**Báo cáo đồ án**

PANORAMA STITCHING

ĐẠI HỌC QUỐC GIA TP HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN



Môn: Nhập môn Thị giác Máy tính

Lớp: KHTN2017

Mã lớp: CS231.K11.KHTN

GVHD: TS. Ngô Đức Thành

Lê Quốc Thịnh

17521087

Hà Quốc Tiến

17521122

Vũ Đình Vi Nghiệm

17520805

MỤC LỤC

[1. GIỚI THIỆU 1](#_Toc31134195)

[2. XÂY DỰNG CHƯƠNG TRÌNH 1](#_Toc31134196)

[2.1. Trích xuất đặc trưng ảnh 1](#_Toc31134197)

[2.2. So khớp đặc trưng 1](#_Toc31134198)

[2.3. Cân chỉnh hình ảnh 2](#_Toc31134199)

[2.4. Khớp các ảnh theo ước lượng Homography để tạo ảnh panorama 3](#_Toc31134200)

[3. ĐÁNH GIÁ 4](#_Toc31134201)

[3.1. Ưu điểm 4](#_Toc31134202)

[3.2. Nhược điểm 5](#_Toc31134203)

[5. TÀI LIỆU THAM KHẢO 5](#_Toc31134204)

DANH MỤC HÌNH ẢNH

[Hình 1. Kết quả của so khớp đặc trưng 2](#_Toc31134189)

[Hình 2. Trước và sau khi cân chỉnh hình ảnh 3](#_Toc31134190)

[Hình 3. Ảnh sau khi được "dán" 3](#_Toc31134191)

[Hình 4. Ảnh panorama hoàn chỉnh 4](#_Toc31134192)

TP. HCM, ngày 25 tháng 01 năm 2020

# GIỚI THIỆU

Ứng dụng Panorama Stitching cho phép người dùng tạo ra ảnh Panorama – ảnh toàn cảnh – bằng cách ghép các ảnh rời rạc lại với nhau. Ảnh Panorama cung cấp một góc nhìn rộng và bao quát hơn so với các ảnh chụp thông thường khác.

Đầu vào của ứng dụng là một đường dẫn thư mục chứa các hình cần ghép. Các hình này phải đảm bảo thuộc cùng một khung cảnh và có phần chồng/gối lên nhau. Đầu ra của ứng dụng là ảnh Panorama của các hình tương ứng.

# XÂY DỰNG CHƯƠNG TRÌNH

## Trích xuất đặc trưng ảnh

Đặc trưng tốt là đặc trưng có khả năng bất biến với các phép biến đổi hình học, cường độ ánh sáng; có tính phân biệt cao; và mang độ lợi về thông tin lớn.

Chương trình sử dụng đặc trưng ORB: đầu tiên, ORB dùng FAST để tìm đặc trưng, sau đó, mỗi đặc trưng sẽ được mô tả bởi BRIEF. So với SIFT, ORB tuy còn hạn chế khi góc xoay thay đổi lớn, nhưng lại đạt hiệu suất gần như tương đương, với thời gian tính toán nhanh hơn hẳn. Mặt khác, ORB được sử dụng tự do mà không vướng phải vấn đề bản quyền [[1]](#_TÀI_LIỆU_THAM). ORB sử dụng bộ mô tả 256 bit nhị phân.

## So khớp đặc trưng

Những đặc trưng được gọi là khớp nhau nếu các bộ mô tả của chúng có độ sai khác nhỏ. Độ sai khác được xác định dựa trên độ đo SSE:

Trong đó:

* : lần lượt là phần tử thứ của đặc trưng .
* : số chiều của bộ mô tả.
* càng nhỏ, cặp khớp nhau càng tốt.

Trên thực tế, hai đặc trưng dù có độ sai khác nhỏ cũng chưa chắc là một cặp khớp nhau tốt. Bởi lẽ, nếu với một đặc trưng *a* thuộc ảnh *A* tồn tại nhiều đặc trưng *b* thuộc ảnh *B* có độ sai khác so với *a* đều thấp, nhưng lại xấp xỉ nhau, thì khó mà chắc chắn được đặc trưng nào là khớp với *a* nhất (tình trạng nhập nhằng). Do đó, cần sử dụng hướng tiếp cận khác để khắc phục vấn đề này [[2]:](#_TÀI_LIỆU_THAM)

Trong đó:

* : đặc trưng có độ sai khác nhỏ nhất so với .
* : đặc trưng có độ sai khác nhỏ thứ hai so với .
* càng nhỏ, cặp khớp nhau càng tốt.

Việc so khớp cung cấp một tập hợp các cặp đặc trưng khớp nhau, đảm bảo nhỏ hơn ngưỡng cho trước (thường là 0.7 – 0.8).



Hình 1. Kết quả của so khớp đặc trưng

## Cân chỉnh hình ảnh

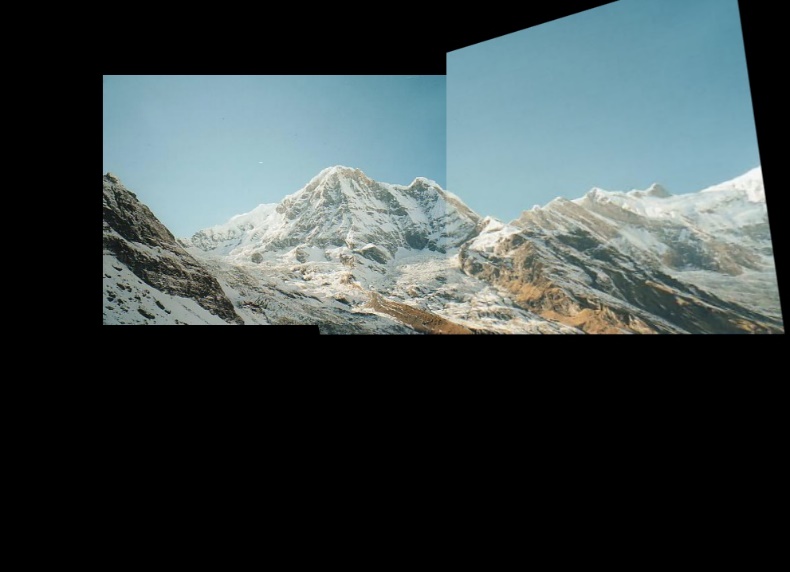
Bất kỳ hai hình ảnh của cùng một khung cảnh nào trong mặt phẳng hai chiều đều liên quan với nhau bằng một phép chuyển đổi Homography (cân chỉnh hình ảnh) [[3]](#_TÀI_LIỆU_THAM). Để thực hiện điều này, ta cần tìm ma trận Homigraphy *H* bằng thuật toán RANSAC với ý tưởng chính: với mỗi mô hình (ma trận) được xây dựng từ 4 cặp khớp nhau bất kì, đếm số cặp phù hợp ứng với từng mô hình và chọn mô hình có số cặp phù hợp nhiều nhất.

Hình 2. Trước và sau khi cân chỉnh hình ảnh

## Khớp các ảnh theo ước lượng Homography để tạo ảnh panorama

Ảnh trái sau khi được cân chỉnh (tạm gọi là ảnh *train*) sẽ được nối vào ảnh phải (ảnh *query*), tạo thành ảnh mới (ảnh *result*).



Hình 3. Minh họa việc nối ảnh

Để tránh việc xuất hiện các phần màu đen ở ranh giới giữa *train* và *query*, cần loại bỏ nền đen ở ảnh *query* trước thao tác nối. Việc này có thể được thực hiện bằng cách sử dụng một kernel có kích thước 5x5 với các phần tử đều là 1, lướt trên ảnh *query*. Các pixel trên ảnh *query* chỉ được dán vào *result* khi và chỉ khi tất cả pixel trong vùng này đều không phải là màu đen. Việc này có thể làm mất mát một phần nhỏ rìa của ảnh (không đáng kể).

Tiếp theo, để loại bỏ các viền đen thừa ở rìa của *result*, ta tìm bao (bounding box) nhỏ nhất sao cho bao này có thể chứa toàn bộ ảnh, không kể nền đen. Ảnh *result* sẽ được cắt tỉa theo bao này để hạn chế sự xuất hiện của nền đen.



Hình 4. Ảnh panorama hoàn chỉnh

Để tạo ảnh panorama từ nhiều hơn hai ảnh, cần chọn ra ảnh trái nhất, sau đó mới lần lượt tìm các ảnh trái thứ hai, thứ 3…, ảnh phải nhất, rồi lần lượt ghép các ảnh này với nhau từ giữa sang hai bên. Các bước thực hiện:

* Bước 1: Với mỗi ảnh thuộc *I* (tập ảnh ban đầu), đếm tổng số đặc trưng khớp giữa nửa trái của ảnh với các nửa phải của các ảnh còn lại.
* Bước 2: Chọn ra ảnh có tổng số đặc trưng khớp nhỏ nhất làm ảnh trái nhất. Cho ảnh này vào tập *O* đang rỗng (tập chứa các ảnh được sắp xếp theo thứ tự từ trái qua phải của khung cảnh) và loại nó khỏi *I*. Gán *ith\_leftmost­* bằng ảnh trái nhất.
* Bước 3: Chọn ảnh có tổng số đặc trưng khớp với *ith\_leftmost­* nhiều nhất làm *ith\_leftmost­.* Đẩy ảnh này vào cuối tập *O* và xóa nó khỏi *I.*
* Bước 4: Lặp lại bước 3 cho đến khi *I* rỗng.
* Bước 5: Chọn ảnh giữa (ảnh có vị trí đứng giữa tập *O*) làm *result.*
* Bước 6: Với mỗi ảnh từ vị trí giữa + 1 cho đến cuối tập *O*, gán result bằng ảnh được nối giữa *result* với lần lượt các ảnh trên (giống như nối 2 ảnh).
* Bước 7: Xoay ngược *result* theo chiều ngang.
* Bước 8: Với mỗi ảnh từ vị trí giữa – 1 cho đến đầu tập *O*, gán result bằng ảnh được nối giữa *result* với lần lượt ảnh xoay ngược của các ảnh trên theo chiều ngang.
* Bước 9: Xoay ngược *result* theo chiều ngang thêm một lần nữa, đây là kết quả cuối cùng.

# ĐÁNH GIÁ

## Ưu điểm

* Có khả năng ghép các ảnh không theo thứ tự.
* Giao diện đơn giản.
* Thời gian thực thi nhanh chóng.
* Kết quả tương đối chính xác, thuận mắt, hợp tỉ lệ.

## Nhược điểm

* Xảy ra lỗi khi thư mục có chứa các loại tệp không phải là hình ảnh.
* Gặp khó khăn khi phần gối nhau giữa các ảnh nhỏ hoặc không rõ ràng.
* Phần nền đen không được loại bỏ triệt để.
* Không có chức năng trộn màu cho ảnh (blend).
* Đôi khi có thể cho ra kết quả xấu khi số ảnh đầu vào nhiều hơn 2.

# TÀI LIỆU THAM KHẢO

[1] P. T. H. Anh, “Kỹ thuật Image Alignment sử dụng phương pháp feature-based trong bài toán nhận diện ký tự OCR,” 2019. [Online]. Available: <https://viblo.asia/p/ky-thuat-image-alignment-su-dung-phuong-phap-feature-based-trong-bai-toan-nhan-dien-ky-tu-ocr-bJzKmyODK9N>.

[2] “Image stitching with OpenCV and Python,” 2019. [Online]. Available: <https://medium.com/pylessons/image-stitching-with-opencv-and-python-1ebd9e0a6d78>.

[3] S. Mallick, “Homography Examples using OpenCV ( Python / C ++ ),” 2016. [Online]. Available: <https://www.learnopencv.com/homography-examples-using-opencv-python-c/>.