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

BÁO CÁO MÔN HỌC

NHẬP MÔN THỊ GIÁC MÁY TÍNH

PANORAMA

GVHD : Nguyễn Vĩnh Tiệp

Lớp: CS321.K11

Thành viên:

Nguyễn Đình Vinh- 16521582

Nguyễn Đắc Phi Hùng 16521688

Phạm Thị Nga 16521743

TP. Hồ Chí Minh, tháng 1 năm 2020

MỤC LỤC

I. Thư viện và kỹ thuật :……..……Trang 2

1. Thư viện :…...……Trang 2

2. Kỹ thuật :…...…….Trang 2

II. Phương pháp thực hiện :……..Trang 3

III. Đánh giá và phân tích hệ thống :…….....Trang 8

1. Chạy thử :……...……Trang 8

2. Đánh giá :…….……Trang 11

3. Cải tiến :……...Trang 11

IV. Tài liệu tham khảo :……....Trang12

I. Thư viện và kĩ thuật

1. **Thư viện**:

- Nhóm em sử dụng 3 thư viện chính:
  - Numpy: Dùng để tính toán ma trận hình ảnh. Tìm mặt nạ ảnh (mask).
  - OpenCV: Thư viện chính sử dụng trong đồ án này, dùng để rút trích đặc trưng, tính ma trận homography, biến đổi ảnh cơ bản như chuyển ảnh xám ,flip, warp…
  - Flask: Thư viện hỗ trợ giúp làm web dễ dàng hơn.

2. **Kỹ thuật:**

a) **Tìm điểm hứng thú (keypoints) và rút trích đặc trưng:** Sau khi tham khảo một số tài liệu nhóm em quyết định xây dựng cả 3 thuật toán là SIFT, SURF, ORB để rút trích đặc trưng. Từ thực nghiệm rút ra ưu điểm và nhược điểm của từng thuật toán

b) **Ghép cặp các điểm tương đồng:** Sử dụng thuật toán FLANN (Fast Library for Approximate Nearest Neighbors) để tìm ra những cặp đặc trưng có khoảng cách Euclid là nhỏ nhất, kết hợp với thuật toán RANSAC(RANdom SAmple Consensus) để loại bỏ những cặp bị nhiễu

c) **Tính toán ma trận Homography và ghép 2 bức ảnh lại với nhau:** OpenCv đã hỗ trợ sẵn hàm tìm homography và ghép 2 ảnh lại với nhau tuy nhiên ảnh ghép xong sẽ bị cắn xén 1 phần, chúng em quyết định áp dụng 1 phép biến đổi translation lên để lấy toàn bộ ảnh

d) **Blending:** Dựa trên thuật toán Alpha Blending vì nó đơn giản dễ thực hiện và có độ thẩm mĩ tương đối tốt

e) **Cropping:** cắt bỏ những phần màu đen dư thừa dựa 4 góc (top-left, bottom-left,bottom-right,top-right) của 2 tấm hình

II. Phương pháp thực hiện

1. Đọc ảnh và chuyển về ảnh xám
2. Dùng ORB để rút trích đặc trưng, mặc định chúng em để ORB với nfeature=3000, việc thực hiện khá đơn giản với OpenCV

```
orb = cv2.ORB_create(nfeatures=3000)
keypoints, features = orb.detectAndCompute(grayImage, None)
```

3. Ghép các điểm tương đồng, sử dụng k-NN với k=2 để chọn ra 2 kết quả tốt nhất, sau đó áp dụng Lowe's ratio = 0.75 để loại bỏ các kết quả không hợp lệ

```
featureMatcher = cv2.DescriptorMatcher_create("FlannBased")
matches = featureMatcher.knnMatch(featuresA,featuresB, k=2)
good = []
for m,n in matches:
    if m.distance<ratio*n.distance:
        good.append(m)
```

4. Tính toán ma trận homography và áp dụng RANSAC để loại nhiễu, đơn giản chỉ cần lấy toạ độ các điểm src_points và dst_points trong các kết quả hợp lệ ở bước 3 rồi dùng hàm findHomography.

```
H,_= cv2.findHomography(src_points, dst_points, cv2.RANSAC,5.0)
```

Áp dụng Homography lên src_img để nó xoay về đúng hướng với dst_img

```
Warp_src_img= cv2.warpPersective(src_img, H,(width,hight)
```

Tuy nhiên ảnh sau khi xoay bị cắn xén 1 phần rất lớn so với ảnh gốc bên dưới

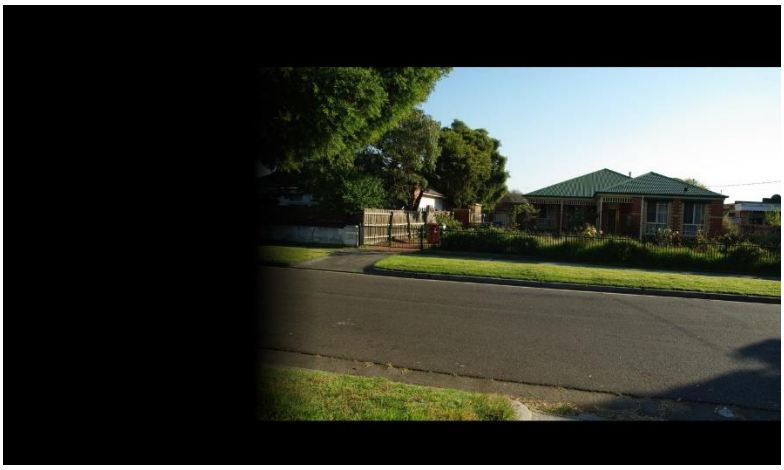
Để giải quyết vấn đề này chúng em áp dụng 1 phép translation lên ma trận homography. Ý tưởng tham khảo tại đây

Kết quả sau khi áp dụng translation lên homography

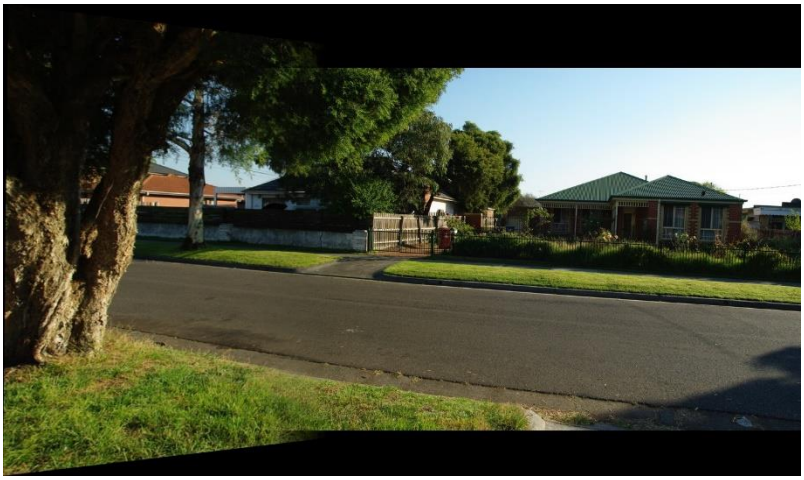
Sau đó chuyển ảnh làm mốc dst_img về đúng size như ảnh trên và cộng

Từng pixel 2 ảnh này lại ta thu được ảnh như bên dưới

5. Blending: Áp dụng tư tưởng của alpha blending, chúng em tạo mask cho từng ảnh rồi nhân nó với ma trận ảnh tương ứng.



Cộng từng pixel 2 ảnh ở trên ta thu được ảnh đã blending



6. Cropping: Sau khi đã ghép 2 ảnh và blending nó, chúng em tiến hành cắt bỏ những phần màu đen, việc cắt bỏ khá đơn giản dựa vào toạ độ 4 điểm góc của 2 tấm ảnh ban đầu, áp dụng kĩ thuật slicing để chọn vùng ảnh.

7. Multi stitching: Để ghép nhiều ảnh lại với nhau, chúng em chia list ảnh ban đầu thành 2 list con là left và right. Lấy bức ảnh ở giữa làm mốc (dst_img) rồi đi ghép với từng ảnh mỗi list. Mỗi lần ghép thì pop 2 ảnh ra rồi push ảnh kết quả ghép được vào. Đến khi còn 1 ảnh thì dừng. Cuối cùng thu được left-panorama và right-panorama đem 2 ảnh này ghép với nhau thu được ảnh hoàn chỉnh.

III. Đánh giá và phân tích hệ thống

1) Chạy thử

- Các kết quả tốt, phần điểm chung mỗi ảnh trên 15% và góc chụp không lệch quá nhiều, điều kiện ánh sáng như nhau.

  - Bộ 1

Kết quả:

  - Bộ 2

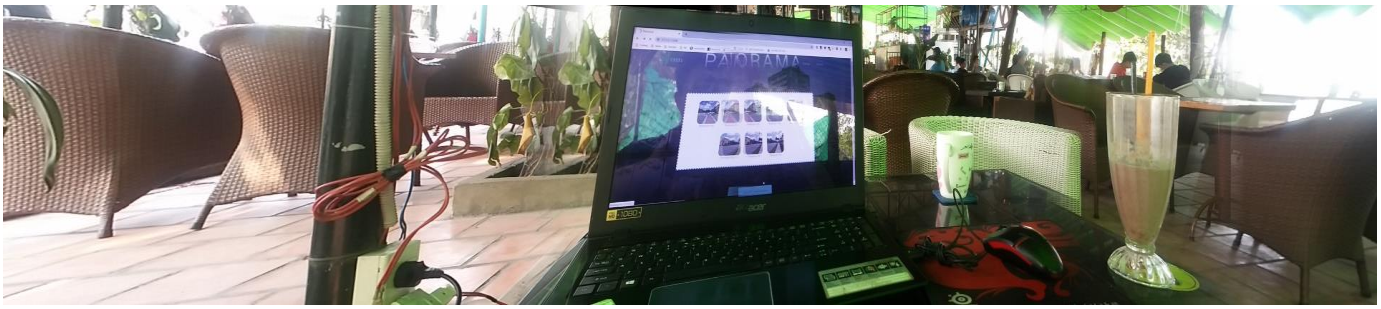
Kết quả:

- Bộ 3:

Kết quả:

- Bộ 4

Kết quả:



- **Các kết quả chưa tốt, góc chụp lệch nhiều, điểm chung ít từ 10~15%, điều kiện chiếu sáng khác nhau**
    - Bộ 1:

Kết quả:

2) Đánh giá

- Phần mềm chưa thực chạy ổn định lắm, phụ thuộc vào cấu hình máy tính của mỗi người, phiên bản python và opencv (chạy ổn định trên python 3.7.1 và opencv 3.4.2)
- Sử dụng SURF cho kết quả ổn định nhất, những trường hợp ảnh có điểm chung dao động từ 10~15% thì SURF chạy được còn ORB thì thi thoảng không chạy được
- ORB cho kết quả chạy nhanh nhất, trong một số trường hợp phải set nfeature tăng hoặc giảm thì mới ghép được
- SURF và SIFT chạy khá chậm, đặc biệt nếu ảnh có độ phân giải lớn sẽ gây hiện tượng giật lag máy
- Phần ghép nhiều ảnh chưa tối ưu vì lặp lại nhiều thao tác
- Ảnh đầu vào phải được chụp có thứ tự từ trái sang phải hoặc ngược lại
- Chỉ là ảnh góc rộng và bị kéo dãn ở 2 đầu → cách khắc phục là tạo ảnh 360

3. Cải tiến

-Sau buổi seminar trên lớp thì chúng em đã cải thiện lại giao diện Reponsive thích hợp trên các thiết bị khác nhau. Thêm một số điều kiện để có thể chạy được cụ thể: số lượng ảnh chọn để ghép phải lớn hơn 1, số cặp điểm tương đồng phải lớn hơn một ngưỡng nhất định…

-Thêm chức năng tạo ảnh panorama 360 từ hàm của openCV (cv2.createStitcher) cùng với các kĩ thuật để crop ảnh panorama 360 này

-Hàm tạo ảnh panorama 360 này có tốc độ xử lí nhanh hơn, ánh sáng, điều kiện chụp cũng như góc chụp có độ lệch lớn cũng xử lí khá tốt và số lượng ảnh đầu vào cũng nhiều hơn.

Panorama tạo bởi cv2.createStitcher (trên) so với hàm tự định nghĩa (dưới bị kéo dãn 2 đầu)

IV. Tài liệu tham khảo:

http://graphics.cs.cmu.edu/courses/15-463/2010_spring/Lectures/blending.pdf

https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/

https://kipalog.com/posts/Dung-RANSAC-de-loai-bo-nhieu-trong-mo-hinh

https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_feature2d/py_matcher/py_matcher.html