ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN

TRUY VẤN THÔNG TIN THỊ GIÁC

Lab 02

Giảng viên hướng dẫn Thầy Võ Hoài Việt
Thầy Nguyễn Trọng Việt
Thầy Phạm Minh Hoàng

Sinh viên thực hiện Võ Nguyễn Hoàng Kim
21127090

Phụ lục

I. Scale-Invariant Feature Transform

- Scale-Invariant Feature Transform (SIFT) là một thuật toán được sử dụng phổ biến trong lĩnh vực thị giác máy tính để tìm kiếm, trích xuất các đặc trưng trong hình ảnh. Các đặc trưng mà SIFT trích xuất được có ưu điểm là bền vững với sự thay đổi của các phép xoay, tỷ lệ và cũng như các phép biến đổi Affine.
- Các bước chính của thuật toán SIFT được trình bày tổng quan như sau:
    - Bước 1: Phát hiển các điểm đặc trưng (Keypoint Detection)
        - Sử dụng Gaussian Blur để làm mờ hình ảnh ở các mức độ khác nhau, tạo ra một chuỗi các hình ảnh bị làm mờ với các hệ số sigma khác nhau. Điều này giúp phát hiện các đặc trưng ở các tỉ lệ khác nhau.
        - Tạo ra các hình ảnh bằng cách trừ đi các hình ảnh đã được làm mờ (bằng Gaussian) ở các mức độ khác nhau.
        - Tìm kiếm các điểm cực trị trong không gian tỷ lệ bằng cách so sánh mỗi điểm ảnh với vùng lân cận 8 của nó (trong cùng một ảnh) và 9 điểm ảnh ở tỷ lệ tiếp theo và 9 điểm ảnh ở tỷ lệ trước đó. Điều này giúp cho đặc trưng đang xét đảm bảo rằng nó được thể hiện tốt nhất trong thang đo đó.
    - Bước 2: Lọc và chỉnh sửa các điểm đặc trưng (Keypoint Localization)
        - Loại bỏ các điểm yếu và không ổn định bằng cách sử dụng thuật toán Taylor để tính toán chính xác vị trí của các điểm đặc trưng và loại bỏ các điểm có độ tương phản thấp hoặc nằm trên các cạnh.
    - Bước 3: Ổn định dướng (Orientation Assignment)
        - Mỗi điểm đặc trưng được gán một hướng dựa trên hướng và độ lớn của gradient trong vùng lân cận của điểm đó. Điều này giúp các điểm đặc trưng trở nên không phụ thuộc vào xoay của hình ảnh.
    - Bước 4: Mô tả các điểm đặc trưng (Keypoint Descriptor)
        - Xung quanh mỗi điểm đặc trưng, một vùng lân cận được chia thành các ô nhỏ. Với mỗi ô, histogram của hướng gradient được tính toán. Các histogram này được kết hợp lại để tạo thành một vector đặc trưng (thường có kích thước 128).
    - Bước 5: So khớp các điểm đặc trưng (Keypoint Matching)
        - Sử dụng các phương pháp như Nearest Neighbor để so sánh các vector đặc trưng giữa các hình ảnh khác nhau.
        - Sử dụng tỷ lệ khoảng cách gần nhất và khoảng cách gần nhì để lọc ra các cặp đặc trưng tốt nhất. Thường thì tỷ lệ này phải nhỏ hơn một ngưỡng định sẵn (thường là 0.8).

II. Oriented FAST and Rotated BRIEF

- Oriented FAST and Rotated BRIEF (ORB) là một thuật toán phát hiện và mô tả đặc trưng hình ảnh nhanh và hiệu quả, được phát triển dựa trên sự kết hợp của các thuật toán FAST (Features from Accelerated Segment Test) và BRIEF (Binary Robust Independent Elementary Features).
- Các bước chính của thuật toán ORB được trình bày tổng quan như sau:
    - Bước 1: Phát hiển các điểm đặc trưng (Keypoint Detection)
        - ORB sử dụng thuật toán FAST để phát hiện các điểm dặc trưng có trong ảnh. FAST kiểm tra một pixel với các pixel xung quanh nó theo mẫu hình tròn và xác định xem đó có phải là một điểm đặc trưng hay không.
        - Sau khi phát hiện các điểm đặc trưng bằng FAST, ORB sử dụng Harris để đánh giá các điểm này và giữ lại các điểm có giá trị cao nhất.

- Bước 2: Chọn hướng cho các điểm đặc trưng (Orientation Assignment)
  - Để làm cho các đặc trưng không phụ thuộc vào xoay, ORB tính toán hướng của mỗi điểm đặc trưng. Điều này được thực hiện bằng cách sử dụng các moment trong một vùng lân cận của điểm đặc trưng.
  - Tiếp theo đó, hướng được tính toán và sau đó được gán cho điểm đặc trưng, giúp nó không phụ thuộc vào xoay của hình ảnh.
- Bước 3: Mô tả các điểm đặc trưng (Keypoint Descriptor)
  - ORB sử dụng thuật toán BRIEF để tạpn các mô tả đặc trưng (nhị phân) cho mỗi điểm đặc trưng. BRIEF tạo ra các mô tả bằng cách so sánh cường độ của các pixel trong một mẫu ngẫu nhiên xung quanh điểm đặc trưng.
  - Tuy nhiên ở đây, ORB cải tiến BRIEF bằng cách xoay các mẫu ngẫu nhiên theo hướng đã gán cho mỗi điểm đặc trưng, làm cho mô tả đặc trưng trở nên không phụ thuộc vào xoay của hình ảnh.
- Bước 4: So khớp các điểm đặc trưng (Keypoint Matching)
  - ORB so sánh các mô tả đặc trưng nhị phân bằng cách sử dụng khoảng cách Hamming, đếm số bit khác nhau giữa hai mô tả.
  - Các điểm đặc trưng từ các hình ảnh khác nhau được so sánh và ghép nối bằng cách tìm các điểm có khoảng cách Hamming nhỏ nhất.

III. Thực nghiệm

1. Xây dựng mã nguồn

- Mã nguồn được xây dựng theo kiến trúc hướng đối tượng (OOP) với lớp chính là Image. Trong lớp này, các thuộc tính và phương thức bao gồm:
  - Thuộc tính:
    - ***image:*** là ma trận ảnh.
  - Phương thức:
    - ***Constructor (Image):*** Khởi tạo đối tượng Image.
    - ***getImage***: được sử dụng lấy thuộc tính ***image*** trong lớp Image.
    - ***featureExtractionBySIFT***: được sử dụng để trích xuất đặc trưng trong ảnh bằng thuật toán SIFT
    - ***featureExtractionByORB***: được sử dụng để trích xuất đặc trưng trong ảnh bằng thuật toán ORB
- Chương trình được chia làm 3 file chính là **main.cpp**, **image.cpp** và **function.h**, trong đó
  - **main.cpp**: chứa mã nguồn chính để thực thi chương trình.
  - **image.cpp**: chứa các nội dung xây dựng các phương thức của đối tượng Image
  - **function.h**: chứa các khai báo thư viện cần thiết cũng như định nghĩa đối tượng Image.
- Chương trình được thực thi trên Terminal dưới lệnh sau:

  <ten_file_thuc_thi.exe> <inputFilePath><outputFilePath>

  Trong đó,

  - ten_file_thuc_thi.exe: là file thực thi
  - <inputFilePath>: là đường dẫn đến nơi lưu trữ ảnh được dùng để trích xuất đặc trưng
  - <outputFilePath>: là đường dẫn đến nơi lưu trữ ảnh kết quả sau khi trích xuất đặc trưng (bằng SIFT và ORB).

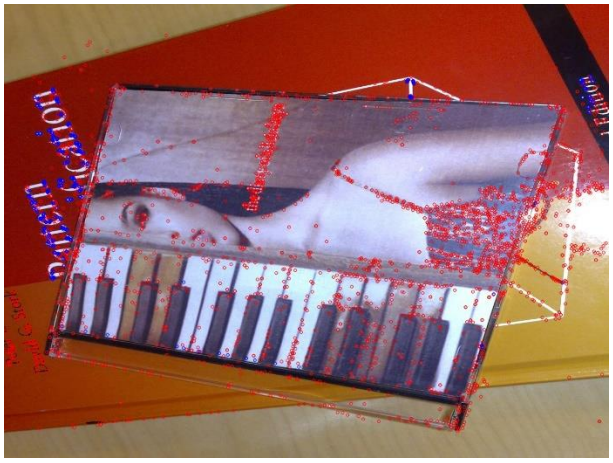
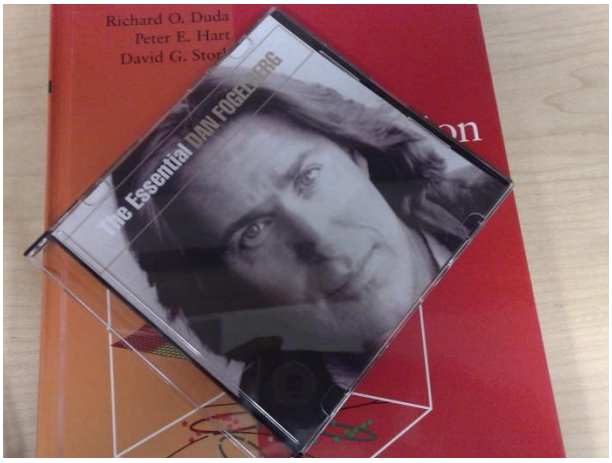
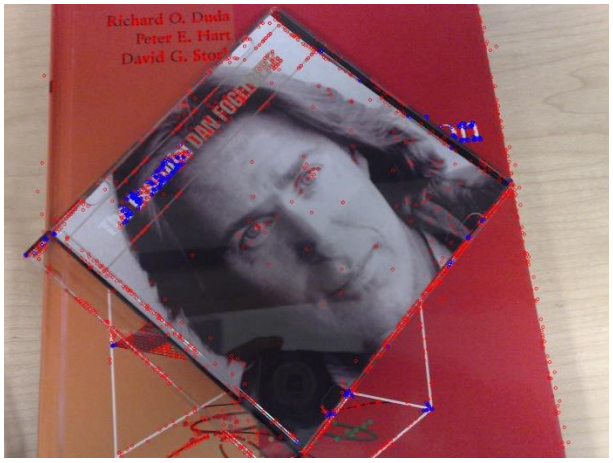
- Ví dụ, để chạy chương trình, ta tiến hành mở terminal, truy cập đến thư mục chứa mã nguồn, và thực hiện gọi lệnh:
    - Nếu ta muốn sử dụng dữ liệu từ webcam, ta sẽ thực thi lệnh sau

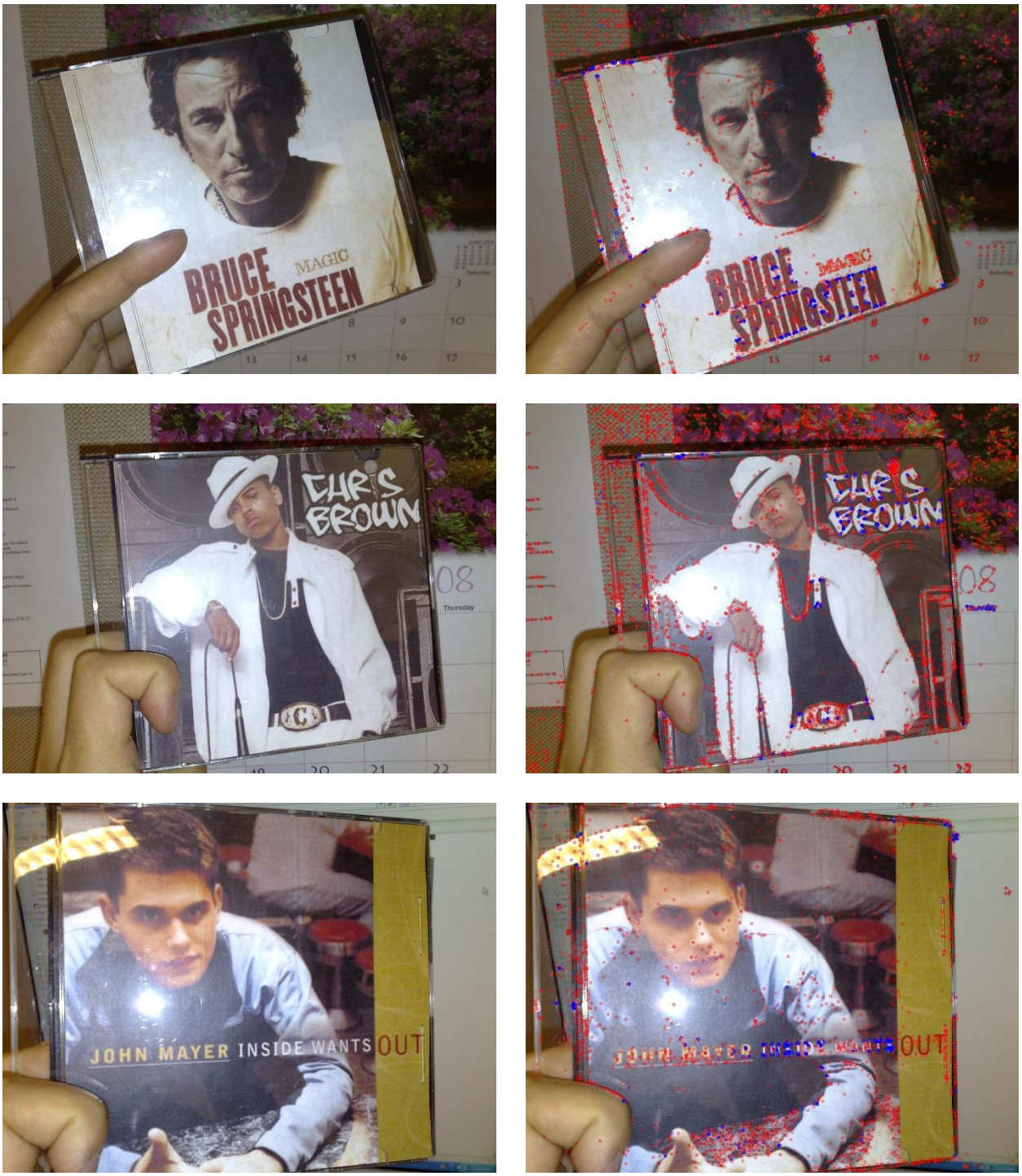
21127090_Lab02.exe ../Data/01.jpg ../Output/01_result.jpg

- Trong đó:
    - 21127090_Lab02.exe: là file thực thi chương trình
    - ../Data/01.jpg: là đường dẫn đến nơi lưu trữ hình ảnh 01.jpg
    - ../Output/01_result: là đường dẫn để lưu kết quả cho ảnh 01.jpg sau khi được trích xuất đặc trưng.

2. Kết quả

- Ảnh sau khi được trích xuất đặc trưng thể hiện các keypoints được tìm thấy bằng thuật toán SIFT (màu đỏ) và ORB (màu xanh). Dưới đây là 5 ảnh kết quả được lấy từ dữ liệu test CD (Trong mô tả bài tập cá nhân):

Ảnh ban đầu	**Ảnh sau khi trích xuất đặc trưng**
	
	



Nhận xét:

- Dựa trên kết quả thực nghiệm mà ta có được, các đặc trưng được trích xuất bằng SIFT (màu đỏ) và các đặc trưng được trích xuất bằng ORB (màu xanh), ta có được một số nhận xét như sau:
    - **Số lượng:** Số lượng đặc trưng mà SIFT trích xuất được nhiều hơn so với số lượng đặc trưng mà ORB trích xuất.

- **Phân bố:** Có thể thấy, các đặc trưng được trích xuất bởi SIFT phân bố đều đặn và rộng khắp hình ảnh, dù ở vùng nhiều hay ít chi tiết. Trong khi đó, cấc đặc trưng mà ORB trích xuất thường tập trung ở vùng có độ tương phản cao (như cạnh của vật thể hoặc vùng có sự thay đổi cường độ mạnh).

- Từ những nhận xét trên, ta có thể rút ra được một số kết luận hữu ích:
  - Với số lượng nhiều và độ phân bố cao, các đặc trưng của SIFT có chất lượng cao và ổn định trong các điều kiện biến đổi khác nhau như tỷ lệ (scale), phép xoay (rotation) cũng như các phép biến đổi Affine. Đồng thời, cũng nhờ vào điều này cho thấy rằng các đặc trưng trích xuất bởi SIFT có khả năng phân biệt tốt hơn giữa các chi tiết của hình ảnh.
  - Ngược lại với SIFT, các đặc trưng mà ORB trích xuất được thường ít nhạy cảm hơn với các biến đổi phức tạp. Đồng thời, số lượng đặc trưng mà ORB trích xuất được khá ít. Điều này phù hợp với lý thuyết của thuật toán ORB.

➔ Dù SIFT cung cấp nhiều đặc trưng và chất lượng của chúng cao, tuy nhiên sử dụng thuật toán SIFT thường khá chậm và tốn tài nguyên tính toán rất nhiều. Ngược lại, ORB tuy cung cấp lượng đặc trưng ít nhưng quá trình trích xuất nhanh hơn nhiều và ít tốn tài nguyên tính toán hơn.