

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN



Vision Information Retrieval
Individual Lab

Họ và tên: Huỳnh Hiệp Phát
MSSV: 21127130
Email: hhphat21@clc.fitus.edu.vn
SĐT: 0357691354

Mục lục

I.	Bảng tự đánh giá.....	3
II.	Giải thích các lớp và hàm	3
1.	Lớp FeatureDatabase.....	3
2.	FeatureExtraction	4
3.	Các hàm trong Int Main.....	4
III.	Các đặc trưng	4
1.	Color Histogram	4
2.	Color Correlogram.....	5
3.	SIFT.....	5
4.	ORB	6
IV.	Hướng dẫn truy vấn	7
1.	Tải dữ liệu	7
2.	Truy vấn.....	7
3.	Kết quả hiển thị.....	8
V.	Tổng kết	8
VI.	Tài liệu tham khảo	9

I. Bảng tự đánh giá

Yêu cầu	Nội dung	Mức độ hoàn thành	Ghi chú
Chuẩn bị dữ liệu	Rút trích các đặc trưng (Color Histogram, Color Correlogram, SIFT, ORB)	100%	
	Xử lý dữ liệu và lưu các đặc trưng vào cơ sở dữ liệu	100%	
Truy vấn ảnh	Cho phép người dùng truy vấn ảnh dựa vào các đặc trưng	100%	
Tính toán MAP	Sử dụng độ đo MAP để đánh giá độ chính xác kết quả truy vấn	100%	
Tính toán tốc độ tìm kiếm	Ghi rõ thời gian truy vấn trên giao diện của chương trình	100%	

II. Giải thích các lớp và hàm

1. Lớp FeatureDatabase

Mô tả: Lớp này quản lý việc lưu trữ và truy xuất các đặc trưng của các hình ảnh.

Các thành phần chính:

- **Các thuộc tính:**
 - **featureFolder:** Đường dẫn thư mục lưu trữ các tệp đặc trưng.
 - **imageFolder:** Đường dẫn thư mục lưu trữ các hình ảnh gốc.
 - **featureData:** Một unordered_map lưu trữ các đặc trưng của từng hình ảnh, được liên kết với tên của hình ảnh.
- **Các phương thức:**
 - **FeatureDatabase(const string& featureFolder, const string& imageFolder):** Constructor để khởi tạo đối tượng FeatureDatabase.
 - **void initializeDatabase():** Khởi tạo cơ sở dữ liệu bằng cách lưu các đặc trưng nếu chưa tồn tại, hoặc tải chúng nếu đã tồn tại.
 - **vector<pair<string, double>> queryImage(const cv::Mat& queryImage, const string& featureType, int k):** Truy vấn hình ảnh dựa trên đặc trưng và trả về một vector các cặp tên hình ảnh và độ tương đồng.
 - **void saveFeatures():** Lưu các đặc trưng của hình ảnh vào các tệp tin.
 - **void loadFeatures(const string& featureType):** Tải các đặc trưng từ các tệp tin lưu trữ vào featureData.
 - **void saveAdditionalFeatures(const string& additionalImageFolder):** Lưu các đặc trưng bổ sung của hình ảnh từ thư mục khác vào các tệp tin.
 - **double calculateHistSimilarity(const cv::Mat& hist1, const cv::Mat& hist2):** Tính toán độ tương đồng giữa hai histogram màu.

- **double calculateDescriptorSimilarity(const cv::Mat& desc1, const cv::Mat& desc2):** Tính toán độ tương đồng giữa hai mô tả ORB hoặc SIFT.

2. Feature Extraction

Mô tả: Tập hợp các hàm để trích xuất các đặc trưng từ hình ảnh.

Các hàm:

- **Mat calculateSIFT(const cv::Mat& image):** Tính toán và trả về mô tả SIFT của hình ảnh đầu vào.
- **Mat calculateORB(const cv::Mat& image):** Tính toán và trả về mô tả ORB của hình ảnh đầu vào.
- **Mat calculateColorHistogram(const cv::Mat& image):** Tính toán và trả về histogram màu chuẩn hóa của hình ảnh đầu vào.
- **Mat calculateColorCorrelogram(const cv::Mat& image):** Tính toán và trả về correlogram màu chuẩn hóa của hình ảnh đầu vào.

3. Các hàm trong Int Main

Mô tả: Thực hiện chương trình và hiện hình ảnh được truy vấn

Các hàm:

- **displayResults(const vector<pair<string, double>>& results, const string& imageFolder):** Hàm này hiển thị kết quả của truy vấn ảnh trong một cửa sổ GUI, bao gồm ảnh và điểm số tương đồng.
- **saveResultsToFile(const vector<pair<string, double>>& results, const string& resultFile):** Hàm này lưu trữ kết quả của truy vấn ảnh vào một tệp tin văn bản.
- **loadResultsFromFile(const string& resultFile):** Hàm này đọc kết quả của truy vấn ảnh từ một tệp tin văn bản.

III. Các đặc trưng

1. Color Histogram

Histogram màu biểu diễn phân bố tần suất của các màu sắc trong ảnh. Khi so sánh hai histogram màu, chúng ta thường sử dụng các phương pháp tính khoảng cách, ví dụ như khoảng cách Euclid hoặc chi-square. Khi thực hiện truy vấn hình ảnh bằng Color Histogram, các ảnh được sắp xếp theo mức độ tương đồng về phân phối màu sắc. Điểm tương đồng càng cao thể hiện rằng hai ảnh có xu hướng chia sẻ một phân bố màu sắc tương tự.

- **Ưu điểm:** Color Histogram là một phương pháp đơn giản và hiệu quả để đại diện cho màu sắc tổng quát của ảnh. Nó tính toán phân bố màu sắc và có thể dễ dàng so sánh bằng cách sử dụng các phương pháp như khoảng cách Chi-square.
- **Nhược điểm:** Phương pháp này không xem xét sự phân bố không gian của màu sắc, do đó không phản ánh được cấu trúc và hình dạng của các đối tượng trong ảnh.



2. Color Correlogram

Color correlogram là một phương pháp biểu diễn màu sắc và không gian trong ảnh, đo lường xác suất một màu xuất hiện ở một khoảng cách nhất định từ một màu khác. Không chỉ xem xét sự phân bố màu sắc, color correlogram còn xem xét mối quan hệ không gian giữa các màu, giúp phản ánh cấu trúc không gian và phân bố màu sắc trong ảnh một cách chi tiết. Phương pháp này cải thiện độ chính xác trong việc nhận dạng và so sánh ảnh so với chỉ sử dụng histogram màu, thường được áp dụng trong các hệ thống truy vấn ảnh để tìm kiếm và so sánh ảnh hiệu quả hơn.

- **Ưu điểm:** Color Correlogram mở rộng từ Color Histogram bằng cách xem xét sự tương quan giữa các cặp màu tại các khoảng cách không gian khác nhau. Điều này giúp phương pháp này phản ánh tốt hơn sự phân bố màu sắc và cấu trúc của ảnh.
- **Nhược điểm:** Tính toán Color Correlogram phức tạp và tốn tài nguyên hơn so với Color Histogram, và có thể không phù hợp cho các ứng dụng yêu cầu tốc độ cao.



3. SIFT

SIFT trích xuất các điểm đặc trưng từ hình ảnh và mô tả chúng thành các vector đặc trưng, không bị ảnh hưởng bởi tỉ lệ hoặc góc nhìn của ảnh. Để so sánh hai ảnh sử dụng SIFT, chúng ta thường tính toán khoảng cách giữa các vector đặc trưng, ví dụ như khoảng cách Euclid hoặc cosine similarity. Kết quả từ SIFT thường phản ánh mức độ tương đồng về cấu trúc và đặc điểm của các vật thể trong ảnh. Khi thực hiện truy vấn, các ảnh được sắp xếp theo mức độ tương đồng của các điểm đặc trưng, tức là các ảnh có các điểm đặc trưng giống nhau sẽ có điểm tương đồng cao.

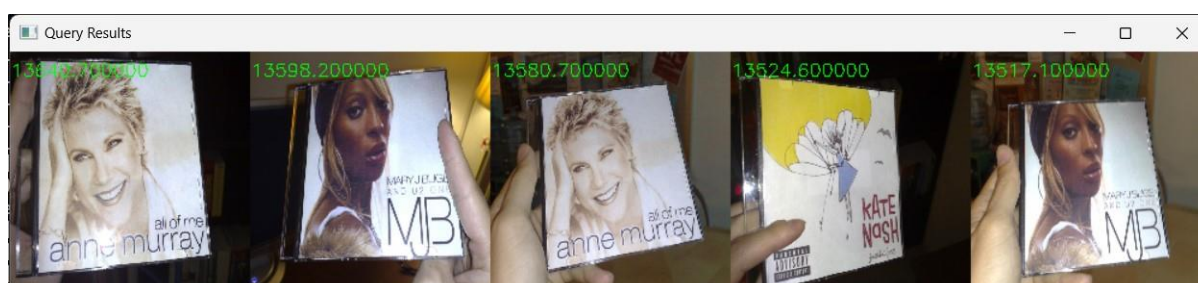
- **Ưu điểm:** SIFT phát hiện các điểm đặc trưng không bị ảnh hưởng bởi thay đổi tỷ lệ và góc nhìn của ảnh. Kết quả thường phản ánh mức độ tương đồng về cấu trúc và đặc điểm của các vật thể trong ảnh.
- **Nhược điểm:** Phương pháp này thường tính toán chậm và yêu cầu tài nguyên xử lý cao, không phù hợp cho các ứng dụng thời gian thực hoặc thiết bị có tài nguyên hạn chế.



4. ORB

ORB (Oriented FAST and Rotated BRIEF) là một phương pháp trích xuất và mô tả đặc trưng trong ảnh, được thiết kế để nhanh chóng và hiệu quả thay thế cho SIFT và SURF. ORB sử dụng tính năng FAST để phát hiện điểm đặc trưng và BRIEF để mô tả đặc trưng, kết hợp với tính năng định hướng để đảm bảo tính bất biến với sự thay đổi góc nhìn và tỷ lệ. ORB giúp nhận dạng và so sánh các đặc trưng của ảnh một cách nhanh chóng và chính xác, thích hợp cho các ứng dụng thời gian thực và thiết bị có tài nguyên hạn chế.

- **Ưu điểm:** ORB là sự thay thế nhanh chóng và hiệu quả cho SIFT và SURF, sử dụng ít tài nguyên hơn và có thể thực hiện trong thời gian thực. ORB bất biến với sự thay đổi góc nhìn và tỷ lệ, phù hợp cho các ứng dụng thời gian thực và trên thiết bị có tài nguyên hạn chế.
- **Nhược điểm:** Độ chính xác của ORB có thể thấp hơn SIFT trong một số tình huống phức tạp, đặc biệt là với các hình ảnh có nhiều chi tiết hoặc biến dạng.



Tổng kết:

- SIFT và ORB phù hợp cho việc nhận dạng và so sánh cấu trúc của các đối tượng trong ảnh.

- Color Histogram và Color Correlogram thích hợp cho việc so sánh tổng quát về màu sắc của ảnh.
- SIFT cung cấp độ chính xác cao hơn nhưng chậm hơn, trong khi ORB nhanh hơn nhưng kém chính xác hơn.
- Color Correlogram cung cấp thông tin chi tiết hơn về phân bố màu sắc nhưng phức tạp và tốn tài nguyên hơn so với Color Histogram.

IV. Hướng dẫn truy vấn

1. Tải dữ liệu

Khi khởi chạy chương trình, chương trình sẽ kiểm tra đường dẫn thư mục chứa các đặc trưng để khởi tạo cơ sở dữ liệu đặc trưng. Nếu thư mục đặc trưng không tồn tại hoặc trống, chương trình sẽ trích xuất các đặc trưng từ đường dẫn thư mục chứa ảnh và lưu lại các đặc trưng. Nếu đã có dữ liệu, hàm sẽ tải các đặc trưng đã tính toán trước đó và lưu vào cơ sở dữ liệu. Việc rút trích, lưu và lấy các đặc trưng đã xử lý là vô cùng quan trọng vì do tập dữ liệu khá lớn, nếu ta không chuẩn bị trước thì việc rút trích các đặc trưng sẽ ảnh hưởng lớn đến tài nguyên của máy. Trong khi máy tính đọc các đặc trưng sẽ hiển thị thông báo như sau:

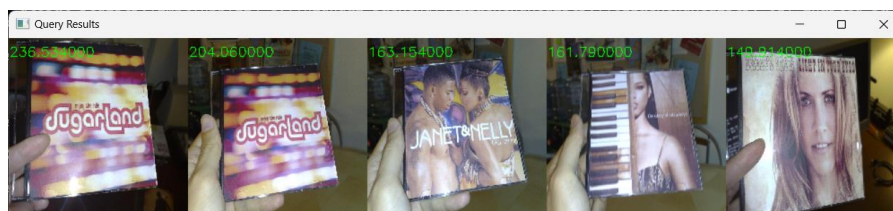
```
Dang load tap tin: ../features_1\01_1_features.yml
Dang load tap tin: ../features_1\01_2_features.yml
Dang load tap tin: ../features_1\01_3_features.yml
Dang load tap tin: ../features_1\02_1_features.yml
```

2. Truy vấn

Sau khi chương trình đã tải hết dữ liệu vào cơ sở dữ liệu. Chương trình sẽ cho phép người dùng được lựa chọn các đặc trưng để truy vấn ảnh. Người dùng có thể lựa chọn các đặc trưng Color Histogram, Color Correlogram, SIFT và ORB. Hoặc có thể dùng chương trình bằng cách ấn phím 0. Sau khi lựa chọn đặc trưng, người dùng sẽ được chọn số lượng bức ảnh tìm kiếm. Ví dụ người dùng muốn tìm kiếm ảnh theo đặc trưng Color Histogram và số lượng bức ảnh hiển thị ra màn hình là 5 thì sẽ thực hiện như ảnh sau:

```
Enter feature type (ColorHistogram/ColorCorrelogram/SIFT/ORB) or 0 to exit: ColorHistogram
Enter number of similar images to display: 5
```

Sau khi thực hiện chương trình sẽ hiển thị kết quả là các ảnh tương tự với ảnh truy vấn



Sau khi thực hiện truy vấn xong, chương trình sẽ hiển thị thời gian truy vấn và sẽ tiếp tục cho phép người dùng tiếp tục truy vấn

```

Enter feature type (ColorHistogram/ColorCorrelogram/SIFT/ORB) or 0 to exit: ColorHistogram
Enter number of similar images to display: 5
Similar image: 25_2_features.yml, similarity score: 236.534
Similar image: 25_3_features.yml, similarity score: 204.06
Similar image: 32_3_features.yml, similarity score: 163.154
Similar image: 01_3_features.yml, similarity score: 161.79
Similar image: 23_2_features.yml, similarity score: 149.914
[ INFO:0@2783.613] global window_w32.cpp:2993 cv::impl::Win32BackendUI::createWindow OpenCV
Query Results (1)
Time taken for the query: 36 milliseconds
Enter feature type (ColorHistogram/ColorCorrelogram/SIFT/ORB) or 0 to exit: |

```

3. Kết quả hiển thị

Chương trình sẽ hiển thị các ảnh trùng khớp với kết quả tìm kiếm và ở phía trên cùng bên trái mỗi ảnh sẽ là khoảng cách giữa ảnh truy vấn với ảnh trong cơ sở dữ liệu (khoảng cách càng nhỏ có nghĩa kết quả càng trùng khớp). Đồng thời chương trình sẽ in ra tên các file đặc trưng của ảnh, đây cũng đồng thời là tên của hình ảnh trong thư mục ảnh và in ra điểm số giống với ảnh gốc (đây là khoảng cách giữa ảnh trong cơ sở dữ liệu và ảnh tìm kiếm). Sau khi ta coi xong ảnh thì chương trình sẽ in ra thời gian thực hiện truy vấn

V. Tổng kết

Dưới đây là kết quả MAP khi thực hiện truy vấn của ảnh trong tập dữ liệu CD.

K	Color Histogram	Color Correlogram	SIFT	ORB
3	0.61	0.72	1	1
5	0.55	0.70	1	0.95
11	0.34	0.49	0.95	0.84
21	0.37	0.38	0.82	0.7

Khi so sánh các đặc trưng với nhau dựa trên kết quả MAP, ta có thể thấy SIFT và ORB đều cho hiệu suất tốt hơn khi so với Color Histogram và Color Correlogram. Và trong các đặc trưng, SIFT duy trì điểm MAP cao hơn khi so với ORB cho thấy nó là đặc trưng tốt nhất trong việc phân biệt ảnh

Dưới đây là thời gian trung bình thực hiện chương trình tính bằng đơn vị miliseconds

K	Color Histogram	Color Correlogram	SIFT	ORB
3	38	4081	318206	1155
5	35	4207	329796	1163
11	35	4049	330461	1174
21	35	4116	330785	1201

Nhận xét về tốc độ, hầu hết thời gian truy vấn 1 ảnh đều khá nhanh, đều dưới 4s, vì sử dụng chung phương pháp nên không có quá nhiều sự chênh lệch giữa việc hiển thị ít kết quả hay nhiều kết quả. Tuy thời gian để truy vấn ảnh bằng SIFT là khá cao (tầm 5p trên một lần truy vấn). Nhưng thay vì trực tiếp so sánh giữa các cặp descriptor tương ứng thì em so sánh toàn bộ các descriptor từ hai hình ảnh để tránh việc không đồng nhất các descriptor. Có thể trong lúc em khởi tạo lại đoạn code thì việc chuẩn bị cơ sở dữ liệu đã sử dụng phương pháp khác nên dẫn tới sự ko đồng nhất các descriptor này. Thông thường thì đặc trưng SIFT em làm thì sẽ có thời gian lâu hơn đặc trưng ORB không quá nhiều.

Sau khi thực hiện các cách kết hợp các đặc trưng với nhau thì em nhận thấy việc chỉnh sửa sự đóng góp giữa các đặc trưng khá là tốn tài nguyên thời gian, tuy đã cố chỉnh để phù hợp thì em thấy việc kết hợp các đặc trưng không mang lại kết quả nhưng mình mong đợi. Đồng

thời còn làm tăng thời gian truy vấn. Nên mã nguồn em nộp sẽ không bao gồm các phương pháp này. Đồng thời vì việc lưu trữ cơ sở dữ liệu của các đặc trưng còn khá hạn chế, nên mỗi lần thao tác lưu cơ sở dữ liệu vào ram của em tốn cực kì nhiều thời gian nên đã hạn chế việc thử nghiệm và thao tác trên bộ dữ liệu lớn.

VI. Tài liệu tham khảo

[1] Tài liệu môn học slide, textbook

[2] <https://nlp.stanford.edu/IR-book/html/htmledition/evaluation-of-ranked-retrieval-results-1.html>

[3] [OpenCV: OpenCV modules](#)