

**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**NGÀNH KHOA HỌC MÁY TÍNH**



Báo cáo đồ án môn học

**NHẬN DẠNG THỊ GIÁC VÀ ỨNG DỤNG**

**Đề tài:**

**HỆ THỐNG TÌM KIẾM ẢNH**

**SỬ DỤNG PHƯƠNG PHÁP BAG OF WORDS**

Thực hiện:

Nguyễn Mạnh Phúc – CH1601033

GVHD:

TS. Lê Đình Duy

TS. Nguyễn Tấn Trần Minh Khang

**Tháng 12/2017**

Mục lục

[1 Mục tiêu và nội dung đồ án 3](#_Toc503121488)

[1.1 Mục tiêu 3](#_Toc503121489)

[1.2 Nội dung đồ án 3](#_Toc503121490)

[2 Harris affine region detector và SIFT descriptor 4](#_Toc503121491)

[2.1 Harris affine region detector 4](#_Toc503121492)

[2.1.1 Phép biến đổi Affine 4](#_Toc503121493)

[2.1.2 Harris-Laplace Detector 5](#_Toc503121494)

[2.1.3 Harris affine region detector 6](#_Toc503121495)

[2.2 SIFT descriptor 6](#_Toc503121496)

[2.2.1 SIFT Descriptor (Scale Invariant Feature Transform) 6](#_Toc503121497)

[2.2.2 rootSIFT descriptor 8](#_Toc503121498)

[3 Cách thực hiện chương trình 8](#_Toc503121499)

[3.1 Giai đoạn huấn luyện 8](#_Toc503121500)

[3.2 Giai đoạn Tìm kiếm ảnh 9](#_Toc503121501)

[3.3 Tìm kiếm ảnh từ giao diện 10](#_Toc503121502)

[3.4 Tạo file dll từ Matlab và gọi hàm từ C# (ASP.Net) 10](#_Toc503121503)

[4 Kết quả đánh giá 10](#_Toc503121504)

[4.1 Một số hình ảnh kết quả Truy vấn: 10](#_Toc503121505)

[4.2 Kết quả thực hiện 14](#_Toc503121506)

[4.2.1 Kết quả sử dụng Harris affine region dectector kết hợp với rootSIFT descriptor 14](#_Toc503121507)

[4.2.2 Kết quả sử dụng Harris affine region dectector kết hợp với SIFT descriptor 16](#_Toc503121508)

[4.2.3 Kết quả sử dụng Hessian affine dectector kết hợp với rootSIFT descriptor 18](#_Toc503121509)

[4.2.4 Kết quả sử dụng Hessian affine dectector kết hợp với SIFT descriptor 20](#_Toc503121510)

[4.2.5 Kết quả sử dụng SIFT detector kết hợp với rootSIFT descriptor 22](#_Toc503121511)

[4.2.6 Kết quả sử dụng SURF features 23](#_Toc503121512)

[5 Lưu ý khi chạy chương trình. 25](#_Toc503121513)

[5.1 Runtime 25](#_Toc503121514)

[5.2 Các tập tin cần thiết 26](#_Toc503121515)

[6 Tài liệu tham khảo 26](#_Toc503121516)

# Mục tiêu và nội dung đồ án

## Mục tiêu

Bài toán tìm kiếm ảnh dựa trên là một trong những bài quan trọng và được nghiên cứu rộng rãi.

Mục tiêu của đồ án này là:

* Vận dụng, ôn tập kiến thức môn học Nhận dạng thị giác và ứng dụng
* Vận dụng các công cụ, thư viện để xây dựng một ứng dụng cho phép tìm kiếm ảnh dựa trên nội dung của ảnh sử dụng phương pháp bag of words.
* Rèn luyện khả năng xây dựng một ứng dụng sử dụng Matlab
* Tìm hiểu chi tiết về Harris affine region detector và cách sử dụng detector để phát hiện ra các keypoints và biểu diễn các keypoints này sử dụng SIFT descriptor

## Nội dung đồ án

Nội dung đồ án này là xây dựng một ứng dụng tìm kiếm ảnh dựa trên các phương pháp bag of words sử dụng Matlab, thư viện vlfeat, thư viện AKM và ASP.Net qua hai giai đoạn và training (trên Matlab) và tìm kiếm ảnh (sử dụng ASP.Net). Ngoài ra, trong đồ án này, em còn chạy thử training và tìm kiếm ảnh sử dụng sử dụng đăc trưng SURF được cung cấp bởi Matlab. Code phần training trên Matlab có tham khảo từ [2] và được thay đổi lại cho phù hợp với nhu cầu. Ứng dụng được giới thiệu tại: <https://youtu.be/mTAa5_c9dJw>

Đồ án sử dụng bộ dữ liệu oxford building (http://www.robots.ox.ac.uk/~vgg/data/oxbuildings/ ) để chạy thử và đánh giá hệ thống thống qua 55 truy vấn được cung cấp sẵn và so sánh hệ thống khi sử dụng các loại detector khác nhau cũng như sử dụng 2 loại descriptor là SIFT và rootSIFT.

Đồ án này còn trình bày nội dụng về phương pháp phát hiện các key points Harris affine region detector và SIFT descriptor cũng như phiên bản rootSIFT descriptor

# Harris affine region detector và SIFT descriptor

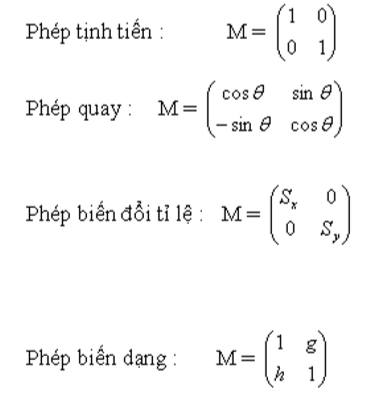
## Harris affine region detector

### Phép biến đổi Affine

Phép biến đổi Affine là phép biến đổi tọa độ điểm đặc trưng của đối tượng thành tập tương ứng các điểm mới để tạo ra các hiệu ứng cho toàn đối tượng

Ví dụ: phép biến đổi tọa độ với chỉ 2 điểm đầu cuối của đoạn thẳng tạo thành 2 điểm mới mà khi nối chúng với nhau tạo thành đoạn thẳng mới. Một số phép biến đổi affine cơ bản là:

* Phép bất biến
* Phép biến đổi tỉ lệ - Scaling
* Phép biến dạng
* Phép xoay



Hình 1. Các phép biến đổi affine transformation

Phép biến đổi affine có một số tính chất sau:

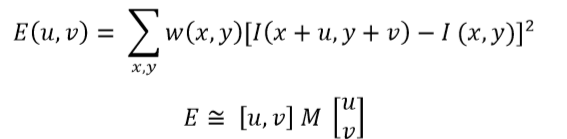
* Bảo tồn đường thẳng
* Bảo tồn sự song song
* Bảo tồn tỉ lệ khoảng cách.

Các phép biến đổi affine có thể kết hợp lại với nhau.

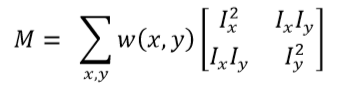
### Harris-Laplace Detector

**Harris corner detector**

Ý tưởng của harris corner detector là di chuyển một khung trong hình để tìm được những điểm có sự thay đổi lớn về cường độ ánh sáng (độ xám)



**Trong đó:**

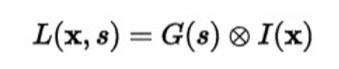


Trong đó Ix là đạo hàm của ảnh theo x, Iy là đạo hàm của ảnh theo y, w là khung trong số.

Khi đó, ta tìm các ảnh có R = det(M) - α x trace(M) lớn hơn một mức nào đó, điểm đó sẽ được coi là key point

**Gaussian scale space**

Biểu diễn Gaussian scale space của một ảnh là tập hợp các ảnh là tích chập của ảnh gốc với nhiều Gaussian kernel ở nhiều tỉ lệ scale khác nhau.



Trong đó L là ảnh đã được làm trơn (smooth) với tỉ lệ s. G(s) là một isotropic, circular Gaussian kernel

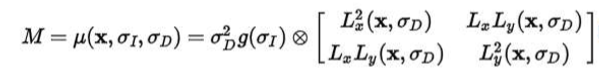
**Harris-Laplace dectector**

Harris-Laplace dectector gồm 2 giai đoạn:

1. Áp dụng Harris corner dectector ở nhiều mức tỉ lệ khác nhau

2. Chọn một tỉ lệ phù hợp

Cụ thể, tương tự ma trận M trong Harris corner detector, ma trận M trong Harris-Laplace detector được tính như sau:



Trong đó là tỉ lệ scale của ảnh, g( ) là kernel tương ứng với tỉ lệ scale , x là điểm ảnh, L(x) là điểm ảnh đã được làm trơn, Lx (x, ), Ly (x, ) là đạo hàm của ảnh tại X với ảnh đã làm trơn khi sử dụng Gaussian kernel . là tỉ lệ scale hiện tại mà key point đang được phát hiện (detect).

### Harris affine region detector

Harris affine region detector hoạt động như sau:

1. Xác định các vùng ban đầu bằng Harris-Laplace Detector.

2. Với mỗi vùng ban đầu, áp dụng thuật toán affine shape adaptation

3. Tìm kiếm các key points, tỉ lệ integration và tỉ lệ differentiation

4. Cập nhật vùng affine region sử dụng các tỉ lệ và các key points

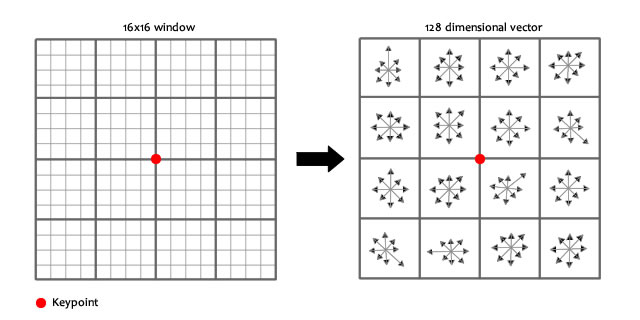
5. Lặp lại bước 3 cho đến khi điều kiện dừng thỏa mãn.

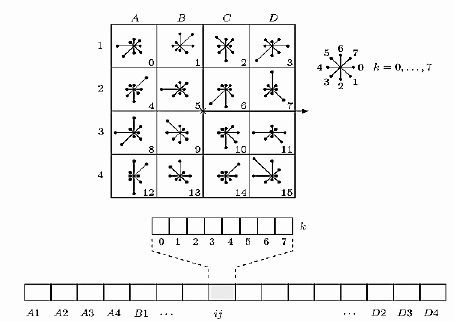
## SIFT descriptor

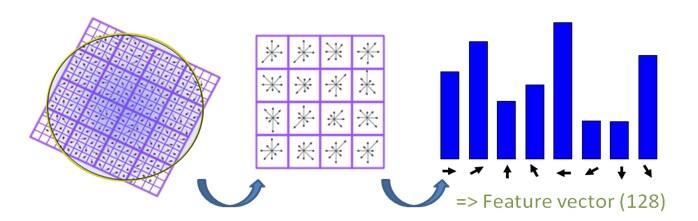
### SIFT Descriptor (Scale Invariant Feature Transform)

SIFT descriptor là một trong những descriptor được sử dụng phổ biến để thể hiện các key points. Ý tưởng chính của SIFT descriptor như sau. SIFT là:

* Lấy 16 x 16 điểm ảnh xung quanh key points và chia thành 4 x 4 block. Có tổng công 16 block.
* Mỗi block được mô tả bởi histogram theo 8 hướng (image gradients) theo chiều kim đồng hồ. Bắt đầu là hướng nằm ngang
* Sau đó, các histogram của mỗi block sẽ được nối lại với nhau theo hàng rồi theo cột để được 128 chiều.







Hình Cách tính SIFT descriptor

### rootSIFT descriptor

rootSIFT là một phiên bản cải tiến của SIFT descriptor. rootSIFT được tính bằng cách:

* Normalize SIFT feature bằng phương pháp L1: lấy mỗi phần tử trong vector chia cho tổng các phần tử
* Lấy căn bậc 2 mỗi phần tử sau khi được normalize

# Cách thực hiện chương trình

## Giai đoạn huấn luyện

Quy trình thực hiện là:

Bước 1: Rút trích đặc trưng

Sử dụng Harris affine region detector để rút trích các keypoint sử dụng thư viện vlfeat (hàm vl\_codet) : vl\_covdet(I, 'Method', 'HarrisLaplace', 'estimateAffineShape', true);

* Tính đặc trưng rootSIFT trên các keypoint
* Số chiều đặc trưng: 128
* Số ảnh: 5063
* Do số lượng features khi để ảnh gốc lớn nên trong quá trình rút trích đặc trưng thì đã giảm thử rút trích trên ảnh có được thu nhỏ với tỉ lệ kích thước là 60% mỗi chiều (imresize (I, 0.6)).

Bước 2: Xây dựng Visual Words (dictionary)

* Sử dụng thuật toán gom cụm Approximate K-Mean (AKM) trong bộ thư viện vlfeat (ccvBowGetDict)
* Sử dụng các đặc trưng đã rút trích ở trên.
* Số lượng cluster (words): 100000
* Số lượng k-d tree: 8
* Số lần lặp: 15
* Mức độ tương tự của các words được tính bằng phương pháp l2 – khoảng cách euclidean

Bước 3: Tính Word-ID cho từng ảnh trong dataset

* Với mỗi đặc trưng trong từng ảnh, ta tìm Word-ID của word dựa trên dictionary (tập Visual Words) đã xây dựng ở bước 2 sử dụng hàm ccvBowGetWordsInit và và ccvBowGetWords

Bước 4: Xây dựng Inverted file

* Inverted file: với mỗi visual word, ta xác định các ảnh có chứa visual word đó theo một thứ tự nhất **ccvInvFileInsert** định sử dụng hàm và **ccvInvFileCompStats**
* Các visual word có trọng số được tính theo phương pháp tf- idf

## Giai đoạn Tìm kiếm ảnh

Bước 1. Rút trích đặc trưng rootSIFT của ảnh cần tìm kiếm sử dụng HarrisLaplace - Affine region detector như bước 1

Bước 2. Tính word-id cho các đặc trưng đã tính ở trên Bước 3. Tính ranked list sử dụng hàm

* Tạo bảng Word Count để thống kê các visual word và tần số xuất hiện tương ứng trong ảnh tìm kiếm sử dụng hàm ccvBowGetWords
* Sử dụng InvertedFile để tạo ra bảng ranked list (hàm ccvInvFileSearch).
* Mức độ tương tự của các words được tính bằng phương pháp l2 – khoảng cách euclidean

Bước 4: Tạo ra kết qủa.

* Dựa trên ranked list, ta tìm được top các ảnh phù hợp với ảnh truy vấn.

## Tìm kiếm ảnh từ giao diện

Bước 1: Chọn ảnh để tìm kiếm

Bước 2: Chọn một vùng trong ảnh để tìm kiếm, vùng không được chọn sẽ được làm tối đi

Bước 3: Hệ thống lấy ảnh được tải lên từ client và lưu lại ảnh với 1 tên duy nhất

Bước 4: Hệ thống sử dụng ảnh ở bước 3 để tìm kiếm

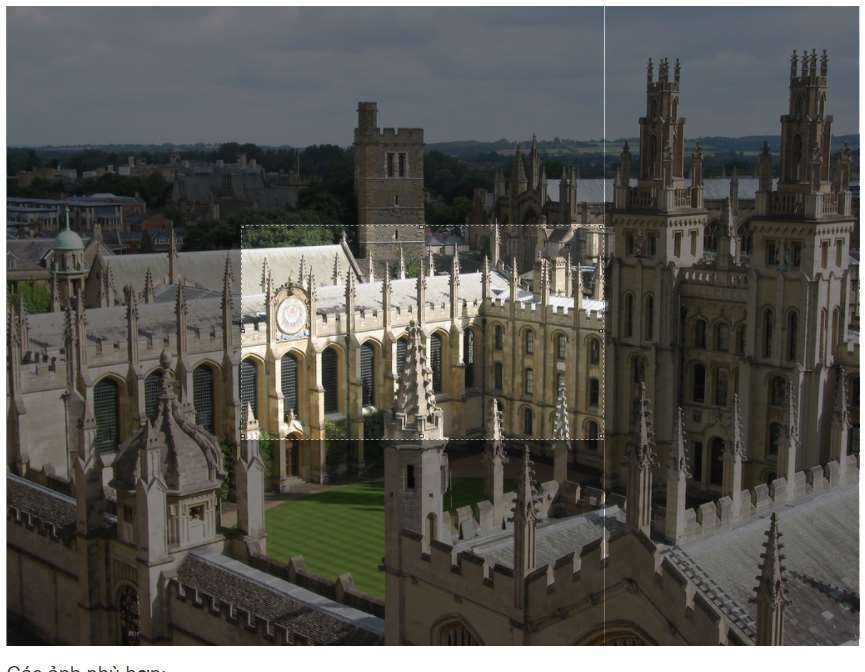
Bước 5. Hiển thị các ảnh kết quả (tối đa 1000 ảnh) có sử dụng phân trang (20 ảnh/trang)

## Tạo file dll từ Matlab và gọi hàm từ C# (ASP.Net)

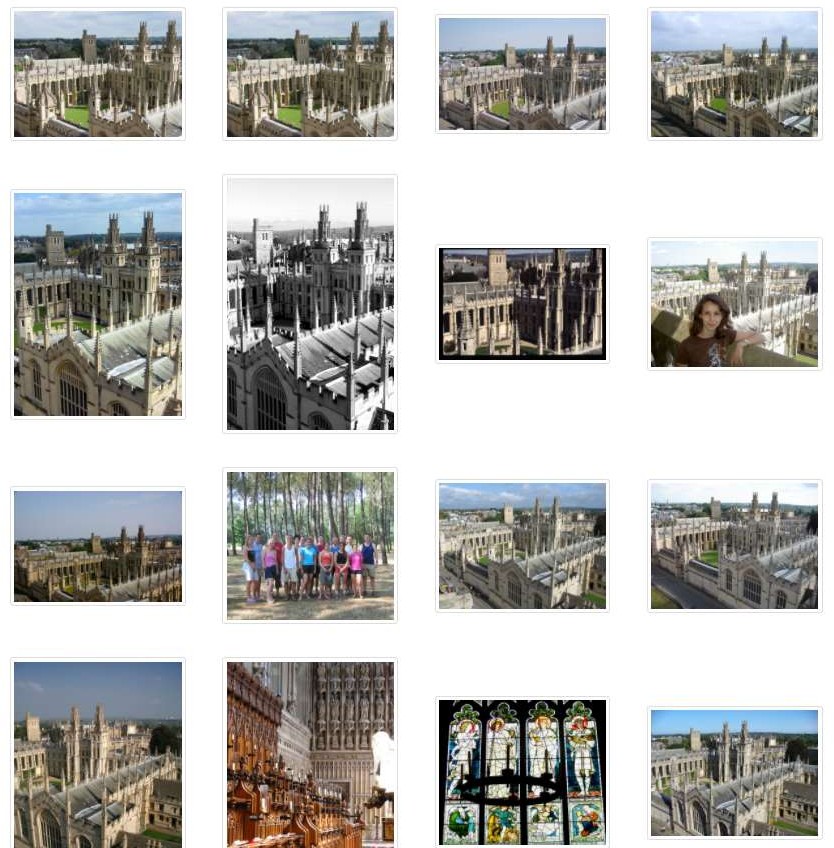
* Sử dụng chức năng library compiler của Matlab để tạo ra các DLL để có thể gọi hàm Matlab từ C#. Trong quá trình build thì xây dựng
* DLL được build từ project Search1.1prj và có thể tìm thấy trong thư mục Search 1.1, file này nằm tại thư mục “Search1.1\for\_redistribution\_files\_only”

# Kết quả đánh giá

## Một số hình ảnh kết quả Truy vấn:



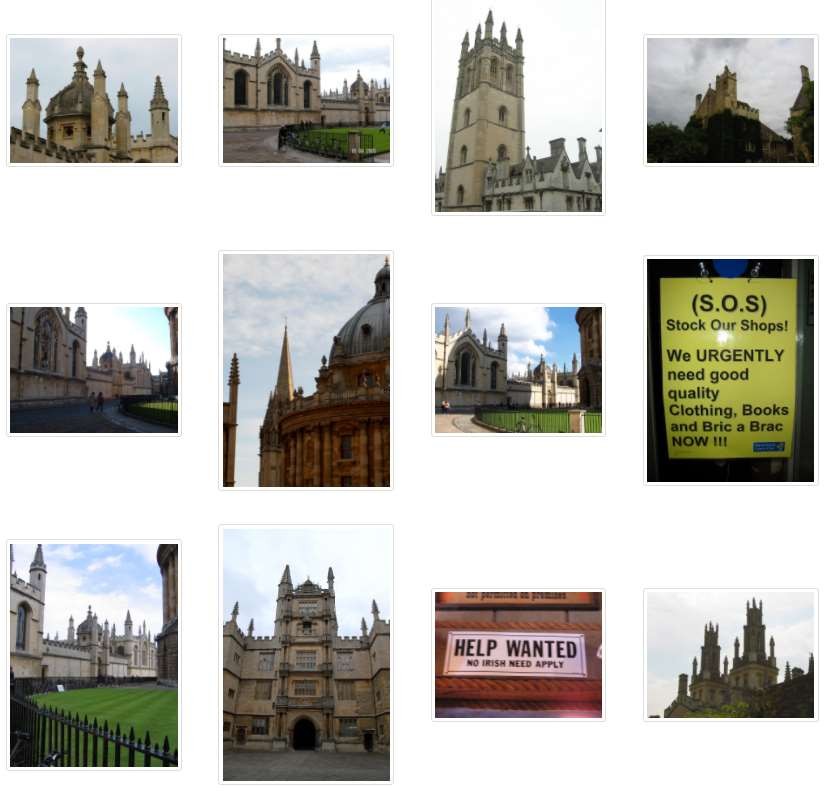
Kết quả:



Truy vấn:



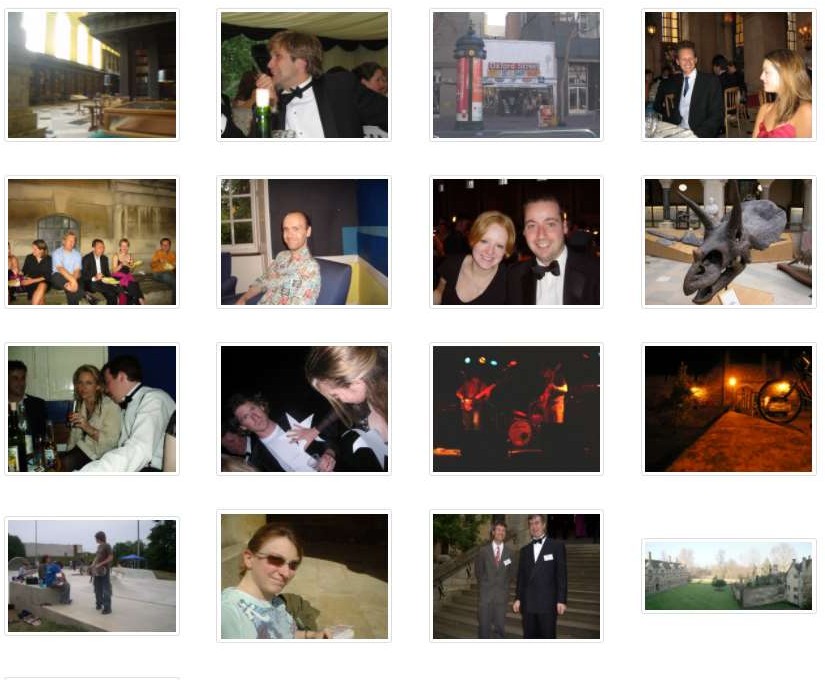
Kết quả:



Truy vấn:



Kết quả:



## Kết quả thực hiện

Các kết quả này được thực hiện dựa trên các bộ query mẫu trong thư mục groundtruth. Kết quả này được so sánh với các phương pháp khác (dùng SIFT descriptor, Hessian- affine detector). Các tỉ lệ dưới được thực hiện với số ảnh tối đa là 5063 ảnh tức là số ảnh trả về từ truy vấn có thể là toàn bộ số ảnh.

### Kết quả sử dụng Harris affine region dectector kết hợp với rootSIFT descriptor

|  |  |  |
| --- | --- | --- |
| STT | Truy vấn | Độ chính xác |
| 1 | all\_souls\_000013 | 0.214455 |
| 2 | all\_souls\_000026 | 0.229988 |
| 3 | all\_souls\_000051 | 0.787867 |
| 4 | ashmolean\_000000 | 0.800504 |
| 5 | ashmolean\_000007 | 0.422111 |
| 6 | ashmolean\_000058 | 0.169143 |
| 7 | ashmolean\_000269 | 0.119844 |
| 8 | ashmolean\_000305 | 0.614251 |
| 9 | balliol\_000051 | 0.690696 |
| 10 | balliol\_000167 | 0.0982744 |
| 11 | balliol\_000187 | 0.219778 |
| 12 | balliol\_000194 | 0.514239 |
| 13 | bodleian\_000107 | 0.0662962 |
| 14 | bodleian\_000108 | 0.31455 |
| 15 | bodleian\_000132 | 0.766667 |
| 16 | bodleian\_000163 | 0.161107 |
| 17 | bodleian\_000407 | 0.0927883 |
| 18 | christ\_church\_000179 | 0.628033 |
| 19 | christ\_church\_000999 | 0.513193 |
| 20 | christ\_church\_001020 | 0.527654 |
| 21 | cornmarket\_000019 | 0.725635 |
| 22 | cornmarket\_000047 | 0.711109 |
| 23 | cornmarket\_000105 | 0.162982 |
| 24 | cornmarket\_000131 | 0.544427 |
| 25 | hertford\_000015 | 0.720003 |
| 26 | hertford\_000027 | 0.783783 |
| 27 | hertford\_000063 | 0.820086 |
| 28 | keble\_000028 | 0.636084 |
| 29 | keble\_000055 | 0.359666 |
| 30 | keble\_000214 | 0.858219 |
| 31 | keble\_000227 | 0.835504 |
| 32 | keble\_000245 | 0.817532 |
| 33 | magdalen\_000058 | 0.141947 |
| 34 | magdalen\_000078 | 0.0485862 |
| 35 | magdalen\_000560 | 0.069651 |
| 36 | oxford\_000317 | 0.420527 |
| 37 | oxford\_000545 | 0.438355 |
| 38 | oxford\_001115 | 0.069833 |
| 39 | oxford\_001752 | 0.685708 |
| 40 | oxford\_001753 | 0.553629 |
| 41 | oxford\_002416 | 0.355181 |
| 42 | oxford\_002562 | 0.213302 |
| 43 | oxford\_002734 | 0.616965 |
| 44 | oxford\_002904 | 0.670368 |
| 45 | oxford\_002985 | 0.342538 |
| 46 | oxford\_003335 | 0.0848855 |
| 47 | oxford\_003410 | 0.388528 |
| 48 | pitt\_rivers\_000033 | 0.56286 |
| 49 | pitt\_rivers\_000058 | 0.7215 |
| 50 | pitt\_rivers\_000087 | 0.717322 |
| 51 | pitt\_rivers\_000119 | 0.812273 |
| 52 | pitt\_rivers\_000153 | 0.36293 |
| 53 | radcliffe\_camera\_000095 | 0.451428 |
| 54 | radcliffe\_camera\_000519 | 0.701208 |
| 55 | radcliffe\_camera\_000523 | 0.740341 |
|  | Trung bình | 0.4745 |

### Kết quả sử dụng Harris affine region dectector kết hợp với SIFT descriptor

|  |  |  |
| --- | --- | --- |
| STT | Truy vấn | Độ chính xác |
| 1 | all\_souls\_000013 | 0.244759 |
| 2 | all\_souls\_000026 | 0.238601 |
| 3 | all\_souls\_000051 | 0.75079 |
| 4 | ashmolean\_000000 | 0.728409 |
| 5 | ashmolean\_000007 | 0.274433 |
| 6 | ashmolean\_000058 | 0.0721619 |
| 7 | ashmolean\_000269 | 0.113234 |
| 8 | ashmolean\_000305 | 0.461302 |
| 9 | balliol\_000051 | 0.625412 |
| 10 | balliol\_000167 | 0.095565 |
| 11 | balliol\_000187 | 0.239062 |
| 12 | balliol\_000194 | 0.319314 |
| 13 | bodleian\_000107 | 0.0795274 |
| 14 | bodleian\_000108 | 0.281791 |
| 15 | bodleian\_000132 | 0.737357 |
| 16 | bodleian\_000163 | 0.123504 |
| 17 | bodleian\_000407 | 0.106418 |
| 18 | christ\_church\_000179 | 0.650381 |
| 19 | christ\_church\_000999 | 0.504389 |
| 20 | christ\_church\_001020 | 0.503231 |
| 21 | cornmarket\_000019 | 0.716591 |
| 22 | cornmarket\_000047 | 0.711675 |
| 23 | cornmarket\_000105 | 0.17004 |
| 24 | cornmarket\_000131 | 0.45092 |
| 25 | hertford\_000015 | 0.750469 |
| 26 | hertford\_000027 | 0.720744 |
| 27 | hertford\_000063 | 0.792891 |
| 28 | keble\_000028 | 0.807536 |
| 29 | keble\_000055 | 0.280608 |
| 30 | keble\_000214 | 0.720102 |
| 31 | keble\_000227 | 0.673534 |
| 32 | keble\_000245 | 0.717043 |
| 33 | magdalen\_000058 | 0.128357 |
| 34 | magdalen\_000078 | 0.0205381 |
| 35 | magdalen\_000560 | 0.0839522 |
| 36 | oxford\_000317 | 0.467255 |
| 37 | oxford\_000545 | 0.49721 |
| 38 | oxford\_001115 | 0.0726797 |
| 39 | oxford\_001752 | 0.639384 |
| 40 | oxford\_001753 | 0.51793 |
| 41 | oxford\_002416 | 0.413962 |
| 42 | oxford\_002562 | 0.109343 |
| 43 | oxford\_002734 | 0.530667 |
| 44 | oxford\_002904 | 0.66757 |
| 45 | oxford\_002985 | 0.313656 |
| 46 | oxford\_003335 | 0.062391 |
| 47 | oxford\_003410 | 0.397696 |
| 48 | pitt\_rivers\_000033 | 0.563685 |
| 49 | pitt\_rivers\_000058 | 0.782454 |
| 50 | pitt\_rivers\_000087 | 0.68084 |
| 51 | pitt\_rivers\_000119 | 0.803412 |
| 52 | pitt\_rivers\_000153 | 0.289765 |
| 53 | radcliffe\_camera\_000095 | 0.450611 |
| 54 | radcliffe\_camera\_000519 | 0.69683 |
| 55 | radcliffe\_camera\_000523 | 0.744193 |
|  | Trung bình | 0.44720 |

### Kết quả sử dụng Hessian affine dectector kết hợp với rootSIFT descriptor

|  |  |  |
| --- | --- | --- |
| STT | Truy vấn | Độ chính xác |
| 1 | all\_souls\_000013 | 0.206126 |
| 2 | all\_souls\_000026 | 0.251789 |
| 3 | all\_souls\_000051 | 0.747347 |
| 4 | ashmolean\_000000 | 0.656052 |
| 5 | ashmolean\_000007 | 0.449561 |
| 6 | ashmolean\_000058 | 0.338013 |
| 7 | ashmolean\_000269 | 0.333826 |
| 8 | ashmolean\_000305 | 0.654974 |
| 9 | balliol\_000051 | 0.336456 |
| 10 | balliol\_000167 | 0.0894032 |
| 11 | balliol\_000187 | 0.153937 |
| 12 | balliol\_000194 | 0.460403 |
| 13 | bodleian\_000107 | 0.0637334 |
| 14 | bodleian\_000108 | 0.494598 |
| 15 | bodleian\_000132 | 0.740875 |
| 16 | bodleian\_000163 | 0.13916 |
| 17 | bodleian\_000407 | 0.12125 |
| 18 | christ\_church\_000179 | 0.688009 |
| 19 | christ\_church\_000999 | 0.585623 |
| 20 | christ\_church\_001020 | 0.727203 |
| 21 | cornmarket\_000019 | 0.756451 |
| 22 | cornmarket\_000047 | 0.686412 |
| 23 | cornmarket\_000105 | 0.162916 |
| 24 | cornmarket\_000131 | 0.518796 |
| 25 | hertford\_000015 | 0.679205 |
| 26 | hertford\_000027 | 0.754978 |
| 27 | hertford\_000063 | 0.77566 |
| 28 | keble\_000028 | 0.0804126 |
| 29 | keble\_000055 | 0.0751754 |
| 30 | keble\_000214 | 0.429494 |
| 31 | keble\_000227 | 0.314886 |
| 32 | keble\_000245 | 0.501001 |
| 33 | magdalen\_000058 | 0.106053 |
| 34 | magdalen\_000078 | 0.0321735 |
| 35 | magdalen\_000560 | 0.11303 |
| 36 | oxford\_000317 | 0.386246 |
| 37 | oxford\_000545 | 0.468502 |
| 38 | oxford\_001115 | 0.0452979 |
| 39 | oxford\_001752 | 0.70242 |
| 40 | oxford\_001753 | 0.234501 |
| 41 | oxford\_002416 | 0.581697 |
| 42 | oxford\_002562 | 0.105334 |
| 43 | oxford\_002734 | 0.629212 |
| 44 | oxford\_002904 | 0.763379 |
| 45 | oxford\_002985 | 0.355372 |
| 46 | oxford\_003335 | 0.0666472 |
| 47 | oxford\_003410 | 0.493071 |
| 48 | pitt\_rivers\_000033 | 0.672541 |
| 49 | pitt\_rivers\_000058 | 0.700136 |
| 50 | pitt\_rivers\_000087 | 0.678102 |
| 51 | pitt\_rivers\_000119 | 0.738379 |
| 52 | pitt\_rivers\_000153 | 0.172242 |
| 53 | radcliffe\_camera\_000095 | 0.581306 |
| 54 | radcliffe\_camera\_000519 | 0.716017 |
| 55 | radcliffe\_camera\_000523 | 0.629401 |
|  | Trung bình | 0.4354 |

### Kết quả sử dụng Hessian affine dectector kết hợp với SIFT descriptor

|  |  |  |
| --- | --- | --- |
| STT | Truy vấn | Độ chính xác |
| 1 | all\_souls\_000013 | 0.207897 |
| 2 | all\_souls\_000026 | 0.258809 |
| 3 | all\_souls\_000051 | 0.801271 |
| 4 | ashmolean\_000000 | 0.547882 |
| 5 | ashmolean\_000007 | 0.404973 |
| 6 | ashmolean\_000058 | 0.202242 |
| 7 | ashmolean\_000269 | 0.244479 |
| 8 | ashmolean\_000305 | 0.508966 |
| 9 | balliol\_000051 | 0.286282 |
| 10 | balliol\_000167 | 0.0877053 |
| 11 | balliol\_000187 | 0.138962 |
| 12 | balliol\_000194 | 0.317442 |
| 13 | bodleian\_000107 | 0.0566266 |
| 14 | bodleian\_000108 | 0.458588 |
| 15 | bodleian\_000132 | 0.769744 |
| 16 | bodleian\_000163 | 0.258068 |
| 17 | bodleian\_000407 | 0.0301638 |
| 18 | christ\_church\_000179 | 0.755518 |
| 19 | christ\_church\_000999 | 0.548242 |
| 20 | christ\_church\_001020 | 0.662017 |
| 21 | cornmarket\_000019 | 0.645882 |
| 22 | cornmarket\_000047 | 0.613575 |
| 23 | cornmarket\_000105 | 0.19797 |
| 24 | cornmarket\_000131 | 0.473692 |
| 25 | hertford\_000015 | 0.692094 |
| 26 | hertford\_000027 | 0.781952 |
| 27 | hertford\_000063 | 0.807928 |
| 28 | keble\_000028 | 0.0969412 |
| 29 | keble\_000055 | 0.120062 |
| 30 | keble\_000214 | 0.417875 |
| 31 | keble\_000227 | 0.449804 |
| 32 | keble\_000245 | 0.485647 |
| 33 | magdalen\_000058 | 0.111759 |
| 34 | magdalen\_000078 | 0.033857 |
| 35 | magdalen\_000560 | 0.0921974 |
| 36 | oxford\_000317 | 0.460362 |
| 37 | oxford\_000545 | 0.474064 |
| 38 | oxford\_001115 | 0.0462257 |
| 39 | oxford\_001752 | 0.696009 |
| 40 | oxford\_001753 | 0.181189 |
| 41 | oxford\_002416 | 0.585441 |
| 42 | oxford\_002562 | 0.0995242 |
| 43 | oxford\_002734 | 0.655671 |
| 44 | oxford\_002904 | 0.739954 |
| 45 | oxford\_002985 | 0.340076 |
| 46 | oxford\_003335 | 0.0600492 |
| 47 | oxford\_003410 | 0.496229 |
| 48 | pitt\_rivers\_000033 | 0.676386 |
| 49 | pitt\_rivers\_000058 | 0.741002 |
| 50 | pitt\_rivers\_000087 | 0.686951 |
| 51 | pitt\_rivers\_000119 | 0.766882 |
| 52 | pitt\_rivers\_000153 | 0.174113 |
| 53 | radcliffe\_camera\_000095 | 0.53593 |
| 54 | radcliffe\_camera\_000519 | 0.692693 |
| 55 | radcliffe\_camera\_000523 | 0.690569 |
|  | Trung bình | 0.4248 |

### Kết quả sử dụng SIFT detector kết hợp với rootSIFT descriptor

|  |  |  |
| --- | --- | --- |
| STT | Truy vấn | Độ chính xác |
| 1 | all\_souls\_000013 | 0.155222 |
| 2 | all\_souls\_000026 | 0.358848 |
| 3 | all\_souls\_000051 | 0.763639 |
| 4 | ashmolean\_000000 | 0.557451 |
| 5 | ashmolean\_000007 | 0.422939 |
| 6 | ashmolean\_000058 | 0.188411 |
| 7 | ashmolean\_000269 | 0.0882302 |
| 8 | ashmolean\_000305 | 0.540082 |
| 9 | balliol\_000051 | 0.620505 |
| 10 | balliol\_000167 | 0.239806 |
| 11 | balliol\_000187 | 0.117328 |
| 12 | balliol\_000194 | 0.46644 |
| 13 | bodleian\_000107 | 0.0684616 |
| 14 | bodleian\_000108 | 0.190789 |
| 15 | bodleian\_000132 | 0.860625 |
| 16 | bodleian\_000163 | 0.242954 |
| 17 | bodleian\_000407 | 0.103806 |
| 18 | christ\_church\_000179 | 0.765871 |
| 19 | christ\_church\_000999 | 0.616039 |
| 20 | christ\_church\_001020 | 0.768875 |
| 21 | cornmarket\_000019 | 0.516147 |
| 22 | cornmarket\_000047 | 0.722106 |
| 23 | cornmarket\_000105 | 0.179135 |
| 24 | cornmarket\_000131 | 0.286273 |
| 25 | hertford\_000015 | 0.757645 |
| 26 | hertford\_000027 | 0.610415 |
| 27 | hertford\_000063 | 0.783979 |
| 28 | keble\_000028 | 0.283481 |
| 29 | keble\_000055 | 0.13775 |
| 30 | keble\_000214 | 0.156363 |
| 31 | keble\_000227 | 0.50888 |
| 32 | keble\_000245 | 0.481087 |
| 33 | magdalen\_000058 | 0.0719374 |
| 34 | magdalen\_000078 | 0.0288762 |
| 35 | magdalen\_000560 | 0.140802 |
| 36 | oxford\_000317 | 0.556257 |
| 37 | oxford\_000545 | 0.51701 |
| 38 | oxford\_001115 | 0.0492437 |
| 39 | oxford\_001752 | 0.717711 |
| 40 | oxford\_001753 | 0.446098 |
| 41 | oxford\_002416 | 0.217604 |
| 42 | oxford\_002562 | 0.191553 |
| 43 | oxford\_002734 | 0.824227 |
| 44 | oxford\_002904 | 0.91767 |
| 45 | oxford\_002985 | 0.379342 |
| 46 | oxford\_003335 | 0.0537878 |
| 47 | oxford\_003410 | 0.533302 |
| 48 | pitt\_rivers\_000033 | 0.842361 |
| 49 | pitt\_rivers\_000058 | 0.846146 |
| 50 | pitt\_rivers\_000087 | 0.848475 |
| 51 | pitt\_rivers\_000119 | 0.824845 |
| 52 | pitt\_rivers\_000153 | 0.844049 |
| 53 | radcliffe\_camera\_000095 | 0.902511 |
| 54 | radcliffe\_camera\_000519 | 0.775305 |
| 55 | radcliffe\_camera\_000523 | 0.812756 |
|  | Trung bình | 0.4709 |

### Kết quả sử dụng SURF features

Sử dụng hàm trainSURF và QuerySURF

|  |  |  |
| --- | --- | --- |
| STT | Truy vấn | Độ chính xác |
|  | all\_souls\_000013 | 0.012821 |
|  | all\_souls\_000026 | 0.085684 |
|  | all\_souls\_000051 | 0.058013 |
|  | ashmolean\_000000 | 0.355481 |
|  | ashmolean\_000007 | 0.146963 |
|  | ashmolean\_000058 | 0.04 |
|  | ashmolean\_000269 | 0.04 |
|  | ashmolean\_000305 | 0.362077 |
|  | balliol\_000051 | 0.105556 |
|  | balliol\_000167 | 0.102183 |
|  | balliol\_000187 | 0.083333 |
|  | balliol\_000194 | 0.083333 |
|  | bodleian\_000107 | 0.052778 |
|  | bodleian\_000108 | 0.256071 |
|  | bodleian\_000132 | 0.061074 |
|  | bodleian\_000163 | 0.510069 |
|  | bodleian\_000407 | 0.168057 |
|  | christ\_church\_000179 | 0.24359 |
|  | christ\_church\_000999 | 0.060628 |
|  | christ\_church\_001020 | 0.187687 |
|  | cornmarket\_000019 | 0.342857 |
|  | cornmarket\_000047 | 0.111111 |
|  | cornmarket\_000105 | 0.111111 |
|  | cornmarket\_000131 | 0.111111 |
|  | hertford\_000015 | 0.210411 |
|  | hertford\_000027 | 0.132998 |
|  | hertford\_000063 | 0.258036 |
|  | keble\_000028 | 0.22619 |
|  | keble\_000055 | 0.519048 |
|  | keble\_000214 | 0.571429 |
|  | keble\_000227 | 0.521773 |
|  | keble\_000245 | 0.157483 |
|  | magdalen\_000058 | 0.024537 |
|  | magdalen\_000078 | 0.018519 |
|  | magdalen\_000560 | 0.024908 |
|  | oxford\_000317 | 0.073117 |
|  | oxford\_000545 | 0.027778 |
|  | oxford\_001115 | 0.021128 |
|  | oxford\_001752 | 0.222615 |
|  | oxford\_001753 | 0.083333 |
|  | oxford\_002416 | 0.139232 |
|  | oxford\_002562 | 0.014049 |
|  | oxford\_002734 | 0.185797 |
|  | oxford\_002904 | 0.007309 |
|  | oxford\_002985 | 0.170209 |
|  | oxford\_003335 | 0.018519 |
|  | oxford\_003410 | 0.001068 |
|  | pitt\_rivers\_000033 | 0.166667 |
|  | pitt\_rivers\_000058 | 0 |
|  | pitt\_rivers\_000087 | 0 |
|  | pitt\_rivers\_000119 | 0 |
|  | pitt\_rivers\_000153 | 0 |
|  | radcliffe\_camera\_000095 | 0.031136 |
|  | radcliffe\_camera\_000519 | 0.012051 |
|  | radcliffe\_camera\_000523 | 0 |
|  | Trung bình | 0.136926 |

# Lưu ý khi chạy chương trình.

## Runtime

* Cài đặt visual studio 2017 và phiên bản .net 4.7
* Cài đặt matlab 2017a trở lên hoặc runtime tương ứng.

## Các tập tin cần thiết

Trong thư mục oxford/feat có chứa các dữ liệu đã được train tương ứng với các dectector và descriptor, sử dụng các file này thay thế cho các file nằm trực tiếp trong oxford/feat để sử dụng dữ liệu mong muốn.. Toàn bộ chương trình có thể tải tại: <https://github.com/nguyenmanhphuc/VRA-classroom>. Cụ thể:

* Source code: [https://github.com/nguyenmanhphuc/VRA-classroom/tree/master/VRA-final](https://github.com/nguyenmanhphuc/VRA-classroom/VRA-final)
* Báo cáo: <https://github.com/nguyenmanhphuc/VRA-classroom/blob/master/VRA.Final.NguyenManhPhuc.pdf>, <https://github.com/nguyenmanhphuc/VRA-classroom/blob/master/VRA.Final.NguyenManhPhuc.docx>

Do các hính ảnh của dataset quá nặng nên không được tải lên cùng với source chương trình. Trước khi chạy chương trình có thể tải về tại: <http://www.robots.ox.ac.uk/~vgg/data/oxbuildings/> và chép các hình giải nén vào thư mục oxford/images trong project VRA-final.

# Tài liệu tham khảo

1. Thư viện lập trình Vlfeat: [http://www.vlfeat.org/index.html,](http://www.vlfeat.org/index.html) truy cập lần cuối vào ngày 10/12/2017.
2. Instance search, https://github.com/nvtiep/Instance-Search, truy cập lần cuối ngày 10/12/2017
3. Le Tan Hung, Các phép biến đổi Đồ hoạ Affine Transformations, [http://www.math.hcmuns.edu.vn/~tatuana/Do%20Hoa/Bai4B.pdf,](http://www.math.hcmuns.edu.vn/%7Etatuana/Do%20Hoa/Bai4B.pdf) truy cập ngày 09/12/2017
4. Mikolajczyk, K. and Schmid, C. 2004. Scale & affine invariant interest point detectors. International Journal on Computer Vision 60(1):63-86
5. Scale Invariant Feature Transform (SIFT), <http://www.vlfeat.org/api/sift.html>, truy cập lần cuối ngày 10/12/2017
6. <https://www.pyimagesearch.com/2015/04/13/implementing-rootsift-in-python-and-opencv/>, truy cập lần cuối ngày 28/12/2017