

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**  
**KHOA KHOA HỌC MÁY TÍNH**



**BÁO CÁO MÔN HỌC**  
**NHẬP MÔN THỊ GIÁC MÁY TÍNH**

**PANORAMA**

**Giảng Viên Hướng Dẫn:      NGÔ ĐỨC THÀNH**

**Thành Viên:**

- 1. Nguyễn Quốc Đạt – 14520146**
- 2. Nguyễn Văn Trưa – 15520934**
- 3. Võ Văn Lộc - 16521721**

*TP. Hồ Chí Minh, tháng 7 năm 2019*



## LỜI CẢM ƠN

Trân trọng gửi lời cảm ơn đến Thầy Ngô Đức Thành đã truyền đạt những kiến thức cơ bản cần thiết để nhóm có thể hoàn thành đồ án này.

Cũng xin cảm ơn thầy cô trong khoa Khoa học Máy tính đã nhiệt tình hỗ trợ, tạo điều kiện nhóm làm đồ án này.

## NHẬN XÉT CỦA GIÁO VIÊN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

## MỤC LỤC

I.	Thư viện và kỹ thuật: .....	4
1.	Thư viện: .....	4
2.	Kỹ thuật: .....	4
II.	Phương pháp thực hiện .....	5
III.	Đánh giá và phân tích hệ thống .....	7
1.	Chạy thử: .....	7
2.	Đánh giá: .....	9
IV.	Các tài liệu tham khảo .....	10

# I. Thư viện và kỹ thuật:

## 1. Thư viện:

- Nhóm em có sử dụng 3 thư viện chính:
  - o [Numpy](#): Dùng để tính toán ma trận vì hình ảnh trên máy tính thực chất chỉ là những ma trận số. Tính toán các phép biến đổi (transformation) và tìm mặt nạ ảnh (mask).
  - o [Scikit-image](#) (skimage): Vì [OpenCV](#) hỗ trợ gần như tự động các bước để ghép ảnh nên nhóm em đã quyết định sử dụng scikit-image để thực hiện nhằm hiểu rõ vấn đề hơn cũng như thử sức với một thư viện mới.
  - o [Flask](#): Thư viện hỗ trợ giúp làm web trở nên dễ dàng hơn.

## 2. Kỹ thuật:

### a. Tìm điểm hứng thú (keypoints) và rút trích đặc trưng: *Oriented FAST and rotated BRIEF (ORB)*

- Sau khi tham khảo một vài tài liệu ([3], [4]) nhóm nhận thấy có thể sử dụng thuật toán ORB để rút trích đặc trưng trong ảnh vì nó cho tốc độ tính toán nhanh và có kết quả khá tốt trong nhiều trường hợp được thử nghiệm.
- Vì SIFT và SURF đã được đăng kí bằng sáng chế và giới hạn phạm vi sử dụng nên thư viện scikit-image không hỗ trợ (Lý do chính, khổ thế thầy ạ).

### b. Loại bỏ điểm dị biệt (outlier) và tìm model ước lượng phép biến đổi: *Random sample consensus (RANSAC)*

- Việc so khớp các điểm hứng thú giữa các bức ảnh sẽ gặp vấn đề so khớp lỗi, ta dùng RANSAC để loại bỏ các trường hợp này và để tìm phép biến đổi tương ứng.
- Vì tính ngẫu nhiên của thuật toán mà model tìm được sẽ có sự khác biệt tương đối cho cùng một bộ ảnh với các lần chạy khác nhau.

### c. Blending:

- Hiện tại nhóm em chỉ tìm hiểu được một vài kỹ thuật blending như:
  - o Feathering
  - o Pyramid blending
  - o Poisson blending
- Tuy nhiên do khả năng cũng như giới hạn thời gian nên nhóm chưa kịp hiện thực phần này. Nếu lựa chọn, nhóm em sẽ chọn kỹ thuật *Poisson blending* vì kỹ thuật này dựa trên việc thay đổi gradient ở những vùng tiếp giáp, cho kết quả tốt vì các bức ảnh ghép panorama thường có màu sắc khá giống nhau.

## II. Phương pháp thực hiện

- Sau khi tham khảo nhiều tài liệu thì nhóm em chọn làm theo hướng dẫn của của MATLAB mà thầy đã giới thiệu [1] vì bài này hướng dẫn chi tiết từng bước thực hiện và đây là phương pháp áp dụng cho nhiều ảnh mà nhóm em tìm mãi cũng không có tài liệu nào ngon lành bằng.
- Bên cạnh đó nhóm em cũng tham khảo cách sử dụng skimage để ghép 3 ảnh từ demo của chính nhóm phát triển [2]
- Vì giới hạn độ dài của báo cáo, các bước thực hiện đã được nêu chi tiết trong các tài liệu tham khảo cũng như thầy đã biết rõ cách làm nên nhóm em xin chỉ tóm tắt các bước chính.

1. Đọc ảnh và chuyển về ảnh xám .

2. Dùng ORB để rút trích đặc trưng trong từng bức ảnh. Nhóm cấu hình với tham số là 1000 keypoints. Bước này rất may là thư viện skimage đã làm tự động.

3. Tìm ma trận biến đổi homography giữa các bức ảnh:

- o Phần làm cho việc ghép 2 bức ảnh khác với ghép nhiều bức ảnh là ma trận biến đổi sẽ có tính tích lũy. Ma trận biến đổi ảnh  $I(n-1)$  thành  $I(n)$  là  $T(n) * T(n-1) * \dots * T(0)$
- o Nhóm em hiện tại chỉ tìm hiểu được đến phần ghép ảnh có thứ tự.
- o Vì là ảnh có thứ tự nên ta giả định 2 bức ảnh gần nhau sẽ có một phần chung để từ đó tìm ra được ma trận biến đổi ảnh này thành ảnh kia.
- o Việc tìm ma trận biến đổi này chỉ là ước lượng nhờ thuật toán RANSAC nên kết quả mỗi lần chạy sẽ không giống nhau và có lúc sẽ ra hình ảnh không như mong đợi.

4. Tìm kích thước ảnh sau khi ghép thành panorama:

- o Áp dụng ma trận đã tìm được phía trên vào các bức ảnh tương ứng để biến đổi góc nhìn, làm cho các bức ảnh khớp với nhau hơn.
- o Do việc chọn bức ảnh đầu tiên làm gốc để ghép các bức ảnh khác vào sẽ gây ra hiện tượng méo mó và làm ảnh sau khi ghép trông quái dị nên ta giải quyết bằng cách:
  - Tìm ma trận biến đổi của ảnh ở trung tâm của khung hình panorama.
  - Đảo ngược ma trận này và nhân nó với tất cả các ma trận còn lại.
- o Ví dụ:



Panorama sử dụng ảnh đầu để ghép.



Panorama sử dụng ảnh ở giữa khung hình để ghép các ảnh khác vào.

5. Tìm mặt nạ (mask) để ghép các ảnh lại với nhau: Tìm phần trùng nhau giữa 2 bức ảnh liên tiếp nhau, cắt phần ảnh phía bên trái gây ra hiện tượng trùng, việc này loại bỏ hiện tượng hình ảnh bị nhòe khi 2 bức ảnh gắn lại nhưng không khớp (do model không đúng).
6. Ghép các ảnh lại theo mặt nạ đã tìm được để tạo panorama hoàn chỉnh.



### III. Đánh giá và phân tích hệ thống

#### 1. Chạy thử:

a. Các ví dụ chạy tốt:

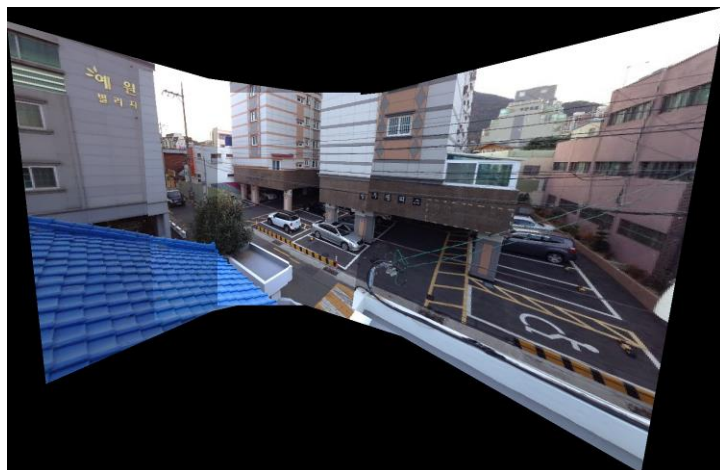
- Bộ 1:



Kết quả:



- Bộ 2:



- Bộ 3: Ảnh dọc

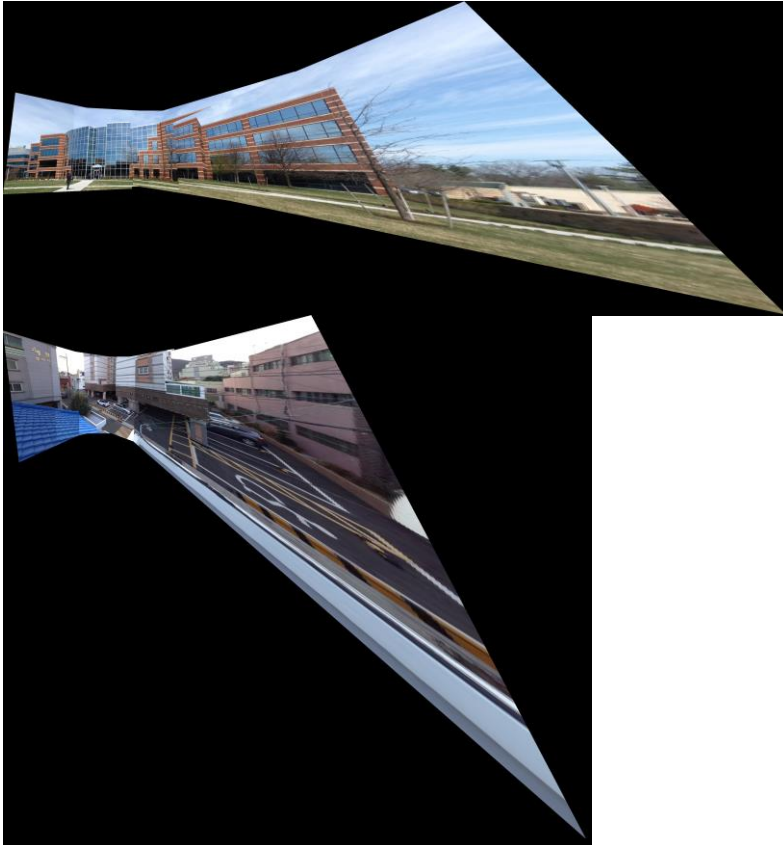


- Bộ 4:





b. Kết quả chạy chưa tốt:



## 2. Đánh giá:

- Phần mềm hoạt động chưa được ổn định lắm, có nhiều trường hợp ra ảnh tốt hơn mong đợi. Việc này nhóm em nghĩ là do khi so khớp các điểm ảnh và ước lượng ma trận biến đổi gây nên (thuật toán RANSAC có tính ngẫu nhiên). Trong ví dụ của MATLAB dùng thuật toán M-estimator Sample Consensus (MSAC) trong hàm `estimateGeometricTransform` cũng có tính ngẫu nhiên nhưng vẫn cho model giống nhau sau nhiều lần chạy. Em không hiểu lắm chỗ này.
- Vì model là ước lượng nên các phép biến đổi không chính xác hoàn toàn, ảnh panorama khi ghép xong không khớp hoàn toàn với nhau.
- Do các bức ảnh được chụp bằng thấu kính hình trụ hoặc hình cầu nhưng dùng phép biến đổi planar homography nên bị biến dạng theo chiều dọc (bộ ảnh 2, 4) [5]. Giải quyết bằng cách dùng phép chiếu hình trụ hoặc hình cầu. Phần này nhóm em không có khả năng thực hiện.
- Vì không blending nên ảnh panorama xuất ra sẽ không có màu đồng nhất. Phần này như đã trình bày phía trên, nhóm em chỉ vừa kịp tìm hiểu chứ chưa hiện thực được.

## IV. Các tài liệu tham khảo

- [1]. <https://www.mathworks.com/help/vision/examples/feature-based-panoramic-image-stitching.html>
- [2]. [https://github.com/scikit-image/skimage-tutorials/blob/5094a14bdc4de2b46be4c5e5a996a8dd4f71379b/lectures/solutions/adv3\\_panorama-stitching-solution.ipynb](https://github.com/scikit-image/skimage-tutorials/blob/5094a14bdc4de2b46be4c5e5a996a8dd4f71379b/lectures/solutions/adv3_panorama-stitching-solution.ipynb)
- [3]. [Karami, Ebrahim, Siva Prasad, and Mohamed Shehata. "Image matching using SIFT, SURF, BRIEF and ORB: performance comparison for distorted images." arXiv preprint arXiv:1710.02726 \(2017\)](#)
- [4]. [Tareen, S. A. K., & Saleem, Z. \(2018\). A comparative analysis of SIFT, SURF, KAZE, AKAZE, ORB, and BRISK. 2018 International Conference on Computing, Mathematics and Engineering Technologies \(iCoMET\). doi:10.1109/icomet.2018.8346440](#)
- [5]. <http://ppwwyyxx.com/2016/How-to-Write-a-Panorama-Stitcher/#Extrema-Detection>