



# Mở rộng khả năng truy vấn đối tượng trong bộ dữ liệu ảnh khổng lồ

## Nhóm 1:

- Chung Kim Khánh (19127644)
- La Trường Phi (19127506)

**Giáo viên hướng dẫn:** PGS.TS. Lý Quốc Ngọc

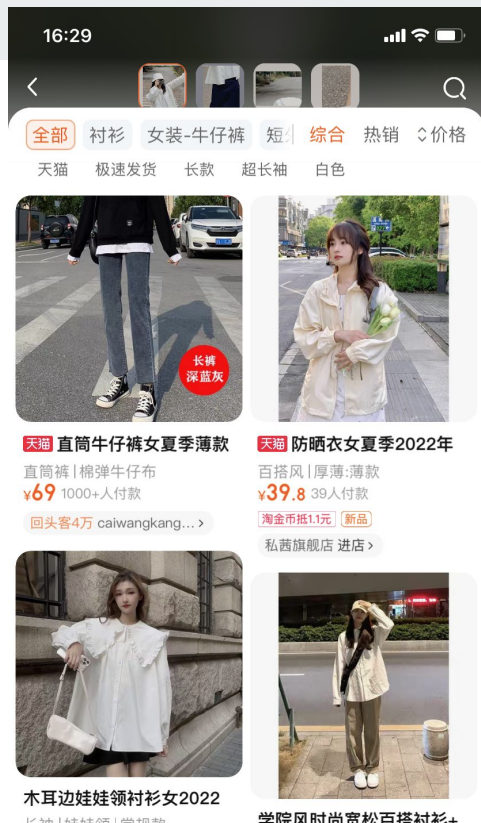
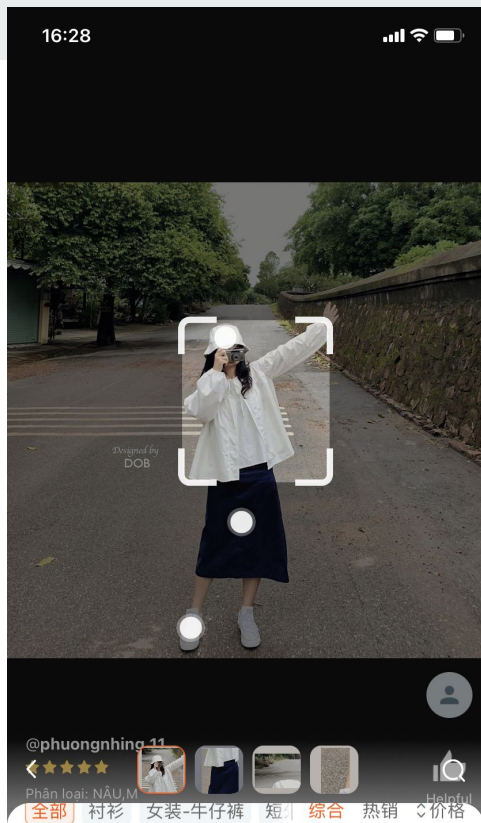
---

# I. Truy vấn quy mô lớn (Chapter 4)

# 1. Động lực nghiên cứu

- Người dùng đưa vào một hình ảnh
- Chọn đối tượng bất kỳ trên hình ảnh đó
- Đối tượng được bao bởi một bounding box xấp xỉ.
- Truy xuất ra những hình ảnh có đối tượng đó trong kho dữ liệu.

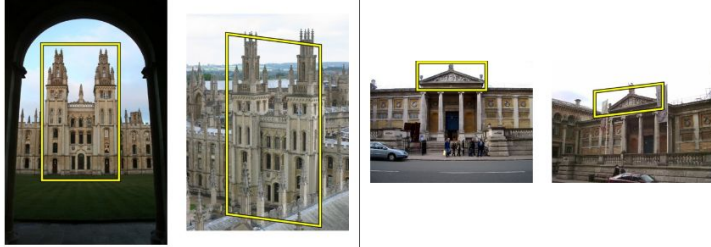
=> Đối mặt với 2 thách thức chính.



(i)



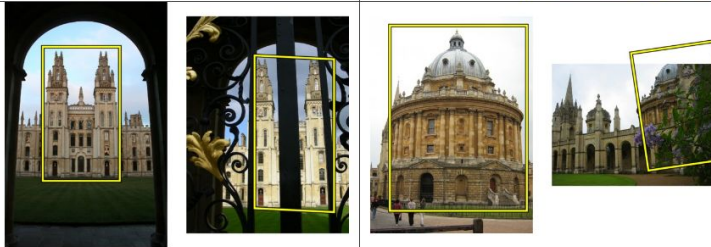
(ii)



(iii)



(iv)



## Thách thức về điều kiện hình ảnh trong thực tế (Điều kiện tự nhiên)

Ở đây là các ví dụ về các loại thách thức (khó khăn) được lấy ra trong bộ dữ liệu của Oxford:

(i) Thay đổi về tỷ lệ của đối tượng đối với khung hình (scale changes)

(ii) Thay đổi góc nhìn (viewpoint changes)

(iii) Thay đổi về điều kiện ánh sáng (lighting changes)

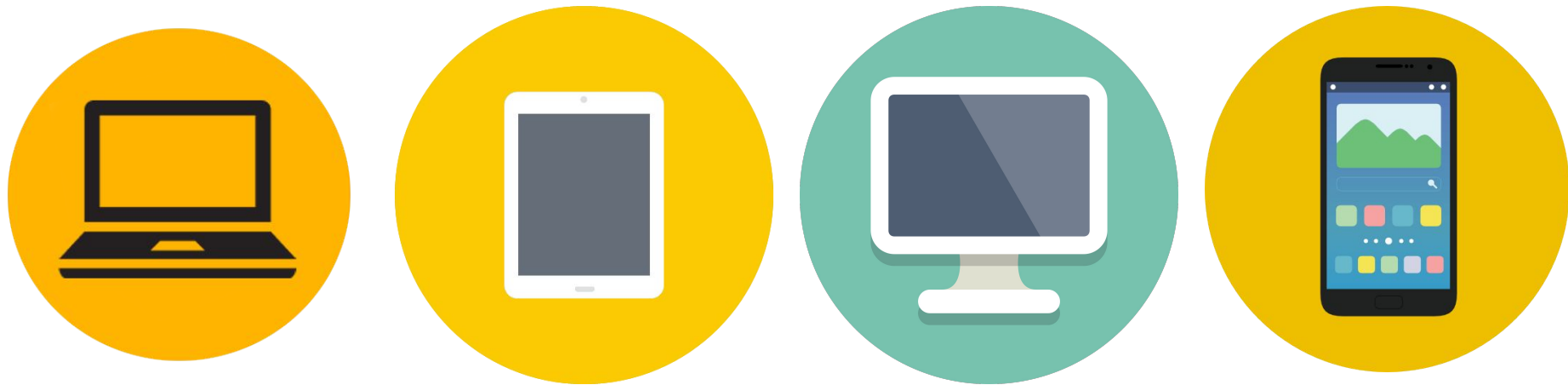
(iv) Che khuất một phần đối tượng (partial occlusions)

## Thách thức về tốc độ truy xuất



Khối lượng **dữ liệu lớn** nhưng **tốc độ** truy xuất ra kết quả **nhANH** (< 1 giây cho mỗi truy vấn)

## Hoạt động hiệu quả trên nhiều thiết bị khác nhau



Không có phần nào của quá trình xử lý hoặc truy xuất sẽ mất thời gian hơn tuyến tính trong kích thước của kho dữ liệu

## Tính tương đối

Không có phương pháp chính xác tuyệt đối để xác định đối tượng mà người dùng cần tìm có trong hình ảnh của kho dữ liệu không

**=> Mỗi phương pháp sẽ có quy tắc xếp hạng khác nhau**



## Nghiên cứu liên quan

15 năm qua, Truy xuất văn bản đã giải quyết và khắc phục được những vấn đề được nêu trên. Cụ thể là sự phát triển vượt bậc của Google.

Mô hình được đề cập: Mô hình giở từ (Chapter 2)

=> Như một bước thiết lập các ký tự cần truy xuất để giảm lượng tài liệu cần được truy xuất.







## Điểm khác biệt giữa truy xuất văn bản và hình ảnh

### Văn bản

Không cần mã hoá dữ liệu

*Ví dụ:* Một người gõ tìm kiếm 3 từ bất kỳ thì nó sẽ truy xuất ra các tài liệu có chứa 3 từ đó.

### Hình ảnh


Mã hoá dữ liệu -> nhiều ý nghĩa nội dung

*Ví dụ:* Tìm kiếm chiếc G63 sơn màu hồng được thiết kế độc quyền từ 1 hình ảnh người dùng chụp được. Rất khó để tìm được chính xác chiếc G63 của người dùng đưa ra.



## Tóm tắt về các phần

- (i) cải thiện khả năng mở rộng và chất lượng của lượng tử hóa từ trực quan (visual word quantization)
- (ii) đưa thông tin không gian vào bảng xếp hạng
- (iii) khắc phục các hiệu ứng lượng tử hóa được thấy khi sử dụng các từ vựng rất lớn
- (iv) các định hướng cho nghiên cứu trong tương lai.



## 2. Mở rộng quy mô lượng tử hoá (Scaling quantization)

Được đề cập ở chương 2, các hệ thống truy xuất dựa trên hình ảnh sẽ trích xuất các vectors đặc trưng sau đó phân cụm chúng thành một kho vocabulary gồm các visual words.

=> Nhiều hình ảnh -> Nhiều vector đặc trưng, nhiều dữ liệu.

Phương pháp K-means có hiệu quả nhưng lại tốn kém khi mở rộng quy mô cho các vocabularies lớn.

=> Nhiều nghiên cứu gần đây áp dụng phương pháp phân cấp cụm để tăng kích thước của các visual words -> Tăng độ chính xác.

=> Không dùng phương pháp phân cấp -> cải tiến phương pháp K-means -> *phương pháp xấp xỉ lân cận*.



## So sánh K-means, HKM, AKM

Tiêu chuẩn	K-means	HKM	AKM
Tốc độ	$O(K^2)$	$O(N \log K)$	$O(N \log K)$
Độ chính xác	Cao	Thấp	Cao



## Phân cụm K-means

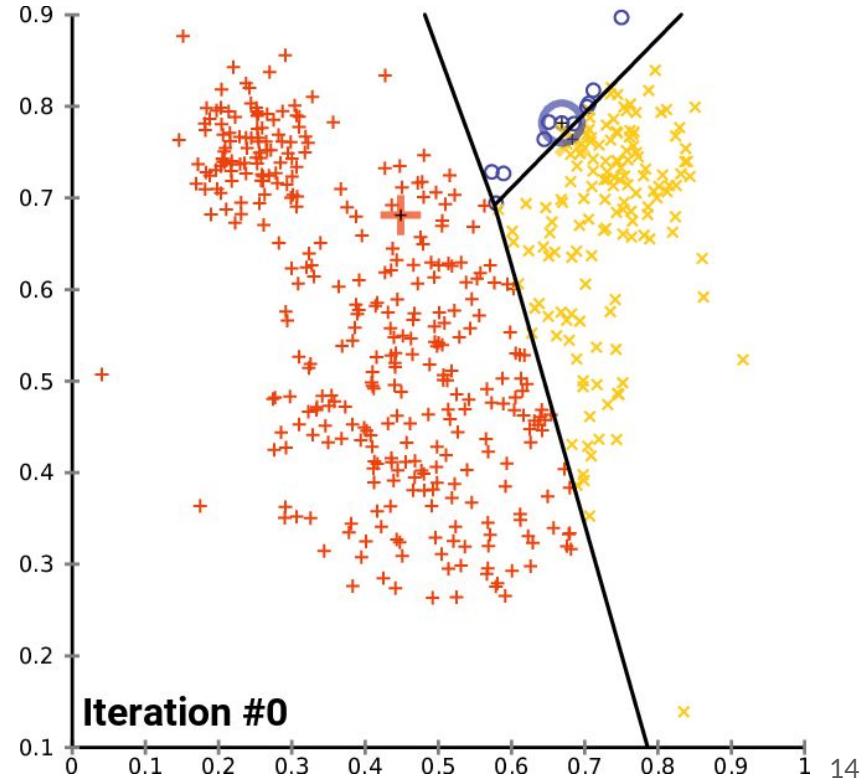
- **Input:** Dữ liệu  $X$  (chưa có nhãn) và  $K$  cụm dữ liệu (cluster)
- **Output:** Điểm trung tâm của mỗi cụm (center)

Phân dữ liệu theo từng cụm khác nhau sao cho ở mỗi cụm, các dữ liệu có cùng tính chất

## Phân cụm K-means

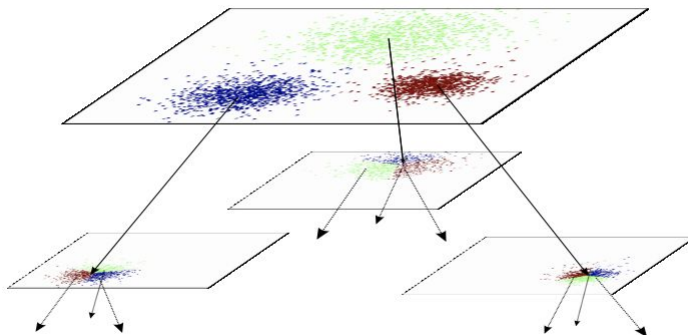
Sau phân cụm, ta thấy, đường phân định giữa các cụm (cluster) là **đường trung trực** giữa các điểm trọng tâm cụm (center) gần nhau.

=> Sử dụng toàn bộ các dữ liệu để xử lý.



## Phân cấp K-means - Hierarchical k-means (HKM)

Nistér và Stewénius đề xuất tạo một "đồ thị nhánh cây" bằng cách sử dụng lược đồ phân cụm k-means phân cấp (còn gọi là lượng tử hóa vector có cấu trúc cây). Thay vì giải quyết một chủ đề phân cụm với một số lượng lớn các cụm, một hệ thống phân cấp có tổ chức cây gồm các chủ đề phân cụm nhỏ hơn sẽ được giải quyết dễ dàng hơn (nhanh hơn).



Hệ số nhánh = 3  
Level 2  
 $\Rightarrow$  Số vector =  $3^2$



## Phân cấp K-means - Hierarchical k-means (HKM)

Làm giảm thiểu sự ảnh hưởng (tầm quan trọng) của lỗi trong quá trình lượng tử hoá (quantization) đối với các trường hợp điểm nằm gần vùng ranh giới giữa các cụm.

**Độ phức tạp thuật toán:**  $O(N \log K)$  -> Lớn hơn 1 triệu visual words và nhiều điểm đặc trưng.

=> Tuy nhiên, chúng ta sẽ thấy rằng các cụm từ do HKM tạo ra không tạo ra visual vocabularies tốt để truy xuất -> Khi đối tượng đã được phân vào sai cụm ngay từ ban đầu thì sẽ dẫn đến các lỗi sai sau của cụm đó.

=> **Độ chính xác thấp.**





## Xấp xỉ K-means (AKM)

Phương pháp này sẽ cải thiện độ chính xác của HKM nhưng vẫn giữ nguyên độ phức tạp thuật toán đối với tập dữ liệu lớn.



## Xấp xỉ k-trung bình (AKM)

Trong k-means truyền thống, phần lớn thời gian dành cho việc tính toán neighbour gần nhất giữa điểm đặc trưng (feature points) và cụm trung tâm (cluster center).

=> AKM thay bằng phương pháp xấp xỉ lân cận gần nhất, sử dụng một rừng gồm một số **cây k-d ngẫu nhiên** (a forest of several randomized k-d trees).



## k-d tree

BUILD-TREE( $node, ys$ )

```
1  if LEN( $ys$ ) = 1
2      then  $node \leftarrow$  NEW-LEAF-NODE( $ys[0]$ )
3      else  $dim, val \leftarrow$  CHOOSE-PSEUDORANDOM-SPLIT( $ys$ )
4            $ls, rs \leftarrow$  PARTITION-POINTS( $ys, dim, val$ )
5            $node \leftarrow$  NEW-INTERNAL-NODE( $dim, val$ )
6           BUILD-TREE( $node_{left}, ls$ )
7           BUILD-TREE( $node_{right}, rs$ )
```



## k-d tree

SEARCH-TREE-ONCE( $pq, node, dsq, query$ )

```
1  while IS-INTERNAL-NODE( $node$ )
2      do
3           $disc \leftarrow (query[node_{dim}] - node_{val})$ 
4          if  $disc \leq 0$ 
5              then  $node \leftarrow node_{left}$ 
6                  PRIORITY-QUEUE-INSERT( $pq, (node_{right}, dsq + disc^2)$ )
7          else  $node \leftarrow node_{right}$ 
8              PRIORITY-QUEUE-INSERT( $pq, (node_{left}, dsq + disc^2)$ )
9  return ( $node_{point}, \text{DISTANCE}(query, node_{point})$ )
```



## k-d tree

SEARCH-KNN(*forest*, *query*, *k*)

1 *pq*  $\leftarrow$  NEW-PRIORITY-QUEUE()

2 *dists*  $\leftarrow$  []

3 **for** each *tree*  $\in$  *forest*

4     **do** SEARCH-TREE-ONCE(*pq*, *tree*, 0, *query*)

5 **while** LEN(*dists*)  $\leq$  *nchecks*

6     **do** *node*, *dsq*  $\leftarrow$  POP(*pq*)

7         APPEND(*dists*, SEARCH-TREE-ONCE(*pq*, *node*, *dsq*, *query*))

8 **return** SELECT-K-SMALLEST(*dists*, *k*)



## Lưu ý

Trong thực tế, chúng ta lưu trữ nhiều điểm trong các nút lá của rừng k-d. Điều này làm giảm thời gian xây dựng và yêu cầu bộ nhớ và tăng vị trí bộ nhớ cache trong quá trình tìm kiếm.



## AKM giảm thiểu chính xác hàm chi phí K-means

Flat K-means cố gắng giảm thiểu hàm chi phí sau đây trên các cluster centers:

$$L(k, a_i) = \sum_{i=1}^N \|x_i - c_{a_i}\|^2$$

Bản cập nhật sau đây được tìm thấy thông qua sự khác biệt của L:

$$a_i^t = \operatorname{argmin}_{k_i} \|x_i - c_{k_i}^{t-1}\|^2$$

$$c_k^t = \frac{1}{|\{i : a_i^t = k\}|} \sum_{i: a_i^t = k} x_i$$



## AKM giảm thiểu chính xác hàm chi phí K-means

Flat K-means cố gắng giảm thiểu hàm chi phí sau đây trên các cluster centers:

$$L(k, a_i) = \sum_{i=1}^N \|x_i - c_{a_i}\|^2$$

Bản cập nhật sau đây được tìm thấy thông qua sự khác biệt của L:

$$a_i^t = \operatorname{argmin}_{k_i} \|x_i - c_{k_i}^{t-1}\|^2$$

$$c_k^t = \frac{1}{|\{i : a_i^t = k\}|} \sum_{i: a_i^t = k} x_i$$



# Đánh giá kết quả

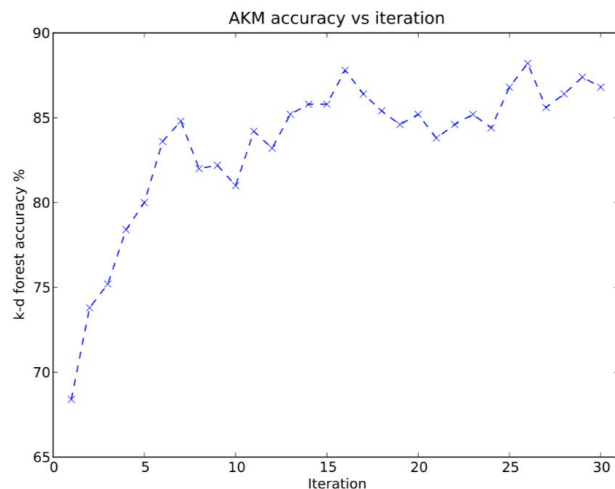
Độ chính xác của xấp xỉ AKM

Search accuracy in a random set of 100,000 SIFT descriptors												
Checks\Trees	1			2			4			8		
128	32.7%	43.2%	49.0 $\mu$ s	36.1%	46.4%	50.6 $\mu$ s	40.0%	50.1%	53.0 $\mu$ s	43.3%	53.9%	56.1 $\mu$ s
256	45.3%	57.5%	92.6 $\mu$ s	50.0%	61.6%	95.1 $\mu$ s	53.0%	64.9%	99.0 $\mu$ s	57.5%	68.5%	104.0 $\mu$ s
512	55.5%	68.8%	179.2 $\mu$ s	64.5%	76.5%	184.0 $\mu$ s	66.8%	78.1%	190.5 $\mu$ s	71.5%	81.5%	199.5 $\mu$ s
1024	67.5%	79.6%	352.7 $\mu$ s	73.8%	85.5%	361.9 $\mu$ s	79.1%	88.2%	373.7 $\mu$ s	83.2%	90.7%	388.5 $\mu$ s

Search accuracy in a random of set of 1,000,000 SIFT descriptors												
Checks\Trees	1			2			4			8		
128	20.9%	27.6%	72.5 $\mu$ s	25.4%	32.6%	74.6 $\mu$ s	27.8%	36.4%	77.3 $\mu$ s	28.8%	37.4%	82.5 $\mu$ s
256	28.5%	36.5%	128.5 $\mu$ s	34.6%	42.6%	131.7 $\mu$ s	37.8%	47.3%	136.6 $\mu$ s	40.6%	50.3%	143.2 $\mu$ s
512	36.8%	46.8%	240.4 $\mu$ s	45.3%	54.5%	245.5 $\mu$ s	50.6%	60.3%	252.9 $\mu$ s	52.9%	62.7%	263.8 $\mu$ s
1024	48.3%	60.1%	462.1 $\mu$ s	56.9%	67.8%	471.6 $\mu$ s	61.4%	71.4%	483.9 $\mu$ s	63.6%	73.6%	500.7 $\mu$ s

# Đánh giá kết quả

Hiệu suất truy xuất bằng cách sử dụng các từ vựng trực quan do AKM tạo ra

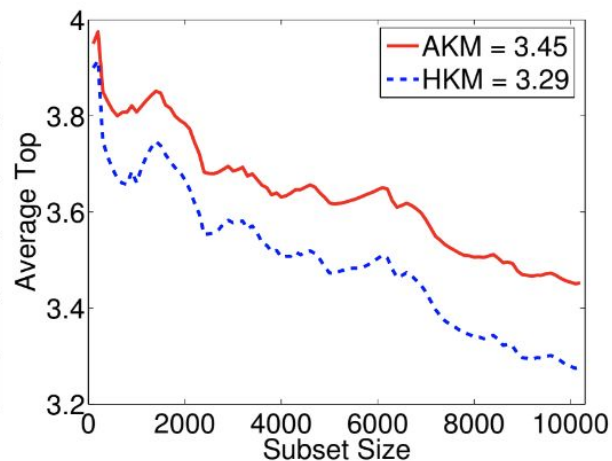


Clustering Parameters		mAP	
Number of descriptors	Vocabulary size	K-means	AKM
800K	10K	0.355	0.358
1M	20K	0.384	0.385
5M	50K	0.464	0.453
16.7M	1M		0.618

## Đánh giá kết quả

Hiệu suất truy xuất bằng cách sử dụng các từ vựng trực quan do AKM tạo ra

Method	# scoring levels	Average top
HKM	1	3.16
HKM	2	3.07
HKM	3	3.29
HKM	4	3.29
AKM		<b>3.45</b>





## Đánh giá kết quả

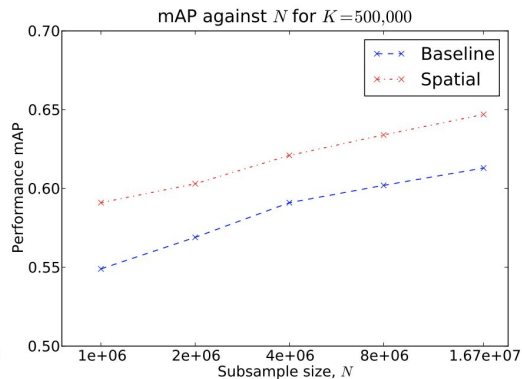
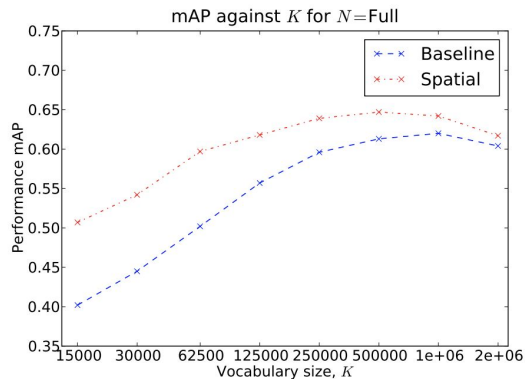
Hiệu suất truy xuất bằng cách sử dụng các từ vựng trực quan do AKM tạo ra

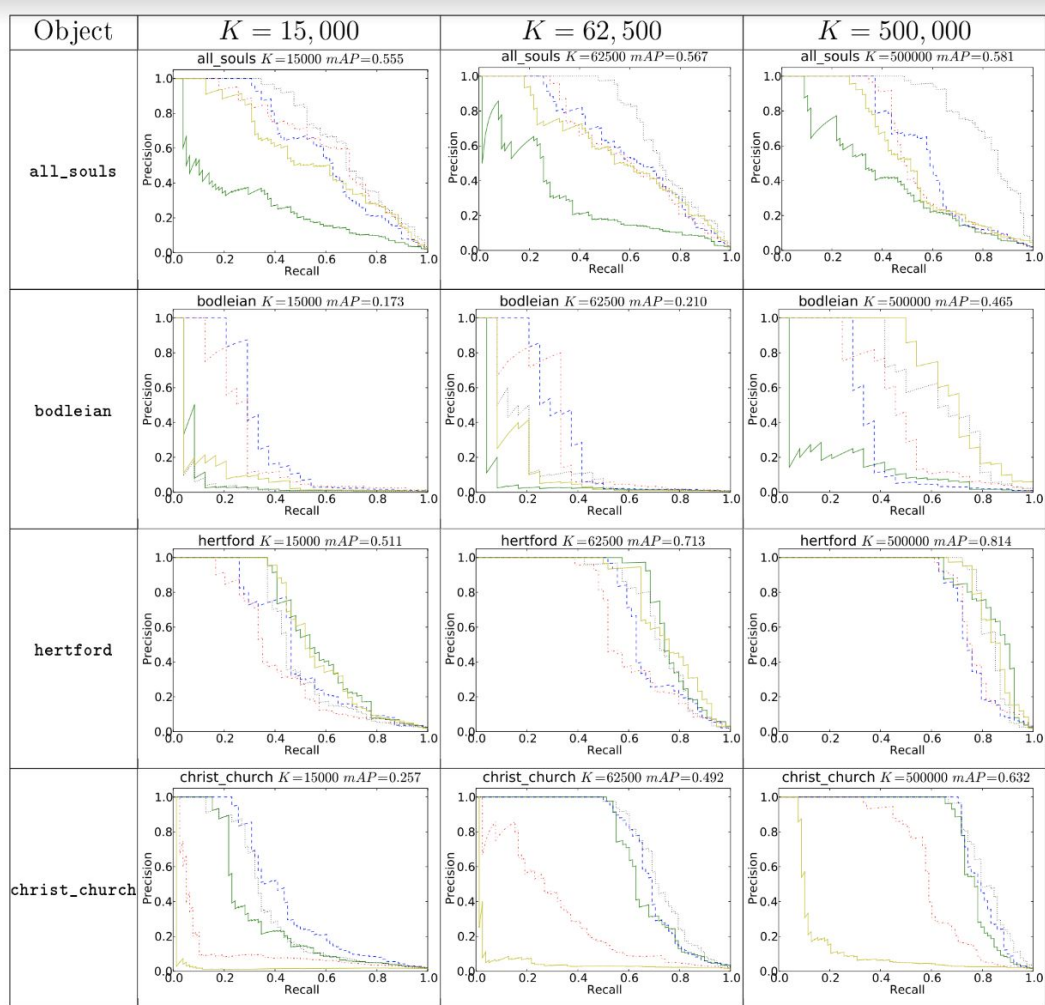
Method	Dataset	mAP	
		Bag-of-words	Spatial
FQM [157]	OXFORD	0.164	
HKM-1	OXFORD	0.439	0.469
HKM-2	OXFORD	0.418	
HKM-3	OXFORD	0.372	
HKM-4	OXFORD	0.353	
AKM	OXFORD	0.618	0.647
AKM	OXFORD-100K	0.490	0.541
AKM	OXFORD-1M	0.393	0.465

# Đánh giá kết quả

Khả năng khái quát hóa của các từ vựng lớn được tạo ra bởi AKM (hoặc các phương pháp khác).

$N \backslash K$	15,000	30,000	62,500	125,000	250,000	500,000	1,000,000	2,000,000
1M	0.380/0.497	0.413/0.516	0.439/0.536	0.486/0.573	0.530/0.588	0.549/0.591		
2M	0.393/0.506	0.425/0.528	0.466/0.551	0.503/0.579	0.556/0.600	0.569/0.603	0.591/0.619	
4M	0.387/0.502	0.419/0.525	0.466/0.569	0.525/0.590	0.557/0.608	0.591/0.621	0.594/0.616	0.594/0.607
8M	0.393/0.518	0.425/0.549	0.477/0.583	0.540/0.612	0.589/0.644	0.602/0.634	0.600/0.622	0.601/0.618
Full	0.402/0.507	0.445/0.542	0.502/0.597	0.557/0.618	0.596/0.639	0.613/0.647	0.620/0.642	0.604/0.617



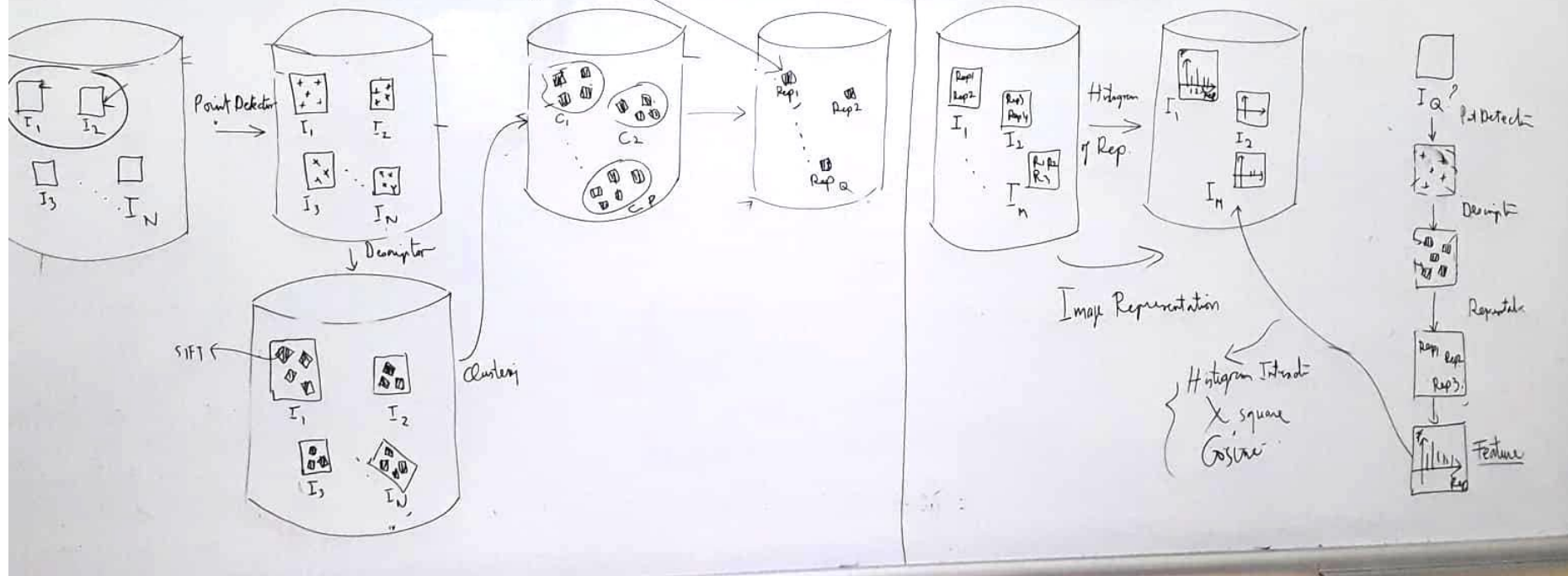




### 3. Sắp xếp lại không gian (Spatial re-ranking)

Tác giả khuyến khích và mô tả một phương pháp xếp hạng lại không gian để cải thiện hiệu suất truy xuất so với mô hình Bag-of-words tiêu chuẩn.

## Mô hình giỏ từ (Bag of Visual Word)



## Mô hình giỏ từ (Bag off Visual Word)





## **RANdom SAmple Consensus (RANSAC)**

Thuật toán RANdom SAmple Consensus (RANSAC) tiêu chuẩn để ước tính ma trận cơ bản đầy đủ 3-D hoặc phép đồng nhất xạ ảnh 2-D giữa hai hình ảnh là quá chung chung và chạy rất chậm.



## Đề xuất thay đổi

1. Tải nhận dạng, vị trí và hình dạng của từng từ trực quan của tài liệu. Nghiên cứu gần đây đã chỉ ra rằng hình dạng elip có thể được lượng tử hóa nếu cần nén nhiều hơn. Khi các từ vựng phân biệt được sử dụng, số lượng tương ứng phù hợp có thể khá ít.
2. Sử dụng các phép biến đổi phẳng đơn giản giữa hai hình ảnh (DOF, 4 DOF và 5 DOF, cùng với các loose thresholds trên khoảng cách bên trong). Do đó, ta vẫn có thể xấp xỉ kết nối một số cảnh có biến dạng phối cảnh đáng kể bằng cách sử dụng loose bound đối với lỗi truyền tải.



## Đề xuất thay đổi

3. Sử dụng hình elip cũng như vị trí của các interest points. Phép kết nối ellipse-ellipse có thể xác định các phép biến đổi lên đến 5 DOF. Điều này có nghĩa là chúng ta có thể tạo ra một giả thuyết từ một correspondence đơn ( $N = 1$  như ở trên). Trong thực tế, sau đó có thể chỉ cần liệt kê tất cả các correspondences và chọn cái có số lượng nội dung cao nhất.

4. Khi đã tìm thấy một phép đồng nhất tốt, ta lấy các giá trị nội tại được tìm thấy và sử dụng chúng để ước tính lại một phép đồng nhất hoàn chỉnh (6 DOF), sử dụng phương pháp bình phương nhỏ nhất.



## Tính toán một số đồng dạng khác nhau

**3 DOF:** Cho phép dịch chuyển và chia tỷ lệ đồng nhất.

**4 DOF:** Cho phép dịch chuyển và chia tỷ lệ dị hướng.

**5 DOF:** Cho phép dịch chuyển, chia tỷ lệ dị hướng và cắt bảo quản theo phương thẳng đứng.



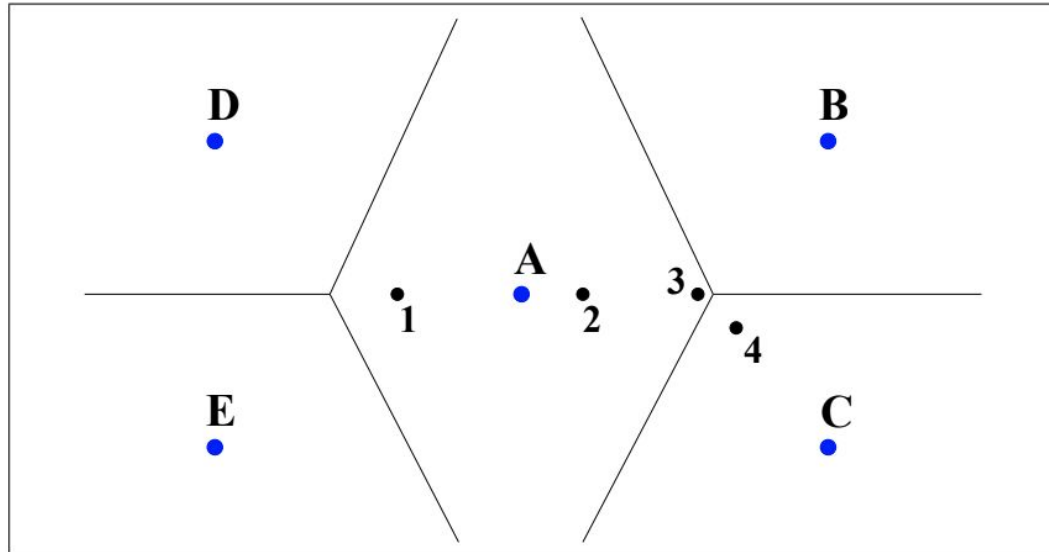
## **4. Mất mát trong quá trình lượng tử hoá (Lost in quantization)**



## Định nghĩa soft-assignment

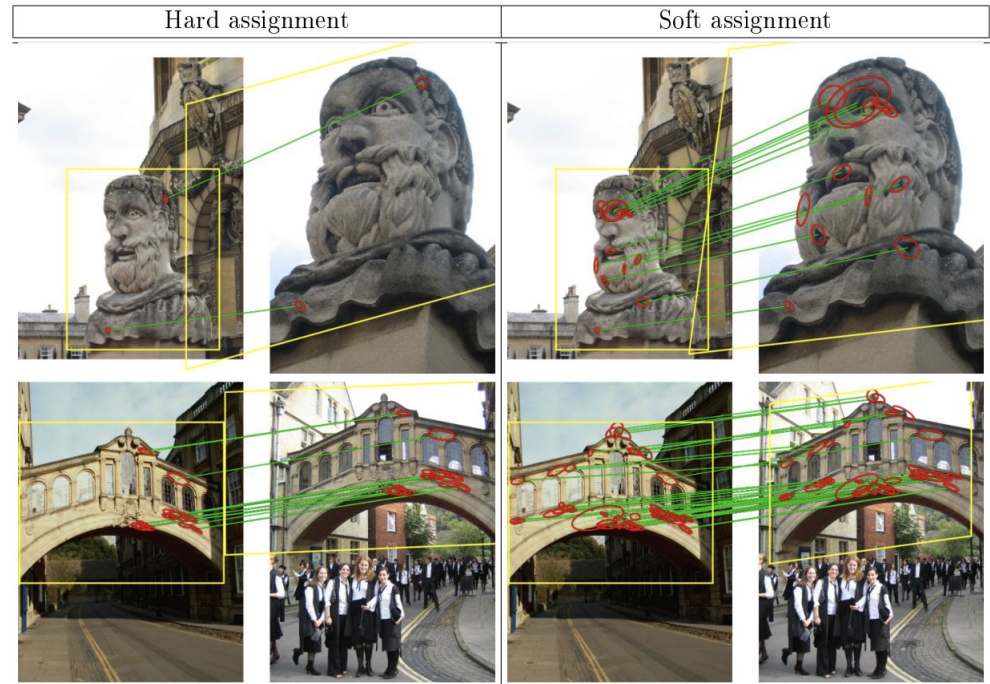
Thuật ngữ “**soft-assignment**” thường được sử dụng trong so sánh biểu đồ. Nó mô tả các kỹ thuật xác định một giá trị liên tục với sự kết hợp có trọng số của các bins lân cận hoặc làm mịn biểu đồ để số lượng trong bin được truyền sang các bins lân cận.

## Lợi ích của gán mềm (soft-assignment)



# Thiết lập

- Trọng số TF-IDF và gán mềm
- Xếp hạng lại không gian và gán mềm







## Đánh giá thực nghiệm

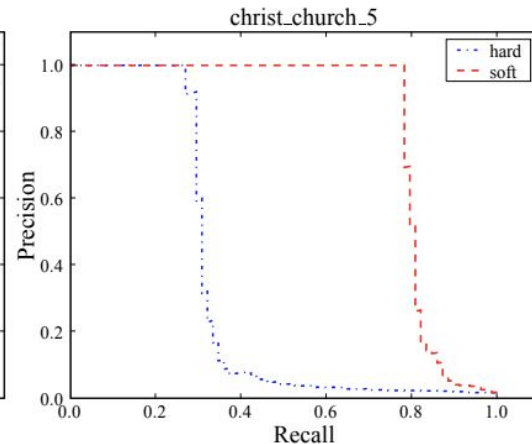
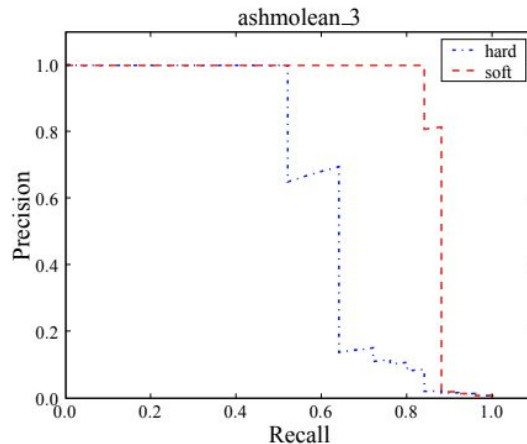
Biến thể tham số

		Training data	
$r$	$\sigma^2$	OXFORD	PARIS
3	5,000	0.671	0.495
3	6,250	0.673	0.494
3	7,500	0.672	0.493
5	5,000	0.674	0.502
5	6,250	0.673	0.499
5	7,500	0.673	0.496

# Đánh giá thực nghiệm

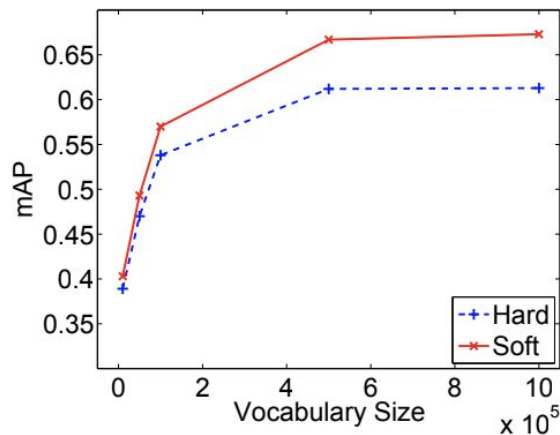
So sánh với các phương pháp khác

Method	Training data	
	OXFORD	PARIS
HKM [105] (1 level)	0.422	0.401
HKM [105] (2 level)	0.410	0.340
Hard [112]	0.614	0.403
Soft	<b>0.673</b>	<b>0.494</b>



# Đánh giá thực nghiệm

Ảnh hưởng của kích thước từ vựng



## Đánh giá thực nghiệm

Xác minh không gian và Tăng cường đến 100K hình ảnh

		Testing OXFORD		Testing OXFORD-100K	
	SP	Training data		Training data	
		OXFORD	PARIS	OXFORD	PARIS
Hard		0.614	0.403	0.498	0.290
Hard	×	0.653	0.460	0.565	0.385
Soft		0.673	0.493	0.534	0.343
Soft	×	<b>0.731</b>	<b>0.598</b>	<b>0.620</b>	<b>0.480</b>



## Tổng kết

Trong chương này đã trình bày **ba phương pháp** giúp mở rộng độ chính xác, khả năng mở rộng và tốc độ truy xuất đối tượng cụ thể trong các tập dữ liệu lớn:

- Large vocabularies do **AKM** tạo ra, đã được chứng minh là cải thiện đáng kể độ chính xác của việc truy xuất so với phương pháp phân cụm HKM hiện đại trước đây.
- Phương pháp để **spatial re-ranking** (xếp hạng lại không gian) sử dụng các tương ứng hình elip và các phép biến đổi giới hạn để đưa ra giả thuyết về các đồng phân phẳng cực kỳ nhanh chóng.
- Kỹ thuật **soft-assignment** được thiết kế để khắc phục một số tác hại của lỗi quantization gia tăng khi các vocabularies lớn được sử dụng, trong khi vẫn giữ được các lợi thế phân biệt.

---

## II. Tăng cường tìm sót (Recall) (Chapter 5)



## Giới thiệu

Trong quá trình truy vấn, chúng ta chỉ cần truy vấn các dữ liệu có quan hệ với query thay vì truy vấn tất cả dữ liệu hiện có vì điều đó sẽ gây hao phí bộ nhớ rất lớn. Như vậy chúng ta cần có giải thuật để đánh giá xem liệu dữ liệu đó có quan hệ với query của chúng ta hay không. Và nếu đó đúng là đối tượng cần truy vấn thì chúng ta phải truy vấn triệt để những đối tượng đó tránh trường hợp bỏ sót đối tượng khi hình ảnh bị thay đổi view, rotation, light condition,...



## Phát biểu bài toán

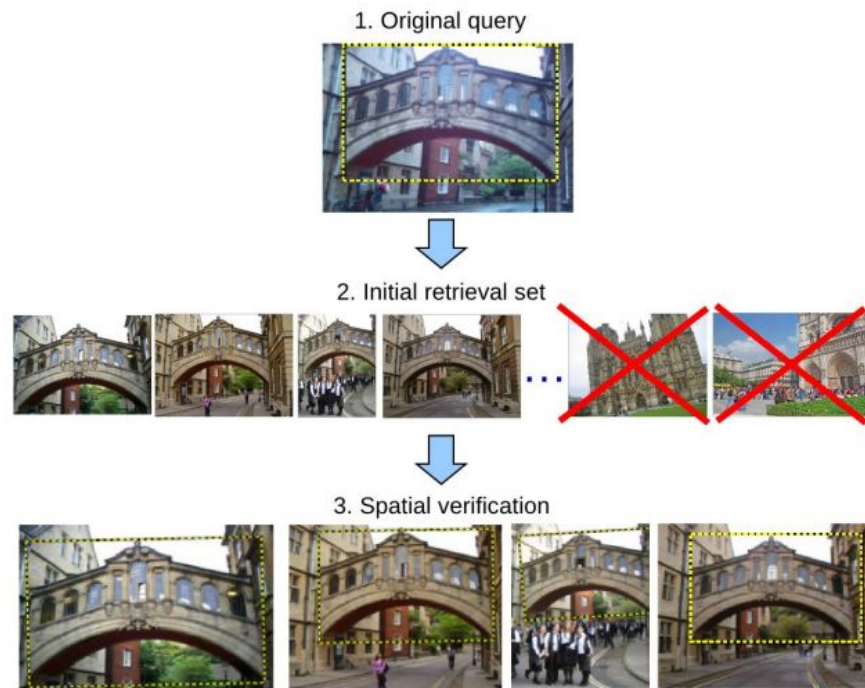
Ở phần trước, các giải thuật đã làm gia tăng giá trị precision. Trong quá trình truy vấn đối tượng, ta gia cần tăng khả năng tìm sót của thuật toán. Điều này sẽ làm tăng giá trị recall nhưng mục tiêu vẫn phải giữ nguyên giá trị precision. Để làm được điều này ta sẽ sử dụng các thông tin mở rộng của đối tượng bên trong kho dữ liệu rồi kết hợp với thông tin ban đầu để truy vấn. Các thông tin mở rộng sẽ bổ sung về mặt hình học trong không gian cho đối tượng chúng ta cần truy vấn



## Giải thuật

**\_B1:** Ta có 1 tập đối tượng truy vấn, đầu tiên ta sử dụng các kĩ thuật ở chapter 4 để truy vấn các ảnh có quan hệ với đối tượng trong query

**\_B2:** Với các ảnh trả về có quan hệ không gian với đối tượng trong query, ta kết hợp với dữ liệu trong query tạo thành 1 bộ dữ liệu cụ thể hơn cho đối tượng cần truy vấn. Điều này sẽ cho thấy được quan hệ hình học của đối tượng trong không gian



## Giải thuật

**\_B3:** Ta sử dụng bộ dữ liệu kết hợp của đối tượng tại bước 2, thực hiện tái truy vấn trên bộ dữ liệu ban đầu

**\_B4:** Lặp lại cho đến khi đạt được kết quả mong muốn, ta có thể thực hiện sàng lọc dữ liệu và tái truy vấn.

↓  
4. New enhanced query



↓  
5. Additional retrieved images





## Models types

Một mô hình đối tượng tiềm ẩn sẽ được xây dựng bằng các kết quả trả về từ việc truy vấn query Q0 và truy vấn mới Q1 hoặc các truy vấn khác được đưa ra, ở đây phát sinh ra 2 vấn đề:

- Trình tự nên kéo dài bao lâu thì mô hình đối tượng tiềm ẩn có thể được xây dựng từ truy vấn Q1 và 1 truy vấn khác
- Sẽ ra sao nếu danh sách kết quả xếp hạng trả về từ Q0 và Q1,... được kết hợp



## Query expansion baseline

Với phương pháp này được ứng dụng trong việc mở rộng query trong text-retrieval. Chúng ta lấy 5 phần tử trong query ban đầu. Sau đó ta thực hiện tính trung bình các vector tần số và thực hiện tái truy vấn đối với tập dữ liệu này. Sau đó thêm các kết quả từ kết quả truy vấn Q1 vào tập truy vấn Q0 ban đầu



## Transitive closure expansion

Một hàng đợi ưu tiên của các hình ảnh đã được xác minh được khóa bởi các từ nằm trong vùng truy vấn. Một mô hình tiềm ẩn sau đó được xây dựng từ số lượng các nội số. Sau đó, một hình ảnh được lấy từ đầu hàng đợi và khu vực tương ứng với vùng truy vấn ban đầu được sử dụng để đưa ra một truy vấn mới



## Average query expansion.

Một truy vấn mới được xây dựng bằng cách tính trung bình các kết quả xác định của truy vấn ban đầu. Đầu tiên, top  $m < 50$  kết quả được xác minh hàng đầu trả về bởi công cụ tìm kiếm được chọn. Một  $Q_{AVG}$  truy vấn mới sau đó được hình thành bằng cách lấy trung bình của truy vấn ban đầu  $Q_0$  và kết quả  $m$

$$d_{avg} = \frac{1}{m+1} \left( d_0 + \sum_{i=1}^m d_i \right),$$

trong đó  $d_0$  là một vector TF được chuẩn hóa của vùng truy vấn và  $d_i$  là vector TF được chuẩn hóa của kết quả thứ  $i$ . Một lần nữa, ta truy vấn lại một lần và kết quả của  $Q_{AVG}$  được thêm vào các kết quả của  $Q_0$ .



## Recursive average query expansion

Phương pháp này cải thiện phương pháp mở rộng truy vấn trung bình, bằng cách tạo các truy vấn đệ quy  $Q_i$  từ tất cả các kết quả xác định không gian được trả về. Phương pháp dừng lại một lần khi có nhiều hơn 30 hình ảnh được xác minh được tìm thấy hoặc sau khi không có hình ảnh mới nào được xác định tích cực.



## Multiple image resolution expansion.

Mô hình trong trường hợp này cũng tính đến xác suất quan sát một đặc trưng được đưa ra một hình ảnh của một đối tượng và độ phân giải của nó. Các đặc trưng bao gồm một khu vực nhỏ của đối tượng chỉ được nhìn thấy trong hình ảnh hoặc hình ảnh cận cảnh với độ phân giải cao. Tương tự, các đặc trưng bao gồm toàn bộ đối tượng không được nhìn thấy trên các chế độ xem chi tiết.



---

# III. Cải thiện phương pháp truy vấn

## Túi từ (Bag-of-Words)

### Mở rộng làm thêm



# Truy xuất đối tượng quy mô lớn gần thời gian thực

Phương pháp tiếp cận tiêu chuẩn:

- Biểu diễn một hình ảnh bằng cách sử dụng một Giỏ từ trực quan (BoW)
- Hình ảnh được xếp hạng bằng cách sử dụng thuật ngữ tần số nghịch đảo tần số tài liệu (tf-idf) -> Thông qua một chỉ mục nghịch đảo.



## Vấn đề

Một đối tượng trong hình ảnh đích có thể không được truy xuất vì một số lý do sử dụng tiêu chuẩn pipeline này:

- Dừng phát hiện đặc trưng.
- Mô tả nhiễu.
- Các thước đo không phù hợp để so sánh với bộ đặc tả.
- Mất do định lượng mô tả.

## Ba nội dung chính trong chương này

RootSIFT

Discriminative query expansion

Database-side feature augmentation

Sử dụng khoảng cách Hellinger thay vì khoảng cách Euclidean tiêu chuẩn để đo độ tương tự giữa các mô tả SIFT dẫn đến **tăng hiệu suất** mạnh mẽ trong tất cả các giai đoạn của pipeline -> thay đổi rất đơn giản chỉ **trong một vài dòng mã** và **không yêu cầu bất kỳ không gian lưu trữ bổ sung** nào vì việc chuyển **đổi từ SIFT sang RooSIFT** có thể được thực hiện trực tuyến.

Các vector BoW các vùng được xác minh không gian được sử dụng để đưa ra các truy vấn mới, giải quyết vấn đề bỏ qua việc xác định đặc trưng ngoài lượng từ hoá và nhiễu trên bộ mô tả. Sử dụng SVM tuyến tính để học phân biệt một vector trọng số để truy vấn lại mang lại **sự cải thiện đáng kể** so với phương pháp Mở rộng truy vấn trung bình tiêu chuẩn (Chum et al., 2007b), trong khi  **duy trì tốc độ truy xuất ngay lập tức** thông qua việc sử dụng hiệu quả Chỉ số đảo ngược.

Hạn chế chính của việc mở rộng truy vấn là dựa vào truy vấn để mang lại đủ số lượng kết quả có độ chính xác cao ngay từ đầu.

**Tăng cường tính năng bên cơ sở dữ liệu** (Turcot và Lowe, 2009) là một bổ sung tự nhiên để mở rộng truy vấn -> hiệu quả nhưng không tính đến cấu trúc không gian của các đặc trưng tăng cường.


=> Sử dụng xác minh không gian bằng phương pháp đồng nhất.



## Tóm lại

Trong mỗi trường hợp, các phương pháp này tăng đáng kể hiệu suất truy xuất và có thể đơn giản được “kết nối vào” kiến trúc truy vấn đối tượng tiêu chuẩn của Philbin et al. (2007) (BoW, chỉ số đảo ngược, tf-idf, xếp hạng lại tính nhất quán trong không gian) mà không làm tăng thời gian xử lý.

*Ví dụ:* RootSIFT và mở rộng truy vấn phân biệt thậm chí không làm tăng yêu cầu lưu trữ.



# Hệ thống truy xuất cơ sở

Baseline retrieval system

Tác giả tuân theo framework truy vấn BoW tiêu chuẩn được mô tả trong (Philbin và cộng sự, 2007).

Sử dụng interest points affine-Hessian (Mikolajczyk và Schmid, 2004b), một vocabulary gồm 1 triệu vision words thu được bằng cách sử dụng giá trị K-means gần đúng (AKM) và xếp hạng lại không gian của 200 kết quả tf-idf hàng đầu bằng cách sử dụng phép biến đổi affine.

Việc triển khai hệ thống gần đây nhất của tác giả đạt được mAP là **0,672** trên bộ dữ liệu 5K của Oxford so với **0,657** ban đầu của Philbin và cộng sự (2007) => Đây là hệ thống cơ sở mà chúng tôi sẽ so sánh khi tác giả giới thiệu các phương pháp mới trong phần tiếp theo.

Triển khai gần đây nhất của tác giả về phương pháp mở rộng truy vấn trung bình từ Chum et al. (2007b) đạt được mAP là **0,726** trên Oxford 105K so với **0,711** ban đầu (Chum và cộng sự, 2007b).

**Lưu ý:** mặc dù *bài báo gốc đã mô tả một số phương pháp để mở rộng truy vấn* (ví dụ: đóng bắc cầu - transitive closure, nhiều độ phân giải hình ảnh - multiple image resolution), phương pháp trung bình đã trở thành tiêu chuẩn để so sánh với (Chum et al., 2011, Mikulik et al., 2010, Philbin et al., 2008) => **nó có hiệu suất tương tự như các phương thức khác và thời gian chạy nhanh hơn vì các phương pháp khác liên quan đến việc đưa ra một số truy vấn mới**. Do đó, tác giả sử dụng nó làm đường cơ sở để mở rộng truy vấn trong các so sánh tiếp theo.

Vì lý do nhất quán (sử dụng **cùng một từ vựng trực quan** và **các tham số khác nhau** của việc sắp xếp lại không gian và mở rộng truy vấn), tác giả so sánh các cải tiến của tác giả với việc triển khai hệ thống cơ sở gần đây nhất của tác giả.




## RootSIFT: Khoảng cách Hellinger cho SIFT

Phương pháp này nổi tiếng với các lĩnh vực như **phân loại kết cấu và phân loại hình ảnh**, rằng việc sử dụng khoảng cách Euclid để so sánh histograms thường mang lại hiệu suất kém hơn so với sử dụng các thước đo như  $\chi^2$  (Chi - bình phương) hoặc Hellinger.

SIFT ban đầu được thiết kế để sử dụng với khoảng cách Euclidean (Lowe, 2004), nhưng vì nó là histogram nên câu hỏi tự nhiên nảy sinh là liệu nó có được lợi khi sử dụng các thước đo khoảng cách histogram thay thế hay không?

=> Tác giả cho thấy rằng việc sử dụng nhân Hellinger thực sự mang lại một lợi ích to lớn.





Giả sử  $\mathbf{x}$  và  $\mathbf{y}$  là  $n$  vectơ có đơn vị Euclid chuẩn ( $\|\mathbf{x}\|_2 = 1$ ), thì khoảng cách Euclid  $d_E(\mathbf{x}, \mathbf{y})$  giữa chúng liên quan đến độ tương tự của chúng (kernel)  $S_E(\mathbf{x}, \mathbf{y})$  là:

$$d_E(\mathbf{x}, \mathbf{y})^2 = \|\mathbf{x} - \mathbf{y}\|_2^2 = \|\mathbf{x}\|_2^2 + \|\mathbf{y}\|_2^2 - 2\mathbf{x}^T\mathbf{y} = 2 - 2S_E(\mathbf{x}, \mathbf{y})$$

trong đó  $S_E(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T\mathbf{y}$ , và bước cuối cùng theo sau từ  $\|\mathbf{x}\|_2^2 = \|\mathbf{y}\|_2^2 = 1$ . Ở đây tác giả quan tâm đến việc thay thế Euclid similarity/kernel bằng Hellinger kernel.

**=> Hữu ích khi sử dụng kết nối tiêu chuẩn giữa khoảng cách (số liệu) và kernels**



## Hellinger kernel

**Hellinger kernel**, còn được gọi là hệ số Bhattacharyya, cho hai histograms chuẩn hóa L1,  $x$  và  $y$  (Ví dụ:  $\sum_{i=1}^n x_i = 1$  và  $x_i \geq 0$ ), được định nghĩa là:

$$H(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^n \sqrt{x_i y_i}$$



# SIFT

Các vector SIFT có thể được so sánh bởi một nhân Hellinger bằng cách sử dụng một thao tác đại số đơn giản theo hai bước:

(i) L1 chuẩn hóa vector SIFT (ban đầu nó có đơn vị L2 chuẩn).

(ii) căn bậc hai mỗi phần tử.

Sau đó,  $S_E(\sqrt{x}, \sqrt{y}) = \sqrt{x}^T \sqrt{y} = H(x, y)$ , và các vector kết quả là L2 chuẩn hóa vì  $S_E(\sqrt{x}, \sqrt{x}) = \sum_{i=1}^n x_i = 1$ .

=> Định nghĩa một bộ đặc tả mới là RootSIFT - phần tử căn bậc hai của các vector SIFT chuẩn hóa L1.

**Điểm mấu chốt:** so sánh các bộ đặc tả RootSIFT sử dụng khoảng cách Euclidean tương đương với việc sử dụng nhân Hellinger để so sánh các vector SIFT ban đầu:  $d_E(\sqrt{x}, \sqrt{y})^2 = 2 - 2H(x, y)$ .



# RootSIFT

RootSIFT được sử dụng trong đường dẫn truy xuất đối tượng cụ thể bằng cách thay thế SIFT bằng RootSIFT tại mọi điểm. Các bộ đặc tả RootSIFT được so sánh bằng cách sử dụng khoảng cách Euclide.

=> Mọi bước có thể được sửa đổi dễ dàng:

- K-means để xây dựng từ vựng trực quan (vì nó dựa trên khoảng cách Euclid) -> vẫn có thể được sử dụng.
- Các phương pháp lân cận gần nhất (cần thiết cho các hệ thống có từ vựng rất lớn) -> vẫn có thể được sử dụng.
- Gán mềm các bộ mô tả cho các từ trực quan (Jégou và cộng sự, 2010a, Philbin và cộng sự, 2008)
- Mở rộng truy vấn và các phần mở rộng khác chỉ yêu cầu khoảng cách Euclide trên SIFT (Jégou và cộng sự, 2008, 2010b, Mikulik và cộng sự, 2010, Philbin và cộng sự, 2010).

## Hiệu suất truy xuất (mAP) của các phương pháp đề xuất khác nhau

Retrieval Method	SIFT		RootSIFT	
	Ox5k	Ox105k	Ox5k	Ox105k
Philbin et al. (2007): tf-idf ranking	0.636	0.515	0.683	0.581
Philbin et al. (2007): tf-idf with spatial reranking	0.672	0.581	0.720	0.642
Chum et al. (2007b): average query expansion (AQE)	0.839	0.726	0.850	0.756
Turcot and Lowe (2009): database-side feature augmentation (AUG)	0.776	0.711	0.827	0.759
This chapter: discriminative query expansion (DQE)	<b>0.847</b>	0.752	0.861	0.781
This chapter: spatial database-side feature augmentation (SPAUG)	0.785	0.723	0.838	0.767
This chapter: SPAUG + DQE	0.844	<b>0.795</b>	<b>0.881</b>	<b>0.823</b>

Tác giả sử dụng cách triển khai của tác giả đối với tất cả các phương pháp đã được liệt kê (Chum và cộng sự, 2007b, Philbin và cộng sự, 2007, Turcot và Lowe, 2009) để so sánh chúng một cách nhất quán bằng cách **sử dụng các từ vựng và bộ thông số trực quan giống nhau**.

=> RootSIFT **tốt hơn** so với SIFT cho tất cả các phương pháp được kiểm tra. Các từ vựng được tạo bằng cách sử dụng bộ mô tả Oxford 5K và tất cả các phương pháp ngoại trừ “xếp hạng tf-idf” đều sử dụng **sắp xếp lại** không gian của 200 kết quả hàng đầu.

**Lưu ý:** đối với AUG và SPAUG, tác giả tính toán lại idf ở phần sau.



## Đánh giá

Sự cải thiện đáng kể về hiệu suất được thể hiện trong bảng trên. Trong đó, đối với mỗi bước (ví dụ: thêm mở rộng truy vấn, thêm tăng cường tính năng) sử dụng SIFT so sánh với sử dụng RootSIFT.

**Ví dụ:** trên Oxford 105K, hệ thống cơ sở (chỉ tf-idf) tăng hiệu suất từ *0,515 lên 0,581* và với việc sắp xếp lại không gian có sự cải thiện từ *0,581 lên 0,642*.

=> Những cải tiến này hầu như **không có chi phí bổ sung** và **không có thêm dung lượng lưu trữ** vì SIFT có thể được chuyển đổi trực tuyến thành RootSIFT với chi phí xử lý không đáng kể.



## Kết luận

Phép biến đổi RootSIFT có thể được coi là **một feature map rõ ràng** từ không gian SIFT ban đầu -> không gian RootSIFT.

sao cho việc thực hiện tích vô hướng (tức là một nhân tuyến tính) trong không gian RootSIFT tương đương với việc tính toán Hellinger kernel trong không gian gốc.

=> Cách tiếp cận này đã được khám phá trong bối cảnh của kernel map cho bộ phân loại SVM bởi (Perronnin và cộng sự, 2010b, Vedaldi và Zisserman, 2010). Những feature maps rõ ràng có thể được xây dựng cho các kernels phụ khác, chẳng hạn như  $\chi^2$  (Chi - bình phương). Tuy nhiên, tác giả nhận thấy có chút khác biệt về hiệu suất so với Hellinger kernel khi được sử dụng trong hệ thống truy xuất đối tượng cụ thể.



# Kết luận

Tác dụng của ánh xạ RootSIFT:

↓ các giá trị **bin lớn** hơn so với các giá trị **bin nhỏ** hơn

Lý do:

Khoảng cách Euclid giữa các vector SIFT ban đầu có thể **bị chi phối bởi các giá trị lớn** này. Sau khi ánh xạ, khoảng cách có tầm ảnh hưởng mạnh hơn với các giá trị bin nhỏ hơn.






## Kết luận

Các công trình trước đây đã so sánh các vector SIFT với các khoảng cách khác Euclidean, nhưng không sử dụng feature map rõ ràng. Vì vậy, lợi ích của việc có thể tiếp tục sử dụng các thuật toán với khoảng cách Euclid (ví dụ: K-mean) là không rõ ràng.

Ví dụ:

Tác giả	Nội dung
Johnson (2010)	sử dụng phân kỳ của Jeffrey để so sánh các vector SIFT tập trung vào nén bộ mô tả
Pele và Werman (2008)	sử dụng biến thể của Khoảng cách của Earth Mover
Pele và Werman (2010)	số liệu bậc hai $\chi^2$ (Chi - bình phương)



# Mở rộng truy vấn mang tính phân biệt

Discriminative query expansion



## Mở rộng truy vấn trung bình

Mở rộng truy vấn có thể cải thiện đáng kể hiệu suất của hệ thống truy xuất. Phương pháp mở rộng truy vấn trung bình tiến hành như sau:

- Cho một vùng truy vấn, hình ảnh được xếp hạng bằng cách sử dụng điểm tf-idf.
- Xác minh không gian được thực hiện trên một danh sách ngắn các kết quả được xếp hạng cao, đồng thời cung cấp vị trí (ROI) của đối tượng truy vấn trong hình ảnh được truy xuất.
- Các vector BoW tương ứng với các từ trong các ROI này được tính trung bình cùng với query BoW.
- Vector BoW mở rộng truy vấn kết quả này được sử dụng để tái truy vấn cơ sở dữ liệu.



## Cách tiếp cận phân biệt để mở rộng truy vấn trong đó dữ liệu phủ định được tính đến và huấn luyện phân loại

- Các vector BoW được sử dụng để làm giàu truy vấn **được thu thập theo cách chính xác** giống như đối với mở rộng truy vấn trung bình. Chúng cung cấp dữ liệu đào tạo tích cực và hình ảnh có điểm tf-idf thấp cung cấp dữ liệu đào tạo tiêu cực.
- Một SVM tuyến tính được đào tạo bằng cách **sử dụng các vector BoW dương và âm** này để **thu được vector trọng số  $w$** .
- Vector trọng số đã học được sử dụng để **xếp hạng hình ảnh** theo khoảng cách của chúng từ ranh giới quyết định, tức là nếu hình ảnh được biểu diễn bằng vector BoW  $x$ , thì hình ảnh được sắp xếp theo giá trị  $w^T x$ . -> Sử dụng chỉ số đảo ngược giống như khi tính toán điểm tf-idf - cả hai phép toán chỉ là tích vô hướng giữa một vector và  $x$ .
- Đối với điểm tf-idf:

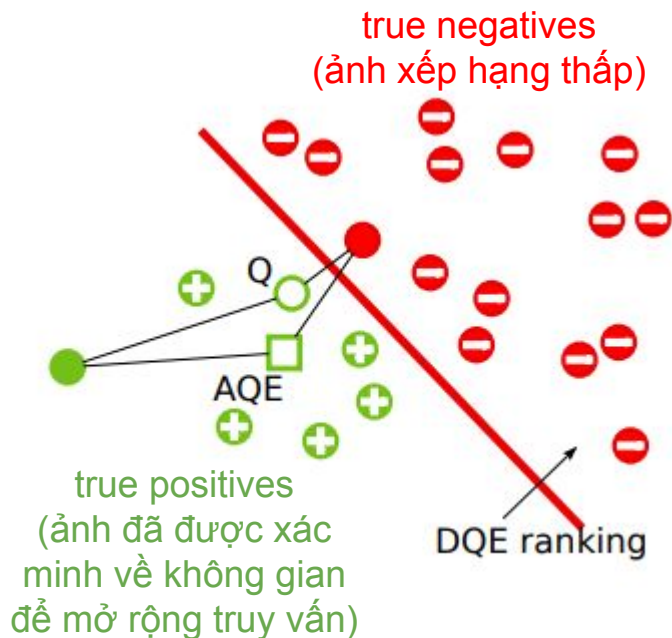
Phương pháp	Mở rộng truy vấn trung bình	Mở rộng truy vấn phân biệt (DQE)
Vector	BoW có trọng số idf truy vấn trung bình	Trọng số đã học $w$



## Lưu ý

Để DQE hoạt động hiệu quả -> vector trọng số phải thưa thớt.

## Hình minh họa không gian đặc trưng BoW



Q và AQE lần lượt biểu thị truy vấn và vector BoW mở rộng truy vấn trung bình. Xếp hạng tf-idf AQE sắp xếp hình ảnh dựa trên khoảng cách của chúng đến vector AQE, trong khi xếp hạng DQE sắp xếp hình ảnh theo khoảng cách đã ký từ ranh giới quyết định. Như minh họa ở đây, DQE xếp hạng chính xác hai hình ảnh có nhãn không xác định trong khi AQE thì không.

Bằng cách lựa chọn cẩn thận dữ liệu phủ định, vector trọng số thu được này ít nhất là thừa thớt như vector được sử dụng trong mở rộng truy vấn trung bình. Do đó, phương pháp này ít nhất cũng hiệu quả về mặt tính toán như mở rộng truy vấn trung bình với chi phí đào tạo SVM tuyến tính không đáng kể. Hình minh họa bằng sơ đồ cách dữ liệu phủ định có thể mang lại lợi ích cho DQE so với việc mở rộng truy vấn trung bình.

## Bảng so sánh đã đề cập trước đó

Retrieval Method	SIFT		RootSIFT	
	Ox5k	Ox105k	Ox5k	Ox105k
Philbin et al. (2007): tf-idf ranking	0.636	0.515	0.683	0.581
Philbin et al. (2007): tf-idf with spatial reranking	0.672	0.581	0.720	0.642
Chum et al. (2007b): average query expansion (AQE)	0.839	0.726	0.850	0.756
Turcot and Lowe (2009): database-side feature augmentation (AUG)	0.776	0.711	0.827	0.759
This chapter: discriminative query expansion (DQE)	<b>0.847</b>	0.752	0.861	0.781
This chapter: spatial database-side feature augmentation (SPAUG)	0.785	0.723	0.838	0.767
This chapter: SPAUG + DQE	0.844	<b>0.795</b>	<b>0.881</b>	<b>0.823</b>

=> Có thể thấy rằng DQE luôn tốt hơn so với AQE. Hiệu suất đạt được đặc biệt rõ ràng khi tăng kích thước tập dữ liệu - đối với Oxford 5K DQE tốt hơn AQE 1% và 1,3% đối với SIFT và RootSIFT, trong khi đối với Oxford 105k mAP cải thiện 3,6% và 3,3%.



## Chi tiết triển khai

- Vector BoW tương ứng với mỗi hình ảnh trước tiên được cắt bớt để chỉ bao gồm các từ xuất hiện trong ít nhất một ví dụ tích cực.
- Bộ phân loại là một SVM tuyến tính được đào tạo với LIBSVM

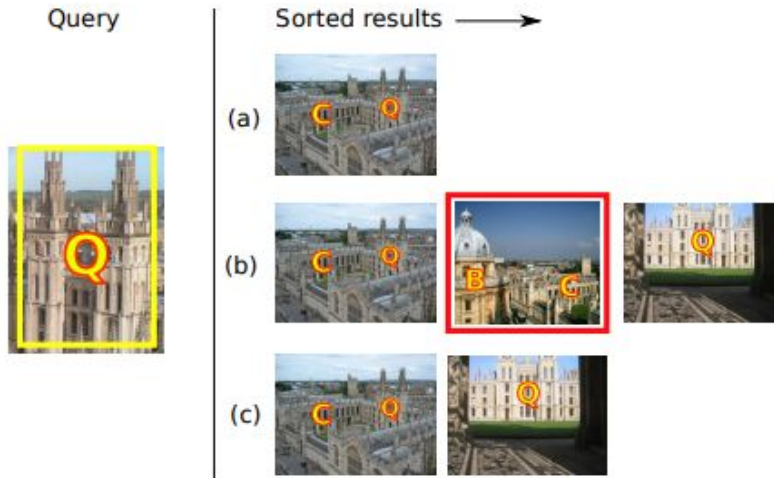




# Tăng cường đặc trưng cơ sở dữ liệu

Database-side feature  
augmentation

# Hiệu suất truy xuất của các phương pháp tăng cường



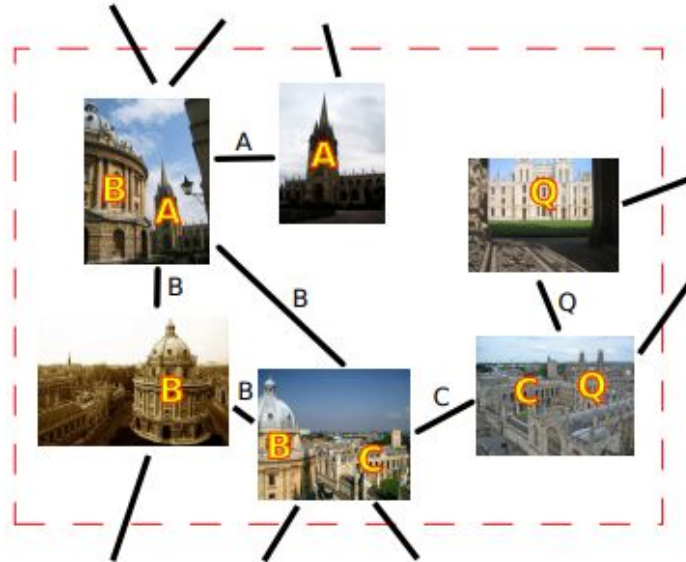
Đối tượng được truy vấn được tô sáng màu vàng trên hình ảnh ngoài cùng bên trái.

(a) Kết quả truy vấn tf-idf khi không dùng thông tin đồ thị ảnh, một hình ảnh trong thách thức của hệ thống truy vấn đã không được truy vấn

(b) Ảnh được truy vấn bằng việc sử dụng phương pháp của Turcot and Lowe (2009) và đồ thị được chỉ ra trong hình: Độ sót được tăng cường nhưng độ chính xác giảm trong các trường hợp dương tính giả (được đánh dấu màu đỏ)

(c) Phương pháp của chúng tôi cho thấy việc độ sót tăng lên trong khi duy trì độ chính xác cao vì hình ảnh chỉ được tăng cường với các từ trực quan từ các khu vực lân cận có liên quan.

Trừ khi hai hình ảnh gần giống nhau, một số lượng lớn các từ bổ sung sẽ thực sự không được nhìn thấy





## Chi tiết triển khai

Tác giả sử dụng cách tiếp cận của Philbin và Zisserman (2008) để xây dựng một biểu đồ hình ảnh phù hợp trong một bộ dữ liệu ngoại tuyến. Mỗi hình ảnh trong bộ dữ liệu được sử dụng làm truy vấn trong một hệ thống truy xuất đối tượng tiêu chuẩn (Philbin et al., 2007) và một cạnh được xây dựng cho mỗi hình ảnh được xác minh theo không gian. Một phương pháp xây dựng đồ thị thay thế sử dụng băm (Chum và Matas, 2010a) có thể được sử dụng cho các bộ dữ liệu quy mô rất lớn trong đó truy vấn sử dụng mỗi hình ảnh lần lượt là không thực tế. Khi xây dựng biểu đồ, chúng tôi không bao gồm các hình ảnh truy vấn được sử dụng để đánh giá bộ dữ liệu để mô phỏng kịch bản thực tế trong đó hình ảnh truy vấn không được biết đến vào thời điểm tiền xử lý.