

**ĐẠI HỌC QUỐC GIA TP. HCM
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**



ĐỒ ÁN

NHẬN DẠNG THỊ GIÁC VÀ ỨNG DỤNG

Giảng viên hướng dẫn: TS. Lê Đình Duy
TS. Nguyễn Tấn Trần Minh Khang
Học viên thực hiện: Phan Quốc Quang
Mã học viên: CH1601016

Tháng 12 năm 2017

Mục lục

1	Thông tin đồ án	1
2	Mục tiêu đồ án	1
3	Phương án thực hiện	1
3.1	Công cụ sử dụng.....	1
3.2	Phương án thực hiện.....	2
4	Hiện thực ứng dụng	2
4.1	Tạo imageIndexs với SURF descriptor	2
4.2	Tạo imageIndexs với SURF descriptor và HOG feature	3
4.3	Tìm kiếm ảnh.....	3
4.4	Giới thiệu giao diện	5
4.5	Đánh giá kết quả tìm kiếm	7
5	Báo cáo mở rộng – Local features.....	8

1 Thông tin đồ án

- Địa chỉ Github: https://github.com/quangpq11uit/VIR_2017
- Địa chỉ Youtube của video demo: <https://youtu.be/4vsgF1UarsQ>
- Địa chỉ thư mục Google Drive lưu các image indexes và dataset (thư mục gồm có dataset, image indexes và một số ảnh query):
<https://drive.google.com/open?id=1KMmrotfsFv1nFrGEW5nxLWhYCRqil89>

2 Mục tiêu đồ án

Mục tiêu của đồ án là xây dựng một chương trình ứng dụng với các yêu cầu:

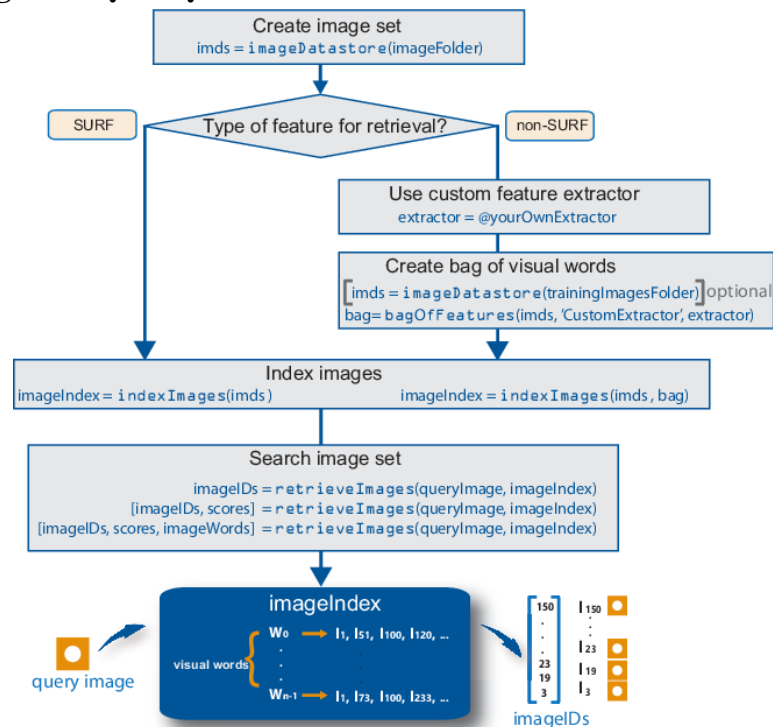
- Tìm kiếm ảnh (image retrieval) sử dụng BoW trên bộ dataset Paris Buildings, với giao diện người dùng được xây dựng bằng Matlab.
- Do dataset Paris Buildings có nhiều ảnh bị hỏng và giới hạn phần cứng, trong đồ án này chỉ sử dụng 1474 ảnh từ dataset gốc.
- Người dùng có thể tìm kiếm toàn bộ bức ảnh hoặc chọn một vùng ảnh mà họ quan tâm.
- Đánh giá hệ thống bằng cách so sánh với groundtruth của dataset.

3 Phương án thực hiện

3.1 Công cụ sử dụng

- Matlab 2017a: <https://www.fshare.vn/folder/F4Y7WNB928RT>
- Bộ công cụ “Computer Vision System Toolbox” của Matlab:
<https://www.mathworks.com/help/vision/ug/image-retrieval-with-bag-of-visual-words.html>
- Bộ công cụ xây dựng giao diện của Matlab:
<https://www.mathworks.com/discovery/matlab-gui.html>
- Bộ dataset Paris Buildings:
<http://www.robots.ox.ac.uk/~vgg/data/parisbuildings/>

3.2 Phương án thực hiện



Hình 1: Cách hoạt động của truy vấn ảnh dựa trên BoW của “Computer Vision System Toolbox”¹

- Tạo image indexes cho bộ dữ liệu Paris Buildings bằng SURF descriptor (không cần tạo bag of words vì Matlab đã hỗ trợ)
- Tạo bag of words và imageIndexes với SURF descriptor kết hợp với HOG feature
- Xây dựng giao diện và sử dụng các image indexes đã tạo để tìm kiếm ảnh.
- So sánh kết quả với groundtruth của bộ dữ liệu Paris Buildings

4 Hiện thực ứng dụng

4.1 Tạo imageIndexes với SURF descriptor

Đầu tiên, tạo một datastore cho tất cả ảnh:

```
imds = imageDatastore(fullfile(datasetDir));
```

Tiếp theo, tạo một image indexes và lưu kết quả vào ổ cứng:

```
imageIndex = indexImages(imds, 'SaveFeatureLocations', false);
save(imageIndexFileName, 'imageIndex');
```

¹ <https://www.mathworks.com/help/vision/ug/image-retrieval-with-bag-of-visual-words.html>

4.2 Tạo imageIndexs với SURF descriptor và HOG feature

Ý tưởng là tìm các điểm đáng quan tâm (interested points) bằng SURF descriptor. Sau đó, sử dụng HOG feature² để tạo các word cho các điểm này.

Đầu tiên, tạo một hàm để tìm các feature cho một ảnh:

```
function [features, featureMetrics] = BagOfFeaturesHOGExtractor(I)
    [height,width,numChannels] = size(I);
    % Chuyển ảnh sang dạng ảnh mức xám
    if numChannels > 1
        grayImage = rgb2gray(I);
    else
        grayImage = I;
    end
    % Tìm các điểm interest points
    points = detectSURFFeatures(grayImage);
    % Xây dựng các word bằng HOG feature
    features = extractHOGFeatures(grayImage, points, 'BlockSize', [5 5]);
    % Rút thông số cần thiết từ các đặc trưng
    featureMetrics = var(features,[],2);
end
```

Input: một ma trận thể hiện cho một ảnh số.

Output: các đặc trưng tìm được và thông số.

Tiếp theo, ta tạo bag of words và sử dụng kết quả để tạo image index:

```
extractorFcn = @BagOfFeaturesHOGExtractor;
bag = bagOfFeatures(imds, 'CustomExtractor', extractorFcn,
    'VocabularySize', 20000);
imageIndex = indexImages(imds, bag, 'SaveFeatureLocations', false);
```

Hai loại bag of words nói trên có thể được sinh ra từ một hàm duy nhất:

```
function buildBagWithCustomExtractor(datasetDir, extractor, bagFileName,
    imageIndexFileName)
```

Input:

- datasetDir: đường dẫn tới thư mục chứa dataset.
- extractor: loại descriptor được dùng, có 2 lựa chọn: 'surf', 'hog-surf'
- bagFileName: tên của file bag of words
- imageIndexFileName: tên của file image index.

4.3 Tìm kiếm ảnh

Hàm retrieval được dùng để tìm ra đường dẫn của các ảnh phù hợp nhất với query:

```
function imagePaths = retrieval(descriptor, queryImg, queryROI,
    numResults)

    % Đường dẫn tới thư mục chứa bộ dataset Oxford Building
```

² <http://lear.inrialpes.fr/pubs/2006/DTS06/eccv2006.pdf>

```
dir = '/Users/quangphan/Documents/data_vra/paris_1k/';
% Tai thu muc dataset va image indexs phu hop
if isequal(descriptor, 'surf')
    load('../VRA_Features/index_surf.mat');
elseif isequal(descriptor, 'hog-surf')
    load('../VRA_Features/index_hog.mat');
end

% neu toa do anh bang 0 thi ham retrieval se bi loi,
% ta phai loai bo trong hop nay
for k = 1 : length(queryROI)
    if queryROI(k) <= 0
        queryROI(k) = 1;
    end
end

% Tim ra 20 index phu hop voi query
if size(queryROI, 1) > 0
    % neu co queryROI thi chi can tim mot phan cuar hinh
    [imageIDs, scores] = retrieveImages(queryImg, imageIndex, 'ROI',
queryROI, 'NumResults', numResults);
else
    [imageIDs, scores] = retrieveImages(queryImg, imageIndex,
'NumResults', numResults);
end

% tao imageDataStore tu thu muc chua dataset
imds = imageDatastore(fullfile(dir));
% Lay duong dan cua hinh tu cac index tim duoc
imagePaths = imds.Files(imageIDs);
% in danh sach duong dan ra man hinh
imagePaths
end
```

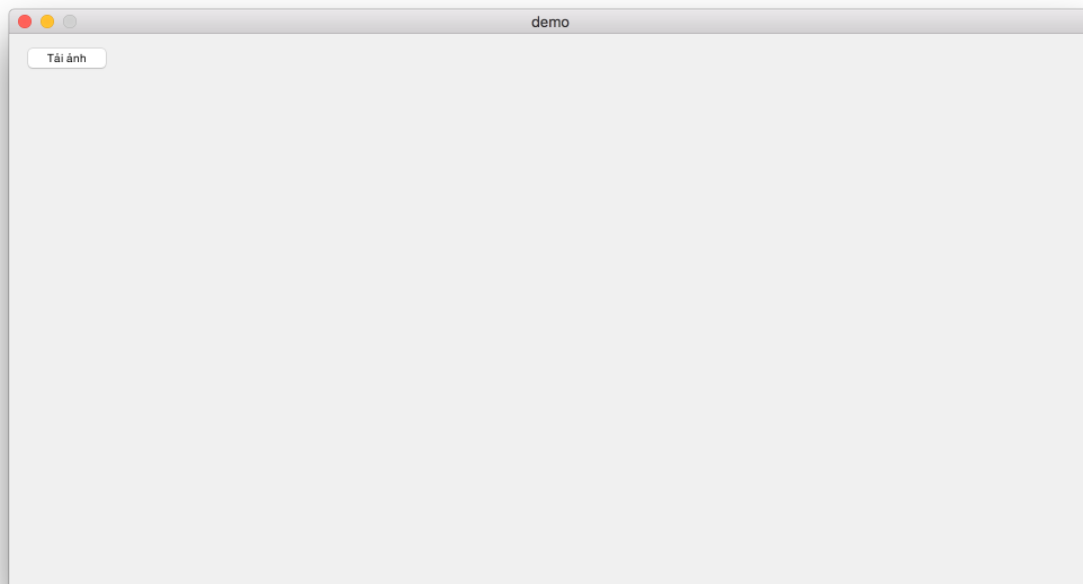
Input:

- descriptor: loại descriptor được dùng, có 3 lựa chọn: 'surf', 'sift', 'hog-surf'
- queryImg: ảnh query (biểu diễn dưới dạng ma trận)
- queryROI: vùng ảnh cần tìm kiếm, là một ma trận có dạng [X Y Width Height]

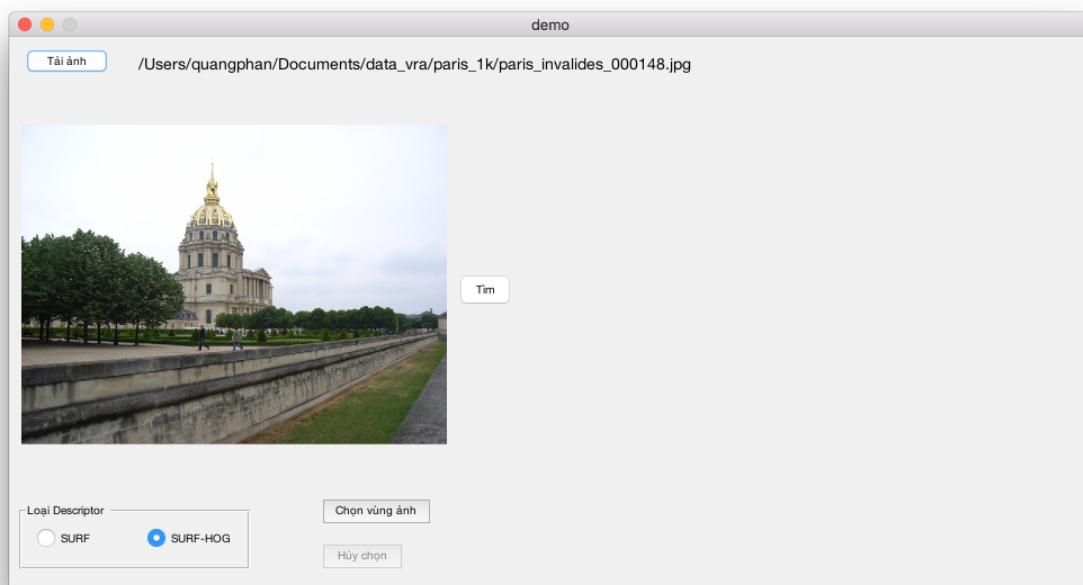
Output: 20 đường dẫn đến các ảnh phù hợp nhất với query.

4.4 Giới thiệu giao diện

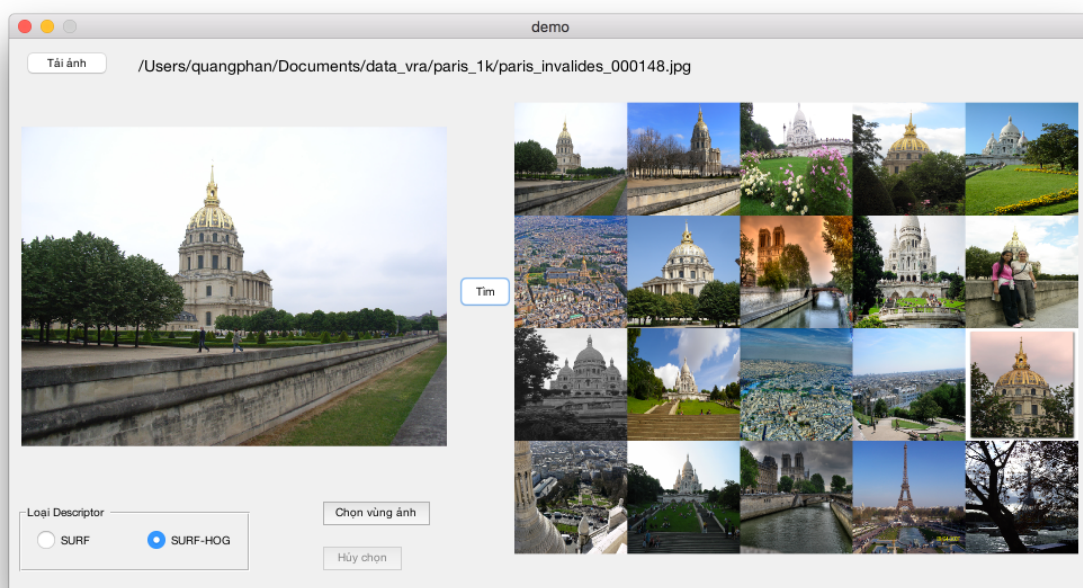
Chạy hàm demo (demo.m) để khởi động giao diện của hệ thống. Giao diện khi mới mở chương trình. Chọn “Tải ảnh” để chọn ảnh query:



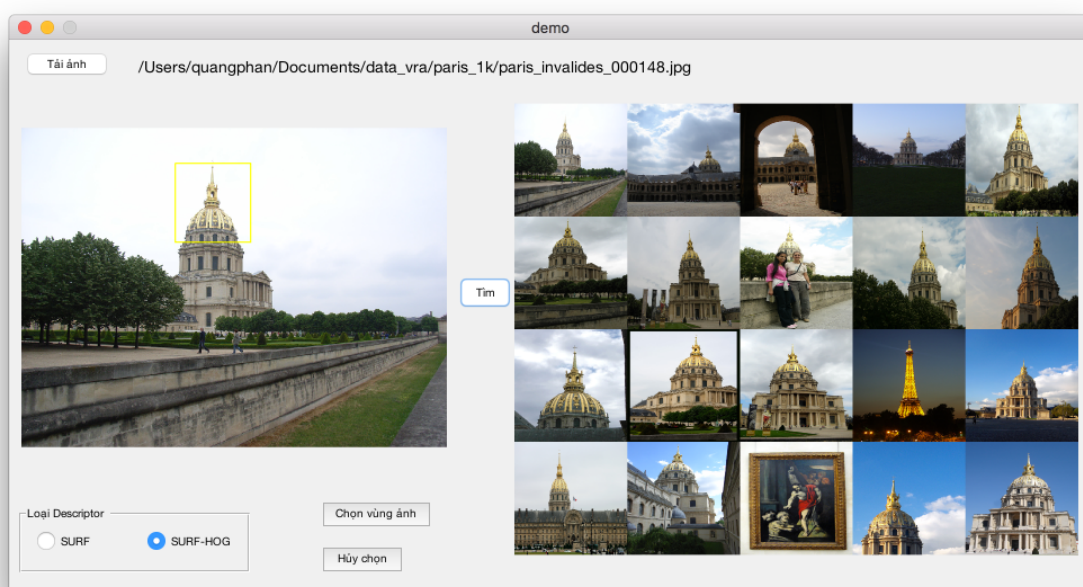
Sau khi chọn ảnh, đường dẫn của ảnh và ảnh sẽ được hiển thị.



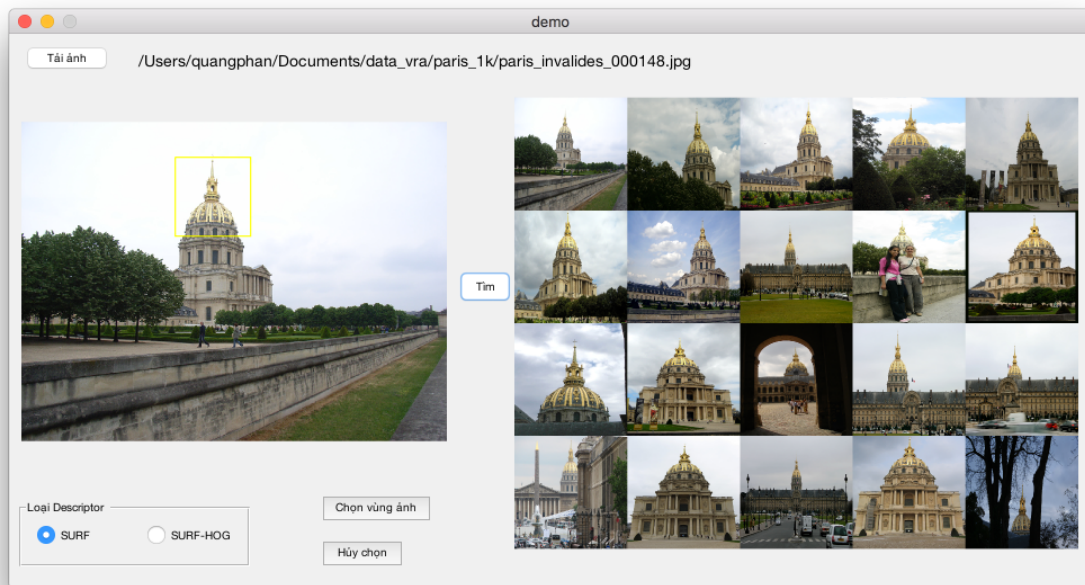
Chọn “Tìm” để xem kết quả, 20 kết quả phù hợp nhất sẽ được hiển thị ở bên phải:



Bên dưới ảnh, có thể chọn vùng cần tìm kiếm bằng “Chọn vùng ảnh” và chọn vùng cần tìm kiếm, nếu không muốn tìm kiếm theo một vùng ảnh nữa thì có thể chọn “Hủy chọn”:



Để thay đổi descriptor khác, chọn loại descriptor trong panel “Loại descriptor”:



4.5 Đánh giá kết quả tìm kiếm

Việc đánh giá kết quả tìm kiếm được thực hiện bằng cách tính trung bình R-Precision (số kết quả trả về cho từng query sẽ bằng với số kết quả đúng của groundtruth query) của tất cả kết quả ở mức OK trong groundtruth của bộ dataset. Do image indexes không được tạo từ dataset hoàn chỉnh, ta không tính các kết quả không nằm trong dataset hiện tại.

Hàm thực thi việc đánh giá là:

```
function evaluationAll(descriptor, datasetDir)
```

Input:

- descriptor: loại descriptor được dùng, có 2 lựa chọn: 'surf', 'hog-surf'
- datasetDir: đường dẫn tới thư mục chứa dataset.

Ví dụ: `evaluationAll('surf', '/Users/quangphan/Documents/data_vra/paris_1k/')`

Kết quả:

Descriptor	R-Precision trung bình
SURF	0.354706
SURF-HOG	0.509183

5 Báo cáo mở rộng – Local features

Local features là một kiểu mẫu hoặc kết cấu đặc trưng được tìm thấy trong ảnh, thường là một vùng ảnh có sự khác biệt với các vùng xung quanh về kết cấu, màu và độ tương phản. Ví dụ: các điểm đốm, góc, cạnh,...

Local features giúp tìm các ảnh của cùng một đối tượng, bỏ qua các sự khác biệt về: độ sáng, góc nhìn, sự ảnh hưởng của các đối tượng khác,...

Một local feature tốt phải đảm bảo các tính chất:

- Lặp lại và chính xác: trong các ảnh khác nhau của cùng một đối tượng, luôn đảm bảo xuất hiện local feature tại cùng một vị trí
- Không trùng lặp: vùng ảnh xung quanh phải có sự khác biệt rõ ràng với local feature.
- Dễ dàng định vị: local feature có một vị trí trên đối tượng mà thay đổi góc nhìn không làm thay đổi vị trí này.

Feature detection:

- Tìm một vùng ảnh có giá trị đặc biệt, như góc hoặc điểm đốm.
- Được dùng để tìm các điểm cần quan tâm cho các bước xử lý sau.
- Các điểm này không cần liên quan tới các yếu tố vật lý của đối tượng, ví dụ: cạnh bàn.
- Điểm quan trọng là cần tìm được các điểm bất biến kể cả khi thay đổi góc nhìn, độ lớn của ảnh.

Features extraction:

- Tạo ra một descriptor từ các điểm ảnh xung quanh một local feature.
- Descriptor được tạo ra bằng cách biểu diễn các điểm xung quanh một local feature dưới dạng một vector.
- Vector biểu diễn này phải không thay đổi khi vùng ảnh thể hiện dưới một góc nhìn khác hoặc dưới tỉ lệ khác.
- Các loại descriptor phổ biến:
 - o Từ gradient: SIFT, SURF.
 - o Nhị phân: BRISK, FREAK

--Hết--