

# Personalized Annotation for Photos with Visual Instance Search

Bao Truong, Thuyen V. Phan, Vinh-Tiep Nguyen, and Minh-Triet Tran

Faculty of Information Technology  
University of Science, VNU - HCM  
{tmbao, ppthuyen}@apcs.vn, {nvtiep, tmtriet}@fit.hcmus.edu.vn

**Abstract.** Emotional and memorable moments are usually kept and shared on different online services such as Facebook, Flickr, Instagram, and Google Photos. As a result, one of users' practical needs is to have their photos annotated automatically, especially with personalized tags. This motivates the authors to propose a system that can suggest personalized annotations for a photo uploaded to online services. Our system provides 2 major features. First, the system automatically recommends personalized annotations for newly uploaded photos based on visually similar photos uploaded in the past. Second, our system propagates manual annotations of users to other similar photos existed in their albums. To evaluate the performance of our system, we use the Oxford 5K Building Dataset and our own dataset consisting of personal photos collected from Facebook. Our systems achieves the mean Average Precision of 0.844 and 0.749 respectively on these two datasets. This demonstrates that our proposed solution can be potentially integrated as a useful utility or extension for online photo sharing services.

## 1 Introduction

In our lives, there are many emotional and memorable moments that worth keeping and sharing with others. Therefore, services allowing users to upload and share their personal photos are always ones of many notable products of different companies such as Facebook, Flickr, Instagram, and Google Photos. This shows that sharing photos is one of greatest demands of users on the Internet.

Online photo services usually allow users to attach some memos to their photos as well as to search their photos more easily using text queries. Currently, the most common way for users to do so is to tag their photos manually which consumes a lot of time and effort. There are also some proposed methods [3, 5] and smart systems which are able to automatically identify noticeable landmarks or locations related to the photos such as Google Photos and Flickr. However, these automated annotation systems suggest tags that are identical for all users and thus do not reflect one's own memories, feelings, or characteristics. For example, these systems would recommend phrases like "Eiffel Tower", "a dog", or "a cat" rather than "where I first met my lover" or the name of your pet.

Therefore, it is necessary to automatically tag users' photos with personalized captions corresponding to their memory and personal characteristics.

In this paper, we propose a system that can suggest appropriate annotations for each photo uploaded by users using Visual Instance Search. In our system, users can assign their personalized annotations for some photos as initial examples, then, the system automatically propagates these annotations to other existed photos in their collection based on visual similarities among the photos. For each uploaded photo, the system bases on visual similarities between that photo and already-annotated photos of the corresponding user to propose a list of suitable annotations for the new uploaded photo in descending order of similarity. Then, the user can choose to approve reasonable annotation for the uploaded photo. In addition, if a user uploads more than one photo and change the annotations, the system has more samples for reference and thus, it tends to better adapt to user's interests. As a result, our system is not only able to recommend proper annotations which are unique for each user but also to interactively and incrementally learn and adapt as users change annotations.

Since the problem of retrieving similar images in a collection corresponding to a single image has been developed for years, there are many different approaches to the problem. One of them is template matching method, i.e. a technique for finding small parts of an image which match a template image [2, 16, 4]. Another popular technique is to evaluate the similarity of two images by comparing some regions which appear to be critical parts of the images, namely features matching [1, 17, 23]. In this paper, the authors develop our own Visual Instance Search framework using Bag-of-Words (BoW) model. In Bag-of-Words model, each image is represented as a histogram of pre-trained visual words (codebook). Since Bag-of-Words allows parts of a query image to appear in a flexible way in the result images, it is a potential approach that is widely used in many Visual Search systems.

Together with the exponential increasing of the number of uploaded images, the system faces lots of difficulty adapting those new images. Since re-training the codebook requires changing Bag-of-Words vectors of users' existing images and is also computationally expensive, the authors propose to use a fixed codebook trained with different types of features (e.g. vehicles, animals, buildings...) and use it universally. Because of the varieties of those different features, it is appropriate to compute and represent any new images' Bag-of-Words vectors without changing the codebook. Therefore, we train our codebook on Oxford Building Dataset and use this codebook for our system.

Our main contributions in this paper are as follows:

- We propose the idea and realize the system that can recommend annotation for photos with visual instance search.
- Our system allows recommended annotation to be personalized and to vary from user to user.
- Our system is interactively user adaptive, i.e. the more a user annotates his/her photos via our system, the more accurate the recommended annotations are.

The rest of this paper is organized as follows. In section 2, we review the background and related works in image retrieval and image classification. Detailed steps of the automatic annotation system and how we use the BoW model is described in section 3. Section 4 contains our experiment result. Conclusion is presented in 5

## 2 Background & Related works

There are many approaches to build an Image Information Retrieval System. Some methods aim at high precision, i.e achieve high quality of top retrieved results, while others focus on high recall, i.e retrieve all positive results. Among them, the first effective and scalable method is Bag-of-Words, Sivic and Zisserman [20], which is inspired by the correspondence algorithm using in text retrieval. Before going into details of BoW model in subsection 2.2, we will first introduce some different methods for image retrieval problem in subsection 2.1.

### 2.1 Different approaches for image retrieval problem

One of many popular methods is histogram comparisons which compares 2 different images based on their color histograms. Some early works of this approach using a cross-bin matching cost for histogram comparison can be found in [19, 24, 12]. In [12], Peleg et al. represent images as sets of pebbles after normalization. The similarity score is then computed as the matching cost of two sets of pebbles based on their distances.

Another well-known technique is template matching, i.e. seeking a given pattern in a image by comparing to candidate regions of the same size in the target image. By consider both the pattern and candidate regions as a length- $N$  vector, we can compare these two vectors by using different kinds of distance metrics, one such metric is the Minkowski distance [10]. The major disadvantage of 2 listed methods is that they require the query and target images to share a similar stationary interrelation, which means that components of the given image are not allowed to change freely in a certain extent. Bag-of-Words, the method that is discussed in this paper, is another approach that can tolerate the flexibility in structures of the object and thus, have a wider variation of application in many problems.

### 2.2 Bag-of-Words

Since Bag-of-Words is originally a text retrieval algorithm, we will first introduce some backgrounds about BoW in text retrieval problem in subsection 2.2 before discussing using BoW in image retrieval in subsection 2.2.

**Bag-of-Words in text retrieval** In text retrieval, a text is represented as a histogram of words, also known as BoW [6]. This scheme is called term frequency

weighting as the value of each histogram bin is equal to the number of times the word appears in the document. Moreover, some words are less informative than others since those words appear in almost every document. Therefore, we need a weighting scheme that address this problem. Such weighting scheme is called inverse document frequency (idf) and is formulated as  $\log(N_D/N_i)$ , where  $N_D$  is the number of documents in the collection and  $N_i$  is the number of documents which contains word  $i$ . The overall BoW representation is thus weighted by multiplying the term frequency (tf) with the inverse document frequency (idf) giving rise to the tf-idf weighting [6]. In addition, extremely frequent words, “stop words”, can be removed entirely in order to reduce storage requirements and query time.

**Bag-of-Words in image retrieval** When applying BoW to image retrieval, a major obstacle is the fact that text documents are naturally broken into words by spaces, dots, hyphens, or commas. In contrast, there is no such separator in images. Therefore, the concept of “visual word” is introduced where each visual word is represented as a cluster obtained using k-means on the local descriptor vectors [20].

The bigger the vocabulary size is, the more different the visual words are. Hence, the vocabulary helps us distinguish the images more effectively. Nonetheless, with bigger vocabulary size, slightly different descriptors can be assigned to different visual words thus not contributing to the similarity of the respective images and causing a drop in performance examined in [9, 18, 15]. Philbin et al. [15] suggests “soft assign” method where each descriptor is assigned to multiple nearest visual words instead of using “hard assignment”, i.e only assign a local descriptor to only one nearest visual word. Despite its effectiveness, this method also significantly costs more storage and time.

### 3 Proposed System

In this section, we will show how our system can learn to annotate different photos and briefly describe main steps in our BoW model.

#### 3.1 How our system learn and automatically annotate new photos

Figure 1 illustrates the overview of our proposed system to automatically recommend personalized annotations for newly uploaded photos. Firstly, users simply use their smartphones cameras to capture scenes or objects in real life such as books, dogs or buildings. These photos are then sent to our server for processing and our server will return the list of visual similar photos. Additionally, each photo attaches a list of annotations and these possible personalized annotations will be re-ranked and sent to the users. The users will have chances to review and approve these personalized annotations before sharing these photos to social networks such as Facebook, Flickr, or Google Plus along with the approved personalized tags.

Figure 2 shows how our system learns to annotate photo from samples provided by users. Firstly, users manually choose suitable tags for some photos and these photos along with the tags are then sent to the server. Subsequently, our server process will identify and recommend the users to also apply these changes to visual similar photos in their albums. The users will have the rights to approve before these changes take effect in the database. From this point of time, our system will automatically annotate new photos based on these new configurations.

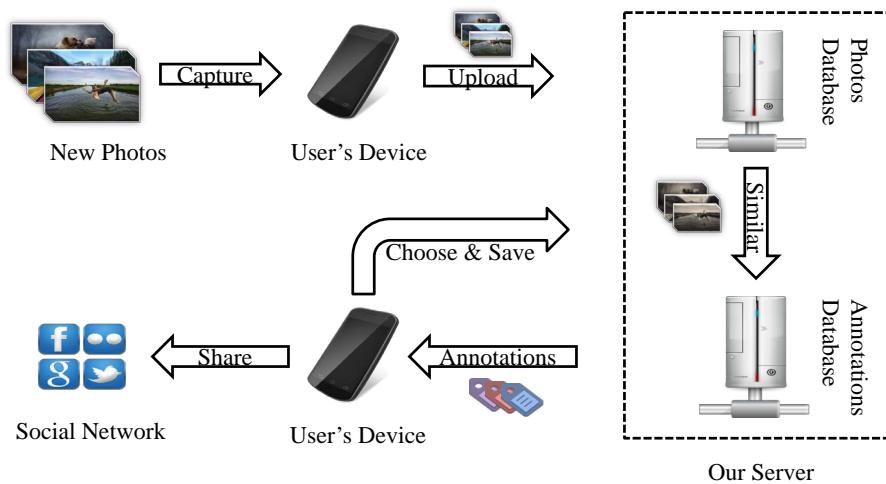


Fig. 1: Overview of our proposed system to automatically recommend personalized tags.

### 3.2 Visual Instance Search Method

**Feature Extraction** To detect and extract features from images, there are many methods that have been proposed (Harris-Affine, Hessian-Affine detectors [8], Maximally stable extremal region (MSER) detector [7], Edge-based region detector [21], Intensity extrema-based region detector [22] ...). The authors choose to use Hessian-Affine detector, for detecting and extracting features from images. In our version of BoW model, we use Perd'och's implementation of SIFT detector, which is shown to perform best on Oxford Building Dataset [13].

**Dictionary Building** Treating each descriptor as an individual visual words in the dictionary results in a worthless waste of resources and time. In order to overcome this obstacle, the authors therefore build the dictionary by considering

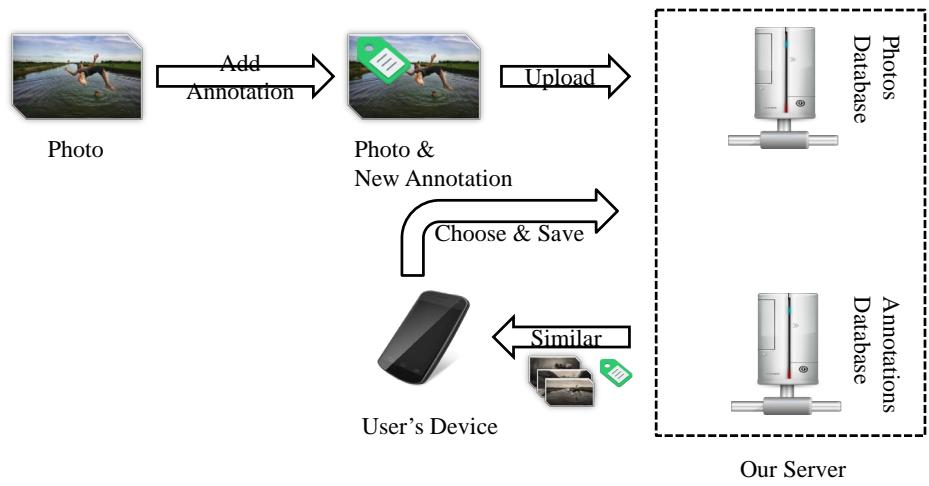


Fig. 2: Overview on how our system learn to annotate photo from samples provided by users.

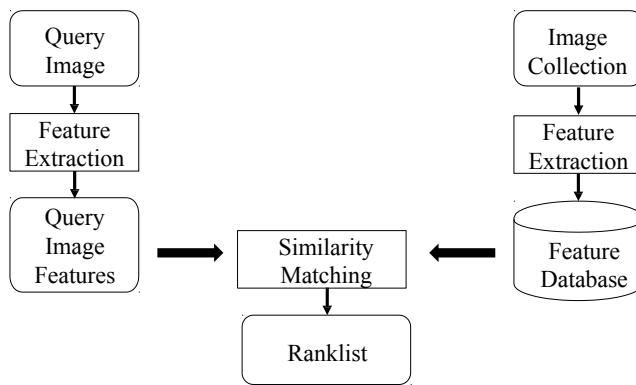


Fig. 3: How an Image Retrieval System works

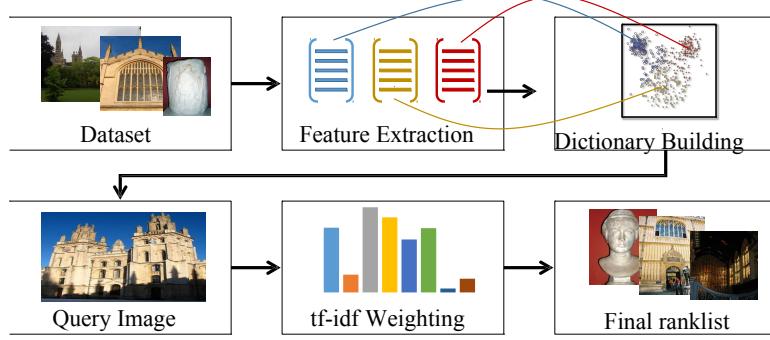


Fig. 4: Proposed framework

some similar descriptors as one. In other words, all descriptor vectors are divided into  $k$  clusters, each representing a visual word. There are many algorithms that are proposed to solve this kind of problem. However, the authors use the approximate k-means (AKM). AKM is proposed by Philbin et al. [14]. Comparing to the original k-means, AKM can reduce the majority amount of time taken by exact nearest neighbors computation but only gives slightly different result. Also, in [14], Philbin et al. shows that using 1M dictionary size would have the best performance on the Oxford Building 5K Dataset [11].

**Quantization** Subsequently, each 128-dimension SIFT descriptor needs to be mapped into the dictionary. Commonly, each descriptor is assigned into the nearest word in the dictionary. Thus, when two descriptors are assigned to different words, they are considered as totally different. In practice, this hard assignment leads to errors due to variability in descriptor (e.g. image noise, varying scene illumination, instability in the feature detection process ...) [15]. In order to handling this problem, the authors use soft assignment instead of hard assignment. In particular, each 128-dimension SIFT descriptor is reduced to a  $k$ -dimension vector of their  $k$  nearest visual words in the dictionary. Each of these  $k$  nearest cluster is assigned with weights calculated from the formula proposed by Sivic et al. [15],  $weight = \exp(-\frac{d^2}{2\delta^2})$ , where  $d$  is the distance from the cluster center to descriptor point. Then, by adding all these weights to their corresponding bins, we will have the BoW representation of an image.

In this work,  $k$  and  $\delta^2$  are chosen to be 3 and 6250, respectively.

**tf-idf Weighting Scheme** As mentioned in section 2, tf-idf is a popular weighting scheme that is used by almost any BoW model. In this section, the authors will show how this scheme is applied to our system.

For a term  $t_i$  in a particular document  $d_j$ , its term frequency  $tf_{i,j}$  is defined as follow:

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \quad (1)$$

Where  $n_{i,j}$  is the number of occurrences of the considered term  $t_i$  in the document  $d_j$ . The denominator is the sum of the number of occurrences of all the terms in document  $d_j$ .

The inverse document frequency  $idf_i$  of a term  $t_i$  is computed by the following formula:

$$idf_i = \log \frac{|D|}{|\{j : t_i \in d_j\}|} \quad (2)$$

Where,  $|D|$  is the total number of documents in the corpus,  $|\{j : t_i \in d_j\}|$  is the number of documents where the term  $t_i$  appears, i.e.  $n_{i,j} \neq 0$

The tf-idf weight of a term  $t_i$  in a document  $d_j$  is then calculated as the product of tf and idf:

$$tfidf_{i,j} = tf_{i,j} \times idf_i \quad (3)$$

The tf-idf weight is then used to compute the similarity score between an image  $d_i$  and a query  $q$ :

$$s_{d_i,q} = \mathbf{tfidf}_i \cdot \mathbf{tfidf}_q = \sum_{j=1}^{|T|} tfidf_{i,j} \times tfidf_{q,j} \quad (4)$$

Finally, by sorting the list of images corresponding to their similarity score with a query, we achieve the raw ranked list of this query which is then used for the Spatial Rerank step.

## 4 Experiment & Result

In this section, first, we present our experiment result on Oxford 5K Building Dataset to prove that our BoW implementation can achieve good enough performance on standard benchmark. The experiment shows that our version of BoW achieves the mean average precision of 0.844 on Oxford 5K Building Dataset with nearly one second average time for each query. This dataset was constructed by Philbin et al. in 2007 [14]. It consists of 5,062 images of resolution  $1024 \times 768$  belongs to 11 different Oxford buildings. Images for each building are collected from Flickr by searching using text queries. Along with the dataset, there are also 55 queries along with their ground-truth, 5 for each landmark. The ground truth of 55 queries are manually constructed. For each query, images are classified into 4 groups: (1) *Good*: the building appears apparently, (2) *OK*: more than 25% of the building is present, (3) *Bad*: the building is not shown up, and (4) *Junk*: less than 25% of the building is captured. The reason why the authors use this dataset is because of its popularity, it is used by many previous works in this field. Thus, we can easily compare our systems with those previous works.

Secondly, we also present and illustrate several typical scenarios of our automatic annotation system with the dataset consisting of our personal photos taken from Facebook. This dataset includes 5 different classes corresponding with 5 social events that are personally annotated. There are 2 classes that share a common annotation. Photos in each class share some particular attributes such as background, mascots, logos. As a result, whenever users create or edit the annotation of these common objects, other photos in the same class can also be tagged similarly thanks to these mutual attributes. The details of these 5 classes in the dataset are described below:

1. #APCS\_Party: Photos taken at a party of my university. Photos in this class contain nearly the same group of people and have similar background and decorators on the stage.
2. #First\_time\_in\_Singapore: These photos are taken at the Merlion in Singapore. They all contain the merlion statue.
3. #Hoi\_An\_with\_family: These photos are taken at Hoi An town in Vietnam with one of the authors' family. The people appearing in them and the background are their common attributes.
4. #My\_favoriate\_competition: These are taken at multiple times I have taken part in the ACM-ICPC, a really famous collegiate programming competition. The mutual characteristic of these photos is the logo of the competition.
5. #My\_firstRegional: These photos are taken at my ICPC regional contest in Phuket, Thailand. The photos all accommodate the mascot of the competition.

We then performed experiment on 5 different queries corresponding to 5 different classes. These queries and some sample result are given in Fig. 5. In the experiment, each query also takes our system nearly one second on average and the mean average precision over 5 queries is 0.749. The detail result is shown in table 1.

APCS query	Singapore query	Thailand query	ACMlogo query	HoiAn query	<b>mAP</b>
0.690	0.775	0.798	0.754	0.728	<b>0.749</b>

Table 1: Experiment on personal dataset

## 5 Conclusion

In this paper, we have proposed our idea to use Visual Instance Search to create a system that can help user in 2 different tasks: automatically suggested personalized annotations for uploaded photos and propagate users' annotations for their photos to similar images. To realize this, we build our system based

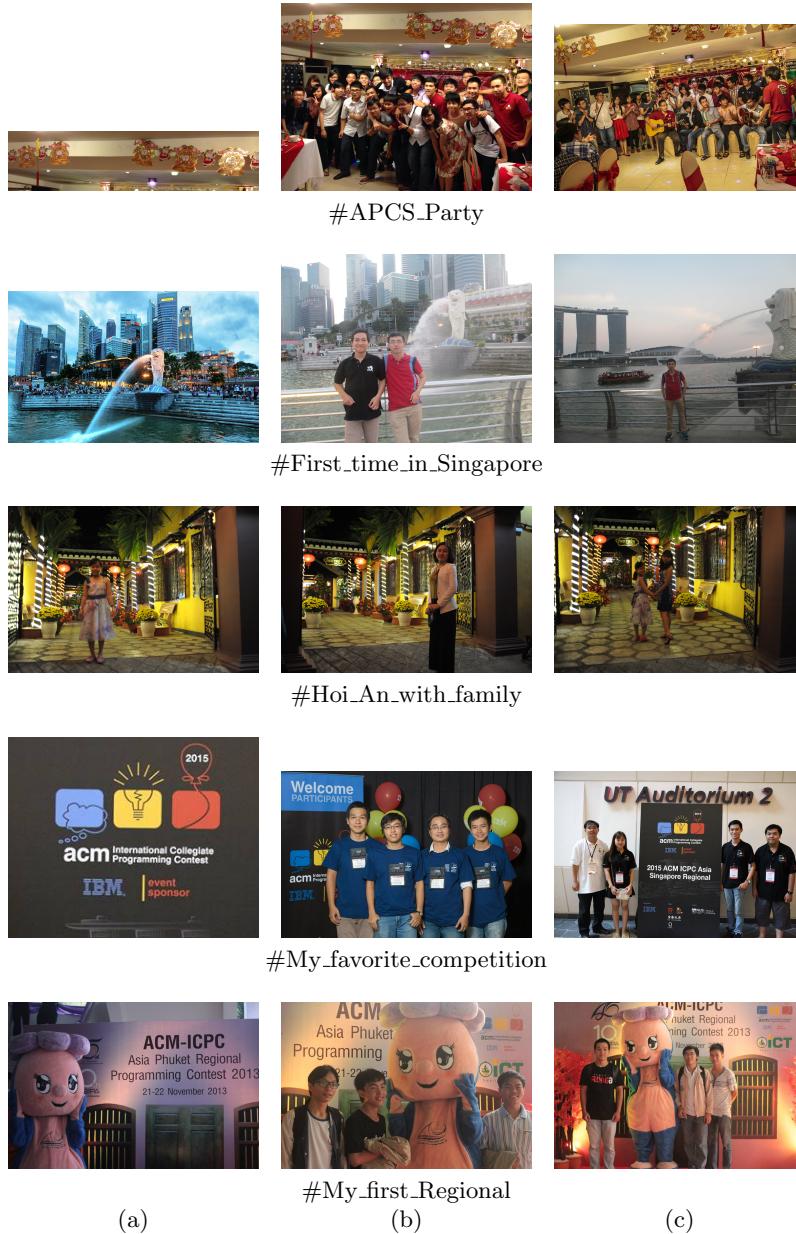


Fig. 5: Our personal dataset. Column (a) shows 5 queries of 5 classes in the dataset. Column (b) and (c) are some examples in the returned result of the queries.

on Bag-of-Words model. To evaluate the performance of the system, we have used Oxford 5K Building Dataset, a really popular benchmark for Visual Instance Search task. In addition, we have also experimented and illustrated our systems with some scenarios taken from personal photos along with their tags on Facebook. Our system has achieves the mAP of 0.844 and 0.749 respectively on these 2 datasets with the processing time less than 1 second for each query. In the future, the authors believe that the system can further be developed to become a valuable extension for online photo sharing services.

## References

1. Belongie, S., Carson, C., Greenspan, H., Malik, J.: Color- and texture-based image segmentation using em and its application to content-based image retrieval. In: Computer Vision, 1998. Sixth International Conference on. pp. 675–682 (Jan 1998)
2. Brunelli, R.: Template Matching Techniques in Computer Vision: Theory and Practice. Wiley (2009)
3. Chen, M., Zheng, A., Weinberger, K.Q.: Fast image tagging. In: Dasgupta, S., Mcallester, D. (eds.) Proceedings of the 30th International Conference on Machine Learning (ICML-13). vol. 28, pp. 1274–1282. JMLR Workshop and Conference Proceedings (May 2013), <http://jmlr.org/proceedings/papers/v28/chen13e.pdf>
4. Gharavi-Alkhansari, M.: A fast globally optimal algorithm for template matching using low-resolution pruning. Image Processing, IEEE Transactions on 10(4), 526–533 (Apr 2001)
5. Lan, T., Mori, G.: A max-margin rifled independence model for image tag ranking. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2013)
6. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press (2008)
7. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide baseline stereo from maximally stable extremal regions. In: Rosin, P.L., Marshall, A.D. (eds.) BMVC. British Machine Vision Association (2002)
8. Mikolajczyk, K., Schmid, C.: Scale & affine invariant interest point detectors. International Journal of Computer Vision 60(1), 63–86 (2004)
9. Nister, D., Stewenius, H.: Scalable recognition with a vocabulary tree. In: Proc. CVPR (2006)
10. Ouyang, W., Tombari, F., Mattoccia, S., Di Stefano, L., Cham, W.K.: Performance evaluation of full search equivalent pattern matching algorithms. Pattern Analysis and Machine Intelligence, IEEE Transactions on 34(1), 127–143 (Jan 2012)
11. <http://www.robots.ox.ac.uk/~vgg/data/oxbuildings/>
12. Peleg, S., Werman, M., Rom, H.: A unified approach to the change of resolution: space and gray-level. Pattern Analysis and Machine Intelligence, IEEE Transactions on 11(7), 739–742 (Jul 1989)
13. Perd'och, M., Chum, O., Matas, J.: Efficient representation of local geometry for large scale object retrieval. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. pp. 9–16 (June 2009)
14. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: Proc. CVPR (2007)
15. Philbin, J., Isard, M., Sivic, J., Zisserman, A.: Lost in quantization: Improving particular object retrieval in large scale image databases. In: Proc. CVPR (2008)

16. Rosenfeld, A., Vanderburg, G.J.: Coarse-fine template matching. *Systems, Man and Cybernetics, IEEE Transactions on* 7(2), 104–107 (Feb 1977)
17. Rubner, Y., Tomasi, C., Guibas, L.: The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision* 40(2), 99–121 (2000)
18. Schindler, G., Brown, M., Szeliski, R.: City-scale location recognition. In: Proc. CVPR (2007)
19. Shen, H.C., Wong, A.K.: Generalized texture representation and metric. *Computer Vision, Graphics, and Image Processing* 23(2), 187 – 206 (1983), <http://www.sciencedirect.com/science/article/pii/0734189X83901123>
20. Sivic, J., Zisserman, A.: Video google: A text retrieval approach to object matching in videos. In: Proc. ICCV (2003)
21. Tuytelaars, T., Gool, L.V.: Content-based image retrieval based on local, affinely invariant regions. In: In Int. Conf. on Visual Information Systems. pp. 493–500 (1999)
22. Tuytelaars, T., Gool, L.V.: Wide baseline stereo matching based on local, affinely invariant regions. In: In Proc. BMVC. pp. 412–425 (2000)
23. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on. vol. 1, pp. I-511–I-518 vol.1 (2001)
24. Werman, M., Peleg, S., Rosenfeld, A.: A distance metric for multidimensional histograms. *Computer Vision, Graphics, and Image Processing* 32(3), 328 – 336 (1985), <http://www.sciencedirect.com/science/article/pii/0734189X85900556>