạnh sẽ không sự thay đổi về cường độ xám.

Trong Hình 2.2(c), cửa sổ trượt nằm trên vùng hình ảnh không có góc cạnh, sau khi di chuyển cửa sổ trượt sẽ không có sự thay đổi về cường độ xám.

Dựa vào điều này ta có thể phát hiện ra điểm nào là điểm góc, điểm nào không phải.

2.1.2 Thuật toán

Giả sử ta có một ảnh xám (I), với mỗi điểm (u, v) và độ dịch chuyển (x, y) ta có thể tính toán sự thay đổi trung bình cường độ xám bằng một cửa sổ dịch chuyển từ (u, v) tới $(u + x, v + y)$ như sau:

$$S(x, y) = \sum_u \sum_v w(u, v) (I(u + x, v + y) - I(u, v))^2 \quad (2.2.1)$$

Trong đó:

$S(x, y)$ là Tổng số bình phương giá trị độ lệch hay còn gọi là sự thay đổi cường độ xám tại (x, y)

$w(u, v)$ là cửa sổ tại (u, v)

$I(u, v)$ và $I(u + x, v + y)$ là giá trị cường độ xám của pixel tại các vị trí (u, v) và $(u + x, v + y)$

Lưu ý rằng, hàm $w(u, v)$ thay vì là hàm nhị phân chỉ trượt được theo 4 hướng như trong thuật toán của Moravec, thì nay đã được thay thế bằng hàm Gaussian trong thuật toán Harris để có thể trượt theo nhiều hướng khác nhau.

Giá trị $I(u + x, v + y)$ có thể được khai triển theo công thức Taylor như sau:

$$I(u + x, v + y) \approx I(u, v) + I_x(u, v)x + I_y(u, v)y \quad (2.2.2)$$

Với, I_x, I_y là đạo hàm theo thành phần x, y .

Từ đó, (2.2.1) có thể được viết lại như sau:

$$S(x, y) = \sum_u \sum_v w(u, v) (I_x(u, v)x - I_y(u, v)y)^2 \quad (2.2.3)$$

Nếu biểu diễn dưới dạng ma trận thì $S(x, y)$, ta có:

$$S(x, y) \approx (x \ y) A \begin{pmatrix} x \\ y \end{pmatrix} \quad (2.2.4)$$

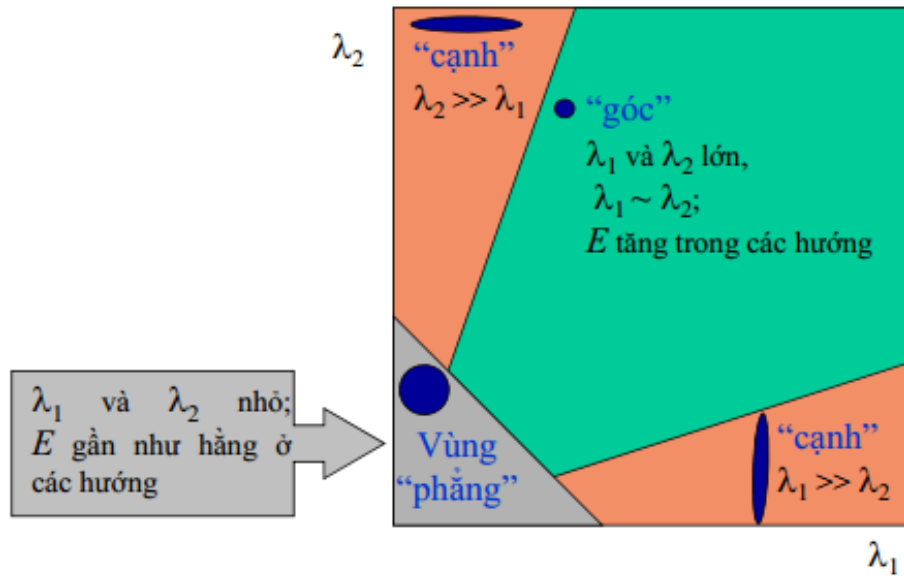
Trong đó, A là một cấu trúc như sau:

$$S(x, y) = \sum_u \sum_v w(u, v) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} = \begin{bmatrix} \langle I_x^2 \rangle & \langle I_x I_y \rangle \\ \langle I_x I_y \rangle & \langle I_y^2 \rangle \end{bmatrix}$$

Gọi λ_1 và λ_2 là các trị riêng của \mathbf{A} , k là hằng số và được xác định thông qua thực nghiệm, thường có giá trị trong khoảng $[0.04, \dots, 0.15]$.

Khi đó, biểu thức dưới đây sẽ quyết định xem cửa sổ w có chứa góc hay là không:

$$M_C = \lambda_1 \lambda_2 - k(\lambda_1 + \lambda_2)^2 = \det(\mathbf{A}) - k(\text{trace}^2(\mathbf{A})) \quad (2.2.5)$$



Hình 2.3: Minh họa các trường hợp λ_1 và λ_2 [4]

Nếu cả λ_1 và λ_2 đều nhỏ. Có nghĩa là hàm $S(x, y)$ gần như không thay đổi theo bất kì hướng nào. Khi đó vùng ảnh nằm trong cửa sổ gần như không có sự thay đổi về cường độ. Tức là trường hợp này không tìm thấy điểm góc.

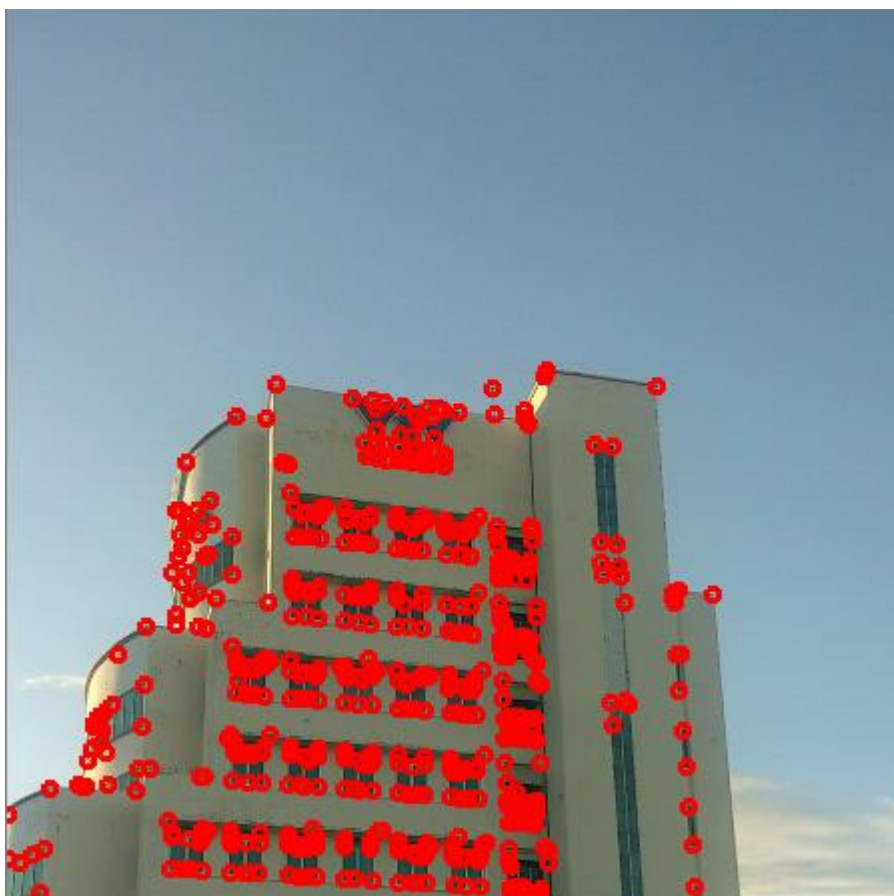
Nếu λ_1 là lớn và λ_2 là nhỏ, hoặc ngược lại, thì có nghĩa là $S(x, y)$ có sự thay đổi nhỏ nếu cửa sổ trượt theo một hướng, và có sự thay đổi đáng kể nếu dịch chuyển theo hướng trực giao. Điều này cho thấy tồn tại một cạnh.

Nếu λ_1 là lớn và λ_2 đều lớn. Khi đó sự dịch chuyển của cửa sổ trượt theo bất kì hướng nào cũng sẽ có sự thay đổi đáng kể về cường độ xám. Điều này biểu thị cho một điểm góc.

2.1.3 Một vài kết quả thực tế



Hình 2.4: Kết quả tìm kiếm góc Harris trên nền C++ và OpenCV2

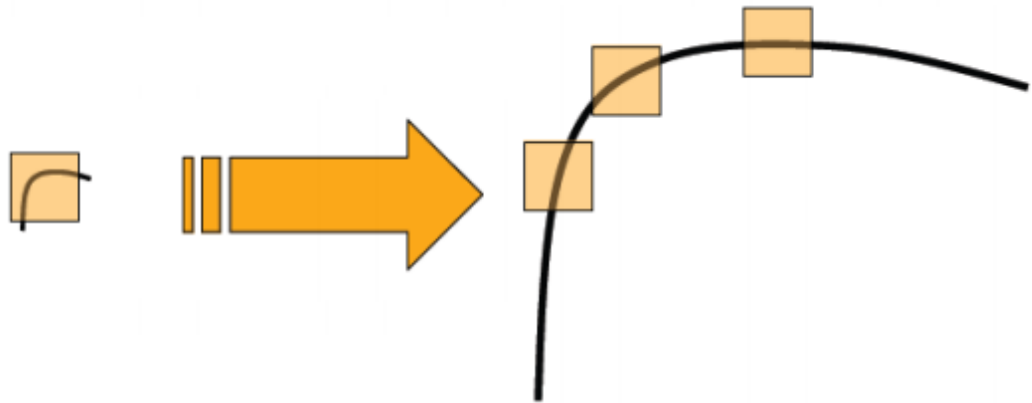


Hình 2.5: Đỉnh tòa nhà VTV Đà Nẵng và những góc Harris

2.2 Trích chọn đặc trưng cục bộ bất biến SIFT

2.2.1 Giới thiệu và định nghĩa

Trong quá trình nghiên cứu việc cứu so sánh đối chiếu các điểm đặc trưng của ảnh số, người ta thường gặp những vấn đề có liên quan đến việc các điểm đặc trưng tương đồng giữa các ảnh không có được sự so sánh đối chiếu chính xác nếu như các điểm đó không có cùng tỉ lệ. Điều này có nghĩa là nếu ta cố gắng tìm ra các điểm đặc trưng tương đồng từ hai hình ảnh được chụp lại từ hai góc độ khác nhau của một khung hình, ta sẽ không có được kết quả tốt nhất[16].



Hình 2.6: Mô tả sự không hoàn thiện của thuật toán Harris[26]

Ví dụ: Trong hình trên, ta dễ dàng nhận ra ở bên trái là một góc đã được thuật toán tìm kiếm góc Harris phát hiện. Thế nhưng nếu như ta xem xét góc này với một tỉ lệ lớn hơn nhiều thì đó không còn là một góc theo thuật toán Harris nữa, nhưng trên thực tế, đó hoàn toàn là một góc.

Để giải quyết vấn đề này, khái niệm “Điểm đặc trưng cục bộ bất biến SIFT” (Scale-Invariant Feature Transform)[12]: đã được David G. Lowe đưa ra hồi năm 2004 trong “Distinctive Image Features from Scale-Invariant Keypoints” với các tính chất sau:

- Là bất biến khi
 - Thay đổi tỷ lệ ảnh
 - Quay ảnh
 - Thay đổi góc nhìn
 - Thêm nhiễu ảnh
 - Thay đổi cường độ chiếu sáng ảnh
- Các đặc trưng cục bộ bất biến được trích rút từ các điểm nổi bật trên ảnh.

2.2.2 Thuật toán

Phương pháp trích chọn điểm đặc trưng cục bộ bất biến SIFT gồm các bước[12]:

- Phát hiện các điểm cực trị Scale-Space: Tìm những khu vực chứa những điểm đặc trưng tiềm năng có tính chất không thay đổi dưới các phép phóng và xoay ảnh.
- Định vị các điểm nổi bật: Một hàm kiểm tra sẽ được đưa ra để quyết định xem các điểm nổi bật tiềm năng có được lựa chọn hay không.
- Xác định hướng cho các điểm nổi bật: Mỗi điểm nổi bật sẽ được gán cho một hoặc nhiều hướng dựa trên hướng gradient của ảnh. Mọi phép toán xử lý ở các bước sau này sẽ được thực hiện trên những dữ liệu ảnh mà đã được biến đổi tương đối so với hướng đã gán, kích cỡ và vị trí của mỗi điểm đặc trưng. Nhờ đó, tạo ra một sự bất biến trong các phép xử lý này.
- Mô tả các điểm nổi bật: Các hướng gradient cục bộ được đo trong ảnh có kích cỡ cụ thể nào đó trong vùng lân cận với mỗi điểm đặc trưng. Sau đó, chúng sẽ được biểu diễn thành một dạng mà cho phép mô tả các tầng quan trọng của quá trình bóp méo hình dạng cục bộ và sự thay đổi về độ sáng.

2.2.2.1 Phát hiện các điểm cực trị Scale-Space

Định nghĩa không gian tỉ lệ của một hình ảnh được mô tả bằng hàm dưới đây:

$$L(x, y, \sigma) = G(x, y, k\sigma) * I(x, y) \quad (2.2.1)$$

Với $G(x, y, k\sigma)$ là Biến tỉ lệ Guassian (Variable scale Gaussian)

$$G(x, y, k\sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \quad (2.2.2)$$

$I(x, y)$ là Ảnh đầu vào

$L(x, y, \sigma)$ là Hàm không gian tỉ lệ của ảnh

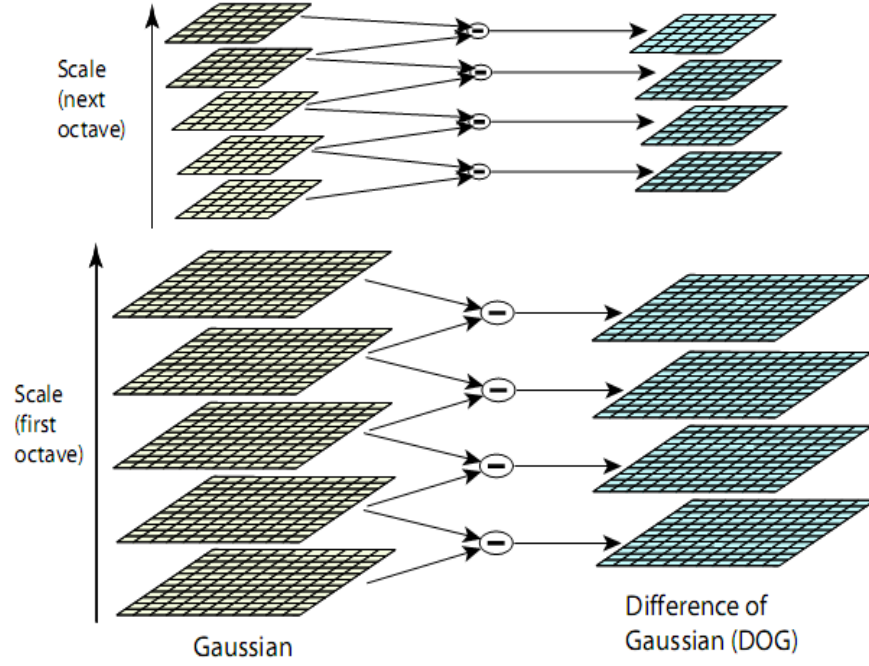
Để tìm những điểm nổi bật có tính bất biến cao, ta sẽ tìm cực trị cục bộ của hàm sai khác DoG (Difference of Gaussian), kí hiệu là $D(x, y, \sigma)$. Hàm này được tính toán từ sự sai khác giữa 2 không gian đo cạnh nhau của một ảnh với tham số đo lệch nhau một hằng số k .

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) = L(x, y, k\sigma) - L(x, y, \sigma) \quad (2.2.3)$$

Với $D(x, y, \sigma)$ được tính bằng phép trừ giữa Hàm không gian tỉ lệ của ảnh $L(x, y, \sigma)$ và k lần của chính nó.

Lý do để lựa chọn hàm Gaussian là vì kỹ thuật này rất hiệu quả để tính toán L (cũng như làm tăng độ mịn của ảnh), mà L thì luôn phải được tính rất nhiều để mô

tả đặc trưng trong không gian đo, và sau đó, D sẽ được tính bằng phép trừ ma trận điểm ảnh, một phép tính toán đơn giản.



Hình 2.7: Quá trình tính không gian đo (L) và hàm sai khác D

Hơn nữa, hàm sai khác DoG có thể được sử dụng để tạo ra một xấp xỉ gần với đạo hàm bậc hai Laplace có kích thước chuẩn của hàm Gaussian ($\sigma^2 \nabla^2 G$) do tác giả Lindeberg đề xuất năm 1994[17]. Ông đã chỉ ra rằng việc chuẩn hóa đạo hàm bậc hai với hệ số σ^2 là cần thiết cho bất biến đo trở nên đúng. Cụ thể, ông đã công bố rằng các giá trị cực đại và cực tiểu của ($\sigma^2 \nabla^2 G$) chính là những giá trị có tính ổn định nhất (bất biến cao) so với một loạt các hàm đánh giá khác như: gradient, Hessian hay Harris.

Mối quan hệ giữa D và ($\sigma^2 \nabla^2 G$) được biểu diễn như sau:

$$\frac{\partial G}{\partial \sigma} = \sigma \nabla^2 G \quad (2.2.4)$$

Như vậy, ($\sigma^2 \nabla^2 G$) có thể được tính thông qua việc xấp xỉ sự sai khác hữu hạn $\frac{\partial G}{\partial \sigma}$ tại các tham số đo gần nhau $k\sigma$ và σ :

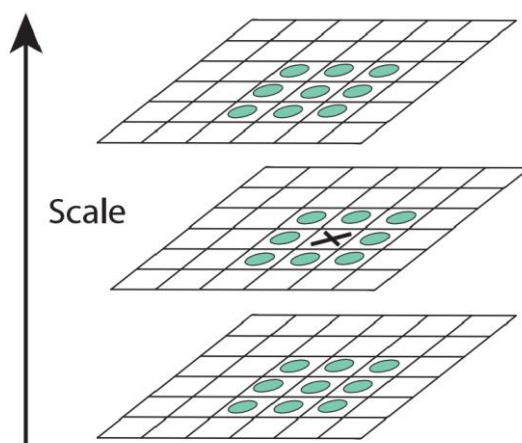
$$\sigma \nabla^2 G = \frac{\partial G}{\partial \sigma} \approx \frac{G(x, y, k\sigma) - G(x, y, \sigma)}{k\sigma - \sigma} \quad (2.2.5)$$

$$G(x, y, k\sigma) - G(x, y, \sigma) \approx (k - 1)(\sigma^2 \nabla^2 G) \quad (2.2.6)$$

Từ công thức này, ta thấy khi mà hàm sai khác DoG được tính toán tại các tham số đo lệch nhau một hằng số k , thì ta có thể sử dụng DoG để xấp xỉ đạo hàm bậc hai Laplace của Gaussian. Vì hệ số $(k - 1)$ trong phương trình trên là hằng số trong mọi không gian đo nên nó sẽ không ảnh hưởng đến việc tìm các vị trí cực trị.

Sai số trong việc xấp xỉ đạo hàm bậc 2 tiến về **0** khi **k** gần với **1**. Tuy nhiên, các kết quả thử nghiệm của tác giả cho thấy quá trình xấp xỉ đạo hàm không ảnh hưởng đến việc dò tìm các vị trí cực trị thậm chí ngay cả khi chọn **k** khá xa, ví dụ **$k = \sqrt{2}$** [12].

Sau khi áp dụng hàm DoG ta thu được các lớp kết quả khác nhau từ ảnh gốc, bước tiếp theo là tìm các cực trị trong các lớp kết quả theo từng miền cục bộ. Cụ thể là tại mỗi điểm trên các lớp kết quả sẽ được so sánh với 8 điểm lân cận trên cùng lớp và 9 điểm lân cận trên mỗi lớp khác.



Hình 2.8: Quá trình tìm điểm cực trị trong các hàm sai khác DoG

Trong hình trên: điểm đánh dấu **x** sẽ được so sánh với 26 điểm lân cận (đánh dấu vòng tròn xanh). Điểm này sẽ được lấy làm điểm tiềm năng (điểm có thể làm điểm đặc biệt – candidate keypoint) nếu nó có giá trị lớn nhất hoặc nhỏ nhất so với 26 điểm lân cận như trên. Giải pháp cho việc tìm các điểm tiềm năng này là sử dụng thuật toán *blob detection* (dò tìm điểm) do Lindeberg đề xuất[19].

Vì số lượng các cực trị là rất lớn, vì vậy để tăng sự hiệu quả khi dò tìm các điểm cực trị (dò các điểm cực trị tốt nhất thay vì phải dò hết), ta cần xác định tần số lấy mẫu trong không gian đo và tần số lấy mẫu trong không gian quan sát (không gian ảnh). Thật không may là ta không thể xác định cả 2 loại tần số này một cách tự động trong mỗi tiến trình dò tìm.

Thay vì vậy, các tần số này sẽ được xác định thông qua phương pháp thử nghiệm. Sau khi thử nghiệm với nhiều nguồn dữ liệu ảnh khác nhau, tác giả đã chỉ ra tần số lấy mẫu trong không gian đo tốt nhất là 3 (giữ lại 3 lớp trong mỗi bộ 8 lớp), và tần số lấy mẫu **$\sigma = 1.6$** .

2.2.2.2 Định vị điểm nổi bật

Sau bước một, ta sẽ thu được rất nhiều điểm tiềm năng có thể làm điểm đặc biệt, tuy nhiên một số điểm trong số đó là không cần thiết. Ở bước này, ta sẽ loại bỏ các điểm có độ tương phản kém (nhạy cảm với nhiễu) hoặc tính đặc trưng cục bộ ít hơn các điểm khác hoặc có xu hướng là đường biên đối tượng. Bước thực hiện này gồm 3 công đoạn:

a. Phép nội suy lân cận cho vị trí đúng của điểm tiềm năng:

Phép nội suy lân cận sử dụng mở rộng Taylor (Taylor expansion) cho hàm Difference-of-Gaussian $D(\mathbf{x}, \mathbf{y}, \sigma)$:

$$D(\mathbf{x}) = D + \frac{\partial D^T}{\partial \mathbf{x}} \mathbf{X} + \frac{1}{2} \mathbf{X}^T \frac{\partial^2 D}{\partial \mathbf{x}^2} \mathbf{X} \quad (2.2.7)$$

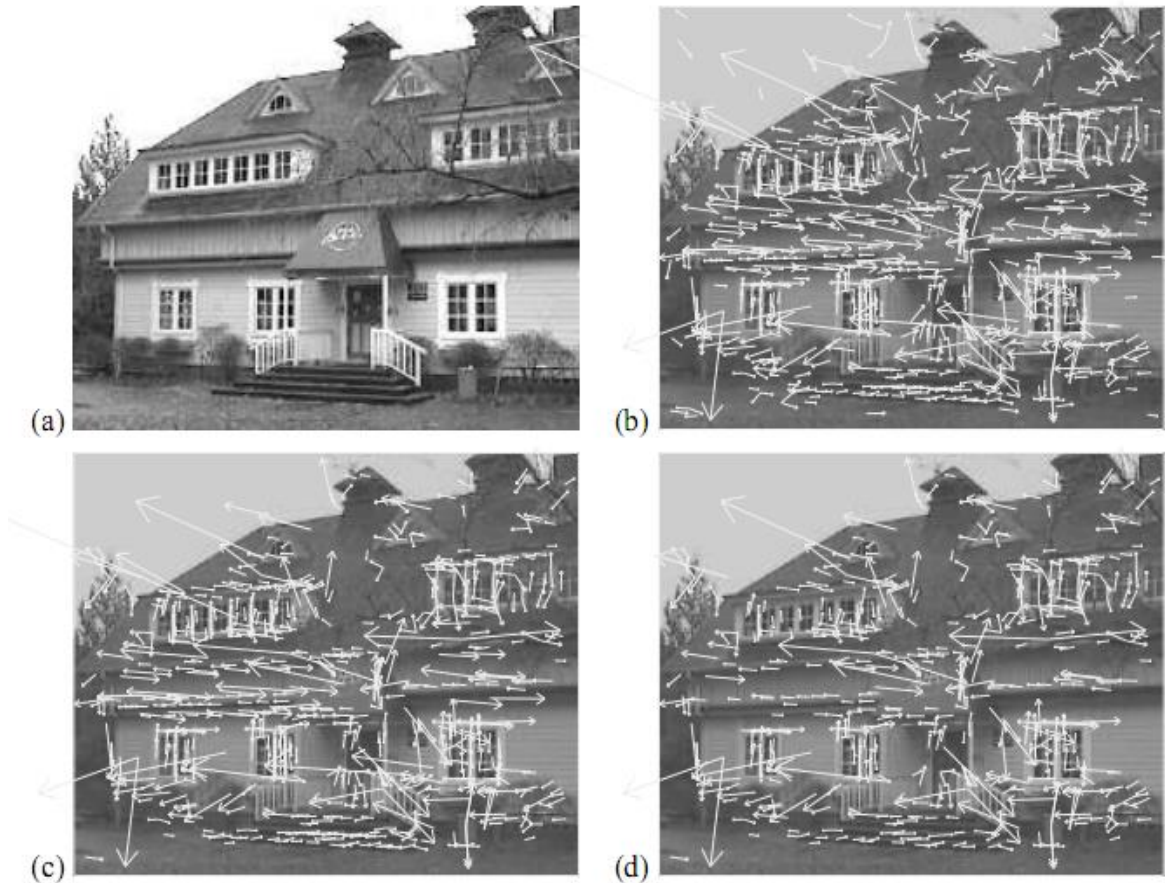
Với $\mathbf{X} = (\mathbf{x}, \mathbf{y}, \sigma)^T$ là Độ dịch so với các điểm lân cận của điểm lấy mẫu.

Vùng chứa điểm nổi bật được xác định qua:

$$\hat{\mathbf{X}} = - \frac{\partial^2 D^{-1}}{\partial \mathbf{x}^2} \frac{\partial D}{\partial \mathbf{x}} \quad (2.2.8)$$

Nếu $\hat{\mathbf{X}} > 0.5$, điểm nổi bật nằm ở gần điểm lấy mẫu khác.

Nếu $\hat{\mathbf{X}} < 0.5$, những điểm này sẽ được đưa vào tập hợp mẫu tốt nhất để tiếp tục phân tích.



Hình 2.9: Minh họa các bước của quá trình lựa chọn các điểm nổi bật

Trong hình trên: (a) là ảnh gốc. (b) mô tả 832 điểm keypoints tìm được, các điểm keypoints được vẽ ở dạng một vector thể hiện 3 thông tin: vị trí, hướng và độ

dài. (c) sau khi đặt ngưỡng tương phản tối thiểu, ta giữ lại được 729 điểm. (d) Giữ lại 536 điểm sau khi áp một ngưỡng nữa về hệ số độ cong.

b. Loại trừ các điểm có tính tương phản kém:

Các điểm nhạy cảm với độ sáng và nhiễu thì không được trở thành điểm đặc biệt và cần loại bỏ khỏi danh sách điểm tiềm năng.

Dùng $D(\hat{X})$ để loại những điểm cực trị không ổn định (có độ tương phản thấp).

Thay \hat{X} vào $D(x)$, ta có:

$$D(\hat{X}) = D + \frac{1}{2} \frac{\partial D^T}{\partial x} \hat{X} \quad (2.2.9)$$

Nếu $D(\hat{X}) < 0.03$ thì điểm lấy mẫu đó sẽ bị loại.

c. Loại bỏ các điểm dư thừa theo biên:

Sử dụng hàm DoG sẽ cho tác động mạnh đến biên khi vị trí của biên là khó xác định và vì vậy các điểm tiềm năng trên biên sẽ không bất biến và bị nhiễu. Và để tăng sự ổn định cho các điểm sẽ được chọn làm điểm đặc biệt ta sẽ loại trừ các điểm tiềm năng khó định vị (tức là vị trí dễ thay đổi khi có nhiễu do nằm ở biên).

Sau khi áp dụng hàm DoG sẽ làm đường biên ảnh không rõ ràng và độ cong chính sẽ có giá trị lớn hơn nhiều so với độ cong dọc theo biên vì vậy cần loại bỏ bớt các điểm đặc biệt dọc theo cùng một biên. Giải pháp cho việc này là sử dụng giá trị của ma trận Hessian cấp 2:

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \quad (2.2.10)$$

Các giá trị riêng của H tỉ lệ thuận với độ cong của D , các giá trị riêng β (giá trị nhỏ) và α (giá trị lớn) có tỉ lệ $r = \alpha/\beta$ sẽ được sử dụng. Các phần tử của H là D_{xx} và D_{yy} .

2.2.2.3 Xác định hướng cho các điểm nổi bật

Bằng việc gán một hướng cho mỗi điểm nổi bật dựa vào các thuộc tính ảnh cục bộ, bộ mô tả điểm nổi bật có thể được biểu diễn tương đối so với hướng này và do đó đạt được tính bất biến đối với các hiện tượng quay ảnh

Độ đo của các điểm nổi bật được sử dụng để tìm ra một ảnh đã lọc Gaussian L với kích thước gần nhất sao cho mọi tính toán sẽ được thực hiện trong cùng một cách bất biến về độ đo.

Tính toán hướng và độ lớn cho điểm nổi bật:

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2} \quad (2.2.11)$$

$$\theta(x, y) = \tan^{-1}((L(x, y + 1) - L(x, y - 1)) / (L(x + 1, y) - L(x - 1, y))) \quad (2.2.12)$$

Với:

$m(x, y)$ là Độ lớn của vector định hướng

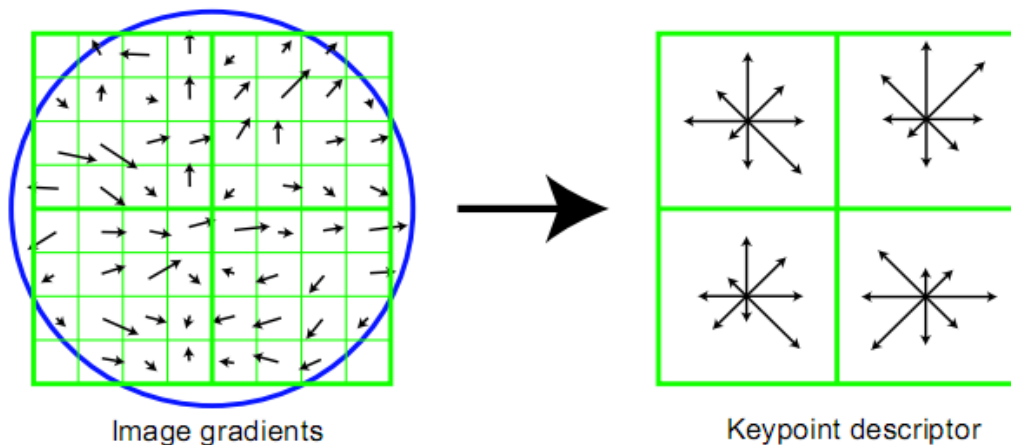
$\theta(x, y)$ là Hướng của vector định hướng (biểu diễn qua góc θ)

$L(x, y)$ là Ảnh Guassian ở tỷ lệ nhỏ nhất

2.2.2.4 Mô tả các điểm nổi bật

Các phép xử lý trên đây đã thực hiện dò tìm và gán tọa độ, kích thước, và hướng cho mỗi điểm nổi bật. Các tham số đó yêu cầu một hệ thống tọa độ cục bộ 2D có thể lặp lại được để mô tả vùng ảnh cục bộ và nhờ vậy tạo ra sự bất biến đối với các tham số đó. Bước này sẽ tính toán một bộ mô tả cho một vùng ảnh cục bộ mà có tính đặc trưng cao (bất biến với các thay đổi khác nhau về độ sáng, thu – phóng ảnh, xoay).

Cách tiếp cận này dựa trên một mô hình thị giác sinh học, cụ thể là mô hình neuron phức tạp trong hệ thống não bộ. Các neuron sẽ tương ứng với một gradient tại một hướng và tần số không gian cụ thể, nhưng vị trí của gradient trên võng mạc được phép trượt trên một phạm vi nhỏ của khung nhìn.



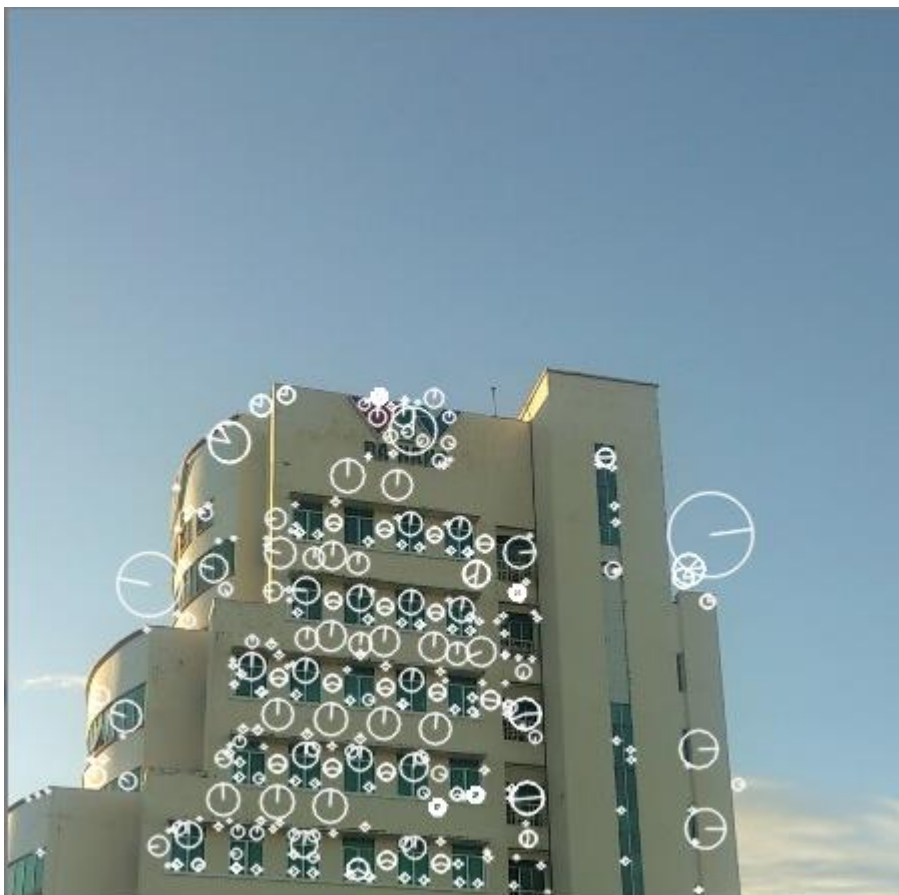
Hình 2.10: Mô tả tạo bộ mô tả cục bộ

Ảnh trái là mô phỏng biên độ gradient và hướng tại mỗi mẫu ảnh trong một vùng lân cận với điểm nổi bật. Các giá trị đó tập trung trong một cửa sổ gaussian (nằm bên trong vòng tròn). Các mẫu này sau đó được gom lại thành một lược đồ hướng mô tả vắn tắt nội dung trong 4x4 vùng con như được mô tả ở bên phải với độ dài của mỗi hàng tương ứng với tổng biên độ gradient gần hướng đó bên trong một vùng.

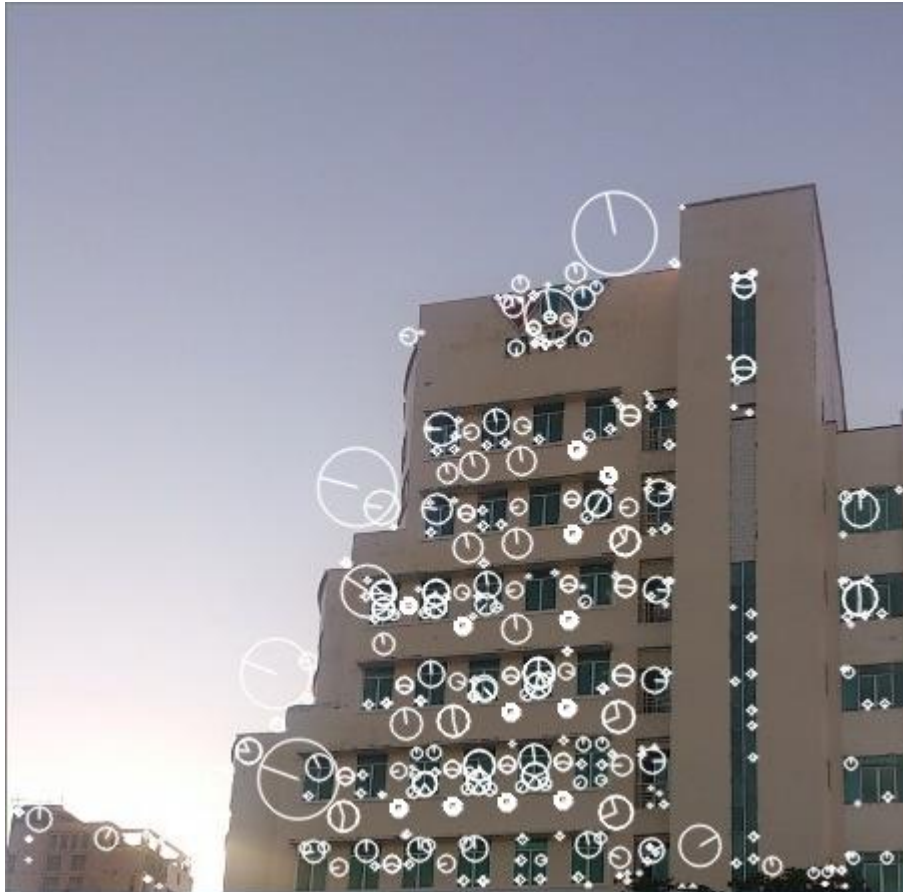
2.1.4 Một vài kết quả thực tế



Hình 2.11: Kết quả tìm kiếm đặc trưng SIFT trên nền C++ và OpenCV2



Hình 2.12: Đỉnh tòa nhà VTV Đà Nẵng và những điểm đặc trưng SIFT(1)



Hình 2.13: Đỉnh tòa nhà VTV Đà Nẵng và những điểm đặc trưng SIFT(2)

Từ hai hình 2.12 và 2.13, bằng mắt thường, ta có thể dễ dàng nhận ra những điểm đặc trưng SIFT khá giống nhau về vị trí và mô tả. Đó chính là những điểm nổi bật không bị ảnh hưởng bởi tỉ lệ mà ta gọi là đặc trưng cục bộ bất biến SIFT.

2.3 Trích chọn đặc trưng SURF

2.3.1 Giới thiệu và định nghĩa

Phương pháp SIFT đã giải quyết được những hạn chế còn tồn tại ở thuật toán tìm kiếm góc Harris và trở thành một trong những thuật toán trích chọn đặc trưng mạnh mẽ nhất. Dù vậy, tốc độ xử lý của SIFT vẫn còn rất chậm và không phù hợp với các ứng dụng thời gian thực[26].

Để giải quyết bài toán này, người ta đã giới thiệu thuật toán trích chọn đặc trưng SURF (Speed Up Robust Features) có được sự cân bằng giữa yêu cầu tốc độ và sự chính xác. Đặc trưng tối ưu cả hai giai đoạn phát hiện đặc trưng (detector) và mô tả đặc trưng (descriptor) về mặt thời gian tính toán nhưng vẫn giữ được tính bền vững của đặc trưng[15].

Bộ phát hiện đặc trưng của SURF sử dụng phép xấp xỉ trên ma trận Hessian và ảnh tích hợp (Integral Image)[13][14] để làm giảm thời gian tính toán một cách đáng kể. Bộ mô tả đặc trưng tương tự như đặc trưng SIFT, sử dụng vector 64 chiều chứa thông tin biến thiên trên ảnh dựa trên sự phân phối bậc nhất Haar wavelet tác động trên trục x và y, kết hợp với ảnh tích lũy làm tăng tốc độ tính toán.

SURF được mô tả bởi vector có số chiều ít hơn SIFT nên tốc độ so khớp nhanh hơn, tuy nhiên độ bền vững vẫn được đảm bảo. Hơn thế nữa, bằng việc đánh chỉ mục dựa trên dấu của Laplacian, đặc trưng SURF không chỉ giữ tính bền vững cho đặc trưng mà còn làm tăng tốc độ so khớp (tăng gấp 2 trong trường hợp tốt nhất).

Ảnh tích hợp (Integral Image) là một khái niệm do Viola và Jones đưa ra trong “Rapid object detection using a boosted cascade of simple Features”[13], là một mảng hai chiều với kích thước bằng kích thước của ảnh cần tìm các đặc trưng. Mỗi phần tử của mảng này được tính bằng cách tính tổng của điểm ảnh phía trên (dòng - 1) và bên trái (cột - 1) của nó. Bắt đầu từ vị trí trên, bên trái đến vị trí dưới, bên phải của ảnh, việc tính toán đơn thuần chỉ dựa trên phép cộng số nguyên đơn giản, do đó tốc độ thực hiện rất nhanh.

Ảnh tích hợp có công thức sau:

$$I_{\Sigma}(X) = \sum_{i=0}^{x} \sum_{j=0}^{y} I(i,j) \quad (2.3.1)$$

2.3.2 Thuật toán

Thuật toán của kỹ thuật SURF gồm những bước dưới đây:

- Sử dụng bộ dò Fast-Hessian để xác định các điểm nổi bật
- Gán hướng cho các điểm nổi bật và mô tả đặc trưng SURF
- So khớp đặc trưng.

2.3.2.1 Xác định điểm nổi bật

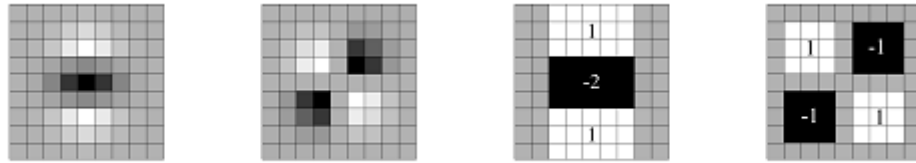
Bộ dò này được xây dựng dựa trên ma trận Hessian vì hiệu suất tốt của nó về thời gian tính toán và độ chính xác, kết hợp với việc sử dụng ảnh tích hợp để giảm thời gian tính toán đặc trưng.

Ma trận Hessian được định nghĩa như sau:

$$H(x, \sigma) = \begin{bmatrix} L_{xx}(x, \sigma) & L_{xy}(x, \sigma) \\ L_{xy}(x, \sigma) & L_{yy}(x, \sigma) \end{bmatrix} \quad (2.3.1)$$

Với $L_{xx}(x, \sigma) = \frac{\partial^2}{\partial x^2} g(\sigma)$ là tích của đạo hàm bậc hai của hàm Gaussian với ảnh I tại điểm $x(x, y)$, có tỉ lệ σ .

Nếu như SIFT xấp xỉ việc tính Laplacian của hàm Gaussian (LoG) bằng việc tính sai khác của hàm Gaussian (DoG) thì SURF xấp xỉ việc tính đạo hàm cấp 2 của hàm Gaussian bằng các hộp lọc (box filters). Dưới đây là một ví dụ của việc tính xấp xỉ đạo hàm cấp hai của hàm Gaussian với hệ số tỉ lệ thấp nhất bằng hộp lọc:



Hình 2.14: Xấp xỉ đạo hàm cấp 2 hàm Gaussian bằng hộp lọc[15]

Trong hình trên: Ảnh thứ nhất là đạo hàm ma trận đạo hàm cấp 2 Gaussian theo trục y, ảnh thứ hai theo trục x và trục y. Ảnh thứ ba và thứ tư lần lượt là các hộp lọc xấp xỉ với hai trường hợp của ảnh một và hai. Phép tích chập xoắn của ảnh I với các hộp lọc này được thực hiện rất nhanh bằng việc sử dụng kết hợp với ảnh tích lũy.

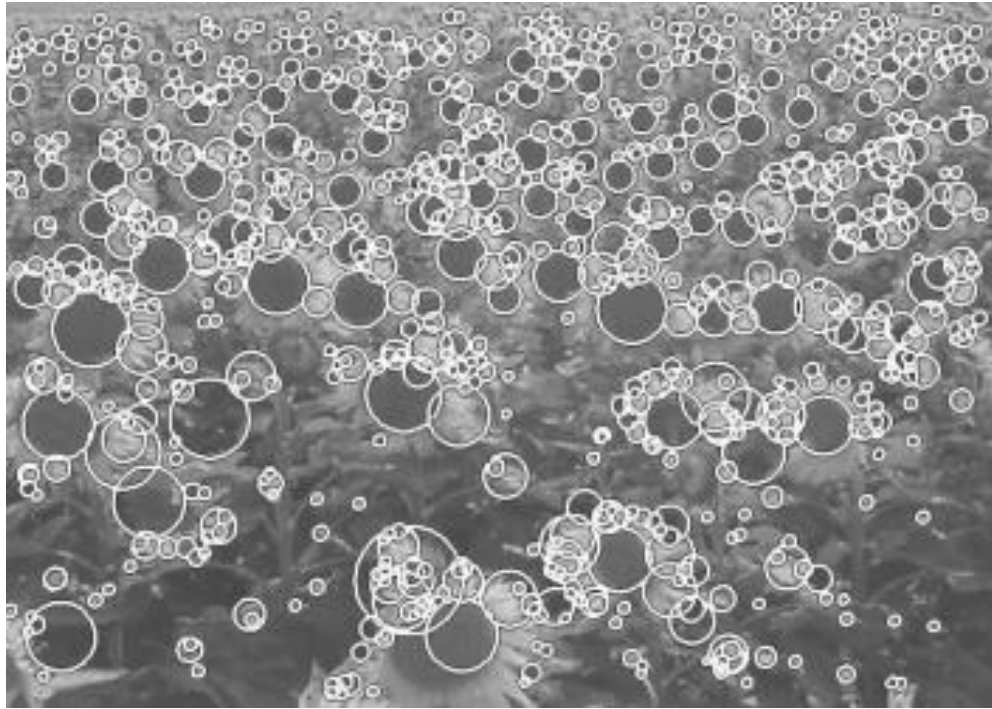
Ta xác định vị trí và hệ số tỉ lệ tương ứng của điểm đặc trưng dựa trên định thức của ma trận Hessian. Công thức tính xấp xỉ định thức ma trận Hessian:

$$\det(H_{xấp xỉ}) = D_{xx}D_{yy} - (w \cdot D_{xy})^2 \quad (2.3.2)$$

Trong đó w là trọng số cân bằng của biểu thức định thức ma trận Hessian tùy thuộc vào hệ số tỉ lệ. D_{xx}, D_{yy}, D_{xy} là các hộp lọc xấp xỉ Gaussian như đã nói ở trên. Đối với tỉ lệ $\sigma = 1.2$, kích thước hộp lọc là 9×9 thì giá trọng số w được tính bằng:

$$w = \frac{|L_{xy}(1.2)|_F |D_{yy}(9)|_F}{|L_{yy}(1.2)|_F |D_{xy}(9)|_F} = 0.912 \approx 0.9 \quad (2.3.3)$$

Vị trí, tỉ lệ và không gian ảnh mà điểm đặc trưng được xác định một phép loại trừ phi cực đại trong một vùng $3 \times 3 \times 3$ sẽ được áp dụng[15].

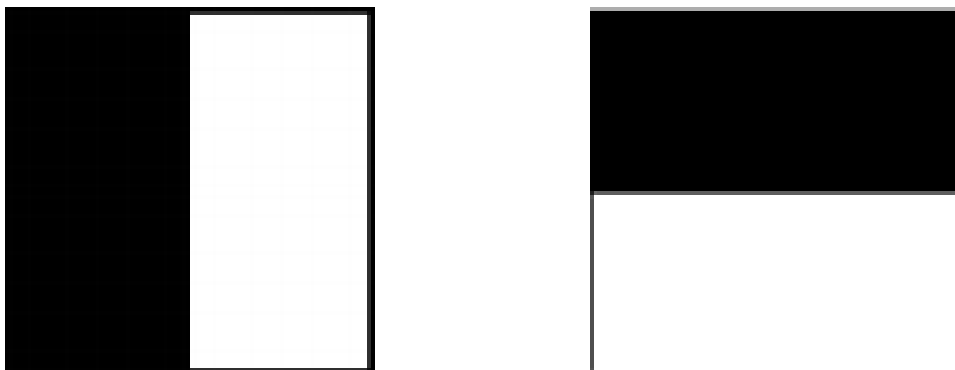


Hình 2.15: Các điểm nổi bật được phát hiện[15]

2.3.2.2 Gán hướng cho điểm nổi bật và mô tả đặc trưng SURF

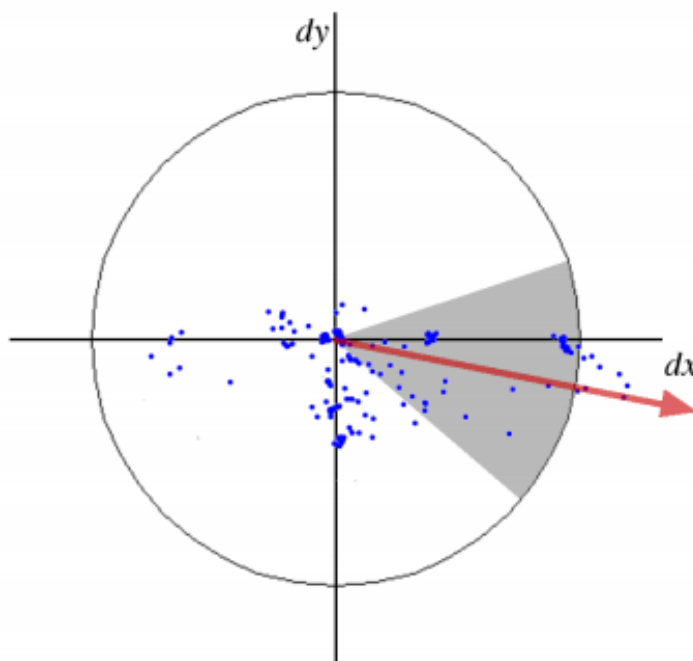
Đầu tiên, ta phải xác định vùng hình xung quanh điểm đặc trưng vừa tìm được, gán một giá trị hướng duy nhất cho điểm đặc trưng. Kích thước của hình tròn phụ thuộc và hệ số tỉ lệ tương ứng trong không gian ảnh mà điểm đặc trưng tìm được.

Ở đây các tác giả chọn bán kính của hình tròn là $6s$, trong đó s là tỉ lệ mà tại đó điểm đặc trưng được tìm thấy. Hướng của đặc trưng được tính bằng Haar wavelet tác động theo hai hướng x và y (Hình 2.16). Trong đó, vùng tối có trọng số -1 , vùng sáng có trọng số $+1$. Kích thước của wavelet cũng phụ thuộc vào hệ số tỉ lệ s , ở đây các tác giả chọn $4s$.



Hình 2.16: Lọc Haar wavelet để tính sự ảnh hưởng trên hai hướng x và y [15]

Haar wavelet có thể được tính một cách nhanh chóng bằng cách sử dụng ảnh tích lũy tương tự như hộp lọc xấp xỉ của đạo hàm cấp 2 hàm Gaussian. Vector hướng nào trội nhất sẽ được ước lượng và gắn vào thông tin của điểm đặc trưng. Hình dưới đây sẽ mô tả hướng và vùng ảnh hưởng của đặc trưng.



Hình 2.17: Vùng hình tròn xung quanh và hướng đại diện cho điểm đặc trưng[15]

Tiếp theo, ta xây dựng các vùng hình vuông xung quanh điểm đặc trưng men theo vector hướng vừa ước lượng được ở bước trước đó. Vùng hình vuông này được chia nhỏ thành 4x4 hình vuông con để ghi nhận thông tin của trên miền không gian ảnh lân cận. Haar wavelet được rút trích trên toàn bộ không gian điểm ảnh. Wavelet tác động trên hai hướng ngang và dọc được cộng dồn các giá trị d_x và d_y trên mỗi hình vuông con.

Hơn thế nữa, các giá trị tuyệt đối $|d_x|$ và $|d_y|$ cũng được cộng dồn để lấy thông tin về độ lớn của sự thay đổi cường độ sáng trên ảnh.

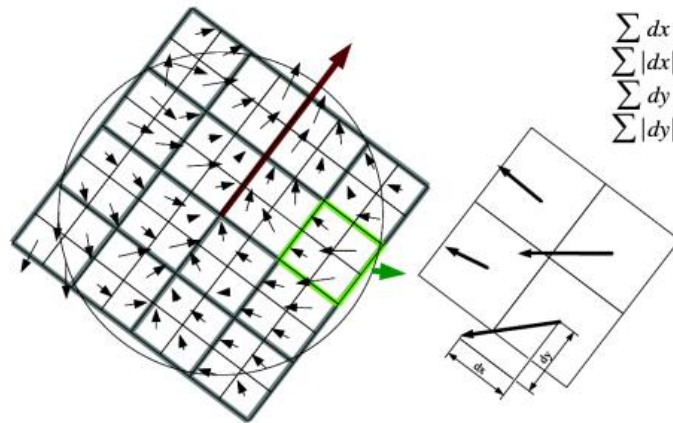
Như vậy mỗi hình vuông con sẽ được mô tả bởi một vector 4 chiều:

$$\mathbf{V} = [\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|] \quad (2.3.4)$$

Như vậy vector mô tả cho tất cả 4x4 hình vuông con là một vector 64 chiều (4x4x4) (Hình 2.18). Đây cũng chính là mô tả đặc trưng chuẩn của SURF (hay còn gọi là SURF-64). Ngoài ra còn có các phiên bản khác dựa trên cách chia hình vuông con như SURF – 36, SURF – 128...

Tuy nhiên thực nghiệm của các tác giả cho thấy rằng SURF – 64 cho tốc độ tính toán tốt nhất mà vẫn đảm bảo tính bền vững của đặc trưng. Haar wavelet bất biến với sự

thay đổi của ánh sáng và sự tương phản khi ta chuẩn hóa vector mô tả đặc trưng về chiều dài đơn vị.

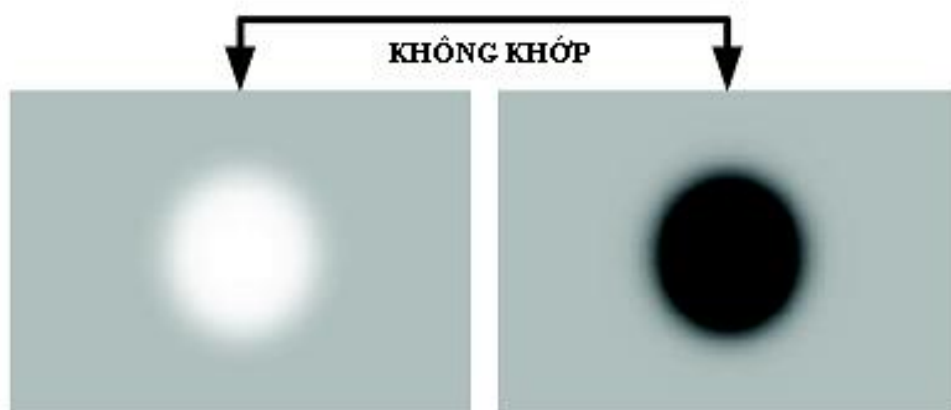


Hình 2.18: 4x4 hình vuông con xung quanh điểm đặc trưng

2.3.2.3 Lập chỉ mục và so khớp

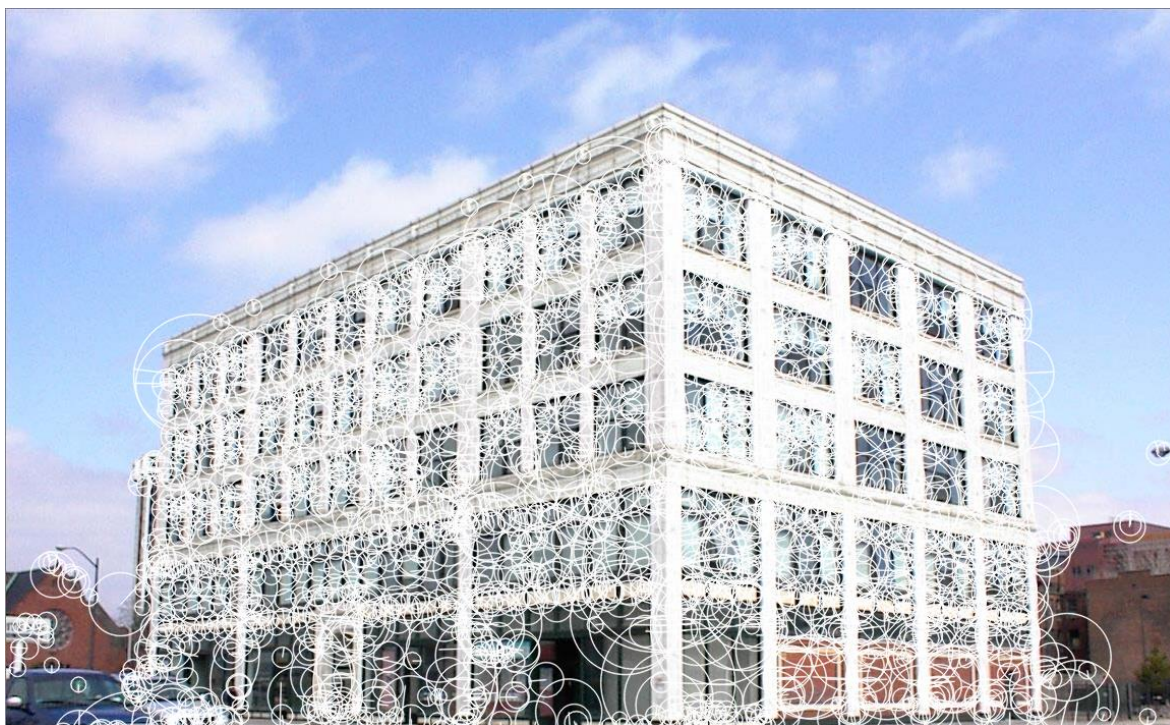
Đặc điểm quan trọng của đặc trưng SURF là quá trình rút trích đặc trưng nhanh do sử dụng kỹ thuật ảnh tích lũy và phép loại trừ phi cực đại. Quá trình so khớp đặc trưng cũng nhanh hơn rất nhiều chỉ bằng một bước đánh chỉ mục đơn giản dựa trên dấu của Laplacian (trace của ma trận Hessian). Ta không phải tốn chi phí tính toán trong bước này do trong quá trình phát hiện đặc trưng đã được tính sẵn.

Dấu của Laplacian giúp phân biệt đốm sáng trên nền tối và đốm tối trên nền sáng. Điểm đặc trưng sáng chỉ có thể khớp với điểm đặc trưng sáng khác, tương tự cho đặc trưng tối (Hình 2.19). Kỹ thuật này có thể giúp cho quá trình so khớp nhanh gấp đôi trong trường hợp tốt nhất do không phải tốn chi phí tính toán dấu của Laplacian.

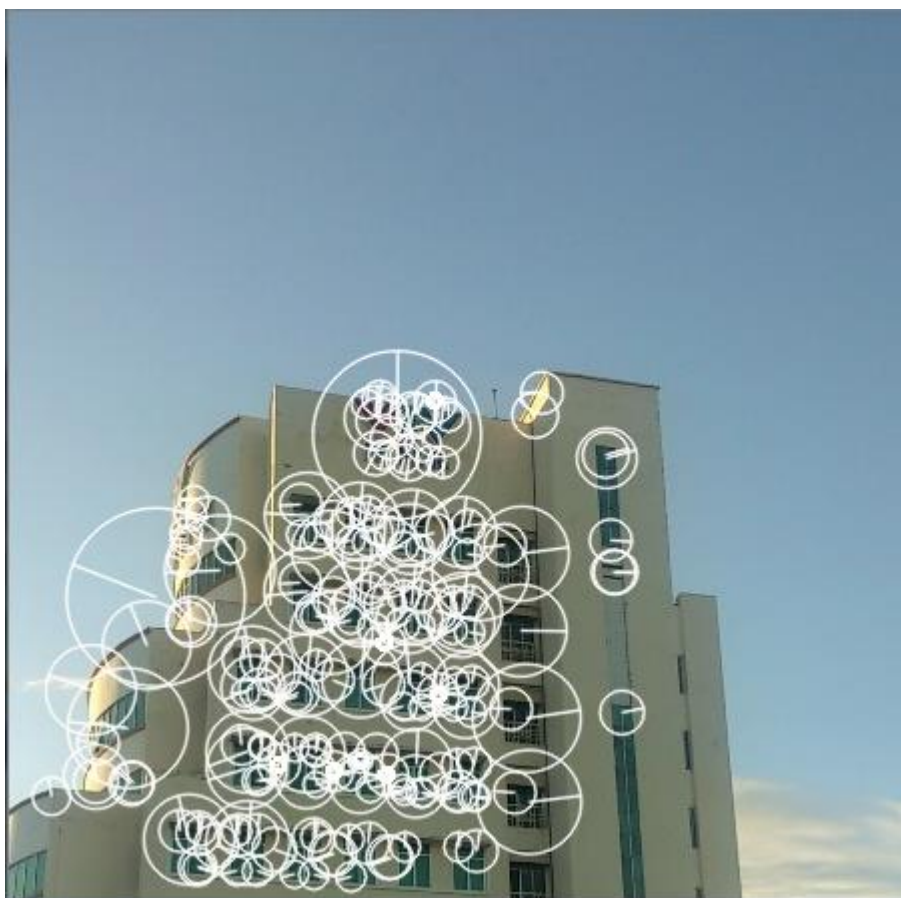


Hình 2.19: Ví dụ về đốm sáng trên nền tối hơn và đốm tối trên nền sáng hơn

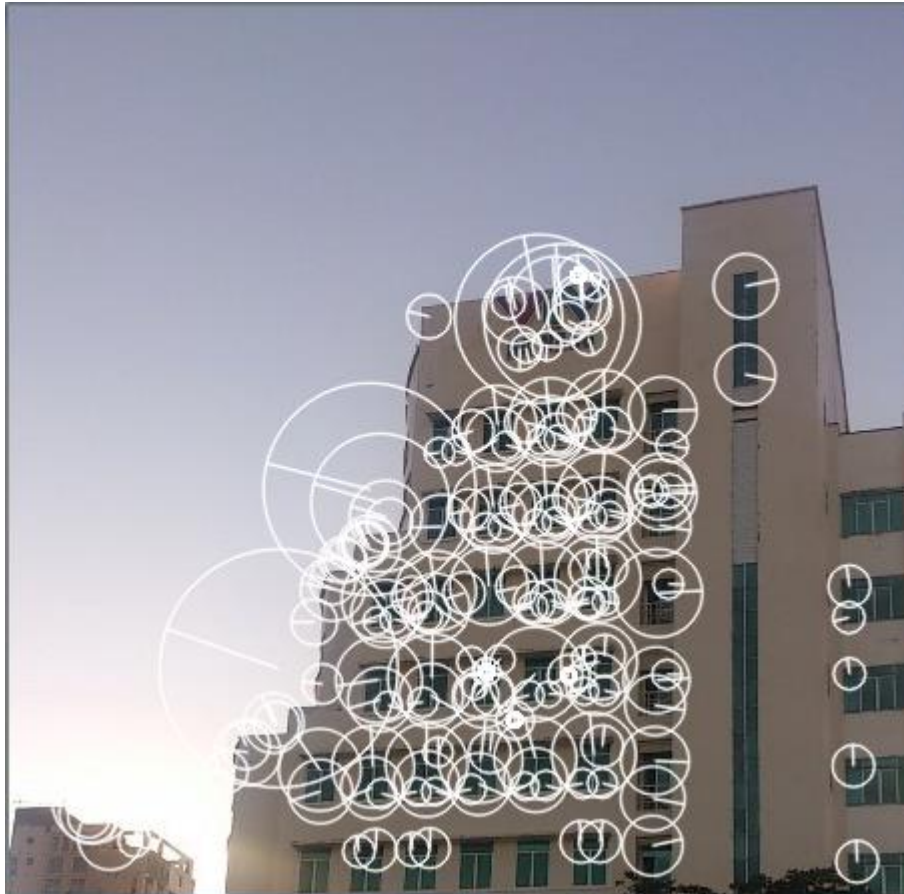
2.3.3 Một vài kết quả thực tế



Hình 2.20: Kết quả tìm kiếm đặc trưng SURF trên nền C++ và OpenCV2



Hình 2.21: Đỉnh tòa nhà VTV Đà Nẵng và những điểm đặc trưng SURF(1)



Hình 2.22: Đỉnh tòa nhà VTV Đà Nẵng và những điểm đặc trưng SURF(2)

CHƯƠNG 3: ỨNG DỤNG VÀO LĨNH VỰC TẠO ẢNH PANORAMA

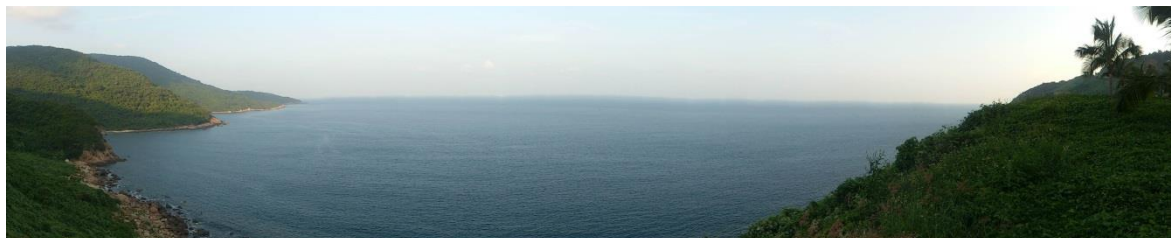
3.1 Giới thiệu

Thuật ngữ panorama xuất hiện trước khi con người có máy ảnh panorama. Nguyên gốc của thuật ngữ này là do họa sĩ người Ireland – Robert Baker – dùng để mô tả những bức tranh diện rộng ở Edinburgh (Đức)[20].

Năm 1881, họa sĩ người Hà Lan Hendrik Willem Mesdag đã tạo nên trường phái panorama Mesdag[17] với những ống hình trụ cuốn các bức tranh toàn cảnh với những kích cỡ khổng lồ, cao 14m và dài có thể từ 40 – 120 m. Thế kỉ XIX, có hai bức tranh panorama được coi là lớn nhất thời kì này, đó là bức tranh mô tả trận chiến tại Atlanta với chiều cao gần 13m, dài 110m và bức tranh ở Wroclaw(Ba Lan) với chiều cao là 15m và chiều dài là 120m.

Ảnh panorama là ảnh được ghép từ những ảnh số chụp từng phần của một phong cảnh (ảnh sau có phần chung với ảnh trước) lại thành một ảnh toàn cảnh hoàn chỉnh.

Một cách đơn giản, Panorama là chế độ chụp ảnh khổ rộng bằng cách chụp nhiều tấm ảnh liên tiếp, với thông tin của tấm ảnh trước được thể hiện một phần trong tấm ảnh sau, nhằm hỗ trợ người dùng. Sau đó với sự trợ giúp của phần mềm xử lý ảnh thì ta sẽ có được 1 tấm ảnh khổ rộng.



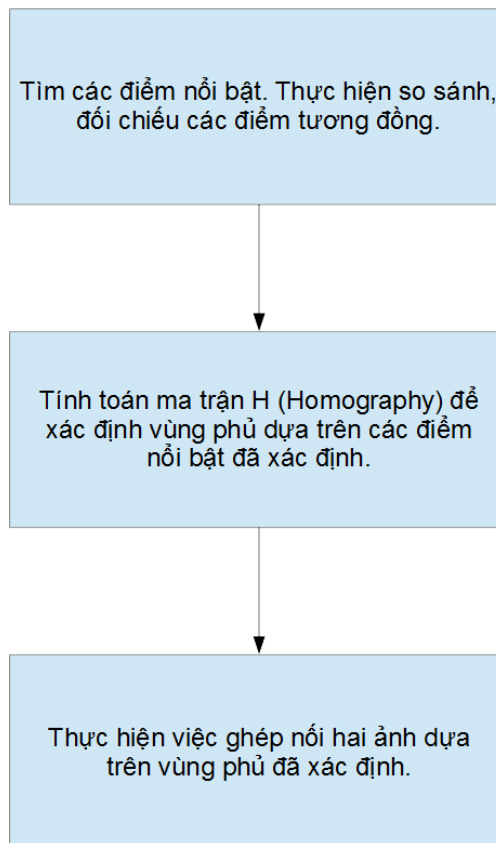
Hình 3.1: Một ảnh Panorama

Panorama được chia làm nhiều loại, trong đó có các loại phổ biến như sau:

- Ảnh phẳng (*flat*): có chiều rộng lớn hơn nhiều lần chiều cao
- Ảnh vertorama: ghép ảnh với nhau theo chiều dọc.
- Ảnh trụ (*cylinder*): chụp một vòng 360 độ để tạo ảnh panorama, sau đó dán nó vào mặt hình trụ.
- Ảnh cầu (*sphere*): Ảnh được phủ lên khắp quả cầu tròn.
- Ảnh hộp (*cube*): Ảnh được phủ lên một hình hộp.

Chương này sẽ trình bày quy trình và phương pháp tạo ra một ảnh Panorama phẳng một cách cơ bản nhất.

3.2 Quy trình thực hiện



Hình 3.2: Minh họa các bước thực hiện thuật toán ghép nối ảnh

Tìm điểm nổi bật và so sánh, đối chiếu

Đầu tiên, ta sẽ sử dụng một trong các kỹ thuật đã trình bày ở trên để đi tìm các điểm nổi bật ở hai hình ảnh cần ghép nối.

Sau đó, dựa trên những điểm nổi bật tìm được được, so sánh, đối chiếu để xác định những điểm nổi bật tương đồng giữa hai hình ảnh và thực hiện.

Tính toán Homography

Sau khi thực hiện so sánh đối chiếu để tìm ra những điểm nổi bật giống nhau, ta sẽ sử dụng những điểm nổi bật giống nhau giữa hai hình ảnh để tính toán ma trận Homography vốn cùng tồn tại ở cả hai ảnh thông qua thuật toán RANSAC.

Cần lưu ý rằng, ta chỉ có thể thực hiện ghép nối ảnh dựa trên Homography khi và chỉ khi cả hai ảnh này chứa các đối tượng được chụp từ hai góc nhìn khác nhau của một Camera duy nhất.

Từ kết quả có được, ta sẽ dễ dàng tìm thấy vùng phủ chung của hai ảnh cần ghép nối. Chi tiết thực hiện tính toán Homography sẽ được trình bày ở phần dưới.

Ghép nối ảnh

Sau khi ma trận Homography được tính toán, bước cuối cùng trong việc tạo ảnh panorama là hòa trộn hai bức ảnh lại với nhau.

3.3 So sánh đối chiếu các điểm tương đồng nổi bật

3.3.1 Cơ sở ban đầu

Sau khi tìm được các điểm nổi bật thông qua các thuật toán tìm kiếm góc Harris, tìm điểm nổi bật SIFT hoặc tìm điểm nổi bật SURF, để xác định được chắc chắn giữa hai ảnh ban đầu có những điểm nổi bật hay vùng tương đồng với nhau.

Mặt khác, đây không chỉ là một bước trong quy trình tạo ảnh Panorama, mà còn là bước cơ bản trong các bài toán yêu cầu tìm kiếm hình vật thể, tìm kiếm ảnh tương đồng với một ảnh cho trước từ cơ sở dữ liệu hay xác định hai ảnh bất kì là giống hay khác nhau.

Hiện nay, có nhiều phương pháp để thực hiện bước so sánh đối chiếu điểm tương đồng nổi bật giữa hai ảnh. Trong phần này, đề tài sẽ trình bày phương pháp so sánh đối chiếu bằng phương pháp vét cạn (Brute-Force)[28].

Trong khoa học máy tính, vét cạn (Brute-Force Search) là thuật ngữ dùng để chỉ phương pháp giải quyết vấn đề một cách tổng quát mà trong đó, tất cả các khả năng để giải quyết vấn đề sẽ được lần lượt sử dụng cho đến khi vấn đề được giải quyết.

Trong lĩnh vực xử lý ảnh, mà cụ thể là so sánh đối chiếu các điểm tương đồng nổi bật, phương pháp vét cạn được khái quát như sau:

So sánh đối chiếu theo kiểu vét cạn là phương pháp mà trong đó, mô tả của mỗi điểm nổi bật của nhóm đầu tiên sẽ lần lượt được so sánh với mô tả của các điểm nổi bật của nhóm thứ hai sao cho tìm ra được các cặp điểm nổi bật được mô tả giống nhau nhất. Mỗi cặp điểm được tìm ra được gọi là một cặp điểm tương đồng nổi bật[26].

3.3.2 Thực hiện trên nền C++ và OpenCV2

Giả sử ta có 2 ảnh được tạo thành từ 2 góc nhìn khác nhau của một tòa nhà như sau:



Hình 3.3: Hai hình ảnh được sử dụng so sánh đối chiếu

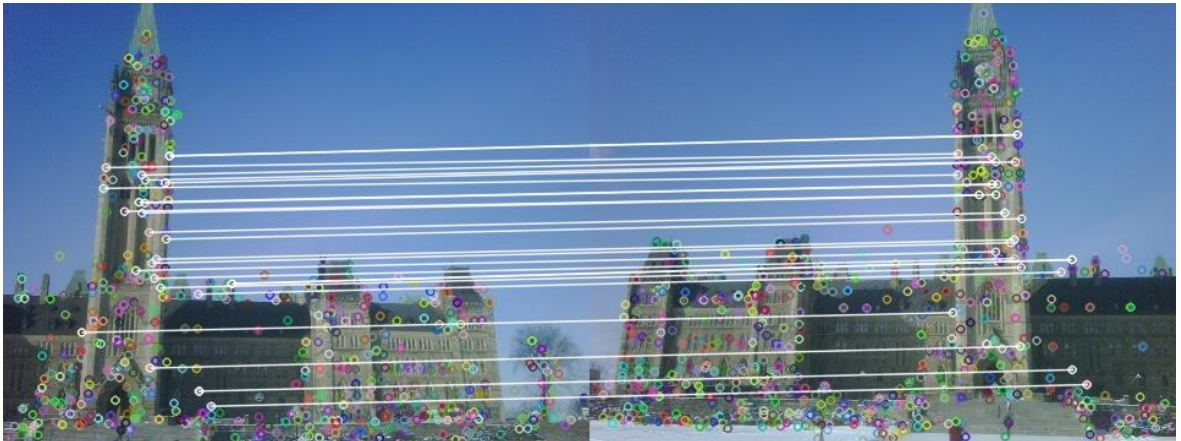
Trong đó, hình bên trái là góc nhìn bên phải của tòa nhà, còn hình bên phải là góc nhìn bên trái của tòa nhà.

Giả sử phương pháp lấy điểm đặc trưng ta chọn sử dụng sẽ là SIFT, ta có kết quả lấy điểm đặc trưng như sau:



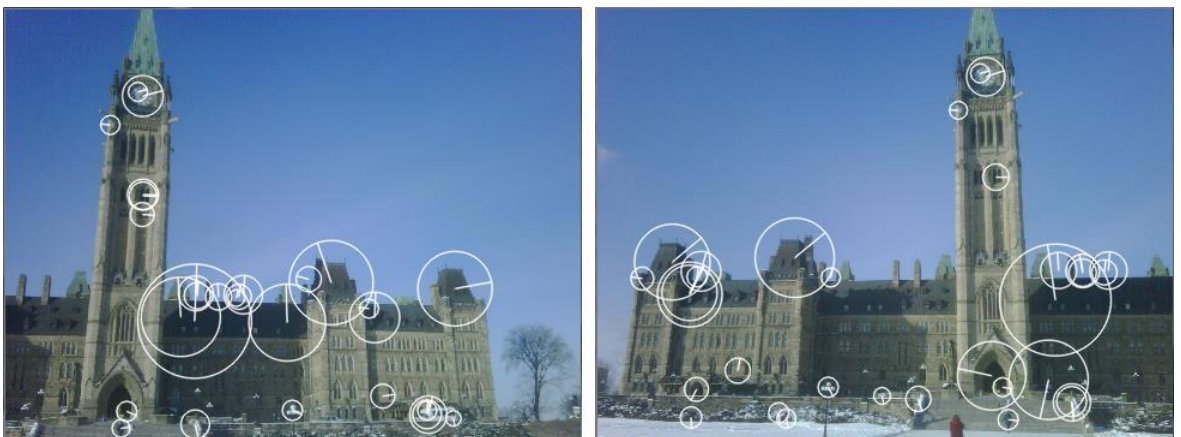
Hình 3.4: Kết quả sau khi lấy đặc trưng SIFT

Thực hiện so sánh đối chiếu bằng thuật toán Brute-Force, kết quả nhận được sẽ là hình sau:

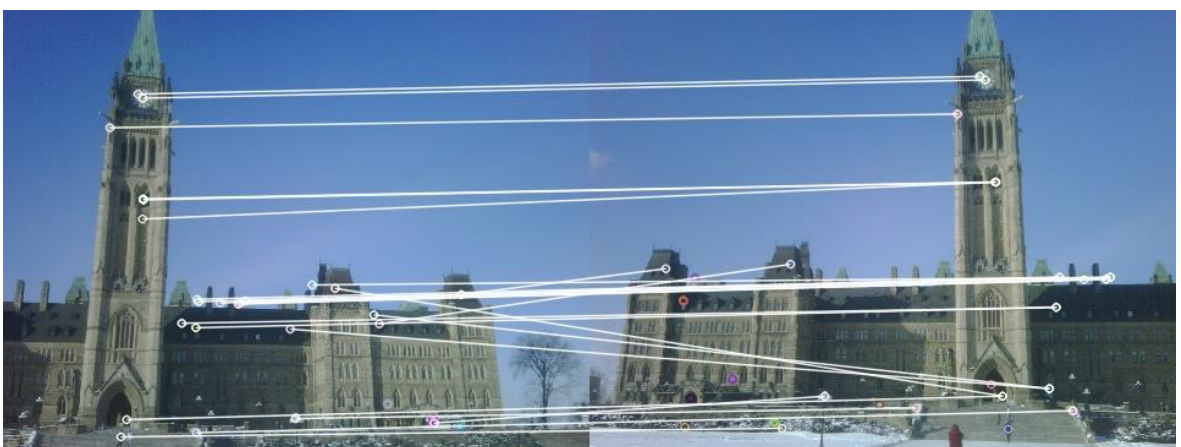


Hình 3.5: Kết quả sau khi so sánh đối chiếu các điểm đặc trưng SIFT

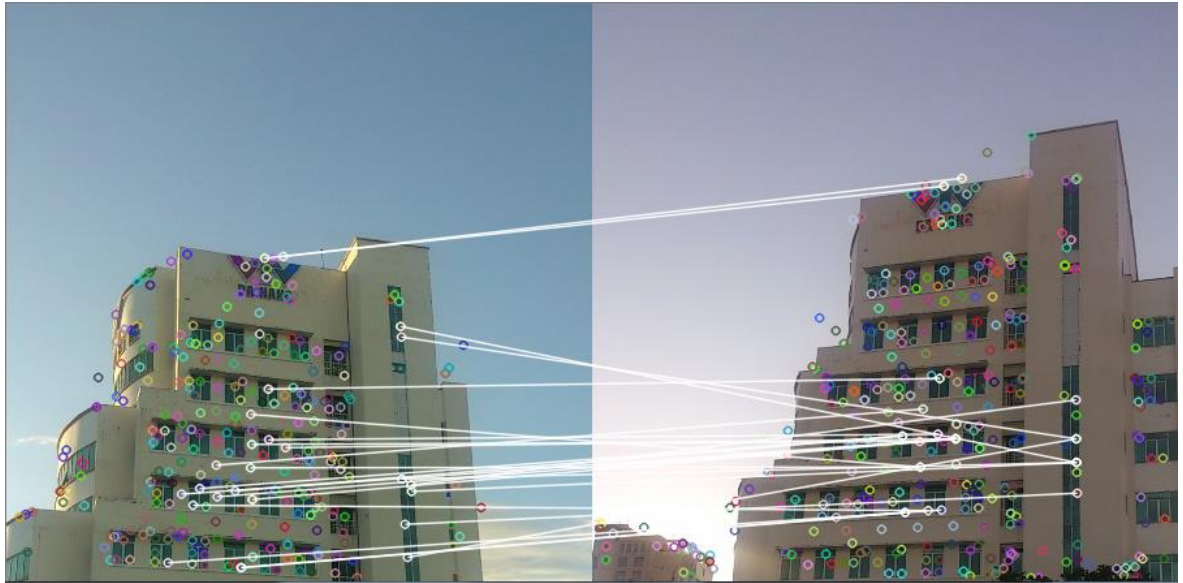
Trong như ta chọn phương pháp lấy đặc trưng SURF thay vì phương pháp lấy đặc trưng SIFT, kết quả lần lượt nhận được sẽ như sau:



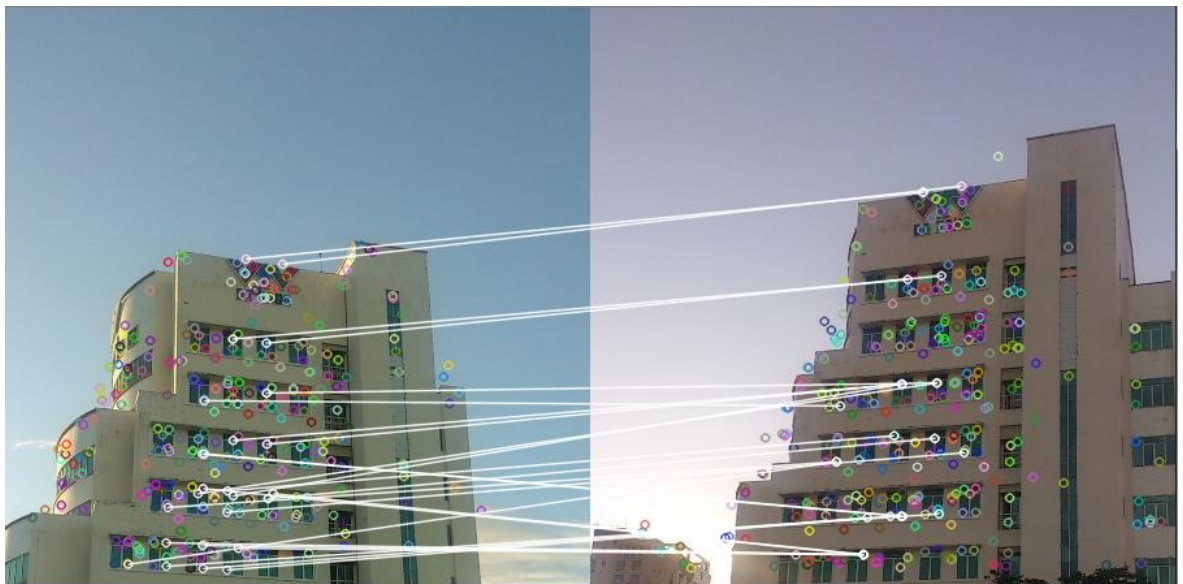
Hình 3.6: Kết quả sau khi lấy đặc trưng SURF



Hình 3.7: Kết quả sau khi so sánh đối chiếu các điểm đặc trưng SURF



Hình 3.8: Kết quả sau khi so sánh đối chiếu các điểm đặc trưng SIFT đối với hình Hình 2.11 và Hình 2.12 ở mục 2.2.3



Hình 3.9: Kết quả sau khi so sánh đối chiếu các điểm đặc trưng SURF đối với hình Hình 2.11 và Hình 2.12 ở mục 2.3.3

3.4 Tính ma trận Homography bằng thuật toán RANSAC

3.4.1 Vài nét về Homography

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 tỷ lệ.

Trong lĩnh vực thị giác máy [23], 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 kỳ 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.

Ta có công thức sau:

$$HX = sX' \quad (3.4.1)$$

Trong đó:

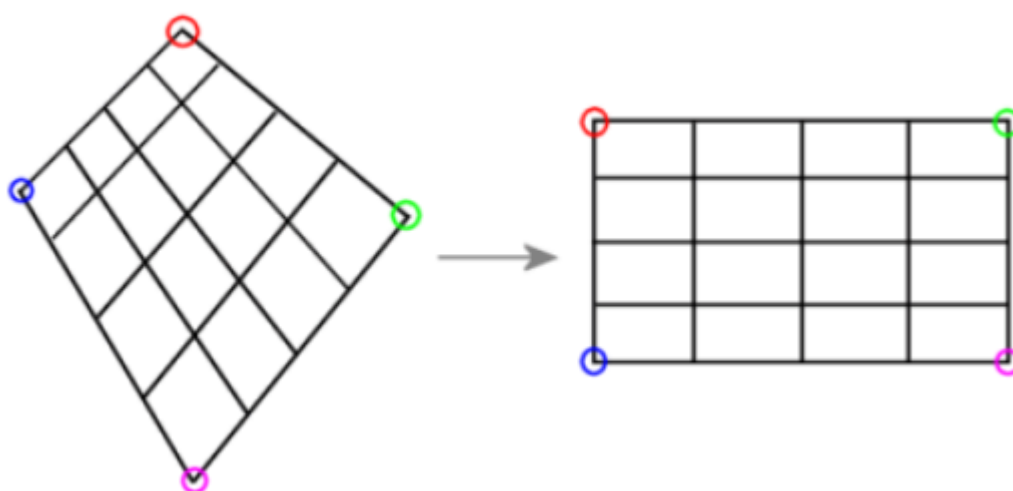
s là hằng số tỉ lệ của phép chiếu và khác 0

X' là kết quả của phép ánh xạ

X là đối tượng ảnh được ánh xạ

H là ma trận Homography, là một ma trận khả nghịch.

Vì H là một ma trận khả nghịch, cho nên trong trường hợp muốn tái tạo ảnh X từ X' , ta chỉ cần xác định được ma trận Homography.



Hình 3.10: Phép chiếu Homography

3.4.2 Tính toán Homography

3.4.2.1. Phân rã giá trị đơn SVD (Singular Value Decomposition)

Phân rã giá trị đơn SVD [6][7] là phương pháp đại số được sử dụng nhiều trong các bài toán yêu cầu việc tính toán ma trận vốn sẽ cho ra kết quả sai số lớn nếu như sử dụng các phương pháp thông thường như khử Gauss hay phân tích LU.

Phân rã SVD dựa trên định lý đại số tuyến tính sau:

Mọi ma trận $[A]$ có kích thước $M \times N$ nào mà có $M \geq N$ thì đều có thể viết lại dưới dạng tích của một ma trận $[U]$ trực giao theo cột có kích thước $M \times N$ với một ma trận chéo $[W]$ có kích thước $N \times N$ với số đường chéo là không âm và một ma trận chuyển vị của một ma trận trực giao $[V]$ có kích thước $N \times N$:

$$[A] = [U] \times \begin{bmatrix} S_1 & \\ & S_n \end{bmatrix} \times [V^T] \quad (3.4.2)$$

Ma trận $[U]$ và ma trận $[V]$ trực giao theo nghĩa các cột của chúng là trực giao:

$$\sum_{i=1}^M U_{ik} U_{in} = \delta_{kn} \quad \begin{cases} 1 \leq k \leq N \\ 1 \leq n \leq N \end{cases} \quad (3.4.3)$$

$$\sum_{j=1}^N U_{jk} U_{jn} = \delta_{kn} \quad \begin{cases} 1 \leq k \leq N \\ 1 \leq n \leq N \end{cases} \quad (3.4.4)$$

Trong đó, $\delta_{kn} = 1$ nếu $k = n$ và bằng 0 nếu $k \neq n$. Ta cũng có thể biểu diễn dưới dạng ma trận:

$$[U^T][U] = [V^T][V] = \mathbf{1} \quad (3.4.5)$$

Vì $[V]$ là ma trận vuông nên cũng đồng thời là ma trận trực giao theo hàng:
 $[V^T][V] = \mathbf{1}$

Các phần tử của $[S]$ khác 0 và được gọi là giá trị đơn. Theo quy ước, vector đơn được xác định bằng cách sắp xếp các giá trị đơn theo thứ tự giảm dần.

Nếu gọi u_i và v_i là tương ứng các cột của ma trận $[U]$ và $[V]$, ta có phương trình phân tích giá trị đơn của A như sau:

$$A = \sum_{i=1}^n u_i s_i v_i^T \quad (3.4.6)$$

s_i là các giá trị đơn của A , do đó $s_1 \geq s_2 \geq s_3 \geq \dots \geq s_n \geq 0$

Chỉ số giá trị đơn khác 0 cuối cùng trong dãy được kí hiệu là ρ . Tức, $s_i = 0$ với $i > \rho$ và $s_i > 0$ với $i \leq \rho$.

Ta có cách tính SVD của ma trận $[A]$:

Tính V^T và S bằng cách chéo hóa ma trận A :

$$A^T A = V S^2 V^T \quad (3.4.7)$$

Sau đó, tính $[U]$ theo công thức:

$$U = AVS^{-1} \quad (3.4.8)$$

Cho phương trình $AX = 0$, để tìm ra X ta sẽ dùng phương pháp giá trị đơn SVD. A sẽ được phân tích thành $A = USV^T$. Khi đó, X sẽ là cột cuối cùng của ma trận $[V]$.

3.4.2.2. Tính Homography bằng phương pháp Direct Linear Transform

Để tính ma trận Homography từ các cặp điểm tương ứng, người ta dùng phương pháp DLT (Direct Linear Transform) [8], gồm có hai bước: Đầu tiên, từ các cặp điểm tương ứng, ta chuyển về dạng ma trận $A_i h = 0$. Sau đó, áp dụng phân rã SVD để tính ma trận H .

Phương pháp DLT với các điểm nổi bật được tìm thấy từ thuật toán Harris:

Trong tọa độ không đồng nhất, công thức (1) được viết lại như sau:

$$c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (3.4.9)$$

Lần lượt chia dòng thứ nhất của công thức trên cho dòng thứ ba và dòng thứ hai cho dòng thứ ba, ta có:

$$\begin{aligned} -h_1x - h_2y - h_3 + (h_7x + h_8y + h_9)_u &= 0 \\ -h_4x - h_5y - h_6 + (h_7x + h_8y + h_9)_u &= 0 \end{aligned}$$

Viết lại dưới dạng ma trận, ta có:

$$A_i h = \begin{bmatrix} -x & -y & -1 & 0 & 0 & 0 & ux & uy & u \\ 0 & 0 & 0 & -x & -y & -1 & vx & vy & v \end{bmatrix} (h_1 h_2 \dots h_9)^T \quad (3.4.10)$$

Với mỗi cặp điểm tương ứng, ta có hai biểu thức. Mặt khác, do ma trận H là ma trận có bậc tự do (D.O.F) là 8 nên chỉ cần 4 cặp điểm tương ứng là ta có thể xác định được nó.

Áp dụng công thức phân rã SVD cho ma trận $[A]$, ta có:

$$A = U \Sigma V^T = \sum_{i=1}^9 s_i u_i v_i^T \quad (3.4.11)$$

Với s_i là các giá trị đơn và được sắp xếp nhỏ dần, nên s_9 là giá trị nhỏ nhất. Khi đó, giá trị của h_i bằng giá trị cuối cùng của cột v_i .

Theo lý thuyết, với 4 cặp điểm tương ứng sẽ tìm được một ma trận H với D.O.F bằng 8. Tuy nhiên, trong thực tế các ảnh đầu vào có thể có gốc tọa độ ở góc trái của ảnh, cũng có thể gốc tọa độ nằm ở tâm ảnh. Nếu để tình trạng như vậy sẽ ảnh hưởng đến các kết quả biến đổi về sau như khi nhân ảnh với một hệ số hay các

biến đổi tương tự, affin. Do vậy, Hartley và Zisserman [8] đã đưa ra bước chuẩn hóa để đảm bảo rằng kết quả của thuật toán sẽ cho ra kết quả chính xác. Các ảnh cần phải chuẩn hóa bằng phép biến đổi quay và dịch chuyển.

3.4.2.3. Thuật toán RANSAC

RANSAC [24], đại diện cho cụm từ “Random Sample Consensus”, tức là “đồng thuận mẫu ngẫu nhiên”, là thuật toán khử nhiễu được công bố bởi Fischler và Bolles vào năm 1981.

Ý tưởng chính của RANSAC như sau: Từ tập dữ liệu ban đầu, ta sẽ có hai loại dữ liệu nhiễu và không nhiễu (outlier và inlier), vì thế ta phải đi tính toán để tìm ra mô hình tốt nhất cho tập dữ liệu. Việc tính toán và chọn ra mô hình tốt nhất sẽ được lặp đi lặp lại k lần, với giá trị k được chọn sao cho đủ lớn để đảm bảo xác suất p (thường rơi vào giá trị 0.99) của tập dữ liệu mẫu ngẫu nhiên không chứa dữ liệu nhiễu.

Nếu gọi u là ước lượng dữ liệu không nhiễu thì $v = 1 - u$ là ước lượng dữ liệu nhiễu và m là số lượng dữ liệu đầu vào cần xây dựng mô hình. Khi đó ta có:

$$1 - p = (1 - u^m)^k \quad (3.4.12)$$

k sẽ được tính theo công thức:

$$k = \frac{\log(1-p)}{\log(1-(1-v)^m)} \quad (3.4.13)$$

Kết quả thu được sẽ là mô hình cần xây dựng phù hợp nhất với dữ liệu đầu vào, tập các dữ liệu nhiễu và tập các dữ liệu không nhiễu.

Quá trình thực hiện thuật toán RANSAC được mô tả như dưới đây:

Từ tập dữ liệu đầu vào gồm có nhiễu và không nhiễu ta chọn n dữ liệu ngẫu nhiên, tối thiểu để xây dựng mô hình:

- Tiến hành xây dựng mô hình với n dữ liệu đó, sau đó đặt ra một ngưỡng dùng để kiểm chứng mô hình.
- Gọi tập dữ liệu ban đầu trừ đi tập n dữ liệu để xây dựng mô hình là tập dữ liệu kiểm chứng. Sau đó, tiến hành kiểm chứng mô hình đã xây dựng bằng tập dữ liệu kiểm chứng. Nếu kết quả thu được từ mô hình vượt quá ngưỡng, thì điểm đó là nhiễu, còn không đó sẽ là ngược lại.
- Quá trình này sẽ được lặp đi lặp lại trong k lần. Với k được tính theo công thức trên. Tại mỗi vòng lặp giá trị của k sẽ được tính lại.
- Kết quả là mô hình nào có số dữ liệu không nhiễu nhiều nhất sẽ được chọn là mô hình tốt nhất.

3.4.2.4. Tính ma trận Homography bằng RANSAC

Trong bài toán tạo ảnh Panorama, ma trận Homography được tính từ tập các cặp điểm nổi bật tương ứng của hai ảnh ban đầu đã được so sánh đối chiếu ở bước hai. Khi có bốn cặp điểm nổi bật tương ứng không thẳng hàng, phương trình $A\mathbf{h} = \mathbf{0}$ theo phương pháp DLT chuẩn hóa đã trình bày ở phần trên. Trong đó, A là ma trận có kích thước 8×9 . Từ đó, ta xác định được ma trận \mathbf{h} .

Với ma trận Homography được tính từ bốn cặp điểm ngẫu nhiên, ta có d là khoảng cách đo mức độ gần nhau của các cặp điểm đã được so sánh đối chiếu. Với cặp điểm nổi bật tương đồng $(\mathbf{x}, \mathbf{x}')$ và $d(\vec{\mathbf{a}}, \vec{\mathbf{b}})$ là khoảng cách của hai vector, ta có công thức khoảng cách như sau:

$$d = d(\vec{\mathbf{x}}, H\vec{\mathbf{x}}') + d(\vec{\mathbf{x}}', H\vec{\mathbf{x}}) \quad (3.4.14)$$

Thuật toán chi tiết[18]:

- Khởi tạo số vòng lặp k , ngưỡng *distance*, \max_{inlier} , và $p = 0.99$
- **for**($i = 1:k$), thực hiện các bước sau:
 - Chọn 4 cặp điểm nổi bật tương đồng ngẫu nhiên
 - Kiểm tra xem các điểm này có cùng nằm trên một đường thẳng không, nếu có, quay lại bước trên.
 - Tính ma trận Homography H từ 4 điểm sử dụng phương pháp DLT chuẩn hóa.
 - Tính khoảng cách d của các cặp điểm nổi bật tương đồng
 - $d = d(\vec{\mathbf{x}}, H_{curr}\vec{\mathbf{x}}') + d(\vec{\mathbf{x}}', H_{curr}\vec{\mathbf{x}})$
 - Tính số lượng m các cặp điểm không nhiễu (inlier) thỏa điều kiện:
 $d_i < distance$
 - Nếu $m > \max_{inlier}$ thì $\max_{inlier} = m$, ma trận Homography $H = H_{curr}$
 - Tính k theo công thức dưới đây với $v = 1 - m/n$:

$$k = \frac{\log(1-p)}{\log(1-(1-v)^4)} \quad (3.4.15)$$

- Tiếp tục tính lại ma trận H cho tất cả các cặp điểm tương đồng được coi là không nhiễu (inlier) bằng phương pháp DLT

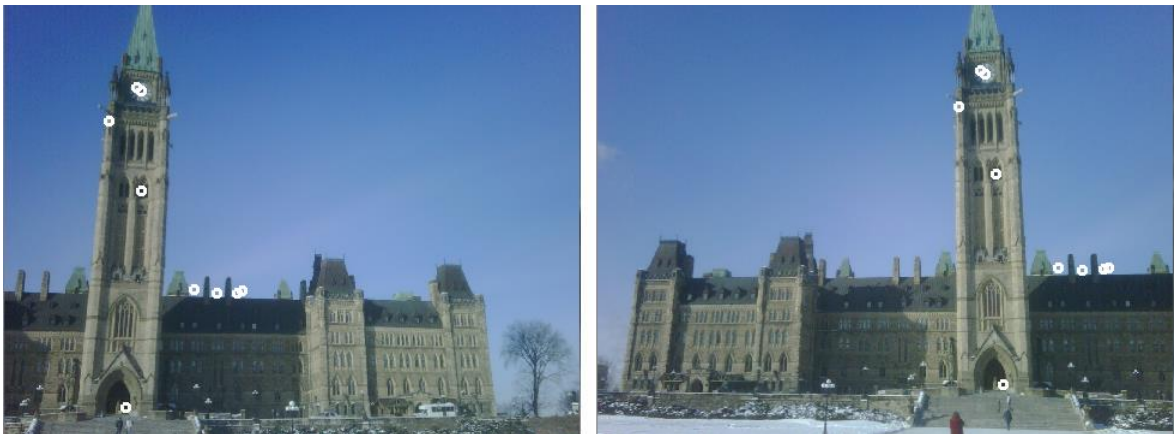
3.3.3 Thực hiện trên nền C++ và OpenCV2

Từ kết quả so sánh đối chiếu điểm đặc trưng SIFT của mục (3.4.2), ta thực hiện phép tính Homography trên nền C++ và OpenCV2 được kết quả như sau:



Hình 3.11: Phép tính Homography từ điểm đặc trưng SIFT

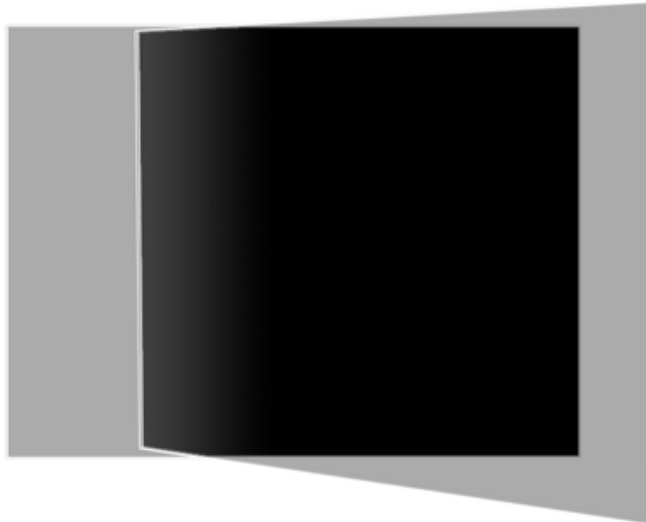
Đối với điểm đặc trưng SURF, ta có kết quả Homography như sau:



Hình 3.12: Phép tính Homography từ điểm đặc trưng SURF

3.5 Ghép nối ảnh

Sau khi ma trận Homography được tính toán, bước cuối cùng trong việc tạo ảnh Panorama là hòa trộn hai bức ảnh lại với nhau. Ý tưởng của bước này là sử dụng một ảnh làm trung tâm, sau đó sử dụng ma trận Homography để chiếu ảnh còn lại tới mặt phẳng ảnh trung tâm.



Hình 3.13: Minh họa ghép nối ảnh

Trong hình trên, bên trái là ảnh trung tâm được sử dụng là mặt phẳng chiếu. Bên phải là ảnh được chiếu lên mặt phẳng ảnh thứ nhất sử dụng ma trận Homography. Phần màu xám là phần riêng của mỗi bức ảnh, còn phần màu đen là phần chung của hai bức ảnh.

Từ kết quả của mục (3.4.3), phép ghép nối ảnh trên C++ và OpenCV2 cho ra kết quả như sau:



Hình 3.14: Kết quả cuối cùng của phép tạo ảnh Panorama

CHƯƠNG 4: THỰC NGHIỆM TRÊN NỀN TẢNG C++ VÀ OPENCV2

4.1 Một vài thông tin ban đầu

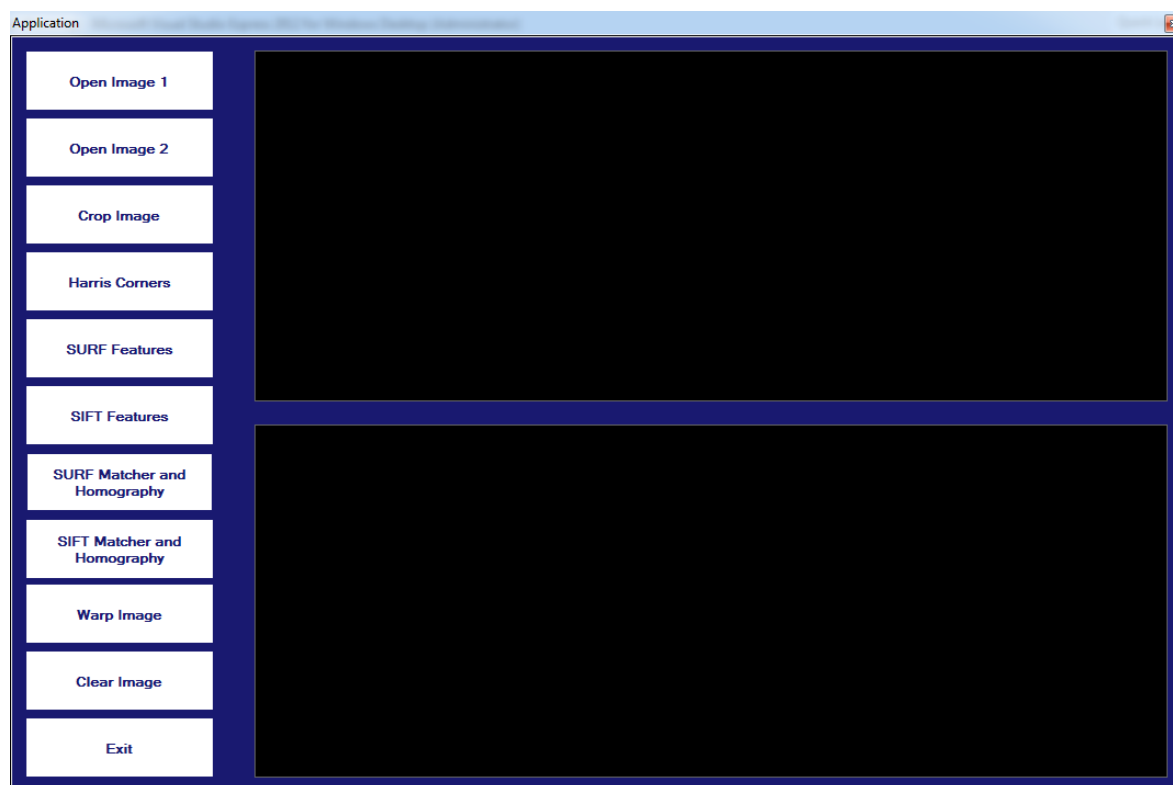
Chương trình sau đây được viết bằng công cụ Visual Studio 2012 dựa trên ngôn ngữ C++ và thư viện Open CV2 phiên bản 2.4.9 với mục đích triển khai các kỹ thuật, thuật toán đã trình bày trong các chương 2, 3, 4.

Trong đó, OpenCV (Open Source Computer Vision) là một thư viện mã nguồn mở về thị giác máy với hơn 500 hàm và hơn 2500 các thuật toán đã được tối ưu về xử lý ảnh, và các vấn đề liên quan tới thị giác máy. OpenCV được có thể chạy trên nhiều nền tảng khác nhau, bao gồm cả hệ điều hành Window, Linux, Mac, iOS, Android...

Dự án về OpenCV được khởi động từ những năm 1999, đến năm 2000 nó được giới thiệu trong một hội nghị của IEEE về các vấn đề trong thị giác máy và nhận dạng, tuy nhiên bản OpenCV 1.0 mãi tới tận năm 2006 mới chính thức được công bố và năm 2008 bản 1.1 (pre-release) mới được ra đời.

Tháng 10 năm 2009, bản OpenCV thế hệ thứ hai ra đời (thường gọi là phiên bản 2.x), phiên bản này có giao diện của C++ (khác với phiên bản trước có giao diện của C) và có khá nhiều điểm khác biệt so với phiên bản thứ nhất. Hiện, thế hệ thứ ba của OpenCV đang được phát triển.

4.2 Giao diện chương trình



Hình 4.1: Giao diện chương trình thực nghiệm

Giao diện chương trình được chia ra thành 2 phần chính: Bên trái là danh sách các phím chức năng, bên phải là cửa sổ hình ảnh. Trong đó:

Phần các phím chức năng từ trên xuống dưới bao gồm:

- **Open Image 1:** Cho phép tải hình ảnh thứ 1 lên chương trình
- **Open Image 2:** Cho phép tải hình ảnh thứ 2 lên chương trình
- **Crop Image:** Cho phép cắt hình ảnh
- **Harris Corners:** Thực hiện thuật toán Harris trên hình ảnh 1 và 2
- **SURF Features:** Thực hiện thuật toán SURF trên hình ảnh 1 và 2
- **SIFT Features:** Thực hiện thuật toán SIFT trên hình ảnh 1 và 2
- **SURF Matcher and Homography:** Đối sánh và tính Homography ảnh 1 và 2 dựa trên đặc trưng SURF
- **SIFT Matcher and Homography:** Đối sánh và tính Homography ảnh 1 và 2 dựa trên đặc trưng SIFT
- **Warp Image:** Ghép nối ảnh 1 và 2 để tạo ảnh Panorama dựa trên đặc trưng SURF
- **Clear Image:** Xóa ảnh 1 và 2 khỏi chương trình
- **Exit:** Thoát chương trình

Phần cửa sổ hình ảnh từ trên xuống dưới bao gồm:

- **Khung đen 1:** Thể hiện hình ảnh thứ 1 đã tải lên
- **Khung đen 2:** Thể hiện hình ảnh thứ 2 đã tải lên

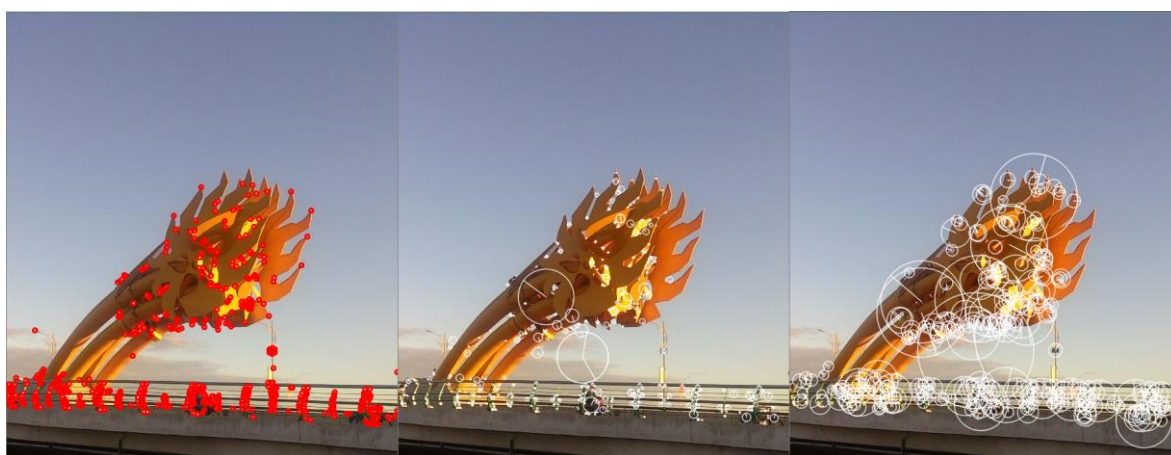
4.3 Một số kết quả có được

4.3.1. Tìm kiếm góc Harris, trích chọn đặc trưng SIFT và SURF

Trong phần này, đề tài sẽ trình bày các kết quả có được từ việc trích chọn điểm nổi bật thông qua các thuật toán tìm kiếm góc Harris, thuật toán trích chọn đặc trưng SIFT và SURF trên một ảnh cho trước.



Hình 4.2: Hình ảnh đuôi cầu Rồng, Tp Đà Nẵng



Hình 4.2: Kết quả trích chọn điểm nổi bật

Trong hình trên, từ trái qua lần lượt là các kết quả của các thuật toán tìm kiếm góc Harris, trích chọn đặc trưng SIFT và trích chọn đặc trưng SURF.

Để so sánh số lượng điểm nhận được và thời gian xử lý, ta có bảng sau:

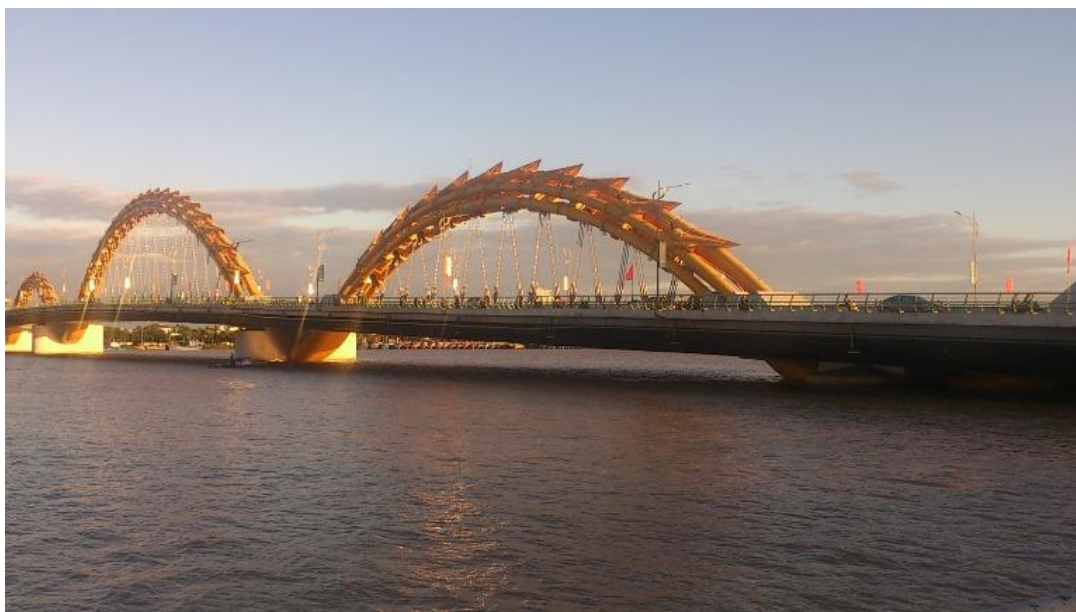
	Số điểm	Thời gian (s)
Harris Corner Detector	534	0.08
SIFT	500	0.42
SURF	330	0.14

Bảng 4.1: So sánh kết quả trích chọn điểm nổi bật

Dễ dàng nhận thấy, thuật toán tìm kiếm góc Harris là phương pháp có tốc độ xử lý nhanh nhất. Về phần SURF và SIFT, SURF cho ra số điểm nổi bật ít hơn so với SIFT nhưng lại có tốc độ xử lý nhanh hơn nhiều.

4.3.2. So sánh đối chiếu các điểm đặc trưng tương đồng SIFT và SURF

Tiếp theo, luận văn sẽ tiến hành so sánh đối chiếu hai hình ảnh cho trước bằng các điểm nổi bật SIFT và SURF. Ta có hai hình ảnh sẽ được sử dụng trong bước này như sau:



Hình 4.3: Hình ảnh cầu Rồng được sử dụng trong thực nghiệm



Hình 4.4: Kết quả so sánh đối chiếu từ điểm đặc trưng SIFT



Hình 4.5: Kết quả so sánh đối chiếu từ điểm đặc trưng SURF

Cần lưu ý rằng, để tiện việc quan sát sự chính xác của phép so sánh đối chiếu, thuật toán chỉ chọn 25 điểm đặc trưng tương đồng giữa 2 ảnh để vẽ đường nối. Thực tế, khi thực hiện bước tạo ảnh Panorama dưới đây, số lượng điểm đặc trưng tương đồng được sử dụng sẽ lớn hơn và phụ thuộc vào thuật toán RANSAC.

Từ kết quả so sánh đối chiếu, ta có bảng sau:

	Số điểm (Hình 1 và 2)	Số điểm nối	Thời gian (s)
SIFT	500 và 500	500	0.78
SURF	472 và 454	472	0.35

Bảng 4.2: So sánh kết quả so sánh đối chiếu điểm đặc trưng tương đồng

4.3.3. *Tạo ảnh Panorama*

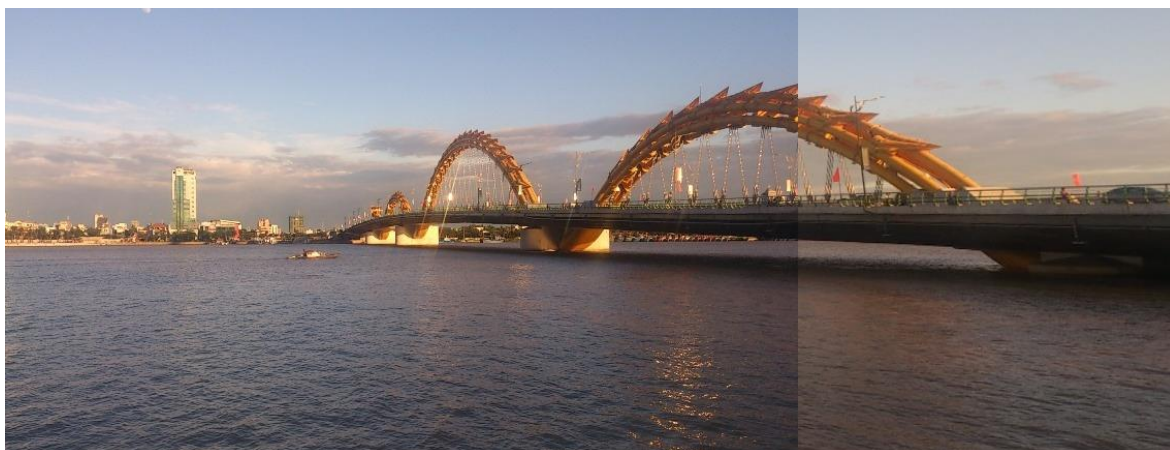
Sau khi thực hiện so sánh đối chiếu điểm đặc trưng tương đồng, bước tiếp theo ta sẽ thực hiện tính toán ma trận Homography. Kết quả có được như sau:



Hình 4.6: Kết quả tính toán ma trận Homography

Trong hình trên, những vòng tròn trắng biểu thị cho những điểm đặc trưng tương đồng nằm trong ma trận Homography giữa 2 ảnh.

Từ kết quả này, ta dễ dàng có được kết quả ảnh Panorama sau phép ghép nối ảnh:



Hình 4.7: Kết quả của phép tạo ảnh Panorama

Dễ thấy rằng, dù kết quả ghép nối ảnh khá hoàn chỉnh, nhưng sự khác biệt về nước màu giữa hai ảnh đã làm cho kết quả cuối cùng không được như ý muốn. Muốn có được kết quả tốt hơn, khi chụp ảnh, ta cần phải có được sự đồng nhất về phương ngang của camera cũng như chế độ ánh sáng. Ví dụ như kết quả sau:

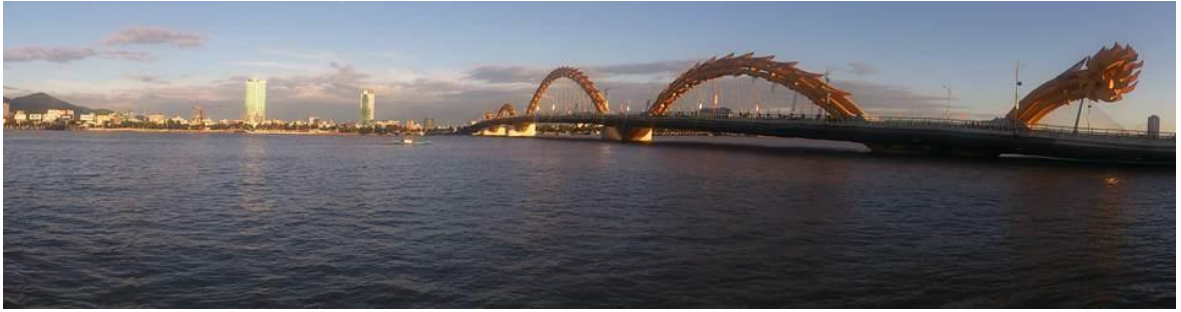
Lý do lớn nhất của điều này là vì giải thuật đề tài áp dụng chỉ cơ bản, chưa đi kèm những giải thuật hiệu chỉnh tinh vi khác về màu sắc, nén ảnh hay chồng ghép.



Hình 4.8: Hình ảnh Panorama khi sử dụng hàm chức năng của OpenCV2

Hiện nay, thư viện OpenCV2 đã tích hợp những hàm chức năng cho phép sự cải tiến mạnh mẽ phép tạo ảnh Panorama, cho phép người lập trình dễ dàng tạo ra những bức ảnh Panorama đẹp mắt. Kết quả được thực hiện trên hàm chức năng đó của OpenCV2 như hình trên.

Áp dụng lại thuật toán trên với số lượng ảnh đầu vào nhiều hơn và một vài công cụ cắt sửa hình ảnh, ta được kết quả sau:



Hình 4.9: Một kết quả hoàn hảo hơn

KẾT LUẬN

Kết quả đạt được

Đề tài đã xem xét tầm quan trọng của công việc “Trích chọn đặc trưng ảnh số” thuộc lĩnh vực xử lý ảnh và thị giác máy. Trong đề tài có nêu các hướng và phương pháp “Trích chọn đặc trưng ảnh số dựa trên điểm nổi bật” như thuật toán Harris, SIFT và SURF, cũng như triển khai các phương pháp này trên thực nghiệm bằng nền tảng C++ và OpenCV2.

Mặt khác, đề tài cũng đã ứng dụng nội dung “Trích chọn đặc trưng ảnh số dựa trên điểm nổi bật” vào quy trình tạo ảnh Panorama từ hai ảnh đầu vào, nêu các bước chính trong quy trình này và triển khai thực nghiệm. Trong đó, mỗi bước đều đã nêu lên các cơ sở phương pháp luận, các kỹ thuật liên quan và đi sâu vào kỹ thuật tiêu biểu như đối sánh điểm đặc trưng tương đồng bằng giải thuật Brute-Force và tính ma trận Homography bằng giải thuật RANSAC.

Hạn chế của đề tài

Có rất nhiều kỹ thuật, phương pháp giải quyết vấn đề trong từng bước của bài toán trích chọn ảnh đặc trưng từ video chưa được xét đến, các kỹ thuật trong đề tài chưa phải là tối ưu nhất hiện nay. Như trong việc tìm kiếm đặc trưng ngoài phương pháp tìm kiếm góc Harris còn có các phương pháp khác như SIFT hay SURF, rồi bài toán đối sánh

Ngoài thuật toán Harris, SIFT và SURF, vẫn còn tồn tại rất nhiều phương pháp trích chọn đặc trưng ảnh số từ điểm nổi bật chưa được xét đến. Mặt khác, giải thuật Harris, SIFT, SURF hay cách thực hiện các bài toán so sánh đối chiếu điểm đặc trưng tương đồng, tính ma trận Homography đều chưa phải là phương pháp tối ưu nhất cả về độ chính xác lẫn tốc độ.

Mặt khác, đặc trưng ảnh số được chia thành nhiều loại với những thuật toán khác nhau ở từng loại. Trong đề tài này mới chỉ nghiên cứu và sử dụng giải pháp “Trích chọn đặc trưng dựa trên điểm nổi bật” để phân tích.

Cuối cùng, ứng dụng tạo ảnh Panorama đã được phát triển vẫn còn cho ra kết quả rất hạn chế nếu như không các ảnh đầu vào không có được sự đồng nhất cao về màu sắc, góc quay theo phương ngang của camera.

Định hướng phát triển

Tối ưu các kỹ thuật trong từng bước của đề tài nhằm cải tiến thời gian tính toán và đưa ra kết quả chính xác hơn và ứng dụng vào các bài toán tìm kiếm, so sánh, theo dõi...

Tìm hiểu, nghiên cứu và sử dụng các phương pháp, kỹ thuật tốt hơn nhằm tối ưu, hoàn thiện bài toán tạo ảnh Panorama.

Triển khai ứng dụng tạo ảnh Panorama lên các thiết bị chụp hình kỹ thuật số chạy các hệ điều hành Android, iOS hay Windows Phone.

TÀI LIỆU THAM KHẢO

Tiếng Anh

- [1] L. Kitchen and A. Rosenfeld (1982), “Gray - Level corner detection. Pattern Recognition Letters”, pp. 95-102.
- [2] H. P. Moravec (1979), “Visual mapping by a robot rover”, *Proc of the 6th International Joint Conference on Artificial Intelligent*, pp. 589-600,
- [3] H. P. Moravec (1977), “Towards automatic visual obstacle avoidance”, *Proc of the International Joint Conference on Artificial Intelligent*, pp. 584.
- [4] C. Harris and M. Stephens (1988), “A combined corner and edge detector”, *Fourth Alvey Vision Conference*, pp. 147-151.
- [5] C. Schmid, R. Mohr, and C. Bauckhage (2000). “Evaluation of interest point detectors”, *International Journal of Computer Vision*, pp.151–172.
- [6] Thomas Melzer (2004), “SVD and its Application to Generalized Eigenvalue problems”, pp. 1-15.
- [7] William. H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery Numerical (1988-1992), “The Art of Scientific computing Second Edition”, *Cambridge University Press*, chapter 15.
- [8] R. Hartley and A. Zisserman (2003), “Multiple View Geometry in Computer Vision”, *Cambridge University Press, second edition*.
- [9] Elan Dubrofsky (2007), “Homography Estimation”, *Carleton University*.
- [10] M.A. Fischler and R.C. Bolles (1981), “Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography”, *Communications of the ACM*, pp.381–395.
- [11] Zhengyou Zhang, Rachid Deriche, Olivier Faugeras and Quang Tuan Luong (2004), “A Robust technique for matching two uncalibrated Images through the Recovery of the Unknown Epipolar Geometry”.
- [12] David G. Lowe (2004), “Distinctive Image Features from Scale-Invariant Keypoints”, *International Journal of Computer Vision*.
- [13] P. Viola, M. Jones (2001), “Rapid object detection using a boosted cascade of simple features”, *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.
- [14] P. Y. Simard, L. Bottou, P. Haffner, Y. Lecun (1998), "A fast convolution algorithm for signal processing and neural networks", *In Proceedings of the 1998 conference on Advances in neural information processing systems II*.
- [15] H. Bay, A. Ess, T. Tuytelaars, L.V. Gool (2008), “SURF - Speeded Up Robust Features”, *Computer Vision and Image Understanding (CVIU)*, Vol. 110, No. 3, pp. 346–359.

-
- [16] Robert Laganière (2011), “OpenCV 2 Computer Vision Application Programming Cookbook”, chapter 8-9.
 - [17] Tony Lindeber (1994), “Scale-space theory: a basic tool for analyzing structures at different scales”.
 - [18] Rong Zhang, “Automatic Computation of a Homography by RANSAC Algorithm”.
 - [19] http://en.wikipedia.org/wiki/Blob_detection#Lindeberg.27s_watershed-based_grey-level_blob_detection_algorithm
 - [20] <http://en.wikipedia.org/wiki/Panorama>
 - [21] http://en.wikipedia.org/wiki/Panorama_Mesdag
 - [22] http://en.wikipedia.org/wiki/Corner_detection
 - [23] <http://en.wikipedia.org/wiki/Homography>
 - [24] <http://en.wikipedia.org/wiki/RANSAC>
 - [25] [http://en.wikipedia.org/wiki/Feature_detection_\(computer_vision\)](http://en.wikipedia.org/wiki/Feature_detection_(computer_vision))
 - [26] <http://docs.opencv.org/>
 - [27] http://en.wikipedia.org/wiki/Interest_point_detection
 - [28] http://en.wikipedia.org/wiki/Brute_force